|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ \_\_Информатика и системы управления \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

КАФЕДРА \_\_\_\_Системы обработки информации и управления\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

***К ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЕ***

***НА ТЕМУ:***

***«Модуль морфологического***

***анализа русскоязычного текста»***

Студент \_\_\_ИУ5-83\_\_\_ **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_М.А. Абибок\_\_\_\_\_**

(Группа) (Подпись, дата) (И.О.Фамилия)

Руководитель ВКР **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_В.А. Михеев\_\_\_\_\_\_**

(Подпись, дата) (И.О.Фамилия)

Консультант **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_Ю.Е. Гапанюк \_\_\_**

(Подпись, дата) (И.О.Фамилия)

Нормоконтролер \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_**Ю.Н. Кротов\_\_\_**\_\_

(Подпись, дата) (И.О.Фамилия)

*2021 г.*

# АННОТАЦИЯ

Расчётно-пояснительная записка квалификационной работы бакалавра содержит 53 страницу. С приложениями объем составляет 73 страницы. Работа включает в себя 5 таблиц и 22 иллюстраций. В процессе выполнения было использовано 13 источников.

Цель работы заключается в создании программного модуля морфологического анализа русскоязычного текста. Разрабатываемый модуль должен включать в себя алгоритмы морфологической разметки и лемматизации русскоязычного текста и пользовательский интерфейс взаимодействия с модулем.

Разработка модуля производилась на основании документа «Техническое задание», утвержденного руководителем выпускной работы.

В процессе выполнения квалификационной работы бакалавра были реализованы все требования и модуль соответствует всем сформулированным требованиям. Была составлена расчетно-пояснительная записка.

Пояснительная записка содержит 3 приложения.

# СОДЕРЖАНИЕ

[АННОТАЦИЯ 2](#_Toc74170851)

[СОДЕРЖАНИЕ 3](#_Toc74170852)

[СПИСОК ОБОЗНАЧЕНИЙ И СОКРАЩЕНИЙ 5](#_Toc74170853)

[ВВЕДЕНИЕ 6](#_Toc74170854)

[1. ПОСТАНОВКА ЗАДАЧ РАЗРАБОТКИ 7](#_Toc74170855)

[1.1 Общетехническое обоснование разработки 7](#_Toc74170856)

[1.1.1 Постановка задачи проектирования 7](#_Toc74170857)

[1.1.2 Описание предметной области 9](#_Toc74170858)

[1.1.3 Функциональные задачи системы 11](#_Toc74170859)

[2. КОНСТРУКТОРСКО-ТЕХНОЛОГИЧЕСКАЯ ЧАСТЬ 12](#_Toc74170860)

[2.1 Выбор программных средств 12](#_Toc74170861)

[2.2 Выбор аппаратных средств 15](#_Toc74170862)

[2.3 Разработка алгоритмов 15](#_Toc74170863)

[2.3.1 Загрузка корпуса 16](#_Toc74170864)

[2.3.2 Обучение CRF модели 21](#_Toc74170865)

[2.3.3 Морфологический анализ 22](#_Toc74170866)

[2.3.4 Оценка полученной модели 23](#_Toc74170867)

[2.3.5 Разработка веб-интерфейса 29](#_Toc74170868)

[3. ИССЛЕДОВАТЕЛЬСКАЯ ЧАСТЬ 37](#_Toc74170869)

[3.1 Морфологические анализаторы для русского языка 37](#_Toc74170870)

[3.1.1 AOT 39](#_Toc74170871)

[3.1.2 TreeTagger 39](#_Toc74170872)

[3.1.3 PyMorphy2 40](#_Toc74170873)

[3.1.4 MyStem 41](#_Toc74170874)

[3.2 Сравнение морфологических анализаторов 42](#_Toc74170875)

[3.3 Метод условных случайных полей 46](#_Toc74170876)

[ЗАКЛЮЧЕНИЕ 51](#_Toc74170877)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 52](#_Toc74170878)

[ПРИЛОЖЕНИЕ А Графическая часть дипломного проекта 52](#_Toc74170879)

[ПРИЛОЖЕНИЕ В Техническое задание 62](#_Toc74170880)

[ПРИЛОЖЕНИЕ С Программа и методика испытаний 69](#_Toc74170898)

# СПИСОК ОБОЗНАЧЕНИЙ И СОКРАЩЕНИЙ

1. CRF – условные случайные поля
2. MEMM – модель Маркова с максимальной энтропией
3. HMM – скрытая марковская модель
4. ID3 – алгоритм построения дерева принятия решений
5. NOUN – существительное
6. PROPN – имя собственное
7. ADJ – прилагательное
8. PRON – местоимение
9. NUM – числительное
10. VERB – глагол
11. ADV – наречие
12. DET – детерминант
13. CONJ – союз
14. ADP – предлог
15. PART – частица
16. H – вводное слово
17. INTJ – междометие
18. Case – падеж
19. Gender – род
20. Number – число
21. Animacy – одушевленность
22. Tense – время
23. Person – лицо
24. VerbForm – форма глагола
25. Mood – наклонение
26. Variant – кратность прилагательного
27. Degree – степень сравнения
28. NumForm – графическая форма числительного
29. АОТ – система автоматической обработки текста
30. MyStem – морфологический анализатор русского языка
31. PyMorphy2 – морфологический анализатор русского языка
32. TreeTagger – морфологический анализатор

# ВВЕДЕНИЕ

Обработка естественного языка является одним из важнейших и актуальных направлений в области синтеза искусственного интеллекта и математической лингвистики.

Создание системы обработки текста на естественном языке дает платформу для решения сразу нескольких проблем. Можно выделить три целевых блока, где возможно использование этих методов. Первый – анализ текста и информационный поиск. Второй – синтез текстов, предполагающий автоматическую генерацию текстов с заданными характеристиками. Третий – естественно-языковое взаимодействие с компьютером.

Квалификационная работа на тему «Модуль морфологического анализа русскоязычного текста» посвящена разработке алгоритма морфологической разметки.

Разработка модуля проводилась в рамках кафедральных исследований по обработке естественного языка, а неотъемлемой частью любой системы обработки текста на естественном языке является модуль морфологической разметки.

# ПОСТАНОВКА ЗАДАЧ РАЗРАБОТКИ

## Общетехническое обоснование разработки

* + 1. Постановка задачи проектирования

Разработке подлежит модуль морфологического анализа русскоязычного текста. Морфологический анализ в лингвистике — определение морфологических характеристик слова. Каждая часть речи имеет набор грамматических категорий, каждая категория — это набор взаимоисключающих граммем. В рамках конкретной словоформы каждая категория может реализоваться единственным образом. Например, для существительных есть категория числа (Number). Число может быть либо единственным (Sing), либо множественным (Plur). Таким образом, Animacy, Case, Gender и Number — это те категории, на которые разбивается грамматическое значение существительного, а неодушевленное, родительный падеж, женский род, единственное число — это вектор реализовавшихся для них граммем. Лемма – это словарная форма слова, т.е. лемма существительного – это его форма именительного падеж и единственного числа. Пример морфологической разметки предложения представлен на рисунке 1.

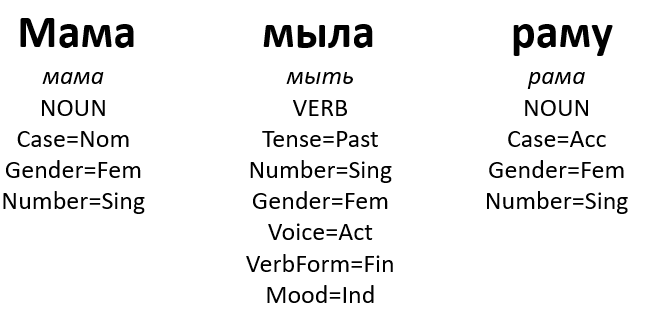


Рисунок 1 – Морфологическая разметка предложения

Таким образом, задача морфологической разметки текста заключается в определении части речи, грамматических значений и леммы для каждого слова в предложении. Сложность маркировки частей речи заключается в неоднозначных и не вошедших в лексику словах.

Эта двусмысленность может иметь место между словами одной и той же лексемы. Например, «стол» может быть как именительной, так и винительной формой. Также может возникнуть двусмысленность между словами разных лексем: «стекло» может быть как существительным («стекло»), так и глаголом («стекать»). Чтобы справиться с двусмысленностью, нам нужно использовать контекстную информацию.

Также необходимо разработать пользовательский интерфейс взаимодействия с модулем. Для удобного и эффективного взаимодействия интерфейс должен удовлетворять следующим требованиям:

1. Естественность. Результаты, выдаваемые приложением, должен быть интуитивно понятными. Должны быть сохранены система обозначений и терминология, используемая в данной предметной области;
2. Ясность. Интерфейс должен быть предсказуемым и иметь очевидную цель. Пользователи должны понимать, с чем они взаимодействуют;
3. Пошаговый информационный поток. На экране должна отображаться только необходимая информация, которой будет достаточно для принятия решения пользователем.

Конкретизируем задачу разработки, разбив её на несколько подзадач. Для достижения поставленной цели и успешной разработки программного модуля нам необходимо:

* Исследование предметной области;
* Определение функциональных требований;
* Выбор основного морфологического анализатора;
* Выбор моделей машинного обучения, улучшающих качество предсказаний;
* Выбор данных для обучения моделей;
* Программная реализация;
* Тестирования программного модуля.

Успешно решив все подзадачи, мы получим эффективный модуль морфологического анализа русскоязычного.

* + 1. Описание предметной области

Предметной областью данной разработки является морфология русского языка. Для того чтобы подчеркнуть различия в употреблении слов люди придумали формы слов или словоформы. Однако какова бы ни была словоформа, она выражает одно и то же понятие. Обсуждая понятие само по себе, принято использовать его нормальную форму – просто одну из словоформ, выделенную для обозначения понятия. Т.е., если у нас есть слово «мама», то для него существует несколько форм: мамы, маме, маму и т.д. К каждой форме приписывается ряд характеристик или параметров (род, падеж, число), характеризующих данную словоформу. Также каждому слову приписывается часть речи, показывающая, какого рода понятием мы оперируем. В речи мы привыкли к тому, что в данном месте должно стоять слово с заданной частью речи в определенной форме, но при машинной обработке подобные интуитивные рассуждения должны быть формализованы. Кроме того, подобное разнообразие вносит известные проблемы при анализе текста. Вместо того, чтобы работать с единственным словом, мы вынуждены обрабатывать все его словоформы.

Список основных частей речи в целом уже устоялся, хотя различные исследователи всё еще спорят о составе служебных частей речи. При реализации конкретного морфологического словаря важно с самого начала определиться с их списком, так как его изменение потом может оказаться дорогостоящей операцией. Для практических задач удобна любая из имеющихся логически обоснованных систем деления слов на части речи.

Морфологический параметр – это пара <имя параметра, значение параметра>. Именем параметра может служить род, число, время, склонение, краткость формы прилагательного и другие признаки слов, принятые в данном языке. Значение параметра – это конкретное значение, которое может принимать данный признак. Так, например, падеж может быть именительным, родительным, местным, аккузативным; род может быть мужским, женским, средним; число – единственным, множественным, двойственным и т.д.

В данной работе используется следующий список частей речи:

1. существительное;

2. имя собственное;

3. прилагательное;

4. местоимение;

5. числительное;

6. глагол;

7. наречие;

8. детерминант;

9. союз;

10. предлог;

11. частица;

12. вводное слово;

13. междометие.

Для самостоятельных частей речи размечаются следующие морфологические параметры:

1. существительное: род, число, падеж, одушевленность;

2. имя собственное: род, число, падеж;

3. прилагательное: род, число, падеж, краткость, степень сравнения;

4. местоимение: род, число, падеж, лицо;

5. числительное: род, падеж, графическая форма;

6. глагол: наклонение, лицо, время, число, род, форма;

7. наречие: степень сравнения;

8. детерминант: род, число, падеж.

В ряде случаев значение морфологического параметра невозможно определить однозначно. Так происходит, если даны омонимичные слова без контекста. Морфологическая омонимия – это ситуация, когда одной словоформе можно приписать несколько кортежей, содержащих нормальную форму, части речи и набор параметров.

* + 1. Функциональные задачи системы

Функциональные задачи формируются на основе целей, поставленных перед разрабатываемым модулем.

Функциональная задача – это программная реализация одной или нескольких функций программного модуля.

Доступ к модулю должен быть возможен как через веб-интерфейс, так и с помощью командной строки.

Разрабатываемый модуль должен выполнять следующие функции, доступные из веб-приложения:

* Морфологическая разметка русскоязычного текста;
* Обучение моделей;
* Оценка качества моделей.

# КОНСТРУКТОРСКО-ТЕХНОЛОГИЧЕСКАЯ ЧАСТЬ

## Выбор программных средств

Выбор операционной системы для разработки пал на ОС Windows, встречающуюся почти на всех современных ПК. Windows – наиболее популярная система, обладающая простым внешним оформлением и упрощающая взаимодействие с пользователем. Windows обладает всеми необходимыми средствами для разработки и поддерживает большинство программных продуктов.

Далее необходимо определиться с языком программирования, который будет наиболее удобным и оптимальным для разработки программного модуля морфологического анализа русскоязычного текста.

Учитывая задачи, поставленные перед программным модулем, в качестве языка программирования был выбран Python3, т.к. он оптимально подходит для задач обработки естественного языка.

Python - это универсальный и очень популярный язык программирования высокого уровня, который активно используются в веб-разработке и машинном обучении. Python позволяет программировать в объектно-ориентированной и процедурной парадигмах. Код, написанный на этом языке, удобочитаем и понятен, а также занимает намного меньше места из-за простоты его синтаксиса, что, в свою очередь, значительно увеличивает скорость разработки. Самое большое преимущество Python перед другими языками программирования – огромная коллекция стандартной библиотеки. Также Python популярен для веб-разработки, так как имеет несколько надежных и проверенных инструментов для неё.

Далее необходимо определиться со средой разработки.

IDE (или интегрированная среда разработки) — это программа, предназначенная для разработки программного обеспечения. Как очевидно из названия, IDE собирает в себе несколько инструментов, специально предназначенных для разработки. Эти инструменты обычно включают редактор, предназначенный для работы с кодом (например, подсветка синтаксиса и автодополнение); средства сборки, выполнения и отладки; и определённую форму системы управления версиями.

Существует много IDE и редакторов кода для Python.

К IDE предъявляются базовые требования:

1. Сохранение файлов.
2. Запуск кода из среды разработки.
3. Отладка (возможность пошагово выполнить код).
4. Подсветка синтаксиса.
5. Автоматическое форматирование кода.

Эти требования не зря называются базовыми, все популярные IDE имеют вышеперечисленные функции.

Для разработки данного модуля была выбрана IDE Visual Studio Code. Visual Studio Code доступен на всех платформах, имеет все необходимые функции для полноценной разработки. Бесплатно распространяется и имеет открытый исходный код.

