DOI: 10.17323/2587-814X.2025.2.41.53

Аватар-модель покупателя на сетях Колмогорова-Арнольда

Ф.В. Краснов
E-mail: krasnov.fedor2@rwb.ru

Ф.И. Курушин
E-mail: kurushin.fedor@rwb.ru

Исследовательский центр ООО «Вайлдберриз СК» на базе Инновационного центра «Сколково», Москва, Россия

Аннотация

В условиях стремительного развития электронной торговли перед специалистами встают новые задачи по персонализации поиска и рекомендаций товаров. Существующие монолитные системы поиска и рекомендаций становятся слишком громоздкими и не могут обеспечить должного уровня понимания пользователей электронной торговой площадки (ЭТП), несмотря на доступ к полной информации об их интересах и истории покупок. Широко используемые механизмы коллаборативной фильтрации также сталкиваются с проблемами: недостатком разнообразия предложений и низкой способностью удивлять. Кроме того, они характеризуются медленной обновляемостью рекомендаций и подменой «персонального» подхода на «такой же, как у других». Мы предлагаем решение этих проблем путём разработки индивидуального для каждого пользователя ЭТП поискового ассистента «Эллочка». Цифровая аватар-модель пользователя непрерывно осуществляет поиск нужных товаров, исходя из истории взаимодействия пользователя с ЭТП. Мы следуем принципу изолированности – аватар-модели не обмениваются информацией друг о друге. При регистрации нового пользователя ему создаётся новая аватар-модель, которая затем самостоятельно развивается и адаптируется к его предпочтениям и поведению. Каждая аватар-модель имеет свой собственный язык для формирования поисковых запросов, что позволяет более точно учитывать предпочтения и интересы пользователя. Сложность аватар-модели может изменяться в зависимости от интенсивности взаимодействий с ЭТП, что обеспечивает более гибкую и адаптивную систему рекомендаций. Благодаря непрерывности взаимодействий аватар-модели могут отслеживать лучшие условия для приобретения товаров, напоминать об окончании срока годности товара и необходимости повторного приобретения товаров массового спроса. Изолированность аватар-моделей позволяет переобучать их после каждого события без значительного влияния на общую систему поиска и рекомендаций. Применение нейросетевых структур и сетей Колмогорова-Арнольда в аватар-моделях позволяет улучшить основные показатели эффективности поиска и рекомендаций, такие как новизна и разнообразие.

Ключевые слова: большие языковые модели, поиск продуктов, рекомендации поисковых запросов, преобразование поисковых запросов, определение намерений пользователей, анализ текста, машинное обучение, электронная коммерция

Цитирование: Краснов Ф.В., Курушин Ф.И. Аватар-модель покупателя на сетях Колмогорова-Арнольда // Бизнес-информатика. 2025. Т. 19. № 2. С. 41–53. DOI: 10.17323/2587-814X.2025.2.41.53

Введение

Искусственный интеллект всё больше проникает в научную методологию и повседневную жизнь. С одной стороны, пересматриваются подходы к классификации научных знаний, например, в ГРНТИ (государственный рубрикатор научно-технической информации) появился раздел «28.23: искусственный интеллект». С другой стороны, в интернет-магазине женской одежды наукоёмкость процессов сравнима с камеральными работами в геологии (процесс обработки и анализа геологических данных, полученных в результате проведения полевых работ) [1].

В данной работе представлена аватар-модель персонального помощника в совершении покупок на электронных торговых площадках (ЭТП), получившая название «Эллочка» – в честь героини произведения Ильфа и Петрова «12 стульев». Напомним, в этом романе Эллочка – красивая и избалованная молодая женщина, живущая в свое удовольствие за счет мужа, занятая в основном приобретением вещей.

Персональные помощники, основанные на искусственных нейронных сетях глубокого обучения, стали привычным атрибутом клиентского обслуживания в транспортной сфере, банковской отрасли и электронной торговле. Однако «персональность» такого помощника ограничивается лишь временным стремлением сэкономить ресурсы в процессе сеанса обслуживания. Личные предпочтения и настроения пользователя никак не влияют на работу системы. Персональный помощник не сможет продолжить прерванный разговор с того же места и, скорее всего, забудет его начало.

В основе персональных помощников и моделей поиска и рекомендаций в электронной коммерции лежат большие языковые модели (англ. Large Language Model, LLM), основанные на архитектуре трансформеров [2].

Однако каждая большая языковая модель содержит файл со словарём, который она использует далее в нейросетевых структурах уже в виде порядковых номеров токенов из словаря. Это связано с тем, что линейная алгебра работает с моделями чисел, а не со строковыми переменными.

Размер словаря LLM во многом определяет количество параметров модели, так как он является одной из основных размерностей LLM наряду с размером окна восприятия модели. В *таблице 1* приведены размеры словарей наиболее результативных LLM согласно эталонному тесту MTEB (Massive Text Embedding Benchmark) [3].

Системы поиска и рекомендаций представляют собой высоконадёжные информационные системы, которые обеспечивают непрерывность процесса обслуживания и минимальные задержки при взаимодействии с покупателями и продавцами.

Быстродействие в обслуживании поисковых запросов достигается за счёт горизонтального мас-

Таблица 1.

Показатели отраслевых исследований

Модель	Размер словаря	Количество русских слов
alan-turing-institute/mt5-large-finetuned-mnli-xtreme-xnli	250100	26427
ai-forever/rugpt3large_based_on_gpt2	50257	43213
aeonium/Aeonium-v1-Base-4B	128000	102913

штабирования вычислительных ресурсов. В рамках одной сессии обслуживания пользователя не закрепляется определённый вычислительный ресурс. Это приводит к необходимости синхронизации пользовательских действий, которые изменяют, например, баланс, остатки на складе или содержимое корзины покупателя. Эти цифровые объекты существуют в единственном числе и управляются централизованно.

Основной вклад данного исследования состоит в формулировании и апробации нового, децентрализованного подхода к построению систем поиска и рекомендаций в электронной коммерции, ставящем в центр аватар-модель покупателя (*рис. 1*).

Вторым по важности вкладом данного исследования является расширение подхода к построению искусственных сетей глубокого обучения для обработки временных рядов новым классом сетей Колмогорова–Арнольда с использованием вейвлетных преобразований.

