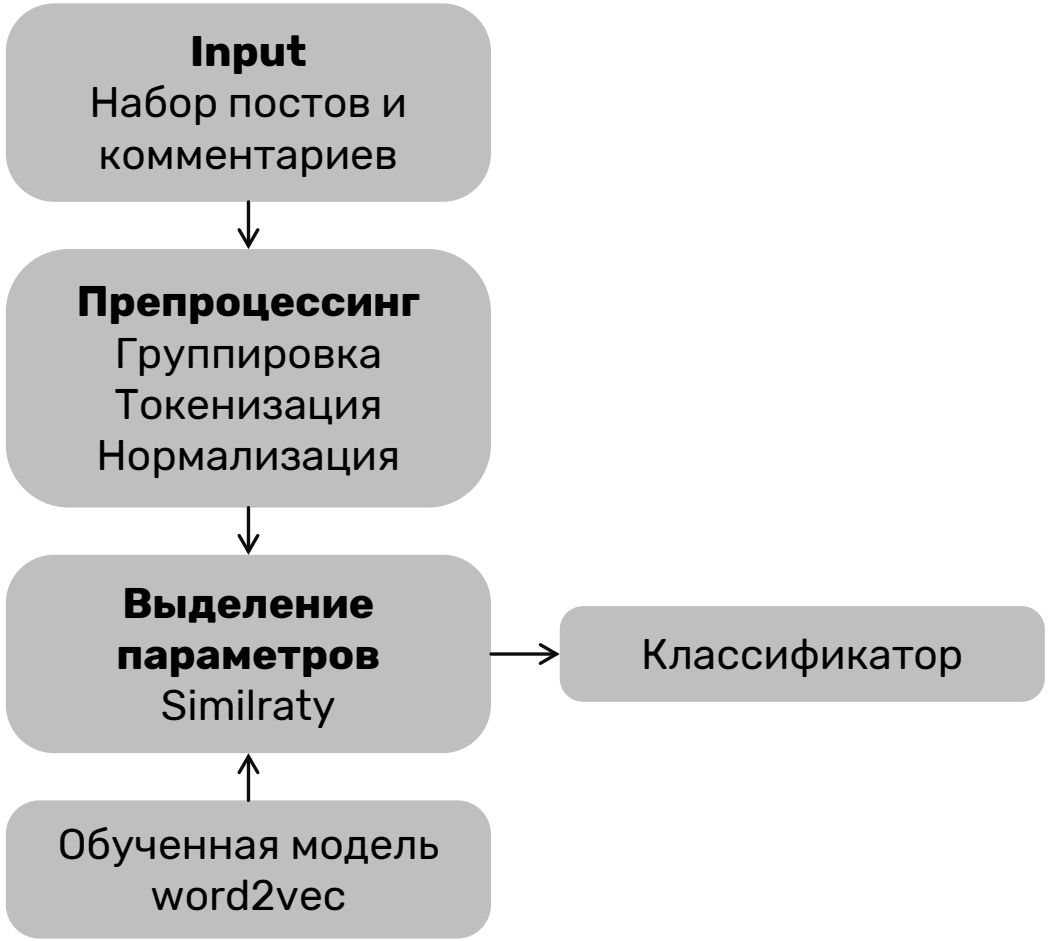
Changellenge

Чемпионат Changellenge >> Cup IT 2023

Команда «Jagermeister»

Трек «Data science»