Далее необходимо выбрать основной морфологический анализатор. В таблице 1 приведено сравнение характеристик рассмотренных морфологических процессоров.

Таблица 1 – Сравнение морфологических анализаторов

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Система | АОТ | MyStem | TreeTagger | PyMorphy2 |
| Открытые исходные коды | Да | Нет | Нет | Да |
| Скорость, слов в секунду | 60-90 тыс. | 100-120 тыс. | 20-25 тыс. | 80-100 тыс. |
| Подключение словарей | Нет | Да | Да | Нет |
| Объем словаря, тыс. слов | 160 | 250 | 210 | 250 |
| Удобство тэгов для поставленной задачи | Хорошо | Хорошо | Удовлетворительно | Отлично |

Все морфопроцессоры предоставляют наиболее важную для русского языка функцию лемматизации словоформ, при этом со снятием омонимии. Данная функция реализуется и для несловарных слов.

Два из представленных процессоров являются закрытыми и распространяются исключительно в виде бинарных файлов. Словарь MyStem является закрытым, словарь TreeTagger доступен в виде бинарного файла.

Существенной проблемой, связанной с морфологическими процессорами, является использование собственной системы морфологических тегов в каждом из них. Из-за несоответствия морфологических тегов сложно сравнивать работу процессоров, оценивать их точность и полноту на размеченных корпусах. Решением данной проблемы мог бы быть универсальный конвертер из одной системы тегов в другую, который отсутствует во всех рассмотренных анализаторах.

Для данной работы был выбран морфологический анализатор PyMorphy2, так он предоставляет оптимальный набор тегов, имеет открытый исходный код и имеет достаточный размер словаря.

Для разработки веб-приложения использовался фреймворк Streamlit. Streamlit – это фреймворк с открытым исходным кодом для создания веб-приложений в задачах машинного обучения и анализа данных [1].

## Выбор аппаратных средств

Аппаратные средства компьютера — совокупность его электрических, электронных и механических компонентов, составляющих его техническое обеспечение. Быстродействие разработанного модуля напрямую зависит от аппаратных средств компьютера. Так, например, объем необходимой оперативной памяти напрямую зависит от размера корпуса данных, на котором будет обучаться и тестироваться модель. Скорость работы процессора влияет на скорость обработки корпуса и скорость предсказания.

Для оптимальной скорости работы системы компьютер должен быть оснащён:

* Оперативной памятью не менее 4 Гбайт;
* Жестким диском не менее 10 Гбайт;
* Процессором Intel core i3 третьего поколения и выше;
* Клавиатурой и мышью для возможности личного управления компьютером.

## Разработка алгоритмов

Базовый морфологический анализатор PyMorphy2, используемый в данной работе, возвращает все допустимые варианты разбора слова [2]. У каждого разбора есть параметр score, равный вероятности того, что данный разбор правильный. Этот параметр рассчитывается исключительно на основании статистических величин корпуса, используемого в PyMorphy2. Поэтому главной целью алгоритма будет являться вывод одного единственно правильного разбора слова с учетом его контекста.

## Загрузка корпуса

Поскольку для обучения CRF модели необходим размеченный корпус данных, первоочередной задачей является считывание и загрузка этого корпуса. Корпус соревнования RU-EVAL придерживается строгих морфологических стандартов [3]. В корпусе присутствуют следующие морфологические признаки [4]:

1. Падеж (Case): именительный - Nom, родительный - Gen, дательный - Dat, винительный - Acc, местный – Loc, творительный – Ins
2. Род (Gender): мужской - Masc, женский - Fem, средний – Neut
3. Число (Number): единственное - Sing, множественное – Plur
4. Одушевленность (Animacy): одушевленное - Anim, неодушевленное – Inan
5. Время (Tense): прошедшее - Past, непрошедшее – Notpast
6. Лицо (Person): первое – 1, второе – 2, третье – 3
7. Форма глагола (VerbForm): инфинитив - Inf, финитная - Fin, деепричастие – Conv
8. Наклонение (Mood): индикатив – Ind, императив – Imp
9. Краткость прилагательного (Variant): краткое – Short (если форма полная, отметка не ставится)
10. Степень сравнения (Degree): позитивная или суперлативная - Pos, сравнительная – Cmp
11. Графическая форма числительного (NumForm): числовая запись – Digit (если форма записи буквенная, метка не ставится)

Соглашения, принятые в морфологическом стандарте MorphoRuEval:

1. Предикативы, омонимичные кратким прилагательным, размечаются как краткие прилагательные.
2. Предикативное слово “нет” считается глаголом.
3. Омонимия кратких прилагательных и наречий разрешена следующим образом: прилагательное обязано быть частью сказуемого.
4. Начальной формой глагола считается инфинитив соответствующего вида.
5. Деепричастия считаются частью глагольной парадигмы.
6. Причастия считаются прилагательными и получают соответствующую лемму именительного падежа.
7. Порядковые числительные считаются прилагательными.
8. Не оценивается одушевленность существительных.
9. Не оценивается частеречная разметка предлогов, союзов, частиц, междометий, а также отдельных наречий (как, пока, так, когда), омонимичных союзам.
10. Не оценивается вид и залог глагола, время глагола делится на прошедшее и непрошедшее.

Пример размеченного предложения из взятого корпуса приведен ниже.

1 Скромный скромный ADJ Case=Nom|Degree=Pos|Gender=Masc|Number=Sing

2 Роман роман NOUN Animacy=Anim|Case=Nom|Gender=Masc|Number=Sing

3 Аркадьевич аркадьевич NOUN Animacy=Anim|Case=Nom|Gender=Masc|Number=Sing

4 заранее заранее ADV Degree=Pos

5 был быть VERB Gender=Masc|Mood=Ind|Number=Sing|Tense=Past|VerbForm=Fin

6 согласен согласный ADJ Degree=Pos|Gender=Masc|Number=Sing|Variant=Short

7 на на ADP \_

8 любой любой ADJ Case=Acc|Degree=Pos|Gender=Masc|Number=Sing

9 поворот поворот NOUN Animacy=Inan|Case=Acc|Gender=Masc|Number=Sing

10 - - PUNCT \_

11 лишь лишь PART \_

12 бы бы PART \_

13 процветал процветать VERB Gender=Masc|Mood=Ind|Number=Sing|Tense=Past|VerbForm=Fin

14 его его DET \_

15 бизнес бизнес NOUN Animacy=Inan|Case=Nom|Gender=Masc|Number=Sing

16 . . PUNCT \_

Каждый токен предложения начинается с новой строки, далее идет его лемма, часть речи и морфологические характеристики. Функция загрузки корпуса возвращает три списка: corpus, X\_raw и y.

Часть списка corpus приведена на рисунке 2.

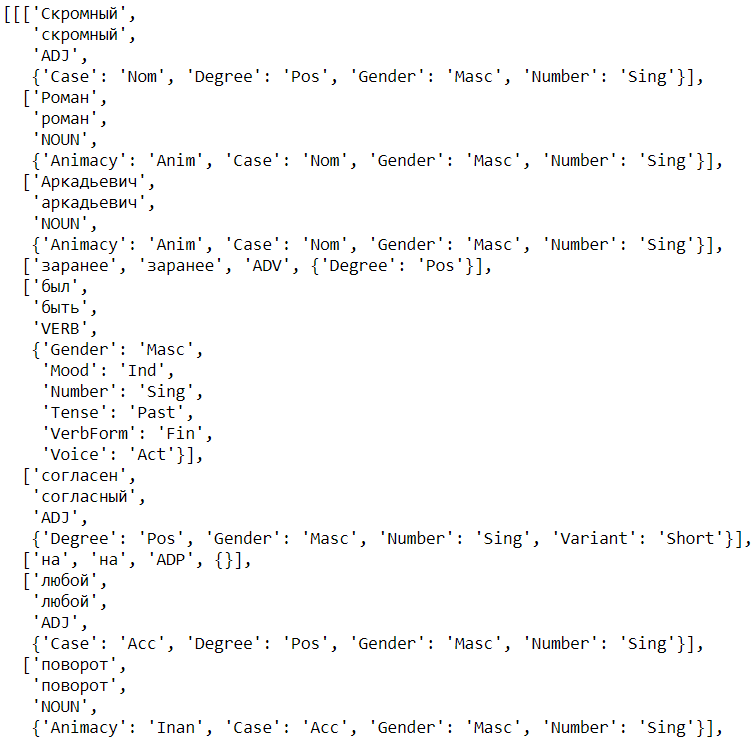


Рисунок 2 – Часть списка corpus

В данном списке содержится вся информация из исходного корпуса, представленная в удобном для работы виде.

Часть списка X\_raw представлена на рисунке 3.

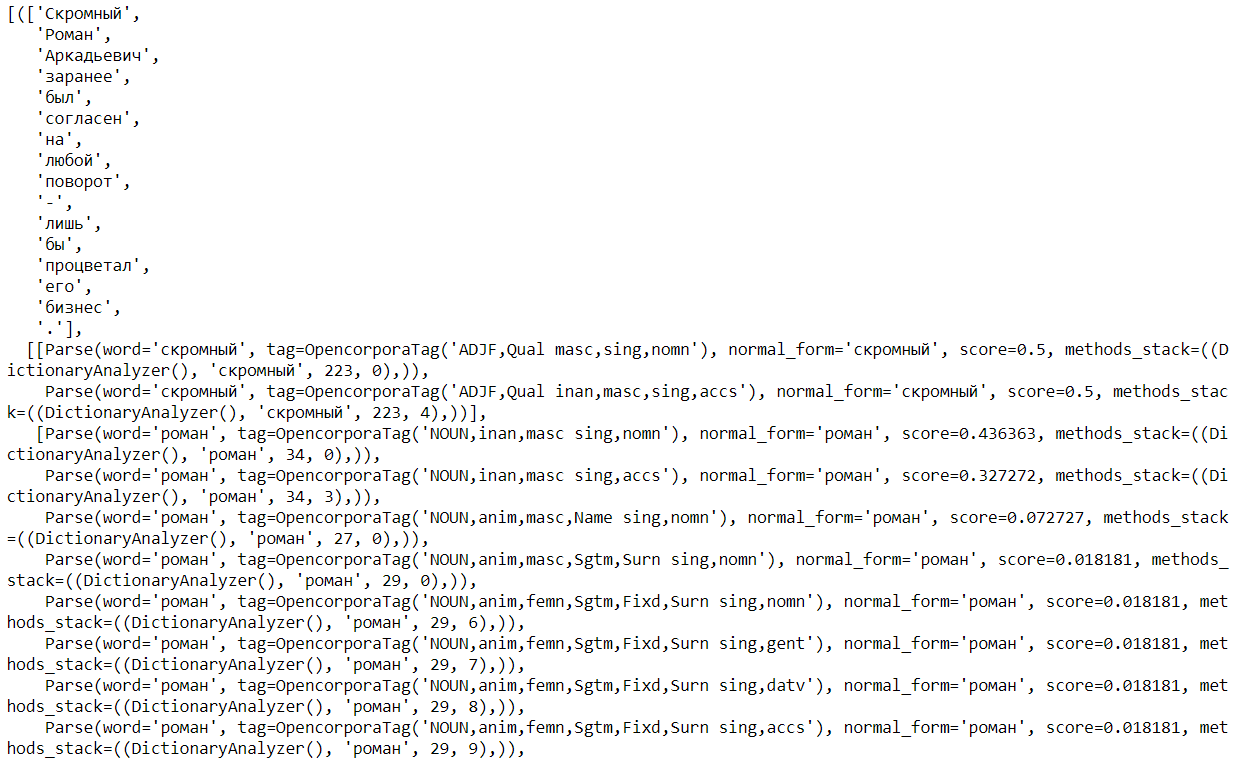


Рисунок 3 – Часть списка X\_raw

В данном списке содержится список токенов каждого предложения и всевозможные разборы для каждого токена, которые поступают от морфологического анализатора PyMorphy2. Данная информация необходима для обучения CRF модели и учета контекста каждого слова [5].

Список y содержит в себе верную разметку для каждого токена. Эта информация нужна для оценки точности построенной модели. Часть такого списка изображена на рисунке 4.

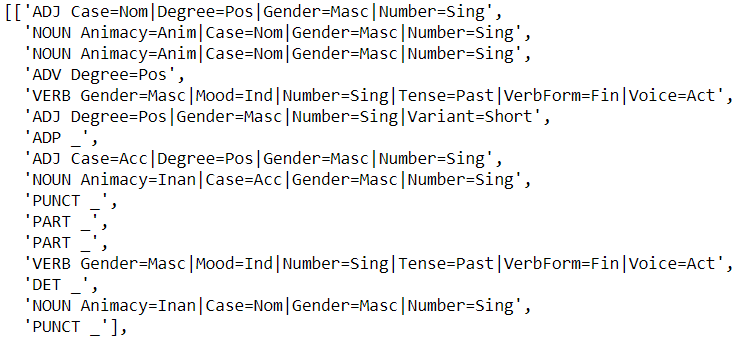


Рисунок 4 – Часть списка y

## Обучение CRF модели

Перед тем, как обучиться CRF модель, необходимо извлечь информацию о контексте слова и величины score, полученные от PyMorphy2 для каждого морфологического параметра [5]. Эту задачу выполняет функция fit\_transform(), принадлежащая классу TaggerFeatureExtractor. Данная функция получает на вход список X\_raw и извлекает информацию по следующим правилам:

1. Извлекается сам токен, предыдущий и следующий за ним токены в нижнем регистре.
2. Извлекаются всевозможные грамматические категории и их score не только для данного токена, но и для следующего и предшествующего токенов.
3. Для первого слова в предложении добавляется метка начала предложения.
4. Для последнего слова в предложения добавляется метка конца предложения.

Пример части списка X, который возвращает функция fit\_transform(), приведен на рисунке 5.

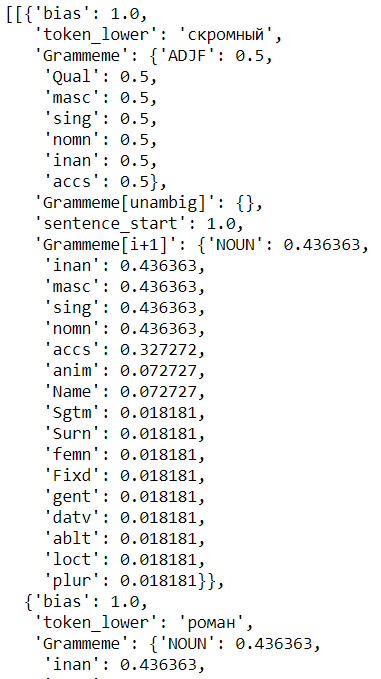


Рисунок 5 – Часть списка, возвращаемого функцией fit\_transform()

Далее полученный список Х и список с верными морфологическими разборами y разбивается на тренировочную и тестовую выборки и подается на вход CRF модели. Обучение модели занимает некоторое время, напрямую зависящее от размера обучающего корпуса данных. По окончании обучения модель сохраняется для дальнейшего использования.

