Классификация намерений в email-обращениях на основе LLM

М. С. Замула, И. А. Лоскутов

Аннотация – При обработке входящих ответов на корпоративные email-рассылки возникает задача определения заинтересованности клиента в продолжении диалога. Мы сравниваем три подхода на потоке действующей платформы (более 1300 аккаунтов, 17 языков): классификатор по ключевым словам, большую языковую модель с контекстными примерами и структурированным выводом и дообученный мультиязычный энкодер XLM-RoBERTa. Эталонной разметкой служит действие оператора; тестовая выборка – 300 подлинных клиентских ответов, отфильтрованных от служебного транзита рассылки. Цитата исходного письма присутствует в 96 % ответов и лишает наивный keyword-метод различающей способности: на полном теле письма он статистически неотличим от тривиальной стратегии «отвечать всем» (F1 0,676 против 0,667). LLM-конвейер даёт F1 = 0,73; энкодер, дообученный на 11 тыс. операторских решений, достигает F1 = 0,78, значимо превосходя LLM (тест Макнемара, p < 0,01) при почти вдвое меньшем числе ложных срабатываний и нулевой стоимости вызова. При этом и LLM, и энкодер нечувствительны к цитированному тексту (p ≥ 0,75): удаление цитат остаётся полезным лишь для лексического резерва и сокращения промпта. Конвейер работает в производственной среде; обсуждаются валидность операторской разметки и экономика подходов.

Ключевые слова – классификация намерений, большие языковые модели, обработка электронной почты, обучение на примерах, дообучение энкодера, XLM-RoBERTa, предобработка текста, цитированный текст, структурированный вывод

I. Введение

По оценке Radicati Group, ежедневно в мире отправляется свыше 300 миллиардов электронных писем. Значительную долю генерируют отделы продаж и клиентской поддержки – они всё чаще используют специализированные платформы для массовой рассылки. Такие платформы обслуживают сотни и тысячи почтовых аккаунтов – от формирования персонализированных писем до отслеживания доставки. При потоке в сотни входящих ответов в день оператору приходится быстро решать, стоит ли продолжать диалог с конкретным клиентом. На каждый тред уходит в среднем две минуты – при 500 тредах это полный рабочий день одного человека. Ручная классификация при таком объёме быстро становится узким местом.

Простейшее решение – поиск ключевых слов в теле ответа [3]. Слова «интересует», «цена», «доступно» помечают ответ как перспективный. Казалось бы, всё просто. Однако на практике этот подход наталкивается на неприятную особенность, которую мы в просмотренных источниках не встретили: email-клиент при формировании ответа автоматически вставляет в тело письма цитату исходного сообщения. Цитата отделяется маркером «>», строками «wrote:», «schrieb:», «a écrit:» или горизонтальной чертой. В итоге ключевые слова из рассылки – того самого письма, что отправила компания – попадают в тело ответа и сбивают классификатор. Покажем на примере:

Исходное: «Наш продукт доступен в вашем регионе. Хотите узнать подробности?» Ответ клиента: «Нет, спасибо.» Тело email, полученное платформой:

Нет, спасибо.

> Наш продукт доступен в вашем регионе. > Хотите узнать подробности?

Классификатор находит «доступен» в цитате и ошибочно относит ответ к категории «заинтересован». На деле клиент отказал. По нашим данным, 95–96 % подлинных входящих ответов содержат такие цитаты – проблема затрагивает практически весь поток.

Зонирование email (разделение на тело, цитату и подпись) изучалось в [4, 12] и расширено на другие языки в [13], но только как отдельная задача предобработки – авторы не проверяли, помогает ли зонирование последующей классификации намерений. Применение LLM к email рассмотрено в обзорах [2, 5], к техподдержке – в [16], к классификации текстов – в [10, 14], включая определение намерений (intent detection) [15]. Специфика корпоративной переписки с цитатами в них не затрагивается. Подбор контекстных примеров для LLM [6] к этой задаче тоже не применялся. Открытым остаётся вопрос: насколько цитаты вообще мешают классификации – и мешают ли при использовании языковых моделей? Ответ, как покажет эксперимент, нетривиален.

Обозначим через T набор тредов электронной почты. Каждый тред $t \in T$ содержит тело ответа клиента $B(t)$ и, возможно, цитированный текст $Q(t)$. Задача классификации:

$$f: B(t) \to C \tag{1}$$

где $C = \{Respond, Ignore\}$ – по сути, нужен ответ оператора или нет.

Реальное намерение клиента обозначим $y(t) \in C$, предсказание модели – $\hat{y}(t)$. Качество оценивается метриками:

$$P = \frac{TP}{TP + FP}, \quad R = \frac{TP}{TP + FN}, \quad F_1 = \frac{2PR}{P + R} \tag{2}$$

где TP, FP, FN, TN – элементы матрицы ошибок. F_1 – гармоническое среднее точности и полноты; значение 1,0 соответствует идеальной классификации. Поскольку пропуск заинтересованного клиента обходится дороже ложного срабатывания, приоритет отдаётся полноте R.

М. С. Замула, И. А. Лоскутов: МИРЭА – Российский технологический университет, Москва, Россия (e-mail: matvey.zamula@mail.ru, Faxvex@ya.ru).

Конвейер, описываемый ниже, включает три этапа:

1) удаление цитированного текста из тела ответа набором из 12 языковых паттернов; 2) бинарная классификация намерения с помощью LLM со структурированным выводом; 3) подбор контекстных примеров из базы исторических диалогов по четырём критериям.

Основные результаты:

1) На выборке из 300 подлинных клиентских ответов production-платформы (эталон – действие оператора) сравнение трёх классов методов дало F1 0,68–0,69 для keyword-классификатора, 0,73–0,74 для few-shot LLM и 0,77–0,78 для дообученного мультиязычного энкодера XLM-R; превосходство энкодера над LLM статистически значимо (тест Макнемара, $p < 0,01$). 2) Цитаты присутствуют в 96 % подлинных ответов и лишают наивный keyword-метод различающей способности (на полном теле письма он неотличим от стратегии «отвечать всем»); при этом и LLM, и дообученный энкодер к цитированному тексту нечувствительны ($p \geq 0,75$) – предобработка нужна только лексическому резерву. 3) Накопленная операторская разметка (~11 тыс. решений) позволяет заменить внешний LLM локальной моделью с выигрышем в качестве и нулевой стоимостью вызова; few-shot LLM остаётся решением холодного старта.

