

Министерство науки и высшего образования Российской Федерации  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО  
ITMO University

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА  
GRADUATION THESIS

Разработка алгоритма генерации диалогов с неигровыми персонажами на основе  
моделируемых языковых групп

Обучающийся / Student Ярославцев Никита Сергеевич

Факультет/институт/кластер/ Faculty/Institute/Cluster школа разработки видеоигр

Группа/Group J4222

Направление подготовки/ Subject area 09.04.03 Прикладная информатика

Образовательная программа / Educational program Технологии разработки  
компьютерных игр 2022

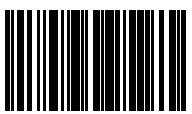
Язык реализации ОП / Language of the educational program Русский

Квалификация/ Degree level Магистр

Руководитель ВКР/ Thesis supervisor Карсаков Андрей Сергеевич, кандидат технических  
наук, Университет ИТМО, школа разработки видеоигр, доцент (квалификационная  
категория "ординарный доцент")

Консультант/ Consultant Нахабин Дмитрий Игоревич, ШРВ, преподаватель  
(квалификационная категория "преподаватель практики"), неосн по совм.

Обучающийся/Student

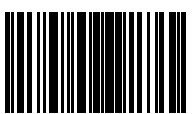
Документ подписан	
Ярославцев Никита Сергеевич	
22.05.2024	

(эл. подпись/ signature)

Ярославцев  
Никита  
Сергеевич

(Фамилия И.О./ name  
and surname)

Руководитель ВКР/  
Thesis supervisor

Документ подписан	
Карсаков Андрей Сергеевич	
22.05.2024	

(эл. подпись/ signature)

Карсаков  
Андрей  
Сергеевич

(Фамилия И.О./ name  
and surname)

**Министерство науки и высшего образования Российской Федерации  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО  
ITMO University**

**ЗАДАНИЕ НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ /  
OBJECTIVES FOR A GRADUATION THESIS**

**Обучающийся / Student** Ярославцев Никита Сергеевич

**Факультет/институт/кластер/ Faculty/Institute/Cluster** школа разработки видеоигр

**Группа/Group** J4222

**Направление подготовки/ Subject area** 09.04.03 Прикладная информатика

**Образовательная программа / Educational program** Технологии разработки компьютерных игр 2022

**Язык реализации ОП / Language of the educational program** Русский

**Квалификация/ Degree level** Магистр

**Тема ВКР/ Thesis topic** Разработка алгоритма генерации диалогов с неигровыми персонажами на основе моделируемых языковых групп

**Руководитель ВКР/ Thesis supervisor** Карсаков Андрей Сергеевич, кандидат технических наук, Университет ИТМО, школа разработки видеоигр, доцент (квалификационная категория "ординарный доцент")

**Консультант/ Consultant** Нахабин Дмитрий Игоревич, ШРВ, преподаватель (квалификационная категория "преподаватель практики"), неосн по совм.

**Характеристика темы ВКР / Description of thesis subject (topic)**

**Тема в области фундаментальных исследований / Subject of fundamental research:** нет / not

**Тема в области прикладных исследований / Subject of applied research:** да / yes

**Основные вопросы, подлежащие разработке / Key issues to be analyzed**

Цель работы: Разработка алгоритма генерации диалогов с неигровыми персонажами на основе моделируемых языковых групп

Задачи работы:

1. Проведение анализа предметной области для изучения современных возможностей в области генерации текста и обработки естественного языка
2. Формирование теоретической модели с учётом проведённого анализа предметной области
3. Практическая имплементация алгоритма
4. Разработка пользовательского интерфейса
5. Оценка результатов работы алгоритма
6. Демонстрация примеров использования алгоритма
7. Описание дальнейших планов развития алгоритма
8. Подготовка отчётных материалов

Содержание работы: введение, анализ предметной области, теоретическая модель алгоритма, практическая имплементация алгоритма, экспериментальные исследования алгоритма, заключение, список использованных источников

Рекомендуемые материалы:

1. Ramalho L. Fluent python. – " O'Reilly Media, Inc.", 2022.
2. Horstmann C. S., Necaie R. D. Python for everyone. – John Wiley & Sons, 2019.
3. Hyman L. M., Plank F. (ed.). Phonological typology. – Walter de Gruyter GmbH & Co KG, 2018. – Т. 23.
4. Bonvillain N. Language, culture, and communication: The meaning of messages. – Rowman & Littlefield, 2019.
5. Khyani D. et al. An interpretation of lemmatization and stemming in natural language processing // Journal of University of Shanghai for Science and Technology. – 2021. – Т. 22. – №. 10. – С. 350-357.
6. Vissio N. C., Zakharov V. A Disambiguator for Pymorphy2 Morphological Analyzer // IMS. – 2021. – С. 81-88.
7. Levshina N. Token-based typology and word order entropy: A study based on Universal Dependencies // Linguistic Typology. – 2019. – Т. 23. – №. 3. – С. 533-572.
8. Wang M., Hu F. The application of nltk library for python natural language processing in corpus research //Theory and Practice in Language Studies. – 2021. – Т. 11. – №. 9. – С. 1041-1049.
9. Gilda S. Notice of Violation of IEEE Publication Principles: Evaluating machine learning algorithms for fake news detection // 2017 IEEE 15th student conference on research and development (SCORED). – IEEE, 2017. – С. 110-115.
10. Janda H. K. et al. Syntactic, semantic and sentiment analysis: The joint effect on automated essay evaluation //IEEE Access. – 2019. – Т. 7. – С. 108486-108503.

**Форма представления материалов ВКР / Format(s) of thesis materials:**

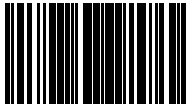
Текст ВКР в формате PDF, презентация

**Дата выдачи задания / Assignment issued on:** 22.02.2024

**Срок представления готовой ВКР / Deadline for final edition of the thesis** 24.05.2024

**СОГЛАСОВАНО / AGREED:**


Руководитель ВКР/  
Thesis supervisor

Документ подписан	
Карсаков Андрей Сергеевич	
08.05.2024	

(эл. подпись)

Карсаков  
Андрей  
Сергеевич

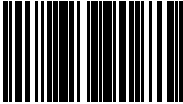
Задание принял к  
исполнению/ Objectives  
assumed BY

Документ подписан	
Ярославцев Никита Сергеевич	
09.05.2024	

Ярославцев  
Никита  
Сергеевич

(эл. подпись)

Руководитель ОП/ Head  
of educational program

Документ подписан	
Карсаков Андрей Сергеевич	
22.05.2024	

(эл. подпись)

Карсаков  
Андрей  
Сергеевич

---

**Министерство науки и высшего образования Российской Федерации  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО  
ITMO University**

**АННОТАЦИЯ  
ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЫ  
SUMMARY OF A GRADUATION THESIS**

**Обучающийся / Student** Ярославцев Никита Сергеевич  
**Факультет/институт/кластер/ Faculty/Institute/Cluster** школа разработки видеоигр  
**Группа/Group** J4222  
**Направление подготовки/ Subject area** 09.04.03 Прикладная информатика  
**Образовательная программа / Educational program** Технологии разработки компьютерных игр 2022  
**Язык реализации ОП / Language of the educational program** Русский  
**Квалификация/ Degree level** Магистр  
**Тема ВКР/ Thesis topic** Разработка алгоритма генерации диалогов с неигровыми персонажами на основе моделируемых языковых групп  
**Руководитель ВКР/ Thesis supervisor** Карсаков Андрей Сергеевич, кандидат технических наук, Университет ИТМО, школа разработки видеоигр, доцент (квалификационная категория "ординарный доцент")  
**Консультант/ Consultant** Нахабин Дмитрий Игоревич, ШРВ, преподаватель (квалификационная категория "преподаватель практики"), неосн по совм.

**ХАРАКТЕРИСТИКА ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЫ  
DESCRIPTION OF THE GRADUATION THESIS**

**Цель исследования / Research goal**

Разработка алгоритма генерации диалогов с неигровыми персонажами на основе моделируемых языковых групп.

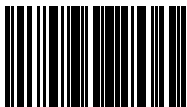
**Задачи, решаемые в ВКР / Research tasks**

1. Проведение анализа предметной области для изучения современных возможностей в области генерации текста и обработки естественного языка. 2. Формирование теоретической модели с учётом проведённого анализа предметной области. 3. Практическая имплементация алгоритма на основе сформированной теоретической модели. 4. Разработка пользовательского интерфейса для удобного взаимодействия с алгоритмом. 5. Оценка результатов работы алгоритма. 6. Демонстрация примеров использования алгоритма с помощью разработанного пользовательского интерфейса. 7. Описание дальнейших планов развития алгоритма.

**Краткая характеристика полученных результатов / Short summary of results/findings**

1. Для достижения поставленной цели работы одних больших языковых моделей оказалось недостаточно. 2. Самым трудозатратным получился метод замены лексических единиц из-за большого количества операций на каждый токен. 3. Итоговое время алгоритма можно считать приемлемым. 4. Имеется возможность в оптимизации алгоритма, а также расширении функционала.

Обучающийся/Student


Документ подписан	
Ярославцев Никита Сергеевич	
22.05.2024	

(эл. подпись/ signature)

Ярославцев  
Никита  
Сергеевич

(Фамилия И.О./ name  
and surname)

Руководитель ВКР/  
Thesis supervisor

Документ подписан	
Карсаков Андрей Сергеевич	
22.05.2024	

(эл. подпись/ signature)

Карсаков  
Андрей  
Сергеевич

(Фамилия И.О./ name  
and surname)

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ .....	10
1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ .....	12
1.1 Примеры использования генерации диалогов в компьютерных играх ..	12
1.1.1 «Façade» .....	12
1.1.2 «AI Dungeon» .....	13
1.2 Большие языковые модели .....	15
1.2.1 «GPT-4» .....	15
1.2.2 «YandexGPT 3» .....	16
1.2.3 «RuGPT-3» .....	16
1.2.4 Возможности больших языковых моделей для изменения корректности согласования и построения текста на примере «ChatGPT»	18
1.3 Увеличение точности языковых моделей .....	21
1.3.1 Предварительное обучение языковой модели .....	21
1.3.2 Тонкая настройка языковой модели .....	22
1.4 Техники обработки естественного языка .....	24
1.4.1 Токенизация .....	25
1.4.2 Удаление стоп-слов из текста .....	27
1.4.3 Лемматизация и стемминг .....	28
1.4.4 Синтаксический анализ текста .....	31
1.5 Библиотеки для обработки естественного языка .....	32
1.5.1 Пакет библиотек NLTK .....	33
1.5.2 Библиотека Rymorphy2 .....	34
1.5.3 Библиотека SpaCy .....	35
1.6 Типология .....	36
1.6.1 Фонетико-фонологическая типология .....	36
1.6.1.1 Вокаличность и консонантность .....	37
1.6.1.2 Бинарная оппозиция .....	39
1.6.2 Морфологическая типология .....	39
1.6.2.1 Изолирующие языки .....	40

1.6.2.2	Агглютинативные языки .....	40
1.6.2.3	Флективные языки .....	41
1.6.2.4	Инкорпорирующие языки .....	42
1.6.3	Просодическая типология.....	42
1.6.4	Синтаксическая типология .....	43
1.6.4.1	Эргативные языки.....	43
1.6.4.2	Номинативные языки.....	44
1.6.4.3	Порядок слов .....	44
1.6.5	Лексическая типология .....	45
1.7	Конкретизация задачи .....	46
2	ТЕОРЕТИЧЕСКАЯ МОДЕЛЬ АЛГОРИТМА .....	48
2.1	Требования к системе.....	48
2.2	Ограничения системы .....	48
2.3	Формализация модели.....	49
3	ПРАКТИЧЕСКАЯ ИМПЛЕМЕНТАЦИЯ АЛГОРИТМА .....	52
3.1	Реализация отдельных задач.....	52
3.1.1	Реализация замены маркированных звуков .....	52
3.1.2	Реализация преобразования флективного текста в определённую степень изолированности .....	53
3.1.3	Реализация преобразования шести основных видов порядка слов ..	53
3.1.4	Реализация замены лексических единиц .....	54
3.2	Объединение реализованных задач в цельный алгоритм .....	55
3.3	Получение данных о проведённых преобразованиях .....	55
3.3.1	Данные об изменённом порядке слов .....	56
3.3.2	Данные о заменённых лексических единицах .....	57
3.3.3	Данные об изолированных слова .....	57
3.3.4	Данные о заменённых маркированных звуках .....	57
3.4	Разработка пользовательского интерфейса .....	57
3.4.1	Вкладка для создания и сохранения моделируемой языковой группы .....	58
3.4.2	Вкладка для демонстрации преобразования текста .....	59



3.4.3 Вкладка для общения с персонажем сервиса «Character.ai» .....	61
4 ЭКСПЕРИМЕНТАЛЬНЫЕ ИССЛЕДОВАНИЯ АЛГОРИТМА .....	63
4.1 Оценка результатов работы алгоритма .....	63
4.1.1 Метод изменения порядка слов .....	63
4.1.1.1 Точность метода изменения порядка слов .....	63
4.1.1.2 Время выполнения метода изменения порядка слов .....	64
4.1.2 Метод замены лексических единиц .....	64
4.1.2.1 Точность метода замены лексических единиц .....	65
4.1.2.2 Время выполнения метода замены лексических единиц .....	65
4.1.3 Метод преобразования текста в степень изолированности .....	67
4.1.3.1 Точность метода преобразования текста в степень изолированности .....	67
4.1.3.2 Время выполнения метода преобразования текста в степень изолированности .....	68
4.1.4 Метод замены маркированных звуков .....	69
4.1.4.1 Точность метода замены маркированных звуков .....	69
4.1.4.2 Время выполнения метода замены маркированных звуков .....	70
4.1.5 Сравнение всех методов преобразования .....	71
4.1.6 Суммарная работа алгоритма .....	71
4.1.6.1 Итоговая точность работы алгоритма .....	72
4.1.6.2 Итоговое время выполнения алгоритма .....	72
4.2 Демонстрация примеров использования алгоритма .....	73
4.3 Описание дальнейших планов развития алгоритма .....	75
ЗАКЛЮЧЕНИЕ .....	76
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ .....	78

## **ВВЕДЕНИЕ**

Диалоги являются одной из важнейших и неотъемлемых частей большого количества жанров компьютерных игр. Технологии в областях генерации текста и обработки естественного языка существенно улучшились, следовательно появилась довольно значительная перспектива в использовании данных технологий в процессе создания диалогов для компьютерных игр.

С широким распространением проблем обработки естественного языка и генерации текста появилось множество технологий, выводящих возможность процедурной генерации контента высокого качества на общедоступный уровень. Несмотря на колоссальный прогресс в решении этих проблем в последний годы, используются далеко не все возможности, которые предлагают современные технологии.

Использование техник обработки естественного языка позволяет создавать более реалистичные и естественные диалоги между персонажами, что делает игровой мир более убедительным и захватывающим для игроков. Обработка естественного языка может помочь в создании персонализированных диалоговых вариантов, учитывающих предыдущие действия и выборы игрока, что делает игровой процесс более адаптивным и вовлекающим [1]. Это способствует более глубокой погруженности игрока в игровой мир и сюжет. Использование обработки естественного языка позволяет в некоторой степени расширить возможности искусственного интеллекта, что в свою очередь приводит к более эффективному взаимодействию с игроками, а также более гибкому реагированию на различные игровые ситуации, которые не были прописаны разработчиками заранее. Даже если взять в пример текстовую компьютерную игру «AI Dungeon» [2], полностью построенную на процедурной генерации диалогов с помощью большой языковой модели, в ней имеется все тот же недостаток. Все неигровые персонажи, встречающиеся в игре, разговаривают без акцента, а также не имеют

практически никаких проблем с построением речи не зависимо от их расы. Иными словами, все неигровые персонажи являются носителями языка. Хотя есть некоторые ограничения и сложности в использовании обработки естественного языка для формирования диалогов в компьютерных играх, его применение может значительно улучшить игровой опыт и сделать игры более захватывающими для игроков.

Цель работы: разработка алгоритма генерации диалогов с неигровыми персонажами на основе моделируемых языковых групп.

Задачи:

- 1) Проведение анализа предметной области для изучения современных возможностей в области генерации текста и обработки естественного языка.
- 2) Формирование теоретической модели с учётом проведённого анализа предметной области.
- 3) Практическая имплементация алгоритма.
- 4) Разработка пользовательского интерфейса.
- 5) Оценка результатов работы алгоритма.
- 6) Демонстрация примеров использования алгоритма.
- 7) Описание дальнейших планов развития алгоритма.

## 1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

### 1.1 Примеры использования генерации диалогов в компьютерных играх

#### 1.1.1 «Façade»

Одним из примеров использования генерации диалогов в компьютерных играх является компьютерная игра «Façade», разработанная независимой студией «Procedural Arts». Ключевыми особенностями компьютерной игры являются продвинутый на момент выхода игры искусственный интеллект, а также способ взаимодействия игрока с неигровыми персонажами посредством текстового ввода фраз с помощью клавиатуры с последующей озвученной реакцией неигровыми персонажами (Рисунок 1.1).



Рисунок 1.1 – Рендеринг неигровых персонажей Трип и Грейс в реальном времени с текстом, набранным игроком [1]

Введенный игроком текст является не просто дополнительной возможностью увидеть реакцию неигровых персонажей на случайные фразы, а представляет из себя возможность непосредственного воздействия на последующие события игры и на взаимоотношения между всеми персонажами. Введенный игроком текст преобразовывается в одну из множества реакций (Рисунок 1.2).

- |                    |                   |                   |
|--------------------|-------------------|-------------------|
| • agree            | • referTo <topic> | • intimate        |
| • disagree         | • praise          | • judgment        |
| • positive exclaim | • ally            | • suggestion      |
| • negative exclaim | • criticize light | • misc-custom     |
| • express happy    | • criticize harsh | • manipulation    |
| • express laugh    | • oppose          | • jokeTestLimits  |
| • express sad      | • flirt           | • inappropriate   |
| • express angry    | • pacify          | • hug             |
| • maybeUnsure      | • provoke         | • comfort         |
| • dontUnderstand   | • greet           | • kiss            |
| • thank            | • goodbye         | • physicallyFavor |
| • apologize        | • getAttention    | • wanderAway      |

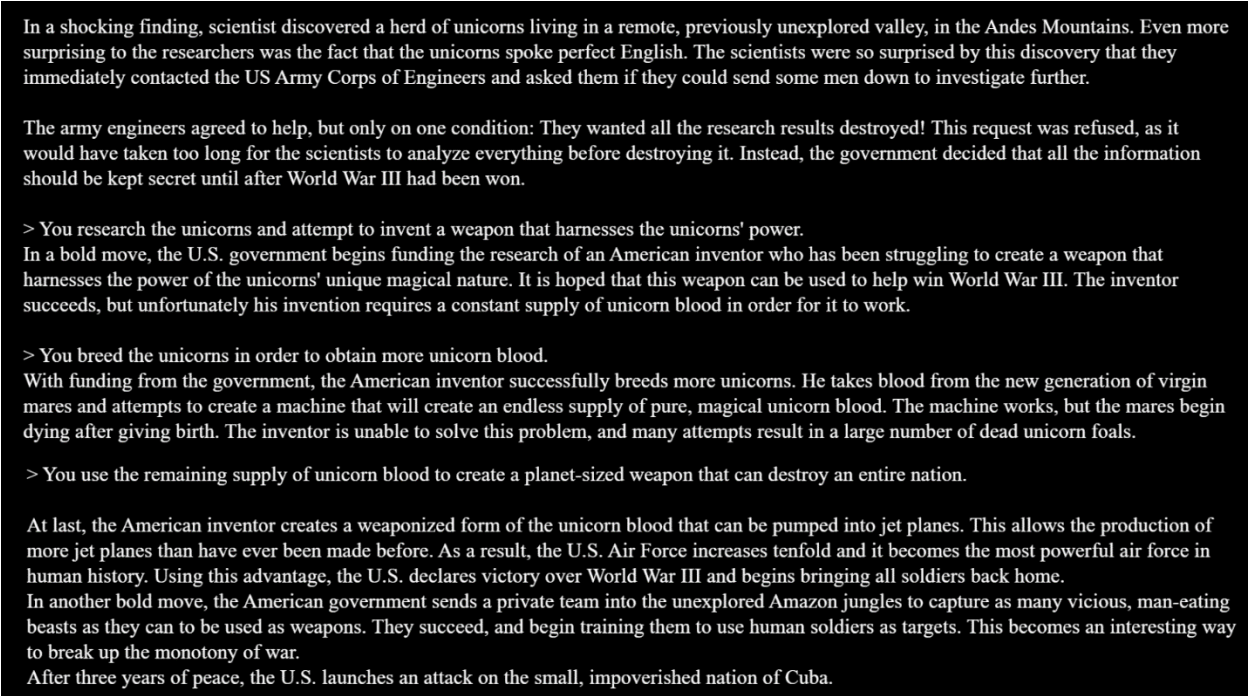
Рисунок 1.2 – Список всех возможных реакций неигровых персонажей на действия игрока [1]

На момент выхода компьютерной игры «Façade» вопросы обработки естественного языка не имели такого широкого распространения, поэтому отдельного внимания достойна система «Natural Language Understanding». Система «NLU» была специально разработана создателями компьютерной игры для обработки естественного языка и последующего преобразования в шаблоны поведения неигровыми персонажами [1]. С учетом того, что разработчики были одними из первых, кто на серьезном уровне попытался внедрить технологии обработки естественного языка в игровой процесс, существует множество вариантов, при которых искусственный интеллект генерирует неоднозначные и противоречащие выходные данные.

### 1.1.2 «AI Dungeon»

Одним из самых современных и продвинутых вариантов в сфере генерации диалогов является компьютерная игра в преимущественно текстовом формате «AI Dungeon» [2]. Последняя версия данной компьютерной игры создана с использованием большой языковой модели

«GPT-4 Turbo». Несмотря на то, что данная компьютерная игра обладает игровым процессом с бесконечным и хаотичным чтением без конечной цели, за счет использования большой языковой модели у любого игрока возникает уникальный опыт из-за полной процедурной генерации игрового контента. Автор статьи «Playing With Unicorns: AI Dungeon and Citizen NLP» привёл пример игрового процесса (Рисунок 1.3)



In a shocking finding, scientist discovered a herd of unicorns living in a remote, previously unexplored valley, in the Andes Mountains. Even more surprising to the researchers was the fact that the unicorns spoke perfect English. The scientists were so surprised by this discovery that they immediately contacted the US Army Corps of Engineers and asked them if they could send some men down to investigate further.

The army engineers agreed to help, but only on one condition: They wanted all the research results destroyed! This request was refused, as it would have taken too long for the scientists to analyze everything before destroying it. Instead, the government decided that all the information should be kept secret until after World War III had been won.

> You research the unicorns and attempt to invent a weapon that harnesses the unicorns' power.  
In a bold move, the U.S. government begins funding the research of an American inventor who has been struggling to create a weapon that harnesses the power of the unicorns' unique magical nature. It is hoped that this weapon can be used to help win World War III. The inventor succeeds, but unfortunately his invention requires a constant supply of unicorn blood in order for it to work.

> You breed the unicorns in order to obtain more unicorn blood.  
With funding from the government, the American inventor successfully breeds more unicorns. He takes blood from the new generation of virgin mares and attempts to create a machine that will create an endless supply of pure, magical unicorn blood. The machine works, but the mares begin dying after giving birth. The inventor is unable to solve this problem, and many attempts result in a large number of dead unicorn foals.

> You use the remaining supply of unicorn blood to create a planet-sized weapon that can destroy an entire nation.