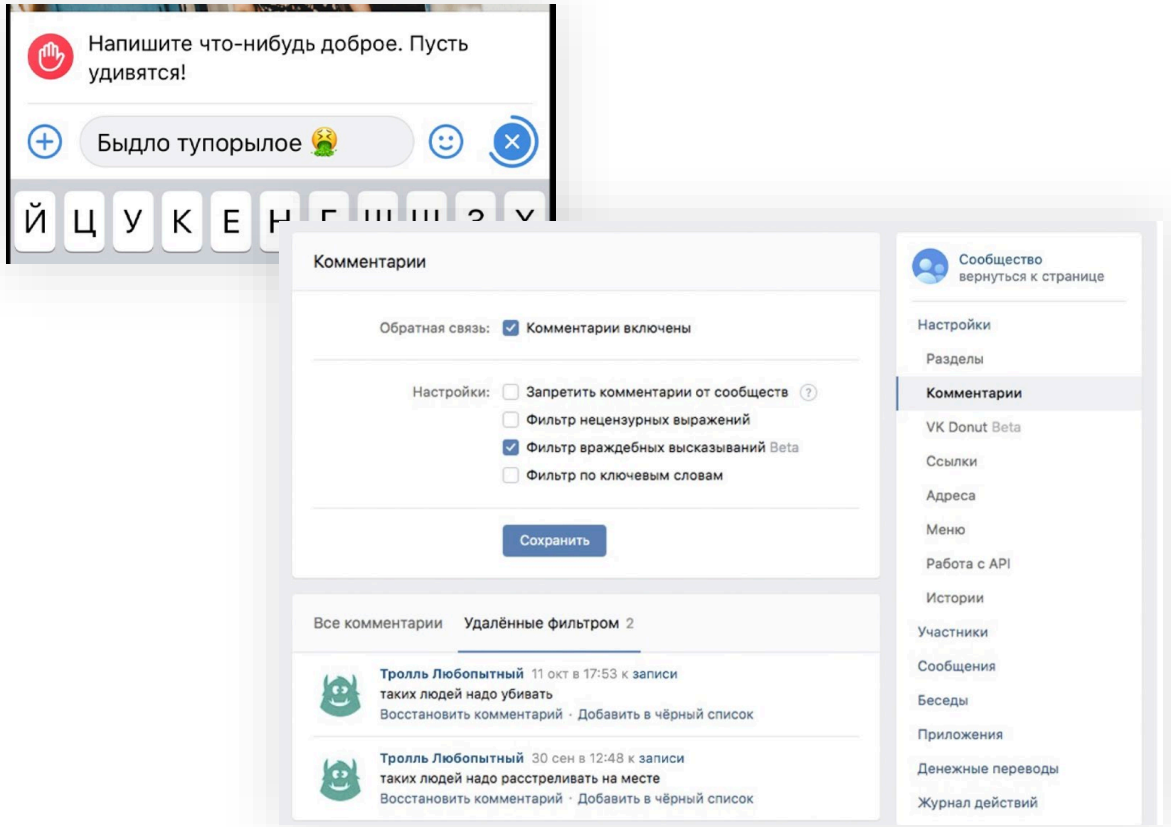
Была получена модель ранжирования комментариев к постам. Архитектура модели выглядит следующим образом:



Результат на валидационных данных: 0.45

Время обучения модели: 8 часов

Возможное практическое применение:



https://github.com/EgorLiutov22/ds_cup/tree/main

Подготовка данных

https://github.com/EgorLiutov22/ds_cup/blob/main/eda_preprocessing.ipynb

Понимание бизнес-целей (Business Understanding)

Для того, чтобы обучить модель ранжировать комментарии к постам, необходимо иметь размеченный набор данных, состоящий из постов и комментариев с уникальными, в рамках одного поста, оценками.

Более того, набор данных должен соответствовать требованию по качеству данных на отсутствие пропусков и полного дублирования строковых значений (текст + комментарий).

Начальное изучение данных (EDA)

Тренировочный датасет содержит в себе 88107 строк текста постов, в каждом из которых содержится 5 комментариев с оценками от 0 до 4, где 0 — оценка самого популярного комментария.

В процессе EDA было получено, что:

- NaN-значения отсутствуют
- 399 дубликатов строк текста с разными комментариями (в количестве от 1 до 6)

Препроцессинг (Preprocessing)

Для работы с данными, с каждым из датафреймов был проведен препроцессинг, который состоял в:

- группировке построчно постов к каждому комментарию
- нормализации данных с отделением текста комментария от его оценки
- конвертировании оценки в интервал [0.2, 0.8], где 0.8 — наиболее популярный комментарий
- замене ссылок на единых для всех ссылок токен «HTTPURL»

Для вычисления «sentence embeddings», который представляет из себя одномерный вектор размерностью 768, была выбрана архитектура «sentence transformers»

При помощи данных вектора для текста поста и соответствующего ему комментария, было рассчитано семантическое сходство при помощи косинусного расстояния. После получения сходства для всех 5 комментариев, они были отсортированы в порядке возрастания.

Мы передаем входное предложение или текст в сеть преобразователя, такую как BERT. BERT создает контекстуальные вложения слов для всех токенов ввода в нашем тексте. Поскольку нам нужно выходное представление фиксированного размера (вектор u), нам нужен объединяющий слой.

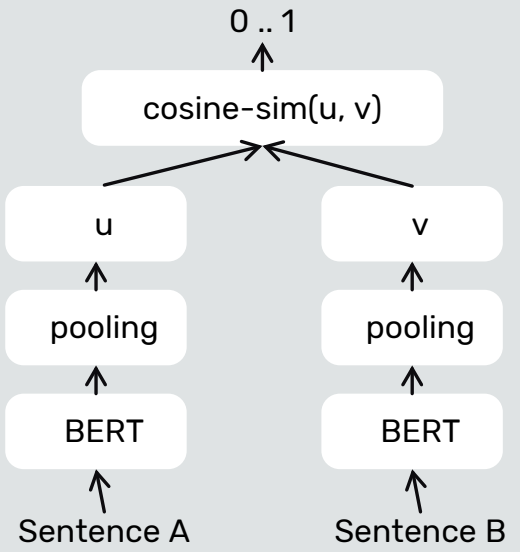
Доступны различные варианты объединения, самый простой из них — объединение среднего значения: мы просто усредняем все контекстуализированные вложения слов, которые нам дает BERT. Это дает нам фиксированный 768-мерный выходной вектор, не зависящий от длины нашего входного текста.