1. Методика

Для того чтобы избежать общеизвестных проблем с новизной и разнообразием в товарных рекомендациях [4–7], в данном исследовании предлагается рассматривать рекомендации поисковых запросов, приводящих к нужным товарам. Выражаясь бытовым языком, рекомендовать как искать товар, а не пытаться угадать искомый товар. Аналогичный подход к рекомендациям с использованием текста предложен в работе [8], в которой авторы получают существенное улучшение метрик за счёт отказа от рекомендаций товаров по их номеру, как в работах [9, 10].

Другой известной проблемой генеративных моделей на основе LLM является склонность к галлюцинациям, когда нейросеть генерирует «придуманную» ложную информацию. Для того чтобы побороть негативные эффекты от галлюцинирования, применяются различные методики регуляризации вывода модели. Например, отражение модели на саму себя [11] или использование адаптивной спектральной нормализации [12]. Авторы настоящего исследования снизили негативный эффект от галлюцинирования за счёт того, что создаваемый текст поискового запроса предназначен не для человека, а для системы поиска, другими словами, для другой модели машинного обучения. Что позволило снизить требования к лексике и связанности генерируемого текста поискового запроса, так как наличие токенов достаточно для корректного семантического поиска по товарам, в свою очередь несогласованность токенов по падежам или повторение влияет на релевантность поиска незначительно.

Методически аватар-модель построена на основании парадигмы информационного поиска – композиции кандидатной и ранжирующей моделей [13]. Авторы настоящего исследования развили этот подход и вместо ранжирующей модели создали модель отбора пар запросов кандидатов с фиксированными пятью уровнями отношений: «сужение», «расширение», «перефразирование», «разные характеристики», «сопутствующие товары». Что дало возможность создавать более сфокусированные на покупателя сценарии взаимодействия в зависимости от наличия или отсутствия покупок в сессии взаимодействия с ЭТП.

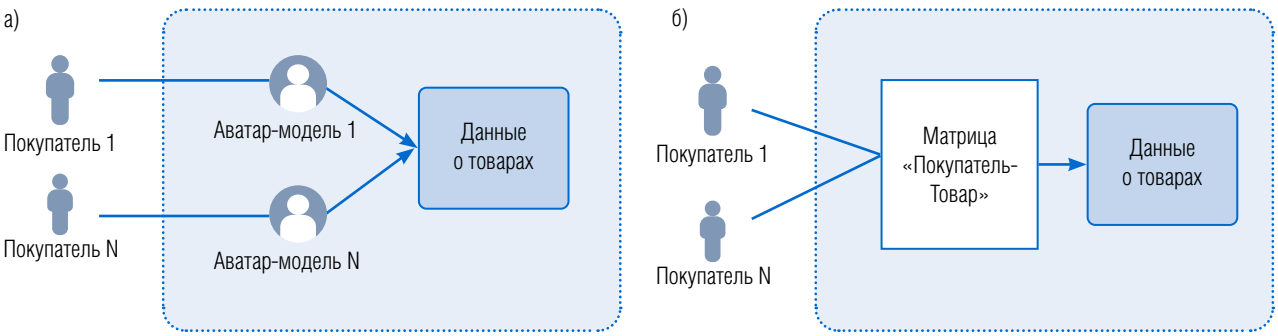

Рис. 1. Схема работы рекомендаций:
а) децентрализованная (предлагается в данном исследовании); б) централизованная.

Современные модели токенизации в LLM построены на сжатых словарях из субсловарных *n*-грамм [14, 15]. Такой подход решает проблему отсутствия слова в словаре и уменьшает потребление памяти, но токенизация производится неоднозначно – одно и то же предложение может быть токенизировано на различные наборы токенов. Это свойство используется при обучении LLM, как бутстрэппинг для более равномерного распределения обратного распространения ошибки. Для аватар-модели нет необходимости в решении проблемы токенизации новых слов, так как аватар-модель использует все токены из словарного запаса данного пользователя. Более того, аватар-модель использует словарные *n*-граммы длинной до четырех слов, чтобы получать события от структур токенов. Например, от названий торговых марок, состоящих из нескольких слов, таких как «Рот Фронт». Очевидно, что на поисковый запрос «Рот Фронт» не стоит искать товары с токенами «рот» и «фронт».

При обновлении аватар-модели модель токенизации создаётся заново, так как аватар-модель обладает в 103 раза меньшим количеством параметров для обучения чем, например, модель GPT2 (120 млн).

2. Сети Колмогорова–Арнольда

Сети Колмогорова–Арнольда (англ. Kolmogorov–Arnold Networks, KAN) демонстрируют многообещающие результаты в различных генеративных моделях [16] и моделях временных рядов [17]. Покажем преимущества KAN по сравнению с многослойным перцептроном Румельхарта (англ. Multilayer Perceptron, MLP). Пусть дан MLP с размерностью n на входе и m на выходе, содержащий полносвязанные слои искусственной нейронной сети от 1 до $l + 1$. Тогда уравнение MLP в матричной форме задается формулой (1):

$$x^{(l+1)} = W_{l+1,l}\,x^{(l)} + b^{(l+1)}, \tag{1}$$

где $W_{l+1,l}$ – матрица весов, соединяющая слой l и слой $l + 1$;

$x^{(l)}$ – входящий вектор;

$x^{(l+1)}$ – исходящий вектор;

$b^{(l+1)}$ – смещение для слоя $l + 1$.

Матрица весов $W_{l+1,l}$ может быть представлена следующим образом:

$$W_{l+1,l} = \begin{pmatrix} w_{1,1} & w_{1,2} & \cdots & w_{1,n} \\ w_{2,1} & w_{2,2} & \cdots & w_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ w_{m,1} & w_{m,2} & \cdots & w_{m,n} \end{pmatrix}, \tag{2}$$

где $w_{i,j}$ – это вес между i-ым узлом в $l + 1$-ом слое и j-ым узлом в l-ом слое ($i = 1, 2, ..., m$ и $j = 1, 2, ..., n$).

