Суммаризация и характеристики разговоров операторов службы поддержки с клиентами

Давид Гулян, Елизавета Ёжикова, Андрей Овсянников, Осип Поддерегин

Декабрь 2023

Аннотация

В работе представлено решение задачи суммаризации разговоров операторов службы поддержки с клиентами. Исследована эффективность современных трансформерных архитектур для решения задач суммаризации. Обучен ряд моделей классификации, автоматически определяющих по тексту диалога часть основных его характеристик. Рассмотрена эффективность дообучения трансформерных моделей на задаче MLM на текстах диалогов для повышения качества получаемых моделей. `https://github.com/o-podder/Tweetsumm_NLP-course-2023`.

1 Введение

Служба технической поддержки - неотъемлемая часть большинства компаний, оказывающих услуги большому кругу клиентов. Эффективность работы техподдержки напрямую влияет на удовлетворенность клиентов качеством обслуживания и их доверие к компании. Помимо этого, наличие такой службы позволяет более оперативно получать и обрабатывать обратную связь о качестве оказываемых услуг, и, как следствие, улучшать продукты компании. В связи с этим, качественная аналитика разговоров службы техподдержки имеет важное значение для бизнеса.

Тем не менее, ручной анализ текстов разговоров занимает большое количество времени и человеческих ресурсов. Для сокращения требуемого объема работ целесообразно применение методов машинного обучения. В данной работе приводится решения ряда задач, призванных повысить качество аналитики сервисных диалогов:

1. Суммаризация текста диалога. Заключается в составлении "summary" исходного текста - текста меньшего объема по сравнению с исходным, но при этом сохраняющего основную его информацию. Позволяет значительно сократить время аналитика на понимание содержания разговора при ручном просмотре.

2. Классификация текста диалога по основным характеристикам: 1) Наличие негатива в диалоге со стороны клиента; 2) Наличие благодарности в диалоге со стороны клиента; 3) Определение цели обращения клиента - вопрос или жалоба на качество оказываемых услуг. При последующей агрегации данные метки позволяют аналитику делать выводы о качестве работы отдельных агентов службы поддержки или о довольстве отдельным продуктом клиентов.

Существует большое количество работ и исследований, посвященных задачам суммаризации и классификации текстовых данных. В данной работе рассматривается решение приведенных выше задач на текстах диалогов, которые обладают специфичной структурой и лексикой в сравнении с часто используемыми текстами, например, новостей или научных статей. Анализируется влияние способов предобработки диалоговых данных на качество работы моделей. Исследуется эффект от дообучения исходных трансформерных моделей на репликах диалогов на задании Masked Language Modeling (MLM).

1.1 Команда

Данная работа была подготовлена четырьмя участниками:

Елизавета Ёжикова - разметка данных.

Давид Гулян - классификация диалогов.

Осип Поддерегин - суммаризация диалогов.

Андрей Овсянников - дообучение трансформерных моделей, классификация диалогов.

2 Работы по данной тематике

2.1 Классификация

Существует множество методов, позволяющих осуществлять классификацию текстов. Одним из успешных ранних подходов является TF-IDF представление текста [Spärck Jones, 1972], которое затем подается на вход классической модели классификации: например, логистической регрессии или градиентного бустинга.

С появлением новых методов преобразования текстов, позволяющих получать вектор-представление слова в непрерывном пространстве (Word2Vec [Mikolov et al., 2013], FastText [Bojanowski et al., 2017]), появились более продвинутые подходы, позволяющие учитывать порядок слов в документе. К таким относятся сверточные нейронные сети (CNN, [Kim, 2014]), а также различные варианты рекуррентных нейронных сетей (RNN) - например, основанных на GRU [Cho et al., 2014].

Более современными являются трансформерные архитектуры [Vaswani et al., 2017], основанные на механизме "attention" [Bahdanau et al., 2014], которые лучше

учитывают контекст документа. К таким относятся BERT [Devlin et al., 2018], RoBERTa [Liu et al., 2019], XLNet [Yang et al., 2019] и многие другие.