## Морфологический анализ

Первое, что необходимо сделать с текстом, введенным пользователем – токенезировать его, т.е. разбить на слова и знаки препинания [6]. Сначала предложение разбивается на токены с помощью регулярных выражений подобно тому, как это представлено на рисунке 6.

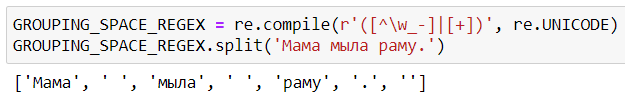


Рисунок 6 – Разбиение на токены с помощью регулярных выражений

Список очищается от пробелов и пустых элементов. Токенизация данным способом – один из простейших вариантов токенизации, который не учитывает некоторые редко встречающиеся варианты. Так, например, название компании «Яндекс.Еда» алгоритм разобьет на три токена. Однако в данной работе это не учитывается, так как выделение именованных сущностей – отдельная и очень обширная задача обработки естественного языка.

После того, как предложения разбиты на токены, они подаются на вход ранее упоминавшейся функции fit\_transform(). Уже обработанные токены подаются на CRF модель, которая определяет часть речи и морфологические параметры.

В том случае, если пользователь предал в функцию морфологической разметки параметр lemmatization=True, токены буду лемматизированы по следующему алгоритму:

1. Берется токен и его разбор, полученный от CRF модели.
2. Берутся всевозможные варианты разборов и лемм для данного токена, полученные от PyMorphy2.
3. Выбирается та лемма, морфологические параметры которой сходятся с разметкой полученной от CRF модели.
4. Если ни один вариант леммы не подходит, выбирается лемма с большим значением score.

Примеры разобранных предложений с включенной и выключенной лемматизацией приведены на рисунке 7 и рисунке 8 соответственно.

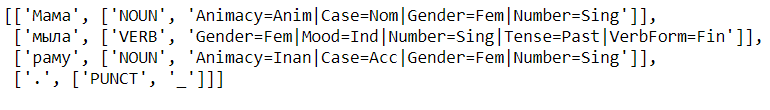


Рисунок 7 – Разбор предложения с выключенной лемматизацией

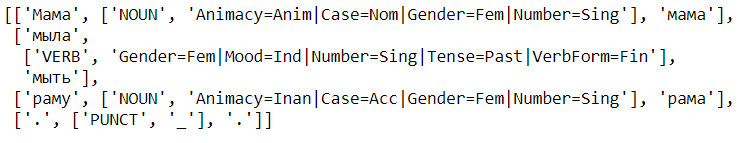


Рисунок 8 – Разбор предложения с включенной лемматизацией

## Оценка полученной модели

В первую очередь необходимо оценить скорость работы модели. Текст, состоящий из 10 тысяч слов, в режиме без лемматизации модель обрабатывает за 4 секунды. Учитывая, что модель обрабатывает не только слова, но и знаки препинания, фактическая скорость обработки несколько выше. В режиме с включенной лемматизацией модель обрабатывает тот же текст за 8 секунд. Уменьшение скорости обработки текста при включенной лемматизации связано с прямым перебором всех вариантов разбора в алгоритме лемматизации.

Далее необходимо выбрать метрики оценки модели. Были выбраны стандартные метрики для задач классификации: precision, recall, F1-score и accuracy [7]. В вычислении каждой из этих метрик используется матрица ошибок. Матрица ошибок – это таблица, которая позволяет визуализировать эффективность алгоритма классификации путем сравнения прогнозируемого значения целевой переменной с ее фактическим значением.

Таблица 2 – Матрица ошибок

|  |  |  |  |
| --- | --- | --- | --- |
|  | | Экспертная оценка | |
| Положительная | Отрицательная |
| Оценка системы | Положительная | TP | FP |
| Отрицательная | FN | TN |

TP — true positive, классификатор верно отнёс объект к рассматриваемому классу.

TN — true negative, классификатор верно утверждает, что объект не принадлежит к рассматриваемому классу.

FP — false positive, классификатор неверно отнёс объект к рассматриваемому классу.

FN — false negative, классификатор неверно утверждает, что объект не принадлежит к рассматриваемому классу.

**Accuracy**

(1)

Как эвристика или эмпирическое правило, метрика accuracy может сразу сказать, правильно ли обучается модель и как она может работать в целом. Однако она не дает подробной информации о её применении к проблеме. Проблема с использованием accuracy в качестве основной метрики производительности заключается в том, что она не работает, когда имеется серьезный дисбаланс классов. Чтобы объективно оценить модель, нужно использовать и другие метрики.

**Precision**

(2)

Precision активно используется, когда важно фиксировать количество FP предсказаний. В некоторых задачах, например, при обнаружении спама, ложное срабатывание является худшей ошибкой, чем ложноотрицательное (как правило, отсутствие важного электронного письма хуже, чем неудобство, связанное с удалением спама, прошедшего через фильтр).

**Recall**

(3)

Recall – это противоположность precision. Эта метрика измеряет ложноотрицательные результаты по сравнению с истинными положительными. Ложноотрицательные результаты особенно важны для задач обнаружения болезней и других прогнозах, связанных с безопасностью.

**F1-score**

(4)

Если необходимо сбалансировать две цели: выcокую точность и высокую отзывчивость, то стоит использовать F1-score. F1-score рассчитывается как среднее гармоническое значение precision и recall. Хотя мы использовать и простое среднее из двух оценок, гармонические средние более устойчивы к выбросам. Таким образом, оценка F1 представляет собой сбалансированный показатель, который надлежащим образом количественно определяет правильность моделей во многих областях.

Оценим полученную модель с помощью приведенных выше метрик. Функция оценки принимает на вход обученную CRF модель, размеченный корпус данных и два параметра для оценки лемматизации и использования сравнения с PyMorphy2.

Функция оценки выводит значения метрик для каждой части, общие значения метрик, accuracy для полного тега и лемм. Результаты оценки PyMorphy2 на корпусе из 170 тысяч токенов приведены на рисунке 9.

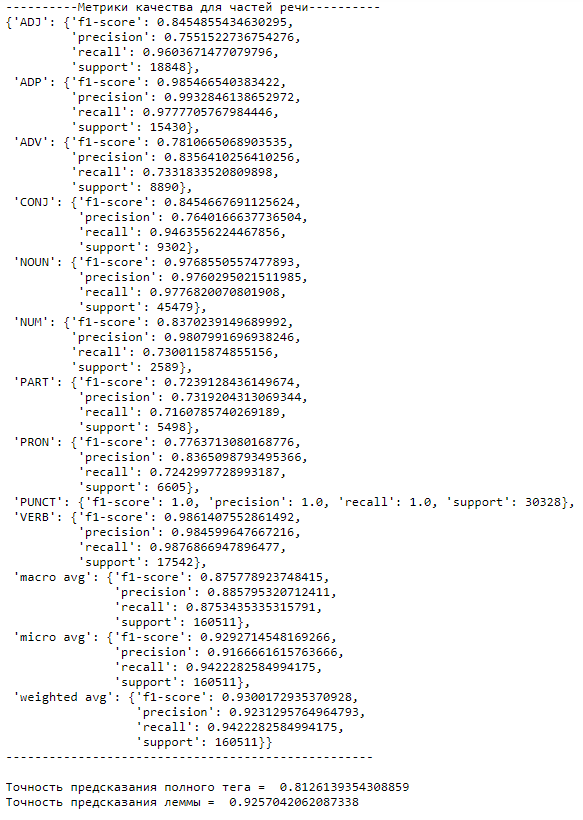


Рисунок 9 – Метрики качества для PyMorphy2

Аналогичным образом производится оценка CRF модели. Результаты представлены на рисунке 10.

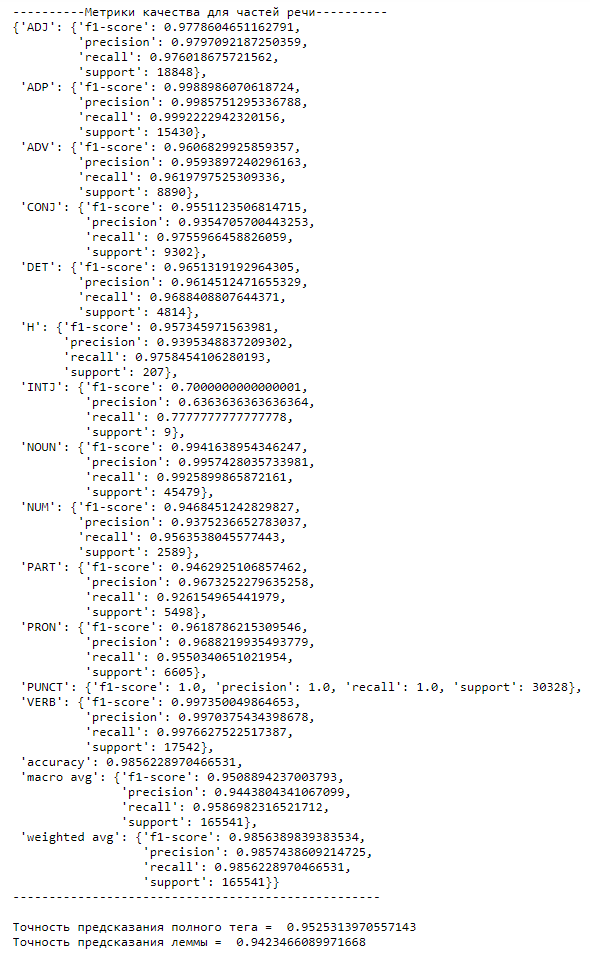


Рисунок 10 – Метрики качества для CRF модели

Диаграмма сравнения F1-score PyMorphy2 с CRF моделью для всех частей речи представлен на рисунке 11.

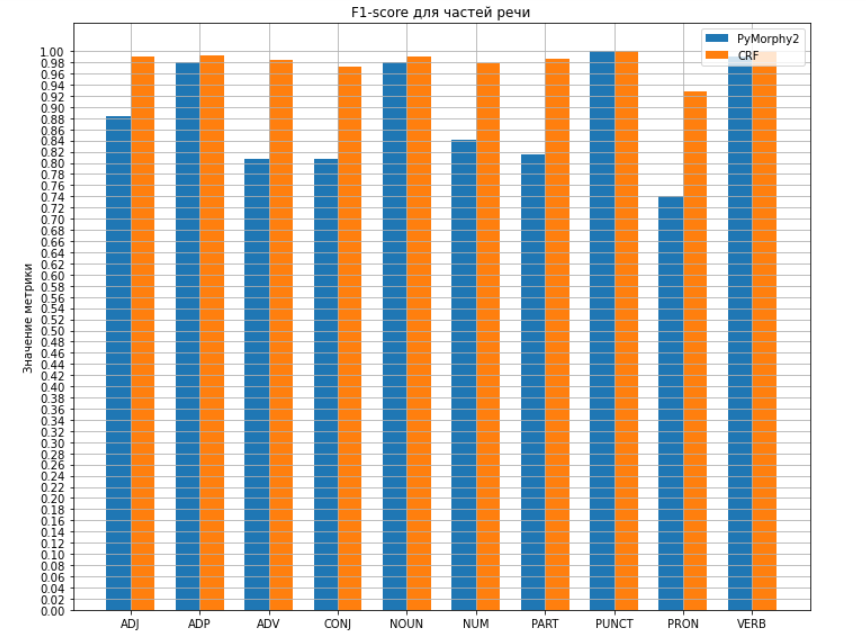


Рисунок 11 – Диаграмма сравнения CRF и PyMorphy2

Из приведенной диаграммы видно, что точность предсказания каждой части речи повысилась. Точность предсказания лемм была повышена на 2%, точность предсказания полного тега была повышена на 14%.

## Разработка веб-интерфейса

В соответствии с поставленными задачами разработки было разработано веб-приложение взаимодействия пользователя с модулем морфологического анализа русскоязычного текста. Для удобного, красивого и понятного пользователю интерфейса была использована библиотека Streamlit.

При запуске веб-приложения пользователь попадает на страницу «Разметка русскоязычного текста» (рисунок 12)

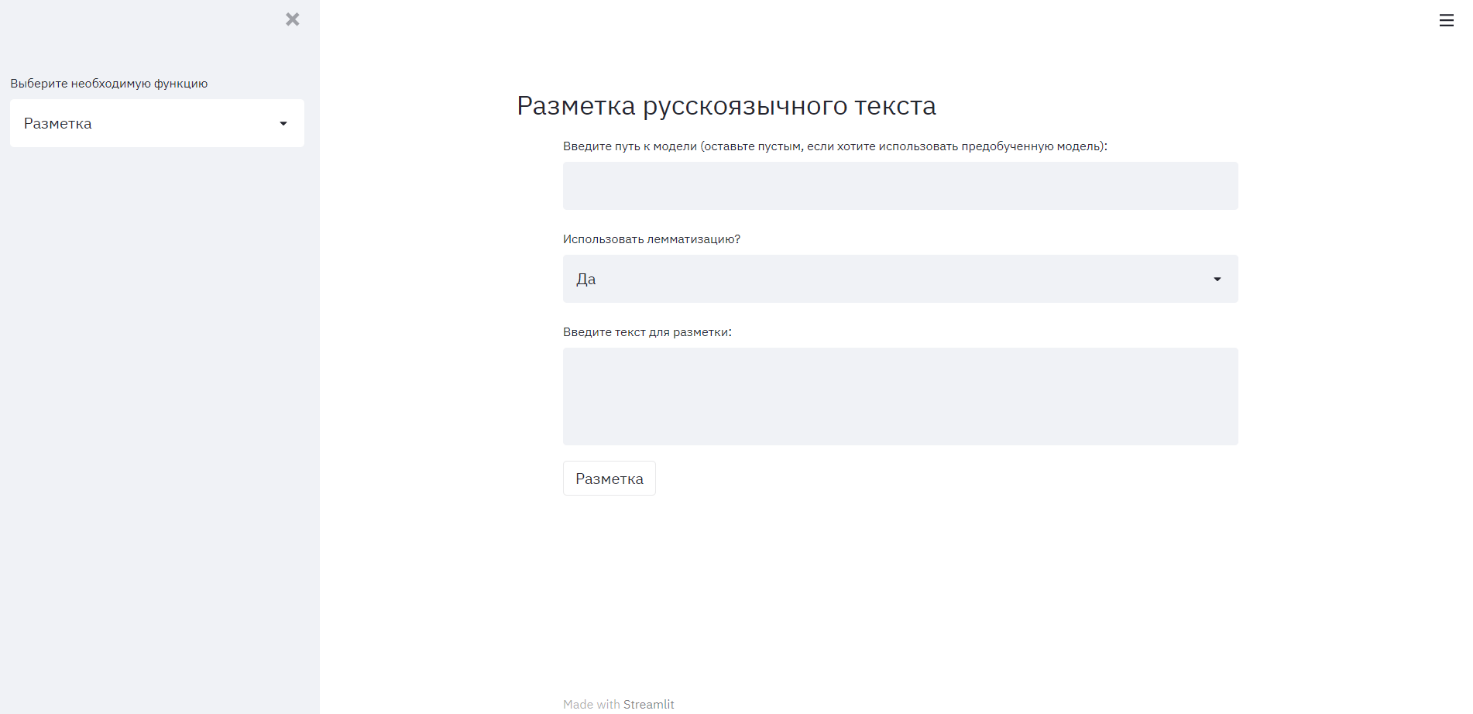


Рисунок 12 – Страница «Разметка русскоязычного текста»

Страница «Разметка русскоязычного текста» содержит поле для ввода пути к модели и поле для ввода текста, выпадающий список для использования лемматизации и кнопку «Разметка». Так как в модуле уже содержится модель, обученная на корпусе RU-EVAL, поле для ввода пути к модели можно оставить пустым. В таком случае будет использована предобученная модель.

