

Пояснительная записка по проекту подготовки данных для обучения и оценки диалоговых эмбеддеров

Введение

Проект направлен на создание платформы для обработки диалоговых данных, с целью обучения диалоговых эмбеддеров, генерации эмбеддингов и их последующего использования в диалоговых системах. Эмбеддинги представляют собой компактные и информативные представления диалогов, которые обеспечивают:

- Распознавание контекстов и намерений пользователей.
- Генерацию осмысленных ответов.
- Адаптацию к различным языкам и задачам.

Проект включает три основные задачи:

1. **Формирования диалогового корпуса для предобучения моделей:** формирование корпуса диалогов.
2. **Подготовка данных для контрастивного обучения:** сбор и подготовка пар реплик диалогов для подходов к обучению диалоговых эмбеддеров, основанных на контрастивном обучении.
3. **Подготовка данных для оценки эмбеддингов:** сбор и обработка данных для downstream-задач для оценки эмбеддингов.

Цели проекта

1. Обеспечить высокое качество входных диалоговых данных.
 2. Создать удобные инструменты для анализа данных.
 3. Автоматизировать процессы обработки данных и формирования диалогового корпуса.
 4. Предоставить возможность оценки качества эмбеддингов через downstream-задачи.
-

Подготовка данных

Процесс подготовки данных разделён на две основные части:

1. Предобучение (Pre-train)

- Обработка различных диалоговых датасетов, таких как MultiWOZ, CamRest676, Schema и другие.
- Формирование единого корпуса данных с диалогами и последовательностями реплик из диалогов для обучения моделей.
- Генерация пар реплик для контрастивного обучения.

2. Downstream-задачи

- Обработка популярных датасетов, таких как Banking77, CLINC150, MASSIVE и других.
 - Объединение данных в единый формат с метками, языками, и разделением на тренировочные, тестовые и валидационные выборки.
 - Формирование диалогового корпуса для задачи классификации намерений.
-

Структура репозитория

Репозиторий структурирован следующим образом:

1. **data_for_pretrain** — подготовка данных для предобучения:
 - Обработка данных из диалоговых датасетов.
 - Генерация пар реплик для обучения моделей.
 - Скрипты для аналитики данных.
2. **data_for_downstream** — подготовка данных для downstream-задач:
 - Скрипты обработки датасетов.
 - Объединение данных в общий формат.
 - Анализ данных в ноутбуке с использованием pandas и визуализаций.
3. **Скрипты автоматизации:**
 - **create_db.py** — создание базы данных для хранения обработанных данных.

- `run_pipeline.py` — запуск полного пайплайна обработки данных.
 - `run_dashboard.py` — запуск дашборда для визуализации метрик.
-

Метрики качества данных

Для обеспечения качества данных используются три ключевые таблицы:

1. **Полные диалоги для предобучения:**
 - Число записей, длина диалогов и реплик.
 - Распределение данных по split и dataset.
 2. **Пары реплик для контрастивного обучения:**
 - Количество пар, длина реплик.
 - Распределение текстов по длине.
 3. **Данные для downstream-задач:**
 - Количество записей, языков, меток.
 - Анализ распределения данных по сплитам.
-

Автоматизация и дашборд

В рамках проекта создан автоматический пайплайн, включающий сбор данных, предобработку и формирование базы данных. Дополнительно разработан интерактивный дашборд на базе Streamlit для визуализации метрик.

Использование репозитория

1. Предобучение данных

Для подготовки данных для предобучения выполните следующие команды:

```
python data_for_pretrain/pretrain_collecting.py
```

```
python data_for_pretrain/pretrain_preprocessing.py
```

2. Подготовка данных для downstream-задач

Для обработки данных и объединения их в единый формат выполните следующую команду:

```
python data_for_downstream/downstream_collecting.py
```

3. Создание базы данных

Для создания базы данных выполните команду:

```
python create_db.py
```

4. Запуск пайплайна

Для автоматического запуска всего процесса обработки данных, включая, создание базы данных и обработку данных для предобучения модели downstream-задач, выполните команду:

```
python run_pipeline.py
```

5. Запуск дашборда

Для визуализации данных и метрик в интерактивном дашборде выполните команду:

```
streamlit run run_dashboard.py
```

Дашборд автоматически использует ранее созданную базу данных. Если база данных отсутствует, будет запущен процесс ее формирования.