2.2 Суммаризация

Существует два подхода к задачам суммаризации: Экстрактивная суммаризация (extractive summarization) заключается в выделении из текста наиболее значимых его частей. Одним из примеров такого подхода является BERTSum [Liu, 2019], в котором эмбеддинг каждого предложения, получаемый на выходе с BERT, подается на вход multi-layer attention декодера. На выходе модели каждому предложению соответствует число от 0 до 1, характеризующий "вероятность" того, данное предложение входит в summary исходного текста.

Абстрактивная суммаризация (abstractive summarization) заключается в генерации нового текста на основе исходного. Среди значимых работ в этом направлении можно выделить BART [Lewis et al., 2019], при обучении которого, помимо маскирования токенов (как в BERT), применяются и другие способы "зашумления" данных - например, перестановка токенов. Необходимо отметить и другие трансформерные модели, например, T5 [Raffel et al., 2020]. В PEGASUS [Zhang et al., 2020] при обучении, помимо маскирования отдельных токенов, производится маскирование некоторых предложений целиком. BRIO [Liu et al., 2022] И SimCLS [Liu and Liu, 2021] используют Contrastive Learning для модификации функции потерь, используемой при обучении.

3 Описание моделей

3.1 Классификация

В качестве основных моделей были выбраны модели семейства BERT. Данная архитектура основана на механизме "attention который позволяет изучать контекстуальные отношения между словами (или подсловами) в тексте. В рамках предобучения BERT используются 2 задачи:

1) MLM. Перед вводом последовательности слов в BERT 15% слов в каждой последовательности заменяется токеном [MASK]. Затем модель пытается предсказать исходное значение замаскированных слов на основе контекста, предоставляемого другими, не замаскированными словами в последовательности.

2) Next Sentence Prediction (NSP). В рамках процесса обучения BERT модель в качестве входных данных получает пары фраз, на которых она учится предсказывать, является ли вторая фраза в паре следующей после первой в исходном тексте. Во время обучения 50% входных данных представляют собой пары, в которых вторая фраза действительно является следующей фразой в исходном тексте, а в остальных 50% в качестве второй

фразы выбирается случайная фраза из того же текста. Предполагается, что случайная фраза будет не связана по смыслу с первой фразой.

Входные данные перед подачей в модель обрабатываются следующим образом: текст разбивается на WordPiece токены, в соответствие которым назначаются эмбеддинги. В начало первой фразы вставляется токен [CLS]. В конец каждой из фраз вставляется токен [SEP]. К каждому эмбеддингу токена добавляется эмбеддинг (векторное представление) фразы, обозначающий Фразу A или Фразу B. К каждому эмбеддингу токена добавляется позиционный эмбеддинг, чтобы указать его положение в последовательности.

Для дообучения модели на задачу классификации на выходное представление CLS токена добавляется полносвязный слой и softmax, при этом число выходных нейронов равно числу классов (во всех поставленных в данной работе задачах решается задача бинарной классификации).

Одной из распространенных модификаций BERT является модель RoBERTa, использующая ту же архитектуру, но имеющая ряд изменений: расширенный словарь токенов, увеличенный объем данных для предобучения, динамическое маскирование.

В работе проверяется качество моделей bert-base-cased, roberta-base, а также собственной дообученной на текстах разговоров модели (на основе bert-base-uncased) [1]

3.2 Суммаризация

В данной работе рассматриваются абстрактивные методы решения задачи суммаризации. В ходе экспериментов применялись трансформерные модели BART, T5, PEGASUS. Рассмотрим модель BART, т.к. при ее использовании удалось достичь наилучших результатов.

BART представляет собой трансформерную архитектуру с двунаправленным энкодером и однонаправленным декодером (см. рис. 1). В базовой версии модели используется 6 слоев в энкодере и 6 слоев в декодере.

