DOI: 10.17323/2587-814X.2025.1.7.21

Модель рекомендательной системы на основе технических событий

К.И. Пашигорев
E-mail: kipashigorev@sberbank.ru

А.О. Резников
E-mail: aoreznikov@sberbank.ru

ПАО Сбербанк, Москва, Россия

Аннотация

Рекомендательные системы широко применяются в коммерческой сфере, алгоритмы и архитектуры рекомендательных систем схожи в различных областях применения и доказали свою эффективность. Рекомендации строятся на основании профиля пользователя, манере его поведения на различных ИТ-ресурсах (ИТ – информационные технологии), а также по схожим пользователям. При этом применение рекомендательных систем в специализированных областях не распространено. Новой перспективной областью применения рекомендательных систем являются подразделения блока «Технологии» Сбера, а пользователями будут являться сами ИТ-эксперты. Рассмотрение комбинации рекомендательной системы, машинного обучения (machine learning, ML) и LLM (Large Language Model, большая языковая модель) и проектирование этих инструментов в единой системе является целью данной статьи. Объемы данных в настоящее время измеряются петабайтами (10^{15} байт) и эксабайтами (10^{18} байт), и чтобы обрабатывать даже техническую информацию (метаданные/технодданные) из окружающего ИТ-ландшафта, из используемых экспертами ИТ-систем, необходимы помощники – AI-агенты (AI – Artificial Intelligence, искусственный интеллект). В статье приводится обзор литературы в части применения рекомендательных систем в комбинации с LLM-приложениями, предлагается модель архитектуры приложения, которое из технических журналов событий формирует человекочитаемые новости. Система спроектирована для группы пользователей, которые работают с большими данными (ML-инженеры, аналитики данных и исследователи данных), представляет собой совокупность технологий рекомендательной системы, LLM и модели машинного обучения. Также в статье приводятся первые результаты проведенного исследования.

Ключевые слова: рекомендательная система, матричная факторизация, промпт-инжиниринг, LLM, AI-агенты

Цитирование: Пашигорев К.И., Резников А.О. Модель рекомендательной системы на основе технических событий // Бизнес-информатика. 2025. Т. 19. № 1. С. 7–21. DOI: 10.17323/2587-814X.2025.1.7.21

Введение

Сегодня мы становимся свидетелями повсеместного создания и использования AI-агентов. Такие инструменты как рекомендательные системы, экспертные системы, голосовые помощники, стали обычным делом. Но для эффективного решения практических задач обособленное использование отдельных инструментов перестает быть эффективным, поэтому архитектуры систем становятся все более сложными, и в организациях проектируются комплексные системы для решения сложных задач. Случаи проектирования сложных систем будут рассмотрены далее в разделе 1.

В части комбинации технологий можно выделить популярные в настоящее время AI-агенты – цифровые помощники для выполнения комплекса задач: несколько AI-агентов смогут взаимодействовать между собой и автономно выполнять сложные задачи без вмешательства человека [1, 2], и быть его безусловным помощником. Реализация AI-агентов во всех сферах жизни, в том числе обработка человеком больших объемов разрозненных данных с помощью умных помощников является актуальной задачей ближайших лет.

Рекомендательные системы стали универсальным инструментом, применяемым в различных сферах, в том числе и несвязанных с интернетом: здравоохранение, образование, логистика, и др. Перспективным является внедрение рекомендательных систем в подразделениях Технологий. Появление мощных языковых моделей на основе архитектуры трансформеров (LLM) открыло новый подход к решению задачи обработки больших объемов данных. Используя LLM стало возможно извлекать релевантную информацию из больших объемов технических данных, что ранее было задачей, требующей значительных человеческих ресурсов; идентифицировать целевые аудитории среди сотрудников подразделения Технологий, которые могут быть заинтересованы в конкретных событиях или информации; работать с различными типами профессиональной информации, включая метаданные, технологические события, новости из разрозненных каналов и чатов, статьи, профессиональные встречи, и др., и объединить их все в едином интерфейсе. Фокусируясь на профессиональных интересах и компетенциях сотрудников, проектируемая система сможет предоставлять более точные рекомендации и акцентировать внимание на релевантных технических событиях, не отвлекая сотрудника на неинтересные события, которые находятся за рамками его профессиональной деятельности.

Применение LLM для анализа журналов логирования было рассмотрено в [3] и [4]. Авторы рассматривают модель снижения числа аномалий, анализируя журналы логов. При этом в рассматриваемом авторами подходе присутствует ограничение на состав полей – так называемое свойство слабой адаптивности, и, как следствие, необходимость дообучения LLM в случае изменения структур входных данных. Т.е. технические данные использовались для проведения анализа и прогнозирования и имеют ограничение на входные структуры данных. С развитием LLM появляются новые сценарии работы с техническими данными, а именно оповещение конечного пользователя об актуальных для него произошедших событиях в ИТ-ландшафте. Также авторы явно указывают на другой недостаток работы моделей с журналами логов – это недостаточная интерпретируемость. При обнаружении аномалий на основе журналов интерпретируемые результаты имеют решающее значение для того, чтобы администратор и аналитики могли доверять автоматическому анализу и действовать в соответствии с ним. Модель системы, предлагаемая в данной статье, позволит интегрировать LLM в процесс промышленной разработки, гарантируя, что информация не будет утрачена и будет своевременно доставлена получателю, а также будет однозначна интерпретируема. Такой подход является новым способом применения LLM на уже существующих технических данных.

Предлагаемая модель системы также предполагает формирование портрета пользователя для кор-

ректного подбора рекомендаций. Для этого в терминологию вводятся такие понятия как портрет пользователя или профиль пользователя. Пользователями системы являются эксперты, работающие с данными – ML-инженеры, аналитики данных и исследователи данных, которые находятся внутри разнородного ИТ-ландшафта, сотни и тысячи систем которого регулярно обновляются, и в которых регулярно происходят десятки тысяч событий ежемесячно, за которыми необходимо следить.

1. Рекомендательные системы в крупных динамических ИТ-ландшафтах для фильтрации метаданных

Рекомендательные системы используются в различных сферах, которые включают как коммерческую деятельность – электронная коммерция, дистанционное образование и развлекательные платформы – эти системы способствуют подбору товаров, услуг и контента, соответствующих индивидуальным предпочтениям и поведенческим характеристикам пользователей; так и все более активно применяются для решения задач технического характера в сферах безопасности и строительства, в качестве инструмента обобщения больших текстов, и др.

В обзорной статье [5] говорится о том, что ключевым элементом, применяемым в современных информационных технологиях, является анализ данных с привлечением технологий искусственного интеллекта. Автор рассматривает машинное обучение, нейронные сети и обработку естественного языка в качестве основных элементов искусственного интеллекта, которые применяются для анализа данных.