Тогда вектор смещения $b^{(l+1)}$ может быть представлен в следующем виде:

$$b^{(l+1)} = \begin{pmatrix} b_1^{(l+1)} \\ b_2^{(l+1)} \\ \vdots \\ b_m^{(l+1)} \end{pmatrix}. \tag{3}$$

Таким образом, в развёрнутом виде уравнение (1) выглядит следующим образом:

$$\begin{pmatrix} x_1^{(l+1)} \\ x_2^{(l+1)} \\ \vdots \\ x_m^{(l+1)} \end{pmatrix} = \begin{pmatrix} w_{1,1} & w_{1,2} & \cdots & w_{1,n} \\ w_{2,1} & w_{2,2} & \cdots & w_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ w_{m,1} & w_{m,2} & \cdots & w_{m,n} \end{pmatrix} \begin{pmatrix} x_1^{(l)} \\ x_2^{(l)} \\ \vdots \\ x_n^{(l)} \end{pmatrix} + \begin{pmatrix} b_1^{(l+1)} \\ b_2^{(l+1)} \\ \vdots \\ b_m^{(l+1)} \end{pmatrix}. \tag{4}$$

Теперь предположим, что у нас есть L слоёв, каждый из которых имеет структуру (4). Пусть $\sigma(\cdot)$ это функция активации. Тогда компактная формула для всей сети MLP $f(x)$, где x – входной вектор, а $f(\cdot)$ – это MLP, задаётся следующим образом (5):

$$f(x) = x^{(L)}, \tag{5}$$

где

$$x^{(l+1)} = \sigma(W_{l+1,l}\,x^{(l)} + b^{(l+1)}), \tag{6}$$

для $l = 0, 1, 2, ..., L - 1$, а $x^{(l+1)}$ обозначает вектор на выходе.

Что эквивалентно следующему представлению (7):

$$f(x) = \sigma\begin{pmatrix} W_L\sigma(W_{L-1}\cdots\sigma(W_2\sigma(W_1x + b_1) + \\ + b_2)\cdots + b_{L-1}) + b_L \end{pmatrix}. \tag{7}$$

Теперь рассмотрим, как создаются отношения между слоями в KAN. Пусть $x^{(l)}$ – вектор с размерностью n. Тогда транспонированный вектор можно представить следующим образом:

$$x^{(l)} \in R^n \Rightarrow (x^{(l)})^{\mathrm{T}} \in R^{1\times n}.$$

Обозначим матрицу транспонированных векторов $x^{(l)}$ с размерностью m строк и n столбцов, как X_l:

$$X_l = \begin{pmatrix} (x^{(l)})^{\mathrm{T}} \\ (x^{(l)})^{\mathrm{T}} \\ \vdots \\ (x^{(l)})^{\mathrm{T}} \end{pmatrix} \in R^{m\times n}.$$

Каждая строка матрицы X_l – это транспонированный вектор $(x^{(l)})^{\mathrm{T}}$. Теперь определим оператор A_o, который действует на матрицу $\Phi_{l+1,l}(X_l)$. Этот оператор суммирует элементы каждой строки матрицы и выводит результирующий вектор r. Формула определения A_o выглядит следующим образом:

$$A_o(\Phi_{l+1,l}(X_l)) = r,$$

где r – вектор, получаемый из следующего выражения (8):

$$r_i = \sum_j \left[\Phi_{l+1,l}\left(X_l\right)\right]_{ij} = \sum_{j=1}^{n} \phi_{i,j}\left(x_j^{(l)}\right), \text{для } i = 1,2,\ldots,m. \quad (8)$$

В выражении (8), $\left[\Phi_{l+1,l}\left(X_l\right)\right]_{ij}$ соответствует элементу в i-й строке и j-ом столбце матрицы $\Phi_{l+1,l}(X_l)$.

Таким образом, оператор A_o может быть записан в следующем виде:

$$A_o\left(\Phi_{l+1,l}\left(X_l\right)\right) = \sum_j \left[\Phi_{l+1,l}\left(X_l\right)\right]_{ij}.$$

По определению A_o осуществляет действие над матрицей $\Phi_{l+1,l}(X_l)$, суммирует элементы в каждой строке и получает на выходе результирующий вектор r. Действительно, $\Phi_{l+1,l}$ получает на входе вектор $x^{(l)}$ и выдает на выходе данные, где каждый элемент из $x^{(l)}$ является суммой в виде одного элемента (9):

$$X_{l+1,l} = \Phi_{l+1,l}(X_l), \quad (9)$$

где:

$$\Phi_{l+1,l}\left(X_l\right) = \begin{pmatrix} \phi_{1,1}(x_1^{(l)}) & \phi_{1,2}(x_2^{(l)}) & \cdots & \phi_{1,n}(x_n^{(l)}) \\ \phi_{2,1}(x_1^{(l)}) & \phi_{2,2}(x_2^{(l)}) & \cdots & \phi_{2,n}(x_n^{(l)}) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_{m,1}(x_1^{(l)}) & \phi_{m,2}(x_2^{(l)}) & \cdots & \phi_{m,n}(x_n^{(l)}) \end{pmatrix}. \quad (10)$$

В выражении (10) $\Phi_{l+1,l}$ является функцией активации между слоями l и $l + 1$. Каждый элемент $\phi_{i,j}(\cdot)$ обозначает функцию активации, которая соединяет j-й нейрон в слое l с i-м нейроном в слое $l + 1$. Вместо умножения уравнение (10) вычисляет функцию с обучаемыми параметрами. Следовательно, если X_o рассматривать как входные данные, которые содержат только входной вектор в виде строк, то для всей KAN выходные данные после L слоёв будут следующими (11):

$$f_{KAN}\left(X_0\right) = x^{(L)} = A_o\left(\Phi_{L,L-1}\left(X_{L-1}\right)\right) =$$

$$= A_o\left(\Phi_{L,L-1}\left(\begin{pmatrix} \left(A_o\left(\Phi_{L-1,L-2}\left(X_{L-2}\right)\right)\right)^{\mathrm{T}} \\ \left(A_o\left(\Phi_{L-1,L-2}\left(X_{L-2}\right)\right)\right)^{\mathrm{T}} \\ \vdots \\ \left(A_o\left(\Phi_{L-1,L-2}\left(X_{L-2}\right)\right)\right)^{\mathrm{T}} \end{pmatrix}\right)\right) = \quad (11)$$

$$= A_o\left(\Phi_{L,L-1}\left(\begin{pmatrix} \left(A_o\left(\Phi_{L-1,L-2}\cdots\left(A_o\left(\Phi_{1,0}\left(X_0\right)\right)\right)\right)\right)^{\mathrm{T}} \\ \left(A_o\left(\Phi_{L-1,L-2}\cdots\left(A_o\left(\Phi_{1,0}\left(X_0\right)\right)\right)\right)\right)^{\mathrm{T}} \\ \vdots \\ \left(A_o\left(\Phi_{L-1,L-2}\cdots\left(A_o\left(\Phi_{1,0}\left(X_0\right)\right)\right)\right)\right)^{\mathrm{T}} \end{pmatrix}\right)\right)$$

Таким образом, традиционные MLP используют фиксированные нелинейные функции активации в каждом узле, линейные веса и смещения для преобразования входных данных по слоям. Выходные данные на каждом слое вычисляются с помощью линейного преобразования, за которым следует фиксированная функция активации. Во время обратного распространения ошибки вычисляются градиенты функции потерь относительно весов и смещений для обновления параметров модели. В отличие от этого, KAN заменяют линейные веса одномерными функциями, которые можно обучать, размещая на рёбрах, а не на узлах. В качестве одномерных функций наиболее часто используют вейвлетные преобразования: MHAT-вейвлет («Мексиканская шляпа»), вейвлет Шеннона, вейвлет Морле, вейвлет Гаусса (*таблица 2*).

