2

# СОДЕРЖАНИЕ

**ВВЕДЕНИЕ 3**

**Глава 1. Погружение в проект 0**

1.1 Выбор темы практики 0

1.2 Описание проекта 0

1.3 Задание на практику 0

**Глава 2. Реализация приложения 0**

2.1 Реализация1 0

2.2 РеализацияN 0

**Глава 3. Описание результатов работы 0**

3.1 Результат1 0

3.2 РезультатN 0

**Глава 4. Описание командной работы 0**

**ЗАКЛЮЧЕНИЕ 0**

**СПИСОК ЛИТЕРАТУРЫ 0**

**ПРИЛОЖЕНИЕ 0**

3.1 Презентация проекта на студенческой научной конференции 0

3.3 Код методов 0

|  |  |  |
| --- | --- | --- |
| Ответственный за соответствие содержания и оформления текста отчёта предъявляемым требованиям | (подпись) | Ступников Андрей Анатольевич |

3

# ВВЕДЕНИЕ

Тип практики: технологическая (проектно-технологическая)

Дата начала: 26.02.2024

Дата окончания: 26.06.2024

Цель практики – получение первичных профессиональных умений и навыков, в том числе первичных умений и навыков научно-исследовательской деятельности, обучающихся по направлению подготовки 02.03.03 Математическое обеспечение и администрирование информационных систем (уровень бакалавриата).

Задачи учебной практики:

* Закрепление теоретических и практических знаний, умений и навыков, полученных на первом и втором курсах обучения;
* Выполнение заданий, предусмотренных программой практики и назначенных руководителем;
* Освоение отдельных компьютерных программ и информационных систем, используемых в профессиональной деятельности;
* Получение навыков работы с периодическими, реферативными и справочными информационными изданиями по прикладной математике и информационным системам;
* Подготовка к осознанному и углубленному изучению дисциплин старших курсов;
* Подготовка и защита в установленный срок отчета по практике.

В результате выполнения технологической (проектно-технологической) учебной практики студент должен обладать следующими компетенциями:

* Анализ и обработка больших данных
* Machine learning
* Создание веб-приложения с помощью фреймворка Flask

4

**Глава 1. Погружение в проект**

**1.1 Выбор темы практики**

Участники нашей команды ранее были знакомы и выполняли совместную работу в рамках хакатона. Нам была интересна сфера машинного и, в частности, глубокого обучения.

Проект основывается на работе с текстом (NLP) – данная тема является одной из наиболее актуальных в сфере машинного обучения. Задача проекта была ясна. Мы владели навыками алгоритмического и объектно-ориентированного программирования, управления проектами, однако мы не владели компетенциями машинного обучения. В ходе практики мы стремились на начальном уровне освоить: машинное обучение, веб-разработку, анализ данных.

**1.2. Описание проекта**

Связность текста – это признак текста, при котором следующее предложение строится на базе предыдущего с помощью языковых средств [1]. Связность также является важным содержательным критерием, от которого зависит качество его реферирования [2] и сложность восприятия читателем [3]. Инструменты для оценки данного критерия могут найти применение в области обработки естественного языка, в частности при анализе сложности восприятия текста читателем.

Существующие сервисы для английского языка используют различные подходы, включающие семантические сети (WordNet), дистрибутивную семантику (LSA, word2vec), эвристические подходы подсчета повторяющихся слов [4-5]. При этом сервисы для русского языка с открытым описанием их принципа работы используют только последний подход, не учитывая словосочетания производных слов, родовидовые и ссылочные отношения, которые также являются средствами связности [6].

Цель работы – разработать веб-приложение для оценки локальной связности перевода на русский язык с помощью модели машинного обучения, учитывающей синтаксические, лексические и морфологические средства связи.

Задачи проекта:

* Сформировать и разметить датасет на основе 3503 текстов публицистической и информационной направленности корпуса несовершенных переводов [1].
* Провести сравнительный анализ регрессионных моделей машинного обучения
* Внедрить модель для оценки локальной связности текста.
* Создать приложение, предоставляющее пользователю оценку связности введённого им текста.