Далее: раздел II – обзор литературы, III – платформа и постановка задачи, IV – метод, V – эксперимент, VI – выводы.

II. Обзор литературы

A. Классификация email и определение намерений

Задачи классификации email варьируются от фильтрации спама до маршрутизации запросов в техподдержку. Наиболее полный обзор методов дан у AlShaikh et al. [3]: авторы проследили путь от наивного Байеса и SVM с TF-IDF через ансамбли к трансформерам, которые на стандартных наборах данных для спама превышают 98% точности. При этом при смене домена модели теряют 5–12 п.п. – проблема переноса так и остаётся актуальной. Интересно другое. Melendez et al. [14]: на корпусе фишинговых писем RoBERTa показала 99,43% F1, но SVM и Random Forest отстали всего на 1–2 п.п. при грамотном отборе признаков. Иными словами, для бинарной классификации email превосходство трансформеров далеко не бесспорно.

Определение намерений (intent detection) – задача посложнее. Здесь вместо «спам/не спам» нужна система из нескольких категорий. Jbene et al. [15] сравнили RNN и трансформеры для определения намерений в диалоговых системах; BERT-подобные модели обгоняют BiLSTM на 3–7% по F1 (наборы данных ATIS, SNIPS). Правда, при увеличении обучающей выборки разрыв сокращается. Guo et al. [17] пошли другим путём и обогатили представления BERT на уровне частей слов информацией о морфемах (ESIE-BERT), улучшив совместное предсказание намерения и слотов.

Для русского языка ситуация развивается параллельно. Куратов и Архипов [9] адаптировали BERT (RuBERT), а Карпов и Коновалов [18] показали, что мультизадачное обучение на русских данных позволяет классифицировать намерения без больших размеченных корпусов – для нас это было важно, учитывая мультиязычность платформы. Косяненко и Болбаков [20] применили CodeBERT для оценки качества коротких текстовых сообщений, подтвердив работоспособность таких моделей на русскоязычных данных. Архитектуры LLM-агентов [8] делегируют классификацию оркестратору, но вычислительные затраты пока делают этот путь для нас избыточным.

Отдельно стоит упомянуть сегментацию email на зоны (тело, цитата, подпись). В [4] SVM на структурных признаках хорошо выделяет зоны – графические маркеры и паттерны цитирования (символ «>») оказались сильнейшими предикторами. Repke и Krestel [12] применили GRU-CRF и обогнали SVM-бейзлайн на 4 п.п. F1, особенно на нестандартных форматах цитирования. Jardim et al. [13] расширили подход на мультиязычные корпуса – перенос модели, обученной на английском, на португальский сохраняет до 89% качества. Но (и это важно) все три работы рассматривают зонирование как самоцель, не проверяя, помогает ли оно последующей классификации. Для нашей задачи это означает, что зонирование стоит оценивать не изолированно, а в связке с классификатором – чему и посвящён эксперимент в разделе V.

B. Большие языковые модели в обработке email

Dam et al. [2] рассмотрели архитектуры LLM-чатботов для клиентской поддержки и зафиксировали разрыв между лабораторными метриками и production-показателями – ситуация, хорошо знакомая практикам. Novelo et al. [5] в PRISMA-обзоре 32 статей по персонализированной генерации email обнаружили, что RAG (дополнение промпта результатами поиска по документам) и PEFT (LoRA, QLoRA – методы дообучения с малым числом параметров) адаптируют LLM к стилю пользователя, но классификацию намерений во входящих письмах ни один из авторов не рассматривал – модели сразу генерируют ответ, не определив тип запроса.

Подробнее стоит остановиться на сравнении Chae и Davidson [10]: они протестировали 10 моделей от BERT-base до GPT-4 в четырёх режимах: без примеров (zero-shot), с несколькими примерами (few-shot), с дообучением на данных задачи (fine-tuning) и с дообучением на инструкциях (instruction-tuning). Вывод, который оказался важен для нашей работы: дообученный BERT-base оказался лучше GPT-4 без примеров на задачах с чёткими метками, тогда как LLM с instruction-tuning выигрывают, когда категории размыты или данных мало. Wulf и Meierhofer [16] на данных телеком-оператора показали, что GPT-4 автоматизирует когнитивные задачи техподдержки (маршрутизация, извлечение фактов, генерация ответов), но сценарий с цитатами в теле ответа не изучали.

Однако ни одна из этих работ не рассматривает ситуацию, когда тело ответа содержит цитату исходного письма – а для корпоративной переписки это, как мы уже отмечали, типичный случай.

C. Few-shot обучение и бенчмарки email

Обучение на малом числе примеров и тестовые наборы для email*

Pichugov et al. [6] систематизировали стратегии обучения LLM на минимальных данных. Один из выводов, который перекликается с нашим опытом: случайная выборка контекстных примеров уступает стратифицированной на 8–15% accuracy – выбор примеров критичен. Николаев [7] предложил метод объяснимости BERT для задач классификации текстов – для нас это интересно в контексте отладки: понять, почему модель приняла то или иное решение.

Shay et al. [11] представили тестовый набор EnronSR – 50 email-цепочек из корпуса Enron с экспертными аннотациями. Набор интересен тем, что фиксирует: качество ответа зависит от понимания контекста цепочки, включая цитаты. Но EnronSR оценивает генерацию, а не классификацию.

Подбор контекстных примеров для классификации email-намерений с учётом цитат ранее не применялся. Существующие тестовые наборы (ATIS, SNIPS для определения намерений; Enron, SpamAssassin для email) не разделяют авторский текст и цитаты. Ни в одной из рассмотренных 19 работ не исследовано, как цитированный текст влияет на точность классификации намерений. Мы решили проверить это экспериментально.

III. Архитектура платформы

Платформа обслуживает email-коммуникации отделов продаж. Управление масштабной почтовой инфраструктурой – задача не новая (ещё в [1] описан кластерный почтовый сервис Porcupine, обслуживавший миллионы почтовых ящиков на кластере из обычных ПК), однако классификация намерений во входящих ответах в подобных системах, насколько мы можем судить, не рассматривалась. Наша платформа оперирует пулом из более чем 1300 email-аккаунтов, разделённых более чем на 40 групп по каналам и регионам (17 языков). Каждая группа обслуживает кампании с параметризованными шаблонами: платформа подставляет переменные (имя адресата, название продукта) и рассылает персонализированные предложения. За время работы платформа обработала около 800 000 входящих сообщений (на момент фиксации экспериментальных данных); из потока сформирована курируемая база из 14 029 завершённых диалогов на 8 языках (немецкий, нидерландский, французский, норвежский и др.), служащая источником контекстных примеров (раздел IV), а для оценки выделено около 12 тысяч подлинных клиентских обращений с зафиксированным решением оператора (раздел V).