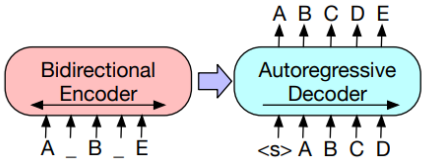

Рис. 1: Схематичное изображение модели BART. Источник: [Lewis et al., 2019]

BART предобучается, восстанавливая предварительно зашумленные данные. Однако, в отличие от BERT, где решается задача предсказания предварительно замаскированного токена, в рассматриваемой модели использу-

[1] Приведены названия моделей на `https://huggingface.co`

ется больше способов "зашумления" данных. К ним, помимо замены токена на [MASK], относятся удаление токена, "вращение" документа относительного случайно выбранного токена, замена целого интервала текста одним [MASK] токеном, перестановка предложений. В общем случае, структура модели позволяет обучаться на восстановлении данных, зашумленных произвольным способом.

Так как BART содержит декодер, его можно напрямую дообучать на задачи генерации текста, в т.ч. на суммаризацию.

4 Датасет

В работе используется датасет TweetSumm [Feigenblat et al., 2021], который включает в себя 1100 диалогов агентов службы поддержки с клиентами в Twitter. Диалоги были составлены из датасета твитов службы поддержки [2], который содержит около 2.7 млн твитов. Данные имеют сплит 80/10/10: в обучающей выборке находится 880 диалогов, в валидационной и тестовой - 110. В табл. 1 представлена информация о средней длине диалога из датасета.

	Полный текст	Текст клиента	Текст оператора
Число реплик	10.17(±2.31)	5.48(±1.84)	4.69(±1.39)
Число предложений	22(±6.56)	10.23(±4.83)	11.75(±4.44)
Число токенов	245.01(±79.16)	125.61(±63.94)	119.40(±46.73)

Таблица 1: Средняя длина диалогов

Каждому диалогу соответствует по 3 абстрактивных и 3 экстрактивных summary, составленных экспертами. Поскольку в данной работе рассматриваются методы абстрактивной суммаризации, экстрактивные summary в дальнейшем не используются. В табл. 2 представлена информация о средней длине abstractive summary.

	Всего	Клиент	Оператор
Abstractive summary	36.41(±12.97)	16.89(±7.23)	19.52(±8.27)

Таблица 2: Средняя длина составленных экспертами summary (в токенах)

Для возможности решения поставленных выше задач классификации была проведена ручная разметка диалогов. Метки проставлялись следующим образом:

[2] https://www.kaggle.com/datasets/thoughtvector/customer-support-on-twitter

1. Негатив со стороны клиента. За негативные реплики принимаются оскорбления, упреки, обвинения в сторону компании. Объяснение самой проблемы клиента и расстроенные чувства, не относящиеся к действиям компании, негативом не считаются.

   Пример негативной реплики: "@116245 how do I make a complaint? You lot are sh*t"

   Пример реплики, которая не считается негативом: "@British_Airways I've got a lost luggage ref number. Got told today that a similar bag got left so that's who probably has mine. I've got the majority of my clothes in it and want them back asap."

2. Благодарность со стороны клиента. Реплика размечается как благодарность, если клиент указывает причину, по которой он благодарен оператору (клиент отмечает хорошо выполненную работу, быстрый ответ, терпение и т.д.). Также за благодарность принимается ярко выраженный позитив ("Thank you very much"). Фразы по типу "Thank you" без пояснений принимаются как вежливость и благодарностями не считаются.

   Пример благодарности: "@AlaskaAir Ah yes that was the issue! Thanks so much for the help, I'm booked!".

   Пример реплики, не размеченной, как благодарность: "@Safaricom_Care Please see DM. Thanks!".