После ввода текста, выбора параметра использования лемматизации и нажатии кнопки «Разметка» выводится размеченный текст (рисунок 14). Если путь к модели был указан неверно, приложение уведомляет об ошибке (рисунок 13).

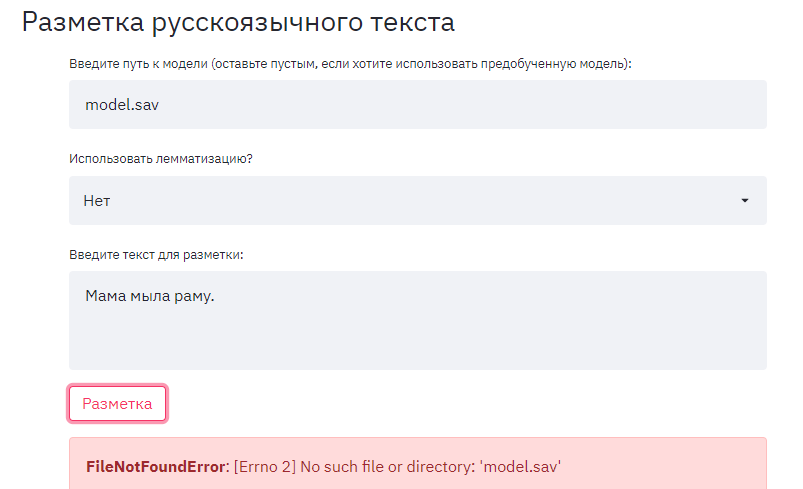


Рисунок 13 – Ошибка при неправильном вводе пути к модели

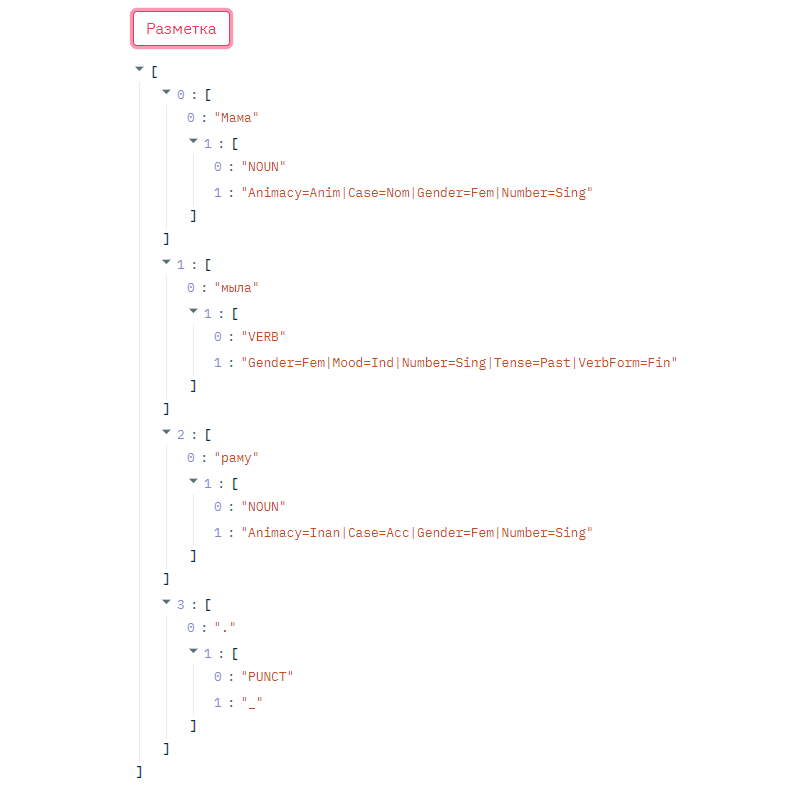


Рисунок 14 – Вывод размеченного текста

Переход на другие страницы веб-приложения производится с помощью нажатия на нужный вариант в выпадающем списке (рисунок 15)

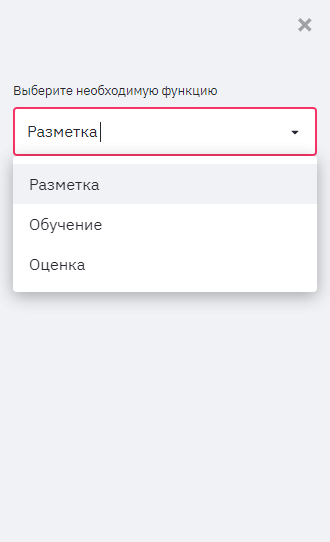


Рисунок 15 – Выпадающий список для перехода по страницам

Страница «Обучение моделей морфологической разметки» представлена на рисунке 16.

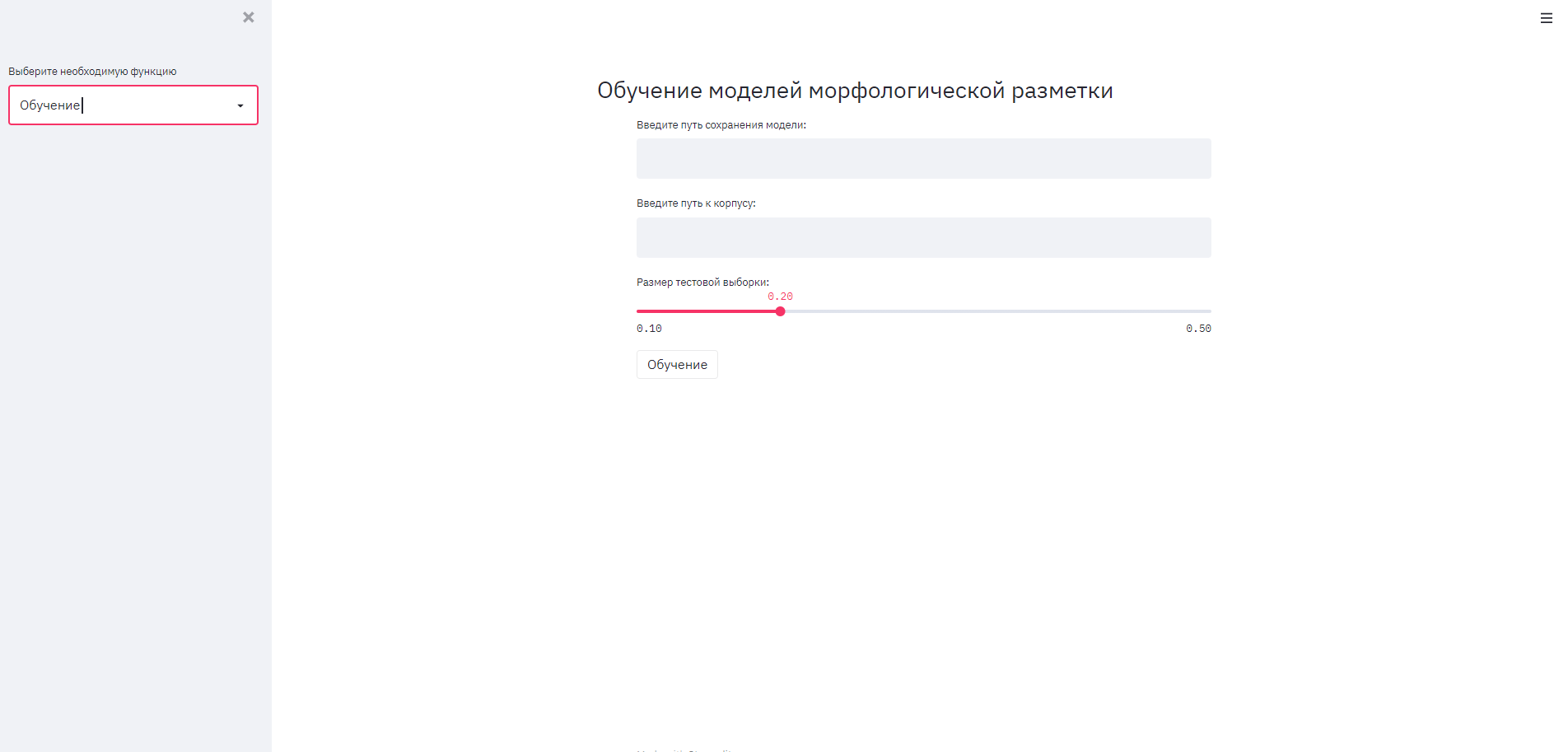


Рисунок 16 – Страница «Обучение моделей морфологической разметки»

На этой странице находятся поля для ввода пути сохранения модели и пути к корпусу, ползунок для выбора размера тестовой выборки и кнопка «Обучение».

В качестве значения размера тестовой выборки по умолчанию установлено значение 0,20. После ввода необходимых путей и нажатия кнопки «Обучение» модель обучается, сохраняется по указанному пути и на странице выводится сообщение о сохранении (рисунок 17).

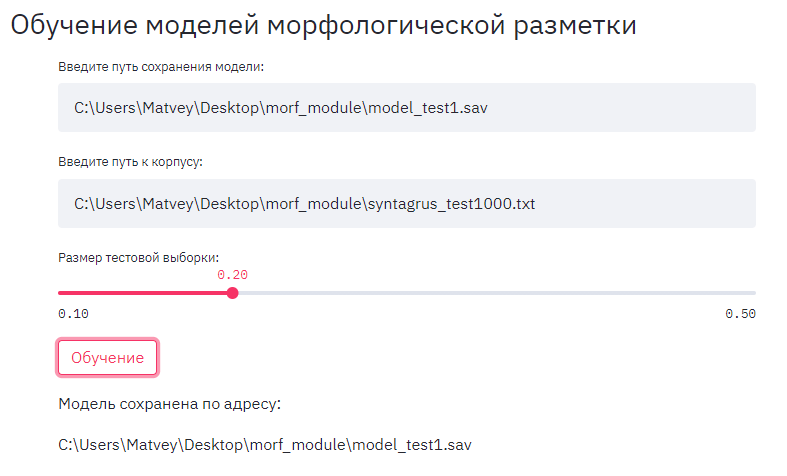


Рисунок 17 – Сообщение об успешном сохранении модели

Веб-приложение обрабатывает всевозможные ошибки, допущенные пользователем (рисунок 18).

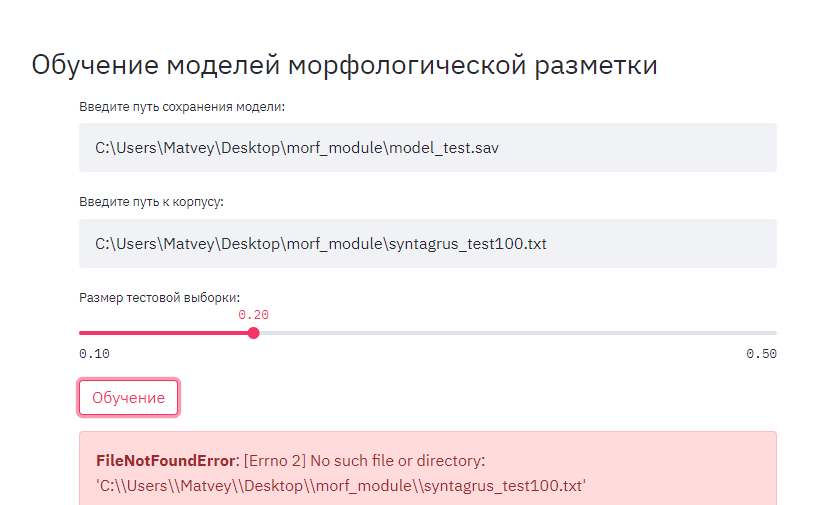


Рисунок 18 – Ошибка ввода пути к корпусу

Страница «Оценка моделей морфологической разметки» представлена на рисунке 19.

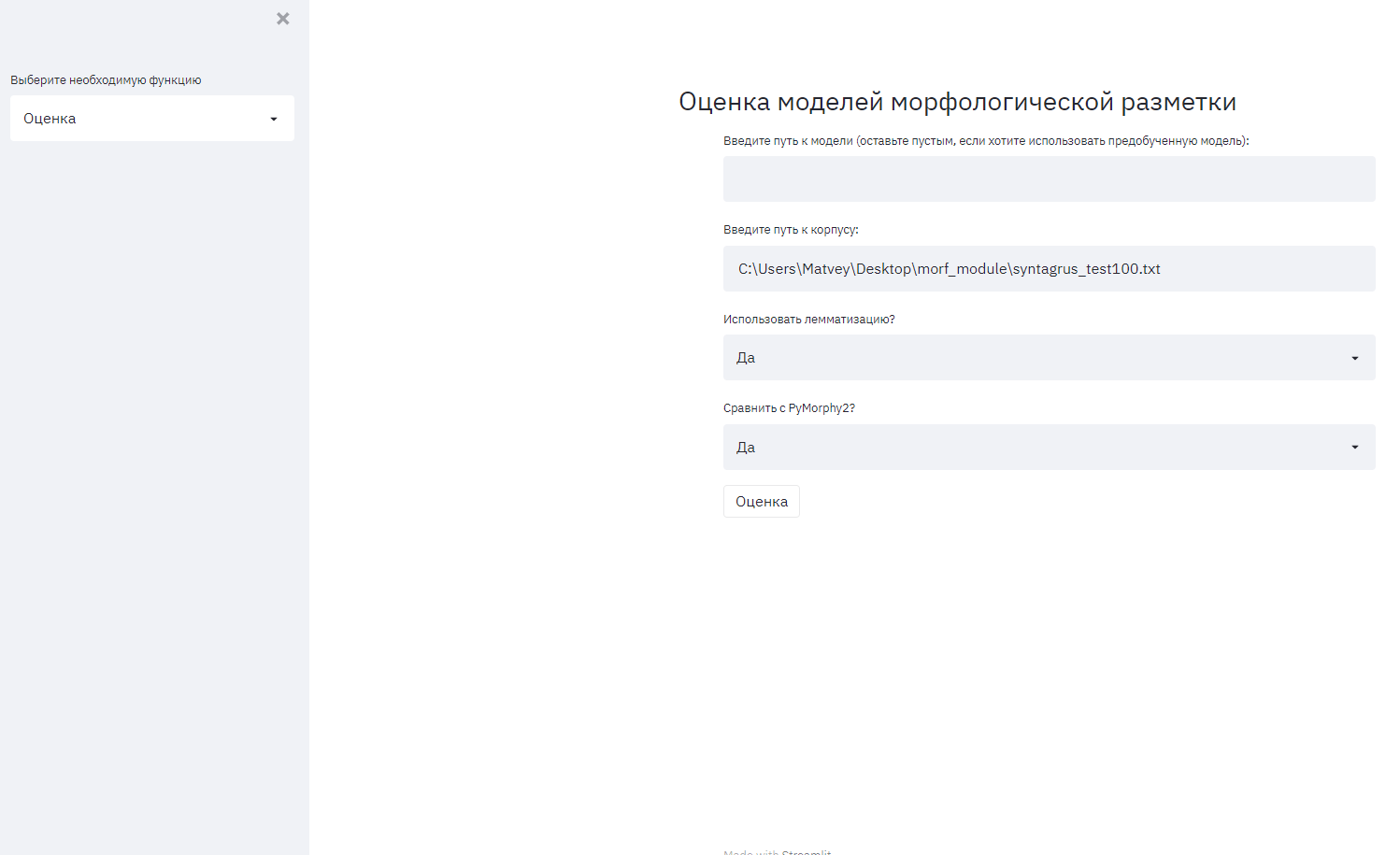


Рисунок 19 – Страница «Оценка моделей морфологической разметки»

На странице «Оценка моделей морфологической разметки» находятся поля для ввода пути к модели и пути к корпусу и выпадающие списки для использования лемматизации и сравнения с PyMorphy2.