Входящие ответы поступают в операторский интерфейс. Для каждого ответа нужно решить: требует ли письмо реакции (клиент заинтересован, задаёт вопрос, готов к сделке) или его можно пропустить (отказ, спам, автоответ). На первых этапах мы обходились ручной классификацией, но при 500+ тредах в день это занимало у оператора несколько часов ежедневно – неприемлемо. При этом цена ошибки асимметрична: пропустить заинтересованного клиента гораздо хуже, чем потратить лишние 10 секунд на прочтение нерелевантного письма. Отсюда приоритет полноты (recall) над точностью (precision).

Рис. 1 показывает конвейер обработки. Входящее письмо проходит модуль удаления цитат. Очищенный текст поступает на LLM-классификатор, который возвращает категорию, оценку уверенности и обоснование. Оператор видит результат и при необходимости корректирует его; корректировки накапливаются и используются для подбора контекстных примеров (своего рода обратная связь). По нашим данным, 95–96 % подлинных входящих ответов содержат цитированный текст – маркеры «>», «wrote:», «schrieb:» и аналогичные. Именно этот поток и стал отправной точкой для конвейера, описанного далее.

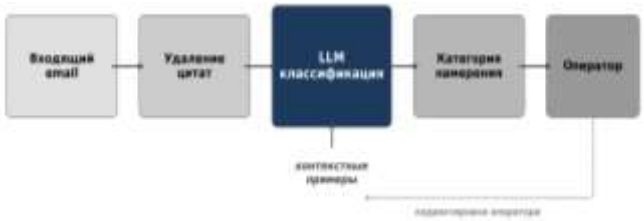

Рис. 1. Конвейер классификации намерений в email-обращениях

IV. Классификация намерений на основе LLM

A. Проблема цитированного текста

Как отмечено в разделе I, почтовые клиенты вставляют в тело ответа цитату исходного сообщения. Обозначим t – тред, $B(t)$ – тело входящего ответа, $Q(t)$ – цитированный фрагмент. Тогда

$$B(t) = B_{client}(t) \cup Q(t) \quad (3)$$

где $B_{client}(t)$ – собственно текст клиента (здесь $\cup$ обозначает конкатенацию текстовых фрагментов). Для keyword-классификатора это проблема. Покажем на двух примерах.

Пример 1 (русский). Исходное: «Наш продукт доступен в вашем регионе. Хотели бы вы узнать подробности?» Ответ: «Нет, спасибо, не нужно.» Тело email:

```
Нет, спасибо, не нужно.

> Наш продукт доступен в вашем регионе.
> Хотели бы вы узнать подробности?
```

Классификатор видит «доступен» из цитаты и относит ответ к «Interested». Реальное намерение – отказ.

Пример 2 (английский). Исходное: «Would you like to order a sample?» Ответ: «No, thank you.» Тело email:

```
No, thank you.

On Mon, Apr 3, 2026 Platform wrote:
> Would you like to order a sample?
```

Слово «order» из цитаты провоцирует ложное срабатывание. Масштаб проблемы велик: как показано

в разделе V, 96 % подлинных ответов клиентов в нашей выборке содержат цитаты.

B. Предобработка: удаление цитат

Для решения проблемы мы реализовали модуль предобработки. Функция предобработки:

$$strip(B) = B[:min_k pos(P_k, B)] \quad (4)$$

где $\{P_1, \dots, P_{12}\}$ – набор регулярных выражений, k – индекс первого совпадения. Основные паттерны:

1) стандартный маркер цитаты – символ «>» в начале строки; 2) англоязычный формат «On ... wrote:» (Outlook, Apple Mail, Thunderbird); 3) немецкоязычный «Am ... schrieb»; 4) франкоязычный «Le ... a écrit»; 5) горизонтальный разделитель – три и более дефисов.

Остальные 7 паттернов покрывают нидерландскую, скандинавские, испанскую и итальянскую локализации, а также заголовки пересылки (Von:, Van:, From:). Идея общая – вырезать всё, что идёт после первого маркера цитаты, какой бы локализацией почтовый клиент ни пользовался. Полный список доступен в исходном коде платформы.

При первом совпадении текст усекается: сохраняется фрагмент до маркера, остальное отбрасывается. Многоуровневое цитирование (маркеры «>>») обрабатывается тем же усечением: вложенные уровни находятся ниже первого маркера и отбрасываются вместе с ним. Подход рассчитан на формат top-posting (ответ над цитатой). Обратный порядок (bottom-posting, когда ответ помещается под цитатой, – стиль, характерный для юридической переписки) и чередование ответа с фрагментами цитаты (interleaved quoting) усечением принципиально не обрабатываются: текст клиента был бы потерян.

Мы количественно оценили распространённость этих форматов в нашем потоке. Автоматическая эвристика выделяла сообщения, в которых после первого маркера цитирования остаётся ≥30 символов текста, не относящегося к служебным строкам (префиксы цитат, заголовки пересылки, подписи); каждый кандидат затем проверялся вручную. Истинный bottom-posting и interleaved-ответ такая эвристика ловит по построению: текст клиента, расположенный ниже маркера, попал бы в остаток. На тестовой выборке (n = 300, из них 289 с цитатами) обнаружено 17 кандидатов, на случайной выборке из корпуса платформы (n = 5000) – 6; ручная проверка не подтвердила ни одного: во всех случаях «остаток» оказался цитированной копией нашего собственного исходящего сообщения под заголовком почтового клиента (без построчных префиксов) либо подписью или контактным блоком клиента. Таким образом, подтверждённых случаев bottom-posting/interleaved – 0 из 390 цитированных сообщений; верхняя граница 95 %-го доверительного интервала («правило трёх») – менее 1 % цитированных сообщений. Объяснение доменное: переписка идёт через почтовые релеи внешних платформ и личные почтовые клиенты в сценарии продаж, где доминирует top-posting; на юридическую и иную формальную корреспонденцию с принятым там bottom-posting наш вывод не распространяется, и переносимость конвейера на такие домены требует отдельной проверки.