Предметная область. В этом параграфе надо описать предметную область, т.е. сферу деятельности, процессы, явления, систему, взаимосвязи в условиях чего выполняется проект

…

…

Постановка задачи. Дана выборка , элементы которой представляют пары предложений , где – два неповторяющихся предложения и бинарные оценки . D разделяется на На входе модель для обучения получает набор векторных представлений для , dim() = m, , m и набор оценок . На выходе модель выдает набор вещественных оценок , определяющих характер взаимосвязи между значениями целевых переменных и векторных представлений , в котором каждая оценка принадлежит отрезку от 0 до 1. Необходимо построить модель машинного обучения, находящую неизвестное отображение .

Необходимо описать технологии

Для проведения сравнительного анализа рассматривались модели машинного обучения на базе архитектуры transformer BERT [11] (rubert-base-cased [12] и sbert\_large\_nlu\_ru), градиентного спуска (CatBoostRegressor), байесовского подхода (Bayessian Ridge), метода LASSO (LassoLarsCV) и сетей с долгой краткосрочной памятью LSTM.

Мы освоили использование этих моделей. Среди них были модели (CatBoostRegressor, LassoLarsCV), архитектура которых изучалась в рамках дисциплины АДиОМО. Также встречались модели незнакомой нам архитектуры BERT, LSTM, принципы работы которых достаточно сложны и их изучение выходит за рамки практики.

Для векторизации данных обучающей и тестовой выборки применялись разные виды представлений, такие как: эмбеддинги sentence transformers исходной модели rubert-base-cased, tf-idf, усредненные по эмбеддингам слов векторы elmo, fasttext и word2vec. Ембеддинг слова (word embedding) – общее название для векторного представления слова из естественного языка.

Самый простой способ векторизации – one-hot encoding. Вектор некоторого слова i из словаря S – вектор длинны , где на i-том месте стоит 1, а на остальных позициях 0. Недостаток данного способа заключается в том, что полученные векторы обладают большой размерностью, также вектор слова никак не связан со смыслом слова. Мы не использовали этот способ векторизации напрямую, но он применяется в других методах.

Способ векторизации word2vec использует нейронные сети и основывается на контексте слова в предложении. Для обучения word2vec используются предобработанный текст (или набор текстов). С помощью скользящего окна размера 2n+1 слов, для каждого слова в тексте определяются другие слова, стоящие к нему на расстоянии не более n. Такие слова для исходного слова считаются близкими по контексту и расстояние между их векторами будет невелико. На основании такой разметки обучается нейронная сеть.

S – множество слов из текста (текстов). – множество слов, находящихся в тексте от слова i на расстоянии не более n, где . На вход нейросети подаётся one-hot encoding вектор слова i, а также правильный ответ в виде one-hot encoding вектора слова . Нейросеть word2vec устроена следующим образом:

Где A – матрица размера B – матрица размера ; j – искомый верктор, состоящий из вещественных чисел в диапазоне [0, 1], отражающий степень контекстной близости слов. Во время обучения . Задача обучения сводится к нахождению коэффициентов матриц A и B. Для каждого эта операция производится раз.

Недостатком word2vec является то, что с его помощью не могут быть представлены слова, не встречающиеся в обучающей выборке. Fasttext построен на схожих идеях, но призван решить проблемы word2vec.

В fasttext слова делятся на n-граммы по буквам скользящим окном и на вход нейросети поступают n-граммы. Вектором слова в таком случае является сумма векторов его n-грамм. Однако такой подход требует больше вычислительных ресурсов.

Word2vec и fasttext не учитывают порядок слов в предложении, в попытке это исправить к задаче векторизации применили идею рекуррентных нейронных сетей – на этом принципе построена модель elmo. Elmo – модель для векторизации, основанная на архитектуре LSTM. Elmo учитывает не только контекст слова, но и его место в предложении, что помогает разделить векторы многозначных слов.