3. Причина обращения (вопрос/жалоба). Причиной обращения считается "вопрос если клиент пишет в службу поддержки за справочной информацией или консультацией. Если клиент задает вопрос, в котором прослеживается недовольство действиями компании или качеством оказываемых услуг, такой диалог размечается как "жалоба".

   Пример жалобы: "@116035 if an ATM doesn't work because of suspicious activity out of your normal withdrawl zone why does it take $ out ur act???"

Также при разметке благодарности и негатива были выделены конкретные реплики, в которых они выражаются. Это позволяет исследовать возможность обучения модели по текстам отдельных реплик, а не целых диалогов.

По итогам разметки получены следующие распределения классов: 35% диалогов содержат негатив, 10% содержат благодарность, 78% является жалобами.

Помимо датасета TweetSumm, для дообучения трансформерных моделей на задаче MLM в данной работе используется исходный датасет твитов службы поддержки, указанный выше. Для дообучения использовались все твиты, за исключением тех, что были выбраны как реплики диалогов при формировании датасета TweetSumm.

5 Эксперименты

5.1 Метрики

5.1.1 Класификация

Из-за наличия дисбаланса в классах для оценки качества моделей классификации используется метрика F1-score.

$$F1 = \frac{2 \cdot precision \cdot recall}{precision + recall};$$

Здесь:

$$precision = \frac{TP}{TP + FP}$$

$$recall = \frac{TP}{TP + FN},$$

где TP - количество верных предсказаний объектов положительного класса, FP - количество ложных предсказаний объектов положительного класса, FN - количество ложных предсказаний объектов нулевого класса.

Для моделей благодарностей и негатива в качестве целевой метрики был выбран F1-score положительного класса, для модели причины обращения - macro average F1-score.

5.1.2 Суммаризация

В качестве целевых метрик был выбран набор F1-метрик ROUGE: ROUGE-1, ROUGE-2, ROUGE-L.

ROUGE-1 метрики для исходного summary (reference summary) и summary-кандидата (candidate summary) вычисляются следующим образом:

$$ROUGE\text{-}1_{precision} = \frac{|unigram\ cand. \cap unigram\ ref.|}{|unigram\ cand.|}$$

$$ROUGE\text{-}1_{recall} = \frac{|unigram\ cand. \cap unigram\ ref.|}{|unigram\ ref.|}$$

$$ROUGE\text{-}1_{F1} = \frac{2 \cdot precision \cdot recall}{precision + recall},$$

где $unigram\ cand.$ - множество слов в candidate summary, $unigram\ ref.$ - множество слов в reference summary.

Набор ROUGE-2 метрик высчитывается аналогично с тем отличием, что вместо множеств слов используются множества биграмм слов.

ROUGE-L метрики вычисляются с использованием величины наиболее длинной общей последовательности (LCS) для reference summary и candidate summary:

$$ROUGE\text{-}1_{precision} = \frac{LCS(cand., ref.)}{|unigram\ cand.|}$$

$$ROUGE\text{-}1_{recall} = \frac{LCS(cand., ref.)}{|unigram\ ref.|}$$

$$ROUGE\text{-}1_{F1} = \frac{2 \cdot precision \cdot recall}{precision + recall},$$

Важно отметить, что в имеющемся датасете одному тексту диалога соответствует несколько reference summary. В этом случае, в соответствии с [Lin, 2004], ROUGE метрика для candidate summary рассчитывается как максимум среди ROUGE метрик, попарно рассчитанных для каждого reference summary, соответствующего тексту диалога.