Ограничение модуля предобработки, однако, не переносится на итоговый конвейер. Как будет показано в разделе V (табл. I), LLM-классификатор, получающий полное тело письма без усечения, статистически неотличим от конвейера с удалением цитат (F1 0,736 против 0,734, p = 1,0); то же верно и для дообученного энкодера (p = 0,75): обе модели сами отделяют текст клиента от цитированного. Поэтому для доменов, где bottom-posting распространён, достаточно отключить усечение и подавать полное тело – штраф за это в нашем эксперименте не обнаружен; предобработка же сохраняет ценность как защита keyword-резерва и способ сокращения промпта (раздел V.B).

C. LLM-классификация со структурированным выводом

Очищенный текст подаётся на вход LLM (Claude Sonnet 4.5, версия от 29.09.2025). Температуру мы подбирали на валидационном подмножестве из 50 тредов и остановились на 0,3: при нуле модель иногда застревает на неоптимальных токенах (жадный выбор наиболее вероятного варианта), при значениях выше 0,5 появляется нежелательная случайность.

Системный промпт описывает роль классификатора, контекст платформы (B2B-коммуникации, пул аккаунтов, операторский интерфейс), определения допустимых категорий с примерами и инструкцию: классифицировать только текст ответа клиента, игнорируя остатки цитат. Эта инструкция дублирует модуль предобработки – страховка на случай, если часть цитаты проскочит через regex. Проблема нестабильности LLM-ответов, исследованная в [19], дополнительно мотивировала нас использовать структурированный вывод вместо свободной генерации.

Вместо генерации произвольного текста модель вызывает функцию с типизированными параметрами. Или в программном виде:

```
{
  "category":      "enum:      Interested      |
    Not    interested    |    Info    request    |
    Conversion          |          Unclassified",
  "confidence":    "float,    0.0    -    1.0",
  "reasoning":                              "string"
}
```

Пример для ответа «No thanks, not interested»:

```
{
  "category":          "Not          interested",
  "confidence":                              0.95,
  "reasoning":        "Customer        explicitly
    declines              the              offer"
}
```

Важно, что выход всегда содержит ровно одну категорию – никаких сюрпризов с форматом. Поле confidence (уверенность) позволяет маршрутизировать ответы с низкой оценкой (мы выбрали порог 0,6) на ручную проверку, а reasoning (обоснование) – понять логику модели.

D. Отбор контекстных примеров

Для повышения качества мы включаем в промпт примеры из базы исторических диалогов (14 029 тредов, 8 языков).

Алгоритм отбора (рис. 2) применяет четыре фильтра. Пусть $d \in D$ – тред из базы D:

1) *Доставленность* – выбираются только треды с успешно доставленным ответом оператора (флаг has_sq_replies(d) = True).

2) *Глубина* – не менее 3 сообщений (total_messages(d) >= 3). Однооборотные диалоги менее информативны: они не показывают модели, как развивается переписка.

3) *Краткость* – ответ оператора не длиннее 100 слов ($|operator_{reply}(d)| \leq 100$). Длинные шаблоны увеличивают расход токенов без пользы.

4) *Дедупликация* – первые 50 символов ответа должны быть уникальны среди уже отобранных.

Итоговая выборка k примеров:

$$FewShot(k) = top_k(sort(D', depth), k) \quad (5)$$

где D' – треды, прошедшие все фильтры, $depth(d)$ – число сообщений.

Пример элемента в промпте:

```
Customer: "Yes, I'm interested.
  Could you tell me more?"
-> Interested
```

Рис. 2 показывает каскад: из 14 029 тредов фильтр доставленности оставляет 8 200, глубины – 3 400, краткости – 2 100. После сортировки по глубине и дедупликации мы отбираем 5 верхних примеров. Число 5 подобрано эмпирически после нескольких итераций: при 3 примерах качество заметно ниже, при 7 – прирост минимален, а стоимость вызова растёт.

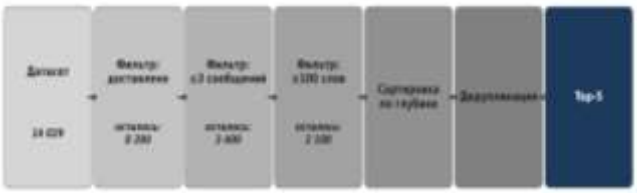

Рис. 2. Алгоритм отбора контекстных примеров (few-shot selection)

V. Экспериментальная оценка

A. Методология

Эталонной разметкой (ground truth) служит действие оператора: он принимал решение в контексте реальной переписки, что снимает необходимость дополнительной экспертной разметки. Окно данных ограничено периодом до запуска на платформе автоматических ответов, поэтому все решения в разметке приняты людьми. Существенная часть конструирования выборки – фильтрация почтового потока: помимо клиентских ответов, входящая почта содержит служебный транзит собственной рассылки (копии наших исходящих сообщений, проходящие через почтовые релеи внешних платформ и аккаунты пула), технические уведомления и автоответы серверов. Мы исключали сообщения, чьё тело совпадает с исходящими того же треда, сообщения с адресов пула платформы и служебных отправителей. Без этой фильтрации выборка вырождается: значительная доля «входящих» оказывается копиями собственных шаблонов рассылки, и оценка классификатора сводится к распознаванию собственных текстов. После фильтрации получено 24 379 подлинных клиентских сообщений в 17 811 тредах; разметка по действию оператора дала 6 103 треда с отправленным оператором ответом (по журналу действий платформы; класс RESPOND) и 6 139 тредов, где последнее сообщение клиента прочитано и оставлено без ответа (класс IGNORE). Из этого пула сформирована стратифицированная по группам тестовая выборка: 150 RESPOND и 150 IGNORE (n = 300); в ней представлены 41 группа (канал × регион) и 17 языков. Выборка невелика, но достаточна для обнаружения различий в F1 порядка 0,15 при уровне значимости 5 %. 96 % тестовых сообщений содержат цитаты – столько же, сколько в полном потоке подлинных ответов (95 %).

