Министерство науки и высшего образования Российской Федерации

Санкт-Петербургский политехнический университет Петра Великого

Физико-Механический институт

Работа допущена к защите

Руководитель ОП

_____К.Н. Козлов

«___»_____2025 г.

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

РАБОТА БАКАЛАВРА

СИСТЕМА СРАВНЕНИЯ МУЛЬТИМОДАЛЬНЫХ ЯЗЫКОВЫХ МОДЕЛЕЙ

по направлению подготовки (специальности)
01.03.02 Прикладная математика и информатика

Направленность (профиль)
01.03.02_04 Биоинформатика

Выполнил
студент гр. 5030102/10401 А. Д. Мартынов

Руководитель
к. ф.-м. н. В. С. Чуканов

Санкт-Петербург

2025

САНКТ-ПЕТЕРБУРГСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ПЕТРА ВЕЛИКОГО

Физико-механический институт

УТВЕРЖДАЮ
Руководитель образовательной программы «Прикладная математика и информатика»
_____К.Н. Козлов
«____» _____2025 г.

ЗАДАНИЕ

на выполнение выпускной квалификационной работы

студенту Мартынову Александру Дмитриевичу, гр. 5030102/10401

1. Тема работы: «Система сравнения мультимодальных языковых моделей»
2. Срок сдачи студентом законченной работы: июнь 2025 г.
3. Исходные данные по работе:

   Перечень используемых информационных технологий

   1. Язык программирования Python
   2. Среда разработки Visual Studio Code

   Ключевые источники литературы

   1. Yang Liu, Zhi-Ping Fan, Yao Zhang A method for stochastic multiple criteria decision making based on dominance degrees // Information Sciences. – 2011. – Vol. 181. – C. 4139–4153
   2. Лемешко Б. Ю., Лемешко С. Б., Постовалов С.Н., Чимитова Е.В. Статистический анализ данных, моделирование и исследование вероятностных закономерностей. Компьютерный подход. – Новосибирск: Изд-во НГТУ, 2011. – С. 93 – 103.

4. Содержание работы (перечень подлежащих разработке вопросов):
   1. Введение
   2. Обзор методов
   3. Разработка системы ранжирования на основе PROMETHEE
   4. Применение на реальных данных
   5. Анализ полученных результатов
   6. Выводы

5. Дата выдачи задания: 31.01.2025г.

Руководитель ВКР _____ к. ф.-м. н. В.С.Чуканов
(подпись)

Задание принял к исполнению

Студент _____ А.Д. Мартынов
(подпись)

РЕФЕРАТ

Ключевые слова: машинное обучение, языковые модели, статистика
Тема выпускной квалификационной работы: «Система сравнения мультимодальных языковых моделей».

Данная работа посвящена разработке метода ранжирования мультимодальных языковых моделей (Multimodal Language Models, MLLM) с целью повышения точности и объективности их сравнения. Актуальность темы обусловлена возросшим количеством подобных моделей и популяризацией их применения в самых различных областях, что усложняет процесс выбора наилучшего алгоритма в каждом приложении, а также недостатком универсальной системы оценки их качества.

В основе метода лежит анализ метрик качества, которые получают модели на валидационном наборе данных, а также их статистических свойств. Разрабатываемая процедура представляет собой улучшенную версию метода PROMETHEE, что позволяет точнее ранжировать модели с учетом противоречивых критериев качества и учитывает специфику работы с моделями машинного обучения. В ходе экспериментальной оценки показано преимущество такого подхода по сравнению с другими методиками.

Полученные результаты свидетельствуют о возможности применения предложенной системы в реальных задачах с разнотипными данными, что особенно полезно при отсутствии возможности определения относительной значимости каждого критерия.

ABSTRACT

Keywords: machine learning, language models, statistics
Thesis Title: *"*A system for comparing multimodal language models"

This work is devoted to the development of a ranking method for multimodal Language Models (MLLM) in order to improve the accuracy and objectivity of their comparison. The relevance of the topic is due to the increased number of such models and the popularization of their use in various fields, which complicates the process of choosing the best algorithm in each application, as well as the lack of a universal system for evaluating their quality.

The method is based on the analysis of quality metrics obtained by models based on a validation dataset, as well as their statistical properties. The procedure being developed is an improved version of the PROMETHEE method, which makes it possible to more accurately rank models based on conflicting quality criteria and takes into account the specifics of working with machine learning models. The experimental evaluation shows the advantage of this approach in comparison with other methods.

The results obtained indicate the possibility of using the proposed system in real-world problems with different types of data, which is especially useful in the absence of the possibility of determining the relative importance of each criterion.

Содержание

Введение

В последние годы языковые модели искусственного интеллекта показали значительный прогресс и получили широкое применение в самых разнообразных прикладных задачах: от генерации текста и автоматического перевода до поиска информации и помощи в программировании. По мере развития технологии, появилась необходимость расширения возможностей подобных моделей за пределы работы исключительно с текстом, что привело к созданию мультимодальных языковых моделей (Multimodal Language Models, MLLM), способных обрабатывать разные типы (модальности) входных данных: изображения, аудио, видео, временные ряды или их комбинации.

Развитие мультимодальных языковых моделей открыло возможности для их применения в ряде прикладных областей: медицина (рентген-диагностика, анализ истории болезней), транспортные системы, системы видеонаблюдения, анализ документов, интеллектуальные ассистенты и другие. Вместе с этим наблюдается также и значительный рост числа моделей, отличающихся архитектурой, обучающими данными и специализированными функциями. Для каждой задачи может найтись несколько моделей, которые были разработанны специально для нее, что привело к необходимости в систематическом и корректном сравнении и ранжировании таких моделей для конкретной задачи с целью выбора наиболее подходящей.

Однако, сравнение MLLM сопряжено с рядом сложностей. Во-первых, каждая задача имеет свой собственный набор метрик, отражающих различные аспекты их качества. Во-вторых, этим метрики не всегда могут быть согласованы между собой: одна модель может превосходить другие по одному критерию, но уступать им по другому. В-третьих, отсутствие универсального критерия приводит к субъективности выбора наилучшей модели и затрудняет автоматизацию процесса ее поиска.

Таким образом, возникает потребность в общей системе оценки и сравнения MLLM, которая бы учитывала результаты по всем необходимым метрикам, минимизировала влияние человеческого фактора и была независима от конкретной предметной области.

Цель данной работы - разработка такого метода ранжирования MLLM, отвечающего данным требованиям.

Формализация задачи

Для эффективного решения проблемы сравнения MLLM с учетом различных метрик, сначала необходимо определить то, какие данные нам доступны и что должны получить в результате.

Пусть $\mathcal{M} = \{M_1, M_2, \dots, M_n\}$ – множество сравниваемых моделей. В контексте рассматриваемой задачи их все можно считать за функции, задающие отображение

$$M_i: \mathcal{X} \to \mathcal{Y}, i = \overline{1,n}$$

где $\mathcal{X}$ – пространство входных данных для модели, например пар (текст, изображение), $\mathcal{Y}$ – пространство выходных данных. Именно $\mathcal{X}$ и $\mathcal{Y}$ определяют модальность модели, т.е. то, с какими данными она работает.

Датасетом (или валидационной выборкой) $\mathcal{S}$ будем называть множество пар $(x_k, y_k), i = \overline{1,l}$ – примеров, на которых будем проверять модель: $x_k \in \mathcal{X}$ – вход, $y_k \in \mathcal{Y}$ – ожидаемый (эталонный) выход. Оценки будем строить именно на основании этих данных, подразумевая, что он хорошо отражает требования к моделям.

Далее, $\mathcal{C} = \{C_1, C_2, \dots, C_m\}$ – множество метрик, то есть функций

$$C_i: (\mathcal{X}, \mathcal{Y}, \mathcal{Y}) \to \mathbb{R}\,,\ i = \overline{1,m}$$

В общем случае они работают с тремя аргументами: вход, ожидаемый выход (x_k, y_k из датасета) и $\hat{y}_k$ – результат работы модели. Часто также встречаются метрики, сравнивающие фактический выход только с ожидаемым или только со входом. Их основная задача – численно оценить, как хорошо сработала модель на данном примере. Для простоты будем считать, что метрики строятся таким образом, что большему значению метрики соответствует лучшая оценка.

