**Сопроводительная записка.**

В ТЗ была поставлена задача по отладке промта OpenAI для генерации задач в CRM.

При реализации этой задачи столкнулся с тем что запросы к OpenAI API отклоняются, т.к. регион не поддерживается. Использование vpn посчитал в данном тесте необоснованным и использовал локальные языковые модели (**sberbank-ai/rugpt3large\_based\_on\_gpt2**  и модели от Hugging Face) .

Первоначально код загружает данные из CSV-файла, который находится по ссылке <https://docs.google.com/spreadsheets/d/1kxoGhGT-e_jPaFJ4Po7giixzFzogLowDBDi_C2pbvME/edit?usp=sharing>.

Затем выполняется очистка текста, с помощью функции clean\_text. В этой функции используются регулярные выражения для удаления ненужных элементов текста, таких как временные метки, веб-ссылки и специальные символы. Это помогает создать более структурированный и однозначный текст, который может служить основой для последующего анализа или генерирования задач.

Далее пробовал делать различную токенизацию по cleaned\_dialog\_text для использования в модели, результатов не дало(из кода удалил).

Для генерации задач на основе очищенных диалогов использовал модели для обработки естественного языка sberbank-ai/rugpt3large\_based\_on\_gpt2 и от Hugging Face EleutherAI/gpt-j-6B и EleutherAI/gpt-neox-20b. Лучший результат получил при sberbank-ai/rugpt3large\_based\_on\_gpt2.

После генерации задач, код также включает функцию remove\_duplicate, которая предназначена для удаления дубликатов среди сгенерированных задач. Это решает проблемы с повторяющимися или схожими задачами, что в свою очередь способствует более четкому и организованному представлению информации.

Без remove\_duplicate не получилось (генерация только одной задачи), наверняка другое решение есть, я пока не нашел.

Вывод результатов в конце кода показывает, как были сформированы сгенерированные задачи для каждого диалога. Тут мы можем видеть соответствие задач входящим диалогам и позволяет оценить результат.

Для того чтобы понять, насколько хорошо отрабатывает промт, необходимо установить четкие критерии оценки сгенерированных задач.

В OpenAI API как я понимаю есть функция create\_report, которая составляет .json-отчет о качестве модели. Как можно оценить качество в данной реализации? Возможно надо составить пары(после сбора данных задач), ожидаемая генерация задачи и полученная генерация задачи, далее конвертируем задачи в множества слов для их для дальнейшего анализа.

Потом рассчитаем precision и recall для каждой пары. После расчета точности и полноты каждой пары можно усреднить результаты, чтобы получить общую оценку качества работы модели.

Задание 2.

Возможно такое направление решения:

Если в диалоге упоминается встреча с клиентом (например, "Не забудьте, у нас встреча тогда-то") или если клиент не отвечает на сообщение менеджера в течение заданного времени (тут как раз можно попробовать сделать токенизацию диалогов и смотреть кто последний ответил, но как-то надо будет понять ждем ли мы ответа клиента), то клиент «помечается» (разными метками по соответствующим событиям) и далее может быть реализовано через API(смотрим метки клиентов), который будет принимать запросы на выполнение конкретных действий, например, отправку сообщений «отмеченным» клиентам (email, SMS или сообщения в мессенджерах) или создание записей о напоминаниях.