Использование действия оператора как эталона требует оговорки об угрозах валидности. Оператор ошибается: усталость, спешка и субъективность вносят в разметку шум, и модель, обучаемая или оцениваемая на таких метках, частично наследует эти ошибки. Мы разделяем две составляющие. Случайный шум (несистематические ошибки внимания) не благоприятствует ни одной из сравниваемых конфигураций: он одинаково занижает абсолютные значения метрик у всех методов, оставляя корректным их попарное сравнение. Систематическая же составляющая (например, устойчивое игнорирование сверхкоротких ответов вида «+») смещает сам эталон, и здесь важно рабочее определение задачи: система автоматизирует операторскую сортировку, а не выносит суждение об «истинном» намерении клиента, поэтому согласие с оператором и есть целевой показатель. Масштаб шума мы оцениваем по ручному разбору ошибок (раздел V.C): порядка 40–50 случаев из 300 (13–16 %) допускают обе метки; характерно, что один и тот же паттерн ответа («предложение актуально, но клиент переводит сделку в сторонний канал») встречается и среди проигнорированных, и среди отвеченных оператором тредов. Приоритет полноты в конвейере дополнительно смягчает цену операторской ошибки типа «пропуск»: спорный тред попадает к оператору, а не теряется. Тем не менее независимая экспертная разметка подвыборки с оценкой межаннотаторского согласия остаётся корректным следующим шагом, и мы отмечаем её в направлениях дальнейшей работы.

В качестве лексического базового метода (baseline) взят keyword-классификатор на 24 многоязычных ключевых словах двух классов. Отрицательные индикаторы (10 слов: явный отказ, неактуальность предложения, просьба прекратить рассылку, спам-маркеры) имеют приоритет: совпадение хотя бы одного даёт IGNORE. Иначе совпадение положительного индикатора (14 слов: выражение интереса, согласие, вопросы о цене, наличии и доставке) даёт RESPOND; при отсутствии совпадений сообщение помечается IGNORE. Списки покрывают

доминирующие языки потока и доступны в исходном коде платформы; их компактность отражает реалистичный для практики уровень ручной настройки лексического метода.

Доверительные интервалы получены bootstrap-методом (многократная повторная выборка с возвращением, $B = 1000$ итераций, random seed = 42, 95 %). Для попарного сравнения на одних и тех же данных – тест Макнемара с поправкой на непрерывность:

$$\chi^2 = \frac{(|b - c| - 1)^2}{b + c} \quad (6)$$

где b – число случаев, когда первая конфигурация ошиблась, а вторая нет, c – наоборот.

Помимо keyword-метода, в качестве локальной альтернативы LLM мы дообучили мультиязычный энкодер XLM-RoBERTa-base (270 млн параметров) [21]. Выбор модели обусловлен мультиязычностью потока: монолингвальные энкодеры типа RuBERT [9] не покрывают 17 языков корпуса, XLM-R – их прямой мультиязычный аналог. Обучающая выборка взята из того же размеченного пула после исключения 300 тестовых тредов и удаления дубликатов: 11 049 обучающих примеров (5 518 RESPOND, 5 531 IGNORE – классы сбалансированы естественно) и 581 валидационный; пересечение тредов между обучением и тестом отсутствует. Дообучение: 3 эпохи, AdamW, скорость обучения $2 \cdot 10^{-5}$, длина последовательности 128 токенов для очищенного текста (батч 32) и 256 для полного тела письма (батч 16), фиксированный seed 42; чекпойнт выбирался по F1 на валидации. Метки обучения порождены теми же действиями оператора, поэтому энкодер наследует систематическую составляющую операторского шума, описанную выше.

Мы сравнивали шесть конфигураций: (a) ключевые слова на полном теле письма (kw_raw), (b) ключевые слова без цитат (kw_stripped), (c) LLM на полном теле (llm_raw), (d) полный конвейер: удаление цитат + LLM + контекстные примеры (pipeline), (e) дообученный энкодер на очищенном тексте (xlmr_stripped), (f) дообученный энкодер на полном теле письма (xlmr_raw). Уточним конфигурацию оценочного стенда: LLM-конфигурации (c) и (d) используют модель Claude Sonnet 4.6 с температурой 0,1 и ограничением вывода до одного токена класса, а контекстные примеры конфигурации (d) представлены компактным фиксированным набором типовых случаев в бинарной разметке; динамический отбор примеров из базы диалогов (раздел IV.D) и параметры из раздела IV.C относятся к продакшен-конвейеру.

B. Результаты

ТАБЛИЦА I. СРАВНЕНИЕ ШЕСТИ КОНФИГУРАЦИЙ (n = 300)

Конфиг.	P	R	F1	95% CI	Acc
(a) KW, raw	0,524	0,953	0,676	[0,62;0,73]	0,543
(b) KW, strip	0,593	0,827	0,691	[0,63;0,74]	0,630
(c) LLM, raw	0,636	0,873	0,736	[0,68;0,79]	0,687
(d) Pipeline	0,633	0,873	0,734	[0,68;0,78]	0,683
(e) XLM-R, strip	0,753	0,793	0,773	[0,72;0,82]	0,767
(f) XLM-R, raw	0,715	0,853	0,778	[0,73;0,83]	0,757

Начнём с keyword-метода. На полном теле письма он срабатывает положительно на 273 из 300 сообщений (91 %), то есть практически вырождается в тривиальную стратегию «отвечать всем», F1 которой на сбалансированной выборке составляет 0,667; результат конфигурации (a) – 0,676 – статистически от этой опорной точки неотличим. Причина – цитаты: 96 % сообщений содержат цитату нашего исходного письма со словами-триггерами («цена», «доставка», «интерес» и их аналоги на языках потока), и классификатор видит их в каждом письме. Удаление цитат поднимает точность с 0,524 до 0,593 ($p = 0{,}0018$) и возвращает методу различающую способность. Однако даже после предобработки преимущество LLM над keyword-методом на выборке n = 300 статистически не подтверждается ($p = 0{,}11$): исходное противопоставление «ключевые слова против LLM» на подлинных клиентских ответах оказывается менее однозначным, чем принято считать.

LLM-конфигурации дают F1 0,736 (c) и 0,734 (d) при полноте 0,873: из 150 заинтересованных клиентов теряются 19. Для квалификации лидов полнота приоритетнее точности, и профиль LLM этому соответствует; точность 0,63 означает, что примерно каждый третий тред, направленный оператору, не требует ответа. Различие между (c) и (d) отсутствует ($p = 1{,}0$): модель отделяет ответ клиента от цитаты без внешней помощи, и предобработка сохраняет ценность только для keyword-резерва и как способ сократить промпт на 30–40 % (экономия на каждом вызове API).

Лучший результат показывает дообученный энкодер: F1 0,773 (e) и 0,778 (f), обе конфигурации значимо превосходят LLM (McNemar, $p = 0{,}007$–$0{,}015$). Выигрыш достигается за счёт точности: 0,753 против 0,633 у конвейера, ложных срабатываний почти вдвое меньше (39 против 76) при сопоставимой полноте варианта (f) (0,853 против 0,873). Причина, по-видимому, в источнике меток: энкодер обучен на одиннадцати тысячах решений операторов этой же платформы и воспроизводит их операционные правила, включая паттерн «положительный ответ с переводом сделки в сторонний канал» (раздел V.C), который few-shot LLM, опирающаяся на универсальное прочтение текста, систематически трактует иначе. Предобработка и для энкодера некритична ($p = 0{,}75$ между (e) и (f)): механизм внимания обучается игнорировать цитированный текст самостоятельно.