В статье [6] автор приводит описание гибридных систем, в которых в рамках построения систем рекомендаций комбинируются различные подходы и алгоритмы для достижения более точных персонализированных рекомендаций. А также описывает преимущество такого комбинирования перед моделями совместной фильтрации и рекомендаций на основе контента. При этом автор утверждает, что гибридные системы решают проблемы холодного старта и агрегирования информации из различных источников. В статье [7] авторы дополнительно классифицируют гибридные системы на монолитные, смешанные и ансамблевые модели, определяя монолитные рекомендаторы как совокупность частей различных типов рекомендательных алгоритмов, а смешанные рекомендаторы – как комбинацию результатов всех входящих в него рекомендаторов. При этом выделяют сложность разработки таких систем, т.к. требуются значительные ресурсы и усилия.

В статье [8] авторы для целей своего исследования также используют методы обобщения и применяют на входном потоке сообщения социальных сетей, исчисляемые миллионами. Авторы рассматривают случай, что длина обобщенного топика является управляемой переменной, и далее озвучивают мнение, что автоматическое обобщение – это задача создания последовательной сокращенной версии документа, в которой изложены его основные положения. При этом в зависимости от выбранного варианта использования целевая длина конечного результата может быть выбрана относительно длины входного документа или может быть ограничена.

Авторы статьи [9] рассматривают применение LLM-приложений в строительстве для составления автоматизированных отчетов на основании технической информации, и также называют оптимизацию ресурсов в качестве одного из получаемых результатов. Авторы применяют в своей работе термин «интеллектуализация инспекции строительства», а также указывают на недостатки в исследованиях, что в настоящее время строительная инспекция в основном полагается на выполнение и анализ вручную.

Авторы статьи [10] рассматривают применение комбинации технологий машинного обучения, LLM и генеративного ИИ. Аналогично предыдущим работам авторы преследуют цель оптимизации рабочего времени экспертов в специфической области, и достигают ее с применением современных технологий. Авторы проводят исследования на модели GPT-4 (Generative Pre-trained Transformer 4) и упоминают про риск получения галлюцинаций в качестве ответов LLM, а также определяют недетерминизм (разные ответы во время разных сеансов) как дополнительный уровень сложности и непредсказуемости в процессе генерации ответов.

Авторы статьи [11] также рассматривают для решения своей задачи комбинацию LLM-приложения и RecSys (Recommender Systems, рекомендательные системы) и особенно акцентируют внимание на проблеме холодного старта, рассматривая различные варианты. Авторы называют

свою систему A-LLMRec (All-round LLM-based Recommender system), т.к. основная идея состоит в том, чтобы позволить LLM напрямую использовать коллективные знания, содержащиеся в предварительно обученной современной системе рекомендаций на основе совместной фильтрации (CF-RecSys) (CF – Collaborative Filtering, совместная фильтрация), чтобы можно было совместно использовать новые возможности LLM, а также высококачественные представления пользователей и товаров, которые уже обучены в современной системе CF-RecSys. Но эксперимент все-таки проводится не на технических данных, а с человекочитаемыми заголовками и описаниями.

Авторы статьи [12] позиционируют свой подход как новаторский. Они подробно описывают применение мультиагентной архитектуры на базе LLM, в которой применяется такая цепочка агентов: Восприниматель (Perceive) – Учащийся (Learn) – Исполнитель (Act) – Критик (Critic) – Мыслитель (Reflect). Архитектура задействует цикл Учись (Learn) – Действуй (Act) – Критикуй (Critic) и механизм рефлексии для повышения эффективности взаимодействия с пользователями. Как и в ранее рассмотренных статьях авторы этой статьи делают акцент на холодном старте. При этом в центре внимания находится баланс между точностью рекомендаций и удовлетворенностью пользователей. Эксперимент также проводился на человекочитаемых данных. В целом в качестве инновационности представлен именно LLM-компонент (состоящий из небольших агентов/модулей).

В рассмотренных источниках большое внимание уделяется большим языковым моделям, с архитектурой которых можно подробно ознакомиться в [13]. Также отметим, что создание интеллектуальных ассистентов в различных предметных областях активно рассматривается не только зарубежными авторами и исследователями, примеры которых в достаточном объеме приведены в данном разделе, но также и отечественными авторами и исследователями [14, 15]. Отличием системы, проектируемой в данном исследовании, является то, что перед проектируемой системой не ставится таких бизнес-целей как удержание аудитории или увеличение потребляемого контента, а преследуется цель уведомить пользователя о релевантных для его специфики и задач изменениях в информационной инфраструктуре компании. Например, выход нового дата-продукта, связанного с проектами пользователя, произошедшие изменения в актуальных для него данных, выход релизов релевантных для пользователя систем, изменение метаданных, и др. Выделим две проблемы, которые были решены в [4] не до конца – слабая адаптивность моделей и недостаточная интерпретируемость результатов, и рассмотрим далее в этой статье способы решения в том числе этих задач.

2. Архитектура системы умной новостной ленты технособытий

Первичной информацией о событии являются технические данные, сгенерированные другими системами (журналы событий), которые трудно воспринимаются человеком. Перед нами стоит задача не только найти актуальное для пользователя событие, но и привести его к виду, который легко воспринимается человеком. Ввиду широкого спектра обрабатываемых событий, универсальным решением для приведения информации о событии к приемлемому для пользователя виду является внедрение большой языковой модели (LLM) для суммаризации технических данных в новостное уведомление, содержащее все ключевые аспекты о данном событии. Кроме того, для улучшения сопоставления пользователей с конкретным событием мы будем использовать выделение ключевых свойств с помощью тегирования.

Для определения контекста исследования введем в использование термин «техноданные»: это могут быть тысячи событий, происходящих в сотнях систем ИТ-ландшафта. В *таблице 1* приведен простой пример структуры таких событий.

Таблица 1.

Пример таблицы события

id	Date	Event's type	Description	...	Author	Source
uuid	Date	String	Text	...	Varchar	Varchar

Для решения задачи исследования была спроектирована интеллектуальная система для обработки большого объема событий, архитектура которой представлена на *рисунке 1.*

Рассмотрим основные элементы этой системы.

- События, генерируемые в системах ИТ-ландшафта, сохраняются в СУБД.
- Микросервис Giga-intgr отправляет данные технического события в LLM. Ответ от LLM (обработанные события) в виде сформированной новости и присвоенных тегов записывает в СУБД.