В узлах производится суммирование одномерных функции из предыдущих уровней. Каждая функция может быть адаптирована, что позволяет KAN обучать как активацию, так и преобразование входных данных. Это изменение приводит к повышению точности и интерпретируемости, поскольку KAN могут лучше аппроксимировать функции с использованием меньшего количества параметров. Во время обратного распространения ошибки в KAN градиенты вычисляются относительно одномерных функций, обновляя их, чтобы минимизировать функцию потерь. Это приводит к более эффективному обучению для сложных и многомерных функций.

Таблица 2.

Вейвлеты, используемые в качестве одномерных функций

Название	Формула
MHAT-вейвлет («Мексиканская шляпа»)	$\psi(t)=\frac{-d^2}{dt^2}e^{-t^2/2}=\left(1-t^2\right)e^{-t^2/2}$
Вейвлет Шеннона	$\psi_k^n(t):=2^{n/2}\psi^{(Sha)}\left(2^n t-k\right)$, где $\psi^{(Sha)}(t):=\frac{\sin\left(\pi t\right)}{\pi t}$
Вейвлет Морле	$\psi_\sigma(t)=c_\sigma\pi^{-1/4}e^{-t^2/2}\left(e^{i\sigma t}-k_\sigma\right)$, где $k_\sigma=e^{-\sigma^2/2}, c_\sigma=\left(1+e^{-\sigma^2}-2e^{-3\sigma^2/4}\right)^{-\frac{1}{2}}$
Вейвлет Гаусса	$\psi(t)=\frac{-d}{dt}e^{-t^2/2}$

3. Модель сокращения разнообразия поисковых запросов

В сфере электронной коммерции существует проблема «длинного хвоста» поисковых запросов, которая затрудняет создание эффективных моделей. Это связано с большим разнообразием запросов, опечатками, синонимами и сленгом.

Современные подходы для переформулировки запросов используют нейросетевые методы. С помощью выделения текстовых факторов строятся векторные представления поисковых запросов, а затем осуществляется поиск «ближайших соседей» в векторном пространстве с использованием Retrieval Augmented Generation (RAG) языковых моделей [18] и моделей трансформеров BERT [19].

Поиск по товарам позволяет использовать поведенческие данные пользователей от совершения покупок. Таким образом, используется коллаборативная парадигма: по связи «пользователь-товар» определяют последовательности, приведшие к покупке, и используют их как варианты для текущего пользователя [20]. В настоящем исследовании задача переформулирования запросов (ПЗ) рассмотрена с позиции связи «запрос-товар» для применения суррогатной функции [22] в аватар-модели.

Помимо переформирования запросов также ставится задача определения типа связи «запрос-запрос» и «запрос-товар». В качестве примера, демонстрирующего различные типы связей, выделенные авторами, приведем *таблицу 3.*

Таблица 3.

Примеры запросов, демонстрирующих различные типы связей

Исходный запрос	Замена	Сценарий
Iphone 16 pro max	Apple iPhone 16 Pro Max	Перефразированные
Платье женское красное	Платье женское	Расширение
Платье	Платье женское красное	Сужение
Платье женское красное	Платье женское вечернее	Другие характеристики
Платье женское вечернее	Туфли на каблуке	Заменители
Туфли на каблуке	Кроссовки для бега	Нерелевантное

Для более наглядного представления, обозначим множество всех поисковых запросов как Q. Приблизительное количество элементов в этом множестве составляет $Q_v \approx 10^{10}$, что сопоставимо с потоком событий, анализируемых при поиске бозона Хиггса [21].

Из исследования [23] известно, что 98% всех покупок совершается с использованием ограниченного множества запросов, которые мы обозначим как Q_{HPQ}. Без ограничения общности можно считать, что Q_{HPQ} не зависит от времени. Оставшееся множество запросов обозначим как Q_{LPQ}, где HPQ и LPQ – общепринятые сокращения для high-performing и low-performing запросов соответственно.

Исходя из определения множества Q, можем записать следующие уравнения:

$$Q = Q_{HPQ} \cup Q_{LPQ}, \quad (12)$$

$$\varnothing = Q_{HPQ} \cap Q_{LPQ}. \quad (13)$$

Из характера распределения Q следует, что $|Q_{LPQ}| \gg |Q_{HPQ}|$. Задача переформулировки запросов тогда может быть рассмотрена как поиск функции F такой, что $F(Q_{LPQ}) \Rightarrow Q_{HPQ}$. Рассмотрим запросы $q^{p_i}_{HPQ} \in Q_{HPQ}$ и $q^{p_i}_{LPQ} \in Q_{LPQ}$ приводящие к покупкам продукта $p_i \in P$ из всего каталога продуктов P. Тогда можем записать, что есть суррогатная функция F, которая приводит $q^{p_i}_{LPQ}$ $q^{p_i}_{HPQ}$ для продукта p_i

$$F\left(Q^{p_i}_{LPQ}\right) \Rightarrow Q^{p_i}_{HPQ}, \text{ для } \forall p_i \in P. \quad (14)$$

Выражение (14) позволяет сформулировать условия получения набора данных для использования численных методов получения F, например с помощью искусственных нейронных сетей глубокого обучения. Для этого необходимо собрать пары $\left\{q^{p_i}_{LPQ}, q^{p_i}_{HPQ}\right\}$ из журналов поиска пользователей.

Для формирования негативных примеров был использован подход Negative Batch Sampling [10], при этом количество негативных примеров варьировалось от 5 до 12. Использование большего количества негативных примеров улучшало показатели модели не значительно, но требовало существенно больших ресурсов для обучения модели.

