МИНОБРНАУКИ РОССИИ

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ**

**ВЫСШЕГО ОБРАЗОВАНИЯ**

**“ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ”**

Факультет компьютерных наук

Кафедра Технологий обработки и защиты информации

Отчетпо *\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_* практике

*указать вид практики*

Разработка системы дообучения русскоязычной NER модели для выделения именованных сущностей

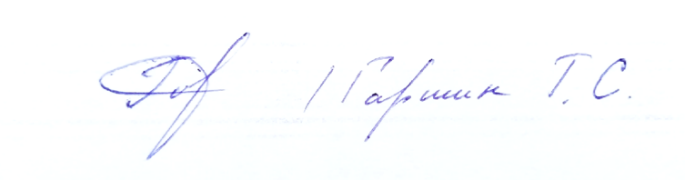
Направление\_\_09.03.02 Информационные системы и технологии\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*Шифр, наименование направления подготовки / специальности*

Профиль\_\_\_\_\_ Обработка информации и машинное обучение \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Зав. кафедрой \_д.т.н., проф. Сирота А.А.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_.\_\_.20\_\_

*Подпись, расшифровка, ученая степень, звание*

Студент\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.\_\_.20\_\_

*Подпись, расшифровка подписи*

Руководитель практики от ВГУ\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.\_\_.20\_\_

*Подпись, расшифровка подписи, ученая степень, звание*

Руководитель практики от предприятия \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.\_\_.20\_\_

*Подпись, расшифровка подписи, ученая степень, звание*

Воронеж2022

Оглавление

1. [Введение 3](#_Toc83733900)

[2.  Постановка задачи 6](#_Toc83733901)

[3.  Выбор инструментов и средств разработки 7](#_Toc83733902)

[4.  Подходы к обучению и предобработке данных 7](#_Toc83733903)

[5.  Сравнение языковых предобученных моделей 8](#_Toc83733904)

[6.  Создание и подключение крафтового датасета 9](#_Toc83733905)

[Заключение 11](#_Toc83733907)

[Список литературы 11](#_Toc83733908)

# Введение

Последние несколько лет все более актуальной становиться задача NLP (Natural Language Processing). Согласно данным компании Research and Market объём мирового рынка обработки естественного языка увеличился с $10,2 млрд. в 2019 году до $26,4 млрд. К 2024 году при совокупном годовом темпе роста (CAGR) на 21,0% в течение прогнозируемого периода. Основными факторами роста рынка NLP являются:

* Повышение спроса на интеллектуальные устройства;
* Развитие облачных решений и приложений на основе NLP, которые улучшают процесс обслуживания клиентов;
* Увеличился спрос и технологические инвестиции в отрасль.

Для каких задач применяется NLP сегодня?

1. Машинный перевод;
2. Анализ текстов

* Классификация (бинарная и мультиклассовая);
* отражение содержания ;
* анализ тональности.

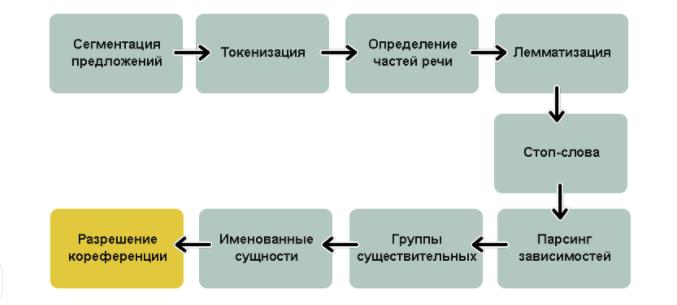
1. Распознавание и синтез речи

* Разработка диалоговых систем умные помощники (Яндекс.Алиса, Siri, Alexa);
* чат-боты — текстовые системы, следующие сценариям диалога (бот «Связного», Facebook Messenger);
* QA-системы.

1. Выделение сущностей и фактов (Named-entity recognition, NER)

Задача NER — понять, что участок текста «1999 года» является датой, «Иван Петров» — персоной, а «Газпром» — организацией.

Реализация любой сложной задачи обычно означает построение пайплайна (Рис.1):



Пример конвейера для NLP задач

Суть этого подхода в том, чтобы разбить задачу на ряд последовательных подзадач и решать каждую из них отдельно. В построении пайплайна можно условно выделить две части: предобработку входных данных (обычно занимает больше всего времени) и построение модели.

1. Первые два шага пайплайна, которые выполняются для решения практически любых задач NLP, — это сегментация (деление текста на предложения) и токенизация (деление предложений на токены, то есть отдельные слова).