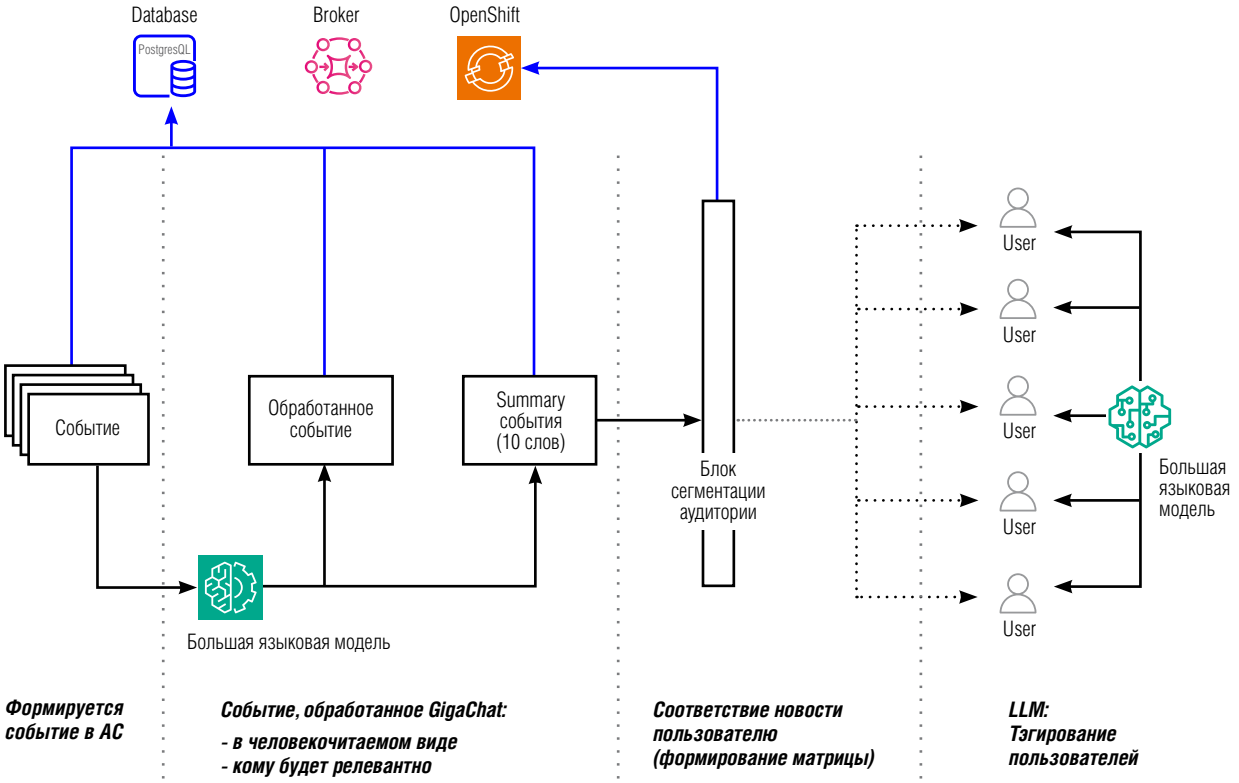

Рис. 1. Основные элементы интеллектуальной системы для обработки большого объема событий.

- Микросервис Jira-intgr запрашивает список задач пользователей в системе учета задач (Jira) и с помощью брокера сообщений передает в Giga-intgr. Giga-intgr передает полученное сообщение в LLM. Ответ от LLM (тегирование пользователей) в виде тегов для каждого пользователя записывает в СУБД.
- Сервис SegmentOfAuditory-serv формирует матрицу соответствия пользователя и новости с помощью модели.
- Сервис главной страницы получает новости для отображения в ленте из СУБД через API.

Общий итог применения LLM в представленной модели архитектуры заключается в способности решать следующие ключевые задачи:

- тегирование пользователей, то есть присвоение им соответствующих меток, отражающих их интересы и предпочтения;
- тегирование новостей, позволяющее классифицировать новостные материалы по различным категориям и тематикам;
- формирование краткого содержания новостей на основе технической информации, чтобы пользователь мог быстро ознакомиться с содержанием сообщения без необходимости читать весь текст;
- на рекомендательную систему приходится решение задачи подбора интересных новостей для каждого конкретного пользователя на основе его персональных интересов и предпочтений.

Представленная архитектура системы способна учитывать большое количество следующих параметров:

- большое количество информационных систем в качестве источников, которое может измеряться в диапазоне от одного до нескольких тысяч;
- большое количество событий с различной структурой, которая также может меняться как в зависимости от события или от системы-источника, так и со временем; количество событий может измеряться сотнями тысяч, и количество атрибутов в структуре этих событий может изменяться в диапазоне от двух до нескольких сотен;
- большое количество ролей пользователей и еще большее количество пользователей.

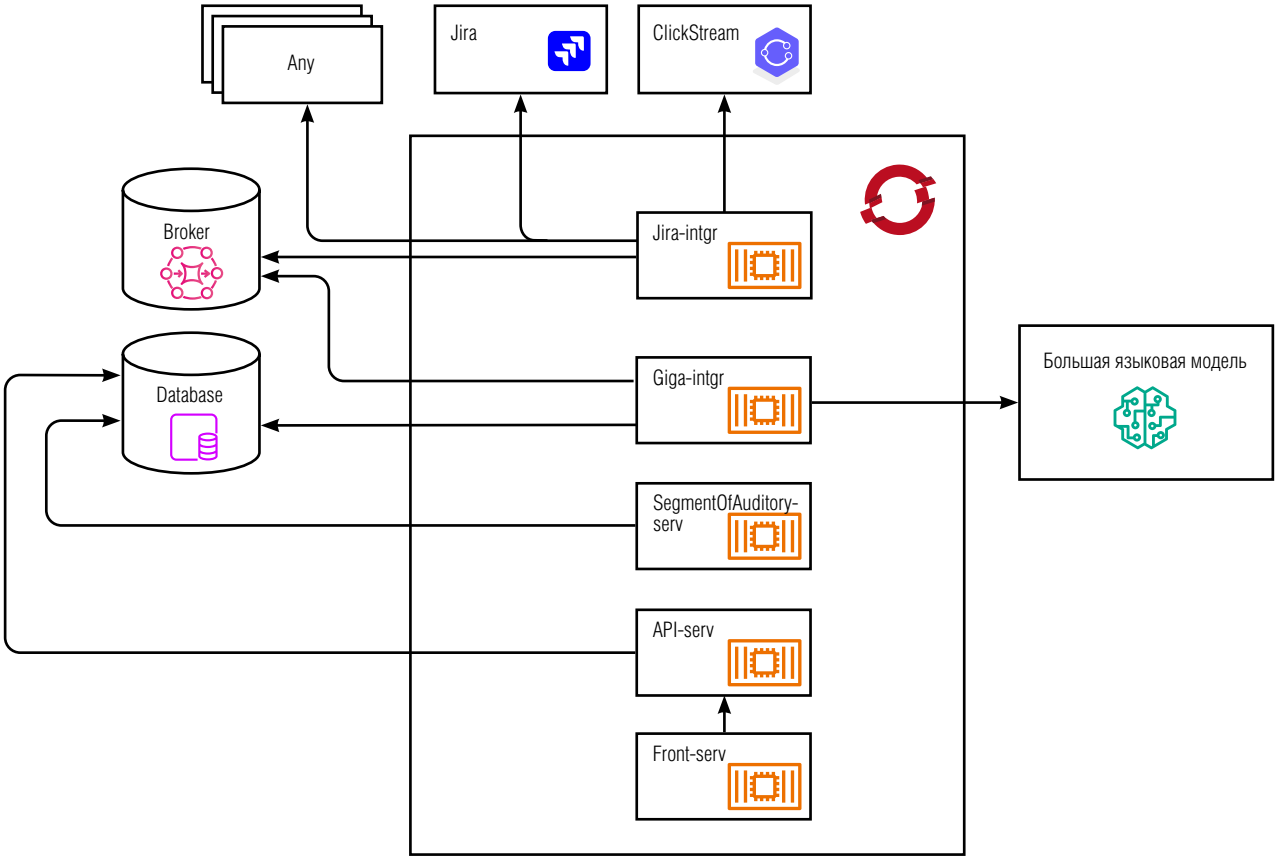

Рис. 2. Компонентная диаграмма.