На рис. 3 видна трёхъярусная картина: keyword-методы у опорной линии тривиального классификатора, LLM-конфигурации выше, энкодер – верхний ярус; доверительные интервалы keyword-метода и энкодера не перекрываются. Практический вывод для вопроса о необходимости внешнего LLM: при накопленной операторской разметке локальный дообученный энкодер предпочтительнее – выше качество, нулевая стоимость вызова, отсутствие зависимости от стороннего сервиса. Few-shot LLM остаётся рациональным выбором холодного старта, когда размеченных решений ещё нет: конвейер на её основе работал в продакшене с первого дня и,

собственно, накопил ту разметку, на которой энкодер затем обучился.

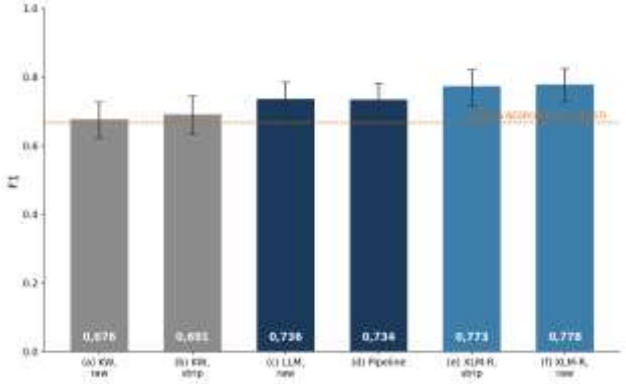

Рис. 3. Сравнение F1-меры шести конфигураций с 95% доверительными интервалами; пунктир – тривиальная стратегия «отвечать всем»

C. Анализ ошибок

При ручном разборе ошибок конвейера (76 ложных срабатываний и 19 пропусков) обнаружились повторяющиеся паттерны, хотя границы между ними условны – часть случаев попадает сразу в два.

Доминирующий паттерн ложных срабатываний – содержательно положительный ответ клиента, который оператор тем не менее оставил без ответа. Не менее 22 из 76 случаев содержат явное условие, не предусмотренное сценарием кампании: требование нестандартных условий сделки, перенос общения в сторонний канал (например: «Да, предложение в силе – свяжитесь со мной напрямую»; здесь и далее примеры приведены в переводе и перефразированы в целях анонимизации), требование телефонного звонка. Текстуально это заинтересованный ответ – модель закономерно предсказывает RESPOND, – но операционно тред бесперспективен, и оператор его закрывает. Ещё около 20 случаев – положительные ответы без видимых ограничений, оставленные без ответа; здесь правдоподобны и операторская усталость, и контекст, невидимый в тексте (например, параллельная переписка в стороннем канале).

Тот же паттерн «положительный ответ с переводом в сторонний канал» встречается и на противоположной стороне: среди 19 пропусков есть случаи, когда на структурно идентичный ответ («Да, актуально – свяжитесь со мной напрямую») оператор всё-таки ответил. Один и тот же тип сообщения оказывается по обе стороны решающей границы – это прямое количественное проявление систематической составляющей операторского шума, обсуждённой в разделе V.A: модель здесь упирается не в понимание текста, а в непоследовательность самой целевой переменной.

Оставшиеся ошибки распределяются между сверхкороткими сообщениями (одиночное слово, «J», эмодзи), ответами «не по адресу» (получатель не считает себя релевантным адресатом обращения или просит исключить его из рассылки – сюда же попало требование об удалении персональных данных по ст. 15 GDPR) и враждебными репликами, на которые оператор иногда отвечает в порядке деэскалации.

D. Стоимость

Стоимость классификации одного треда – около \$0,004 (вызов LLM с промптом в несколько сотен токенов). Тарифы провайдера менялись за последний год, так что цифра приближённая. Ручная классификация обходится примерно в \$0,50/тред – при ставке \$15/час и средней скорости 2 минуты на тред (чтение, решение, маркировка в CRM). Эти 2 минуты – средний показатель по внутренней статистике нашей платформы.

Обозначим N – ежедневное количество тредов, $C_{manual} = 0{,}50$ \$/тред, $C_{LLM} = 0{,}004$ \$/тред, P – точность (precision). Эффективная стоимость обработки одного заинтересованного клиента с учётом ложных срабатываний:

$$C_{eff} = \frac{1-P}{P} \cdot C_{manual} + C_{LLM} \quad (7)$$

Годовая экономия при 250 рабочих днях:

$$\Delta_{savings} = N \cdot (C_{manual} - C_{LLM}) \cdot 250 \quad (8)$$

Точка безубыточности:

$$N_{breakeven} = \frac{C_{fixed}}{C_{manual} - C_{LLM}} \quad (9)$$

где $C_{fixed} \approx 2$ \$/день (серверная инфраструктура, мониторинг). Соотношение стоимости – примерно 125:1 в пользу автоматизации. При 500 тредах в день дневные расходы: \$2 (конвейер) против \$250 (оператор). Годовая экономия при 250 рабочих днях – порядка \$62 000, при условии стабильного потока и неизменного тарифа API.

Безубыточность достигается при 4 тредах в день. Рабочий объём нашей платформы (500+ тредов/день) превышает эту точку более чем в 100 раз.

Точность 63,3 % означает, что 36,7 % тредов, переданных оператору, не требуют ответа. По формуле (7) издержки ложных срабатываний составляют около \$0,29 на одного заинтересованного клиента, а эффективная стоимость его обработки – \$0,294. Даже с учётом этого автоматизация дешевле ручной классификации в 1,7 раза.

Дообученный энкодер меняет экономику задачи. Стоимость вызова нулевая (локальный инференс на CPU), внешний API не нужен, а точность 0,753 снижает издержки ложных срабатываний по формуле (7) до \$0,16 на заинтересованного клиента. Плата – потребность в накопленной разметке (порядка 10^4 операторских решений) и периодическое переобучение при дрейфе потока; few-shot LLM этих требований не имеет и потому остаётся решением холодного старта.