2. Вычисление признаков каждого токена. Вычисляются контекстно-независимые признаки токена. Это набор признаков, не зависящих от соседних с токеном слов.

* Один из самых часто использующихся признаков — часть речи.



Пример токенизации текста

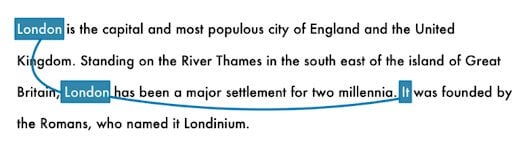
* Для языков со сложной морфологией (русский язык) также важны морфологические признаки: например, в каком падеже стоит существительное, какой род у прилагательного. Из этого можно сделать выводы о структуре предложения.
* Морфология также нужна для приведения слов к начальной форме, с помощью которой мы можем уменьшить объём признакового пространства.

3. Определение значимости и фильтрация стоп-слов. В русском и английском языках очень много вспомогательных слов, например «and», «the», «a». При статистическом анализе текста эти токены создают много шума, так как появляются чаще, чем остальные. Поэтому их отмечают как стоп-слова и отсеивают.



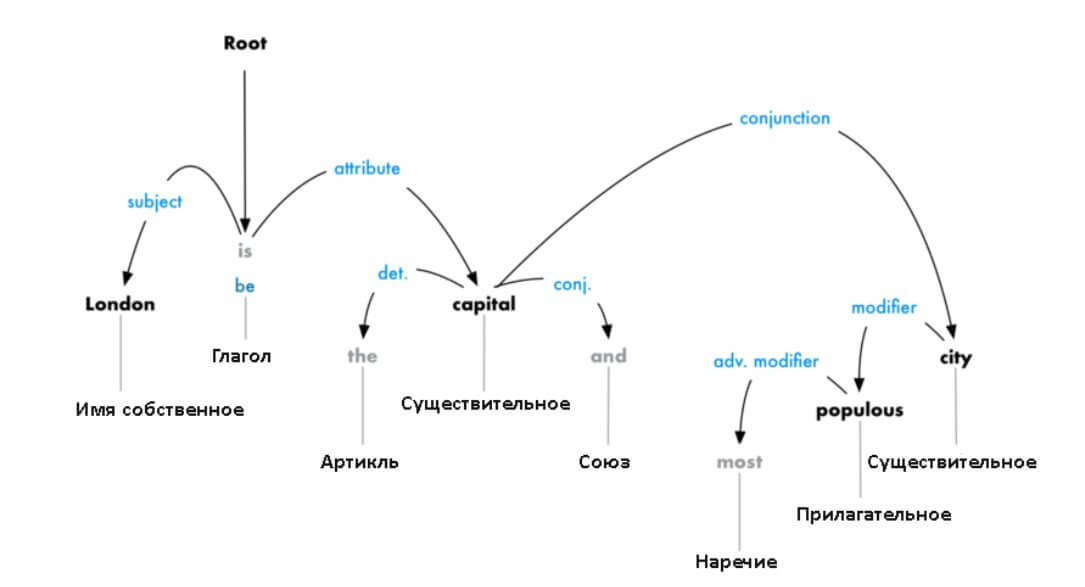
Пример фильтрации стоп-слов в предложении (серые)

4. Разрешение кореференции. В русском и английском языках очень много местоимений вроде he, she, it или ты, я, он и т. д. Это сокращения, которыми мы заменяем на письме настоящие имена и названия. Человек может проследить взаимосвязь этих слов от предложения к предложению, основываясь на контексте. Но NLP-модель не знает, что означают местоимения, ведь она рассматривает всего одно предложение за раз.



Пример предложения, содержащего кореференцию

5. Парсинг зависимостей. Конечная цель этого шага — построение дерева, в котором каждый токен имеет единственного родителя. Корнем может быть главный глагол. Также нужно установить тип связи между двумя словами:



Выделение зависимых слов в предложении древовидным разбором

Это дерево парсинга демонстрирует, что главный субъект предложения — это существительное «London». Между ним и «capital» есть связь «be». Вот так мы узнали, что Лондон — это столица. Если бы мы проследовали дальше по веткам дерева (уже за границами схемы), то могли бы узнать, что «London is the capital of Great Britain».

6. Перевод обработанного текста в векторную форму. Данный шаг позволяет сформировать векторные представления слов. Таким образом, у слов, используемых в одном и том же контексте, похожие векторы.