Таблица 2.

Компоненты системы

Компонент	Описание
Большая языковая модель	В качестве средства доступа к большой языковой модели будет использована технология GigaChat API [16].
Any, Jira, ClickStream	Данные компоненты являются внешними системами – источниками данных. В качестве источника может выступать любая произвольная система.
СУБД	Компонент (stateful) для хранения данных в структурированном виде.
Брокер Kafka	Компонент (stateful) для реализации взаимодействия между микросервисами.
Giga-intgr	Компонент (микросервис) для реализации взаимодействия с большой языковой моделью. Реализован на Python.
Jira-intgr	Компонент (микросервис) для реализации получения данных из окружающего ИТ-ландшафта. Реализован на NodeJS.
SegmentOfAuditory-serv	Данный компонент (микросервис) служит для выбора, преобразования, комбинирования, и иных способов подготовки данных для поиска полезных закономерностей, а также для извлечения закономерностей из полученных данных.
API-serv	API для главной страницы ленты реализована на NodeJS.
Front-serv	Главная страница ленты (GUI) реализована на JavaScript – React.

Архитектура системы изображена на *рисунке 2*, а описание технических компонент приведено в *таблице 2*.

Архитектура спроектирована в микросервисном стиле и предполагает размещение в динамической инфраструктуре на кластере OpenShift в целях повышения уровня горизонтального масштабирования.

3. Оптимизация работы с техническими событиями

3.1. Постановка задачи рекомендации новостей

В целях сохранения связности изложения и для формализации задачи приведем описание метода матричной факторизации, который был неоднократно и подробно рассмотрен в [17–19], и представляет собой декомпозицию матрицы-источника на произведение двух других матриц меньшего ранга. В исследовании, которое описывается в данной работе, итоговая матрица R декомпозируется на матрицы A и B: рассматривается множество сотрудников $U = \{u_1, u_2, ..., u_m\}$, множество задач $T = \{t_1, t_2, ..., t_n\}$ и множество новостей $N = \{n_1, n_2, ..., n_k\}$. Матрица задач сотрудника $A = (a_{ij}) \in \{0;1\}^{m \cdot n}$ определяется таким образом, что $a_{ij} = 1$, если сотрудник u_i занимается задачей t_j, и $a_{ij} = 0$ в противном случае. Матрица новостей $B = (b_{ij}) \in \mathbb{R}^{n \cdot k}$, представляет собой релевантность новости n_i к задаче t_j. Целью исследования является нахождение матрицы рекомендаций $R = (r_{ij}) \in \mathbb{R}^{m \cdot k}$, где r_{ij} – степень релевантности новости n_j для сотрудника u_i.

Оценка релевантности новостей для задач выполняется путем вычисления степени релевантности b_{ij} для каждой задачи t_j и новости n_i, используя методы текстового анализа BERT. Оценка релевантности новостей для сотрудника u_i осуществляется через вычисление степени релевантности r_{ij} для каждого сотрудника u_i и новости n_j, используя матрицу задач сотрудника A и матрицу новостей B. Ранжирование новостей для каждого сотрудника u_i производится по степени релевантности r_{ij} в порядке убывания.

Данный подход может быть представлен с помощью следующей формулы:

$$r_{ij} = \sum_{l=1}^{n} a_{il} \cdot b_{jl},$$

где a_{il} – степень участия сотрудника u_i в задаче t_l, а b_{jl} – степень релевантности новости n_j к задаче t_l.

Вводятся ограничения по задаче исследования: ограничение на максимальное количество рекомендаций L для каждого сотрудника и установление минимальной степени релевантности r_{min}, ниже которой новость не рекомендуется пользователю.

Для оценки используемой модели используются следующие критерии: точность (Precision), полнота (Recall) и F1-score. Целью является максимизация этих метрик для обеспечения наиболее релевантных рекомендаций для каждого сотрудника. Комбинация критериев для оценки представлена в *таблице 3*.

Таблица 3.

Матрица критериев

Новость должна быть рекомендована и действительно попала в рекомендацию (TP)	Новость должна быть рекомендована, но в действительности не попадает в рекомендацию (FP)
Новость не должна быть рекомендована, но в действительности попадает в рекомендацию (FN)	Новость не должна быть рекомендована, и действительно не попадает в рекомендацию (TN)

Такая оценка позволит определить, насколько смело модель должна подбирать получателей. Первичная оценка сначала выполняется человеком.

Важно отметить, что при обучении модели в процессе исследования будет сделано допущение о доверительном интервале: если отсутствуют ошибки первого рода и существует допустимое количество ошибок второго рода, то та же картина сохранится на всем множестве (выборке). Ошибки должны быть агрегированы рецензентом и принято решение, требуется ли дообучение модели и коррекция промпта. Ошибки второго рода исправляются промптом, ошибки первого рода требуют дообучения.

Также необходимо учитывать необходимость решения задачи “холодного старта” [20], который возникает единоразово при запуске системы.

На представленном этапе исследования сделано допущение, что технические новости предлагаются пользователям случайным образом. Чтобы оценить качество предсказаний, было выполнено разделение множества оценок V на отдельные множества V_{train} для обучения и V_{test} для тестирования. Проведение тестов для модели будет выполнено на количественных показателях, которые представлены в *таблице 4*.

Таблица 4.

Количественные характеристики эксперимента

Характеристика	Значение	Характеристика	Значение
Количество событий «Создание ДП»	40000	Обучающая выборка	75%
Количество событий «Изменение МД»	10000	Тестовая выборка	25%
Количество релизов АС КД	10–1500, включая дочерние задачи	Ранг матриц	будет принято значение *X*
Количество релизов АС СМД	10–1500, включая дочерние задачи	Шаги факторизации	будет принято значение *Y*
Количество пользователей	50	Диапазон значений исходной матрицы	{0, 1} – неявный фидбек

Характеристики, представленные в *таблице 4*, имеют отношение к специфике данной работы:

- ДП (дата-продукт) – вид информационного актива, представляющий собой структурированное описание набора промышленных данных (вид и состав данных, источники данных и способы поставки), доступных для заказа и получения через тракты распространения промышленных данных КАП с целью использования в интересах продуктов Банка и Экосистемы.
- КАП (корпоративная аналитическая платформа) – общее название решения, отвечающего за получение данных, их обработку и предоставление обработанных данных заинтересованным лицам. Централизованная платформа по сбору (загрузке), поставке (выгрузке), обработке, интеграции, исследованию и анализу данных. КАП состоит из Ядра и Пользовательского пространства (User space).
- МД (менеджер данных) – работник Банка, назначается на роль в соответствии с Политикой по управлению данными (в зависимости от объекта назначения).
- АС КД (автоматизированная система «Карта данных») – система, отвечающая за управление дата-продуктами: их атрибутами, структурой и признаком доступности к распространению в АС Супермаркет Данных.
- АС СМД (автоматизированная система «Супермаркет Данных») – единый портал взаимодействия с КАП, который позволяет изучить и заказать данные из КАП. Предназначена для распространения данных в КАП из реплик промышленных источников.