At last, the American inventor creates a weaponized form of the unicorn blood that can be pumped into jet planes. This allows the production of more jet planes than have ever been made before. As a result, the U.S. Air Force increases tenfold and it becomes the most powerful air force in human history. Using this advantage, the U.S. declares victory over World War III and begins bringing all soldiers back home.

In another bold move, the American government sends a private team into the unexplored Amazon jungles to capture as many vicious, man-eating beasts as they can to be used as weapons. They succeed, and begin training them to use human soldiers as targets. This becomes an interesting way to break up the monotony of war.

After three years of peace, the U.S. launches an attack on the small, impoverished nation of Cuba.

Рисунок 1.3 – Пример игрового процесса «AI Dungeon» («Пользовательское приключение») [2]

Если опустить все положительные черты игрового процесса «AI Dungeon», можно выявить явные ограничения и недостатки. Данные ограничения и недостатки скорее присущи большим языковым моделям, нежели конкретному игровому процессу некоторой компьютерной игры. Под самое заметное ограничение попадает малый объем памяти о происходящих событиях у большой языковой модели, что приводит к тому, что события, которые происходили пару сцен назад, могут быть напрочь забыты неигровыми персонажами, что может привести к выходу из погружения. Явным же недостатком можно посчитать большие трудности при попытке изменить главную ветку повествования, что приводит к тому, что любые действия игрока рано или поздно сводятся к определенным событиям.

Хоть компьютерная игра «AI Dungeon» обладает некоторыми недостатками, которые могут испортить игровой опыт, она потенциально является одним из самых прогрессивных и успешных попыток использовать современные технологии для процедурной генерации контента.

## **1.2 Большие языковые модели**

На сегодняшний день одной из самых передовых возможностей в области обработки естественного языка, и, в частности, генерации диалогов, являются большие языковые модели. Большие языковые модели представляют из себя комплексные программы, использующие передовые алгоритмы глубокого обучения, которые способны качественно обрабатывать естественный язык, генерировать текст, а также исполнять множество других задач, которые не относятся напрямую к данному исследованию.

С учетом обширных возможностей больших языковых моделей, многие разработчики компьютерных игр стремятся использовать их для упрощения создания игрового контента, или же непосредственно внедрять функционал больших языковых моделей в игровой процесс для процедурной генерации контента.

### **1.2.1 «GPT-4»**

С выходом большой языковой модели «GPT-3», которая была представлена компанией «OpenAI» в 2020 году, частота использования больших языковых моделей, а также количество областей их применения начали увеличиваться [3].

GPT-4 является большой мультимодальной моделью, которая способна обрабатывать изображения и текст, а также генерировать текстовые выходные данные [4]. Авторы статьи «GPT-4 Technical Report» проанализировали и сравнили «GPT-4» с другими моделями. Авторы статьи пришли к выводу, что «GPT-4» превосходит текущие крупные языковые модели в различных задачах обработки естественного языка и опережает большинство современных систем, часто требующих тонкой настройки под

конкретную задачу. Также было обнаружено, что улучшения, наблюдаемые в мультимодальной модели «GPT-4», обычно измеряемые на английском языке, также проявляются на многих других языках.

### 1.2.2 «YandexGPT 3»

В качестве альтернативы «GPT-4», поддерживающей русский язык, в 2024 году вышла «YandexGPT 3» от компании «Яндекс». «YandexGPT 3» представляет собой новое поколение генеративных моделей от компании «Яндекс», способное эффективно решать сложные задачи и строго следовать заданному формату ответа [5]. Это делает ее особенно подходящей для использования в ИТ-продуктах и других коммерческих целях. «YandexGPT 3» не только более интеллектуальна по сравнению с предыдущими моделями, но и превосходит как их, так и «ChatGPT-3.5 Turbo» в ответах на узкие классы запросов, критически важных для бизнеса. Для достижения этого разработчиками было проведено дополнительное обучение модели на задачах, связанных с рабочими процессами.

### 1.2.3 «RuGPT-3»

В большинстве случаев большие языковые модели поддерживают конкретный язык, с помощью которого отправляется запрос, и на котором возвращается последующий ответ. В отличие от стандартного функционала языковых моделей в статье «Building a Bilingual QA-system with ruGPT-3» был представлен подход к межъязыковому трансферному обучению английскому и русскому языкам [6]. Реализованный метод позволяет отправлять запрос на русском языке и получать ответ на английском, а также наоборот. Подход подразумевает использование генеративной модели, которая предварительно обучалась на текстах «Википедии» на обоих языках, а затем была тонко настроена. Результаты представленного подхода показывают, что точно настроенная модель демонстрирует возможность взаимодействия на двух языках и может генерировать ответы, близкие к правильным.



Metrics for ruGPT-3 XL	Ru-Ru	En-Ru	En-En	Ru-En
Lemmas intersection ratio over 0.7	0.39	<b>0.12</b>	0.09	<b>0.1</b>
Named entities ratio	0.83	0.75	0.52	0.51
Named entities precision	0.52	0.2	0.32	0.27
Fuzz ratio	0.59	<b>0.28</b>	0.24	<b>0.27</b>
Exact match	0.32	<b>0.1</b>	0.08	<b>0.1</b>
METEOR mean	0.48	0.21	0.22	0.22
METEOR ratio over 0.7	0.36	0.11	0.1	0.11
ROUGE-L mean	0.57	<b>0.28</b>	0.25	<b>0.27</b>
ROUGE-L ratio over 0.7	0.46	0.16	0.12	0.14
SBERT cosine distance 0.4 or less	0.56	0.59	0.31	0.35
SBERT cosine distance 0.1 or less	0.41	0.19	0.13	0.15

Рисунок 1.4 – Оценочные метрики межъязыкового подхода на примере языковой модели «ruGPT-3» [6]

Помимо оценочных метрик, результаты экспериментов оставили два варианта для интерпретации: научилась ли модель переводить тексты с одного языка на другой или она научилась сравнивать два похожих источника на разных языках? Чтобы проверить эту гипотезу, модели были заданы вопросы, не относящиеся к ресурсу «Википедия», на основе новостных абзацев [6]. В результате проверки гипотезы языковая модель вполне успешно перевела запрос и сгенерировала правильный (Рисунок 1.5).

**paragraph:** Как передает RegioNews, об этом сообщил первый заместитель директора Одесского припортового завода Николай Шуриков. «Завод уже в процессе остановки. На сегодня причины две — переполненный состав карбамида и заоблачная цена сырья (природного газа)», — сообщил он. По его словам, «окно возможностей» для стабилизации работы завода закрылось. Как отмечает Шуриков, пока остается лишь шанс провести приватизацию завода. Напомним, Одесский припортовый завод выпускал свою продукцию с 1978 года. Специализация предприятия — производство аммиака, карбамида и другой химической продукции. Предприятие возобновило свою работу два года назад после длительного простоя.

**question:** What are two main reasons of the plant shutdown?

**ruGPT-3 answer:** odessa carbamid and over-billing for raw materials

**gold answer:** overfilled composition of urea and sky-high price of raw materials (natural gas)

Рисунок 1.5 – Проверка гипотезы об успешном обучении «ruGPT-3»  
межъязыковым подходом [6]

1.2.4 Возможности больших языковых моделей для изменения  
корректности согласования и построения текста на примере «ChatGPT»

Большие языковые модели по большей части используются для генерации качественного и связного текста, но поскольку большие языковые модели являются не просто генераторами текста, их также можно использовать в целях синтаксического, морфологического и семантического взаимодействия с текстом [3].

Воспользовавшись Telegram чат-ботом, использующим большую языковую модель ChatGPT, можно продемонстрировать условно-встроенные возможности в большие языковые модели для изменения корректности согласования и построения текста.

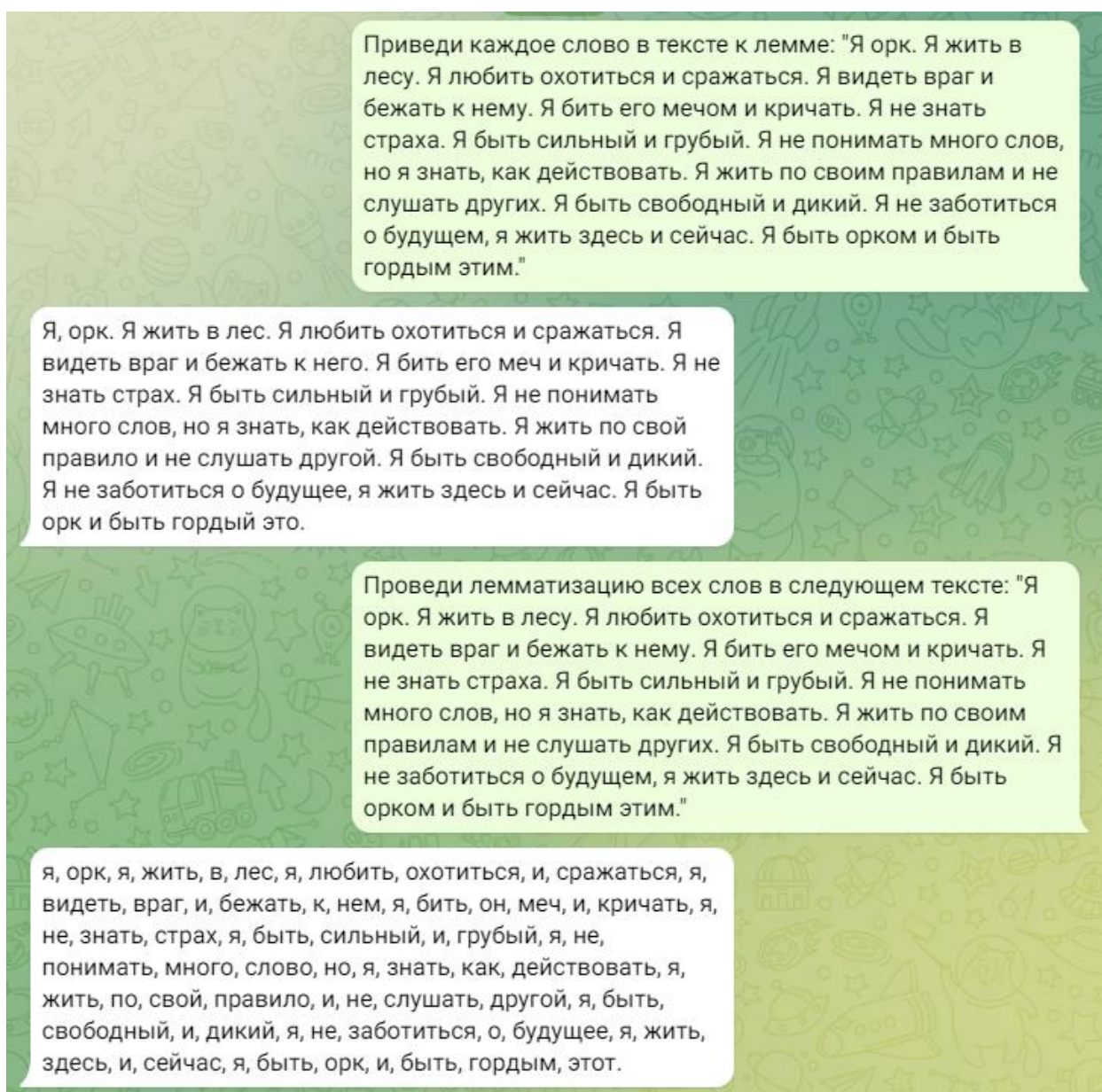


Рисунок 1.6 – Лемматизация всех слов в тексте с помощью «ChatGPT»

Языковая модель способна к лемматизации лексических единиц (Рисунок 1.6). Также видно, что языковая модель очень чувствительна к отправляемым запросам. Оба запроса имеют одну и ту же задачу – лемматизация текста, но выходные данные кардинально различаются из-за немного разных формулировок запроса. Данная проблема широко распространена среди большинства больших языковых моделей, поэтому проблема релевантна не только по отношению к «ChatGPT».

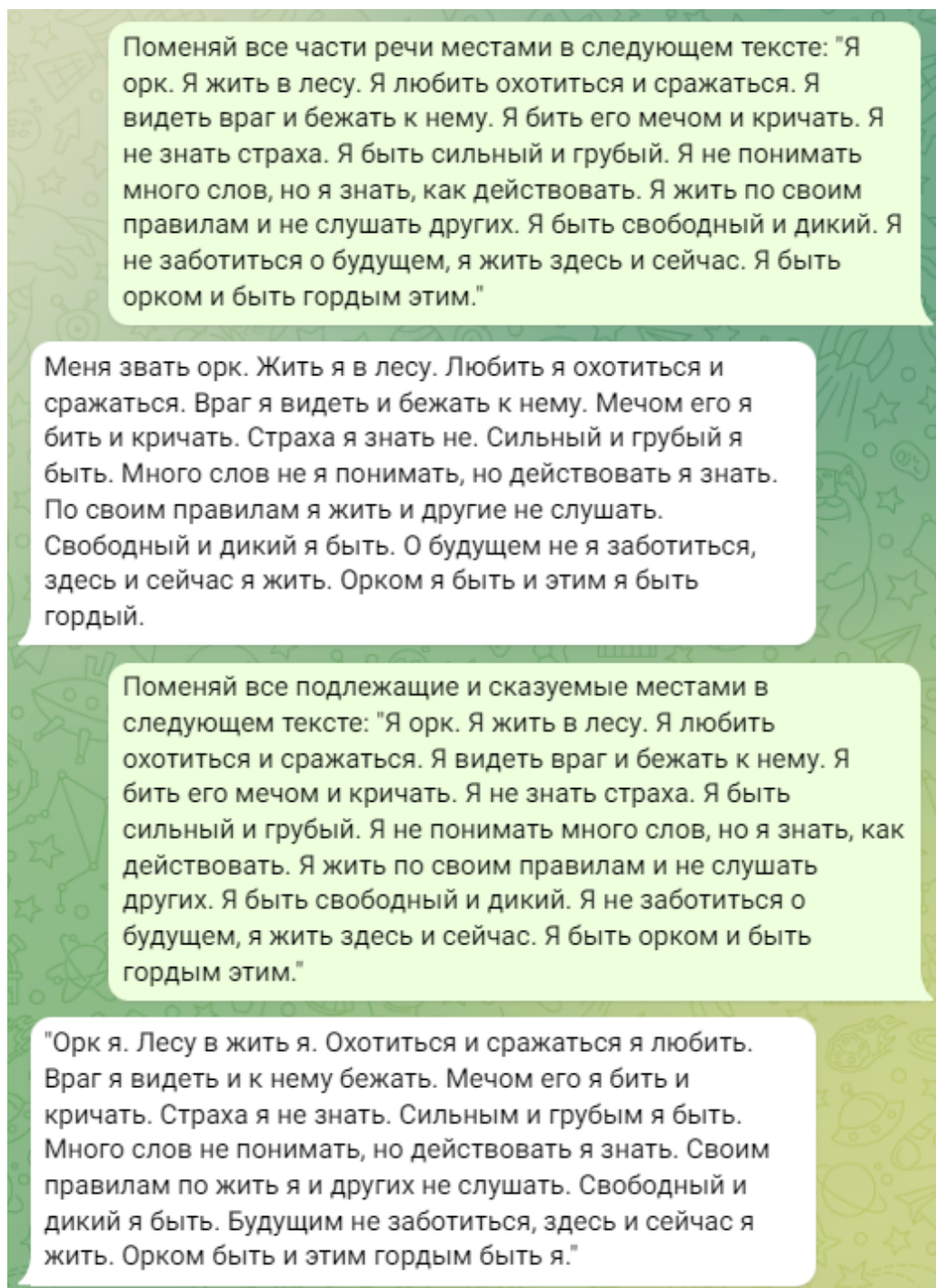


Рисунок 1.7 – Изменение последовательности слов в тексте с помощью «ChatGPT»

Языковая модель способна к изменению последовательности слов в тексте в некоторой степени (Рисунок 1.7). На данном примере также заметна проблема с чувствительностью к отправляемым запросам и последующим выходным данным.

Может показаться, что возможностей больших языковых моделей достаточно для решения задачи изменения корректности согласования и



построения текста. Главная проблема кроется в огромной чувствительности больших языковых моделей к отправляемым запросам, а также к тому, что нет гарантий, что выходные данные будут в необходимом формате.

### **1.3 Увеличение точности языковых моделей**

Повышение корректности согласования и построения речи является одной из главных задач технологий, связанных с обработкой естественного языка и генерацией текста.

#### **1.3.1 Предварительное обучение языковой модели**

Предварительно обученные языковые модели (PRLM) добились большого успеха в широком спектре задач обработки естественного языка благодаря способности универсального языкового представления, полученной в результате самостоятельного обучения на больших данных.

Автора статьи «Dialogue-adaptive language model pre-training from quality estimation» представляют подход «Dialogue-adaptive pre-training objectives» (DAPO), для повышения качества согласованности, специфичности и информативности текста. В качестве основы для предварительной подготовки модели авторы создают обучающий набор с помощью двух методов [7]:

1) Искажение контекста, ориентированное на согласованность, включая упорядочивание высказываний, вставку и замену, чтобы способствовать усваиванию согласованности моделью внутри диалоговых контекстов.

2) Побуждение модели в измерении качества синтезированных данных для предварительной подготовки, адаптированных к диалогу, с учетом их специфичности и информативности.

В своем подходе авторы статьи использовали заранее обученную языковую модель «ELECTRA» (Рисунок 1.8).

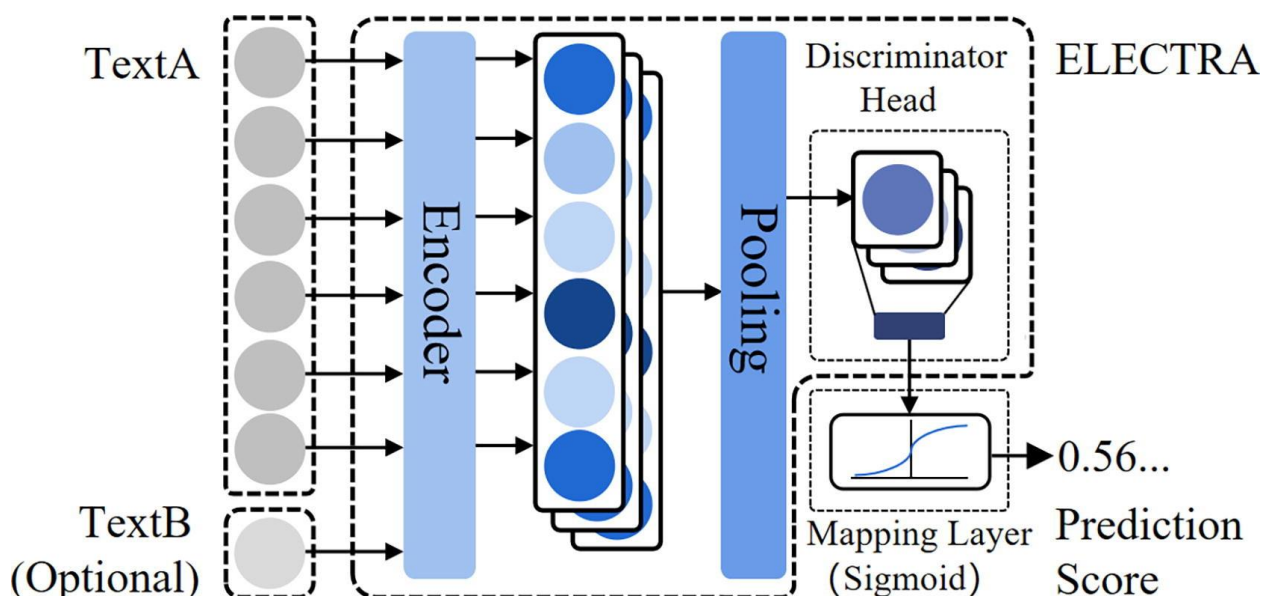


Рисунок 1.8 – Структура модели «ELECTRA» [8]

### 1.3.2 Тонкая настройка языковой модели

Одним из самых распространенных подходов в повышении качества текста и его обработки является тонкая настройка языковой модели [9, 10]. Под тонкой настройкой чаще всего подразумевается обучение языковой модели на узкоспециализированных наборах данных. В большинстве случаев тонкая настройка производится на последних этапах обучения языковой модели.

Авторы статьи «Fine-Tuning GPT-3 for Russian Text Summarization» использовали тонкую настройку языковой модели в целях улучшения качества краткого изложения русских текстов (Рисунок 1.9).

human summary	ruGPT3 summary
Депутаты Госдумы одобрили в третьем чтении законопроект о пенсионном обеспечении граждан. Согласно документу, с 1 января 2020 года пенсии будут проиндексированы на 6,7%. Это позволит повысить размер пенсий, повысить их размер и размер страховых взносов.	Работающие пенсионеры дождались индексации. Это следует из уже одобренных в первом чтении поправок в Конституцию. Так что работающие пенсионеры смогут получить замороженную с 2016 года прибавку к пенсии. В России их насчитывается более девяти миллионов. Эксперты называют это восстановлением справедливости.
20 января состоится заседание Совета Госдумы, который рассмотрит вопрос о включении в повестку нижней палаты парламента на 21 января создание должности зампреда Совета безопасности РФ, которую займет бывший премьер-министр Дмитрий Медведев. Некоторые эксперты трактуют это как понижение, однако на деле Медведев фактически станет правой рукой Владимира Путина в сфере безопасности и вторым лицом одного из важнейших совещательных органов России.	Дмитрий Медведев назначен на должность председателя совета безопасности при президенте РФ. Соответствующий указ был подписан президентом РФ Владимиром Путиным в ходе послания Федеральному собранию в декабре прошлого года. До этого Медведев занимал пост главы администрации президента.
Владимир Путин позвонил премьер-министру Израиля Биньямину Нетаньяху и поздравил его с юбилеем. В ходе диалога они обсудили ситуацию в Сирии и некоторые актуальные двусторонние вопросы.	Премьер-министр Израиля Биньямин Нетаньяху в ходе встречи с президентом России Владимиром Путиным заявил о готовности к диалогу с Россией по вопросу о транзите гуманитарных грузов через сирийскую территорию. Израильский премьер также выразил надежду на диалог с Москвой

Рисунок 1.9 – Примеры хорошей производительности модели [9]

Автор статьи «Тонкая настройка BERT и GPT-3 для решения задачи генерации русскоязычных новостей» использовал тонкую настройку для узкоспециализированной задачи, а именно повышение качества генерируемого текста в сфере русскоязычных новостей [10]. Применение тонкой настройки к языковой модели дало хорошие результаты. Базовая модель плохо справлялась с генерацией новостей, но тонкая настройка произвела нужный эффект (Рисунок 1.10).

Название метрики	base model		fine-tuned model	
	nucleus	beam	nucleus	beam
MSJ2	0.326	0.373	0.412	0.678
MSJ3	0.162	0.155	0.250	0.386
MSJ4	0.057	0.034	0.148	0.169
BLEU	0.705	0.793	0.914	0.928
Self-BLEU	0.927	0.916	0.948	0.879
SUN2	0.068	0.093	0.073	0.104
SUN3	0.153	0.292	0.159	0.238
SUN4	0.474	0.564	0.512	0.633
Human	0.16	0.27	2.25	3.54

Рисунок 1.10 – Оценка генерации с «ruGPT-3» [10]

Несмотря на то, тонкая настройка является популярным и полезным подходом в обучении языковых моделей, она подходит не во всех случаях. В случаях широко специализированных нужд тонкая настройка может оказаться весьма бесполезным подходом, и не оказать необходимого положительного эффекта.