5.2 Постановка экспериментов

5.2.1 Классификация

Сплит данных, представленный в исходном датасете мало пригоден для обучения и оценки, поскольку: 1) Распределение меток в тестовой выборке не совпадает с распределением меток во всей выборке; 2) Тестовый набор данных имеет всего 110 объектов, что с учетом явного дисбаланса классов для некоторых меток (например, в тестовом наборе содержится всего 11 диалогов, размеченных, как имеющие благодарность), приводит к ненадежным оценкам качества обученных моделей. По этой причине было принято решение при обучении всех моделей классификации использовать стратифицированную кросс-валидацию на 5 фолдов следующим образом: 1) на каждой итерации новый фолд становится тестовым набором данных; 2) среди оставшихся четырех фолдов делается случайная подвыборка размером с тестовый датасет, эта подвыборка становится валидационным набором данных. 3) Оставшиеся данные включаются в обучающую выборку. Итоговая оценка рассчитывается как среднее арифметическое f1-метрик на тестовом наборе данных на каждой итерации.

Проверялось качество моделей bert-base-cased, bert-base-uncased, roberta-base, а также собственной, дообученной на текстах твитов модели (на основе bert-base-uncased) [3]. Дообучение модели проводилось с $lr = 1e^{-5}$ в течение 2 эпох.

Было рассмотрено 3 метода предобработки данных:

1. Очистка текстов от ссылок, ников, html-тегов;

2. Помимо очистки, из диалога удалялся весь текст оператора, реплики клиента конкатенированы в один текст;

[3] `https://huggingface.co/SoooSlooow/tweet-bert-uncased`

3. Для моделей благодарностей и негатива: помимо очистки, из диалога удалялся весь текст оператора, был проведено разделение текста клиента на отдельные реплики. Каждой реплике была проставлена метка класса (напомним, соответствующая разметка по репликам была проведена в п.4). Обучение модели производилось по репликам. Метка диалогу в тестовом наборе данных ставится следующим образом: 1) делается предсказание для каждой реплики клиента диалога; 2) если хотя бы одной реплике был присвоен положительный класс, диалогу присваивается положительный класс, иначе, диалогу присваивается отрицательный класс.

5.2.2 Суммаризация

При обучении моделей суммаризации использовался сплит данных, указанных в п.4. Предварительно текст был очищен от ссылок и html-тегов.

В ходе работы проверялись следующие модели суммаризации: t5-base, bart-base, pegasus-cnn_dailymail. В качестве алгоритма оптимизации использовался Adam с $lr = 5e\text{-}6$. Также использовался weight decay, равным 0.01, остальные параметры обучения и оптимизатора были оставлены по умолчанию.

Помимо этого, исследуется качество дообученной модели bart-base на датасете твитов техподдержки, описанном в п.4, на задаче MLM (с вероятностью маскирования токена, равной 15%).[4] Дообучение модели проводилось с $lr = 1e\text{-}5$ в течение 2 эпох (обучение заняло около 24ч).

5.3 Базовые модели

5.3.1 Классификация

В качестве базовой модели при решении задач классификации была выбрана реализация градиентного бустинга CatBoost. Предварительная обработка текстов включала в себя очистку текстов от пунктуации, ссылок, никнеймов и лемматизацию. После очистки текстов проводилось TF-IDF преобразование (выполнялось при помощи реализации в sklearn TfIdfVectorizer() с параметром ngram_range = (1, 2)).

5.3.2 Суммаризация

В качестве бэйзлайна суммаризации были выбраны 2 эвристики:

1) Из текста диалога случайным образом были выбраны 2 предложения (не реплики) клиента и 2 предложения оператора. Эти 4 предложения, отсортированные в порядке их следования диалога, составляют итоговое summary.

2) Из текста диалога выбираются первые 2 предложения клиента и первые 2 предложения оператора. Эти 4 предложения отсортированные в порядке их следования диалога, составляют итоговое summary.

[4] `https://huggingface.co/SoooSlooow/TweetBART2`

6 Результаты

6.1 Классификация

В таблице 3 представлены F1-оценки работы моделей классификации.