На начальной стадии эксперимента более 95% ячеек матрицы *V* будут не заполнены, т.е. матрица будет являться очень разреженной.

В качестве метрики будет использована среднеквадратичная ошибка (*RMSE*) [21]. *RMSE* позволит оценить долю новостей, которые должны были быть показаны конкретной группе пользователей и были показаны:

$$RMSE = \sqrt{\frac{1}{N-d}\sum_{i=1}^{N}\left(M_i - O_i\right)^2},$$

где M_i – прогнозируемое значение для *i*-го наблюдения в наборе данных;

O_i – наблюдаемое значение для *i*-го наблюдения в наборе данных;

N – размер выборки;

d – степени свободы в конфигурации модели. Для линейной подгонки $d = 2$.

3.2. LLM и промпт-инжиниринг

Для целей нашего исследования мы будем пользоваться уже обученной языковой моделью GigaChat [22], и сосредоточимся на промпт-инжиниринге для получения желаемого результата без дообучения модели. Значение этих терминов представлено в [23]: промпт-инжиниринг – это процесс создания эффективных и точных промптов для работы с большими языковыми моделями (LLM). Промпт (prompt – подсказка) – это текстовое описание задачи, которую необходимо выполнить с помощью ИИ-модели. Запрос, который задается модели для генерации текста, изображений, кода или других видов контента.

Для передачи в большую языковую модель будут использованы следующие элементы:

- промпт с типом «Обобщение» (Summarization) – выделение основных тезисов из текста, ключевые слова для данных типов промптов: сократи, суммаризуй;
- промпт с типом «Генерация» (Generation), ключевые слова: напиши, сочини, придумай;
- роль – для поведения в роли определенного персонажа;
- метод Zero-shot – не содержит примеров ответа, ответ большой языковой модели будет получен в свободной форме.

Предварительный список атрибутов журнала событий для передачи в большую языковую модель, на основании которой будет формироваться новость:

- ФБ трайба;
- трайб;
- название ДП;
- дата-продукт;
- категория;
- тип ДП;
- кластер;
- платформа схемы;
- схема БД;
- количество тэгов;
- имя тэга;
- таб. номер МД ДП;
- UUID;
- последний статус;
- дата назначения МД;
- флаг публикации;
- имя системы-приемника;
- площадка распространения;
- SLA и др.

Под трайбом здесь понимается группа кросс-функциональных команд, работающих над продуктовым или сервисным направлением. Каждая команда включает в себя специалистов всех необходимых профилей для создания продукта под ключ.

Таким образом, на основании атрибутов события «Изменение дата-продукта»:

- должна появиться новость: *“Витрина W для онлайн-помощника запущена в Блоке Bl”*;
- новости должны быть присвоены тэги: *{“Фокус-группа”: “Блок Bl”}, {“Тема”: “Витрина W для онлайн-помощника”}, {“Категория”: “Новости Блока Bl”}*.

При подготовке промптов важно убедиться, что не возникнет эффекта галлюцинаций [16, 24]. Также обязательным пунктом при подготовке промпта является задача обязательного прохождения проверки Цензора. Возможны случаи, когда на корректный запрос к большой языковой модели Цензор расценивает его как нежелательный для обсуждения и возвращает ответ о неуместности данной темы с просьбой сменить ее на корректную.

По результатам первичного исследования были получены следующие результаты, представленные в *таблице 5*.

В *таблице 5* представлено сравнение расстояний между промптами и результатами генерации на их основе.

Таблица 5.

Результаты первичного исследования

Расстояние между промптами	Промпт 2	Промпт 3
Промпт 1	0,03732394628917002	0,095708435241878
Промпт 2	0	0,0855076122459778
Усредненное расстояние между новостями, сгенерированными при помощи этих промптов	AI-новость промпта 2	AI-новость промпта 3
AI-новость промпта 1	0,15939979460493073	0,1789409953210833
AI-новость промпта 2	0	0,13067957637939245

- Промпт 1: *«Сгенерируйте ясную и краткую новостную статью, суммирующую ключевые детали события, описанные в предоставленных технических данных. Фокусируйтесь на представлении фактуальной информации в человекочитаемом формате, избегая любых спекуляций, предположений или избыточных украшений. Убедитесь, что статья имеет четкую и логичную структуру, использует правильную грамматику и синтаксис предложений. Используйте нейтральный тон и избегайте сенсационного языка. Цель – предоставить информативный и объективный обзор события, делая его легко понятным для читателей»*.
- Промпт 2: *«Сгенерируйте краткую и информативную новостную статью, резюмирующую ключевые детали события, описанного в предоставленных технических данных. Сосредоточьтесь на представлении фактической информации в нейтральном тоне, избегая любых спекуляций, предположений или эмоционального языка. Используйте четкую и логичную структуру и избегайте сенсационного или привлекающего внимание языка. Целью является предоставление объективного резюме события, чтобы читатели могли быстро понять, что произошло»*.
- Промпт 3: *«Как эксперт в составлении информативных уведомлений составь короткое сообщение длиной не больше 10–12 слов, которое сообщит ключевую информацию о событии»*.

Из данных в *таблице 5* можно увидеть, что даже близкие по структуре и содержанию промпты могут давать результаты с существенным отклонением друг от друга. Такое поведение можно сравнить с поведением жесткой системы обыкновенных дифференциальных уравнений (ОДУ). Исходя из того факта, что в проекте применяется уже обученная LLM без глобальной необходимости ее дообучения, а также на основании полученных результатов делается вывод о необходимости четкого формулирования промпта для получения стабильного результата.

Исходя из исследования поведения модели с различными промптами было отобрано три промпта, данные о которых были приведены в *таблице 5*. При помощи этих данных была сгенерирована тестовая выборка новостей, которая была предоставлена на разметку на платформе TagMe [25] целевой группе пользователей данного продукта, для выбора оптимального по информативности и простоте восприятия варианта сформированных новостей. По итогу был выбран второй промпт для применения в пилотной версии продукта.

3.3. Холодный старт и первые результаты

Учитывая, что пользователями данного сервиса будут являться сотрудники нашей компании, целесообразно подумать о том, как можно «подогреть» холодный старт для новых пользователей, чтобы уменьшить воронку сходимости системы к рекомендации максимально релевантных новостей. Так как наша целевая аудитория существует в том же ИТ-ландшафте, возможно помимо сбора технодannых также собирать и цифровые следы работы пользователей. Первым и самым очевидным следом являются задачи, которые заводились на пользователя, и которые пользователь заводил сам. Исходя из этого мы можем сразу выдвинуть гипотезу о том, что задачи, которые связаны с пользователем, отражают род его профессиональной деятельности, и, следовательно, сравнивая новость по близости с задачами пользователя в Jira, мы можем предполагать, насколько они актуальны для пользователя. Получив задачи пользователя и построив эмбеддинги для задач, которыми занимался пользователь, и эмбеддинги для новостей и технодannых, мы можем построить матрицу близости (косинусных расстояний) между задачами и новостями. Так мы сможем получить первичную информацию о том, какие новости вероятнее всего будут пересекаться с профессиональной деятельностью пользователя.