7. Построение модели в зависимости от поставленной цели. Например, модель NER, для выделения сущностей и фактов.

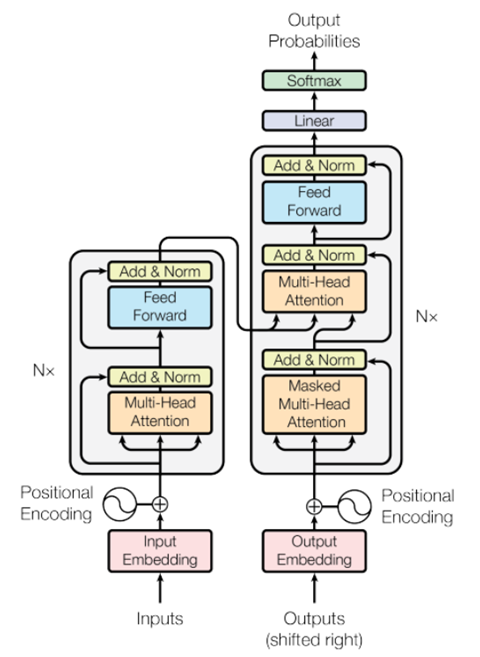
Постановка задачи

В ходе практики мне была поставлена задача разработать систему дообучения русской NER модели для выделения требуемых для конкретных задач сущностей.

Проблема разработки состоит в том, что в первую очередь NER модели обучаются на английских и испанских текстах (как самых распространенных в мире). Для русского языка существует ряд моделей:

* RuBERT, Russian, cased, 12-layer,
* SlavicBERT, Slavic (bg, cs, pl, ru), cased, 12-layer
* bert-base-multilingual-cased
* ner\_rus\_bert\_torch

**BERT** — двунаправленная модель с transformer-архитектурой. Модель также предобученна на двух задачах без учителя — моделирование языковых масок и предсказание следующего слова в предложении. Новая архитектура не является по сути ни RNN, ни CNN. Новая интересная вещь в этих слоях — это Multi-head attention. Это специальный новый слой, который дает возможность каждому входному вектору взаимодействовать с другими словами через attention mechanism, вместо передачи hidden state как в RNN или соседних слов как в CNN.



Transformer-архитектура

Выбор инструментов и средств разработки

В качестве языка программирования был выбран Python, так как именно на нем развивается анализ данных, под него существует большое количество библиотек и для него написаны NER модели BERT. В качестве IDE был выбран Google Collab, так как он предоставляет доступ к серверу с GPU, что просто необходимо при обучении модели.

На основе руководства к моделям и в ходе ряда дополнительных экспериментов подобранны оптимальные аргументы обучения: batch size=60, 10000 примеров ,15 эпох.

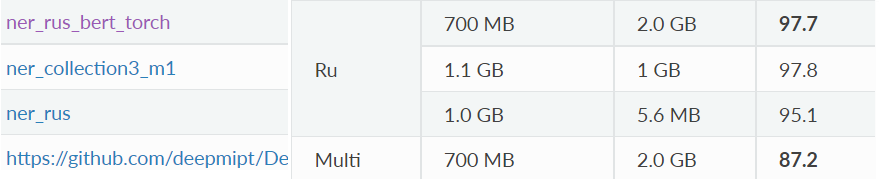
Подходы к обучению и предобработке данных

Для дальнейшего обучения модели нам нужно было определиться с предобработкой данных для обучения. Для этой задачи была найдено несколько готовых решений:

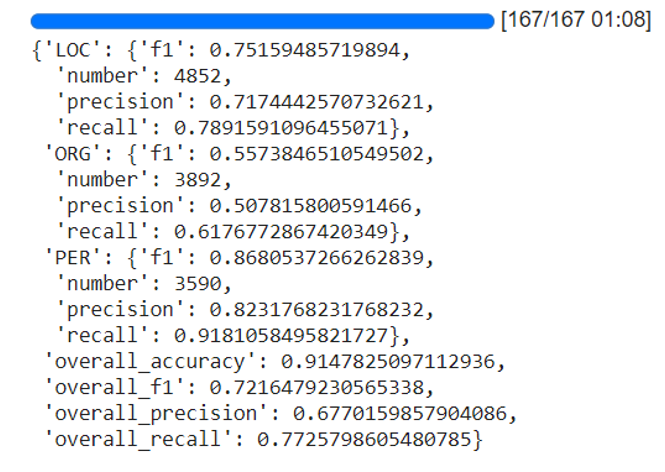
* Библиотека configs, решение от деппавлова. Для каждой предобученной модели формируется конфигурационный файл с заранее прописанными переменными весами модели. Эта библиотека была использована в нашем решении.
* Библиотека tokenaizer, решение от haggin face. Помогает разбивать текст на набор токенов (на предложения и слова). После ряда экспериментов это решение в проект включено не было, так как регулирование уровня токенизации было невозможно, что приводило к разбиению не только предложений на слова, но и слова на слоги, что в процессе обучения вводило модель в заблуждение.
* Библиотека datasets, решение от haggin face. Эта библиотека для легкого доступа к готовым наборам данных, а также оценочным показателям для обработки естественного языка. Datasets была использована нами на уровне тестирования моделей и подбора оптимальных аргументов.
* Библиотека sklearn эффективный инструмент для предиктивного анализа данных. Была использована в проекте для оценивания модели после обучения.