При нажатии кнопки «Оценка» выводятся метрики качества модели. Если включена лемматизация, выводятся метрики её оценки. Если включено сравнение с PyMorphy2, выводятся метрики оценки PyMorphy2 и график сравнения. (рисунки 20, 21)

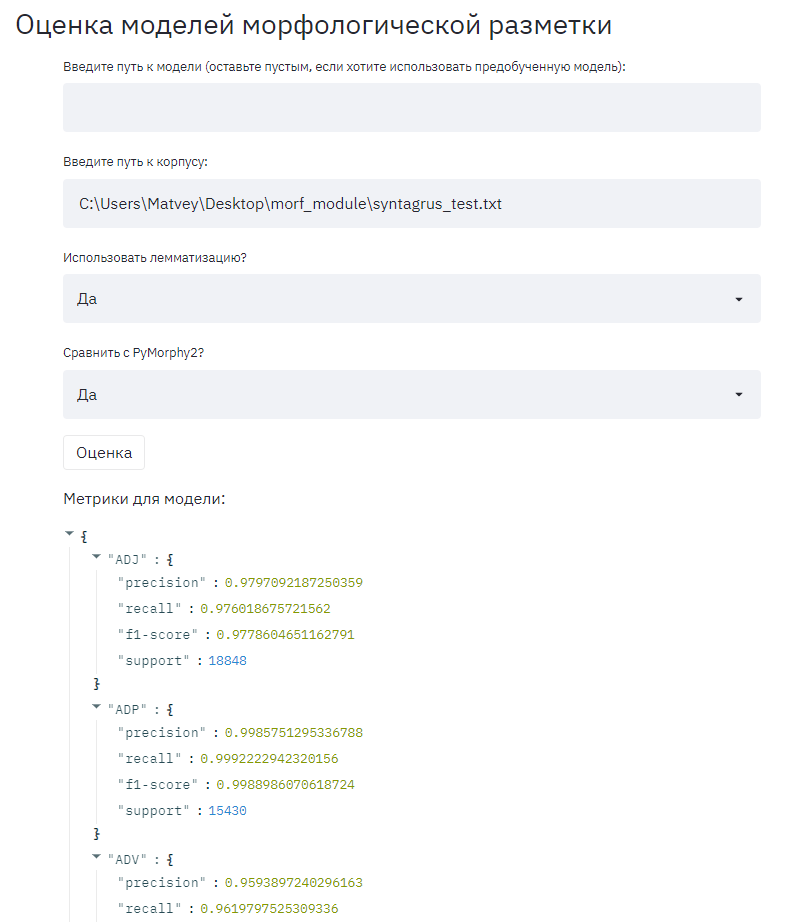


Рисунок 20 – Метрики оценки CRF модели

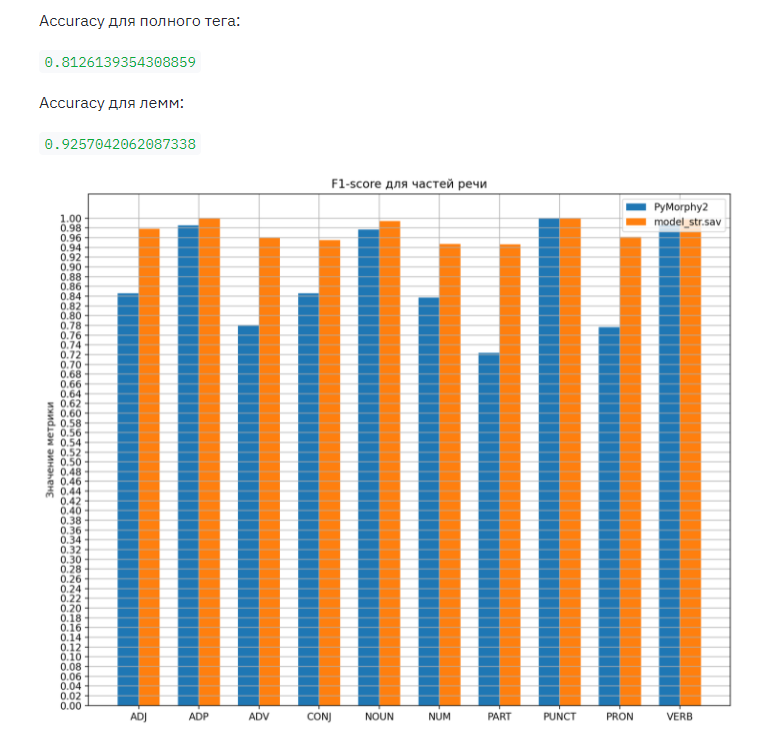


Рисунок 21 – Метрики оценки PyMorphy2 и диаграмма сравнения

# ИССЛЕДОВАТЕЛЬСКАЯ ЧАСТЬ

## Морфологические анализаторы для русского языка

Морфологический анализатор – это набор алгоритмов, который занимается сопоставлением отдельных слов и словоформ в словаре и выяснением грамматических характеристик слов[8].

По функциональным возможностям морфологические процессоры делятся на несколько видов:

1. Выполняющие только лемматизацию или стемминг
2. Определяющие часть речи
3. Осуществляющие полный морфологический анализ, т.е. лемматизацию и определение всех морфологических характеристик
4. Осуществляющие морфемный анализ, выделяющие морфы, входящие в состав слова

Помимо этого, анализаторы используют различные подходы к морфологическому анализу [9].

Бессловарные морфологии являются одним из первых подходов к решению задачи морфоанализа русского языка. Однако бессловарными они являются лишь условно: в них отсутствуют большие словари лексических единиц, но фактически используются небольшие словари с информацией о флексии. В первом столбце такого словаря содержатся флексии словоформ, во втором – флексия нормальной формы, а в третьем – морфологические характеристики, соответствующие исходной форме с данной флексией.

Таблица 3 – Часть словаря с информацией о флексиях

|  |  |  |
| --- | --- | --- |
| Флексия | Флексия начальной формы | Морфологические характеристики |
| -онок | -онок | СУЩ., неод., м. р., ед. ч., им. п. |
| -онока | -онок | СУЩ., неод., м. р., ед. ч., род. п. |
| -оноку | -онок | СУЩ., неод., м. р., ед. ч., дат. п. |
| -оноком | -онок | СУЩ., неод., м. р., ед. ч., твор. п. |

Для бессловарной морфологии анализ сводится к поиску наиболее длинного окончания анализируемой словоформы в этом словаре и выборке соответствующих морфологических характеристик.

Одним из классических подходов к морфологическому анализу русского языка является построение словарной морфологии на основе словаря основ. Основой модели выступает словарь основ, который содержит все основы лексем языка. Он связан со вспомогательными словарями, в которых содержится список флексий всех словоизменительных классов, для каждой из которых указан набор значений морфологических характеристик, которые она может выражать.

Разбор словоформ происходит по следующей схеме:

1. Последовательно отсекаются возможные окончания длиной от 0 до 𝑛 букв, таким образом слово разбивается на основу и флексию.
2. Для полученного окончания находится его словоизменительный класс.
3. Проверяется наличие полученной основы в словаре основ и находится номер её словоизменительного класса.
4. В случае совпадения словоизменительных классов выбираются соответствующие морфологические характеристики и строится лемма, которые и являются результатом анализа.

Для высокофлективного языка наиболее частым подходом к решению задачи морфологического анализа является словарная морфология на основе словаря словоформ. База морфологического процессора, построенного на такой морфологии, заключается в создании словаря всех форм языка. Определение нормальной формы и морфологических характеристик сводится к поиску словоформы в таблице.

## AOT

Морфологический процессор АОТ предоставляет все функции полного морфологического анализа, включая нахождение леммы и морфологических характеристик словоформы, а также синтез словоформ [10]. АОТ базируется на словарной морфологии, в качестве словаря используется русский грамматический словарь, который включает более 161 тыс. лемм. Логическая структура компьютерного словаря представляется в виде нескольких таблиц: лемм, флексий, приставок, морфологических характеристик. Таблица лемм содержит список псевдооснов слов со ссылками на таблицы флексий и приставок. В таблице флексий каждому из окончаний слов соответствует ссылка на соответствующие морфологические характеристики. Морфологический разбор слова по словарю состоит в поиске максимально совпадающей псевдоосновы в таблице лемм, поиск соответствующих приставки и окончания, а затем нахождение по таблице окончаний соответствующих морфологических характеристик.

В настоящий момент морфопроцессор проекта АОТ является полностью открытым и распространяется под лицензией LGPL. Однако, проект не поддерживается и не имеет удобных средств для пополнения словаря.

## TreeTagger

Процессор TreeTagger позволяет определять части речи слов и другие морфологические характеристики, а также их нормальную форму [11]. Основной упор в данном процессоре сделан на разрешение морфологической омонимии и предсказание характеристик неизвестных слов. TreeTagger базируется на словарной морфологии и использует словарь английского языка из проекта Penn TreeBank, содержащий более 2 млн. словоформ. Объем русского словаря неизвестен, однако по объему бинарного файла можно судить о сопоставимости размера словаря с английской версией. В процессе анализа используются 2 вида словарей: словоформ и суффиксов.

Для снятия частеречной омонимии в TreeTagger используются решающие деревья для частей речи, обученные на размеченном корпусе. В узлах такого дерева находятся предикаты с ответом «да» или «нет» для двух предшествующих слов. При этом в листьях хранятся значения вероятностей для возможных ответов. Построение дерева происходит рекурсивно, с помощью модифицированного алгоритма ID3. На каждом шаге для двух предыдущих слов проверяются предикаты на равенство всем возможным частям речи, при этом для определения предиката, наилучшим образом разбивающего пространство признаков, используется правило максимизации энтропии Шенона. Для определения части речи входного слова достаточно, используя информацию о предыдущих словах, пройти по дереву от корня до листьев и выбрать наиболее вероятное значение.

В настоящий момент TreeTagger распространяется в виде бинарного файла, код самого процессора является закрытым. Проект поддерживается, для него создаются новые словари под различные языки.

## PyMorphy2

Pymorphy2 – морфологический процессор с открытым исходным кодом, предоставляет все функции полного морфологического анализа и синтеза словоформ [2].

Процессор базируется на словарной морфологии и использует словарные данные проекта OpenCorpora [12]. Словарь содержит около 250 тыс. лемм, а также является полностью открытым и регулярно пополняемым.

В процессе морфологического синтеза, по исходной словоформе и тегам выполняется поиск нормальной формы слова, а затем перебор всех возможных пар <окончание, теги> в найденной лексеме, пока не будет найдена пара с заданными морфологическими тегами. После этого от нормальной формы отсекается её окончание, а найденное окончание приписывается к полученной псевдооснове.

Для анализа неизвестных слов в Pymorphy2 используются несколько методов, которые применяются последовательно. Изначально от слова отсекается префикс из набора известных префиксов и если остаток слова был найден в словаре, то отсеченный префикс приписывается к результатам разбора. Если этот метод не сработал, то аналогичные действия выполняются для префикса слова длиной от 1 до 5, даже если такой префикс является неизвестным. Затем, в случае неудачи, словоформа разбирается по окончанию. Для этого используется дополнительный автомат всех окончаний, встречающихся в словаре с имеющимися разборами. В процессе построения из автомата удаляются редкие окончания и разборы. Метод анализа по окончанию аналогичен тому, что используется в процессоре АОТ.

В настоящее время Pymorphy2 поддерживается, при этом происходит постоянное пополнение корпуса OpenCorpora, что улучшает характеристики точности и полноты морфологического разбора.

## MyStem

MyStem – морфологический анализатор, разработанный компанией Яндекc [13]. В настоящий момент MyStem версии 3.0 предоставляет все функции полного морфологического анализа, однако не имеет функции синтеза. Данная версия является наиболее стабильной и доступной для скачивания в бинарном виде.

Морфоанализатор MyStem базируется на словаре НКРЯ, который содержит более 200 тыс. лемм. Исходные коды MyStem являются закрытыми, поэтому характеристики использованной структуры данных не известны, однако размер полученного бинарного словаря более 20 МБ.

MyStem производит разрешение морфологической омонимии и делает разбор несловарных словоформ. Для решения этой задачи используются различные методы машинного обучения. В зависимости от входных данных MyStem снимает омонимию двумя способами: с учетом контекста и без учета контекста.

В настоящее время MyStem поддерживается и используется в ряде проектов, таких как НКРЯ. Также он доступен в виде динамической библиотеки для некоммерческих приложений и позволяет подключать собственные словари через опции командной строки или интерфейса библиотеки.

## Сравнение морфологических анализаторов

Русский язык представляет определенные проблемы в отношении морфологического анализа, так как является высокофлективным языком с множеством грамматических категорий [8]. Не существует стандарта даже для аннотации частей речи, не говоря уже о некоторых грамматических категориях. Теоретические споры, касающиеся русской морфологии, приводят к разнообразию решений для аннотации морфологии – от позиционных тегов в соответствии с рекомендациями MULTITEXT-East до комбинации тегов, используемых в RNC. Дополнительная проблема возникает из-за того, что теги на русском языке можно комбинировать и упрощать по-разному. Некоторые системы не учитывают ту или иную грамматическую категорию, в то время как другие системы определят некоторые значения категорий как значения по умолчанию. Сравнение тегов для русского языка также осложняется тем, что существуют различные теоретические традиции для процесса лемматизации. Например, некоторые системы лемму причастий считают глаголов, другие лемматизируют причастия как прилагательные [9].