После построения матрицы корреляций возникает задача отображения ее в ранжированный список новостей для рекомендации пользователю. Простейшим подходом будет расчет средней корреляции для каждой новости и сортировка по полученным значениям. Однако в ходе исследования был выбран другой способ, алгоритм которого выглядит следующим образом: для каждой новости выбирается топ N ($N = 5$) новостей для каждой задачи, каждой новости присваивается вес, равный ($N - i$), где i – номер новости в топе для каждой из задач; далее все веса для каждой новости суммируются по полученным значениям и происходит сортировка. Особенностью такого подхода является то, что в конечное ранжирование попадают не все новости, однако распределение новостей происходит более «честным» способом

за счет того, что некоторые задачи плохо коррелируются как с релевантными для пользователя новостями, так и с другими задачами пользователя. Такое распределение позволяет, во-первых, снизить влияние выбросов и сделать ранжирование новостей более устойчивым к аномальным активностям пользователя, при этом сохранив релевантные для аномальных активностей новости, и, во-вторых, исключить из ранжируемой выборки нерелевантные новости, что в дальнейшем позволит более уверенно использовать на такой отфильтрованной выборке более сложные модели для выдачи новостей. Но в период холодного старта мы можем просто выводить K (количество новостей, рекомендуемых конкретному пользователю) первых новостей из этого списка.

Таким образом, когда пользователь в первый раз заходит в систему, система уже имеет первичные данные о специфике работы пользователя, основываясь на его задачах, полученных из системы учета задач. Система рассчитывает эмбеддинг по задачам пользователя, который в дальнейшем будет использован для поиска косинусного расстояния с эмбеддингами новостей. Далее пользователю предлагается ознакомиться в ленте новостей с рекомендациями, которые для него рассчитала предлагаемая модель.

Преимуществом применения предложенной модели системы является исключение применения классических условных алгоритмов для определения целевых групп пользователей, в предложенной модели системы эту функцию будет выполнять рекомендательная система, используя алгоритм косинусного сходства и матрицы наложения. Способ обработки большого количества типов событий классическими методами с использованием условных операторов не дает желаемого результата. Тем временем использование языковой модели для категоризации событий и пользователей позволяет быстро и оптимально обработать такие типы событий. Также стоит учитывать, что один и тот же тип события, но с различными входными характеристиками, такими, как, например, объект логирования, в одном случае может быть актуален для конкретного пользователя, но в другом случае с другим объектом логирования уже будет не актуален. Обработка всех возможных комбинаций входных параметров с помощью условных операторов – это дорогая операция с технической точки зрения.

Заключение

В статье представлена концепция архитектуры системы управления событиями и модель маршрутизации событий в формате новостей конкретным пользователям, а также определены критерии оценки применения модели. Данный подход использует LLM для преобразования сырых технических данных в короткие новости, которые затем доставляются пользователям через систему рекомендаций. Подобное построение интеллектуальной системы комбинирует технологии нейронных сетей, рекомендательных систем и машинного обучения для минимизации эффекта спама и своевременного оповещения пользователей. Результатом автоматизации процесса генерации новостей будет являться сокращение времени, затрачиваемого экспертом на поиск информации, и как следствие минимизирован риск критических инцидентов. Предложенная архитектура программной системы реализует взаимодействие несвязанных компонентов, объединяя их в единый AI-агент, минимизируя поток новостей в адрес одного пользователя. Также предложенная архитектура позволяет обеспечить дальнейшее развитие системы с наименьшими затратами – интегрировать в систему компоненты распознавания речи, что будет делать систему полноценным AI-помощником. Комбинация этих технологий является незаменимым помощником для поддержки экспертов в их ежедневных задачах, связанных с данными.

По итогам первого применения предложенной модели к техническим данным, которые были накоплены исторически, был получен первый результат: обычно на таких данных решалась задача детекции инцидентов. Но в новых реалиях, когда скорость изменения ИТ-ландшафта регулярно растет, приходится искать новые способы применения LLM к техническим данным.

На примере модели системы, представленной в данной статье, было извлечено более 0,5% полезных данных из общего объема сообщений, и найдены более 1% релевантных потребителей этих данных. Данный результат так же позволяет сократить временные затраты сотрудников на отслеживание изменений ИТ ландшафта. В отличие от классических подходов решения задачи обнаружения аномалий по логам, представленным в [3] и [4], подход, предлагаемый в настоящей статье, акцентируется не на анализе логов, а на упрежде-

нии потенциальных инцидентов в связи с чувствительными изменениями в ИТ-ландшафте, в продуктах, потребляемых системами пользователя, за счет своевременного оповещения об этом пользователей. Верность выбранного нами направления также подтверждает гипотеза из [4], что высокий процент ложных срабатываний может привести к пропуску важных сбоев в системе, а высокий процент ложных пропусков может привести к напрасной трате усилий разработчиков. Предлагаемый нами подход позволяет предупредить сбои и затраты на усилия разработчиков.

Также была получена метрика конечного релевантного объема информации в размере 0,05–0,15% для тестовой выборки сотрудников и сокращение объема целевой информации с сохранением информативности до не более чем 10% от изначального объема. Средний объем единицы поступающего материала с 8629 символов сократился до 189–538 символов, что более чем в 23 раза сократило объем потребляемой информации при сохранении 96% смысловой нагрузки. ■

Благодарности

Авторы выражают благодарность Управлению исследований и инноваций Сбера за проведение питч-сессии Техно-Идея и возможность проведения исследования, Управлению распространения данных и Управлению развития технологий искусственного интеллекта и машинного обучения Сбера за предоставление ресурсов на проведение исследования, а также доценту ФКН НИУ ВШЭ, кандидату педагогических наук Виденину С.А. за методические рекомендации при написании статьи.