На *рисунке 2* изображена схема аватар-модели в виде последовательности моделей токенизации, модели переформулирования запросов (ПЗ) и KAN.

4. Исследовательские вопросы

Данное исследование носит прикладной характер. Изложенная методика служит источником для исследовательских вопросов и проверок с помощью численных методов. При создании методики авторы сформулировали следующие исследовательские вопросы:

ИВ-1: Каково распределение размеров словаря, используемого пользователями при формировании поисковых запросов?

ИВ-2: Как количественно улучшает аватар-модель использование KAN по сравнению с MLP?

5. Эксперимент

Для проверки нашей методики мы собрали последовательности поисковых запросов и покупок за год по 92 миллионам случайных покупателей. Получился набор данных D с уникальным ключом «пользователь» и временным рядом [«запрос1», ..., «покупка1», «конец сессии», «запрос N»]. Каждая покупка представлена в виде текста из названия, бренда и характеристик товара. В поисковых запросах исправлены опечатки, добавлены синонимы и сделано отображение низко производительных запросов на высоко производительные. Так для каждого пользователя в D получен упорядоченный по времени текст, состоящий из запросов и

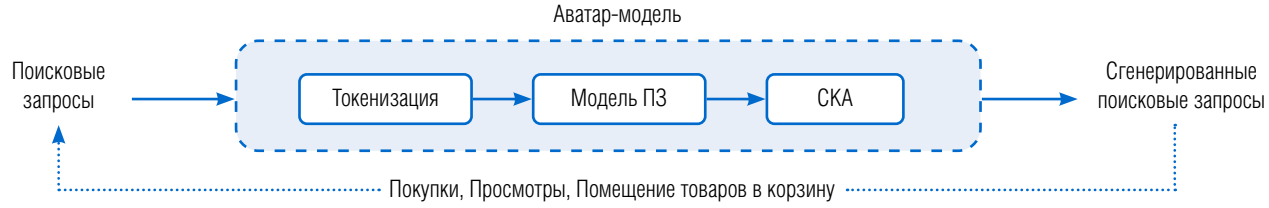

Рис. 2. Схема аватар-модели.

описаний купленных товаров. На основании этого текста производится авторегрессионное обучение аватар-модели каждого пользователя.

При помощи полученной аватар-модели для каждого пользователя была построена прогнозная модель: на основании введённых поисковых запросов прогнозируется следующий поисковый запрос из индекса поисковых запросов. Производится пессимизация на слова из купленных товаров с учётом их удалённости по времени взаимодействия с пользователем. Прогнозируется несколько поисковых запросов для пользователя. Приведём сценарий использования результатов данного исследования: Пользователь заходит на ЭТП, не вводит поисковый запрос, видит поисковую выдачу по товарам, которые ему близки по словарному запасу и при этом не содержат недавних покупок на первых местах. Поисковая выдача отражает текущее состояние каталога товаров, запасов, региона, популярности и других условий, повышающих релевантность. Поисковая выдача обладает высоким разнообразием и новизной, так как содержит товары из нескольких спрогнозированных запросов для пользователя. Выдача не содержит купленные товары на первых позициях.

Для экспериментального ответа на исследовательский вопрос ИВ-1 было построено распределение количества уникальных слов для каждого пользователя из набора данных *D* (*рис. 3*).

Из распределения на *рисунке 3* получаем, что максимальный размер словаря не превышает 1500 слов. По сравнению размерами словаря в LLM, приведённых в *таблице 1*, размер словаря отдельного пользователя меньше на порядок. Этот факт даёт основания для значительного ускорения работы аватар-модели по сравнению с LLM.

Для поиска ответа на исследовательский вопрос ИВ-2 была проведена серия экспериментов по обучению аватар-модели на наборе данных *D*. В качестве гиперпараметров аватар-модели были рассмотрены следующие варианты:

Модель токенизации:

- Unigram, byte pair encoding (BPE), **Word**
- Размер словаря: 100, 1000, **«без ограничений»**
- Количество примеров для субсловарной токенизации: 3, 5, 7, 9, **без сэмплирования**

Модель ПЗ:

- Размерность векторного пространства представления токенов: 32, **64**, 128, 256
- Параметры рекуррентной нейронной сети (RNN):
  - Количество слоев: **2**, 3, 4, 5, 6
  - Двунаправленность: да, **нет**
  - Тип: LSTM, GRU
- In-batch negative sampling: 3, 5, 7, 12
- Dual Margin Cosine Embedding Loss: positive margin **0.9**, negative margin **0.2**
- Общее векторное представление токенов: нет, **да**

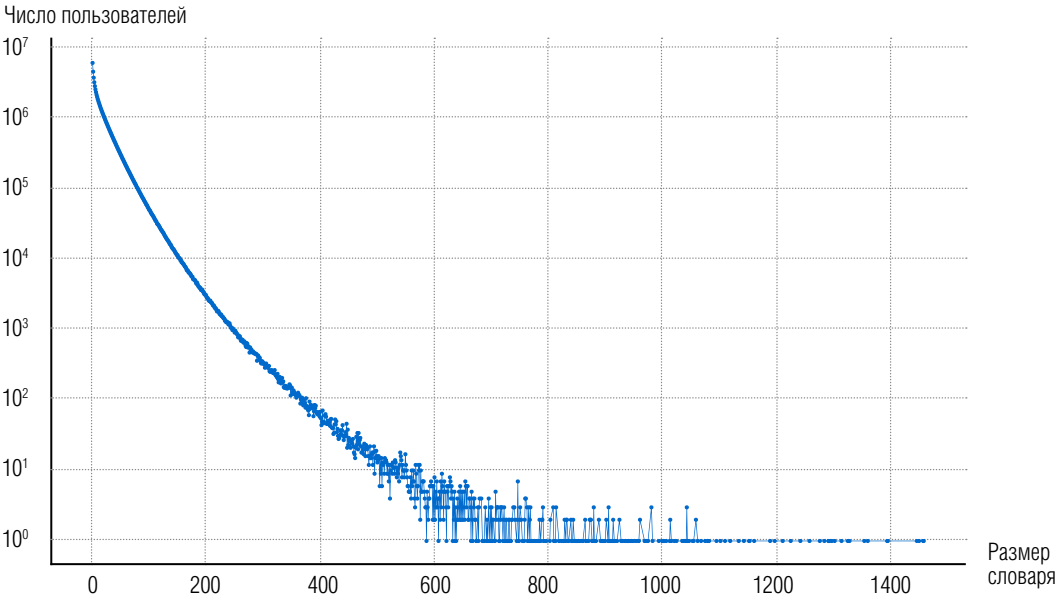

Рис. 3. Распределение количества уникальных слов для каждого пользователя.