#### 1.4 Техники обработки естественного языка

В области обработки естественного языка существует множество полезных техник, которые позволяют решать различные задачи [11]:

- 1) Токенизация по предложениям и токенизация по словам;
- 2) Синтаксический, семантический, морфологический анализы;
- 3) Лемматизация и стемминг;
- 4) Анализ эмоциональной окраски текста;
- 5) Определение языка;
- 6) Извлечение ключевых слов из текста;
- 7) Формирование краткого пересказа текста;



- 8) Машинный перевод;
- 9) Аннотация документа;
- 10) Информационный поиск;
- 11) Дедубликация.

Потенциально необходимыми для решения задачи изменения корректности согласования и построения текста были выявлены следующие техники:

- 1) Токенизация [12, 13];
- 2) Удаление стоп-слов из текста [12, 14];
- 3) Лемматизация и стемминг [12, 15, 16];
- 4) Синтаксический анализ [17];
- 5) Морфологический анализ слов [20].

#### 1.4.1 Токенизация

Токенизация представляет из себя процесс разделения письменного языка на отдельные компоненты [12]. Токенизация может воспроизводиться по предложениям и по словам (Рисунок 1.12). При токенизации по предложениям, как можно понять по названию, идёт разделение текста на цельные предложения. При токенизации по словам всё немного иначе. Разделение идёт не только на слова, но и на знаки препинания, следовательно токеном может являться не только слово, но и знак препинания. В целом концепция токенизации может показаться уж больно простой, но на самом деле не всё так тривиально, поэтому существует большое количество направлений в развитии токенизации (Рисунок 1.11). Например, в английском и русском языках в текстах часто присутствуют сокращения со знаком точки, из-за которых не получится просто так провести токенизацию. В решении этой проблемы могут помочь таблицы сокращений.

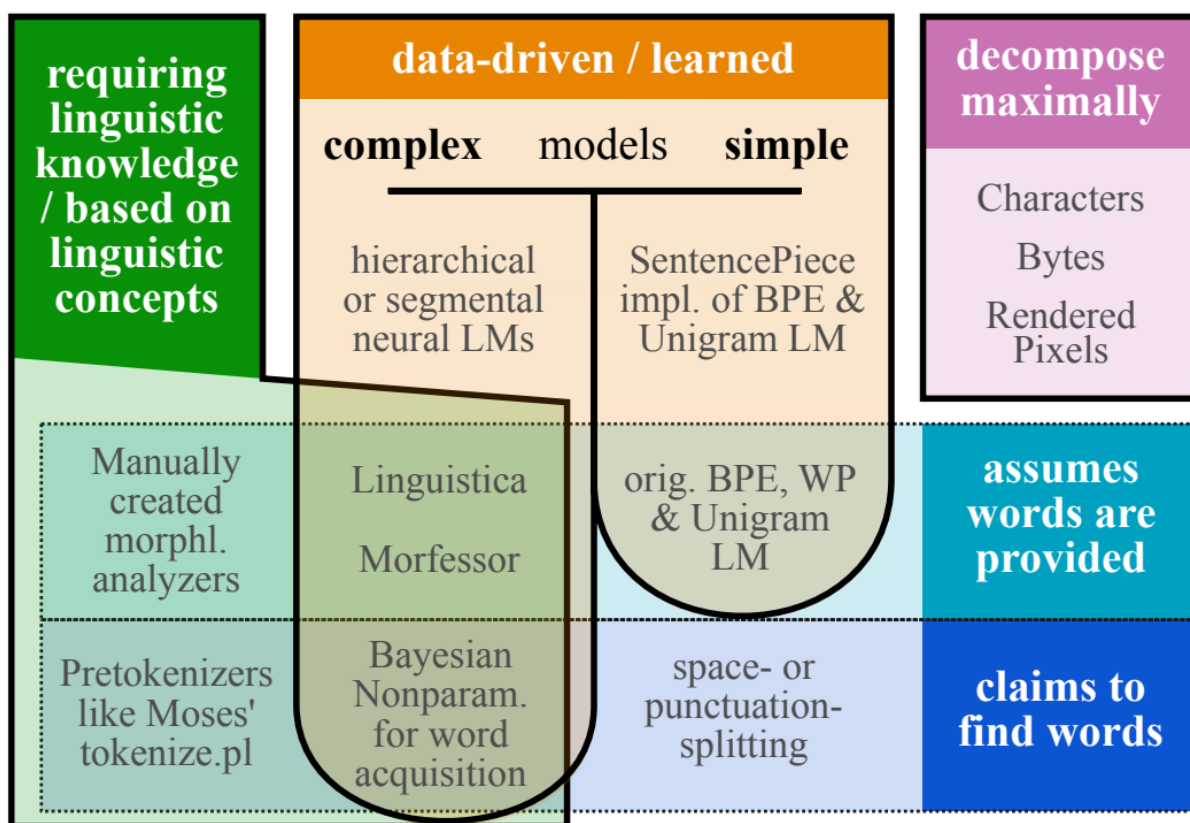


Рисунок 1.11 – Таксономия алгоритмов сегментации, токенизации и направлений исследований [13]

Несмотря на то, что токенизация развивается с самого начала формирования технологий в области обработки естественного языка, не существует идеального метода для воспроизведения токенизации. Все варианты имеют свои недостатки и сильные стороны [13]. Суть заключается в том, что под каждую задачу всё же необходимо подбирать конкретный вариант токенизации.

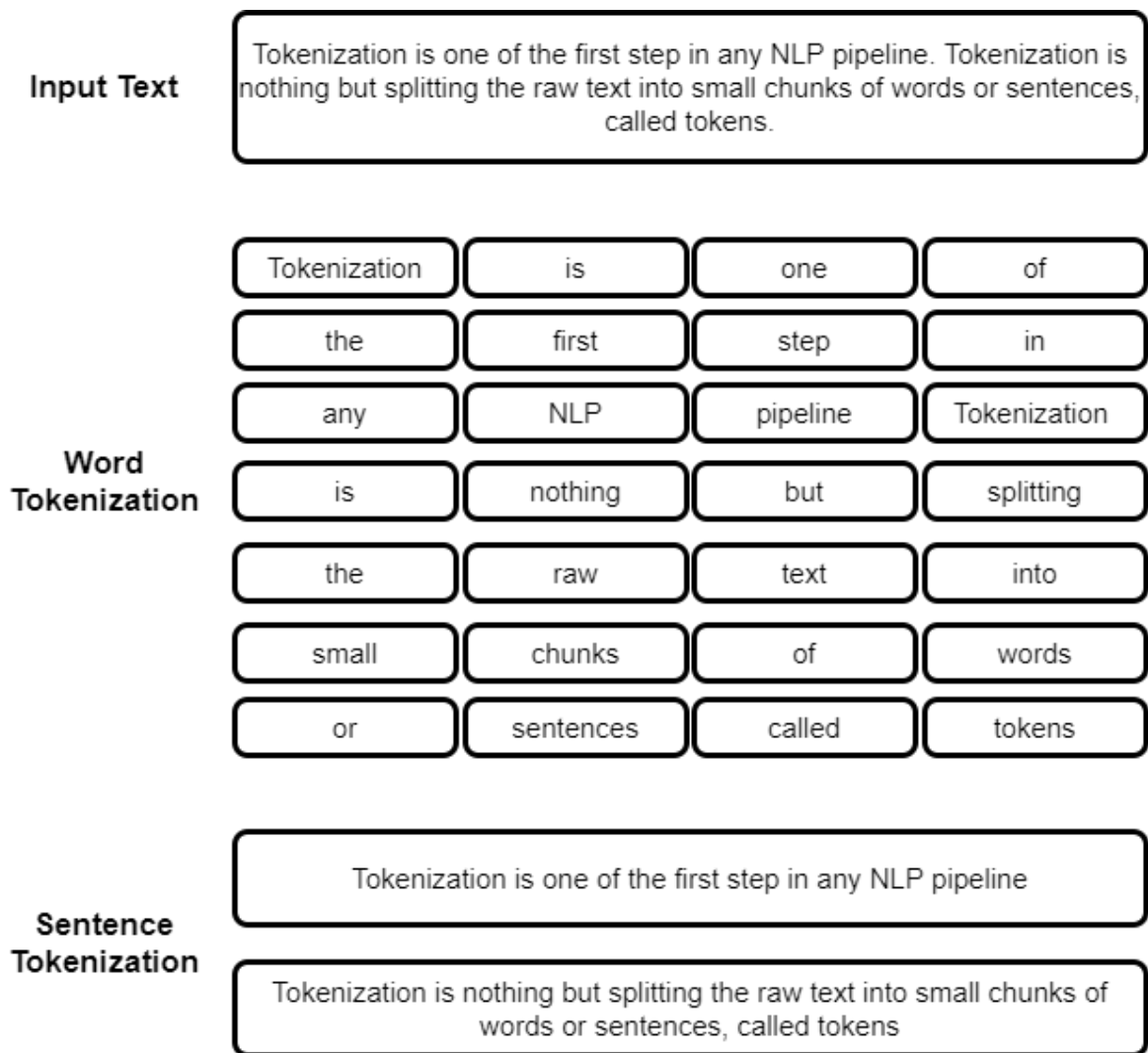


Рисунок 1.12 – Пример токенизации по предложениям и токенизации по словам с учётом знаков препинания

#### 1.4.2 Удаление стоп-слов из текста

Под стоп-словами в области обработки естественных языков обычно подразумеваются артикли, союзы, междометия, а также любые другие слова, которые не несут смысловой нагрузки в тексте и добавляют мешающий для последующей обработки шум. Авторами статьи «Stop Word Lists in Free Open-source Software Packages» были проанализированы 52 популярных списка стоп-слов на английском языке (Рисунок 1.13).

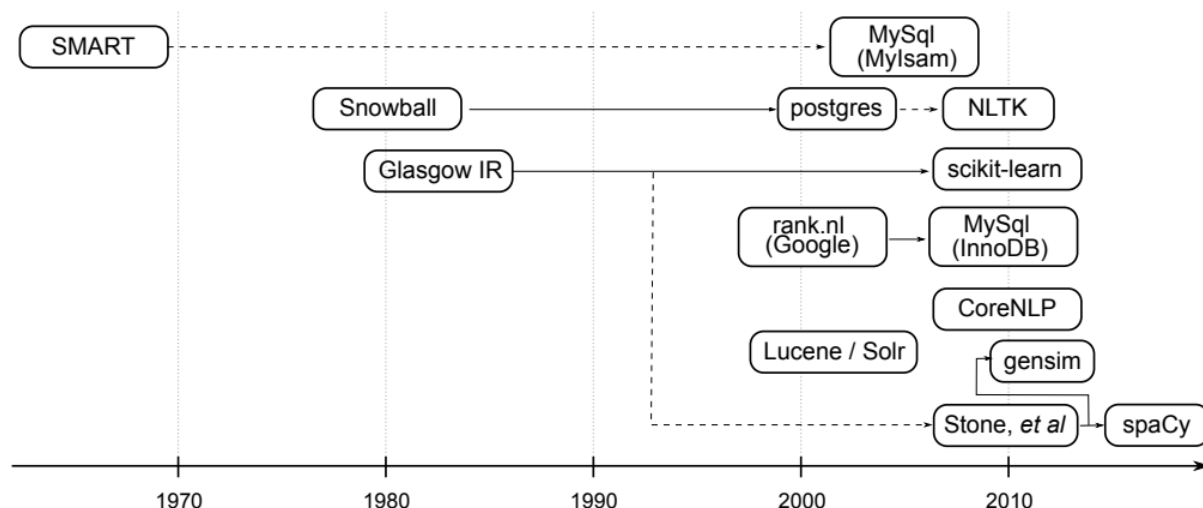


Рисунок 1.13 – Генеалогическое древо популярных списков стоп-слов [14]

Списки стоп-слов – это простой, но полезный инструмент для управления шумом с повсеместной поддержкой программного обеспечения для обработки естественного языка. Пользователи часто применяют вслепую популярные стоп-листы, которые могут страдать от неожиданных пропусков и включений или от их несовместимости с конкретными токенизаторами. Многие из этих проблем могут возникнуть в результате создания стоп-листов с использованием корпусной статистики. Помимо анализа авторами статьи «Stop Word Lists in Free Open-source Software Packages» также были предложены стратегии для устранения проблем в использовании списков стоп-слов [14], в которые входят следующие пункты:

- 1) Улучшение документации;
- 2) Динамическая адаптация стоп-листов во время предварительной обработки;
- 3) Создание инструментов для контроля качества стоп-листов и автоматической генерации стоп-листов.

#### 1.4.3 Лемматизация и стемминг

В текстах часто встречаются различные грамматические формы одного слова, а также слова с одинаковым корнем. Лемматизация и стемминг направлены на приведение всех этих слов к общей словарной форме

(Рисунок 1.14). Лемматизация и стемминг являются частными случаями нормализации слов, а также кардинально различаются в своей реализации.

Лемматизация – это процесс нахождения базовой морфологической формы (леммы) слова. Это довольно важный шаг во многих задачах обработки естественного языка, поиска информации и извлечения информации [15].

Авторами статьи «Highly Language-Independent Word Lemmatization Using a Machine-Learning Classifier» был представлен независимый от языка лемматизатор с открытым исходным кодом, основанный на модели классификации случайного леса. Модель является контролируемым алгоритмом машинного обучения с деревьями решений, которые строятся в соответствии с грамматическими особенностями языка. Авторы статьи сравнили производительность своего лемматизатора с производительностью лемматизатора UDPipe на 22 из 25 языков. Метод лемматизации из статьи успешно работает на разных языках и легко расширяется на другие (Таблица 1). Исходный код лемматизатора доступен для использования в учебных целях [16].

Таблица 1 Список языков, используемых в реализации лемматизатора [16]

Язык	Код	Языковая группа	Пары слов	Источник
Астурийский	ast	Романская	108,792	Списки лемматизации
Болгарский	bg	Славянская/балтийская	30,323	Списки лемматизации
Каталонский	ca	Романская	591,534	Списки лемматизации
Чешский	cs	Славянская/балтийская	36,400	Списки лемматизации
Английский	en	Германская	41,649	Списки лемматизации
Эстонский	et	Уральско-алтайская	80,536	Списки лемматизации
Фарси	fa	Иранская	6,273	Списки лемматизации
Французский	fr	Романская	223,999	Списки лемматизации
Галисийский	gl	Романская	392,856	Списки лемматизации
Немецкий	de	Германская	358,473	Списки лемматизации
Венгерский	hu	Уральско-алтайская	39,898	Списки лемматизации
Ирландский	ga	Гэльская	415,502	Списки лемматизации

Итальянский	it	Романская	341,074	Списки лемматизации
Мэнский гэльский	gv	Гэльская	67,177	Списки лемматизации
Португальский	pt	Романская	850,264	Списки лемматизации
Румынский	ro	Романская	314,810	Списки лемматизации
Русский	ru	Славянская/балтийская	2,657,468	Зарганский словарь
Шотландский гэльский	gd	Гэльская	51,624	Списки лемматизации
Словацкий	sk	Славянская/балтийская	858,414	Списки лемматизации
Словенский	sl	Славянская/балтийская	99,063	Списки лемматизации
Испанский	es	Романская	496,591	Списки лемматизации
Шведский	sv	Германская	675,137	Списки лемматизации
Турецкий	tr	Уральско-алтайская	1,337,898	Зарганский словарь
Украинский	uk	Славянская/балтийская	193,704	Списки лемматизации
Валлийский	cy	Гэльская	359,224	Списки лемматизации

Стемминг, в отличие от лемматизации, является несколько более примитивным и грубым методом нормализации слов. Стеммер отличается от лемматизатора тем, что он не учитывает контекст и не различает значения слов в зависимости от части речи. Тем не менее, стеммеры имеют свои преимущества: они проще в использовании и работают быстрее. Кроме того, в некоторых случаях более низкий уровень точности не имеет значения [15]. Стеммеры не так сильно распространены, как лемматизаторы, но тем не менее могут использоваться в следующих случаях:

- 1) Системы поиска информации;
- 2) Определение словарей предметной области при анализе предметной области.

# Stemming vs Lemmatization



Рисунок 1.14 – Пример нормализации слова при стемминге и лемматизации [15]

Подытожив, можно сказать, что несмотря на быструю работу стемминга, лемматизация является наиболее предпочтительным методом в большинстве задач обработки естественного языка.

## 1.4.4 Синтаксический анализ текста

Синтаксический анализ, или парсинг, или синтаксический разбор – это третий этап в обработке естественного языка. Синтаксический анализ подразумевает анализ значения текста в сравнении с формальными грамматическими правилами, следовательно синтаксический анализ может определить соблюдаются ли грамматические правила в тексте (Рисунок 1.15). В качестве примера рассмотрим предложение «Мэри Джо дорога олень водить». Токены и точка будут указывать на полное предложение, но, по своей сути, предложение не будет передавать какой-либо смысл. Процессами обеспечения соответствия токенов определенной грамматике специалисты по вычислительной технике и вычислительной лингвистике также называют синтаксическим анализом. В идеале для обработки текста необходим как лексический анализатор, так и синтаксический анализатор, поскольку есть множество тонких моментов, которые не отловить лишь одним анализатором [17].

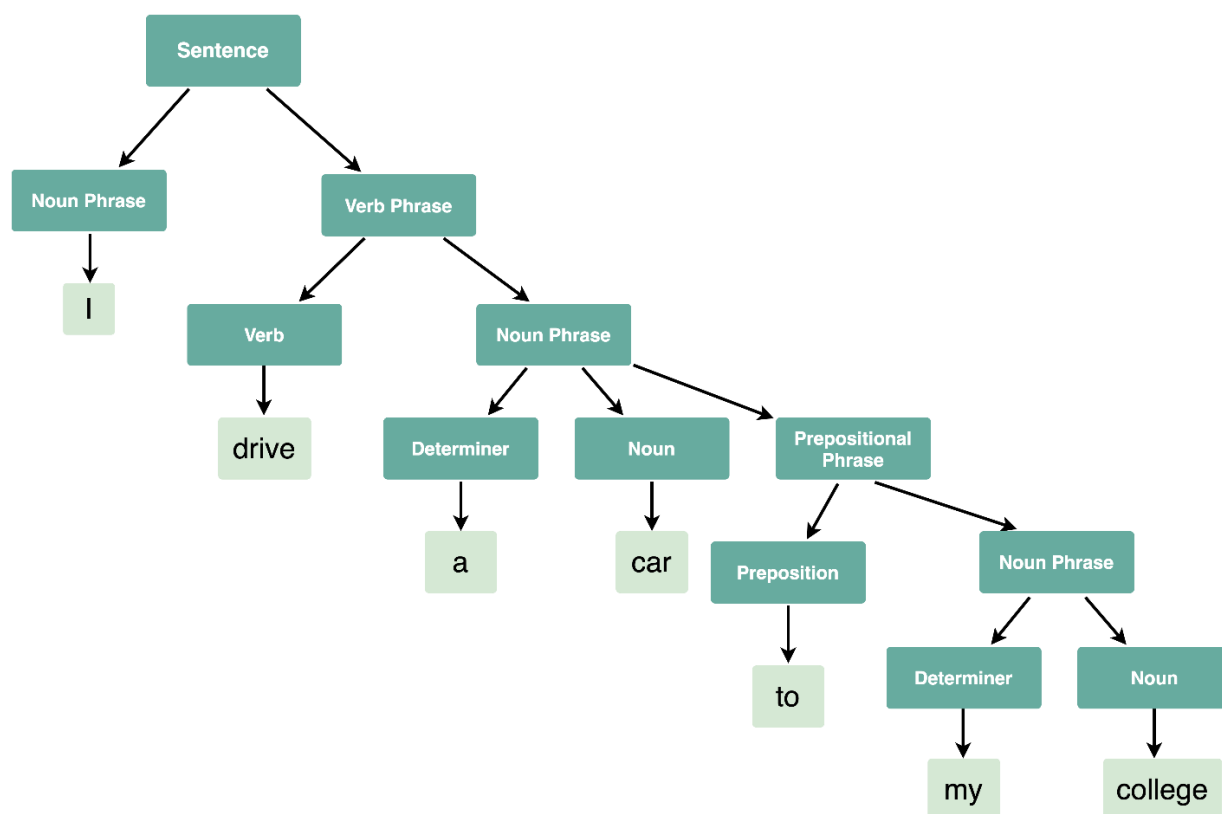


Рисунок 1.15 – Синтаксическое дерево на основе предложения

### 1.5 Библиотеки для обработки естественного языка

Одним из ключевых компонентов в создании высокоэффективных систем автоматической обработки естественного языка являются программные библиотеки. На сегодняшний день существует множество библиотек для обработки естественного языка на разных языках программирования. Каждая из этих библиотек имеет свои преимущества и недостатки, поэтому рекомендуется их использовать в первую очередь, ориентируясь на выполнение поставленной задачи.

Авторами статьи «Choosing an NLP Library for Analyzing Software Documentation: A Systematic Literature Review and a Series of Experiments» были проанализированы популярные библиотеки для обработки естественного языка (Рисунок 1.16).



	Comparison	tokens for first library	tokens for second library	identical tokens	identical tokens %	identical token/POS <sub>g</sub>	identical token/POS <sub>g</sub> %	identical token/POS <sub>s</sub>	identical token/POS <sub>s</sub> %
Stack Overflow	Stanford vs. SyntaxNet	8,743	8,485	8,137	94.46	7,507	87.15	7,322	85.00
	Stanford vs. spaCy	8,743	8,589	8,312	95.92	7,743	89.35	7,548	87.10
	Stanford vs. NLTK	8,743	8,614	8,454	97.41	7,062	81.37	6,605	76.11
	SyntaxNet vs. spaCy	8,485	8,589	8,016	93.90	7,678	89.94	7,511	87.98
	SyntaxNet vs. NLTK	8,485	8,614	8,281	96.86	6,876	80.43	6,441	75.34
	spaCy vs. NLTK	8,589	8,614	8,326	96.80	7,035	81.79	6,585	76.56
GitHub ReadMe	Stanford vs. SyntaxNet	6,289	6,167	5,996	96.27	5,271	84.63	4,998	80.25
	Stanford vs. spaCy	6,289	6,278	6,034	96.03	5,332	84.86	5,100	81.16
	Stanford vs. NLTK	6,289	6,232	6,133	97.96	5,017	80.14	4,395	70.20
	SyntaxNet vs. spaCy	6,167	6,278	5,910	94.98	5,512	88.58	5,307	85.29
	SyntaxNet vs. NLTK	6,167	6,232	6,093	98.28	4,901	79.05	4,399	70.96
	spaCy vs. NLTK	6,278	6,232	6,040	96.56	4,885	78.10	4,313	68.95
Java API Doc.	Stanford vs. SyntaxNet	11,675	10,896	10,488	92.93	9,103	80.66	8,564	75.88
	Stanford vs. spaCy	11,675	11,015	10,611	93.53	9,390	82.77	8,832	77.85
	Stanford vs. NLTK	11,675	11,044	10,788	94.97	8,711	76.68	7,889	69.45
	SyntaxNet vs. spaCy	10,896	11,015	10,008	91.35	9,171	83.71	8,697	79.38
	SyntaxNet vs. NLTK	10,896	11,044	10,735	97.86	8,734	79.62	7,950	72.47
	spaCy vs. NLTK	11,015	11,044	10,311	93.49	8,602	77.99	7,713	69.93

Рисунок 1.16 – Степень совпадения результатов библиотек обработки естественного языка [18]

Следуя из проведённого авторами статьи анализа, можно понять, что результаты, получаемые с помощью библиотек, часто схожи. Исходя из этого, выбор библиотеки в основном будет основываться на поддержке русского языка.

