ДОКЛАД НА ПРЕДЗАЩИТУ

Здравствуйте, уважаемые члены комиссии!

Хочу предложить вашему вниманию выпускную квалификационную работу на тему: Разработка интеллектуальной системы автоматизированной обработки обращений граждан.

СЛАЙД 2 Актуальность

C развитием информационной культуры граждан растет число обращений в различные социальные службы и государственные учреждения.

Часто обращения проходят по длинной цепочке различных инстанций.

В таких случаях высока вероятность увеличения времени на рассмотрение обращений, а также могут происходить факты утери обращений. Между тем, ответ на обращение, как правило, должен быть дан как можно скорее.

Имеющиеся технические решения слабо готовы к реализации полного цикла работы с обращениями граждан. Нужны новые современные системы, построенные на основе машинного обучения

СЛАЙД 3 Цель и задачи исследования

Целью моей работы является разработка интеллектуальной системы автоматизированной обработки обращений граждан.

Передо мной стояли следующие задачи:

1) Осуществить поиск и анализ существующих решений,

2) Выполнить предобработку исходного набора данных,

3) Исследовать статистические и нейросетевые модели машинного обучения для данной задачи,

4) Выбрать метрики качества,

5) Сравнить результаты разных моделей.

Также необходимо было разработать приложение, реализующее API работы с моделью, и выполнить тестирование полученного решения.

СЛАЙД 4 Обзор аналогов

В результате проведенного поиска было найдено несколько систем анализа обращений граждан. Но большинство из них либо не содержат в себе алгоритмов классификации на основе машинного обучения, либо данные закрыты.

Принцип работы одного из найденных классификаторов обращений граждан приведен на слайде. Выводятся топ-3 категории обращения, темы обращения и топ-3 исполнителей. Полученные метрики качества данного приведены в таблице.

Слайд 4

На основе поставленных задач, были сформулированы функциональные и нефункциональные требования к системе.

**Функциональные требования:**

1. Поддержка многопользовательского режима.

2. Система должна предоставлять интерфейс для загрузки текста обращений граждан.

3. Система должна для каждого обращения рассчитывать результаты классификации: топ-3 категорий, тем, исполнителей.

4. Система должна предоставлять интерфейс для просмотра обращений граждан.

**Нефункциональные требования:**

1. Система должна иметь возможность запуска на Linux сервере.

2. Система должна быть реализована с помощью языка Python.

3. Система должна хранить обращения граждан в базе данных.

Слайд 5. Диаграмма прецедентов

Имеется 2 эктора, Пользователь и администратор.

Также представлены прецеденты – описания поведения системы с точки зрения эктора.

Варианты использования включают в себя другие варианты использования,

Вариант использования «Просмотреть историю своих запросов к системе» расширяется вариантом использования «Просмотреть необходимое количество запросов».

Слайд 6. Диаграмма развертывания системы.

Имеется 2 узла (вычислительных ресурса): Сервер и клиент, включающие в себя комноненты.Показаны зависимости, REST – сервер использует базу данных, нейросетевую модель и тематическую модель. Сервер предоставляет интерфейс REST API для клиента по протоколу HTTP (прикладной уровень OSI).

Слайд 7. Диаграмма деятельности для варианта использования «Выполнить анализ обращения».

Показаны действия и деятельности для сервера, узлы соединены дугами деятельности.

Показаны начальный узел и узел финала деятельности.

На диаграмме присутствуют 2 узла решения: один узел принимает решение о том, верный пароль или нет, второй узел принятия решения проверяет выходные данные на корректность.

Также имеется узел соединения, который синхронизирует 2 потока.

Один из потоков является действием приема события, что в базу данные произведена успешная запись.

Слайд 8. Схема базы данных сервера.

Показаны 7 отношений. У нас ведется таблица с логом всей поступающей информации, в которую записаны ID пользователя, ID категории, ID темы. Эта таблица (лог) содержит также сообщения, отправленные пользователем на анализ. Имеются также таблицы Пользователь, Категория, Тема, Исполнитель, они содержат полные наименования данных сущностей. Также имеется таблица Оценок, которая содержит ID категории, темы и исполнителя, а также значение оценки вероятности исполнителя. То есть по данным категории, теме и исполнителю мы можем получить оценку вероятности, которая рассчитывается заранее для каждого исполнителя и для каждой из 260 подкатегории (темы).

Слайд 9. Диаграмма состояния пользовательского интерфейса.

Для интерфейса пользователя имеется 3 состояния: Активное состояние, состояние обработки ошибки и состояние ожидания кнопки.

Если пользователем произведено нажатие кнопки, то объект поль-зовательского интерфейса переходит из состояния «Ожидание нажатия кнопки» в суперсостояние «Активное состояние», в котором сменяет состояния формирования запроса, предобработки данных, считывания данных, подтверждения ответа сервера и отображения информации пользователю.

Если в каком-то из этих состояний возникает исключение, объект переходит из суперсостояния «Активное состояние» в состояние «Обработка ошибки» и затем в конечную точку.

Слайд 10.

Приведены используемое GPU, язык программирования, а также этапы реализации.

Слайд 11. Приведена диаграмма деятельности алгоритма предобработки данных. Имеем исходный набор данных, далее проводим нормализацию и очистку данных.

Если имеем текст обращения, то заменяем именованные сущности тэнами, для исполнителей именованные сущности удаляем. Получаем набор данных для нейросетевых алгоритмов, из которого можем получить набор данных для статистических алгоритмов, где применяется модель мешка слов.

Слайд 12.

Результаты обучения статистических и нейросетевых моделей.

Здесь введен коэффициент MCC, это коэффициент корреляции Мютьюза – мера, достаточно устойчивая к дисбалансу классов, хотя и точность, полнота и Ф-мера я рассчитываю взвешенные, т.е. здесь дисбаланс классов учтен.

По итогам лучшие результаты показали логистическая регрессия и нейросеть на основе механизма внимания, BERT.

Слайд 13. Алгоритм построения модели для определения подлкатегории