Модель KAN:

Тип вейвлета: «**Мексиканская шляпа**», вейвлет Шеннона, вейвлет Морле, вейвлет Гаусса

Всего было проведено 120 экспериментальных сессий, каждая из которых длилась от 12 до 18 часов. В ходе исследования были определены параметры, для которых аватар-модель продемонстрировала наилучшие результаты по скорости сходимости и минимальному значению ошибки обучения.

При всех прочих равных условиях, из распределения, представленного на *рисунке 4*, получен вывод о положительных изменениях в точности при использовании KAN в аватар-модели в сравнении с MLP.

Заключение

В данном исследовании предлагается новый подход к созданию рекомендательной системы для пользователей в сфере электронной коммерции, названный «Эллочка» в честь героини романа «12 стульев» с небольшим словарным запасом, которая успешно справлялась с любыми коммуникационными задачами.

Авторами разработана и протестирована методика, основанная на следующих принципах:

1. Отказ от использования монолитной, единой рекомендательной системы для всех пользователей электронной торговой площадки в пользу создания отдельных рекомендательных моделей для каждого пользователя.

2. Построение языковых моделей для рекомендации текстов поисковых запросов на основе словарей токенов малых размеров вместо огромных словарей с субсловарной токенизацией.

3. Применение математического аппарата сетей Колмогорова-Арнольда для улучшения скорости сходимости моделей при обучении.

Методика, предложенная в статье, успешно прошла апробацию на данных работающей электронной торговой площадки и позволила улучшить автономные показатели рекомендательной системы подсказок. ■

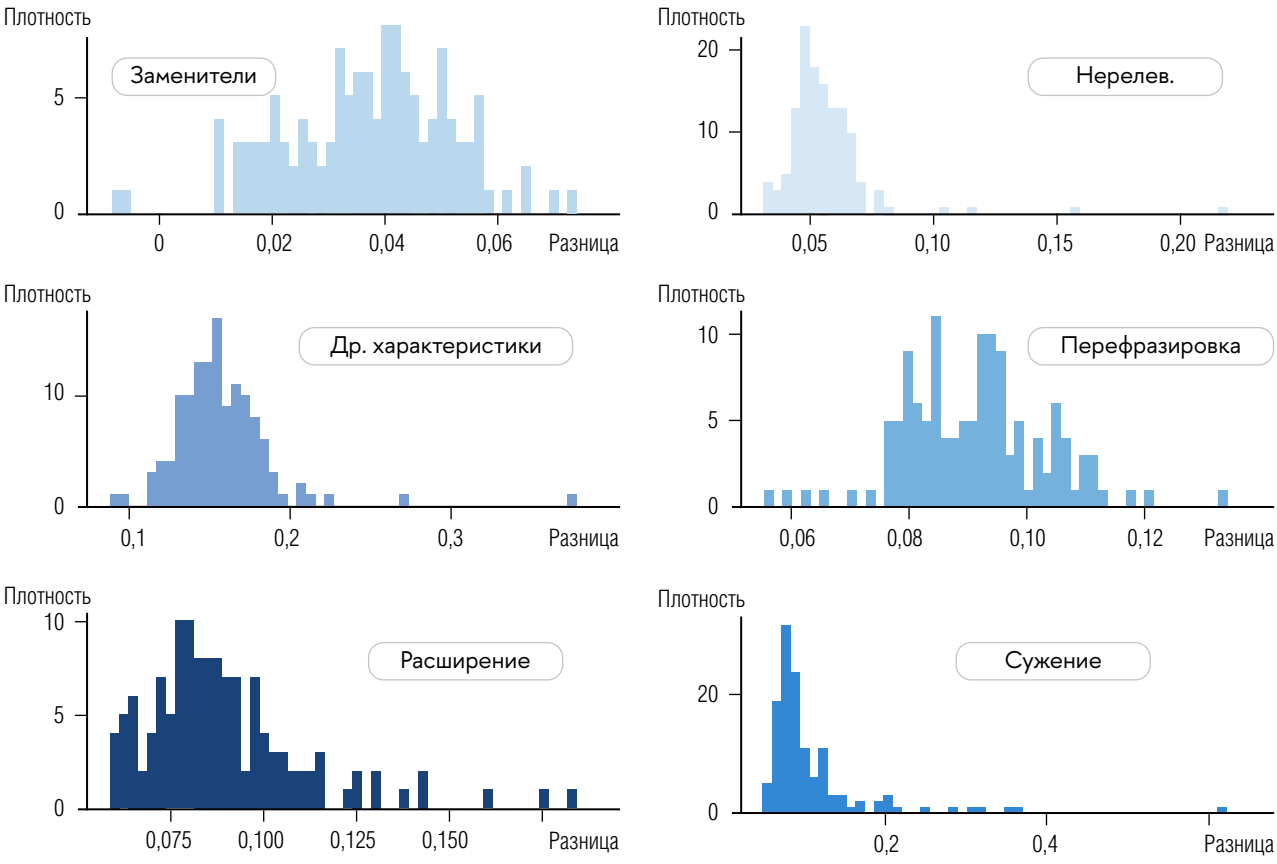

Рис. 4. Распределение разниц в метрике Точность между KAN и MLP.