### 1.5.1 Пакет библиотек NLTK

Одним из самых распространённых и популярных пакетов библиотек является NLTK на языке Python. NLTK обладает большинством возможностей в обработке естественного языка [19]. Единственным значимым минусом является то, что некоторые функции не поддерживают русский язык. Из пакета библиотек планируется использовать технику токенизации.

```
Исходный текст: Меня зовут Грумшак. Я орк из клана Кровавый Клык. Я сильный воин.

Токенизация по предложениям:
0  токен:  Меня зовут Грумшак.
1  токен:  Я орк из клана Кровавый Клык.
2  токен:  Я сильный воин.

Токенизация по словам:
0  токен:  Меня          1  токен:  зовут
2  токен:  Грумшак      3  токен:  .
4  токен:  Я            5  токен:  орк
6  токен:  из           7  токен:  клана
8  токен:  Кровавый     9  токен:  Клык
10 токен:  .            11 токен:  Я
12 токен:  сильный     13 токен:  воин
14 токен:  .
```

Рисунок 1.17 – Токенизация по словам и по предложениям с использованием пакета библиотек NLTK

Исходя из результатов токенизации (Рисунок 1.17) можно понять, что важным моментом является наличие знаков препинания в списке токенов, а следовательно, может понадобиться дополнительная обработка для исключения знаков препинания из списка токенов.

### 1.5.2 Библиотека Rymorphy2

Rymorphy2 – это морфологический анализатор для русского языка, реализованный на Python. Поскольку Rymorphy2 анализирует по одному слову за раз (Рисунок 1.18), он возвращает список возможных вариантов разбора с соответствующей вероятностью. Однако анализ изолированного слова редко происходит без какой-либо двусмысленности, поскольку часто слова имеют больше одного значения. При работе с моделями «мешка слов» обычной практикой является просто выбор для каждой словоформы синтаксического анализа с наибольшей вероятностью. В этом случае, предполагая, что требуется только словарная форма слова, Rymorphy2 получит правильную часть речи примерно в 92% случаев. Однако, если принимать во внимание грамматический падеж слова, точность снизится до 82%, и при синтаксическом анализе часто будут получаться неграмматичные

последовательности [20]. Из библиотеки Rymorphy2 планируется использовать техники лемматизации слов и определения части речи.

```
Исходный текст: Сила заключается в твёрдой верности своей расе и преданности своим братьям

Parse(word='сила',          tag='NOUN,inan,femn sing,nomn',      normal_from='сила')
Parse(word='заключается',   tag='VERB,impf,intr sing,3per,pres,indc', normal_from='заключаться')
Parse(word='в',             tag='PREP',                          normal_from='в')
Parse(word='твёрдой',       tag='ADJF,Qual femn,sing,abl',      normal_from='твёрдый')
Parse(word='верности',     tag='NOUN,inan,femn sing,gent',      normal_from='верность')
Parse(word='своей',        tag='ADJF,Apro,Anph femn,sing,gent', normal_from='свой')
Parse(word='расе',         tag='NOUN,inan,femn sing,dativ',     normal_from='раса')
Parse(word='и',            tag='CONJ',                          normal_from='и')
Parse(word='преданности',  tag='NOUN,inan,femn sing,gent',      normal_from='преданность')
Parse(word='своим',        tag='ADJF,Apro,Anph plur,dativ',     normal_from='свой')
Parse(word='братьям',     tag='NOUN,anim,masc plur,dativ',     normal_from='брат')
```

Рисунок 1.18 – Морфологический анализ всех слов в предложении с использованием библиотеки Rymorphy2

### 1.5.3 Библиотека SpaCy

SpaCy является комплексной библиотекой, которая может проводить различные операции обработки естественного текста, но самой главной операцией является синтаксический анализ текста (Рисунок 1.19). SpaCy реализован на Cython, что обеспечивает очень высокую производительность по сравнению с другими библиотеками обработки естественного языка, например NLTK. Несколько оценок, проведенных в рецензируемых журналах, показывают, что SpaCy обеспечивает производительность при выполнении задач синтаксического анализа и распознавания сущностей, сопоставимую с другими широко используемыми инструментами, при этом имеет значительное преимущество в отношении скорости [21].

Исходный текст: Сила заключается в твёрдой верности своей расе и преданности своим братьям

Слово	Часть речи	Синтаксическая связь	Родитель
Сила	NOUN	nsubj (nominal subject)	заключается
заключается	VERB	ROOT (root)	заключается
в	ADP	case (case marking)	верности
твёрдой	ADJ	amod (adjectival modifier)	верности
верности	NOUN	obl (oblique nominal)	заключается
своей	DET	det (determiner)	расе
расе	NOUN	iobj (indirect object)	верности
и	CCONJ	cc (coordinating conjunction)	преданности
преданности	NOUN	conj (conjunct)	расе
своим	DET	det (determiner)	братьям
братьям	NOUN	iobj (indirect object)	преданности

Рисунок 1.19 – Синтаксический анализ предложения с использованием библиотеки SpaCy

## 1.6 Типология

Типология является разделом лингвистики, изучающим разнообразие языковых систем и их классификацию на основе общих структурных и функциональных характеристик. Типология сравнивает различные языки с целью выявления общих закономерностей и различий между ними, а также классифицирует языки по различным признакам, таким как порядок слов, системы флексии, морфологическую, фонетико-фонологическую, просодическую, синтаксическую и лексическую составляющие [22].

Знание типологических особенностей различных языков помогает понять и учесть особенности структуры языка при разработке алгоритмов обработки текста, поэтому был сделан большой упор в исследовании типологии и её видов для уточнения теоретической модели.

### 1.6.1 Фонетико-фонологическая типология

Фонетико-фонологическая типология является областью лингвистики, которая изучает различия и сходства в звуковых системах языков различных культур и регионов. Она фокусируется на анализе звуковой структуры языков, включая фонетические и фонологические особенности звуков, исследуя их распространение, вариативность и системные закономерности.

Целью фонетико-фонологической типологии является выявление универсальных и специфических характеристик звуковых систем различных языков, а также определение тенденций и закономерностей, которые могут быть общими для большинства языков или для определенных языковых семей [23]. Исследования в области фонетико-фонологической типологии могут включать в себя анализ звуковых наборов, фонетических свойств звуков, фонологические процессы и другие аспекты звуковой структуры языков. Изучение фонетико-фонологической типологии позволяет понять разнообразие звуковых систем в мире и выявить универсальные законы, лежащие в их основе, что способствует более глубокому пониманию природы человеческого языка.

#### 1.6.1.1 Вокаличность и консонантность

Одной из важных составляющих фонетико-фонологической типологии являются вокаличность и консонантность. Вокалические языки характеризуются тем, что в них звуки, называемые гласными (или вокалами), играют важную роль и могут быть доминирующими в звуковой структуре слов (Таблица 2). В таких языках можно наблюдать большое количество различных гласных звуков, а также сложные системы гласных дифтонгов. Примерами вокалических языков являются французский, испанский, итальянский. Консонантные языки, напротив, характеризуются тем, что в них основную роль играют звуки, называемые согласными (или консонантами). В таких языках можно обнаружить богатство согласных звуков, а также сложные системы согласных кластеров. Примерами консонантных языков являются некоторые языки индоевропейской семьи, такие как немецкий, русский, английский.

На данный момент в каждом языке мира существует ограниченное количество фонем, которые образуют систему звуковых характеристик. Согласно фонетико-фонологическим универсалиям, количество фонем в различных языках обычно колеблется от 10 до 81, со средним значением около 40–50. Есть минимальное число фонем, необходимое для языка, а

также фонемы, которые присутствуют в большинстве языков. Например, практически все языки имеют определенные консонантные и вокальные звуки, такие как взрывные звуки p, t, k; фрикативный звук s или h; сонорные звуки j, r или l, носовой звук n и не менее двух гласных фонем [24].

Таблица 2 – Вокалический коэффициент [24]

Язык	Всего фонем (Ph) (100 %)	Гласные (V) (% от Ph)	Согласные (C) (% от Ph)	Вокалический коэффициент (vk)
Шведский	52	17 (33%)	35 (67%)	0,49
Готский	30	10 (33%)	20 (67%)	0,5
Древнеисландский	44	16 (36%)	28 (64%)	0,57
Древнеанглийский (вариант 1)	54	20 (37 %)	34 (63 %)	0,57 (0,59)
Древнеанглийский (вариант 2)	39	23 (59 %)	16 (41%)	1,44
Фарерский	59	23 (39 %)	36 (61%)	0,64
Норвежский	64	25 (39–40 %)	39 (61–60 %)	0,66 (0,64)
Исландский	62–65	26 (42–40 %)	36–39 (58–60%)	0,72–0,67
Французский	36	16 (44 %)	20 (56 %)	0,8
Латинский	35	16 (46 %)	19 (54 %)	0,84
Английский	45	21 (47 %)	24 (53 %)	0,88
Немецкий	40	19 (47,5 %)	21 (52,5 %)	0,9
Нидерландский	42	21 (50 %)	21 (50 %)	1,00
Финский (вариант 1)	39	26 (67 %)	13 (33 %)	2,00
*Финский (вариант 2)	56	34 (61 %)	26 (39 %)	1,55
Датский	47	32 (68 %)	15 (32 %)	2,13
Фризский	64	45 (70–69 %)	19 (30–31 %)	2,37–2,25

### 1.6.1.2 Бинарная оппозиция

Бинарная оппозиция в фонологии представляет собой противопоставление двух фонем, которое играет важную роль в организации звуковой системы языка. Это означает, что звуки могут быть различными по одному конкретному признаку, который имеет значение для различения слов. Например, по признаку глухости-звонкости или по признаку мягкости-твёрдости. В паре противопоставленных звуков один является немаркированным, а другой маркированным. Немаркированный звук имеет большее употребление в языке и легче произносится. В паре звуков «т – д» звук «т» является немаркированным, поэтому данный звук чаще употребляется в языке, и он легче произносится. Маркированный звук является менее удобным для произношения и реже встречается в языке. В определенных ситуациях противопоставленные звуки могут терять свои различия. Например, в конце слова в русском языке звук «д» произносится как «т» (код = кот), то есть менее употребляемый звук теряет свою характеристику.

Бинарные оппозиции помогают организовать фонологическую систему языка, позволяя различать слова на основе минимальных звуковых различий. Они также являются важным аспектом в сравнительно-историческом анализе языков, так как изменения в бинарных оппозициях могут служить индикаторами фонологических изменений в языке.

### 1.6.2 Морфологическая типология

Морфологическая типология – это раздел лингвистики, который изучает различные типы морфологических систем в языках мира. Она занимается классификацией языков по структуре и организации их морфологических категорий и форм. Морфологическая типология исследует различия в образовании слов, изменении их формы для выражения грамматических значений (например, падежей, числа, времени), а также изучает, какие типы морфологических систем существуют в различных языках и как они отличаются друг от друга [25].

### 1.6.2.1 Изолирующие языки

Изолирующие языки, также известные как аналитические языки, представляют собой тип морфологической структуры, в которой слова обычно не изменяются с помощью аффиксов или окончаний, чтобы выразить грамматические отношения [26]. Вместо этого они используются словосочетания, порядок слов и служебные слова для передачи грамматической информации. Изолирующие языки имеют тенденцию к использованию отдельных слов для выражения различных грамматических отношений вместо того, чтобы изменять форму самого слова. Китайский язык (включая китайский мандарин) и вьетнамский язык - примеры изолирующих языков.

### 1.6.2.2 Агглютинативные языки

Агглютинативные языки являются типом морфологической структуры, в котором грамматические отношения выражаются путем добавления аффиксов к основе слова [27]. В отличие от изолирующих языков, в агглютинативных языках каждый аффикс обычно несет одну грамматическую функцию, такую как падеж, число, время. Аффиксы добавляются к основе слова в определенном порядке, что позволяет легко различать различные грамматические категории. Примером агглютинативного языка является финский, в котором аффиксы добавляются к корню слова для выражения различных грамматических отношений. Например, в финском слово «talo» означает «дом», а если добавить аффикс «-ssa», получится «talossa», что означает «в доме». Также одним из примеров агглютинативных языков является индонезийский (Таблица 3). Таким образом, агглютинативные языки отличаются от изолирующих языков тем, что грамматическое отношение выражается через аффиксы, в то время, как в изолирующих языках грамматическая информация передается через словосочетания и служебные слова.



Таблица 3 – Пример использования аффиксов в индонезийском языке [27]

Индонезийский	Русский
ajar	учить
ajaran	учение
belajar	учить (самому)
mengajar	учить (кого-то)
mempelajari	изучать
dipelajari	быть изучаемым
pelajar	ученик
pengajar	учитель
pelajaran	предмет, образование
pengajaran	урок, мораль истории
pebelajaran	изучающий
terpelajar	хорошо образованный

#### 1.6.2.3 Флективные языки

Флективные языки представляют собой тип морфологической структуры, в котором грамматические отношения выражаются путем изменения формы слова. В отличие от агглютинативных языков, в которых грамматические категории обычно выражаются путем добавления аффиксов к основе слова, флективные языки используют грамматические окончания, изменения корня слова и другие формы изменения для передачи различных грамматических значений. Ярким примером флективного языка является русский, в котором слова изменяют свои формы в зависимости от времени, лица, числа, падежа.

Из-за своих особенностей флективные языки могут быть сложны для изучения по сравнению с изолирующими и агглютинативными языками, поскольку в них может присутствовать большое количество дополнительных склонений и неочевидных форм. Авторы научной статьи «On the Complexity and Typology of Inflectional Morphological Systems» проанализировали 36 языков на предмет «сложности» (Рисунок 1.20).

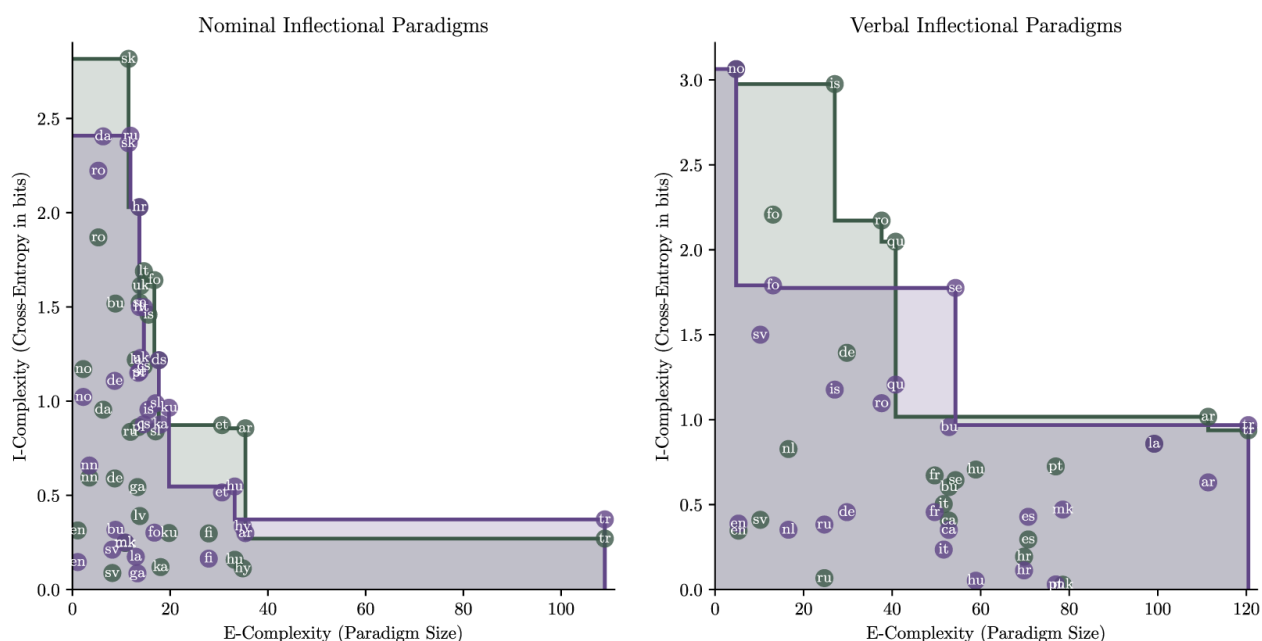


Рисунок 1.20 – Результаты анализа 36 языков на предмет «сложности» [28]

#### 1.6.2.4 Инкорпорирующие языки

Инкорпорирующие языки представляют собой тип морфологической структуры, в котором элементы грамматики могут быть включены непосредственно в основу слова. Это означает, что в инкорпорирующих языках грамматические отношения выражаются путем объединения корней слов с элементами, которые в других языках могли бы быть выражены отдельными словами. Например, в инкорпорирующем языке такие конструкции, как «съесть рыбу» или «пить воду», могут быть выражены одним словом, включающим корень глагола и объект (рыба, вода) внутри этого слова. Инкорпорирующие языки обычно имеют богатую систему аффиксации, позволяющую создавать сложные слова, включающие в себя несколько грамматических категорий. Такие языки часто встречаются у народов индейцев Северной Америки и некоторых народов Сибири.

#### 1.6.3 Просодическая типология

Просодическая типология относится к классификации языков на основе их просодических характеристик, таких как интонация, ритм, тон, длительность звуков. Процесс классификации языков по просодическим характеристикам позволяет выявить общие черты и различия между языками

и группами языков [29]. В интонациях изучаются различия в тоне и интонационных контурах, которые могут использоваться для выражения эмоционального состояния, вопросительности, удивления и других нюансов значения. В ритме изучаются характерные особенности распределения ударений и длительности звуков в словах и предложениях. Изучение характеристики тона подразумевает использование высоты голоса для выделения значимых элементов в предложении. Изучение характеристики длительности связано с продолжительностью звуковых сегментов, которая может быть значима для различения слов или выражения эмоций. Исследования просодической типологии могут помочь установить общие закономерности и различия между языками, а также понять, как просодические характеристики влияют на структуру языка и его восприятие.

#### 1.6.4 Синтаксическая типология

Синтаксическая типология относится к классификации языков на основе их синтаксических характеристик, таких как порядок слов, структура предложений, способы выражения отношений между словами и фразами. Процесс классификации языков по синтаксическим характеристикам позволяет выявить общие черты и различия между языками и группами языков [30]. Различные языки могут иметь различный порядок слов в предложениях. Языки могут иметь различия в структуре предложений, а также различные типы подчинения. Языки могут иметь различия в способах выражения отношений между словами и фразами с помощью морфологии и синтаксиса. Исследования синтаксической типологии позволяют установить общие закономерности и различия между языками, а также понять, как синтаксические характеристики влияют на структуру языка, его выразительные возможности и восприятие.

##### 1.6.4.1 Эргативные языки

Эргативные языки представляют собой тип языков, в которых существует особая морфологическая и синтаксическая система для выражения аргументов глагола. Основным отличительным признаком

эргативных языков является то, что они используют эргативную конструкцию для образования формы глагола в зависимости от того, является ли подлежащее агентом или пациенсом [31]. В эргативных языках форма глагола для агента (того, кто совершает действие) может отличаться от формы для пациенса (того, на кого направлено действие). При этом форма для агента может быть похожа на форму для инструмента или аблатива. Таким образом, в эргативном языке одна и та же морфема может использоваться для выражения агента в одних конструкциях и пациенса в других. Эргативные языки могут также отличаться от номинативно-аккузативных языков (где форма глагола зависит от того, является ли подлежащее субъектом или объектом) в структуре предложений и порядке слов. Примером эргативного языка является баскский язык, а также некоторые языки индоарийской и абхазо-адыгских семей.

#### 1.6.4.2 Номинативные языки

Номинативные языки – это языки, в которых основная форма существительных, местоимений и прилагательных используется для обозначения подлежащего в предложении [32]. В таких языках подлежащее именуется как номинатив (от латинского «*nominativus*», что означает «именительный»). Номинативные стратегии присущи большинству современных индоевропейских языков. В номинативных языках существительные, местоимения и прилагательные изменяются по числам (единственное и множественное число) и по родам (мужской, женский, средний). Формы этих слов могут меняться в зависимости от их роли в предложении (подлежащее, дополнение). В отличие от номинативных языков, в эргативных языках форма существительных, местоимений и прилагательных может изменяться в зависимости от их роли в качестве агенса (действующее лицо) или пациенса (пассивное лицо) в предложении.

#### 1.6.4.3 Порядок слов

Типология порядка слов – это область лингвистики, которая изучает различные способы упорядочения слов в предложениях в различных языках.

Она исследует, каким образом языки могут отличаться в распределении слов по разным позициям в предложении, а также какие универсальные и различные черты могут быть выявлены у различных языков. Типология порядка слов помогает классифицировать языки по их особенностям в распределении слов в предложении [33]. Например, она может рассматривать, какие языки имеют строгий порядок подлежащего, глагола и дополнения, а какие языки допускают более свободное распределение этих элементов. Типология порядка слов также изучает различные явления, такие как инверсия, топиализация, фокусирование и другие явления, связанные с порядком слов в предложении. Изучение типологии порядка слов позволяет понять, каким образом разные языки организуют информацию в предложениях, и выявлять универсальные черты языковой структуры, а также культурные и грамматические особенности конкретных языков.

Существует несколько основных типов порядка слов: SVO (подлежащее, сказуемое, дополнение), SOV (подлежащее, дополнение, сказуемое), VSO (сказуемое, подлежащее, дополнение), VOS (сказуемое, дополнение, подлежащее), OSV (дополнение, подлежащее, сказуемое), OVS (дополнение, сказуемое, подлежащее) [33]. Это основные типы порядка слов, которые могут быть использованы в различных языках. Важно отметить, что многие языки могут иметь более сложные системы порядка слов, а также различные варианты порядка слов в зависимости от контекста, стиля речи и других факторов.

#### 1.6.5 Лексическая типология

Лексическая типология – это область лингвистики, которая изучает различные типы лексических единиц в языках и их структурные особенности. Она исследует лексические системы разных языков, классифицирует слова и вырабатывает типы и категории лексических единиц [34]. Цель лексической типологии - выявить общие и отличительные черты лексического строения разных языков и установить типологические закономерности в развитии лексикона.

Лексическая типология может быть полезна для обработки естественного текста в нескольких аспектах. Она может помочь в построении эффективных моделей для автоматического извлечения информации и обработки текста. Изучение различных типов лексических единиц и их классификация позволяют определить их структурные свойства и общие особенности, что может быть полезно при разработке алгоритмов для обработки и анализа текста [35]. Лексическая типология может быть использована для разработки лексических ресурсов, таких как словари и тезаурусы, которые могут быть полезны при разработке систем машинного перевода, извлечения информации, а также для построения семантических моделей [36]. Изучение типологии лексики может помочь в определении и распознавании языковых единиц в тексте, таких как имена собственные, термины, фразеологизмы и другие специфические лексические конструкции. Следовательно, лексическая типология может способствовать созданию более точных и эффективных моделей для работы с естественным языком, таких как модели машинного обучения и искусственного интеллекта, обеспечивая более точные и связные результаты в обработке текста.

### **1.7 Конкретизация задачи**

Рассмотренные техники и библиотеки обработки естественного языка значительно расширили возможности семантического, синтаксического и морфологического анализа текста, а также сделали всё это довольно доступным для разработчиков.

Большие языковые модели являются невероятно широкопрофильными инструментами, с помощью которых можно решать массу задач. Правда в этом преимуществе кроется недостаток. Часто происходит так, что довольно сложно предоставить вводные данные большой языковой модели, которые приведут к ожидаемым пользователем выходным данным. Тем не менее, конкретно для генерации текста и диалогов большие языковые модели являются наилучшим на данный момент вариантом.

Для достижения цели в генерации диалогов с неигровыми персонажами, в которых корректность согласования и построения естественной речи зависит от расы неигрового персонажа, требуется разделить необходимые инструменты на конкретные задачи. Большие языковые модели следует использовать для генерации текста. Библиотеки обработки естественного языка следует использовать для обработки текста, полученного от большой языковой модели, с учётом выбранных пользователем настроек.