Сравнение языковых предобученных моделей

В начале для работы была выбрана модель ner\_rus\_bert\_torch, тк она специализировалась на русском языке (процесс предобучения модели проходил на датасете собранном с русскоязычной википедии) и имела высокие показатели распознавания после предобучения по F1(Рис.7)

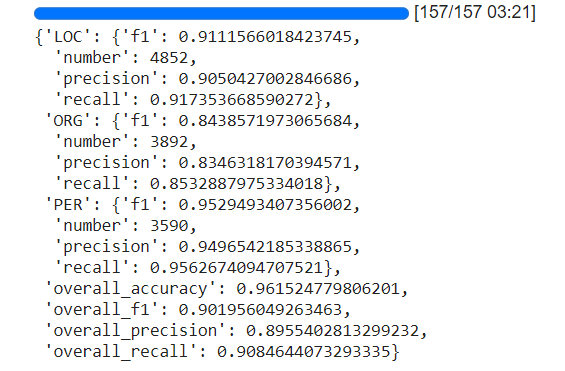


Сравнение предобученных моделей для русского языка

Реальность внесла свои коррективы и при собственном детальном тестировании модели после обучения на размеченном к обучению корпусе русского языка результаты были уже значительно скромнее. (Рис.8)

Результат обучения модели ner\_rus\_bert\_torch на чужом датасете

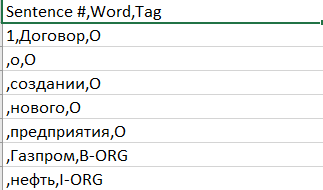
Как видно из результатов тестирования по классам распознавания, результаты на порядок ниже заявленных. Уже в этот момент появилась потребность создания и подключения к процессу обучения собственного датасета. Как было сказано выше, в этот момент мы использовали чужой датасет (wikiann ru). В ходе длительных тестов и сравнений, для работы была выбрана bert-base-multilingual-cased. Не смотря на скромные заявленные показатели, в работе она показала себя очень хорошо и даже предварительные тесты модели показали хороший потенциал для дальнейшего дообучения.(Рис.9)



Результат обучения модели bert-base-multilingual-cased на чужом датасете

Создание и подключение крафтового датасета

Следующая задача, которую требовалось решить- это подключение крафтового (удовлетворяющего нашим требованиям) датасета к модели, этим шагом мы смогли бы регулировать кол-во классов распознавания (что бы модель выделяла из текста именно интересующие нас теги). Такое решение было найдено. С помощью библиотеки pandas мы считываем заранее размеченный csv файл. (Рис.10)