Литература

1. Что такое агенты ИИ? // Microsoft. [Электронный ресурс]: https://learn.microsoft.com/ru-ru/azure/cloud-adoption-framework/innovate/best-practices/conversational-ai (дата обращения 20.07.2024).
2. Представлена платформа для запуска автономных AI-агентов // Сбер. [Электронный ресурс]: https://ai.sber.ru/en/post/predstavlena-platforma-dlya-zapuska-avtonomnyh-ai-agentov (дата обращения 20.07.2024).
3. Shah A., Pasha D., Zadeh E., Konur S. Automated log analysis and anomaly detection using machine learning // Frontiers in Artificial Intelligence and Applications. Vol. 358: Fuzzy Systems and Data Mining. 2022. P. 137–147. https://doi.org/10.3233/FAIA220378
4. Experience report: Deep learning-based system log analysis for anomaly detection / Z. Chen [et al.] // arXiv:2107.05908. 2021. https://doi.org/10.48550/arXiv.2107.05908
5. Мокшанов М.В. Применение искусственного интеллекта в анализе данных: обзор текущего состояния и будущих направлений // Universum: технические науки: электрон. научн. журн. 2024. № 5(122). https://doi.org/10.32743/UniTech.2024.122.5.17513
6. Еремин О.Ю. Методы реализации гибридных рекомендательных систем // E-Scio. 2023. № 3(78).
7. Куренных А.Е., Судаков В.А. Подход к разработке гибридных рекомендательных систем // Бюллетень науки и практики. 2022. Т. 8. № 11.
8. Völske M., Potthast M., Syed S., Stein B. TL;DR: Mining reddit to learn automatic summarization // Proceedings of the Workshop on New Frontiers in Summarization, Copenhagen, Denmark, 2017. P. 59–63. Association for Computational Linguistics. https://doi.org/10.18653/v1/W17-4508
9. Pu H., Yang X., Li J., Guo R. AutoRepo: A general framework for multimodal LLM-based automated construction reporting // Expert Systems with Applications. 2024. Vol. 255. Part B. Article 124601. https://doi.org/10.1016/j.eswa.2024.124601
10. Sivakumar M., Belle A.B., Shan J., Shahandashti K.K. Prompting GPT–4 to support automatic safety case generation // Expert Systems with Applications. 2024. Vol. 255. Part C. Article 124653. https://doi.org/10.1016/j.eswa.2024.124653
11. Large Language Models meet Collaborative Filtering: An efficient all-round LLM-based recommender system / Sein Kim [et al.] // arXiv:2404.11343. 2024. https://doi.org/10.48550/arXiv.2404.11343
12. RAH! RecSys-Assistant-Human: A human-centered recommendation framework with LLM agents / Y. Shu [et al.] // arXiv:2308.09904. 2023. https://doi.org/10.48550/arXiv.2308.09904
13. Attention is all you need / A. Vaswani [et al.] // arXiv:1706.03762. 2017. https://doi.org/10.48550/arXiv.1706.03762

14. Морозевич Е.С., Коротких В.С., Кузнецова Е.А. Разработка модели формирования индивидуальных образовательных траекторий с использованием методов машинного обучения // Бизнес-информатика. 2022. Т. 16. № 2. С. 21–35. https://doi.org/10.17323/2587-814X.2022.2.21.35

15. Пальчунов Д.Е., Якобсон А.А. Разработка интеллектуального помощника для подбора товаров в процессе диалога с пользователем // Бизнес-информатика. 2024. Т. 18. № 1. С. 7–21. https://doi.org/10.17323/2587-814X.2024.1.7.21

16. Побочные эффекты галлюцинаций искусственного интеллекта / А.В. Ameницкий и [др.] // Наука, инновации, образование: актуальные вопросы и современные аспекты. С. 224–235. Пенза, 2024.

17. Strömqvist Z. Matrix factorization in recommender systems: How sensitive are matrix factorization models to sparsity? // Uppsala University Publications. 2018. [Электронный ресурс]: https://uu.diva-portal.org/smash/get/diva2:1214390/FULLTEXT01.pdf (дата обращения 22.07.2024).

18. Мойсюк-Дранько П.А., Ревотюк М.П. Методы матричной факторизации для систем рекомендации // Информационные технологии и системы 2020 (ИТС 2020): материалы международной научной конференции. С. 193–194. Минск: БГУИР, 2020. [Электронный ресурс]: https://libeldoc.bsuir.by/bitstream/123456789/41339/1/Moysyuk_Dranko_Metody.pdf (дата обращения 22.07.2024).

19. Кузнецов И.А. Методы и алгоритмы машинного обучения для предобработки и классификации слабоструктурированных текстовых данных в научных рекомендательных системах. Дис. ... канд. техн. наук / ФГАОУ ВО НИЯУ «МИФИ». Москва, 2019. [Электронный ресурс]: https://ds.mephi.ru/documents/90/Кузнецов_И_А_Текст_диссертации.pdf (дата обращения 22.07.2024).

20. Yuan M., Lin H.-T., Boyd-Graber J. Cold-start active learning through self-supervised language modeling // arXiv:2010.09535. 2020. https://doi.org/10.48550/arXiv.2010.09535

21. RMSE is not enough: Guidelines to robust data-model comparisons for magnetospheric physics / M.W. Liemohn [et al.] // Journal of Atmospheric and Solar-Terrestrial Physics. 2021. Vol. 218. Article 105624. https://doi.org/10.1016/j.jastp.2021.105624

22. GigaChat API // Сбер. [Электронный ресурс]: https://developers.sber.ru/portal/products/gigachat-api (дата обращения 20.07.2024).

23. Промпт-инжиниринг // Сбер. [Электронный ресурс]: https://developers.sber.ru/docs/ru/gigachat/prompt-engineering (дата обращения 26.12.2024).

24. Ameницкий А.В., Рухович И.В., Ameницкая Л.А., Ameницкий Д.А. Причины, этические проблемы и профилактика галлюцинации LLM // Интеллект. Сборник статей Международного конкурса молодых ученых. Пенза, 2024. С. 12–15.

25. Платформа разметки данных TagMe // Сбер. [Электронный ресурс]: https://developers.sber.ru/portal/products/tagme (дата обращения 09.12.2024).

Об авторах

Пашигорев Кирилл Игоревич

руководитель направления, Департамент управления данными, ПАО Сбербанк, Россия, 117105, г. Москва, Варшавское шоссе, 25А стр. 6;

E-mail: kipashigorev@sberbank.ru

ORCID: 0009-0008-3478-4874

Резников Андрей Олегович

главный инженер по разработке, Департамент управления данными, ПАО Сбербанк, Россия, 117105, г. Москва, Варшавское шоссе, 25А стр. 6;

E-mail: aoreznikov@sberbank.ru

Recommendation system model based on technical events

Kirill I. Pashigorev
E-mail: kipashigorev@sberbank.ru

Andrei O. Reznikov
E-mail: aoreznikov@sberbank.ru

PJSC Sberbank, Moscow, Russia

Abstract

Recommendation systems are widely used in the commercial field. The algorithms and architectures of recommendation systems are similar in various fields of application and have proven their effectiveness. Recommendations are based on the user's profile, the manner of his behavior on various IT (Information Technology) resources, as well as on similar users. At the same time, the use of recommendation systems in specialized areas is not widespread. Technology divisions are a promising new area of application for recommendation systems, and IT experts themselves will be the users. The purpose of this article is to consider a combination of a recommendation system, machine learning (ML) and LLM (Large Language Model) and to design these tools in a single system. Data volumes are currently measured in petabytes (10^{15} bytes) and exabytes (10^{18} bytes). In order to process even technical information (metadata/technodata) from the surrounding IT landscape, from the IT systems used by experts, AI (Artificial Intelligence) agents are needed. This article provides a literature review regarding the use of recommendation systems in combination with LLM applications, and suggests an application architecture model that generates human-readable news from technical event logs. The system is designed for a group of users who work with big data (ML engineers, data analysts, and data researchers). It is a combination of recommendation system technologies, LLM, and machine learning models. The article also provides the first results of the research that was carried out.