Помимо отсутствия единых правил аннотации морфологии русского языка, до недавнего времени не существовала ни одного стандартного корпуса. В настоящее время существуют два корпуса, которые могут служить моделями для задач аннотации: неоднозначный подкорпус RNC и OpenCorpora.

В данной работе проведем оценку морфологических анализаторов на наборе данных RNC. Корпус RNC содержит 5,9 миллионов лексем, аннотированных с помощью MyStem. Омонимия в данном корпусе снята вручную, все токены имеют только один морфологический разбор. Пример аннотированного предложения приведен ниже.

<se>

<w><ana lex="береза" gr="S,f,inan=sg,nom"/>Берёза</w>

<w><ana lex="ждать" gr="V,ipf,tran,act=sg,praes,3p,indic"/>ждёт</w>

<w><ana lex="мороз" gr="S,m,inan=sg,gen"/>мор‘оза</w>!"

</se>

Однако, при тестировании морфологических анализаторов только на корпусе RNC возникает ряд проблем. Во-первых, TreeTagger был обучен на корпусе RNC, поэтому у него есть некоторое преимущество перед другими анализаторами. Во-вторых, RNC имеет очень подробный набор тегов, часть из которых можно исключить из анализа, поскольку они не так важны для задачи морфологического анализа. Таким образом, для оценки также будет использован корпус данных из соревнования RU-EVAL. Пример аннотированного предложения из этого корпуса представлен ниже.

как как CONJ

казалось казаться V n,past,sg

раньше раньше ADV

Главная проблема в сравнении морфологических анализаторов заключается в том, что все анализаторы имеют разные обозначения для частей речи и морфологических категорий. Расхождения между наборами тегов могут быть разного рода:

* Некоторая морфологическая категория присутствует в наборе тегов золотого стандарта, но отсутствует в наборе тегов морфологического анализатора, например MyStem различает одушевленность и неодушевленность, поскольку у него есть специальные словари, где эти характеристики определены для каждого слова. TreeTagger, однако, не считает эту информацию важной и не включает ее в анализ.
* Морфологические анализаторы имеют различные стандарты в отношении идентификации частей речи, например Freeling идентифицирует причастия как отдельную часть речи, в то время как другие морфологические анализаторы идентифицируют причастия как глагольные формы.
* Следовательно, наряду с различными стандартами в отношении идентификации частей речи, морфологические анализаторы присваивают словам различные леммы.
* Если часть речи была идентифицирована одинаково, проблемы с лемматизацией все равно могут возникнуть.

Из-за этих проблем необходимо определить соглашения, которые позволят сделать сравнение теггеров возможным и корректным. Преобразуем все теги в теги MyStem. Правила конвертации представлены в таблице 4.

Таблица 4 – Правила конвертации тегов

|  |  |
| --- | --- |
| Тег золотого стандарта | Тег, принимаемый за верный |
| A-NUM | NUM |
| PARENTH | ADV |
| ADV-PRO | PRO |
| A-PRO | PRO |
| m-f | Не важно |
| anim | Не важно |
| inan | Не важно |
| dat2 | dat |
| gen2 | gen |
| ccc2 | acc |
| loc2 | loc |
| adnum | NUM |
| intr | Не важно |
| tran | Не важно |

Правила преобразования в набор тегов RUE VAL были такими же. Кроме того, из анализа были исключены следующие случаи:

* отсутствие наклонения у глаголов;
* путаница между предикатами и другими частями речи;
* глаголы, заканчивающиеся на “ся”;
* различие между полными и сокращенными формами прилагательных и причастий.

Данные правила означают, что в качестве правильного вывода принимаются менее конкретные теги, например, дательный падеж вместо второго дательного падежа. Это приводит к потере некоторой лингвистической информации, но учитывает наборы тегов с менее строгими лингвистическими правилами.

Сравнение морфологических анализаторов проводилось в соответствии со следующими этапами:

1. Извлечение токенов и их морфологических характеристик из корпуса;
2. Анализ токенов различными морфологическими анализаторами;
3. Преобразование выходных данных;
4. Сравнение выходных данных с данными из корпуса.

В таблице 5 представлены результаты сравнения четырех морфологических анализаторов.

Таблица 5 – Результаты сравнения морфологических анализаторов

|  |  |  |  |
| --- | --- | --- | --- |
| Анализатор | Режим оценки | Точность | |
| RNC | RU-EVAL |
| AOT | Лемма  Часть речи  Полный тег | 0,822  0,904  0,833 | 0,816  0,911  0,851 |
| PyMorphy2 | Лемма  Часть речи  Полный тег | 0,822  0,915  0,811 | 0,871  0,904  0,841 |
| TreeTagger | Лемма  Часть речи  Полный тег | 0,950  0,932  0,911 | 0,871  0,904  0,742 |
| MyStem | Лемма  Часть речи  Полный тег | 0,961  0,952  0,924 | 0,869  0,882  0,863 |

## Метод условных случайных полей

К числу методов, успешно применяемых в обработке текста, относится метод случайных Марковских полей и его модификация — метод условных случайных полей (CRF — Conditional Random Fields), который нашел широкое применение в лингвистических приложениях, требующих разметки больших объемов текста на основе некоторых параметров. Чаще всего этот метод применяют в задачах распознавания специальных терминов, именных групп, поверхностного синтаксиса и т.п. Также данный метод находит свое применение в задачах разрешения лексической омонимии, анафорических ссылок, сентиментанализе, машинном переводе.

Метод CRF хорошо исследован для английского, немецкого, арабского, китайского и некоторых других широко распространенных языков. К сожалению, для русского языка этот метод еще не нашел столь широкого применения.

Метод CRF относится к статистическим лингвистическим методам. Данный метод является одной из возможных реализаций Марковских случайных полей.

Марковским случайным полем или Марковской сетью называют графовую модель, которая используется для представления совместных распределений набора нескольких случайных переменных. Формально Марковское случайное поле состоит из следующих компонентов:

* неориентированный граф или фактор-граф G=(V, E), где каждая вершина v∈V является случайной переменной Х и каждое ребро {u,v}∈E представляет собой зависимость между случайными величинами u и v.
* набор потенциальных функций (potential function) или факторов {k}, одна для каждой клики в графе G (полный подграф). Функция k ставит каждому возможному состоянию элементов клики в соответствие некоторое неотрицательное вещественнозначное число.

Вершины, не являющиеся смежными, должны соответствовать условно независимым случайным величинам. Группа смежных вершин образует клику, набор состояний вершин является аргументом соответствующей потенциальной функции.

Совместное распределение набора случайных величин Х={xk} в Марковском случайном поле вычисляется по формуле 5:

, (5)

где – потенциальная функция, описывающая состояние случайных величин в k-ой клике;

Z – коэффициент нормализации, который вычисляется по формуле 6:

(6)

Множество входных лексем X={xt} и множество соответствующих им типов Y={yt} в совокупности образуют множество случайных переменных V=X∪Y. Для решения задачи извлечения информации из текста достаточно определить условную вероятность P(Y|X). Потенциальная функция имеет вид:

, (7)

где Σ{λk} – вещественнозначный параметрический вектор;

Σ{fk(yt, yt-1, xt)} – набор признаковых функций.

Тогда линейным условным случайным полем называется распределение вероятности вида:

(8)

Коэффициент нормализации Z(x) вычисляется по формуле 9:

(9)

Метод CRF, как и метод MEMM (Maximum Entropy Markov Models), относится к дискриминативным вероятностным методам, в отличие от генеративных методов, таких как HMM (Hidden Markov Models) или метод «Наивного Баеса».

По аналогии с MEMM, выбор факторов-признаков для задания вероятности перехода между состояниями при наличии наблюдаемого значения xt зависит от специфики конкретных данных, но в отличии от того же МЕММ, CRF может учитывать любые особенности и взаимозависимости в исходных данных. Вектор признаков L={λk} рассчитывается на основе обучающей выборки и определяет вес каждой потенциальной функции. Для обучения и применения модели используются алгоритмы, аналогичные алгоритмам HMM: Витерби и его разновидность — алгоритм «вперед-назад».

Cкрытую Марковскую модель можно рассматривать как частный случай линейного условного случайного поля. В свою очередь, условное случайное поле можно рассматривать как разновидность Марковского случайного поля (Рисунок 22).

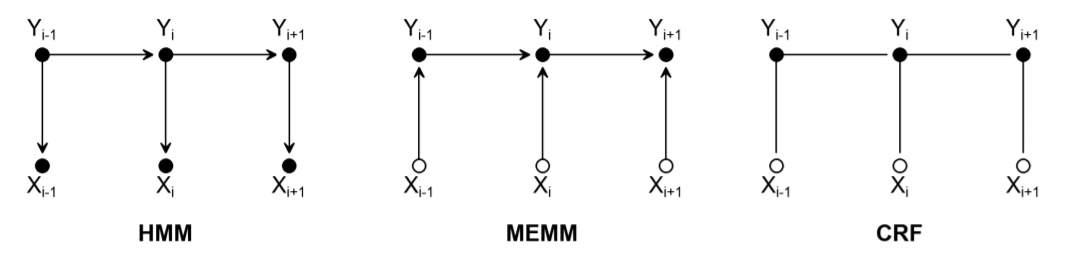


Рисунок 22 – Изображение в виду графов для методов HMM, MEMM и CRF

В условных случайных полях отсутствует т.н. label bias problem — ситуация, когда преимущество получают состояния с меньшим количеством переходов, так как строится единое распределение вероятностей и нормализация производится в целом, а не в рамках отдельного состояния. Это, безусловно, является преимуществом метода: алгоритм не требует предположения независимости наблюдаемых переменных. Кроме того, использование произвольных факторов позволяет описать различные признаки определяемых объектов, что снижает требования к полноте обучающей выборки. Недостатком подхода CRF является вычислительная сложность анализа обучающей выборки, что затрудняет постоянное обновление модели при поступлении новых обучающих данных.

# ЗАКЛЮЧЕНИЕ

При разработке модуля морфологического анализа русскоязычного языка были получены следующие результаты:

* Была исследована предметная область – морфология русского языка;
* Были сформулированы требования к модулю морфологического анализа;
* Было проведено сравнение морфологических анализаторов;
* Была выбрана модель машинного обучения;
* Разработанный модуль была интегрирован в веб-сервис;
* Модуль был протестирован и показал результаты, значительно превосходящие исходный морфологический анализатор.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Streamlit 0.82.0 documentation [Электронный ресурс] // https://docs.streamlit.io/en/stable/ (Дата обращения: 02.05.2020).
2. Документация PyMorphy2 [Электронный ресурс] // https://pymorphy2.readthedocs.io/en/stable/user/ (Дата обращения: 02.05.2020).
3. Морфологический стандарт MorphoRuEval-2017 [Электронный ресурс] // https://github.com/dialogue-evaluation/morphoRuEval-2017/blob/master/morphostandard (Дата обращения: 03.05.2020).
4. Зализняк А.A. Грамматический словарь русского языка. — М., Русский язык, 1980. – 880 с
5. Sklearn-crfsuite documentation [Электронный ресурс] // https://sklearn-crfsuite.readthedocs.io/en/latest/ (Дата обращения: 20.05.2020).
6. Вероятностная модель токенизации в проекте Открытый корпус [Электронный ресурс] // https://docplayer.ru/29320594-Veroyatnostnaya-model-tokenizacii-v-proekte-otkrytyy-korpus.html (Дата обращения: 09.05.2020).
7. Основные метрики задач классификации в машинном обучении [Электронный ресурс] // https://webiomed.ai/blog/osnovnye-metriki-zadach-klassifikatsii-v-mashinnom-obuchenii/ (дата обращения: 04.04.2020).
8. Morphological Analysis for Russian: Integration and Comparison of Taggers [Электронный ресурс] // https://www.hse.ru/data/2016/06/10/1117658168/morphological-analysis-russian-1.pdf (Дата обращения: 01.05.2020).
9. Automatic morphological analysis for Russian: a comparative study [Электронный ресурс] // http://www.dialog-21.ru/media/3473/dereza.pdf (Дата обращения: 20.05.2020).
10. Официальный сайт AOT [Электронный ресурс] // http://aot.ru/ (Дата обращения: 10.05.2020).
11. TreeTagger [Электронный ресурс] // https://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/ (Дата обращения: 10.05.2020).
12. Открытый корпус [Электронный ресурс] // http://opencorpora.org/ (Дата обращения: 10.05.2020).
13. MyStem [Электронный ресурс] // https://yandex.ru/dev/mystem/ (Дата обращения: 10.05.2020).