Исходя из проанализированных видов типологии, можно выделить следующие пункты, которые возможно реализовать на русском языке:

- 1) Замена маркированных звуков (фонетико-фонологическая типология).
- 2) Увеличение степени изолированности языка (морфологическая типология).
- 3) Изменение порядка слов (синтаксическая типология).
- 4) Замена лексических единиц (лексическая типология).

Остальные проанализированные пункты либо невозможно представить в текстовой форме, либо невозможно представить на русском языке из-за кардинально отличающегося строения.

## **2 ТЕОРЕТИЧЕСКАЯ МОДЕЛЬ АЛГОРИТМА**

### **2.1 Требования к системе**

В качестве требований к системе были сформированы следующие пункты:

1) Алгоритм должен иметь возможность принимать любой исходный текст. Исходный текст может быть предоставлен как пользователем, так и любой большой языковой моделью.

2) Выходные данные должны содержать то же самое количество токенов, что и в исходном тексте. Алгоритм не должен добавлять или удалять какие-либо слова.

3) Алгоритм должен принимать конечное количество входных параметров для преобразования текста.

4) Помимо преобразованного текста, алгоритм должен возвращать данные о проведенных преобразованиях для возможности их отображения.

5) Алгоритм должен иметь функционал для сохранения моделируемой языковой группы в JSON файл.

### **2.2 Ограничения системы**

Любая система и алгоритм имеет свои ограничения, поэтому алгоритм генерации диалогов с неигровыми персонажами на основе моделируемых языковых групп не исключение. В качестве ограничений системы были сформированы следующие пункты:

1) Алгоритм является языкозависимым. Исходный текст для преобразования может быть предоставлен только на русском языке.

2) Алгоритм способен работать только с корректными входными данными. Корректность входных данных должна проверяться на уровне пользовательского интерфейса.

3) Для корректной работы метода замены лексических единиц, лексические единицы необходимо заполнять в нормальной форме.

4) Помимо входных параметров корректность выходных данных также напрямую зависит от исходного текста. Если исходный текст будет



некорректно написан с точки зрения русского языка, то используемые морфологические и синтаксические анализаторы также не дадут корректные выходные данные.

5) Точность алгоритма сильно зависит от точности используемых морфологических и синтаксических анализаторов.

6) Итоговое время работы алгоритма зависит от размера входных данных.

### **2.3 Формализация модели**

В качестве получения исходного текста будет использоваться сервис «Character.ai», поскольку сервис предоставляет необходимый функционал для демонстрации алгоритма без нужды собственной имплементации. Сервис имеет большой набор качественных созданных персонажей, а также сервис сохраняет пользовательский контекст с персонажами.

Моделируемая языковая группа представляет из себя набор из следующих параметров:

- 1) Последовательность слов в предложении;
- 2) Набор лексических единиц для замены;
- 3) Степень изолированности текста;
- 4) Набор маркированных звуков для замены.

В качестве параметра последовательности слов в предложении присутствует шесть вариантов:

- 1) SVO (подлежащее, сказуемое, дополнение);
- 2) SOV (подлежащее, дополнение, сказуемое);
- 3) VSO (сказуемое, подлежащее, дополнение);
- 4) VOS (сказуемое, дополнение, подлежащее);
- 5) OSV (дополнение, подлежащее, сказуемое);
- 6) OVS (дополнение, сказуемое, подлежащее).

В качестве набора лексических единиц для замены должны быть представлены пары лексических единиц, которые включают в себя

заменяемую лексическую единицу в нормальной форме и заменяющую лексическую единицу в нормальной форме.

В качестве степени изолированности текста присутствует четыре варианта:

1) Степень 0 – язык является флективным (обычное представление русского языка).

2) Степень 1 – лемматизируются глаголы.

3) Степень 2 – лемматизируются глаголы, существительные и местоимения.

4) Степень 3 – лемматизируется весь текст.

Данное распределение было выбрано из-за приоритетности связанности и логичности текста при лемматизации различных частей речи.

В качестве набора маркированных звуков для замены должны быть представлены пары маркированных звуков, которые включают в себя заменяемую последовательность символов и заменяющую последовательность символов.

Для наглядного изображения теоретической модели была сформирована диаграмма (Рисунок 2.1).

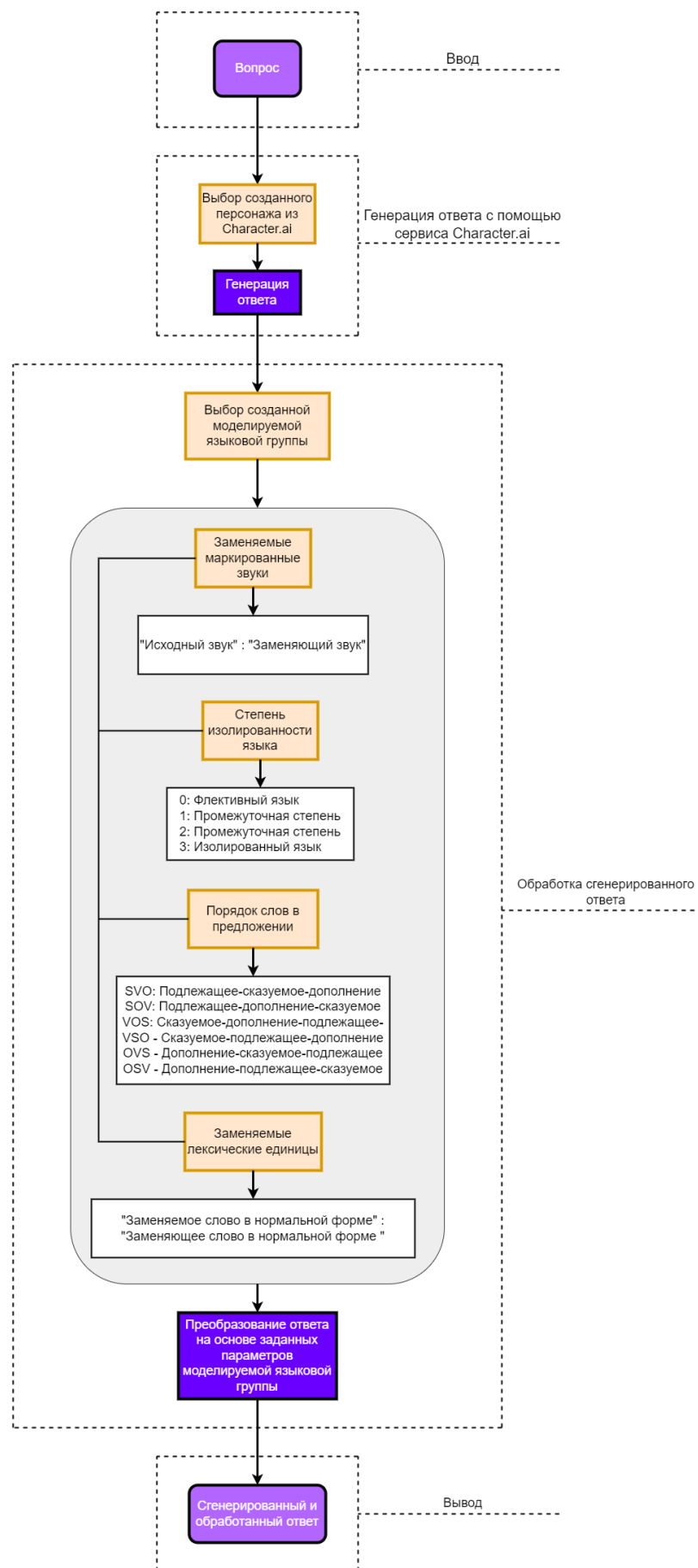


Рисунок 2.1 –Теоретическая модель

## 3 ПРАКТИЧЕСКАЯ ИМПЛЕМЕНТАЦИЯ АЛГОРИТМА

### 3.1 Реализация отдельных задач

Для моделирования языковой группы необходима теоретическая модель с параметрами, на которую можно будет опираться во время реализации алгоритмов.

Для реализации был выбран язык программирования Python по причине большой скорости программирования, а также лёгкого доступа к различным библиотекам [37, 38].

#### 3.1.1 Реализация замены маркированных звуков

Для реализации маркировки звуков было принято решение использовать примитивную замену представленных пользователем звуков. Данное решение было принято, поскольку система является языкозависимой, следовательно маркировку звуков можно производить только по существующим характеристикам русского языка. Помимо языкозависимости, вариант с примитивной заменой представленных пользователем звуков является более наглядным для любого пользователя без необходимости в глубоком понимании всех характеристик маркировки звуков конкретного языка (Рисунок 3.1).

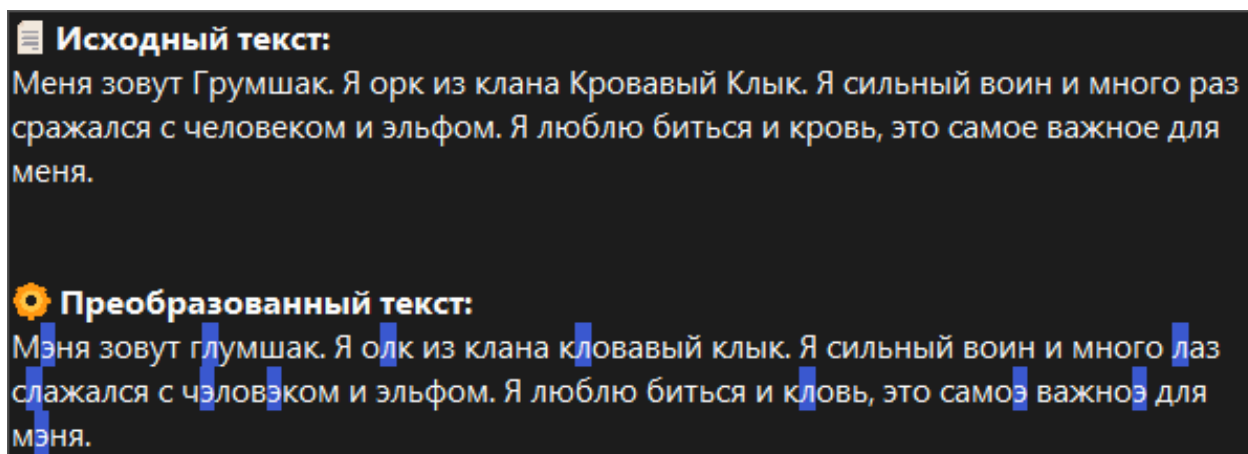


Рисунок 3.1 – Пример замены звуков «р» на «л» и «е» на «э»

### 3.1.2 Реализация преобразования флективного текста в определённую степень изолированности

Для достижения изолированного стиля в русском языке необходимо использовать технику лемматизации слов. Для этой цели используется морфологический анализатор Rymorphy2, реализованный на Python для русского языка. Поскольку Rymorphy2 анализирует по одному слову за раз, он возвращает список возможных вариантов разбора с соответствующей вероятностью [20]. В морфологический анализ входит нахождение нормальной формы слова (Рисунок 3.2).

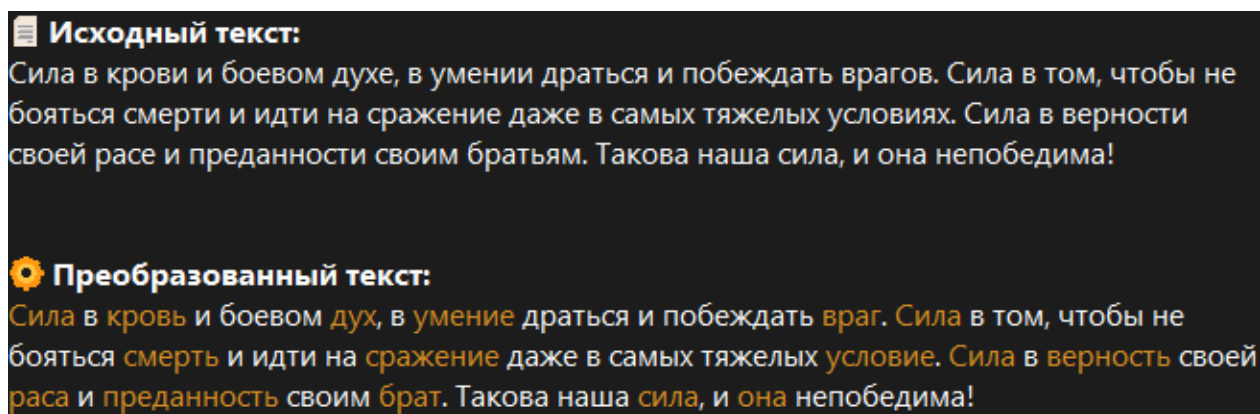


Рисунок 3.2 – Пример преобразования флективного текста во вторую степень изолированности

### 3.1.3 Реализация преобразования шести основных видов порядка слов

Для реализации преобразования текста к другому порядку слов крайне необходим синтаксический анализатор. На основе значительного преимущества над другими инструментами было принято решение использовать SpaCy для синтаксического анализа.

В ходе реализации разных порядков слов стало понятно, что синтаксический анализ предложений целиком приводит к усложнению изменения порядка из-за синтаксических связей, особенно в сложных предложениях [39]. На основе этой проблемы было решено делить предложения на «подпредложения». Это позволяет упростить синтаксические связи, а следовательно изменение порядка слов. В качестве демонстрации разных типов порядка слов возьмём предложение: «Сила

закljučается в верности своей расе» (Рисунок 3.3, Рисунок 3.4). В исходной форме порядок слов является SVO.

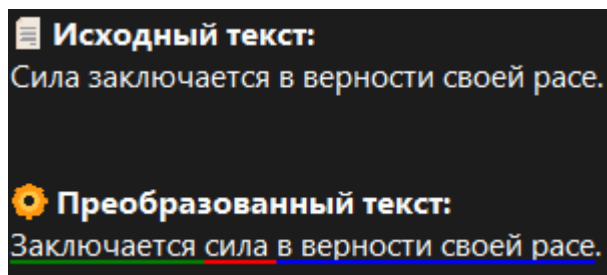


Рисунок 3.3 – Преобразование текста к порядку слов «Сказуемое – подлежащее – дополнение»

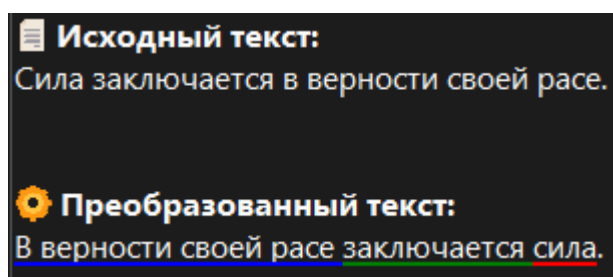


Рисунок 3.4 – Преобразование текста к порядку слов «Дополнение – сказуемое – подлежащее»

#### 3.1.4 Реализация замены лексических единиц

Для реализации замены лексических единиц был использован морфологический анализатор Rymorphy2 для приведение заменяющей лексической единицы в форму (Рисунок 3.5), в которой была заменяемая лексическая единица. Также был использован контекстный лемматизатор PyMystem3 для корректной лемматизации, поскольку Rymorphy2 не учитывает контекст слова в предложении.

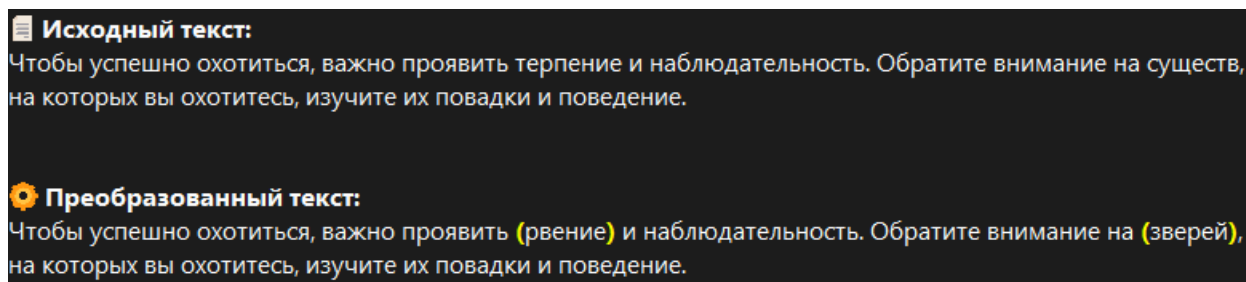


Рисунок 3.5 – Замена лексических единиц с набором «терпение – рвение» и «существо – зверь»

### 3.2 Объединение реализованных задач в цельный алгоритм

После реализации отдельных задач для преобразования текста необходимо применить функции в определённом порядке для корректности выходных данных. Синтаксический анализ необходимо делать самым первым, поскольку для качественного синтаксического анализа текстов на русском языке необходимо учитывать все морфемы. Далее необходимо приводить текст в определённую степень изолированности, поскольку при замене звуков до морфологического анализа может произойти преобразование слов в несуществующие с точки зрения морфологического анализатора. Для демонстрации заполним все параметры моделируемой языковой группы (Рисунок 3.6).

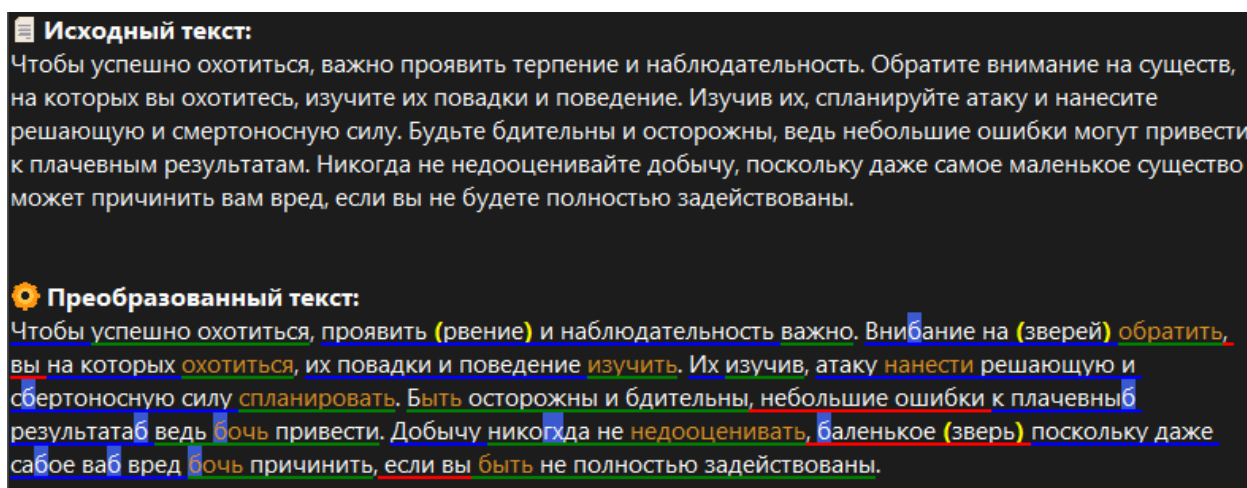


Рисунок 3.6 – Работа алгоритма с порядком слов «SOV», набором лексических единиц «терпение – рвение» и «существо – зверь», первой степенью изолированности и маркированными звуками «г – гх» и «м – б»

### 3.3 Получение данных о проведённых преобразованиях

Во время создания пользовательского интерфейса для демонстрации алгоритма генерации диалогов с неигровыми персонажами на основе моделируемых языковых групп появилась нужда в визуальном отображении всех проведённых преобразований текста для максимальной наглядности выходных данных. На основе этого была проведена работа по дополнению логики функций преобразования текста. Помимо этого, были улучшены несколько аспектов в функциях преобразования текста для получения более

качественных выходных данных.

Помимо самого преобразования текста с помощью четырёх функций преобразования текста было необходимо получить данные о преобразованных словах и символах, поскольку в дальнейшем следует визуально отобразить все проведённые преобразования текста. Данный функционал необходим не только для визуального отображения проведённой работы, но также весьма полезен для конечного пользователя, которому потенциально может понадобиться получение данных о проведённых преобразованиях для определённых целей.

Поскольку четыре функции преобразования текста в определённой мере отличаются своим исходным кодом, логика для получения данных о проведённых преобразованиях будет также отличаться для каждой из функций.

### 3.3.1 Данные об изменённом порядке слов

Самым ресурсоёмким функционалом обладает метод преобразования порядка слов, следовательно и функционал по получению данных о проведённых преобразованиях является самым ресурсоёмким в сравнении с другими тремя функциями преобразования текста.

Сам способ получения представляет из себя заполнение списка индексами токенов, относящихся к фразе с подлежащим, со сказуемым или с дополнением.

Помимо добавления функционала для получения по получению данных о проведённых преобразованиях, также был улучшен основной функционал по преобразованию порядка слов. Раньше для всех шести порядков слов применялась однотипная логика, которая была похожа на дублирование кода. Вместо этого был создан словарь, в котором заранее расположены списки с фразами и индексами фраз, для создания универсальной логики для всех шести случаев.



### 3.3.2 Данные о заменённых лексических единицах

Метод замены лексических единиц в сравнении с методом изменения порядка слов является намного менее трудоёмким, поскольку в отличие от синтаксического анализа полного предложения ограничивается лишь морфологическим анализом слов.

В качестве результата проведённых изменений возвращается множество с индексами заменённых лексических единиц. При условии успешного преобразования предоставленной пользователем лексической единицы в форму, соответствующей исходной лексической единице, индекс добавляется в множество. Индекс соответствует позиции слова во входящем списке со всеми словами предложения.

### 3.3.3 Данные об изолированных слова

Трудоёмкость метода преобразования слов в определённую степень изолированности сравнима с методом замены лексических единиц, поскольку метод преобразования слов в определённую степень изолированности также ограничивается лишь морфологическим анализом слов.

В качестве результата проведённых изменений также, как и в методе замены лексических единиц, возвращается множество с индексами преобразованных слов. При условии, что исходное слово подходит по части речи переданной степени изолированности индекс добавляется в множество.

### 3.3.4 Данные о заменённых маркированных звуках

В качестве результата проведённых изменений возвращается словарь, в котором в качестве ключа используется индекс преобразованного слова, а в качестве значения используется список с индексами изменённых символов в слове. Для получения изменённых символов использовалась функция «ndiff» из библиотеки «difflib». Данная функция позволяет получить разницу двух строк, а также показывает какие символы были удалены, а какие добавлены.

## 3.4 Разработка пользовательского интерфейса

Для наглядной визуальной демонстрации работы алгоритма генерации диалогов с неигровыми персонажами на основе моделируемых языковых

групп и удобного взаимодействия с алгоритмом было необходимо создать приложение с пользовательским интерфейсом.

В качестве разделения функционала пользовательского интерфейса было принято решение создать три вкладки:

- 1) Вкладка для создания и сохранения моделируемой языковой группы;
- 2) Вкладка для демонстрации преобразования текста;
- 3) Вкладка для общения с персонажем сервиса «Character.ai».