Также введем обозначение $V_i^j = \{C_j(x_k, y_k, M_i(x_k)),\ k = \overline{1,l}\},\ j = \overline{1,m},\ i = \overline{1,n}$ – множество значений, которые приняла метрика C_j, посчитанная на результатах модели M_i для всех примеров из датасета.

Таким образом, именно тройка $(\mathcal{M}, \mathcal{S}, \mathcal{C})$ полностью описывает все необходимые входные данные: мы оценивает работу моделей из $\mathcal{M}$ на выборке $\mathcal{S}$ по метрикам из $\mathcal{C}$.

Нам же необходимо предложить способ, как по этим данным составить ранжирование моделей, то есть упорядочить их от лучшей к худшей. Для этого каждую модель нужно сначала сопоставить некоторому числу T отражающему ее «силу», а затем сравнивать уже их.

Обзор существующих подходов

Метод средних

Наиболее простым и распространенным методом на данный момент является метод средних. Он заключается в том, чтобы для каждой модели посчитать среднее значений по каждой метрике, посчитанной на каждом элементе датасета, а затем их взвешенное среднее, которое и будет отождествляться с "силой".

Более формально, каждой модели M_i ставится в соответствие вектор

$$\boldsymbol{\mu}^{(i)} = \left(\mu_1^{(i)}, \mu_2^{(i)}, \dots, \mu_m^{(i)}\right)$$

где

$$\mu_j^{(i)} = \frac{1}{l} \sum_{v \in V_i^j} v$$

После чего T_i вычисляется как взвешенная сумма $\sum_{j=1}^{m} w_j \mu_j^{(i)}$, считая, что w_j - заранее заданные весовые коэффициенты, отражающие «важность» каждой метрики.

У такого подхода, несмотря на его простоту, есть целый ряд недостатков. Во-первых, метод крайне чувствителен к неоднородности шкал: метрики могут измеряться в разных единицах и быть ненормированными (например, первая принимает значения от -1 до 1, а вторая - от 0 до 1), что требует дополнительной предобработки. Во-вторых, метод требует определения весов w_j, которые должны задаваться "снаружи" экспертом или аналитиком, что само по себе является непростой задачей и делает метод субъективным. Не всегда можно обоснованно оценить, насколько один критерий важнее другого, а ошибочная или произвольная расстановка весов может приводить к смещенному ранжированию. В-третьих, мы теряем большое количество статистической информации, считая исключительно средние значения. Например, единственный выброс (что может происходить довольно часто в случае, если модель хорошо обучилась работе с каким-либо узким классом задач, но плохо работает с

любыми другими примерами) будет приводить к существенному смещению среднего значения.

В итоге, метод средних может стать хорошим первым приближением, так как не сложен вычислительно и прост в реализации, однако все перечисленные недостатки приводят к необходимости дальнейшего поиска более универсальных подходов.

PCRA

Алгоритм PageRank [1] – это метод вычисления относительной значимости узлов в направленном графе, изначально предложенный для анализа ссылочной структуры Интернета и широко применяемый для ранжирования страниц в выдаче поисковых машин. В более общем виде, его можно воспринимать как модель стохастического блуждения по графу: каждая вершина "накапливает" значимость, полученную от других вершин, указывающих на нее.

В алгоритме PCRA, предложенном в [2] предлагается метод построения графа доминирования альтернатив на основе усреднения оценок по множеству критериев для дальнейшего применения к нему PageRank.

Сначала, как и в методе средних, для каждой альтернативы определяется вектор средних значений по всем метрикам $\boldsymbol{\mu}^{(i)}$. Затем вводится бинарная функция предпочтения

$$P_j(i,i') = \begin{cases} 1, & \text{при } v_j\left(\mu_j^{(i)}\right) > v_j\left(\mu_j^{(i')}\right) \\ 0, & \text{при } v_j\left(\mu_j^{(i)}\right) \le v_j\left(\mu_j^{(i')}\right) \end{cases}$$

где $\mathrm{v_j}(\cdot)$ – монотонная нормализующая функция, принимающая значения от 0 до 1. В статье предлагается взять

$$v_j\left(\mu_j^{(i)}\right) = \frac{\mu_j^{(i)} - \min\limits_{k=1..n} \mu_j^{(k)}}{\max\limits_{k=1..n} \mu_j^{(k)} - \min\limits_{k=1..n} \mu_j^{(k)}} \in [0,1]$$

Эта функция определяет, выигрывает ли альтернатива i у альтернативы i' по j-ому критерию.

На основании введенной функции предпочтения, для каждой пары (M_i, M_j) вычисляется общее количество критериев, по которым M_i доминирует M_j:

$$G_{ij} = \sum_{k=1}^{m} P_k(i,j)$$

Таким образом формируется матрица $G \in \mathbb{N}^{n \times n}$, из которой формируется направленный взвешенный граф и матрица переходов

$$\Phi_{ij} = \begin{cases} G_{ij}\left(\sum_{k=1}^{n} G_{kj}\right)^{-1}, \text{если} \sum_{k=1}^{n} G_{kj} > 0 \\ 1 / n, \quad \text{иначе} \end{cases}$$

– это стохастическая по столбцам матрица, определяющая вероятность перехода из вершины i в вершину j.

Далее находится вектор $\boldsymbol{r} \in \mathbb{R}^n$, удовлетворяющий

$$\boldsymbol{r} = Q\boldsymbol{r},$$

$$\sum_{j=1}^{n} r_j = 1, Q = (1-\alpha)\Phi + \frac{\alpha}{n}E$$

где $E \in \mathbb{R}^{n \times n}, E_{ij} = 1 \; \forall i,j = \overline{1,n}$. Параметр α (фактор демпфирования) обычно принимается равным 0.85 и отвечает за баланс между двумя способа перехода: по ребру и в случайную вершину, обеспечивая тем самым устойчивость алгоритма. Чем больше α – тем большее значение имеет реальная структура графа и его веса. Вектор r и будет содержать значения «силы» каждой альтернативы: $\mathrm{r}_i = T_i$.

Использование PageRank позволяет методу в целом не просто суммировать количество доминирований, а оценивать их относительное качество. Модель, которая доминирует над другими, которые, в свою очередь, сами доминируют над третьими, получит более высокую оценку – это ключевое свойство PageRank, суть которого состоит переходе от попарных сравнений к оценке всей структуры. Однако, алгоритм все еще полагается на средние значения для попарных сравнений и, как и метод средних, теряет существенную часть известной информации, редуцируя ее до единственного числа для каждой пары (модель,

метрика). Таким образом, основные недостатки метода средних сохраняются несмотря на то, что в некоторой степени компенсируются построенным алгоритмом.

PROMETHEE

Метод PROMETHEE, описанный в статье [3], предназначен для решения задач многокритериального стохастического выбора (MSCDM). Его основная идея - ввести величину степени доминирования одной альтернативы (в нашем случае - модели) над другой по какому-либо критерию (метрике), а затем агрегировать их для получения T_i, но вводится она не так, как описано в PCRA.