# ПРИЛОЖЕНИЕ А Графическая часть

А.1. Граф-диалога;

А.2. Диаграмма сравнения PyMorphy2 и полученной CRF модели;

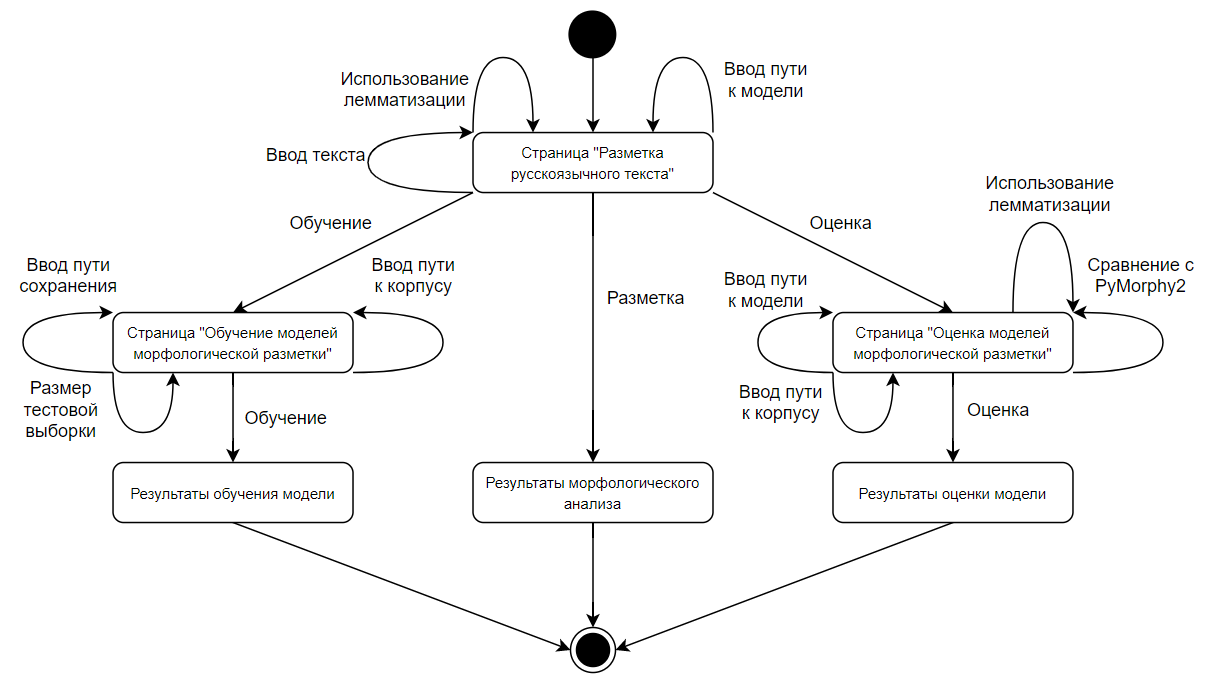
А.3. Методы морфологического анализа;

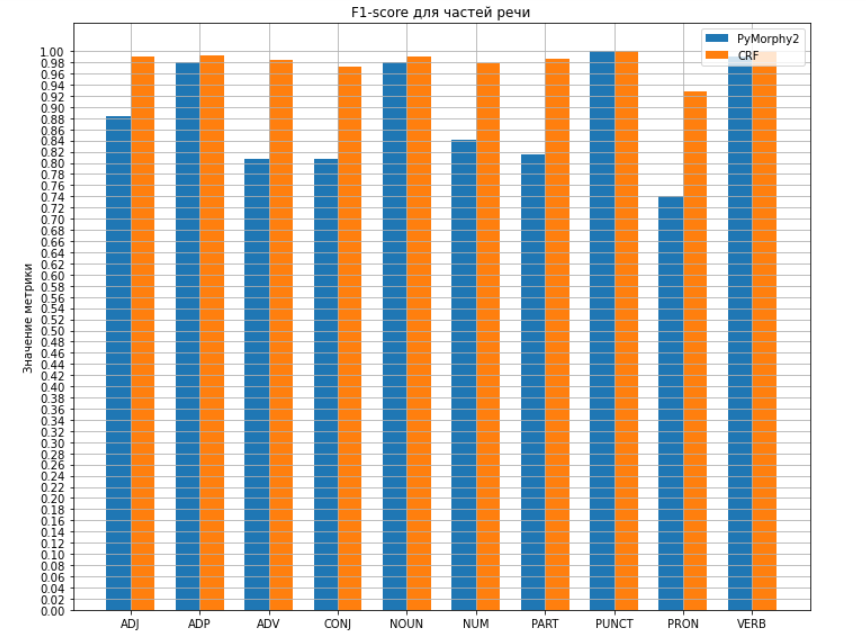
А.4. Предметная область;

А.5. Блок-схема алгоритма морфологического анализа;

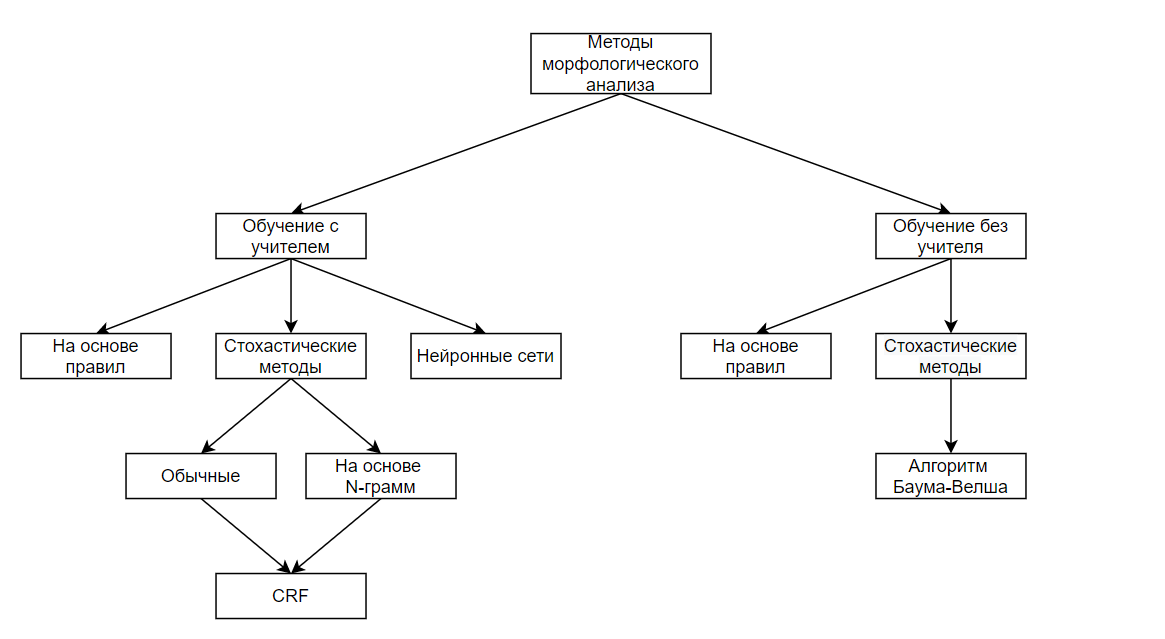
А.6. Экранные формы;

А.1. Граф-диалога

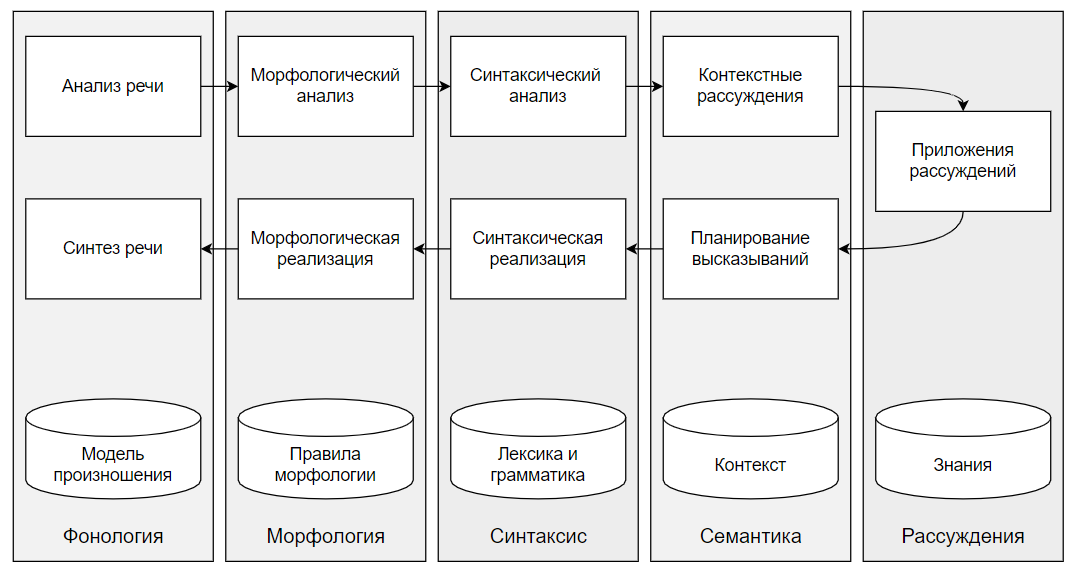


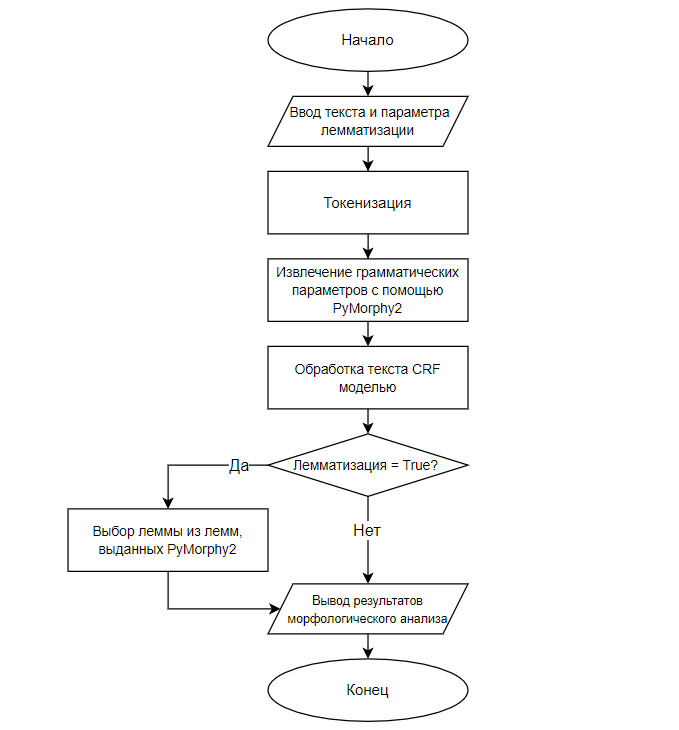
А.2. Диаграмма сравнения PyMorphy2 и полученной CRF модели 