Литература

1. Буторин А.В., Муртазин Д.Г., Краснов Ф.В. Способ и система прогнозирования эффективных толщин в межскважинном пространстве при построении геологической модели на основе метода кластеризации спектральных кривых // Пат. RU 2718135 C1 Рос. Федерация; заявка 2019128334 от 09.09.2019, опубл. 30.03.2020. Бюл. № 10.
2. Краснов Ф.В. Использование языковых моделей на основании архитектуры трансформеров для понимания поисковых запросов на электронных торговых площадках // International Journal of Open Information Technologies. 2023. Т. 11. № 9. С. 33–40.
3. Muennighoff N., Tazi N., Magne L., Reimers N. MTEB: Massive text embedding benchmark // arXiv:2210.07316. 2022. https://doi.org/10.48550/arXiv.2210.07316
4. Li P., Tuzhilin A. When variety seeking meets unexpectedness: Incorporating variety-seeking behaviors into design of unexpected recommender systems // Information Systems Research. 2023. Vol. 35. No. 3. https://doi.org/10.1287/isre.2021.0053
5. Beyond item dissimilarities: Diversifying by intent in recommender systems / Y. Wang [et al.] //arXiv:2405.12327. 2024. https://doi.org/10.48550/arXiv.2405.12327
6. Castells P., Hurley N., Vargas S. Novelty and diversity in recommender systems // Recommender Systems Handbook (eds. F. Ricci, L. Rokach, B. Shapira). 2015. Springer, New York, NY. P. 603–646. https://doi.org/10.1007/978-1-0716-2197-4_16
7. A hybrid bandit framework for diversified recommendation / Q. Ding [et al.] // Proceedings of the AAAI conference on artificial intelligence. 2021. V. 35. No. 5. P. 4036–4044. https://doi.org/10.1609/aaai.v35i5.16524
8. Text is all you need: Learning language representations for sequential recommendation / J. Li [et al.] // Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. 2023. P. 1258–1267. https:// doi.org/10.1145/3580305.3599519
9. BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer / F. Sun [et al.] // Proceedings of the 28th ACM international conference on information and knowledge management. 2019. P. 1441–1450. https://doi.org/10.1145/3357384.3357895
10. Klenitskiy A., Vasilev A. Turning dross into gold loss: is BERT4Rec really better than SASRec? // Proceedings of the 17th ACM Conference on Recommender Systems. 2023. P. 1120–1125. https://doi.org/10.1145/3604915.3610644
11. Towards mitigating LLM hallucination via self reflection / Z. Ji [et al.] // Findings of the Association for Computational Linguistics: EMNLP 2023. 2023. P. 1827–1843.
12. Егоров Е.А., Рогачев А.И. Исследование влияния адаптивной спектральной нормализации на качество генеративных моделей и стабильность их обучения // Доклады Российской академии наук. Математика, информатика, процессы управления. 2023. Т. 514. № 2. С. 49–59.
13. Extreme multi-label learning for semantic matching in product search / W.C. Chang [et al.] // Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. 2021. P. 2643–2651. https://doi.org/10.1145/3447548.3467092
14. Sennrich R., Haddow B., Birch A. Neural machine translation of rare words with subword units // Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics. 2016. P. 1715–1725. https://doi.org/10.18653/v1/P16-1162
15. Kudo T., Richardson J. SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing // Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations. 2018. P. 66–71.
16. KAN: Kolmogorov-Arnold networks / Z. Liu [et al.] // arXiv:2404.19756. 2024. https://doi.org/10.48550/arXiv.2404.19756
17. Vaca-Rubio C. J., Blanco L., Pereira R., Caus M. Kolmogorov-Arnold networks (KANs) for time series analysis // arXiv:2405.08790. 2024. https://doi.org/10.48550/arXiv.2405.08790
18. Query rewriting for retrieval-augmented Large Language Models / X. Ma [et al.] // arXiv:2305.14283. 2023. https://doi.org/10.48550/arXiv.2305.14283
19. Chen Z., Fan X., Ling Y. Pre-training for query rewriting in a spoken language understanding system // 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2020), Barcelona, Spain. 2020. P. 7969–7973. https://doi.org/10.1109/ICASSP40776.2020.9053531
20. Bhandari M., Wang M., Poliannikov O., Shimizu K. RecQR: Using Recommendation Systems for Query Reformulation to correct unseen errors in spoken dialog systems // 17th ACM Conference on Recommender Systems (RecSys'23), Singapore. 2023. P. 1019–1022.
21. The CDF central analysis farm / T.H. Kim [et al.] // IEEE Transactions on Nuclear Science. 2004. Vol. 51. No. 3. P. 892–896. https://doi.org/10.1109/TNS.2004.829574
22. Кулешов А.П. Когнитивные технологии в адаптивных моделях сложных объектов // Информационные технологии и вычислительные системы. 2008. № 1. С. 18–29.
23. ROSE: Robust caches for Amazon product search / C. Luo [et al.] // Companion Proceedings of the Web Conference 2022. P. 89–93.

Об авторах

Краснов Федор Владимирович

кандидат технических наук;

старший специалист по обработке и анализу данных, сотрудник Исследовательского центра ООО «ВБ СК” на базе Инновационного Центра Сколково, Россия, г. Москва, Западный административный округ, Можайский район, ул. Нобеля, 5;

E-mail: krasnov.fedor2@rwb.ru

ORCID: 0000-0002-9881-7371

Курушин Федор Иванович

аспирант;

специалист по обработке и анализу данных, сотрудник Исследовательского центра ООО «ВБ СК” на базе Инновационного Центра Сколково, Россия, г. Москва, Западный административный округ, Можайский район, ул. Нобеля, 5;

E-mail: kurushin.fedor@rwb.ru

ORCID: 0009-0007-5126-4507

A customer avatar model based on Kolmogorov–Arnold networks

Fedor V. Krasnov
E-mail: krasnov.fedor2@rwb.ru

Fedor I. Kurushin
E-mail: kurushin.fedor@rwb.ru

Research Center of LLC Wildberries SK, Moscow, Russia

Abstract

The increasing pace of development of e-commerce continues to present new challenges in terms of personalizing product search and recommendations. Monolithic search and recommendation systems have become cumbersome and are unable to effectively address the need for a deeper understanding of users on electronic trading platforms (ETPs) despite having access to comprehensive information about their interests and purchase histories. Collaborative filtering mechanisms which are widely used suffer from a lack of diversity in offerings and a reduced capacity to surprise users. Additionally, the low frequency of recommendation updates and the replacement of “personalized” with “similar to others” concepts contribute to these issues. We have approached the resolution of these issues by developing a shopping assistant named “Ellochka” that is individual for each user of ETP. The digital avatar model of the user continually searches for relevant products based on their history of interaction with ETP. We were guided by the principle of independence – avatar models do not share information with each other. When a new user joins, they are assigned a unique avatar model that evolves independently. Each avatar has its own language to generate search queries. The level of complexity of each avatar can vary

depending on the intensity of its interaction with ETP. Continued interaction with the avatar allows for tracking of optimal purchase conditions, reminding users of expiration dates and the need for re-purchasing frequently purchased items. Isolating the avatar allows it to be retrained after each event, without significantly impacting the overall search and recommendation system. The use of neural network architecture-based and Kolmogorov–Arnold networks in the avatar-model has led to improvements in the main indicators of search and recommendation effectiveness, namely, novelty and diversity.