Для двух случайных величин с известными непрерывными функциями распределения

$$X_1 \sim f_1(x), X_2 \sim f_2(x)$$

степень доминирования X_1 над X_2 обозначается $D_{f_1 \succ f_2}$ и вычисляется по формуле:

$$D_{f_1 \succ f_2} = \int_{-\infty}^{\infty} \int_{-\infty}^{x_1} f_1(x_1) f_2(x_2) dx_2\, dx_1$$

Для величин с дискретными распределениями

$$Y_1 \sim g_1(y),\ Y_2 \sim g_2(y)$$

$$D_{g_1 \succ g_2} = \sum_{y_1=-\infty}^{\infty} \sum_{y_2=-\infty}^{y_1} g_1(y_1) g_2(y_2) - 0.5 \sum_{y_1=-\infty}^{\infty} g_1(y_1) g_2(y_2)$$

В случае, когда одна величина обладает дискретным распределением, а вторая - непрерывным:

$$D_{f \succ g} = \sum_{y=-\infty}^{\infty} \left[g(y) \int_{y}^{\infty} f(x) dx \right]$$

Также показано, что эти величины обладают следующими свойствами, выполняющимися при всех трех определениях степени доминирования

$$D_{f_1 \succ f_2} + D_{f_2 \succ f_1} = 1$$

$$0 \leq D_{f_1 \succ f_2}, D_{f_2 \succ f_1} \leq 1$$

что позволяет существенно сократить количество вычислений при применении метода

Все три этих определения вводятся таким образом, чтобы отражать значение $\mathbb{P}(x > y)$, $x \sim X$, $y \sim Y$. Действительно, $\mathbb{P}(x > y)$ в точности равна степени доминирования X над Y, что будет использовано при дальнейшем анализе.

Для каждой метрики C_j составляется матрица

$$\begin{array}{c} \\ A_n \\ A_n \\ \vdots \\ A_n \end{array} \begin{array}{c} \begin{array}{cccc} A_n & A_n & \cdots & A_n \end{array} \\ \begin{pmatrix} D_{11j} & D_{12j} & \cdots & D_{1nj} \\ D_{21j} & D_{22j} & \cdots & D_{2nj} \\ \vdots & \vdots & \ddots & \vdots \\ D_{n1j} & D_{n2j} & \cdots & D_{nnj} \end{pmatrix} \end{array}$$

которая обозначается как $D_j, j = \overline{1, m}, D_{ikj}$ - степень доминирования альтернативы i над альтернативой k по критерию j. Это матрица попарных доминирований альтернатив друг над другом по j-ой метрике, $D_{ikj} + D_{ijk} = 1$ по указанному ранее свойству.

После этого строится матрица

$$D_{n \times n} = \sum_{j=1}^{m} D_j$$

– общая агрегированная матрица доминирования в попарных сравнениях. Основываясь на ней, для каждой альтернативы определяются входящий и исходящий потоки:

$$\Phi^+(A_i) = \frac{1}{n-1} \sum_{k=1, k \neq i}^{n} D_{ik}$$

$$\Phi^{-}(A_i) = \frac{1}{n-1} \sum_{k=1, k \neq i}^{n} D_{ki}$$

То есть для каждой A_i исходящий поток – это сумма i-ой строке, а входящий - по i-ому столбцу в матрице D. По этим данным вычисляется чистый поток $\Phi(A_i) = \Phi^{+}(A_i) - \Phi^{-}(A_i)$, которому и будет сопоставляться сила альтернативы.

Такой метод является более продвинутым в сравнении, например, с методом средних. Он не требует сведения величин к единственному числу - среднему, но работает с функциями плотности распределения, то есть получает из данных намного больше информации, от чего оценки оказываются более точными и обоснованными. Потенциально разнородные метрики не "смешиваются" друг с другом, а на этапе построения матриц D_j приводятся к единой величине - степени доминирования, из-за чего не чувствителен к заданию весов w_j (их все еще можно использовать на этапе агрегации D_j в D, но этот шаг не является критическим). Итоговая матрица попарных сравнений D поддается интерпретации: каждый ее элемент показывает силу предпочтения одной альтернативы другой. Метод также сопровождается готовыми формулами для разных типов входных распределений, что делает его более универсальным. В нашей задаче значения, которые могут принимать метрики, могут быть как дискретными (часто встречаются в задачах классификации), так и непрерывными (задачи сегментации или анализа текста)

С другой стороны, такой подход не лишен и минусов. Так, например, многократное вычисление двойных несобственных интегралов или сумм (для каждой пары критериев) является довольно сложной задачей. Кроме того, основной недостаток заключается в том, что изначально требуется знание плотностей распределения всех величин. В поставленной формулировке задачи, эта информация не доступна. Это очень существенное ограничение, которое требует дополнительной нетривиальной работы со входными данными алгоритма.

Метод PROMETHEE, основанный на степенях доминирования случайных величин друг на другом, дает концептуально строгий и обоснованный подход к анализу задачи, обеспечивает интерпретируемость результатов и не требует задания заранее определенных весов. Тем не менее, его все еще нельзя использовать напрямую, потому что мы не обладаем всеми необходимыми данными, но его все еще можно использовать в качестве отправной точки для разработки усовершенствованного алгоритма.

Точность оценок чистых потоков

Ранее было указано, что метод PROMETHEE в своей стандартной формулировке предполагает наличие готовых функций плотности вероятности для сравнения объектов по различным критериям. Однако в условиях поставленной задачи мы располагаем лишь выборками соответствующих случайных величин ξ_i^j, которые характеризуют силу i-ой модели по j-ому критерию. В таких ситуациях часто прибегают к использованию приближений истинных функций, найденных параметрическими или непараметрическими методами.

Для вычисления степени доминирования, определенной в методе PROMETHEE, одной случайной величины над другой в случае, если обе имеют непрерывные функции распределения, можно проделать следующие выкладки:

$$D_{f_1>f_2} = \int_{-\infty}^{\infty}\int_{-\infty}^{x_1} f_1(x_1)f_2(x_2)dx_2\,dx_1 = \int_{-\infty}^{\infty} f_1(x_1)\int_{-\infty}^{x_1} f_2(x_2)dx_2\,dx_1$$

$$= \int_{-\infty}^{\infty} f_1(x_1)F_2(x_1)dx_1 = \int_{-\infty}^{\infty} f_1(x)F_2(x)dx$$

Общий чистый поток для каждой модели

$$\Phi_i = \Phi_i^+ - \Phi_i^- = \frac{1}{n-1}\sum_{k=1,k\neq i}^{n} D_{ik} - D_{ki}$$

$$= \frac{1}{n-1}\sum_{k=1,k\neq i}^{n}\left(\sum_{j=1}^{m} D_{ikj} - \sum_{j=1}^{m} D_{kij}\right)$$

$$= \frac{1}{n-1} \sum_{k=1,k\neq i}^{n} \left(\sum_{j=1}^{m} D_{f_i^j \succ f_k^j} - D_{f_k^j \succ f_i^j} \right)$$
$$= \frac{1}{n-1} \sum_{k=1,k\neq i}^{n} \left(\sum_{j=1}^{m} \int_{-\infty}^{\infty} f_i^j(x) F_k^j(x) dx - \int_{-\infty}^{\infty} f_k^j(x) F_i^j(x) dx \right)$$
$$= \frac{1}{n-1} \sum_{k=1,k\neq i}^{n} \sum_{j=1}^{m} \int_{-\infty}^{\infty} f_i^j(x) F_k^j(x) - f_k^j(x) F_i^j(x) dx$$

То есть вычисление чистого потока сводится к вычислению величины следующего вида

$$\begin{aligned}
\Psi_{12} &= \int_{-\infty}^{\infty} f_1(x)F_2(x) - f_2(x)F_1(x)dx \\
&= \int_{-\infty}^{\infty} f_1(x)F_2(x) + f_2(x)F_1(x) - 2f_2(x)F_1(x)dx \\
&= 1 - 2\int_{-\infty}^{\infty} f_2(x)F_1(x)dx \\
&= -\int_{-\infty}^{\infty} f_2(x)F_1(x) - f_1(x)F_2(x)dx \\
&= -\int_{-\infty}^{\infty} f_2(x)F_1(x) + f_1(x)F_2(x) - 2f_1(x)F_2(x)dx \\
&= -\left(1 - 2\int_{-\infty}^{\infty} f_1(x)F_2(x)dx\right) \\
&= 2\int_{-\infty}^{\infty} f_1(x)F_2(x)dx - 1
\end{aligned}$$

Введем обозначения $J = \int_{-\infty}^{\infty} f_1(x)F_2(x)dx$, $\hat{J} = \int_{-\infty}^{\infty} \hat{f}_1(x)\hat{F}_2(x)dx$, где под выражениями со знаком "^" понимаются оценки соответствующих величин, полученные из выборок значений, без них – истинные величины. Тогда будем иметь $\Psi_{12} = 2J - 1$.