#### 3.4.1 Вкладка для создания и сохранения моделируемой языковой группы

Для того, чтобы пользователь мог удобно взаимодействовать с моделируемыми языковыми группами, было принято решение создать вкладку для создания и сохранения моделируемых языковых групп (Рисунок 3.8). Пользовательский интерфейс обладает минимальным набором требуемых полей ввода. После заполнения всех необходимых пользователю полей моделируемую языковую группу можно сохранить. Сохранение происходит с помощью записи данных в JSON файл (Рисунок 3.7). При отсутствии JSON файла создаётся новый.

```
{
  "language_group_name": "Двойной звук",
  "word_order": 0,
  "lexical_units": {
    "слово": "понятие",
    "истина": "достоверность"
  },
  "isolation_degree": 0,
  "labeled_sounds": {
    "г": "гг",
    "щ": "сч",
    "чт": "ш"
  }
},
```

Рисунок 3.7 – Сохранённая моделируемая языковая группа в JSON файле

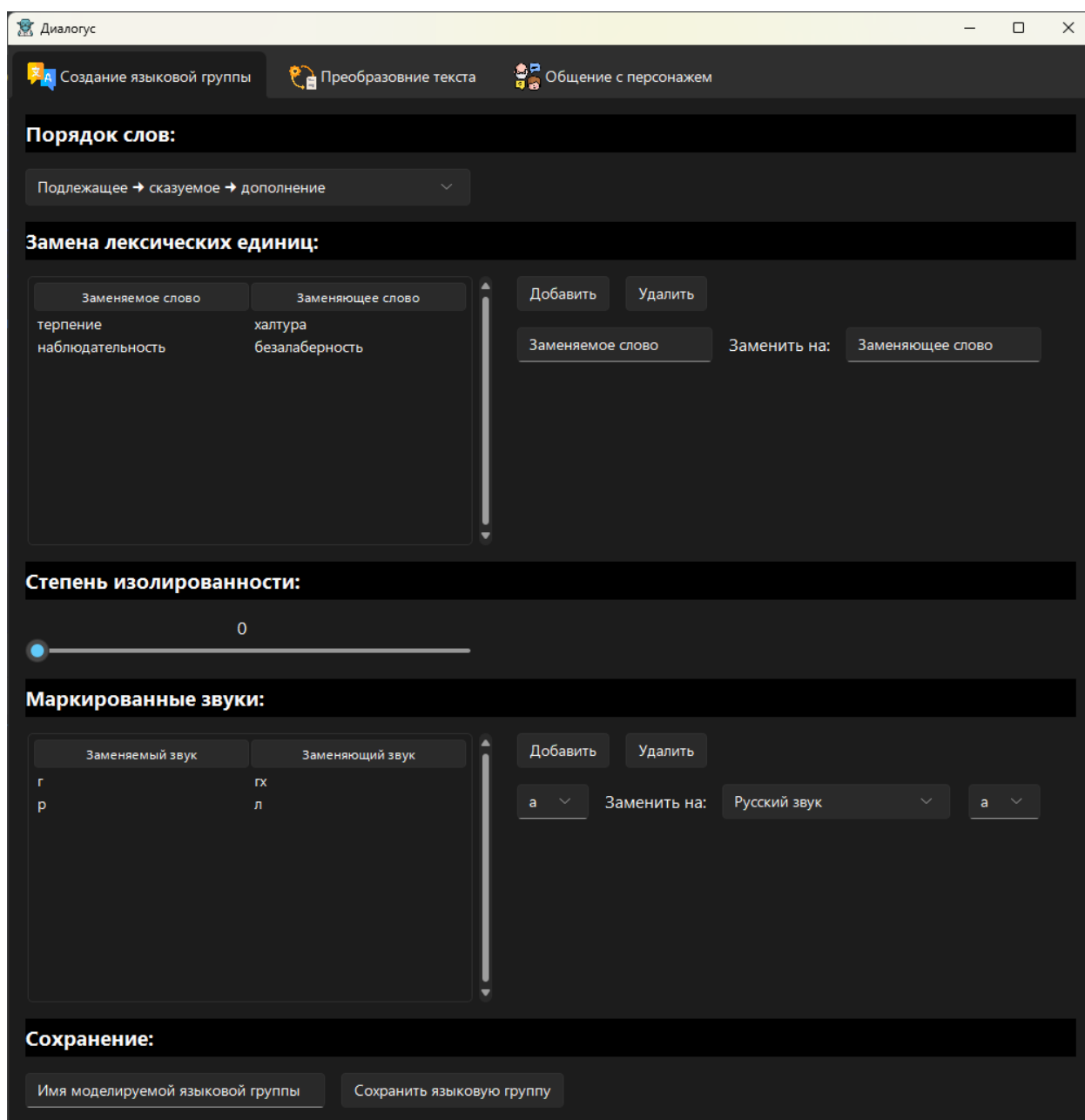


Рисунок 3.8 – Вкладка для создания и сохранения моделируемой языковой группы

### 3.4.2 Вкладка для демонстрации преобразования текста

Для наглядной визуальной демонстрации работы алгоритма генерации диалогов с неигровыми персонажами на основе моделируемых языковых групп и изменённых алгоритмом данных была создана вкладка для демонстрации преобразования текста (Рисунок 3.9). Левая часть данной вкладки очень похожа на вкладку для создания моделируемой языковой группы, но все элементы расположены более компактно для того, чтобы вместить большие текстовые поля в правой части вкладки.

В качестве исходных настроек можно выбрать одну из сохранённых моделируемых языковых групп. После выбора сохранённой моделируемой группы все поля ввода заполнятся соответствующими параметрами. Данные параметры можно менять в реальном времени для более удобного взаимодействия с преобразованием текста.

В правой части вкладки располагаются легенда проведённых преобразований, текстовое поле для вывода результата, текстовое поле для ввода исходного текста и кнопка для отправки запроса на преобразование текста.

После преобразования текста можно увидеть наглядную разницу исходного текста и преобразованного текста за счёт цветного оформления. Все полученные данные о проведённых преобразованиях используются при выводе преобразованного текста.

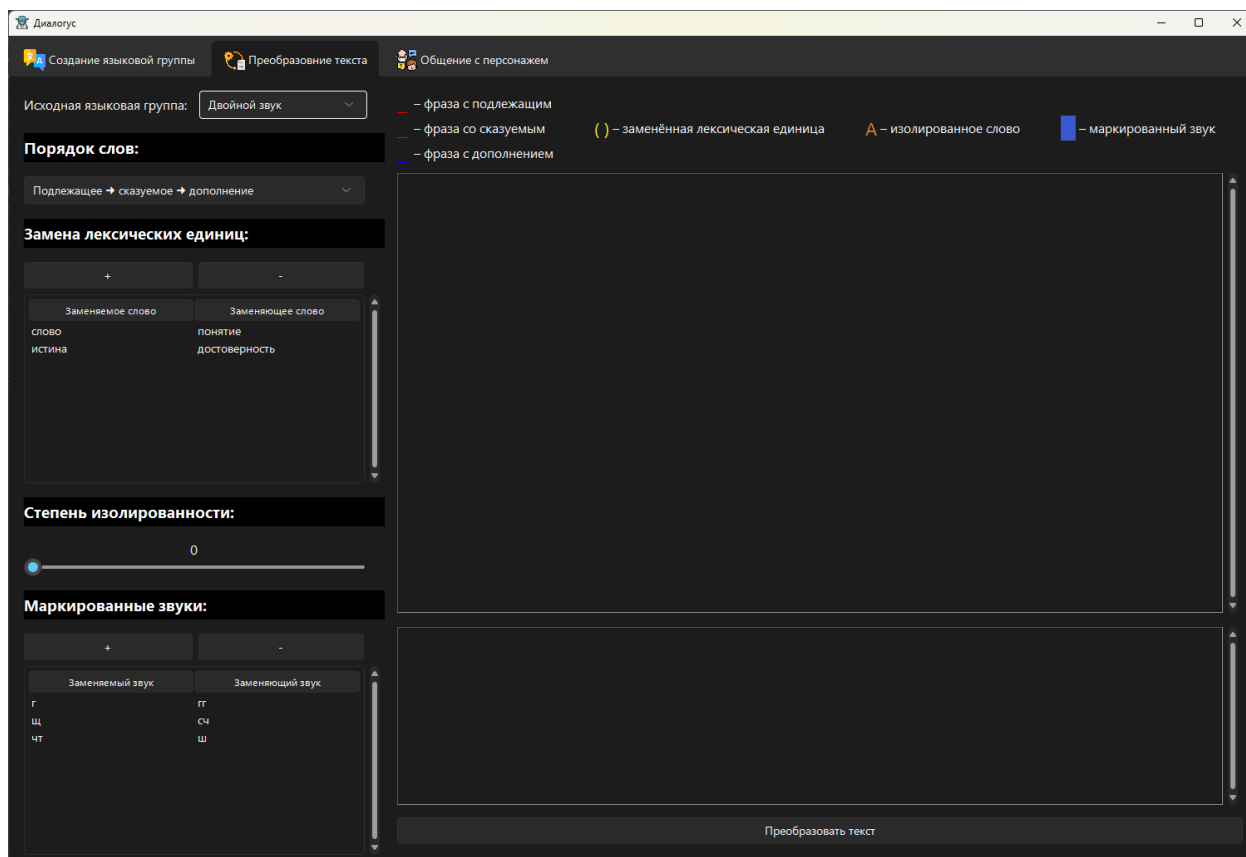


Рисунок 3.9 – Вкладка для демонстрации преобразования текста

### 3.4.3 Вкладка для общения с персонажем сервиса «Character.ai»

В качестве примера интеграции алгоритма была создана вкладка для общения с персонажем сервиса «Character.ai».

На данной вкладке располагаются поле выбора сохранённой моделируемой группы, токен персонажа сервиса «Character.ai», поле вывода результата, поле ввода вопроса персонажу и кнопка для отправки запроса на генерацию ответа и его последующего преобразования (Рисунок 3.11).

Изначально на вкладке отсутствуют поле ввода вопроса и кнопка для отправки запроса, поскольку есть шанс, что с персонажем сервиса «Character.ai» не получится связаться. На данном этапе присутствует кнопка «Начать общение» для связи с персонажем (Рисунок 3.10).

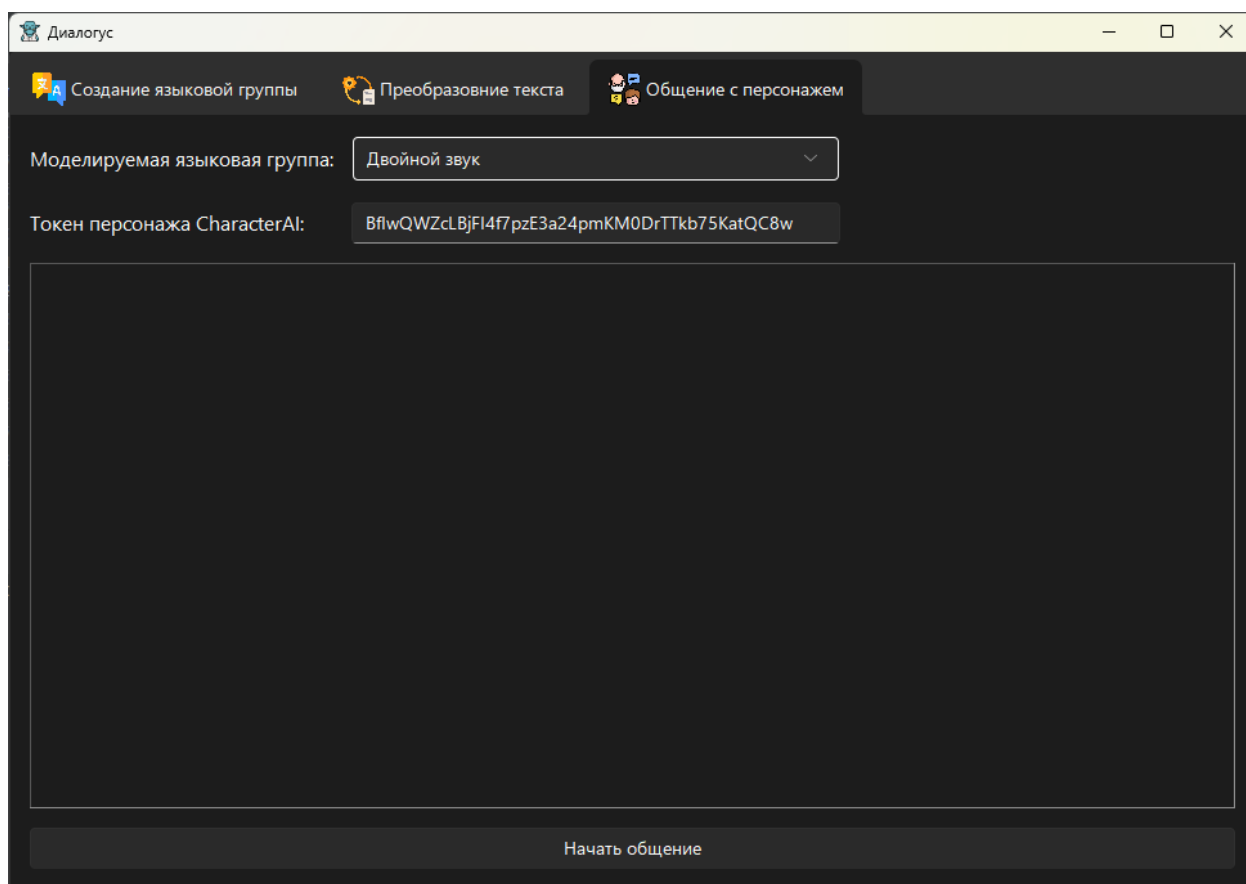


Рисунок 3.10 – Начальное состояние вкладки для общения с персонажем сервиса «Character.ai»

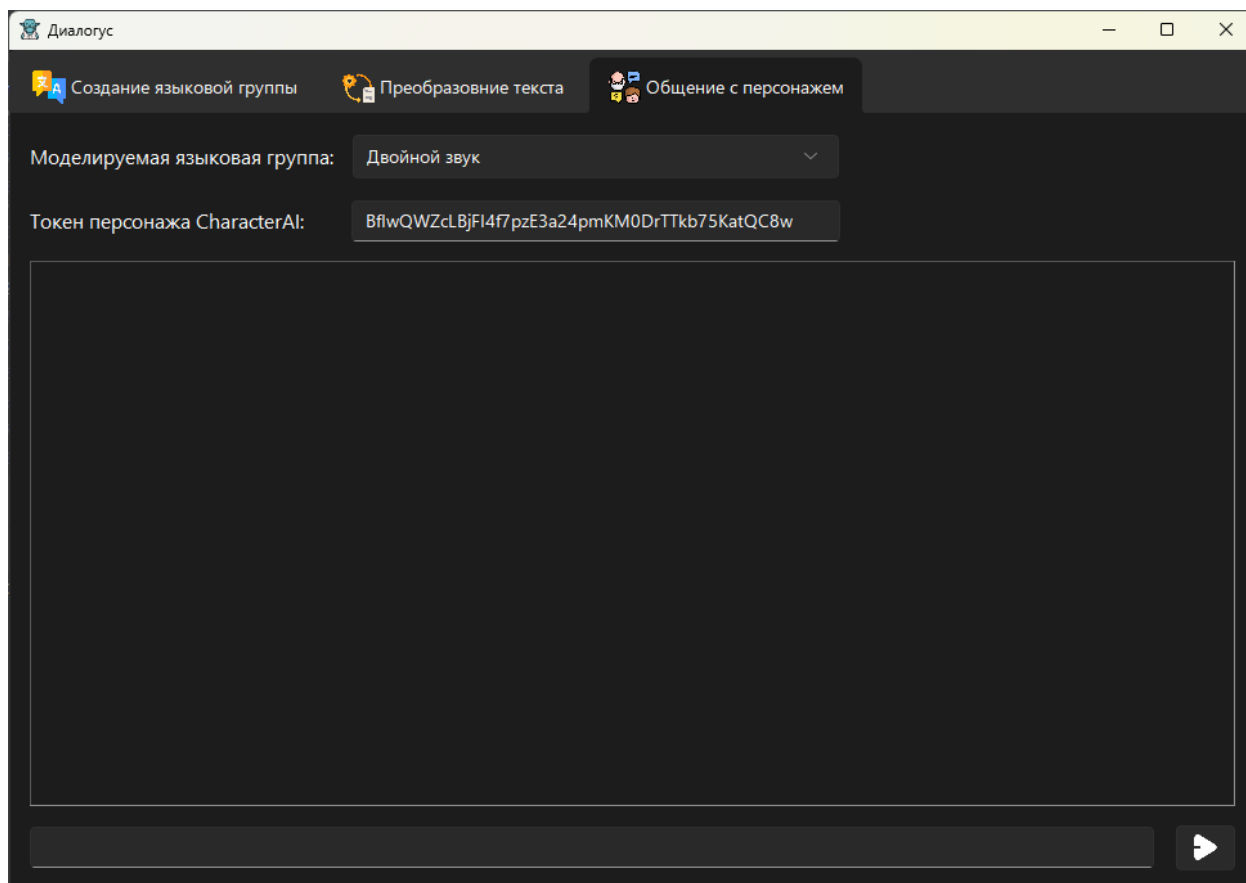


Рисунок 3.11 – Состояние вкладки для общения с персонажем сервиса «Character.ai» при успешном соединении с персонажем сервиса

## 4 ЭКСПЕРИМЕНТАЛЬНЫЕ ИССЛЕДОВАНИЯ АЛГОРИТМА

### 4.1 Оценка результатов работы алгоритма

#### 4.1.1 Метод изменения порядка слов

Несмотря на то, что метод изменения порядка слов использует синтаксический анализатор, который анализирует весь текст (Рисунок 4.1), а не отдельные слова, попавшие под определённые условия проверки, есть предположение, что данный метод будет не самым трудозатратным из всех четырёх методов преобразования текста.

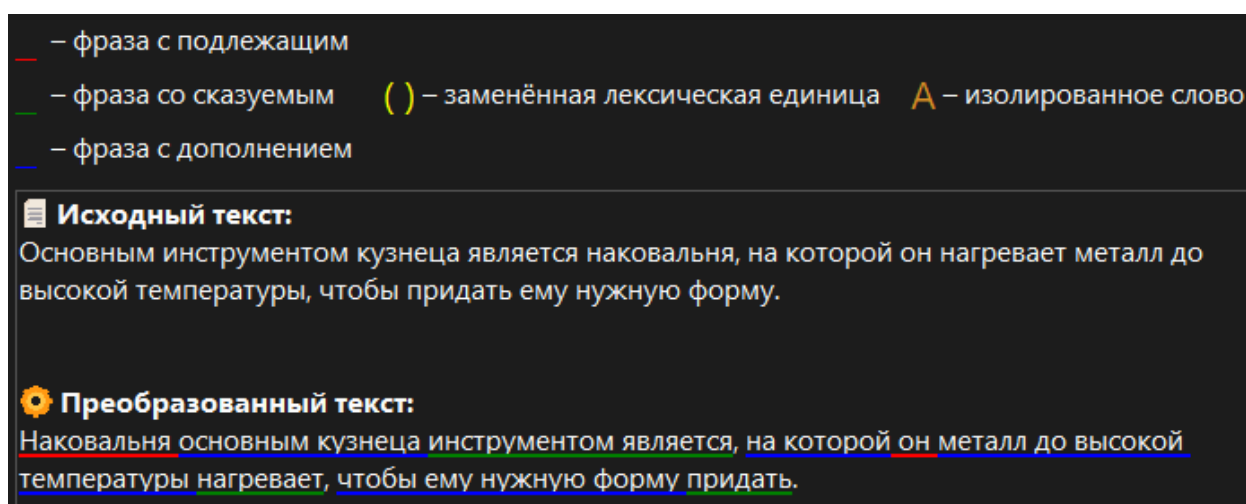


Рисунок 4.1 – Преобразование текста к порядку слов «подлежащее – дополнение – сказуемое»

##### 4.1.1.1 Точность метода изменения порядка слов

Точность метода изменения порядка слов, в отличие от остальных методов, сложно определить с большой точностью, поскольку данный метод не является какой-либо моделью, обучающейся на большом количестве данных. Следуя из этого, точность метода изменения порядка слов была вычислена приблизительно, поскольку выборка была довольно мала из-за её ручного создания. Выборка создавалась вручную, поскольку практически невозможно подобрать необходимое количество текстов со всеми шестью видами порядков слов, так как некоторые порядки слов никто не использует в русском языке.

Приблизительная точность метода изменения порядка слов равняется 73 процентам.

#### 4.1.1.2 Время выполнения метода изменения порядка слов

Время выполнения метода изменения порядка слов напрямую зависит от общего количества токенов в отправленном запросе. Причём выбранный из шести вариантов порядок слов практически никак не влияет на итоговое время выполнения метода, поскольку самой затратной частью метода является синтаксический анализ текста.

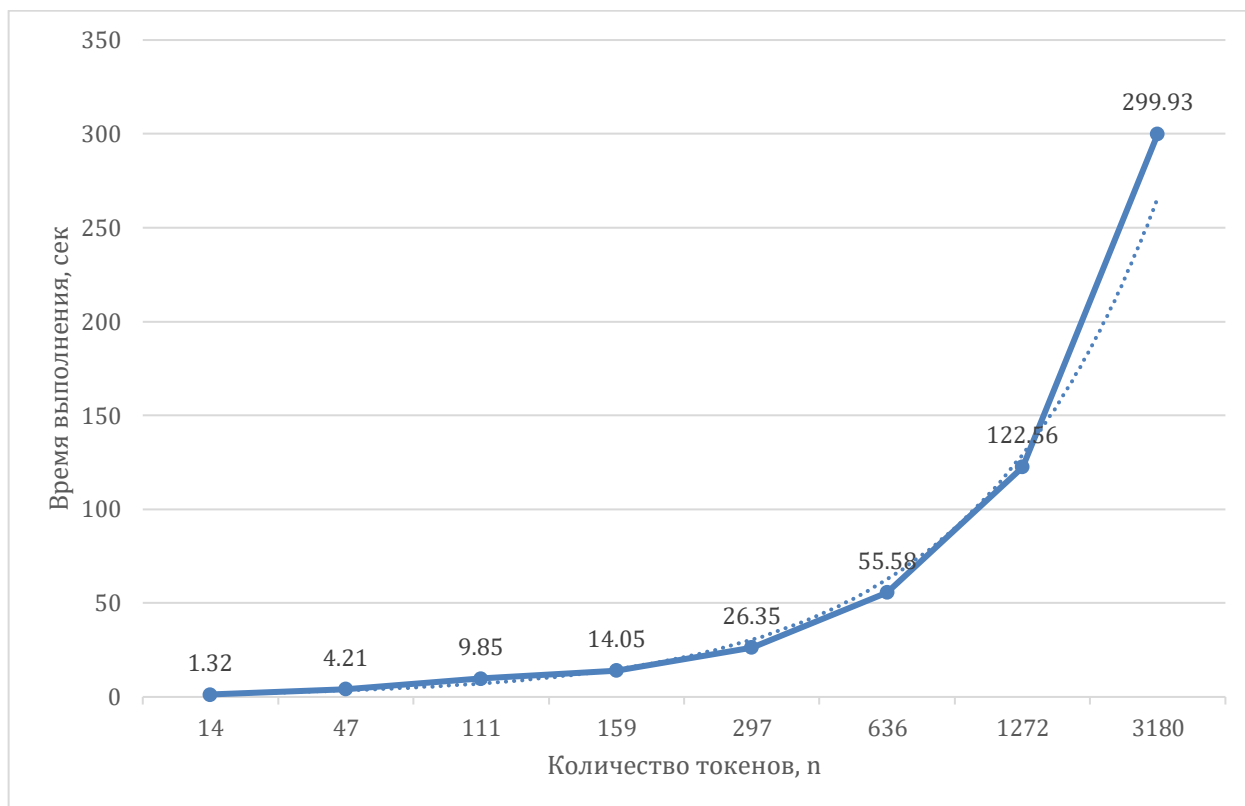


Рисунок 4.2 – Зависимость времени выполнения метода изменения порядка слов от количества токенов в запросе

Исходя из графика зависимости времени (Рисунок 4.2) от количества токенов и его линии тренда, можно сказать, что рост времени похож на экспоненциальный.