Keywords: large language models, product search, search query recommendations, search query transformation, user intent determination, text analysis, machine learning, e-commerce

Citation: Krasnov F.V., Kurushin F.I. (2025) A customer avatar model based on Kolmogorov–Arnold networks. *Business Informatics*, vol. 19, no. 2, pp. 41–53. DOI: 10.17323/2587-814X.2025.2.41.53

References

1. Butorin A.V., Murtazin D.G., Krasnov F.V. (2020) *Method and system for predicting effective thicknesses in the interwell space when constructing a geological model based on the method of clustering spectral curves.* Patent for invention RU 2718135 C1, 03/30/2020. Application No. 2019128334 dated 09/09/2019.

2. Krasnov F. (2023) Query understanding via Language Models based on transformers for e-commerce. *International Journal of Open Information Technologies*, vol. 11, no. 9, pp. 33–40 (in Russian).

3. Muennighoff N., Tazi N., Magne L., Reimers N. (2022) MTEB: Massive text embedding benchmark. *arXiv:2210.07316*. https://doi.org/10.48550/arXiv.2210.07316

4. Li P., Tuzhilin A. (2023) When variety seeking meets unexpectedness: Incorporating variety-seeking behaviors into design of unexpected recommender systems. *Information Systems Research*, vol. 35, no. 3. https://doi.org/10.1287/isre.2021.0053

5. Wang Y., Banerjee C., Chucri S., et al. (2024) Beyond item dissimilarities: Diversifying by intent in recommender systems. *arXiv:2405.12327*. https://doi.org/10.48550/arXiv.2405.12327

6. Castells P., Hurley N., Vargas S. (2022) Novelty and diversity in recommender systems. *Recommender Systems Handbook* (eds. F. Ricci, L. Rokach, B. Shapira). Springer, New York, NY, pp. 603–646. https://doi.org/10.1007/978-1-0716-2197-4_16

7. Ding Q., Liu Y., Miao C., et al. (2021) A hybrid bandit framework for diversified recommendation. Proceedings of the *AAAI conference on artificial intelligence*, vol. 35, no. 5, pp. 4036–4044. https://doi.org/10.1609/aaai.v35i5.16524

8. Li J., Wang M., Li J., et al. (2023) Text is all you need: Learning language representations for sequential recommendation. Proceedings of the 2*9th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 1258–1267. https:// doi.org/10.1145/3580305.3599519

9. Sun F., Liu J., Wu J., et al. (2019) BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer. Proceedings of the *28th ACM international conference on information and knowledge management*, pp. 1441–1450. https://doi.org/10.1145/3357384.3357895

10. Klenitskiy A., Vasilev A. (2023) Turning dross into gold loss: is BERT4Rec really better than SASRec? Proceedings of the *17th ACM Conference on Recommender Systems*, pp. 1120–1125.

11. Ji Z., Yu T., Xu Y., et al. (2023) Towards mitigating LLM hallucination via self reflection. *Findings of the Association for Computational Linguistics*: EMNLP 2023, pp. 1827–1843.

12. Egorov E.A., Rogachev A.I. (2023) Adaptive spectral normalization for generative models. *Doklady Mathematics*, vol. 108(suppl. 2), pp. S205–S214. https://doi.org/10.1134/S1064562423701089

13. Chang W.C., Jiang D., Yu H.F., et al. (2021) Extreme multi-label learning for semantic matching in product search. Proceedings of the *27th ACM SIGKDD conference on knowledge discovery & data mining*, pp. 2643–2651.

14. Sennrich R., Haddow B., Birch A. (2016) Neural machine translation of rare words with subword units. Proceedings of the *54th Annual Meeting of the Association for Computational Linguistics,* pp. 1715–1725. https://doi.org/10.18653/v1/P16-1162

15. Kudo T., Richardson J. (2018) SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing. Proceedings of the *2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 66–71.

16. Liu Z., Wang Y., Vaidya S., et al. (2024) KAN: Kolmogorov-Arnold networks. *arXiv:2404.19756*. https://doi.org/10.48550/arXiv.2404.19756

17. Vaca-Rubio C. J., Blanco L., Pereira R., Caus M. (2024) Kolmogorov-Arnold networks (KANs) for time series analysis. a*rXiv:2405.08790*. https://doi.org/10.48550/arXiv.2405.08790

18. Ma X., Gong Y., He P., et al. (2023) Query rewriting for retrieval-augmented Large Language Models. *arXiv:2305.14283*. https://doi.org/10.48550/arXiv.2305.14283

19. Chen Z., Fan X., Ling Y. (2020) Pre-training for query rewriting in a spoken language understanding system. *2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2020), Barcelona, Spain*, pp. 7969–7973. https://doi.org/10.1109/ICASSP40776.2020.9053531

20. Bhandari M., Wang M., Poliannikov O., Shimizu K. (2023) RecQR: Using Recommendation Systems for Query Reformulation to correct unseen errors in spoken dialog systems. *17th ACM Conference on Recommender Systems (RecSys'23), Singapore*, pp. 1019–1022.

21. Kim T.H., Neubauer M., Sfiligoi I., et al. (2004) The CDF central analysis farm. *IEEE Transactions on Nuclear Science*, vol. 51, no. 3, pp. 892–896. https://doi.org/10.1109/TNS.2004.829574

22. Kuleshov A.P. (2008) Cognitive technologies in adaptive models of complex objects. *Informatsionnye Tekhnologii i Vychslitel'nye Sistemy*, vol. 1, pp. 18–29.

23. Luo C., Lakshman V., Shrivastavaet A., et al. (2022) ROSE: Robust caches for Amazon product search. *WWW '22: Companion Proceedings of the Web Conference 2022*, pp. 89–93. https://doi.org/10.1145/3487553.3524213

About the authors

Fedor V. Krasnov

Candidate of Sciences (Technology);

Senior Data Scientist, Research Center of WB SK LLC based on the Skolkovo Innovation Center, 5, Nobel St., Mozhaisk District, Western Administrative District, Moscow, Russia;

E-mail: krasnov.fedor2@rwb.ru

ORCID: 0000-0002-9881-7371

Fedor I. Kurushin

Doctoral Student;

Data Scientist, Research Center of WB SK LLC based on the Skolkovo Innovation Center, 5, Nobel St., Mozhaisk District, Western Administrative District, Moscow, Russia;

E-mail: kurushin.fedor@rwb.ru

ORCID: 0009-0007-5126-4507