VI. Заключение

На выборке из 300 подлинных клиентских ответов с эталоном по действию оператора сравнение шести конфигураций дало трёхъярусную картину: keyword-метод (F1 0,68–0,69; на полном теле письма он практически вырождается в стратегию «отвечать всем»), few-shot LLM (0,73–0,74) и дообученный мультиязычный энкодер XLM-R (0,77–0,78), значимо превосходящий LLM-конфигурации ($p < 0{,}01$) при почти вдвое меньшем числе ложных срабатываний.

Практически важны два вывода. Во-первых, чувствительность к цитированному тексту – свойство только лексических методов: и LLM (p = 1,0), и дообученный энкодер (p = 0,75) классифицируют полное тело письма не хуже очищенного, поэтому модуль удаления цитат нужен keyword-резерву и экономии промпта, но не самим моделям. Во-вторых, накопленная операторская разметка конвертируется в качество: локальная модель, обученная на ~11 тыс. решений, обходит внешний LLM, обнуляет стоимость вызова и снимает зависимость от стороннего провайдера. Few-shot LLM при этом остаётся рациональной точкой входа: конвейер на её основе работал с первого дня без разметки и накопил те самые решения, на которых затем обучился энкодер.

Следует оговорить ограничения: результаты получены на одной платформе и одном типе потока (переписка через почтовые релеи внешних платформ, доминирующий top-posting), задача поставлена бинарно, а эталонная разметка наследует шум операторских решений, систематическую составляющую которого мы наблюдаем напрямую (разделы V.A, V.C). В дальнейшей работе мы видим независимую экспертную переразметку подвыборки с оценкой межаннотаторского согласия, переход к многоклассовой таксономии намерений (заинтересован, отказ, запрос информации, конверсия), дистилляцию энкодера для удешевления инференса и перенос конвейера на мессенджеры, где структура сообщений иная.

БИБЛИОГРАФИЯ

[1] Saito Y., Bershad B. N., Levy H. M. Manageability, availability and performance in Porcupine: A highly scalable, cluster-based mail service // Proc. 17th ACM Symposium on Operating Systems Principles (SOSP). 1999. С. 1–15.

[2] Dam S. K., Hong C. S., Qiao Y., Zhang C. A complete survey on LLM-based AI chatbots // arXiv:2406.16937. 2024.

[3] AlShaikh M., Alrajeh Y., Alamri S., Melhem S., Abu-Khadrah A. Supervised methods of machine learning for email classification: A literature survey // Systems Science & Control Engineering. 2025. Т. 13, № 1. DOI: 10.1080/21642583.2025.2474450.

[4] Lampert A., Dale R., Paris C. Segmenting email message text into zones // Proc. 2009 Conference on Empirical Methods in Natural Language Processing (EMNLP). Singapore, 2009. С. 919–928.

[5] Novelo R., Rocha Silva R., Bernardino J. A literature review of personalized large language models for email generation and automation // Future Internet. 2025. Т. 17, № 12, ст. 536. DOI: 10.3390/fi17120536.

[6] Пичугов А., Намиот Д., Зубарева Е. Современные методы обучения больших языковых моделей с минимумом данных: от одного примера к абсолютному нулю – академический обзор // International Journal of Open Information Technologies. 2025. Т. 13, № 6. С. 114–124.

[7] Николаев П. Л. Метод объяснимости трансформера BERT при решении задачи классификации текстов // International Journal of Open Information Technologies. 2026. Т. 14, № 3. С. 43–47.

[8] Намиот Д., Ильюшин Е. Архитектура LLM-агентов // International Journal of Open Information Technologies. 2025. Т. 13, № 1. С. 67–74.

[9] Kuratov Y., Arkhipov M. Adaptation of deep bidirectional multilingual transformers for Russian language // Computational Linguistics and Intellectual Technologies: Proc. International Conference "Dialogue 2019". Moscow, 2019. Т. 18. С. 333–339.

[10] Chae Y., Davidson T. Large language models for text classification: From zero-shot learning to instruction-tuning // Sociological Methods & Research. 2025. DOI: 10.1177/00491241251325243.

[11] Shay M., Davidson R., Grinberg N. EnronSR: A benchmark for evaluating AI-generated email replies // Proc. International AAAI Conference on Web and Social Media (ICWSM). 2024. Т. 18. С. 2063–2075. DOI: 10.1609/icwsm.v18i1.31448.

[12] Repke T., Krestel R. Bringing back structure to free text email conversations with recurrent neural networks // Advances in Information Retrieval (ECIR 2018). Lecture Notes in Computer Science, т. 10772. С. 114–126. DOI: 10.1007/978-3-319-76941-7_9.

[13] Jardim B., Rei R., Almeida M. S. C. Multilingual email zoning // Proc. 16th Conference of the European Chapter of the Association for Computational Linguistics: Student Research Workshop (EACL-SRW 2021). С. 88–95. DOI: 10.18653/v1/2021.eacl-srw.13.

[14] Melendez R., Ptaszynski M., Masui F. Comparative investigation of traditional machine-learning models and transformer models for phishing email detection // Electronics. 2024. Т. 13, № 24, ст. 4877. DOI: 10.3390/electronics13244877.

[15] Jbene M., Chehri A., Saadane R., Tigani S. Intent detection for task-oriented conversational agents: A comparative study of recurrent neural networks and transformer models // Expert Systems. 2025. Т. 42, № 2, ст. e13712. DOI: 10.1111/exsy.13712.

[16] Wulf J., Meierhofer J. Exploring the potential of large language models for automation in technical customer service // Proc. Spring Servitization Conference (SSC 2024). arXiv:2405.09161.

[17] Guo Y., Xie Z., Chen X., Chen H., Wang L., Du H., Wei S., Zhao Y., Li Q., Wu G. ESIE-BERT: Enriching sub-words information explicitly with BERT for intent classification and slot filling // Neurocomputing. 2024. Т. 591, ст. 127725. DOI: 10.1016/j.neucom.2024.127725.

[18] Карпов Д. А., Коновалов В. П. Энкодер-агностичные модели типа Трансформер: перенос знаний на разговорных задачах для русского языка // Речевые технологии. 2023. № 2. С. 64–77.

[19] Исаев Р. Р., Ильюшин Е. А. Разработка метода самокоррекции больших языковых моделей с помощью обучения с подкреплением // International Journal of Open Information Technologies. 2025. Т. 13, № 6. С. 1–9.

[20] Косяненко И. А., Болбаков Р. Г. Сбор набора данных для автоматической генерации сообщений коммитов // Russian Technological Journal. 2025. Т. 13, № 2. С. 7–17. DOI: 10.32362/2500-316X-2025-13-2-7-17.

