|  |  |
| --- | --- |
| Слайд 1 | Здравствуйте.  ФИО  Заказчик работы – компания “Интерсвязь” |
| Слайд 2  Актуальность | C развитием информационной культуры граждан растет число обращений в социальные и государственные службы.  Обращения граждан часто проходят по длинной цепочке различных инстанций. Это увеличивает время на их рассмотрение и может приводить к утере информации.  Необходимы современные решения на базе машинного обучения.  Актуальность работы подтверждается указом Президента от 10 октября 2019 г. № 490 «О развитии искусственного интеллекта в Российской Федерации». |
| Слайд 3  Цель и задачи исследования | Цель исследования:  Разработка интеллектуальной системы автоматизированной обработки обращений граждан.  Задачи:  1. Осуществить поиск и анализ существующих решений.  2. Разработать алгоритм предобработки данных.  3. Исследовать статистические и нейросетевые модели для работы с текстом.  4. Выбрать метрики качества.  5. Сравнить полученные результаты и выбрать лучшую модель.  6. Разработать приложение, реализующее API для работы с моделью.  7. Выполнить тестирование. |
| Слайд 4  Обзор аналогов | На следующем слайде приведен обзор аналогов системы автоматизированной обработки обращений граждан.  Первый аналог - это система министерства обороны РФ, внедренная в 2016г.  Также здесь приведены проекты, разработанные в 2020г. в рамках конкурса от Агентства стратегических инициатив (АСИ), которые были представлены к внедрению в регионах. |
| Слайд 5  Исходные данные | Исходными данными к работе служит открытый набор данных обращений граждан региона Татарстан.  Набор данных содержит следующие поля, которые были мной использованы:  - описание заявки;   * категория; * последний исполнитель. |

|  |  |
| --- | --- |
| Слайд 6  Диаграмма прецедентов | На следующем слайде приведена диаграмма вариантов использования системы.  Пользователю доступны следующие действия.  1. Выполнить анализ обращения. Этот вариант использования включает в себя вариант использования «ввод текста обращения в форму».  2. Скрыть историю своих запросов.  3. Зарегистрировать нового пользователя.  4. Вывести статистику своих запросов.  5. Просмотреть историю своих запросов к системе.  Администратор – администратор системы, которому доступны все действия пользователя. Кроме того, администратор системы может просматривать историю запросов всех пользователей, статистику по обращениям к системе всех пользователей, а также может очистить историю всех запросов пользователей. |
| Слайд 7  Используемые технологии | Для обучения моделей использовались облачные ресурсы платформы Paperspace Gradient.  Для предобработки текста использовались библиотеки natasha и pymorphy2.  При разработке нейросетевой модели классификатора использовался фреймворк torch и библиотека pytorch-transformers.  При разработке тематической модели использовалась библиотека bigartm.  REST-сервиса разработан на фреймворке Flask, использовалась база данных sqlite.  Пользовательский веб-интерфейс разработан на фреймворке streamlit.  Контейнеризация сервисов выполнена с помощью приложения docker, docker-compose.  Развертывание системы выполнено с использованием технологий AWS ElasticCloud2. |
| Слайд 8  Обучение нейросетевой модели | Мной было проведено исследование различных статистических и нейросетевых моделей.  Лучшие значения метрик показала модель BERT от Google.  Это двунаправленная нейросетевая модель с transformer-архитектурой, т.е. в ее работе используется механизм внимания. В данной работе использовал BERT в задаче классификации.  Также BERT используют в задачах моделирования языка, в вопросно-ответных системах и для других задач NLP.  Модель BERT уже обучена на большом объеме текстов Википедии. Необходимо лишь провести тонкую настройку, обучить нейронную сеть прямого распространения (классификатор) на целевом наборе данных.  На вход нейросети подаются тексты обращений, предварительно токенизированные с помощью токенизатора подслов WordPiece.  Эта последовательность токенов затем продвигается вверх по стеку энкодеров. В обучаемой мной модели RuBERT 12 слоев энкодера.  Каждый слой энкодера, в свою очередь, состоит из нескольких слоёв следующего вида. Сначала используется механизм внимания (self-attention) для учёта глобального контекста. Механизм внимания представляет собой усовершенствованный механизм пулинга, который взвешивает каждый токен в зависимости от его значимости относительно всех остальных токенов. Затем признаки каждого токена нормализуются (применяется функция softmax) и независимо преобразовываются с помощью двухслойной нейронной сети. Также здесь есть связи в обход нелинейностей — это ускоряет процесс обучения за счёт лучшего протекания градиентов. |