Было бы полезно иметь оценку сверху для $|\Psi_{12} - \hat{\Psi}_{12}|$ как функцию от величины выборки $a(n)$, поскольку это позволило бы точно определить ошибку, с которой могут быть найдены чистые потоки Φ_i, и, следовательно, меры силы каждой модели.

$$
\begin{aligned}
&|\Psi_{12} - \widehat{\Psi}_{12}| \\
&= 2|J - \hat{J}| \\
&= 2\left|\int_{-\infty}^{\infty} f_1(x)F_2(x) - \hat{f}_1(x)\hat{F}_2(x)\,dx\right| \\
&= 2\left|\int_{-\infty}^{\infty} (f_1(x) - \hat{f}_1(x))F_2(x)dx + \int_{-\infty}^{\infty} \hat{f}_1(x)(F_2(x) - \hat{F}_2(x))dx\right| \\
&\leq 2\left(\int_{-\infty}^{\infty} |f_1(x) - \hat{f}_1(x)|dx + \sup_{x\in\mathbb{R}} |F_2(x) - \hat{F}_2(x)|\right)
\end{aligned}
$$

Выкладки выполнены при условии, что $\hat{f}_1$ также обладает всеми свойствами функции плотности распределения, а $\hat{F}_2$ – функции распределения.

Здесь второе слагаемое – L_∞-норма ошибки оценки F_2. Для нее известно неравенство Дворецкого-Кифера-Вольфовица [4]:

$$\mathbb{P}\left(\sup_{x\in\mathbb{R}}\left|F(x) - F_p(x)\right| > \varepsilon\right) \leq 2e^{-2p\varepsilon^2}\ \forall\varepsilon > 0$$

где F_p – эмпирическая функции распределения, построенная по гистограмме выборки размера p.

Первое же слагаемое является L_1-нормой ошибки оценки f_1. Для ядерных оценок плотности вероятности существуют некоторые теоретические результаты [5][6], например

$$M\left\|f(x) - f_n(x)\right\|_{L1} \leq C(h + 1\,/\,\sqrt{nh})$$
$$M\left\|f(x) - f_n(x)\right\|_{L1} = O(n^{-\beta/(2\beta+1)})$$

где h – ширина окна, $\beta -$ гладкость f, C зависит от свойств истинной функции f и выбранного ядра. Эти результаты можно использовать для точной оценки хотя бы математического ожидания ошибки, если было бы задано ограничение на гладкость. Однако в условиях задачи мы не располагаем никакой априорной информацией о гладкости или иных свойствах истинных плотностей. В общем случае, никакая детерминированная $a(n) \to 0$ не может гарантировать

выполнение $|\Psi_{12} - \widehat{\Psi}_{12}| < a(n)$, что следует из двухточечного аргумента Ле Кама [7]. Таким образом, невозможно построить строго обоснованный интервал, в котором гарантированно содержится истинное значение Φ_i лишь на основании оценок, полученных из выборок. Тем не менее, на практике такие приближения нередко демонстрируют удовлетворительную точность даже при отсутствии строгих гарантий.

.

Предлагаемое решение

В данной главе рассмотрено предлагаемое решение, которое основывается на вышеописанном методе PROMETHEE. Его главный недостаток заключается в том, что в условиях поставленной задачи мы не можем работать напрямую с функциями распределения значений метрик для всех моделей. Но нам даны V_i^j, которые мы ввели при формализации задачи, которые можно интерпретировать как выборку случайной величины ξ_i^j, соответствующую силе i-ой модели по j-ой метрике. Понимание силы игрока в соревновании или турнире как случайной величины – это стандартный прием, показавший свою эффективность, который лежит в основе, например, системы рейтингов Elo, широко применяемого в шахматах, или Bradely-Terry – его многомерном аналоге.

Более того, языковые модели, сравнению которых посвящена эта работа, в том числе и их мультимодальные версии, сами по себе имеют вероятностную природу. Они генерируют текст, изображения и другие данные не детерминировано, а на основе распределения вероятностей над возможными последовательностями токенов (сочетаний символов) и пикселей. Даже при одинаковых входных данных модели могут выдавать отличающиеся результаты. Сведение ξ_i^j до среднего значения приводит к значительной потере информации, а значит делает итоговые оценки менее обоснованными, даже если после этого используются такие мощные алгоритмы, как PageRank. Такой упрощенный подход игнорирует не только присущую моделям неопределенность, но и разброс в их производительности, что может оказаться критичным при составлении итогового ранжирования.

В дальнейшем мы будем понимать степень доминирования случайной величины $X_1 \sim f_1$ над $X_2 \sim f_2$ как $\mathbb{P}(X_1 > X_2)$. Это, как уже было указано, полностью согласуется с определениями, данными в статье про PROMETHEE, но универсально для разных типов распределений и проще вычислительно, т.к. не требует вычисления несобственных интегралов. Для обеспечения точности полученных оценок доминирования, будем использовать Bootstrap-подход,

который может компенсировать возможный недостаток данных в исходных выборках.

Применение Bootstrap

Бутстрэп — это непараметрический статистический метод, позволяющий оценить распределение выборочных статистик (таких как среднее, медиана, дисперсия, коэффициент корреляции или, в данном случае, степень доминирования) на основе имеющихся данных. Он не требует строгих предположений о форме распределения генеральной совокупности. Практически, бутстрэп позволяет имитировать процесс многократного сбора выборки, используя доступные данные.

Основная идея бутстрэпа заключается в том, что имеющаяся выборка (размера n) может быть рассмотрена как "суррогат" истинной генеральной совокупности. Если мы не можем взять много выборок из реальной популяции, как в случае нашей задачи, то имеет смысл взять большое число "псевдовыборок" из уже имеющейся. Это достигается путем сэмплирования с возвращением (sampling with replacement) из исходных данных. Каждая такая псевдовыборка, называемая бутстрэп-выборкой, имеет тот же размер n, что и исходная. Поскольку элементы выбираются с возвращением, некоторые элементы могут повторяться в бутстрэп-выборке, а некоторые – отсутствовать. Повторяя этот процесс B раз (B — большое число), получаем B бутстрап-выборок. Для каждой из этих B выборок вычисляем интересующую нас статистику, и набор из B таких статистик формирует ее эмпирическое бутстрэп-распределение. Несмотря на то, что бутстрэп не может создать новую информацию, он эффективно использует ту, что уже есть, для получения более робастных оценок по сравнению с методами, основанными на асимптотических предположениях, которые требуют большого объема данных.

В предлагаемом методе будем использовать бутстрэп для более надежного вычисления степеней доминирования по определенному критерию через

эмпирическую вероятность. Для каждой данной выборки X_i^j строится B бутстрэп выборок $X_{i1}^j, X_{i2}^j, \dots, X_{iB}^j$. После, для каждой b-ой пары $\left(X_{ib}^j, X_{i'b}^j\right)$ вычисляется эмпирическая вероятность того, что случайно выбранный элемент из $\mathrm{X}_{\mathrm{ib}}^{\mathrm{j}}$ окажется больше случайно выбранного элемента из $X_{i'b}^j$, считая долю случаев, когда значение из первой превосходит значение из второй

$$\widehat{\mathbb{P}}^{(b)}\left(\xi_i^j > \xi_{i'}^j\right) = \frac{1}{n}\sum_{k=1}^{B} \mathbb{I}\left(X_{ib,k}^j > X_{i'b,k}^j\right)$$

где $\mathbb{I}$ – индикаторная функция.