	Негатив	Благодарности	Причина обращения
TF-IDF + CatBoost	0.65	0.27	0.59
BERT	0.73	0.58	0.70
RoBERTa	0.74	0.57	0.69
TweetBERT	0.69	0.48	**0.80**
BERT (текст клиента)	**0.76**	**0.62**	0.71
TweetBERT(текст клиента)	0.73	0.58	0.78
BERT (реплики)	0.72	0.60	-
TweetBERT (реплики)	0.72	0.59	-

Таблица 3: Результаты работы моделей классификации. TweetBERT - собственная дообученная модель BERT.

Из таблицы видно, что использование только текста клиента дает прирост качества для моделей негатива и благодарностей. В то же время, TweetBERT показывает себя несколько хуже в задачах определения негатива и благодарностей, однако работает значительно лучше в задаче определения причины обращения.

6.2 Суммаризация

В таблице 4 представлены оценки работы перечисленных в прошлом разделе моделей.

	ROUGE-1	ROUGE-2	ROUGE-L
Random	0.319	0.109	0.255
Lead	0.415	0.192	0.361
T5-base	0.556	0.314	0.501
BART-base	**0.585**	**0.345**	**0.541**
PEGASUS-cnn_dailymail	0.581	0.330	0.528
TweetBART2	0.573	0.332	0.530

Таблица 4: Результаты работы моделей суммаризации. Здесь Random - бейзлайн с выбором случайных предложений, Lead - бейзлайн с выбором первых предложений. TweetBART2 - собственная дообученная модель BART.

Как видно из таблицы, базовая версия модели BART показала лучшие результаты по всем трем метрикам. Совсем незначительно от нее отстает

модель PEGASUS. Дообучение модели на текстах твитов не привело к увеличению метрик, а напротив, привело к небольшому ухудшению качества.

7 Заключение

Таким образом, в ходе работы была собрана разметка по основным характеристикам диалога. Была рассмотрена эффективность дообучения BART-модели на текстах, близких к текстам датасета, проведено сравнение существующих моделей суммаризации. Было приведено решение поставленных задач классификации с использованием трансформерных моделей, показана эффективность определенных методов преобразований текстов диалогов для решения подобных задач.

Список литературы

[Bahdanau et al., 2014] Bahdanau, D., Cho, K., and Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

[Bojanowski et al., 2017] Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. (2017). Enriching word vectors with subword information. *Transactions of the association for computational linguistics*, 5:135–146.

[Cho et al., 2014] Cho, K., Van Merriënboer, B., Bahdanau, D., and Bengio, Y. (2014). On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*.

[Devlin et al., 2018] Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

[Feigenblat et al., 2021] Feigenblat, G., Gunasekara, C., Sznajder, B., Joshi, S., Konopnicki, D., and Aharonov, R. (2021). Tweetsumm– a dialog summarization dataset for customer service. *arXiv preprint arXiv:2111.11894*.

[Kim, 2014] Kim, Y. (2014). Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*.

[Lewis et al., 2019] Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., and Zettlemoyer, L. (2019). Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

[Lin, 2004] Lin, C.-Y. (2004). Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

[Liu, 2019] Liu, Y. (2019). Fine-tune bert for extractive summarization. *arXiv preprint arXiv:1903.10318*.

[Liu and Liu, 2021] Liu, Y. and Liu, P. (2021). Simcls: A simple framework for contrastive learning of abstractive summarization. *arXiv preprint arXiv:2106.01890*.

[Liu et al., 2022] Liu, Y., Liu, P., Radev, D., and Neubig, G. (2022). Brio: Bringing order to abstractive summarization. *arXiv preprint arXiv:2203.16804*.

[Liu et al., 2019] Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

[Mikolov et al., 2013] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26.

[Raffel et al., 2020] Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.

[Spärck Jones, 1972] Spärck Jones, K. (1972). A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*, 28:11–21.

[Vaswani et al., 2017] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.

[Yang et al., 2019] Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R. R., and Le, Q. V. (2019). Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32.

[Zhang et al., 2020] Zhang, J., Zhao, Y., Saleh, M., and Liu, P. (2020). Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In *International Conference on Machine Learning*, pages 11328–11339. PMLR.