Среднее время, затрачиваемое на обработку одного токена, приблизительно равняется 0,09 секундам.

#### 4.1.2 Метод замены лексических единиц

Можно предположить, что самым трудозатратным методом окажется метод замены лексических единиц. Так может произойти, поскольку в методе замены лексических единиц количество операций на один токен может



доходить до трёх. В большинстве случаев количество операций на токен будет равняться двум, но никогда не будет равняться одной операции. Контекстная лемматизация и морфологический анализ всех слов будет выполнен в любом случае, иначе будет недостаток данных для более точного преобразования заменяющей лексической единицы. А вот постановка слова в определённую форму является опциональной операцией морфологического анализатора (Рисунок 4.3).

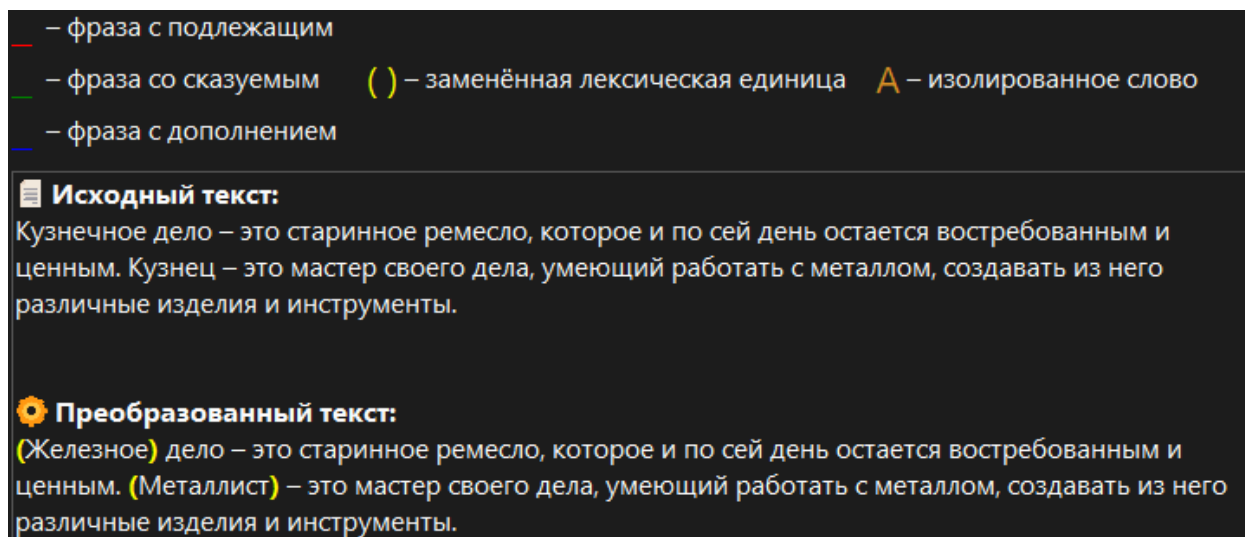


Рисунок 4.3 – Успешная замена лексических единиц с набором «кузнечный – железный» и «кузнец – металлист»

#### 4.1.2.1 Точность метода замены лексических единиц

Точность метода лексических единиц по своей сути будет очень близка к точности морфологического анализатора Rymorphy2, которая находится в диапазоне 70–80 процентов [20]. Точность метода всё же не равна точности морфологического анализатора, поскольку, в методе также используется контекстный лемматизатор, что в какой-то степени способно увеличить точность метода замены лексических единиц примерно на 5 процентов.

#### 4.1.2.2 Время выполнения метода замены лексических единиц

Время выполнения метода замены лексических единиц практически напрямую зависит от общего количества токенов в отправленном запросе, поскольку морфологический анализ будет произведён совершенно на всех словах в тексте, иначе не предоставится возможность проверить соответствие

проанализированного слова слову из предоставленных пользователем параметров. Также помимо морфологического анализатора в данном методе используется контекстный лемматизатор для увеличения точности метода. Контекстный лемматизатор может занимать приличную часть времени выполнения метода. Помимо количества токенов в запросе, время выполнения также зависит от количества слов, необходимых для преобразования в определённую форму. Иными словами, время выполнения метода также зависит от количества заменяемых лексических единиц.

При желании, в методе замены лексических единиц можно не использовать контекстный лемматизатор, что может существенно уменьшить время выполнения метода, но, с другой стороны, это может привести и к заметному уменьшению точности метода вплоть до 5 процентов.

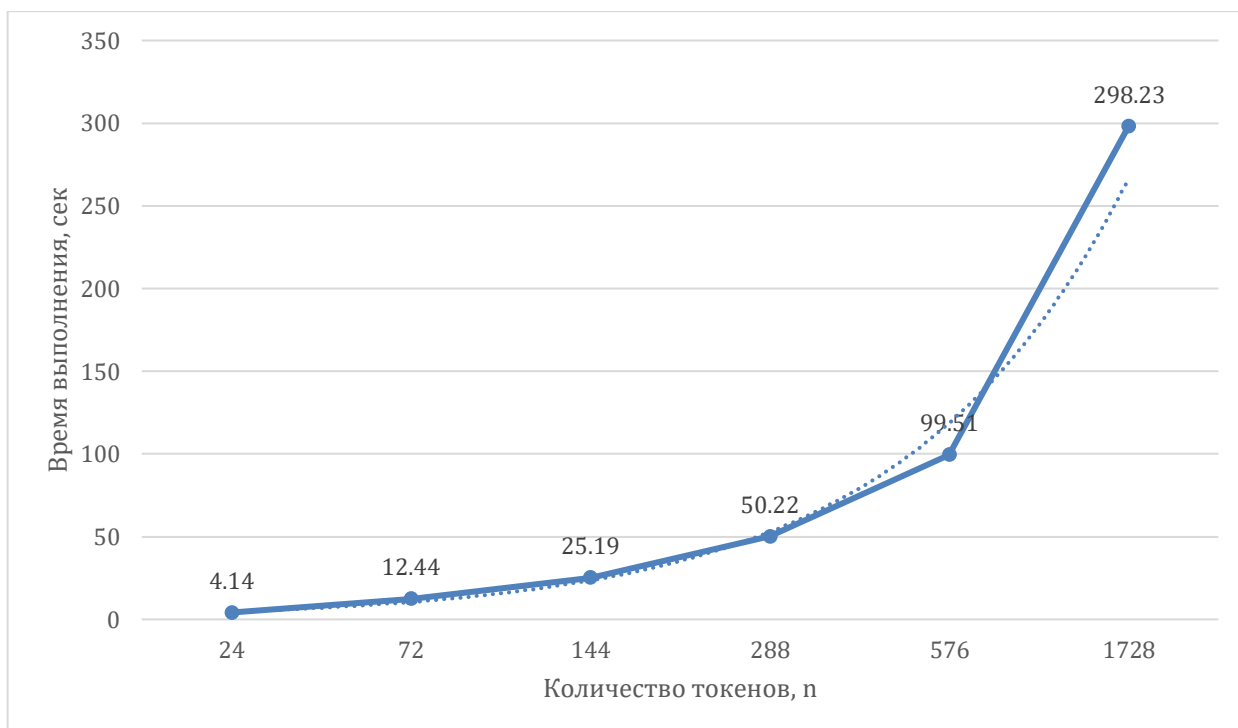


Рисунок 4.4 – Зависимость времени выполнения метода замены лексических единиц от количества токенов в запросе

Исходя из графика зависимости времени от количества токенов (Рисунок 4.4) и его линии тренда, можно сказать, что рост времени работы метода замены лексических единиц похож на экспоненциальный.

Среднее время, затрачиваемое на обработку одного токена, приблизительно равняется 0,17 секундам.

#### 4.1.3 Метод преобразования текста в степень изолированности

Поскольку метод преобразования текста в степень изолированности ограничивается лишь одной операцией морфологического анализа на каждый токен (Рисунок 4.5), есть предположение, что трудозатратность данного метода будет находиться между методом изменения порядка слов и методом замены маркированных звуков.

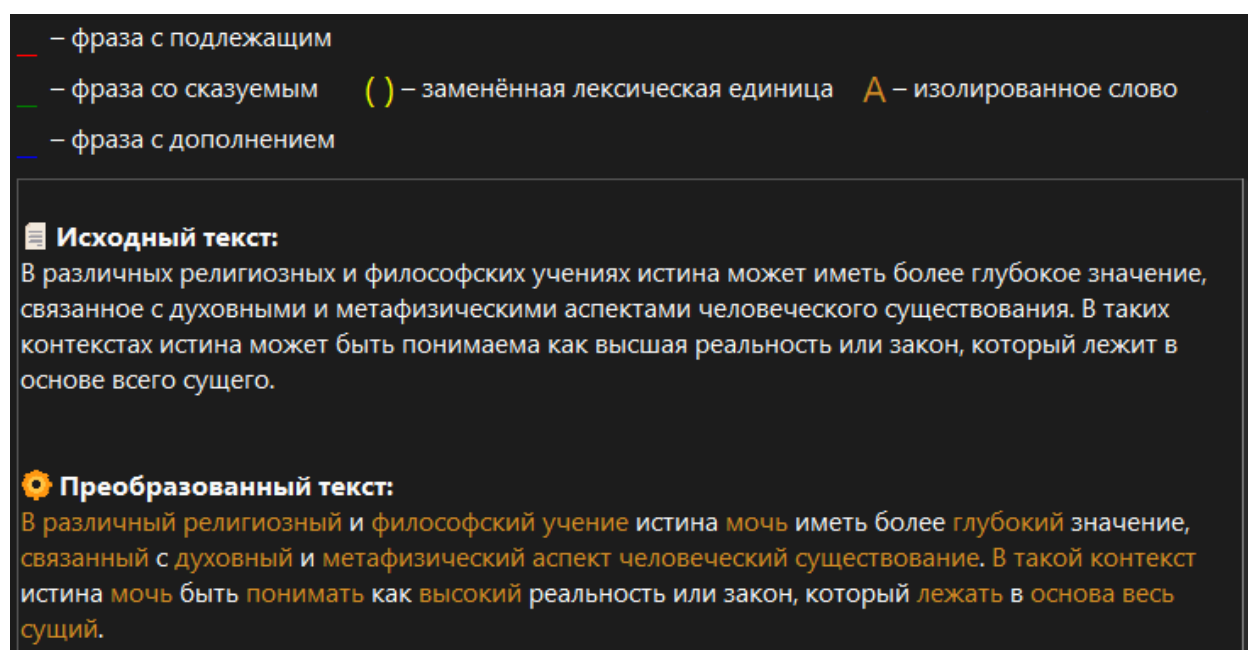


Рисунок 4.5 – Успешное преобразования текста в третью степень изолированности

##### 4.1.3.1 Точность метода преобразования текста в степень изолированности

Точность метода преобразования текста в степень изолированности по своей сути будет очень близка к точности морфологического анализатора `rumorphy2`, которая находится в диапазоне 70–80 процентов [20]. В данном методе используется только морфологический анализатор `rumorphy2`, поэтому точностью метода можно считать 70–80 процентов.

#### 4.1.3.2 Время выполнения метода преобразования текста в степень изолированности

Также, как и метод замены лексических единиц, метод преобразования текста в степень изолированности практически напрямую зависит от общего количества токенов в отправленном запросе, поскольку морфологический анализ будет произведён совершенно на всех словах в тексте. Отличие данного метода заключается в том, что в нём не используется контекстная лемматизация, что существенно экономит время выполнения. Также, помимо этого, главной задачей данного метода является постановка слова, соответствующего определённой части речи, в нормальную форму. Все эти данные получаются при морфологическом анализе, поэтому для одного токена формально применяется всего одна операция, в то время как в методе замены лексических единиц операций может выходить вплоть до трёх на каждый токен.

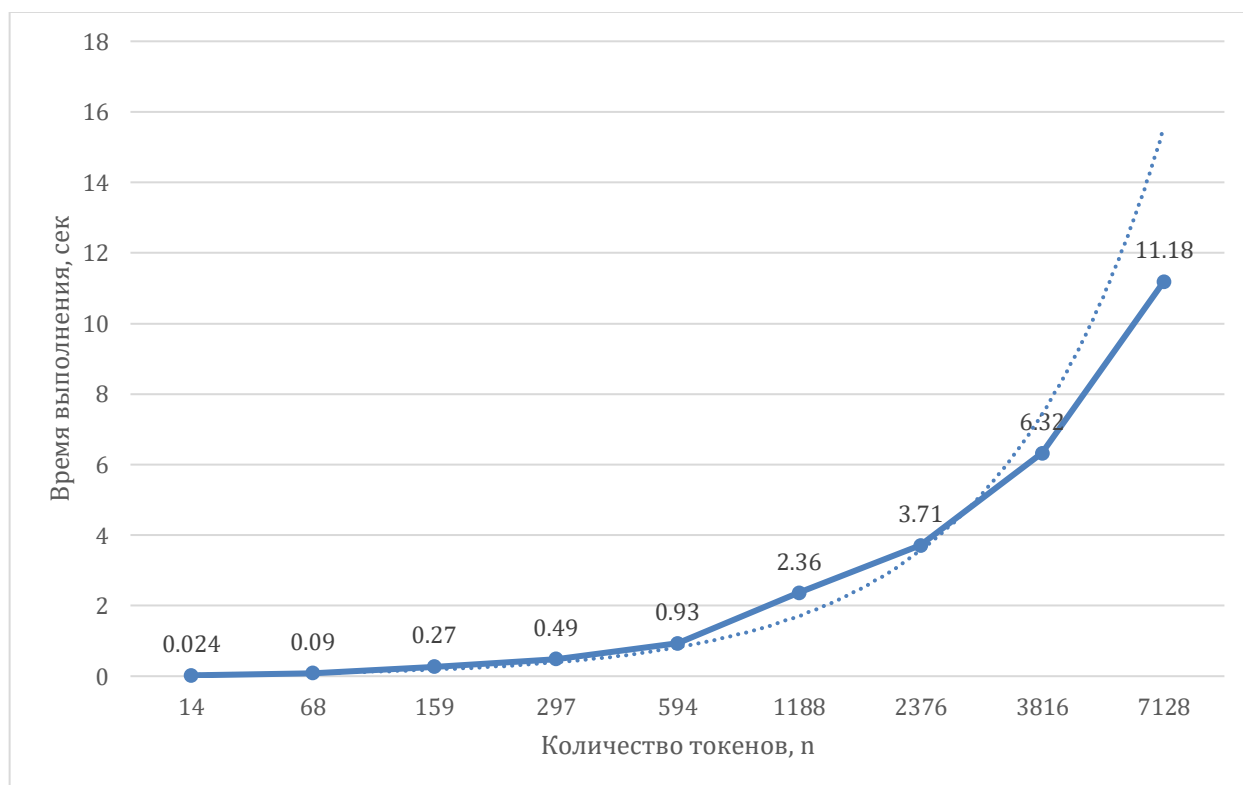


Рисунок 4.6 – Зависимость времени выполнения метода преобразования текста в степень изолированности от количества токенов в запросе

Исходя из графика зависимости времени от количества токенов (Рисунок 4.6) и его линии тренда, можно сказать, что рост времени работы метода преобразования текста в степень изолированности похож на экспоненциальный.

Среднее время, затрачиваемое на морфологический анализ одной лексической единицы, приблизительно равняется 0,001 секундам, что является наглядным примером того, что количество операций на один токен существенно снижает время выполнения метода.

#### 4.1.4 Метод замены маркированных звуков

Есть предположения, что метод замены маркированных звуков является наименее трудозатратным, поскольку логика, присутствующая в нём, максимально тривиальна (Рисунок 4.7).

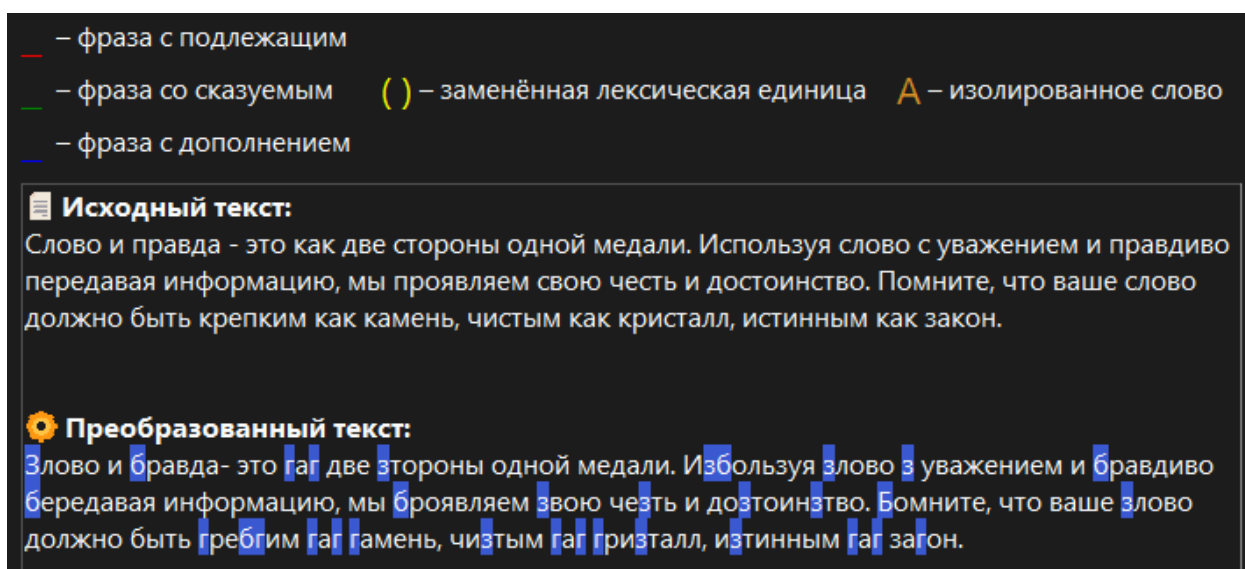


Рисунок 4.7 – Замена маркированных звуков «с – з», «п – б», «к – г»

##### 4.1.4.1 Точность метода замены маркированных звуков

Поскольку метод замены маркированных звуков использует базовый функционал по работе со строками, и не использует какую-либо модель, точность данного метода приблизительно равна 100 процентам. Метод может не сработать только в том случае, если на вход были предоставлены не поддерживаемые операционной системой символы.

#### 4.1.4.2 Время выполнения метода замены маркированных звуков

Время выполнения метода замены маркированных звуков является ничтожно малым на фоне остальных трёх методов преобразования текста. В методе используется базовый функционал по работе со строками в Python, поэтому скорость выполнения максимально мала. Метод является оптимизированным в своей текущей форме, а какие-либо оптимизации метода не принесут итогового результата, поскольку время выполнения данного метода практически никак не повлияет на итоговое время работы всего алгоритма.

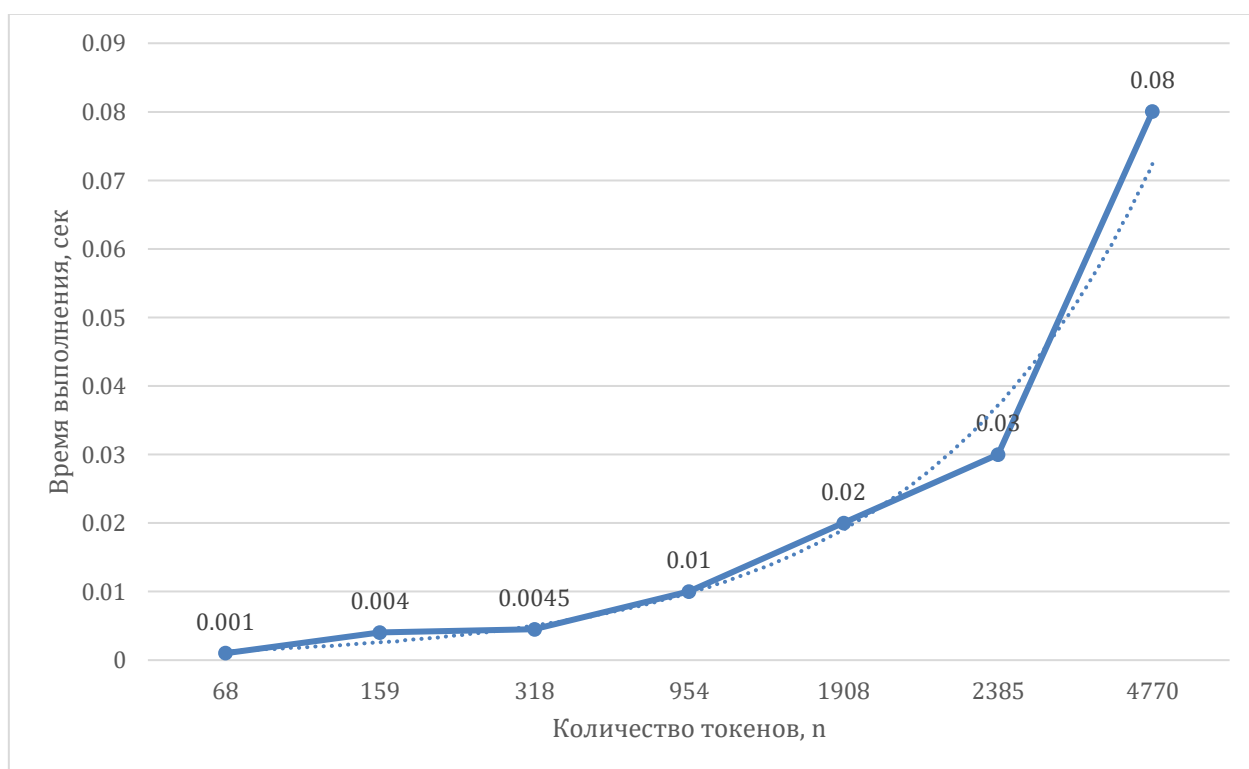


Рисунок 4.8 – Зависимость времени выполнения метода замены маркированных звуков от количества токенов в запросе

Исходя из графика зависимости времени от количества токенов (Рисунок 4.8) и его линии тренда, можно сказать, что рост времени работы метода замены маркированных звуков, как и в других случаях, похож на экспоненциальный.

Среднее время, затрачиваемое на обработку одного токена, приблизительно равняется  $1,42 \times 10^{-5}$  секундам.

#### 4.1.5 Сравнение всех методов преобразования

Для наглядности трудозатрат всех методов был создан единый график со всеми методами преобразования текста при одинаковых условиях в количестве токенов в запросе (Рисунок 4.9).