В результате этого процесса, для каждой пары моделей по какому-то критерию будет получено не одно значение, а целый массив из B эмпирических значений степеней доминирования, который представляет собой эмпирическое бутстрэп-распределение степени доминирования. Для того, чтобы определить какое-то единственное значение, с которым будет ассоциироваться сила каждой модели, можно взять среднее по всему бутстрэп-распределению

$$\widehat{D}_{ik}^j = \frac{1}{B}\sum_{b=1}^{B} \widehat{\mathbb{P}}^{(b)}\left(\xi_i^j > \xi_{i'}^j\right)$$

С помощью описанного алгоритма все еще можно найти лишь оценки истинных степеней доминирования, хотя благодаря бутстрэпу они будут являться робастными. Для того, чтобы метод можно было считать законченным, необходимо продолжить дальнейшую работу.

Анализ статистической значимости

В рамках поставленной задачи необходимо определить, различаются ли модели между собой по значению их чистого потока. Для каждой пары моделей i и k возникает вопрос: является ли поток Φ_i статистически значимо выше, ниже либо сопоставим с потоком Φ_k? Эта задача может быть сведена к проверке следующей пары гипотез:

$$H_0: \Phi_i = \Phi_k, H_1: \Phi_i \neq \Phi_k$$

Проверка статистических гипотез обычно осуществляется с помощью p-значения (p-value), которое отражает вероятность наблюдать величину различия между моделями, не меньшую, чем та, что получена в эксперименте, при условии, что нулевая гипотеза верна. Таким образом, p-значение характеризует степень противоречия между наблюдаемыми данными и гипотезой H_0.

Случай, когда p-value оказывается достаточно малым (ниже установленного порогового значения α, обычно $\alpha = 0.05$), интерпретируется как статистически значимое свидетельство в пользу того альтернативной гипотезы. В противном случае различие между потоками признается статистически несущественным.

Для вычисления этих p-значений можно вновь использовать бутстрэп. Для этого для каждой Φ_i сгенерировать B' ее бутстрэп-оценок, причем для этого можно использовать уже построенные результаты $\widehat{D}_{ik}^{j}$, случайно выбирая одно из B значений для каждой пары моделей по каждому критерию и пересчитывая на них значения $\widehat{\Phi}_{ib\prime}, b' = \overline{1, B'}$. Чтобы проверить различие между Φ_i, Φ_k формируются эмпирические распределение их разностей $\Delta^{(b')} = \widehat{\Phi}_{ib\prime} - \widehat{\Phi}_{kb\prime}$, считается p-value как

$$\frac{1}{B'}\sum_{b'=1}^{B'} \mathbb{I}\left(|\Delta^{(b')}| > \widehat{\Phi}_i - \widehat{\Phi}_k\right)$$

Важно отметить, что при выполнении множества попарных сравнений возрастает риск получения ложноположительных результатов. Чтобы контролировать эту проблему, можно применить один из методов, регулирующих уровень значимости таким образом, чтобы обеспечить более строгий контроль над ошибками. В предлагаемом методе будем использовать поправку Бонферрони. Это гарантирует, что вероятность совершить хотя бы одну ошибку первого рода во всей серии тестов не превысит α.

После вычисления p-value целесообразно объединять модели, различия в чистом потоке между которыми не являются статистически значимыми. Для этого можно использовать следующий алгоритм: сначала все модели упорядочиваются по убыванию их оценок чистого потока. Далее, начиная с модели с наивысшим

значением, последовательно просматриваются остальные. Каждая следующая модель сравнивается с уже включёнными в текущую группу: если её чистый поток статистически не отличается от потока хотя бы одной из моделей, уже входящих в группу, она присоединяется к ней. В противном случае формируется новая группа, в которую модель включается в качестве первого элемента. Такой подход позволяет выявить множества моделей с сопоставимым качеством, не делая заведомо необоснованных различий между ними.

Применение на реальных данных

Визуально-текстовая задача

В первом примера рассмотрим задачу Visual Question Answering, (VQA) [8], в которой модели проверяются на способность воспринимать и интерпретировать изображения и генерировать ответы на поставленные вопросы по нему. Таким образом VQA - задача вида (text, image) → text.

В качестве датасета будем использовать VQA v2, широко используемый для подобных задач. Он состоит из трех компонентов: вопросы, аннотации (ожидаемые ответы) и изображений. Вопросы охватывают определение представленных объектов, их количества, цвета, совершаемых действий или местоположения. В отличие от первой версии, примеры были подобраны таким образом, чтобы нельзя было угадать ответ, основываясь исключительно на вопросе и игнорируя изображение. Для тестирования было выбрано 5 моделей, которые получали одинаковые входные данные и системные инструкции: deepseek-vl2[9], deepseek-vl2-tiny[9], qwen2.5-vl-7b-instruct[10], qwen2-vl-7b-instruct[10], gemma-3-4b-it[11]. Модели оценивались по 3 распространенным метрикам: BERTScore, CosineSimilarity, RougeL и цене использования. Все модели запускались с помощью сервиса YandexCloud, а стоимость рассчитывалась на основе тарифов, указанных в официальной документации [12]