RTX 3090	**~8 ч**	**1 000**	**50**
Видеокарта для расчетов	Заняло обучение	Шагов за эпоху	Кол-во эпох

Функция потерь играет важную роль при тонкой настройке модели. Она определяет, насколько хорошо наша модель внедрения будет работать для конкретной последующей задачи.
В качестве функции потерь была использована CosineSimilarity.

В качестве оценщика (evaluator) применялся EmbeddingSimilarityEvaluator , с количеством шагов 2500 и размером батча 32.

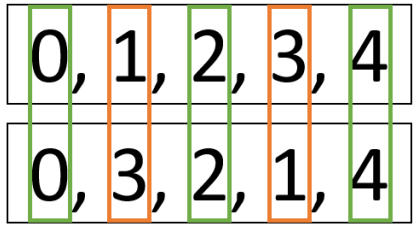
Модели сохранялись каждую эпоху.



После валидации модели была получена точность в 45%.

Это обуславливается тем, что для вычисления точности был использован алгоритм количества совпадения индексов (рангов) между предсказанными и фактическими значениями.

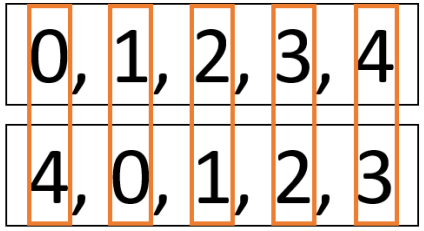
Для улучшения точности предсказания можно предложить улучшение: использовать коэффициент вхождения предсказанной последовательности индексов в фактическую последовательность (т.к. фактические индексы упорядочены в порядке возрастания, то использовать коэффициент упорядоченности)



- Фактическая последовательность
- Предсказанная последовательность

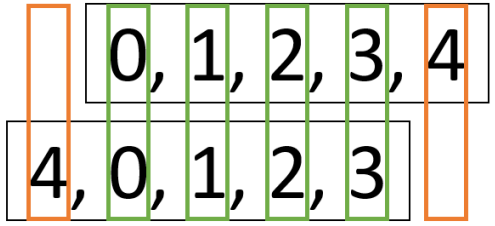
0,6
Точность модели

Зачастую возникают такие случаи из-за 1 ошибки модели, когда он дает слишком высокую/низкую косинусную схожесть



- Фактическая последовательность
- Предсказанная последовательность

0
Точность модели



- Фактическая последовательность
- Предсказанная последовательность

0,8
Точность модели (вместо 0)
Улучшение: коэффициент последовательности

Были сделаны следующие выводы

1. Была найдена корреляция ранга с длиной текста комментария — чем больше длина, тем выше ранг.

2. При помощи модели трансформер-классификатор получена «эмоциональность» текста, которую можно разделить на 7 классов :

- anger (злость);
- disgust (отвращение);
- fear (страх);
- joy (удовольствие);
- neutral (нейтральность);
- sadness (грусть);
- surprise (удивление).

https://github.com/EgorLiutov22/ds_cup/blob/main/train_st.ipynb

3. Результаты оценки комментариев к постам показали, что комментарии с низким уровнем выраженности отвращения и грусти имеют более высокие оценки и, вероятно, будут более популярны среди пользователей.

По полученным выводам можно произвести потенциальные улучшения текущего пайплайна, если в качестве дополнительных параметров модели будут учтены:

- относительная длина токенов в комментарии;
- косинусное сходство между комментарием и текстом поста;
- эмоциальность;
- субъективность.

Кроме того, комментарии, содержащие больше ключевых слов, относящихся к теме поста, также получают более высокие оценки и считаются более популярными.

Однако эти комментарии также должны соблюдать и первое условие. Эти результаты могут быть полезны при формировании контента и улучшении взаимодействия пользователей с платформой.

Создание внутренней рейтинговой системы для комментариев: Рейтинговая система может оценивать качество комментариев на основе содержания и эмоциональной окраски.

Пользователи будут стараться писать более релевантные посту и нетоксичные комментарии, чтобы увеличить свой рейтинг и стать лучшими в сообществе.

Если комментарий короткий:

)))

Попробуйте развернуть свою мысль шире, не бойтесь писать больше. Всем очень интересно!

Если комментарий грубый:

Пост – полная фигня

Ваш комментарий, вероятно, содержит слишком много негатива. Добавь положительных моментов и лайков точно будет больше!

Команда «Jagermeister»

Трек «Data science»

https://github.com/EgorLiutov22/ds_cup/tree/main

Аюпов Айдар Фидратович

sssad@zippp.ru

Разработка и выбор архитектуры модели обучения

Нагимов Рустем Шамилевич

r.sh.nagimov@gmail.com

Препроцессинг, разработка и выбор архитектуры модели обучения. Оформление презентации

Иванушкин Иван Александрович

anvi.inlae@gmail.com

Разведочный анализ данных, интерпретация результатов

Лютов Егор Вячеславович

eliutov22@gmail.com

Модерирование репозитория, дополнительные исследования