[21] Conneau A., Khandelwal K., Goyal N., Chaudhary V., Wenzek G., Guzmán F., Grave E., Ott M., Zettlemoyer L., Stoyanov V. Unsupervised cross-lingual representation learning at scale // Proc. 58th Annual Meeting of the Association for Computational Linguistics (ACL 2020). 2020. С. 8440–8451. DOI: 10.18653/v1/2020.acl-main.747.

LLM-Based Intent Classification in Corporate Email Communications

M. S. Zamula, I. A. Loskutov

Abstract – Incoming replies to corporate email campaigns must be triaged: does the customer intend to continue the conversation? We compare three approaches on live traffic of a production platform (1,300+ accounts, 17 languages): a keyword classifier, an LLM with few-shot examples and structured output (tool use), and a fine-tuned multilingual encoder (XLM-RoBERTa). Ground truth is the operator's action; the test set contains 300 genuine customer replies, filtered from the campaign's own relay transit. Quoted text of the original message is present in 96% of replies and destroys the naive keyword baseline: on raw bodies it is statistically indistinguishable from the trivial respond-to-all strategy (F1 0.676 vs. 0.667). The LLM pipeline reaches F1 = 0.73; the encoder fine-tuned on 11k operator decisions reaches F1 = 0.78, significantly outperforming the LLM (McNemar $p < 0.01$) with nearly half the false positives and zero per-call cost. Both the LLM and the encoder are insensitive to quoted text ($p \geq 0.75$); quote stripping remains useful only for the keyword fallback and prompt economy. The pipeline runs in production; validity of operator-action labels and cost trade-offs are discussed

.

Keywords – intent classification, large language models, email processing, few-shot learning, encoder fine-tuning, XLM-RoBERTa, text preprocessing, quoted text, structured output

REFERENCES

[1] Y. Saito, B. N. Bershad, and H. M. Levy, "Manageability, availability and performance in Porcupine: A highly scalable, cluster-based mail service," in *Proc. 17th ACM Symp. on Operating Systems Principles (SOSP)*, 1999, pp. 1–15.

[2] S. K. Dam, C. S. Hong, Y. Qiao, and C. Zhang, "A complete survey on LLM-based AI chatbots," arXiv:2406.16937, 2024.

[3] M. AlShaikh, Y. Alrajeh, S. Alamri, S. Melhem, and A. Abu-Khadrah, "Supervised methods of machine learning for email classification: A literature survey," *Systems Science & Control Engineering*, vol. 13, no. 1, 2025.

[4] A. Lampert, R. Dale, and C. Paris, "Segmenting email message text into zones," in *Proc. Conf. on Empirical Methods in Natural Language Processing (EMNLP)*, Singapore, 2009, pp. 919–928.

[5] R. Novelo, R. Rocha Silva, and J. Bernardino, "A literature review of personalized large language models for email generation and automation," *Future Internet*, vol. 17, no. 12, art. 536, 2025.

[6] A. Pichugov, D. Namiot, and E. Zubareva, "Modern methods for training large language models with minimal data: From one example to absolute zero – an academic review," *Int. J. Open Inf. Technol.*, vol. 13, no. 6, pp. 114–124, 2025.

[7] P. L. Nikolaev, "Explainability method of BERT transformer for solving text classification problem," *Int. J. Open Inf. Technol.*, vol. 14, no. 3, pp. 43–47, 2026.

[8] D. Namiot and E. Ilyushin, "On architecture of LLM agents," *Int. J. Open Inf. Technol.*, vol. 13, no. 1, pp. 67–74, 2025.

[9] Y. Kuratov and M. Arkhipov, "Adaptation of deep bidirectional multilingual transformers for Russian language," in *Computational Linguistics and Intellectual Technologies: Proc. Int. Conf. "Dialogue 2019"*, Moscow, 2019, vol. 18, pp. 333–339.

[10] Y. Chae and T. Davidson, "Large language models for text classification: From zero-shot learning to instruction-tuning," *Sociological Methods & Research*, 2025.

[11] M. Shay, R. Davidson, and N. Grinberg, "EnronSR: A benchmark for evaluating AI-generated email replies," in *Proc. Int. AAAI Conf. on Web and Social Media (ICWSM)*, vol. 18, pp. 2063–2075, 2024.

[12] T. Repke and R. Krestel, "Bringing back structure to free text email conversations with recurrent neural networks," in *Advances in Information Retrieval (ECIR 2018)*, LNCS, vol. 10772, pp. 114–126.

[13] B. Jardim, R. Rei, and M. S. C. Almeida, "Multilingual email zoning," in *Proc. EACL Student Research Workshop*, 2021, pp. 88–95.

[14] R. Melendez, M. Ptaszynski, and F. Masui, "Comparative investigation of traditional machine-learning models and transformer models for phishing email detection," *Electronics*, vol. 13, no. 24, art. 4877, 2024.

[15] M. Jbene, A. Chehri, R. Saadane, and S. Tigani, "Intent detection for task-oriented conversational agents: A comparative study of recurrent neural networks and transformer models," *Expert Systems*, vol. 42, no. 2, art. e13712, 2025.

[16] J. Wulf and J. Meierhofer, "Exploring the potential of large language models for automation in technical customer service," in *Proc. Spring Servitization Conf. (SSC 2024)*, arXiv:2405.09161.

[17] Y. Guo *et al.*, "ESIE-BERT: Enriching sub-words information explicitly with BERT for intent classification and slot filling," *Neurocomputing*, vol. 591, art. 127725, 2024.

[18] D. A. Karpov and V. P. Konovalov, "Encoder-agnostic transformer models: Knowledge transfer for conversational tasks for the Russian language," *Rechevye Tekhnologii*, no. 2, pp. 64–77, 2023.

[19] R. R. Isaev and E. A. Ilyushin, "Development of method for self-correction of large language models via reinforcement learning," *Int. J. Open Inf. Technol.*, vol. 13, no. 6, pp. 1–9, 2025.

[20] I. A. Kosyanenko and R. G. Bolbakov, "Dataset collection for automatic generation of commit messages," Russian Technological Journal, vol. 13, no. 2, pp. 7–17, 2025.

[21] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, and V. Stoyanov, "Unsupervised cross-lingual representation learning at scale," in Proc. 58th Annual Meeting of the Association for Computational Linguistics (ACL), 2020, pp. 8440–8451.