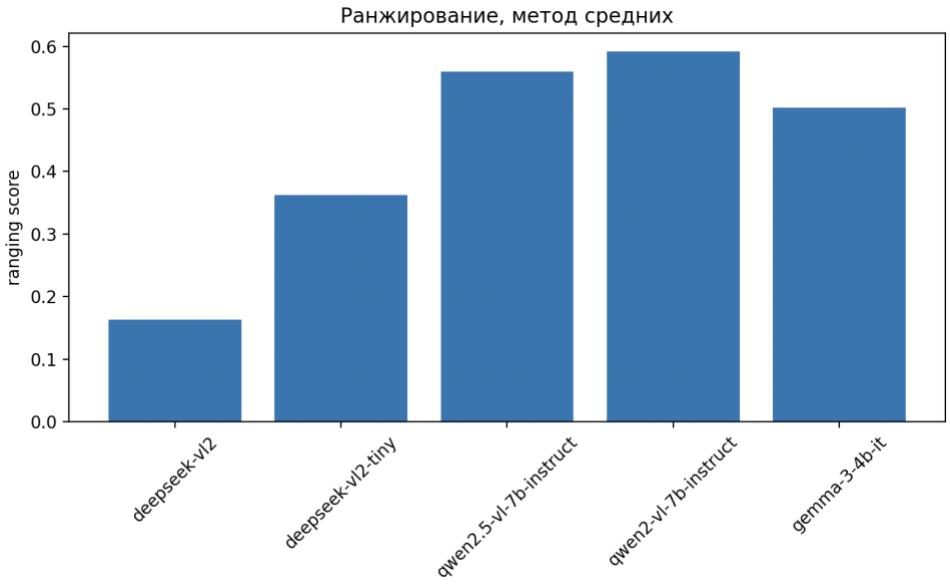

Рисунок 1. Ранжирование, метод средних. VQA

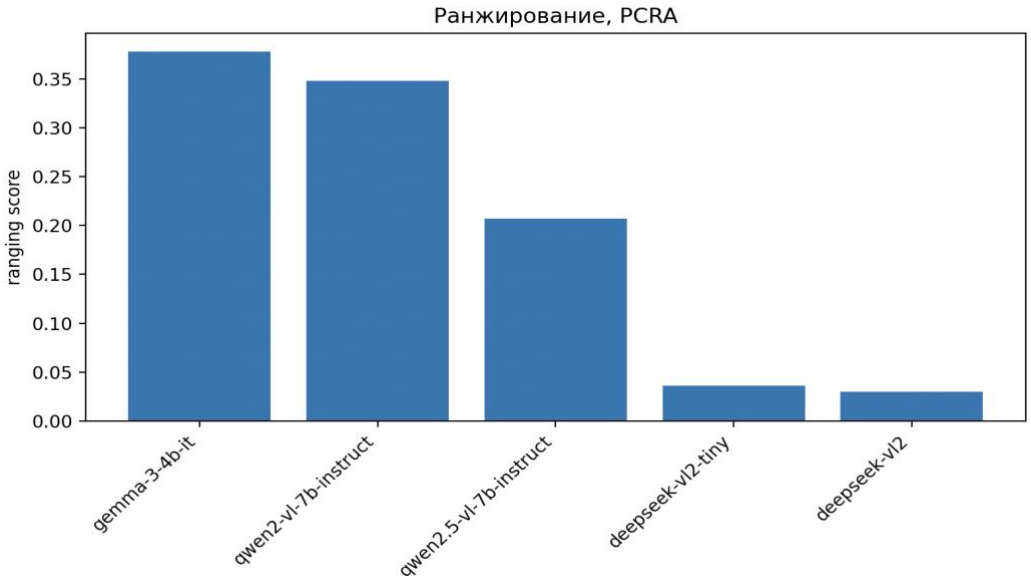

Рисунок 2. Ранжирование, PCRA. VQA

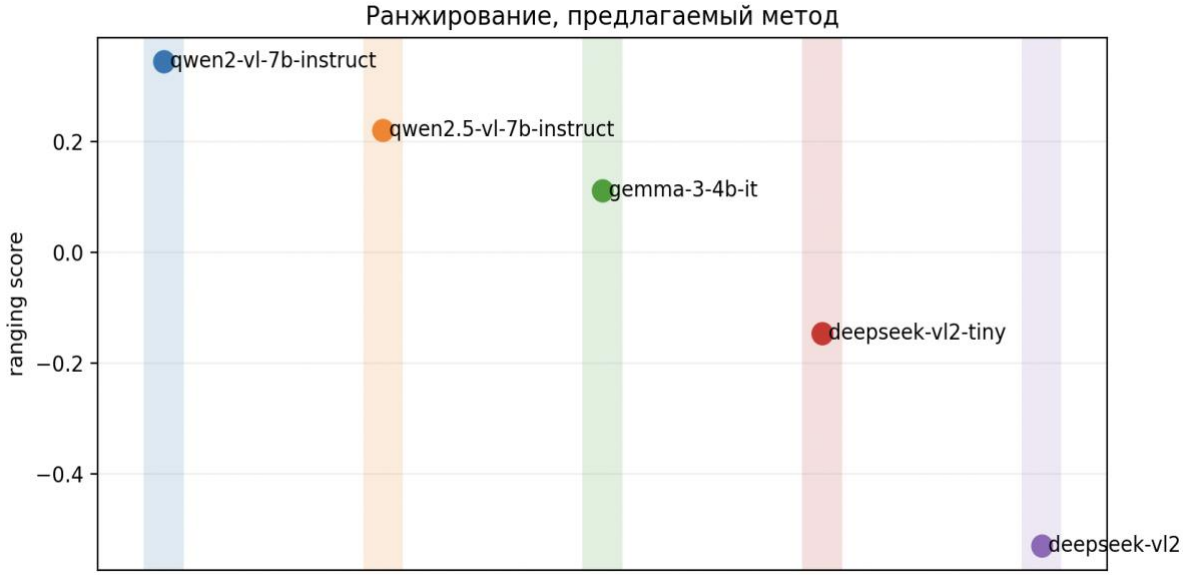

Рисунок 3. Ранжирование, предлагаемый метод. VQA

Во всех показанных методах выделяются три лидера: qwen2.5-vl-7b-instruct, qwen2-vl-7b-instruct, gemma-3-4b-it, а две версии модели deepseek показывают худшие результаты. Ранжирование методом средних и предложенным показали одинаковые результаты, PCRA же отличается в определении порядка внутри категории лидеров. Для определения того, какой способ ранжирования оказался наиболее соответствующим действительности, можно провести анализ усредненных значений результатов работы моделей.

Средние значения метрик

	BERTScore	Cosine Similarity	Price	ROUGE-L
deepseek-vl2	0.83	0.29	-0.40	0.12
deepseek-vl2-tiny	0.87	0.44	0.10	0.29
qwen2.5-vl-7b-instruct	0.97	0.82	0.10	0.59
qwen2-vl-7b-instruct	0.98	0.85	0.10	0.68
gemma-3-4b-it	0.95	0.74	0.10	0.46

Рисунок 4. Средние значения метрик. VQA

Показатели цены были домножены на -1, чтобы соответствовать правилу «чем больше, тем лучше», т.к. все предыдущие рассуждения велись именно в таком предположении.

Из этих данных можно сделать вывод о том, что получившиеся метрики являются непротиворечивыми, т.е. если модель является лучшей по одной метрике, то она является лучшей и по всем остальным. Ранжирование определяется однозначно: qwen2-vl-7b-instruct, qwen2.5-vl-7b-instruct, gemma-3-4b-it, deepseek-vl2-tiny, deepseek-vl2 (от лучшей к худшей), что полностью соответствует результатам предлагаемого метода и метода средних и показывает, что на данном примере PCRA отработал некорректно.

Задача резюмирования текста

В задаче резюмирования (суммаризации) текста требуется по данному тексту создать его более короткое, но сохраняющее смысл изложение. Эта задача является фундаментальной в обработке естественного языка (NLP) и имеет множество практических применений, таких как создание дайджестов новостей, выжимка информации из научных статей, сокращение документов и другие.

Будем использовать датасет CNN/Daily Mail (CNN/DM), который является одним из наиболее широко используемых в области автоматической суммаризации текста. Он был создан путем сбора новостных статей с веб-сайтов CNN и Daily Mail и сопоставления их с аннотациями, предназначенных для краткого

изложения статьи их авторами. Примеры состоят из полного текста исходной статьи на различные темы, и их сжатой версии, которые, как правило являются очень близкими парафразами ключевых предложений. Работа проводилась с моделями yandexgpt-lite, yandexgpt, yandexgpt-32k, llama-lite, gigachat_lite [14], которые оценивались по метрикам: Cosine Similarity, BERTScore, Coverage, Density и цена. Как и в прошлый раз, будем сравнивать три способа ранжирования.