Keywords: recommendation system, matrix factorization, prompt engineering, LLM, AI agents

Citation: Pashigorev K.I., Reznikov A.O. (2025) Recommendation system model based on technical events. *Business Informatics*, vol. 19, no. 1, pp. 7–21. DOI: 10.17323/2587-814X.2025.1.7.21

References

1. Microsoft (2024) *What are AI agents?* Available at: https://learn.microsoft.com/ru-ru/azure/cloud-adoption-framework/innovate/best-practices/conversational-ai (accessed 20 July 2024).
2. Sber (2024) *A platform for launching autonomous AI agents is presented.* Available at: https://ai.sber.ru/en/post/predstavlena-platforma-dlya-zapuska-avtonomnyh-ai-agentov (accessed 20 July 2024).
3. Shah A., Pasha D., Zadeh E., Konur S. (2022) Automated log analysis and anomaly detection using machine learning. *Frontiers in Artificial Intelligence and Applications*, vol. 358: Fuzzy Systems and Data Mining, pp. 137–147. https://doi.org/10.3233/FAIA220378
4. Chen Z., Liu J., Gu W., et al. (2021) Experience report: Deep learning-based system log analysis for anomaly detection. *arXiv:2107.05908.* https://doi.org/10.48550/arXiv.2107.05908
5. Mokshanov M.V. (2024) The use of artificial intelligence in data analysis: an overview of the current state and future directions. *Universum: technical sciences: electronic scientific journal,* no. 5(122) (in Russian). https://doi.org/10.32743/UniTech.2024.122.5.17513

6. Eremin O.Y. (2023) Methods of implementation of hybrid recommendation systems. *E-Scio*, no. 3(78) (in Russian).

7. Kurennykh A.E., Sudakov V.A. (2022) Approach to the development of hybrid recommendation systems. *Bulletin of Science and Practice*, vol. 8, no. 11 (in Russian).

8. Völske M., Potthast M., Syed S., Stein B. (2017) TL;DR: Mining Reddit to Learn Automatic Summarization. Proceedings of the *Workshop on New Frontiers in Summarization, Copenhagen, Denmark, 2017*, pp. 59–63. Association for Computational Linguistics. https://doi.org/10.18653/v1/W17-4508

9. Pu H., Yang X., Li J., Guo R. (2024) AutoRepo: A general framework for multimodal LLM-based automated construction reporting. *Expert Systems with Applications*, vol. 255, part B, article 124601. https://doi.org/10.1016/j.eswa.2024.124601

10. Sivakumar M., Belle A.B., Shan J., Shahandashti K.K. (2024) Prompting GPT–4 to support automatic safety case generation. *Expert Systems with Applications*, vol. 255, part C, article 124653. https://doi.org/10.1016/j.eswa.2024.124653

11. Kim S., Kang H., Choi S., et al. (2024) Large Language Models meet Collaborative Filtering: An efficient all-round LLM-based recommender system. *arXiv:2404.11343*. https://doi.org/10.48550/arXiv.2404.11343

12. Shu Y., Zhang H., Gu H., et al. (2023) RAH! RecSys-Assistant-Human: A human-centered recommendation framework with LLM agents. *arXiv:2308.09904*. https://doi.org/10.48550/arXiv.2308.09904

13. Vaswani A., Shazeer N., Parmar N., et al. (2017) Attention is all you need. *arXiv:1706.03762*. https://doi.org/10.48550/arXiv.1706.03762

14. Morozevich E.S., Korotkov V.S., Kuznetsova E.A. (2022) Development of a model for the formation of individual educational trajectories using machine learning methods. *Business Informatics*, vol. 16, no. 2, pp. 21–35. https://doi.org/10.17323/2587-814X.2022.2.21.35

15. Palchunov D.E., Yakobson A.A. (2024) Development of an intelligent assistant for the selection of goods in the process of dialogue with the user. *Business Informatics*, vol. 18, no. 1, pp. 7–21. https://doi.org/10.17323/2587-814X.2024.1.7.21

16. Amenitsky A.V., Rukhovich I.V., Amenitskaya L.A., et al. (2024) Side effects of hallucinations of artificial intelligence. *Science, innovation, education: current issues and modern aspects*, pp. 224–235. Penza, 2024 (in Russian).

17. Strömqvist Z. (2018) *Matrix factorization in recommender systems: How sensitive are matrix factorization models to sparsity?* Uppsala University Publications. Available at: https://uu.diva-portal.org/smash/get/diva2:1214390/FULLTEXT01.pdf (accessed 22 July 2024).

18. Moisyuk-Dranko P.A., Revotyuk M.P. (2020) Methods of matrix factorization for recommendation systems. Proceedings of the international scientific conference *Information technologies and systems 2020 (ITS 2020)*, pp. 193–194. Minsk: BGUIR (in Russian). Available at: https://libeldoc.bsuir.by/bitstream/123456789/41339/1/Moysyuk_Dranko_Metody.pdf (accessed 22 July 2024).

19. Kuznetsov I.A. (2019) *Methods and algorithms of machine learning for preprocessing and classification of weakly structured text data in scientific recommendation systems*. Moscow: MEPhI (in Russian). Available at: https://ds.mephi.ru/documents/90/Кузнецов_И_А_Текст_диссертации.pdf (accessed 22 July 2024).

20. Yuan M., Lin H.-T., Boyd-Graber J. (2020) Cold-start active learning through self-supervised language modeling. *arXiv:2010.09535*. https://doi.org/10.48550/arXiv.2010.09535

21. Liemohn M.W., Shane A.D., Azari A.R., et al. (2021) RMSE is not enough: Guidelines to robust data-model comparisons for magnetospheric physics. *Journal of Atmospheric and Solar-Terrestrial Physics*, vol. 218, article 105624. https://doi.org/10.1016/j.jastp.2021.105624

22. Sber (2024) *GigaChat API* (in Russian). Available at: https://developers.sber.ru/portal/products/gigachat-api (accessed 22 July 2024).

23. Sber (2024) *Prompt engineering* (in Russian). Available at: https://developers.sber.ru/docs/ru/gigachat/prompt-engineering (accessed 26 December 2024).

24. Amenitsky A.V., Rukhovich I.V., Amenitskaya L.A., Amenitsky D.A. (2024) Causes, ethical problems and prevention of hallucination LLM. *Intelligence. Collection of articles of the International Competition of Young Scientists*. Penza, pp. 12–15 (in Russian).

25. Sber (2024) *TagMe Data Markup Platform* (in Russian). Available at: https://developers.sber.ru/portal/products/tagme (accessed 09 December 2024).

About the authors

Kirill I. Pashigorev

Head of the Department, SberData, PJSC Sberbank, 25A bld. 6, Warsaw Highway, Moscow 117105, Russia;

E-mail: kipashigorev@sberbank.ru

ORCID: 0009-0008-3478-4874

Andrei O. Reznikov

Chief Development Engineer, SberData, PJSC Sberbank, 25A bld. 6, Warsaw Highway, Moscow 117105, Russia;

E-mail: aoreznikov@sberbank.ru