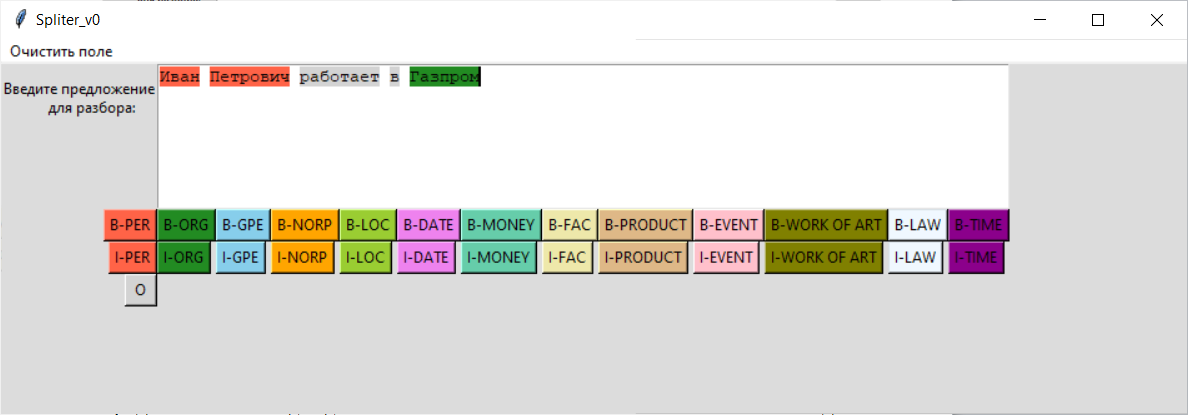


Содержиме датасета для дообучения

Внутри файла есть три колонки:

1. Номер предложения;
2. Само слово;
3. Тег, которым слово должно быть помечено.

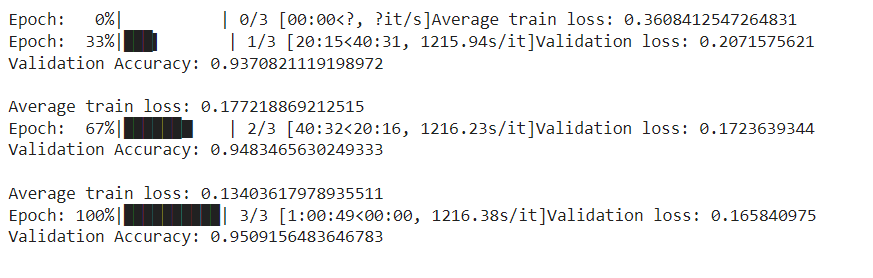
Размечать предложения вручную оказалось очень долго и трудоемко. Было решено создать специальное приложение, которое бы упростило и ускорило процесс. Ниже представлен прототип приложения (Рис.11). Человеку достаточно выделить слово мышью и выбрать на панели управления соответствующий слову тег, слово будет автоматически добавлено в датасет с соблюдением правил разметки.



Прототип приложения разметки датасета

С помощью данного приложения и при помощи студентов 3курса РГФ была организована масштабная разметка датасета для дообучения модели. Студентам было сформулирована задача, подготовлены варианты (предложения для разметки), проведено обучение работе с приложением-разметчиком. Так же осуществлялась техподдержка по вопросам работы и адаптации приложения на ПК студентов, и контроль выполнения поставленной задачи. Размеченные предложения были сформированы в общий csv файл и проверены на наличие логических ошибок при разметке.

Итоговый объем датасета составил 27 500 тегов. Был проведен эксперимент по дообучению модели и результат стал сопоставим с результатом обучения англоязычной модели.



Результат обучения модели bert-base-multilingual-cased на своём датасете

Заключение

В ходе данной работы были решены следующие задачи:

* Проведено исследование и сбор информации о понятиях NLP и NER. Рассмотрена и изучена новая архитектура нейронной модели.
* Проведено исследование и сравнение существующих BERT моделей, выбрана наиболее удовлетворяющая требованиям;
* Изучены различные подходы к дообучению модели, выбраны и модернизированы под нашу задачу;
* Подобранны оптимальные настройки аргументов модели для обучения;
* Программно реализовано приложение для разметки датасета;
* Создан крафтовый датасет и проведены эксперименты с его использованием. Результатом такого эксперимента стало повышение качества распознавания модели.

Список литературы

* + - 1. Burtsev M. DeepPavlov: Open-Source Library for Dialogue Systems / M. Burtsev [и др.] // Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics-System Demonstrations (Melborne, Australia, 15-20 июля 2018 г.) – 2018. – С. 122-127.

Руководство для начинающих по BERT для мультиклассифиции. [Электронный ресурс]. – Режим доступа : https://www.machinelearningmastery.ru/beginners-guide-to-bert-for-multi-classification-task-92f5445c2d7c/

Классификация текста с помощью BERT Tokenizer и TF 2.0 в Python [Электронный ресурс]. – Режим доступа : https://pythobyte.com/text-classification-with-bert-tokenizer-and-tf-2-0-in-python-44cafd87/

Техническая литература к библиотеке datasets Python [Электронный ресурс]. – Режим доступа : <https://huggingface.co/docs/datasets/>

Учебник по BERT WORD IMBEDINGS [Электронный ресурс]. – Режим доступа : <https://russianblogs.com/article/60231465736/>

BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding [Электронный ресурс]. – Режим доступа : https://arxiv.org/pdf/1810.04805.pdf