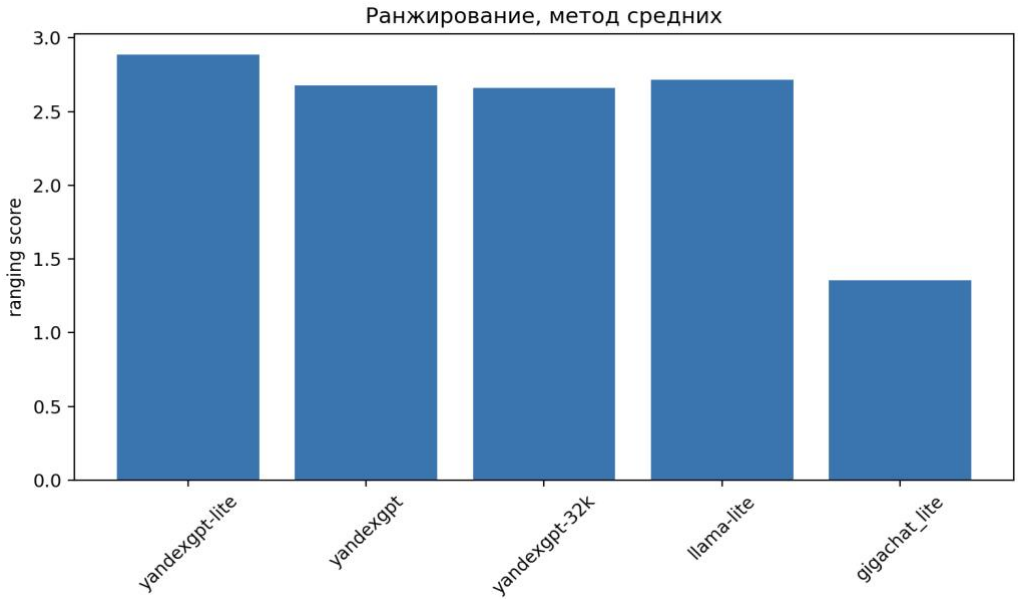

Рисунок 5. Ранжирование, метод средних. CNN

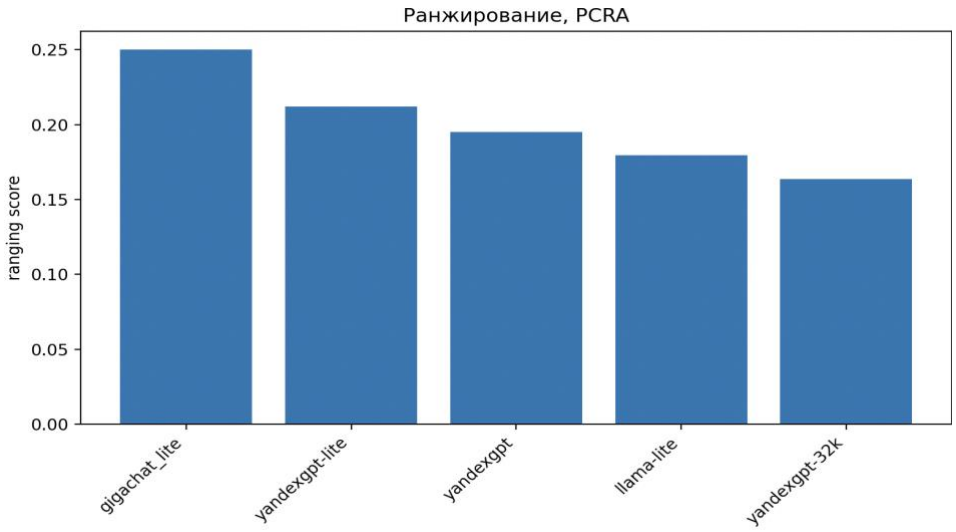

Рисунок 6. Ранжирование, PCRA. CNN

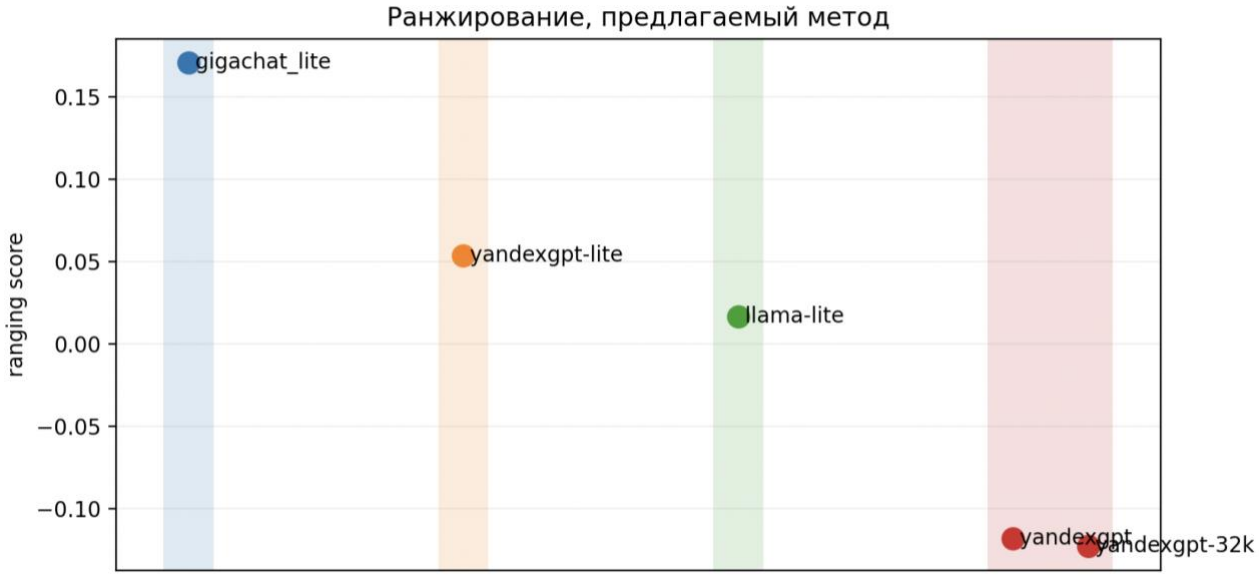

Рисунок 7. Ранжирование, предлагаемый метод. CNN

Вновь результаты получились разными. В отличие от предыдущего случая, в этом примере методы не сходятся даже в определении лидеров: PCRA и предлагаемый метод считают gigachat_lite лучшей опцией, а метод средних ставит его сразу на последнее место. Еще одним отличием является то, что третий метод нашел различия между yandexgpt и yandexgpt-32k статистически неразличимыми и объединил их в одну группу. Обратимся к анализу средних значений

Средние значения метрик

	Cosine Similarity	BERTScore	Coverage	Density	Price
yandexgpt-lite	0.48	0.65	0.19	13.30	-0.20
yandexgpt	0.49	0.65	0.17	13.28	-1.20
yandexgpt-32k	0.49	0.65	0.17	13.19	-1.20
llama-lite	0.48	0.65	0.19	12.47	-0.20
gigachat_lite	0.73	0.69	0.26	5.31	-0.20

Рисунок 8. Средние значения метрик. CNN

В этот раз не нашлось модели, которая оказалась бы лучшей по всем критериям. Однако, gigachat_lite значительно превосходит остальные по Cosine Similarity и Coverage, и немного – по BERTScore; является одним из лучших по цене и уступает остальным лишь по одной метрике, но достаточно существенно, что и привело к ошибочной оценке методом средних. Затем, llama-lite и yandexgpt-lite

имеют практически одинаковые показатели по всем метрикам, кроме Density, где llama-lite оказался хуже, причем достаточно для того, чтобы третий метод не объединил их в одну группу. Две худшие модели из представленных – yandexgpt и yandexgpt32k обладают близкими значениями метрик и находятся примерно на одном уровне с предыдущими, но сильно уступают им по критерию стоимости. Между собой они отличаются настолько слабо, что предлагаемый метод их сгруппировал, т.е. различие между ними оказалось статистически незначимым. Ранжирования, полученные с помощью PCRA и разработанного метода отличаются порядком llama-lite и yandexgpt. Последний несильно уступает первому по Coverage и Density, в то же время критически проигрывая ему в цене, т.е. ожидаемый результат не соответствует предложенному PCRA варианту.
Таким образом, предложенный метод и на этом примере показал себя лучше, чем альтернативные.

Интеграция с ClearML

ClearML — это открытая платформа для управления жизненным циклом разработки моделей машинного обучения и анализа данных. Она обеспечивает инструменты для отслеживания экспериментов, управления вычислительными ресурсами и автоматизации рабочих процессов. ClearML позволяет работать с моделями, наборами данных и результатами экспериментов, включая поддержку мультимодальных данных. В то же время платформа имеет ряд недостатков. Во-первых, ClearML не позволяет напрямую работать с метриками как с функциями, что усложняет анализ и сравнение моделей. Во-вторых, платформа не обладает средствами автоматического определения совместимостьи моделей с наборами данных, что особенно критично в случаях, когда используется сразу нескольких моделей для различных задач.
Для ликвидации подобных недостатков и повышения точности выбора наилучшей модели из множества доступных в проекте ClearML была создана система автоматического сравнения и ранжирования мультимодальных моделей на базе предлагаемого метода.

Решение данной задачи заключается в явно выраженной спецификации метаданных, которые должны быть представлены как для моделей, так и для наборов данных. Эти сведения включают URL модели, описание ожидаемых на вход данных (названия параметров и их модальность) и результата работы модели. В случае с датасетами также используется описание их содержимого — каждый набор данных представляет собой CSV-файл, в котором может присутствовать информация о файлах с образцами в виде относительных путей до них. Эти меры не требуют существенных трудозатрат для реализации, так как вся необходимая информация предоставляется лишь один раз в стандартизированном формате и не накладывает никаких ограничений на внутренне устройство моделей и датасетов, но лишь на интерфейс взаимодействия с ними. Система также имеет готовый набор распространённых метрик для задач вида (image → image) и (text → text), а также позволяет определить собственные, специализированные метрики в зависимости от характера задач.