А.3. Методы морфологического анализа

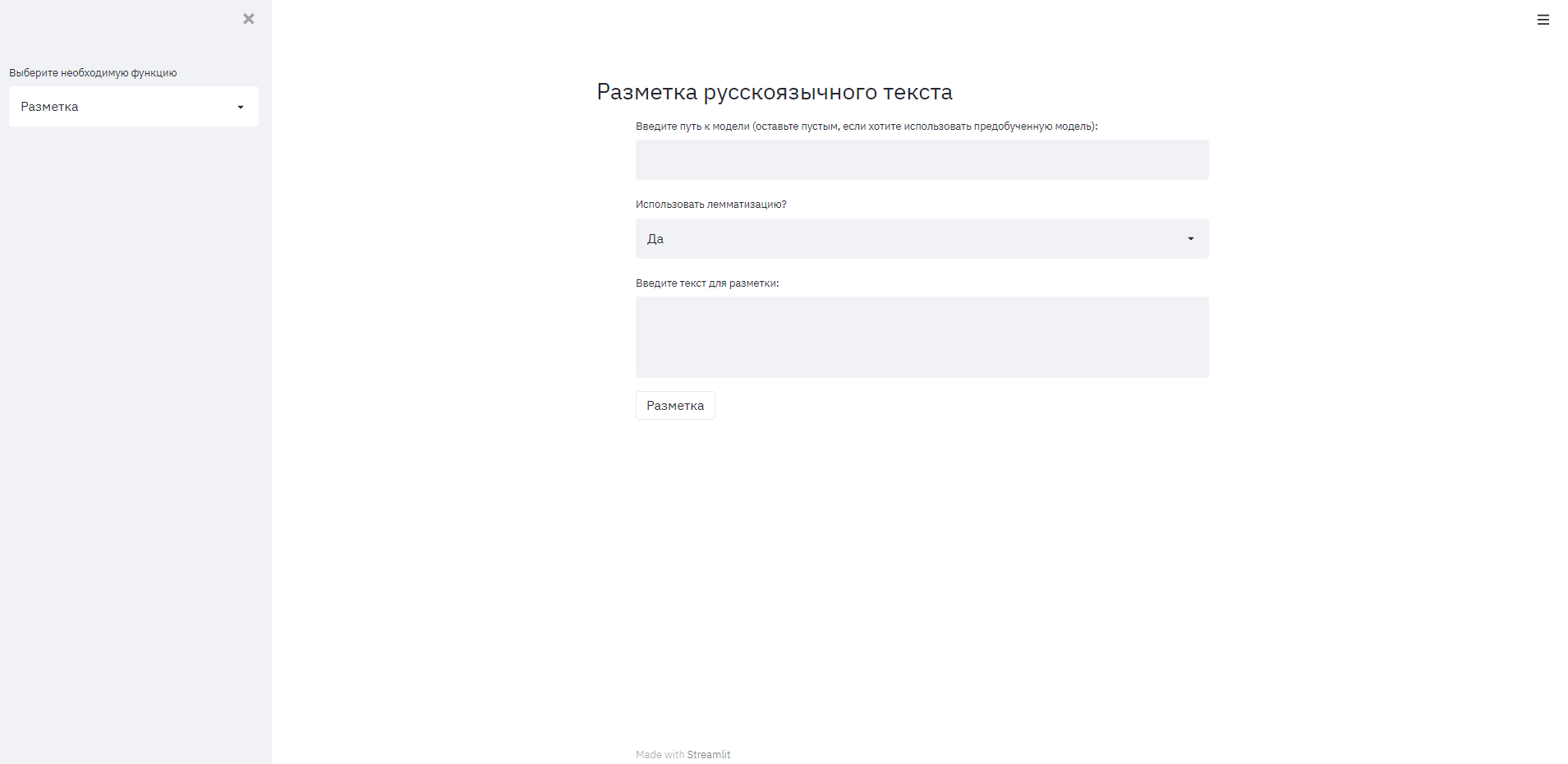


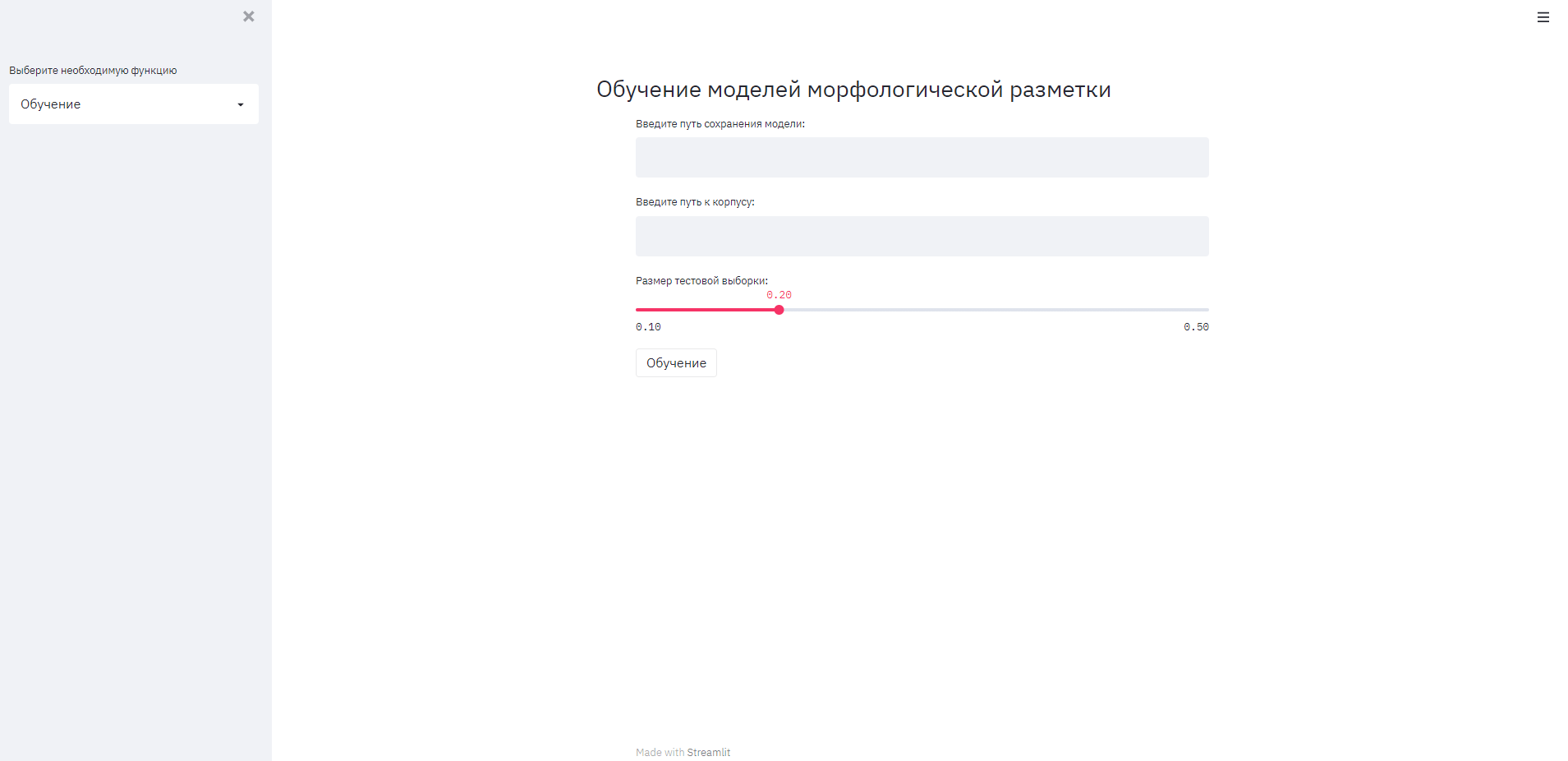
А.4. Предметная область

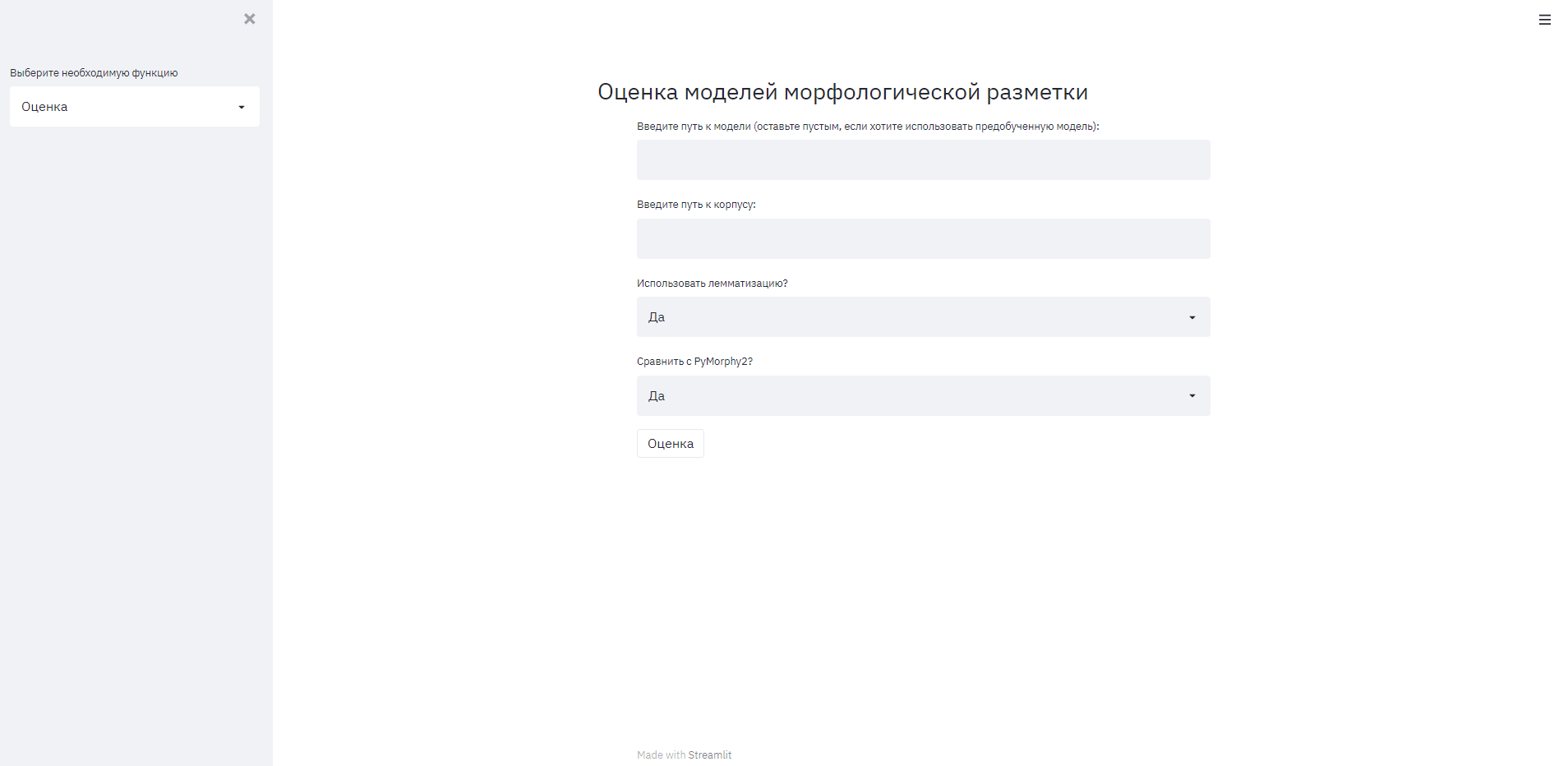


**А.5. Блок-схема алгоритма морфологического анализа** 

А.6. Экранные формы







# ПРИЛОЖЕНИЕ В Техническое задание

МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

им. Н.Э. Баумана

Кафедра «Системы обработки информации и управления»

|  |  |
| --- | --- |
|  | Утверждаю  Научный руководитель |
|  | \_\_\_\_\_\_\_\_\_\_\_ Михеев В.А.  "\_\_"\_\_\_\_\_\_\_\_\_\_2021 г. |

**Модуль морфологического анализа**

**русскоязычного текста**

Техническое задание

(вид документа)

писчая бумага

(вид носителя)

6

(количество листов)

|  |  |
| --- | --- |
| ИСПОЛНИТЕЛЬ: |  |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Абибок Матвей Александрович |
| "\_\_"\_\_\_\_\_\_\_\_\_\_\_2021 г. |  |

Москва – 2021

1. **Наименование**

Модуль морфологического анализа русскоязычного текста.

1. **Основание для разработки**

Основанием для разработки является задание на выпускную работу, подписанное руководителем выпускной работы и утвержденное заведующим кафедрой. Задание утверждено кафедрой ИУ5 МГТУ им. Н.Э. Баумана.

1. **Исполнитель**

Студент МГТУ им. Н.Э. Баумана группы ИУ5-83Б Абибок Матвей Александрович.

1. **Назначение и цель разработки**

Данная разработка предназначена для реализации научно-исследовательской работы кафедры «Системы обработки информации и управления».

Цель разработки модуля морфологического анализа русского текста – достижение достаточного уровня точности морфологической разметки для дальнейшего использования в сфере обработки естественного языка.

1. **Содержание работы**
   1. **Задачи, подлежащие решению**

* Исследование предметной области;
* Определение функциональных требований;
* Выбор основного морфологического анализатора;
* Выбор моделей машинного обучения, улучшающих качество предсказаний;
* Выбор данных для обучения моделей;
* Программная реализация;
* Тестирования программного модуля;
* Разработка эксплуатационной документации к программному модулю.
  1. **Требования к функциональности программного изделия**

Разработанный модуль должен предоставлять пользователю следующие функции:

* + 1. Морфологическая разметка и лемматизация русскоязычного текста;
    2. Обучение моделей;
    3. Оценка качества моделей.
  1. **Требования к входным и выходным данным** 
     1. Входные данные:
* Размеченный текст на русском языке для обучения и оценки точности;
* Текст на русском языке, который необходимо разметить.
  + 1. Выходные данные:
* Данные морфологической разметки;
* Леммы слов;
* Метрики качества.
  1. **Требования к надежности**

Модуль не должен выдавать ошибок, не предусмотренных в работе модуля.

* 1. **Требования к составу технических средств**
     1. **Требования к аппаратному обеспечению**

Для высокой эффективности работы системы компьютер должен быть оснащён:

* Оперативной памятью не менее 4 Гбайт;
* Жестким диском не менее 10 Гбайт;
* Процессором Intel core i3 третьего поколения и выше;
* Клавиатурой и мышью для возможности личного управления компьютером.
  + 1. **Требования к программному обеспечению**
* Операционная система не ниже Windows 7;
* Браузер Google Chrome не ниже версии 70;
* Python 3.6.

1. **Этапы разработки**

График приведен в соответствии с учебными неделями МГТУ им. Н.Э. Баумана 4 курса 8 семестра.

|  |  |  |
| --- | --- | --- |
| **№ п/п** | **Наименование этапов выпускной квалификационной работы** | **Сроки выполнения этапов** |
| 1. | Задание на выполнение работы. Формулирование проблемы, цели и задач работы. | *26.01.2021*  *Планируемая дата* |
| 2. | Разработка и формулирование ТЗ | *12.03.2021*  *Планируемая дата* |
| 3. | 1 часть. Постановка задач разработки | *26.03.2021*  *Планируемая дата* |
| 4. | 2 часть. Конструкторско- технологическая часть | *10.05.2021*  *Планируемая дата* |
| 5. | Разработка и показ макета программы | *11.05.2021*  *Планируемая дата* |
| 6. | 3 часть. Заключение, приложения, оформление работы | *25.05.2021*  *Планируемая дата* |
| 7. | Подготовка доклада и презентации. Предзащита | *30.05.2021*  *Планируемая дата* |
| 8. | Заключение руководителя | *30.05.2021*  *Планируемая дата* |
| 9. | Допуск работы к защите на ГЭК (нормоконтроль) | *01.06.2021*  *Планируемая дата* |
| 10. | Защита работы на ГЭК | *22.06.2021*  *Планируемая дата* |

1. **Техническая документация, предъявляемая по окончании работы**
2. Техническое задание;
3. Расчётно-пояснительная записка;
4. Программа и методика испытаний;
5. Руководство пользователя.
6. **Порядок приема работы**

Прием и контроль программного изделия осуществляется в соответствие с методикой испытаний (см. документ «Программа и методика испытаний»).

1. **Дополнительные условия**

Данное техническое задание может дополняться и уточняться в установленном порядке.

# ПРИЛОЖЕНИЕ С Программа и методика испытаний

МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

им. Н.Э. Баумана

Кафедра «Системы обработки информации и управления»

|  |  |
| --- | --- |
|  | Утверждаю  Научный руководитель |
|  | \_\_\_\_\_\_\_\_\_\_\_ Михеев В.А.  "\_\_"\_\_\_\_\_\_\_\_\_\_2021 г. |

**Модуль морфологического анализа**

**русскоязычного текста**

Программа и методика испытаний

(вид документа)

писчая бумага

(вид носителя)

4

(количество листов)

|  |  |
| --- | --- |
| ИСПОЛНИТЕЛЬ: |  |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Абибок Матвей Александрович |
| "\_\_"\_\_\_\_\_\_\_\_\_\_\_2021 г. |  |

Москва – 2021

1. **Объект испытаний**

Объектом испытаний является программный модуль морфологического анализа русскоязычного текста.

1. **Цель испытаний**

Испытания проводятся с целью проверки соответствия клиента требованиям к функциональным характеристикам, описанным в п. 5.2 Технического задания.

1. **Состав предъявляемой документации**

На испытания программного продукта предъявляются следующие документы:

* Техническое задание
* Программа и методика испытаний

1. **Технические требования**
   1. **Требования к аппаратному обеспечению**

Для высокой эффективности работы системы компьютер должен быть оснащён:

* Оперативной памятью не менее 4 Гбайт;
* Жестким диском не менее 10 Гбайт;
* Процессором Intel core i3 третьего поколения и выше;
* Клавиатурой и мышью для возможности личного управления компьютером.
  1. **Требования к программному обеспечению**
* Операционная система не ниже Windows 7;
* Браузер Google Chrome не ниже версии 70;
* Python 3.6.

1. **Методы испытаний**

Испытания системы будут проводиться в следующем порядке:

1. Взаимодействие с web-интерфейсом страницы.

2. Просмотр результатов работы системы.

Приемочные испытания включают проверку:

1. Полноты и качества реализации функций, указанных в ТЗ

Последовательность проведения испытаний:

|  |  |  |  |
| --- | --- | --- | --- |
| **№ п.п.** | **№ пункта ТЗ** | **Действие** | **Ожидаемый результат** |
| 1 | 5.2.1  Морфологическая разметка и лемматизация русскоязычного текста. | Запустить модуль. Ввести путь к модели или оставить поле пустым, чтобы использовать базовую модель. В выпадающем списке выбрать, использовать лемматизацию или нет. Ввести текст для разметки и нажать кнопку «Разметка». | ПМ запущен. Открытие страницы «Разметка русскоязычного текста» с полями для ввода пути к модели и текста, выпадающим списком для использования лемматизации и кнопкой «Разметка».  При нажатии кнопки «Разметка» выводится размеченный текст. Если лемматизация включена, также выводятся леммы слов. |
| 2 | 5.2.2  Обучения моделей. | Перейти на страницу «Обучение моделей морфологической разметки» по нажатию кнопки «Обучение» в выпадающем списке. Ввести путь сохранения модели и путь к корпусу. Ползунком выбрать размер тестовой выборки и нажать кнопку «Обучение». | Открытие страницы «Обучение моделей морфологической разметки» с полями для ввода пути сохранения модели и пути к корпусу, ползунком для выбора размера тестовой выборки и кнопкой «Обучение».  При нажатии кнопки «Обучение» модель обучается и сохраняется по введенному пути. |
| 3 | 5.2.3  Оценка качества моделей. | Перейти на страницу «Оценка моделей морфологической разметки» по нажатию кнопки «Оценка» в выпадающем списке. Ввести путь к модели или оставить поле пустым, чтобы использовать базовую модель. В выпадающих списках выбрать, использовать лемматизацию и сравнение с PyMorphy2 или нет. Нажать кнопку «Оценка». | Открытие страницы «Оценка моделей морфологической разметки» с полями для ввода пути к модели и пути к корпусу и выпадающими списками для использования лемматизации и сравнения с PyMorphy2.  При нажатии кнопки «Оценка» вывод метрик качества модели. Если включена лемматизация, выводятся метрики её оценки. Если включено сравнение с PyMorphy2, выводятся метрики оценки PyMorphy2 и график сравнения. |

1. **Результат испытаний**

Основой испытаний является демонстрация работы основных функций программного модуля.

Испытание считается пройденным успешно, если в процессе демонстрации все действия прошли успешно и результат соответствовал ожидаемому с учетом используемых данных.