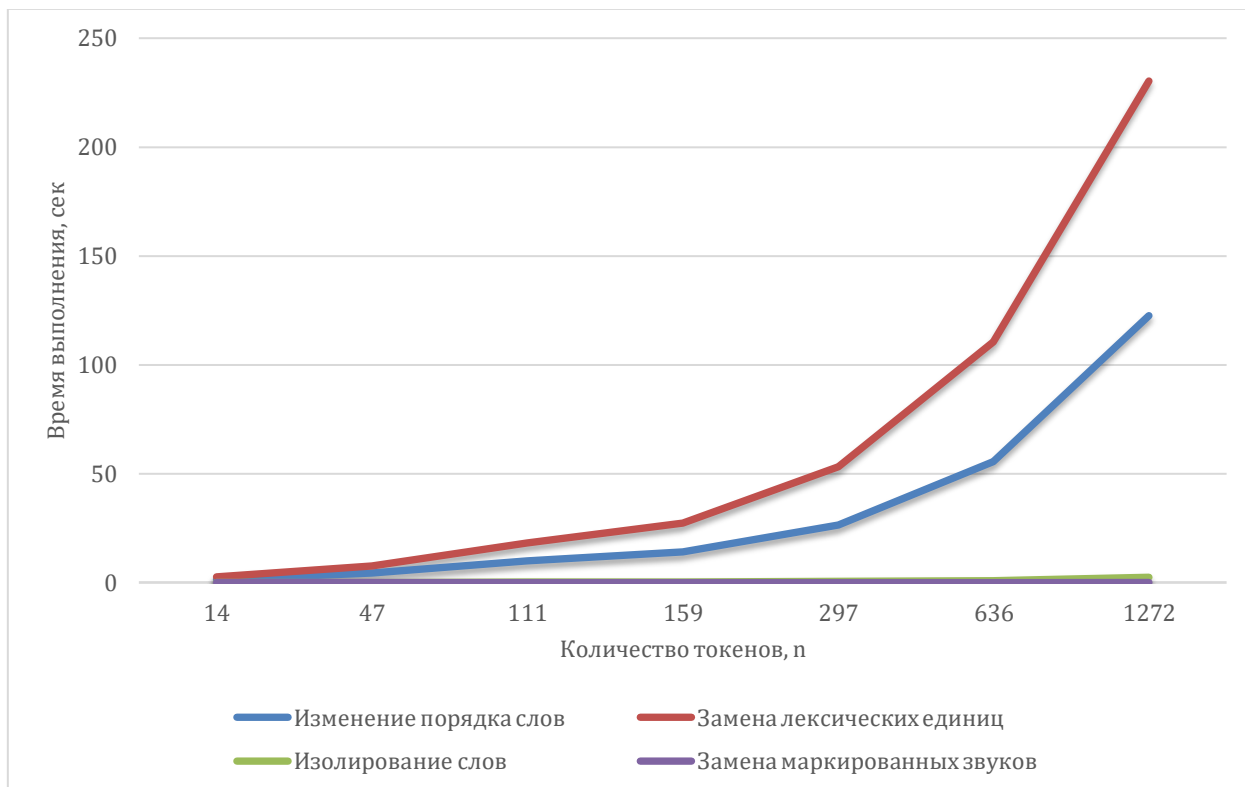


Рисунок 4.9 – Сравнение зависимости время выполнения от количества токенов четырёх методов преобразования

Исходя из графика (Рисунок 4.9), можно сказать, что, как и предполагалось ранее, самым трудозатратным методом оказался метод замены лексических единиц из-за своего количества операций на каждый токен. Метод изменения порядка слов также является довольно сильно трудозатратным. А вот трудозатратность остальных методов приемлемо мала.

#### 4.1.6 Суммарная работа алгоритма

Для оценки работы алгоритма было использовано не более 700 токенов в запросе, поскольку большее количество токенов не будет являться релевантным для данного алгоритма и его возможностей.

#### 4.1.6.1 Итоговая точность работы алгоритма

Приблизительная точность работы алгоритма генерации диалогов с неигровыми персонажами на основе моделируемых языковых групп равна 80 процентам.

#### 4.1.6.2 Итоговое время выполнения алгоритма

Во время составления графика зависимости итогового времени выполнения алгоритма от количества токенов в запросе значение максимального количества токенов было использовано не настолько большое, как на графиках с зависимостью времени других методов. Сделано это было из-за того, что данный алгоритм в первую очередь рассчитан на преобразования диалоговых текстов, количество токенов в которых не велико.

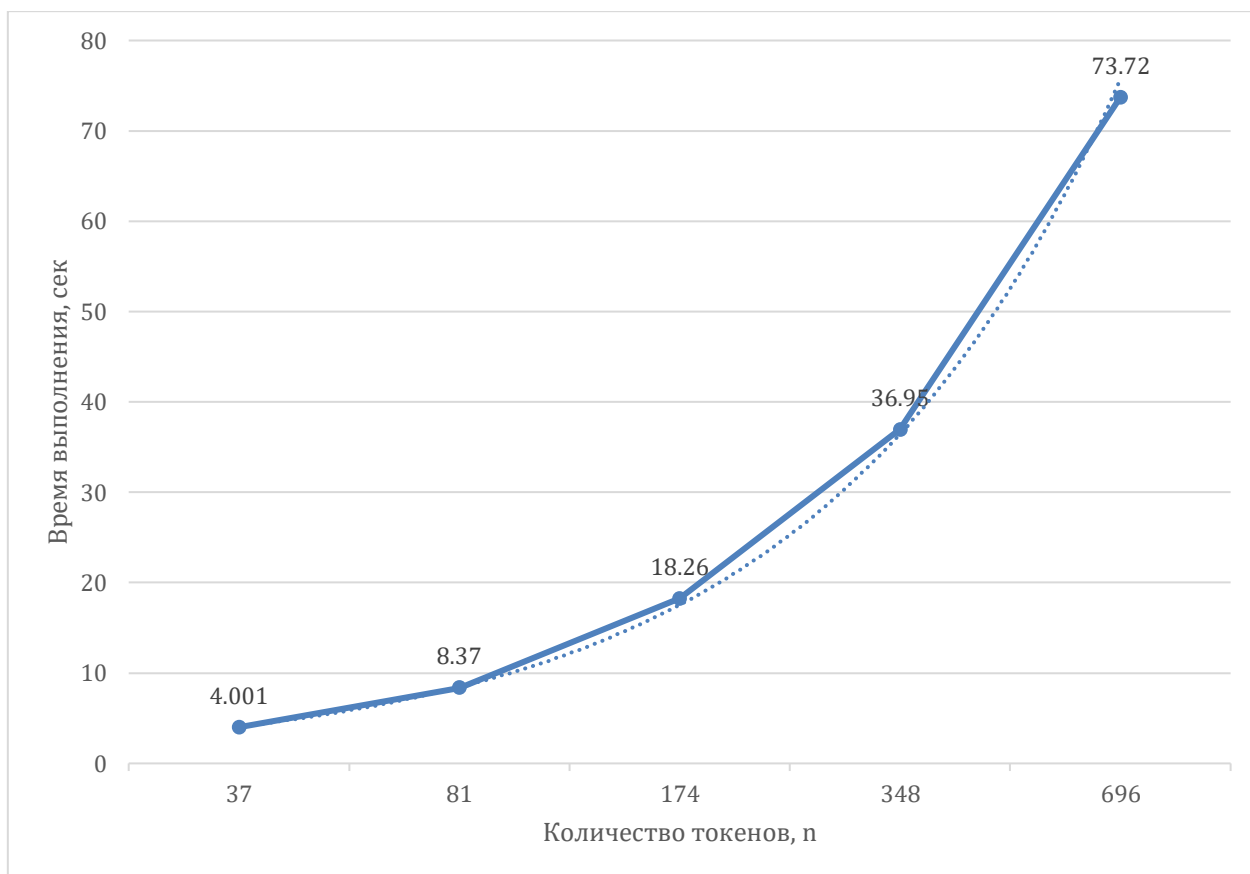


Рисунок 4.10 – Зависимость итогового времени выполнения алгоритма от количества токенов в запросе

Исходя из графика зависимости итогового времени выполнения алгоритма от количества токенов в запросе (Рисунок 4.10) и его линии



тренда, можно сказать, что рост времени работы алгоритма, как и во всех методах, похож на экспоненциальный.

Среднее время, затрачиваемое на обработку одного токена во время работы всего алгоритма, приблизительно равняется 0,104 секундам.

#### **4.2 Демонстрация примеров использования алгоритма**

Демонстрации примеров будут отображены с помощью вкладки для общения с персонажем сервиса «Character.ai», поскольку это является хорошим примером интеграции разработанного алгоритма преобразования текста на основе моделируемых языковых групп. Все примеры будут приведены с участием созданного персонажа орка в сервисе «Character.ai» для наглядного примера того, как кардинально может меняться восприятие диалога в зависимости от разных параметров алгоритма.

Первым интересным примером является моделируемая языковая группа, которая похожа на казахский язык (Рисунок 4.11). В данной моделируемой языковой группе заменяются русские звуки за твёрдые казахские звуки для создания грубой речи. В качестве порядка слов используется «подлежащее – дополнение – сказуемое», поскольку именно так работает синтаксис в казахском языке. В качестве степени изолированности была выбрана первая степень. В таком случае будут лемматизированы глаголы, что может подчеркнуть, что персонаж не является носителем русского языка. Набор лексических единиц для замены в данной моделируемой языковой группе является пустым.

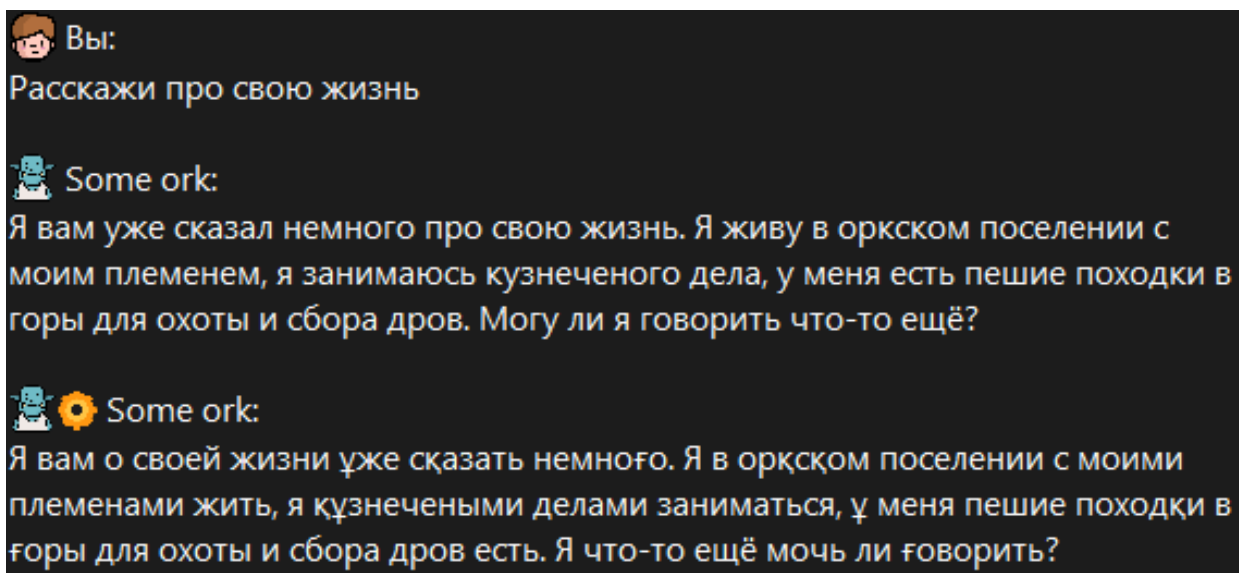


Рисунок 4.11 – Общение с орком с использованием моделируемой языковой группы, похожей на казахский язык

Вторым примером является группа, которая пытается создать эффект плохо говорящего на русском языке заики (Рисунок 4.12). В качестве порядка слов используется «сказуемое – подлежащее – дополнение» для создания эффекта того, что персонаж плохо размышляет на неродном языке. В заменяемых лексических единицах был использован набор «кузнец – железо» и «кузнечный – железный» для создания чувства, что русский лексикон персонажа довольно мал. Степень изолированности была установлена вторая, чтобы подчеркнуть проблемы со склонениями на русском языке. Для создания эффекта заикания и проблемы с произношением некоторых звуков был использован определённый набор (Таблица 4).

Таблица 4 – Набор маркированных звуков для второй моделируемой группы

Заменяемый звук	Заменяющий звук
«Г»	«ГГГ»
«Щ»	«сч»
«Я»	«йа»
«Ю»	«йу»
«Ф»	«ффф»
«Н»	«ннн»

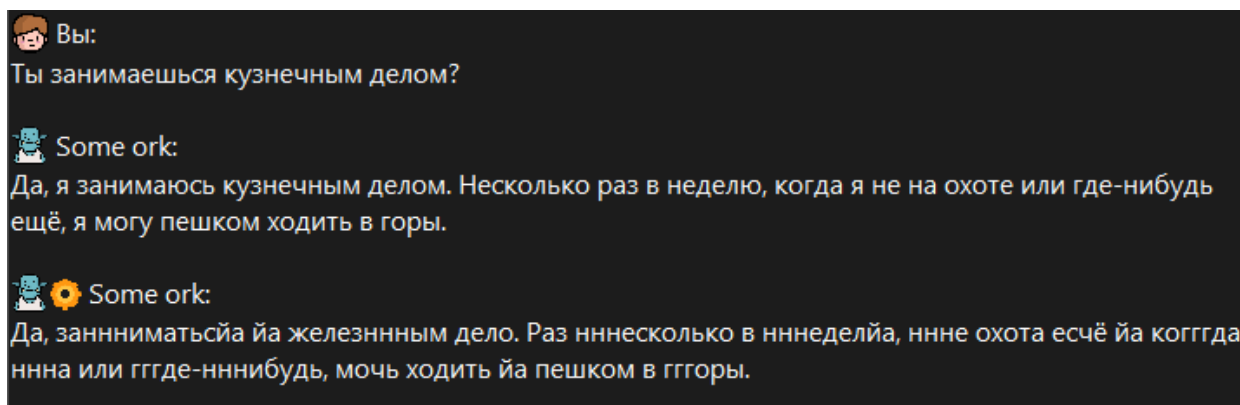


Рисунок 4.12 – Общение с орком с использованием моделируемой языковой группы плохо говорящего заики

### 4.3 Описание дальнейших планов развития алгоритма

В качестве дальнейших планов развития алгоритма можно выделить несколько пунктов:

1) Улучшение функций преобразования текста для получения наилучших выходных данных.

2) Улучшение метода изменение порядка слов за счёт улучшенной модели на русском языке, точность которой будет превышать новостные модели, предоставляемые разработчиками spaCy.

3) Улучшение метода замены лексических единиц за счёт расширения функционала морфологического анализатора для возможности ставить слова, которые отсутствуют в словаре анализатора, в определённую форму.

4) Добавление параметра эргативной стратегии кодирования глагольных актантов в моделируемую языковую группу.

5) Использование алгоритма во время разработки игрового прототипа с уклоном в нарратив и с большим количеством диалогов.

## ЗАКЛЮЧЕНИЕ

В первом разделе работы был проведён анализ предметной области для изучения современных возможностей в области генерации текста и обработки естественного языка. Были указаны возможности и ограничения технологий генерации текста.

Во втором разделе работы была сформирована теоретическая модель алгоритма с требованиями и ограничениями системы.

В третьем разделе был описан процесс практической имплементации отдельных методов преобразования текста, а также приведение их к цельному алгоритму. Был продемонстрирован созданный пользовательский интерфейс для удобного взаимодействия с алгоритмом и демонстрации.

В четвёртом разделе были проведены экспериментальные исследования алгоритма с демонстрацией примеров работы, точности и времени выполнения алгоритма. Также были описаны дальнейшие планы развития алгоритма.

На основе проведённой работы можно сделать следующие выводы:

1) Анализ предметной области дал понять, что большие языковые модели безусловно являются мощным инструментом для генерации и обработки текста, но имеют свои ограничения, которые не позволяют использовать их для обработки текста в текущей работе.

2) Основой для создания алгоритма стали несколько библиотек для обработки естественного языка, поскольку они позволяют иметь наибольший контроль в обработке текста, что является главным фактором в разработке алгоритма генерации диалогов с неигровыми персонажами на основе моделируемых языковых групп.

3) Практическая имплементация алгоритма была представлена с использованием сервиса «Character.ai, поскольку сервис предоставляет необходимый функционал для демонстрации алгоритма без нужды собственной имплементации. Разработка пользовательского интерфейса для алгоритма привела к тому, что понадобилось расширить функционал всех

методов преобразования текста. Следовательно, функционал четырёх методов преобразования текста был дополнен функционалом для получения данных о проведённых преобразованиях.

4) Экспериментальные исследования помогли сформировать точность, время выполнения и примеры работы имплементированных методов обработки текста. Самым трудозатратным оказался метод замены лексических единиц, поскольку количество операций на каждый токен в данном методе может достигать до трёх. Количество операций на каждый токен можно снизить, но вместе с этим упадёт и точность выходных данных.

5) Итоговое время выполнения алгоритма можно считать приемлемым, поскольку алгоритм не предназначен для полноценного использования во время работы игровых приложений.

6) За счёт архитектуры алгоритма имеется возможность использовать исходный текст, полученный любым образом, а не только сгенерированный сервисом «Character.ai». Исходный текст может быть предоставлен как пользователем, так и любой большой языковой моделью.

7) В качестве будущего развития алгоритма существует возможность имплементировать преобразование текста в эргативную стратегию кодирования глагольных актантов и возможность выставления шанса каждого конкретного преобразования. Также присутствует возможность оптимизации алгоритма для уменьшения итогового времени выполнения всех преобразований текста.

Была достигнута основная цель работы. Все поставленные задачи были успешно выполнены. У алгоритма существуют возможности улучшения, оптимизации и будущего развития.

Текст выпускной квалификационной работы, исходный код и инструкции к запуску опубликованы на GitHub. Доступ к репозиторию можно получить по ссылке <https://github.com/Yar-ick/DialogsGenerationBasedOnSimulatedLanguageGroups>.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Mateas M., Stern A. Façade: An experiment in building a fully-realized interactive drama // Game developers conference. – 2003. – Т. 2. – С. 4-8.
2. Hua M., Raley R. Playing With Unicorns: AI Dungeon and Citizen NLP // DHQ: Digital Humanities Quarterly. – 2020. – Т. 14. – №. 4.
3. Tamkin A. et al. Understanding the capabilities, limitations, and societal impact of large language models // arXiv preprint arXiv:2102.02503. – 2021.
4. Achiam J. et al. Gpt-4 technical report // arXiv preprint arXiv:2303.08774. – 2023.
5. Новое поколение генеративных текстовых нейросетей [Электронный ресурс]. – 2024. – URL: <https://ya.ru/ai/gpt-3> (дата обращения 04.05.2024)
6. Shavrina T., Pisarevskaya D., Malykh V. Building a Bilingual QA-system with ruGPT-3 // International Conference on Analysis of Images, Social Networks and Texts. – Springer, Cham, 2022. – С. 124-136.
7. Li J., Zhang Z., Zhao H. Dialogue-adaptive language model pre-training from quality estimation // Neurocomputing. – 2023. – Т. 516. – С. 27-35.
8. Clark K. et al. Electra: Pre-training text encoders as discriminators rather than generators // arXiv preprint arXiv:2003.10555. – 2020.
9. Alexandr N. et al. Fine-tuning gpt-3 for russian text summarization // Proceedings of the Computational Methods in Systems and Software. – Springer, Cham, 2021. – С. 748-757.
10. Бажин В. А. Тонкая настройка BERT и GPT-3 для решения задачи генерации русскоязычных новостей // Актуальные научные исследования в современном мире. – 2021. – №. 5-2. – С. 43-58.
11. Обработка естественного языка // ИТМО. [Электронный ресурс]. – URL: [https://neerc.ifmo.ru/wiki/index.php?title=Обработка\\_естественного\\_языка](https://neerc.ifmo.ru/wiki/index.php?title=Обработка_естественного_языка) (дата обращения 05.06.2023)
12. Основы Natural Language Processing для текста // Хабр. [Электронный ресурс]. – URL:

<https://habr.com/ru/companies/Voximplant/articles/446738/> (дата обращения 05.06.2023)

13. Mielke S. J. et al. Between words and characters: A brief history of open-vocabulary modeling and tokenization in NLP // arXiv preprint arXiv:2112.10508. – 2021.

14. Nothman J., Qin H., Yurchak R. Stop word lists in free open-source software packages // Proceedings of Workshop for NLP Open Source Software (NLP-OSS). – 2018. – С. 7-12.

15. Khyani D. et al. An interpretation of lemmatization and stemming in natural language processing // Journal of University of Shanghai for Science and Technology. – 2021. – Т. 22. – №. 10. – С. 350-357.

16. Akhmetov I. et al. Highly language-independent word lemmatization using a machine-learning classifier // Computación y Sistemas. – 2020. – Т. 24. – №. 3. – С. 1353-1364.

17. Rajput A. Natural language processing, sentiment analysis, and clinical analytics // Innovation in health informatics. – Academic Press, 2020. – С. 79-97.

18. Al Omran F. N. A., Treude C. Choosing an NLP library for analyzing software documentation: a systematic literature review and a series of experiments // 2017 IEEE/ACM 14th international conference on mining software repositories (MSR). – IEEE, 2017. – С. 187-197.

19. Wang M., Hu F. The application of nltk library for python natural language processing in corpus research // Theory and Practice in Language Studies. – 2021. – Т. 11. – №. 9. – С. 1041-1049.

20. Vissio N. C., Zakharov V. A Disambiguator for Pymorphy2 Morphological Analyzer // IMS. – 2021. – С. 81-88.

21. Gilda S. Notice of Violation of IEEE Publication Principles: Evaluating machine learning algorithms for fake news detection // 2017 IEEE 15th student conference on research and development (SCOREd). – IEEE, 2017. – С. 110-115.

22. Ravshanovna T. M., Abdulakhatovna K. R. Typology // Central Asian Journal of Literature, Philosophy and Culture. – 2022. – Т. 3. – №. 12. – С. 101-107.
23. Davenport M., Hannahs S. J. Introducing phonetics and phonology. – Routledge, 2020.
24. Генидзе Н. К. Вокалический коэффициент как один из важнейших параметров фонетической классификации языков мира // Дискурс. – 2020. – Т. 6. – №. 5. – С. 87-96.
25. Lieber R. Introducing morphology. – Cambridge University Press, 2021.
26. Bańcerowski J., Le Dinh T. Phonetic and morphological coding of minimal syntactic units in isolating languages. – 2012.
27. Febrian G., Novitasari U., Hidayat A. Agglutinative language in bahasa Indonesia // LADU: Journal of Languages and Education. – 2022. – Т. 2. – №. 4. – С. 135-141.
28. Cotterell R. et al. On the complexity and typology of inflectional morphological systems // Transactions of the Association for Computational Linguistics. – 2019. – Т. 7. – С. 327-342.
29. Ladd D. R., Arvaniti A. Prosodic prominence across languages // Annual Review of Linguistics. – 2023. – Т. 9. – С. 171-193.
30. Shibatani M. Syntactic typology // Oxford Research Encyclopedia of Linguistics. – 2021.
31. Anatolievna B. A., Glebovna E. N., Mikhail M. Traces Of The Ergative System In The History Of The Russian Language In Comparison With Sanskrit, Latin And Ancient Greek (Based On The Material Of Infinitive Turns) // Webology (ISSN: 1735-188X). – 2022. – Т. 19. – №. 1.
32. Deo A. On mechanisms by which languages become [nominative-] accusative // On looking into words (and beyond). – 2017. – С. 347.
33. Levshina N. Token-based typology and word order entropy: A study based on Universal Dependencies // Linguistic Typology. – 2019. – Т. 23. – №. 3. – С. 533-572.



34. Koptjevskaja-Tamm M. Approaching lexical typology // From polysemy to semantic change: Towards a typology of lexical semantic associations. – 2008. – C. 3-52.
35. Levshina N. Corpus-based typology: Applications, challenges and some solutions // Linguistic Typology. – 2022. – T. 26. – №. 1. – C. 129-160.
36. Polis S. Diachronic and areal patterns: New applications of the semantic map model in lexical typology // Seminats at the Higher School of Economics. – 2018.
37. Ramalho L. Fluent python. – " O'Reilly Media, Inc.", 2022.
38. Horstmann C. S., Nicaise R. D. Python for everyone. – John Wiley & Sons, 2019.
39. Janda H. K. et al. Syntactic, semantic and sentiment analysis: The joint effect on automated essay evaluation //IEEE Access. – 2019. – T. 7. – C. 108486-108503.