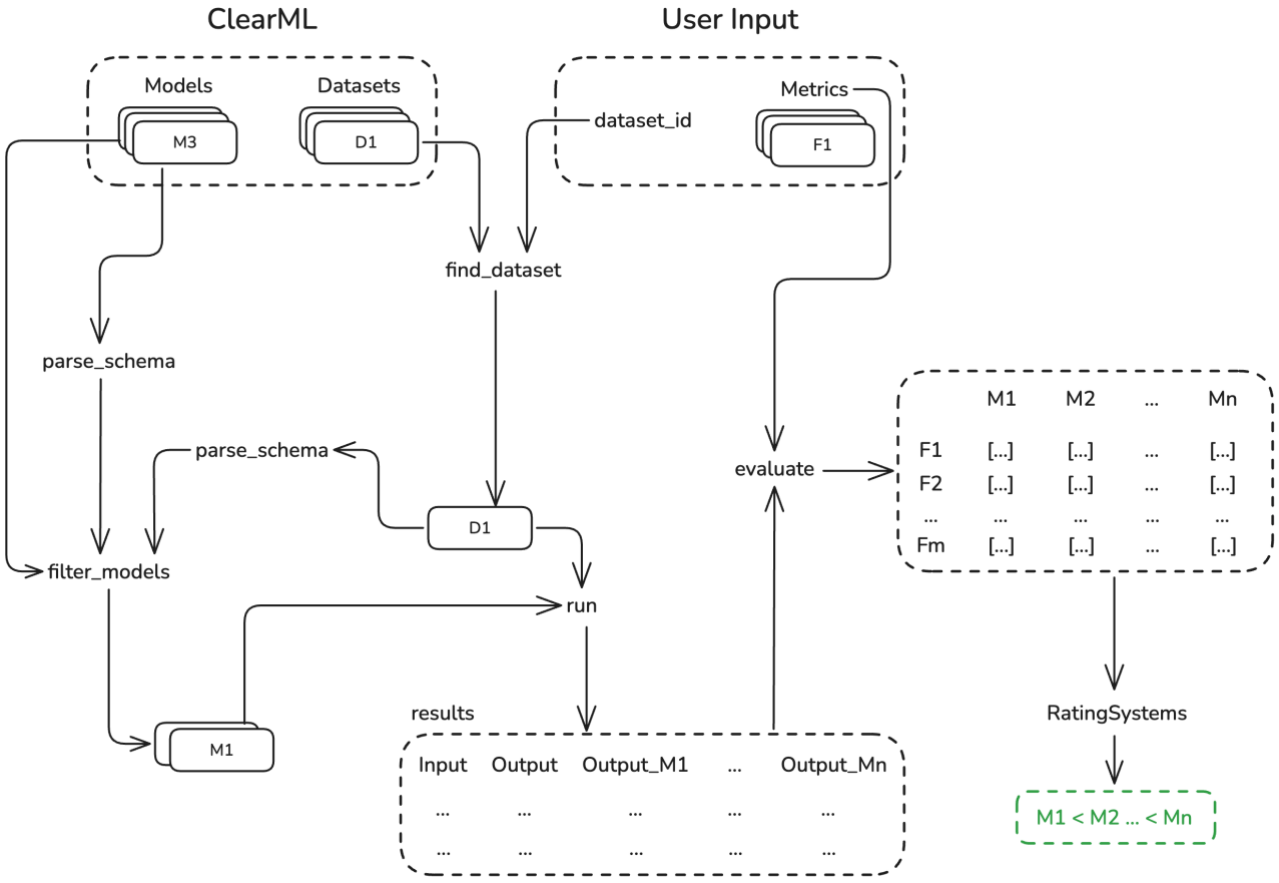

Рисунок 9. Схема работы системы

Разработанная система требует для запуска лишь id датасета и, возможно, python-файлов с определениями пользовательских метрик. Все расчеты происходят автоматически и не требуют вмешательства пользователя, а результатом работы является итоговый график ранжирования, загруженный на платформу. Запуск происходит через интерфейс командной строки, что делает систему удобной для использования на серверах. Так, второй эксперимент, описанный в этой главе, был полностью проделан с помощью разработанной программы.

Заключение

Актуальность настоящего исследования обусловлена стремительным ростом числа и разнообразия мультимодальных языковых моделей**,** а также их повсеместным применением для решения прикладных задач. С увеличением сложности и количества таких моделей возникает острая потребность в объективных и надёжных методах их сравнения и ранжирования.

Целью данной дипломной работы являлась разработка метода, который обеспечил бы объективное сравнение и ранжирование нескольких моделей, основываясь на заданном датасете и наборе метрик качества. Особое внимание уделялось специфике случайной природы моделей машинного обучения, что требовало подхода, способного учитывать стохастичность их результатов.

Для достижения поставленной цели был проведён анализ существующих подходов к решению подобных задач. В результате этого анализа было разработано улучшение одного из них, адаптированное для условий поставленной задачи. В работе были не только продемонстрированы теоретические ограничения предложенного метода, но и показано его превосходство над альтернативными подходами. Разработанный алгоритм лёг в основу системы автоматизированного сравнения моделей, интегрированной в платформу ClearML, что является значимым практическим результатом.

Разработанный метод и реализованная система автоматизированного сравнения открывают перспективы применения в промышленности: система может быть использована для выбора наиболее подходящей MLLM для конкретного производственного процесса или продукта, что приводит к повышению эффективности и качества.

Дальнейшие исследования могут быть сосредоточены на поиске оптимальных гиперпараметров алгоритма, изучении его устойчивости к зашумленным данным и коррелированным метрикам.

Список использованных источников

1. Page, L., Brin, S., Motwani, R., & Winograd, T. (1998). The PageRank Citation Ranking: Bringing Order to the Web. *World Wide Web Internet And Web Information Systems*, *54*(1999–66)
2. Scholz, M., Pfeiffer, J., & Rothlauf, F. (2017). Using PageRank for non-personalized default rankings in dynamic markets. *European Journal of Operational Research*, *260*(1)
3. Liu, Y., Fan, Z. P., & Zhang, Y. (2011). A method for stochastic multiple criteria decision making based on dominance degrees. *Information Sciences*, *181*(19), 4139–4153.
4. Massart, P. (2007). The Tight Constant in the Dvoretzky-Kiefer-Wolfowitz Inequality. The Annals of Probability
5. Silverman B. W. *Density estimation for statistics and data analysis* / B. W. Silverman. – London: Chapman & Hall, 1986. - 175 с.
6. Taylor C.C., Devroye L., Györfi L. Nonparametric density estimation: The L1 view // The Annals of Statistics. 1985. Vol. 13, №4. P. 1364–1369.
7. Tsybakov A. B.Introduction to Nonparametric Estimation /A. B. Tsybakov.– 2-е изд. – New York: Springer, 2009. – 376 с.
8. Goyal, Y., Khot, T., Agrawal, A., Summers-Stay, D., Batra, D., & Parikh, D. (2019). Making the V in VQA Matter: Elevating the Role of Image Understanding in Visual Question Answering. *International Journal of Computer Vision*, *127*(4)
9. DeepSeek-VL: Towards Real-World Vision-Language Understanding / H. Lu, W. Liu, B. Zhang, B. Wang, K. Dong, B. Liu, J. Sun, T. Ren, Z. Li, H. Yang, Y. Sun, C. Deng, H. Xu, Z. Xie, C. Ruan — 2024
10. Qwen-VL: A Versatile Vision-Language Model for Understanding, Localization, Text Reading, and Beyond / J. Bai, S. Bai, S. Yang, S. Wang, S. Tan, P. Wang, J. Lin, C. Zhou, J. Zhou. — 2023
11. Gemma: Open Models Based on Gemini Research and Technology / Gemma Team, T. Mesnard, C. Hardin – 2024
12. Yandex Cloud. Тарифы на использование Foundation Models [Электронный ресурс]. https://yandex.cloud/ru/docs/foundation-models/pricing, Режим доступа: свободный. – Дата обращения: 24.04.2025
13. Chen D., Bolton J., Manning C. D. A Thorough Examination of the CNN/Daily Mail Reading Comprehension Task // *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. — Berlin, Germany: Association for Computational Linguistics, 2016. — С. 2358–2367
14. GigaChat Family: Efficient Russian Language Modeling Through Mixture of Experts Architecture / GigaChat team, Mamedov V., Kosarev E. и др. – 2025.