|  |  |
| --- | --- |
| Слайд 9  Метрики качества модели | На данном слайде приведены используемые метрики качества.  Метрика accuracy является базовой метрикой, которая измеряет количество верно классифицированных объектов относительно общего количества всех объектов. Она имеет недостаток – она не подходит для несбалансированных классов.  Точность и полнота не зависят, в отличие от accuracy, от соотношения классов, так как рассчитываются для конкретного класса.  Точность можно интерпретировать как способность классификатора отличать положительный класс от других классов, а полнота характеризует способность классификатора обнаруживать положительный класс в принципе.  F-мера — среднее гармоническое точности и полноты.  Также я использовал коэффициент корреляции Мэтьюcа – это сбалансированная мера, которую можно использовать, даже если имеющиеся классы очень разных размеров. Коэффициент возвращает значение от -1 до +1. Коэффициент +1 представляет собой идеальное предсказание, а -1 указывает на полное несоответствие. |
| Слайд 10  Обучение тематической модели | Следующим этапом была обучена иерархическая тематическая модель на основе аддитивной регуляризации (BigARTM).  Это модель обучения без учителя для нахождения распределения тем в коллекции документов.  Если каждый документ коллекции представить в виде «мешка слов», то коллекция представима в виде матрицы F содержания слов в документах.  Тогда тематическая модель представима в виде матричного разложения матрицы F на матрицы и  Матрица представляет вероятности терминов (слов) в каждой теме, а матрица - вероятности тем в каждом документе.  Задача разложения имеет множество решений.  Аддитивная регуляризация тематических моделей (ARTM) основана на максимизации взвешенной суммы логарифма правдоподобия и регуляризаторов. В библиотеке BigARTM эта задача решается итерационно, используется EM-алгоритм.  При обучении модели я ориентировался на следующие метрики качества:  Перплексия – метрика, позволяющая понять, насколько сильно распределение F отличается от равномерного, и, следовательно, насколько хорошо модель описывает данные.  Также я использовал метрику разреженности матрицы Φ и метрику разреженности матрицы Θ.  Необходимо добиваться как можно больших значений разреженности этих матриц. |
| Слайд 11  Диаграмма развертывания системы | Система состоит из двух узлов – сервер и клиент. Взаимодействие между узлами осуществляется через API сервера. Сервер включает в себя следующие компоненты:  - REST-сервер предоставляет API для взаимодействия клиента и сервера. Он взаимодействует с базой данных, нейросетевой моделью классификации BERT и тематической моделью на основе аддитивной регуляризации BigARTM.  - База данных SQLite осуществляет хранение всех анализируемых пользователем обращений граждан и информацию о них.  - Нейросетевая модель осуществляет предсказание категории (классификацию) обращения на основе модели машинного обучения BERT.  - Тематическая модель осуществляет предсказание темы обращения, 10 ключевых слов данной темы и топ-3 исполнителей на основе тематической модели с аддитивной регуляризацией BigARTM. |

|  |  |
| --- | --- |
| Слайд 12  Диаграмма деятельности | Получив REST запрос, содержащий текст анализируемого сообщения, сервер начинает его предобработку. Далее с помощью нейросетевой модели BERT определяется категория обращения, с помощью иерерхической тематической модели BigARTM определяется тема обращения. По полученным данным вычисляются наиболее вероятные исполнители. Результаты заносятся в базу данных на сервере и отправляются клиенту. |
| Слайд 13  Пользовательский веб-интерфейс | На слайде показан общий вид пользовательского веб-интерфейса, реализованного на фреймворке Streamlit.  Слева представлен десктопный вариант, справа- вариант для мобильных устройств.  С веб-интерфейсом и работой системы можно ознакомиться, пройдя по ссылке, представленной на QR-коде. |
| Слайд 14  Тестирование | Перед тестированием REST-сервер и веб-клиент были контейнеризованны с помощью утилиты docker и docker-compose.  Оба сервиса развернуты на виртуальном сервере AWS EC2.  - Произведены попытки ввода некорректных сообщений – пустых, состоящих только из нескольких символов или набранных на английской раскладке.  - Проведены тесты неудачной идентификации пользователя.  Во всех случаях сервером корректно обрабатывалось исключение, пользователю выдавалось соответствующее диагностическое сообщение, пример на слайде. |
| Слайд 15  Основные результаты | Основные результаты:  1. Осуществлен поиск и анализ существующих решений.  2. Выполнена предобработка исходного набора данных.  3. Исследованы различные векторные модели для работы с текстом: статистические и нейросетевые.  4. Выбраны метрики качества.  5. Выполнено сравнение полученных результатов.  Наилучшие результаты классификации обращений граждан показала нейросетевая модель BERT.  Для определения тематики обращений была выбрана иерархическая тематическая модель на основе аддитивной регуляризации.  6. Разработано приложение, реализующее API для работы с моделями. Разработан прототип пользовательского интерфейса.  7. Выполнено тестирование разработанной системы. |
| Слайд 16  Акт о внедрении научно-технической продукции | Также имеется акт о внедрении научно-технической продукции в фирме “Интерсвязь”, который представлен на слайде.  Спасибо за внимание! |