МИНИСТЕРСТВО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
МОСКОВСКИЙ ФИЗИКО-ТЕХНИЧЕСКИЙ ИНСТИТУТ
(ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ)

ФИЗТЕХ-ШКОЛА ПРИКЛАДНОЙ МАТЕМАТИКИ И ИНФОРМАТИКИ
КАФЕДРА ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ

Оценка частичных доказательств в Lean 4 на основе анализа матриц внимания большой языковой модели

Выпускная квалификационная работа на степень бакалавра
студентки Б05-112 группы ФПМИ Линич А.А.

Научный руководитель
к.ф.-м.н. Баранников С.А.

Москва 2024

Оглавление

Глава 1

Аннотация

В работе рассматривается задача бинарной классификации частичных доказательств в формальном языке Lean 4. Для этого был подготовлен датасет с частичными доказательствами на языке Lean4, а также предложено два интерпретируемых метода, основанных на анализе матриц внимания большой языковой модели (БЯМ) DeepSeek-Prover-v2-7b. Построенные классификаторы достигли большей точности предсказания, чем бейзлайны. Один из методов показал улучшение в точности предсказания на более, чем 3% относительно собственной оценки БЯМ в режиме запроса без обучения.

Глава 2

Введение

В последние годы большие языковые модели (БЯМ) продемонстрировали высокие результаты в разнообразных областях, однако связные логические рассуждения для них по-прежнему остаются остаются нерешённой в полной мере задачей (Benchmarking Formal Mathematical Reasoning of Large Language Models). В этом контексте особый интерес представляет домен математических доказательств, формализованных в строго логических языках, таких как Lean 4, Isabelle и Coq.

Языки доказательств были разработаны как инструмент поддержки математиков для проверки доказательств. В условиях стремительно возрастающей сложности современных математических результатов задача их ручной верификации становится крайне сложной и трудоёмкой. Языки доказательств позволяют автоматизировать проверку: успешная компиляция доказательства эквивалентна его корректности. Именно поэтому языки доказательств становятся всё более популярными (xenaproject.wordpress.com/2022/09/12/beyond-the-liquid-tensor-experiment).

В данной работе рассматривается язык Lean 4. Lean 4 опирается на исчислении индуктивных конструкций (10.1007/978-3-030-79876-5_37). Доказательство здесь строится при помощи тактик: каждая тактика либо изменяет состояние самого доказательства, либо уточняет текущую цель. Отсутствие активной цели означает, что доказательство завершено.

Существует несколько подходов к автоматической генерации доказательств. Один из них — конечная генерация, при которой модель сразу выдаёт полную последовательность тактик, необходимых для завершения доказательства (ProofNet: A Benchmark for Autoformalization and Theorem Proving). Альтернативный метод — пошаговая генерация, при которой тактики генерируются и немедленно применяются по одной, что позволяет учитывать отклик среды Lean 4 при генерации последующих шагов (LeanDojo: Theorem Proving with Retrieval-Augmented Language Models, LEAN-STAR: LEARNING TO INTERLEAVE THINKING AND PROVING). Наконец, активно развиваются подходы, основанные на обучении с подкреплением (DeepSeek-Prover-V1.5: Harnessing Proof Assistant Feedback for Reinforcement Learning

and Monte-Carlo Tree Search).

Описанные выше базовые подходы к генерации доказательств могут использоваться с усовершенствованными стратегиями управления поиском, например (Learn from Failure: Fine-Tuning LLMs with Trial-and-Error Data for Intuitionistic Propositional Logic Proving). В стратегиях управления поиском: выборка с пересмотром (Machine learning for first-order theorem proving: Learning to select a good heuristic) и обучение с внешним сигналом (Learning to Utilize Shaping Rewards: A New Approach), – важна оценка незавершённых состояний доказательства. Например, в Tree of Thoughts (Tree of Thoughts: Deliberate Problem Solving with Large Language Models), такая оценка используется для отбора наиболее перспективных цепочек рассуждений. В подходах с обучением с подкреплением она может служить источником вспомогательного сигнала (Learning to Utilize Shaping Rewards), ускоряющего обучение. Так, оценщик доказательств — модели, прогнозирующей успешность частичных попыток — оказывается полезным инструментом. При этом простые эвристики, такие как оценка самой модели, уже доказали свою полезность в задачах логического вывода (Self-Consistency Improves Chain of Thought) и служат одним из бейзлайнов в данной работе.

В работе строятся два варианта бинарного классификатора неоконченных доказательств. Актуальность поставленной цели объясняется возможностью использовать классификатор в роли оценщика в вышеописанных сценариях для повышения качества генерации математических доказательств.

Глава 3

Постановка задачи

В данной работе ставится задача бинарной классификации частичных доказательств в Lean 4. Каждому примеру в датасете соответствует пара (φ, T), где φ — формулировка теоремы, а $T = (t_1, \ldots, t_k)$ — последовательность тактик длины k. Каждому примеру сопоставлена метка $y \in \{0, 1\}$:

$$y = \begin{cases} 1, & \text{если отрывок } T \text{ может продолжается до полного доказательства,} \\ 0, & \text{иначе.} \end{cases}$$

Необходимо построить классификатор $f\colon (\varphi, T) \mapsto \hat{y} \in \{0, 1\}$, на тестовой выборке превосходящий бейзлайны по метрике Accuracy:

$$\mathit{Accuracy}\big(f\big) \;>\; \max\big\{\, \mathit{Accuracy}(f),\; \mathit{Accuracy}(f) \big\},$$

где f — собственная оценка модели, а f — градиентный бустинг (Prokhorenkova et al. (2018). CatBoost: unbiased boosting with categorical features.) над вектором специального токена начала последовательности (Attention Is All You Need.). Последний был выбран в качестве бейзлайна, так как является типичным подходом в задаче классификации текстов (Text Classification Algorithms: A Survey). Дополнительным ограничением является использование одинаковой БЯМ для f, f, f. Для f также введено требование строить классификатор исключительно на основе матриц внимания всех слоёв и голов БЯМ. Такой подход обеспечивает интерпретируемость решения.

Глава 4

Решение

4.0.1 MTD_0

Предлагаемый метод основывается на топологическом анализе графов, построенных на матрицах внимания БЯМ. Перед применением алгоритма матрицы внимания необходимо преобразовать, чтобы получить на их основе матрицы смежности соответствуюшего неориентированного графа, описывающего структуру взаимодействия между токенами доказательства.

Пусть задана матрица внимания

$$A \in R^{n \times n},$$

где $n = |P| + |\varphi| + |T|$ — суммарное число токенов, P – промпт, присоединяющийся к теореме перед ее фомулировкой. Поскольку рассматривается декодер трансформера, матрица A является нижнетреугольной и несимметричной. Чтобы задать на её основе неориентированный граф, используется симметризация, переход от меры близости к расстоянию и зануление диагонали:

$$D := A + A^{\top}.$$

$$D := \mathbf{1}_{n \times n} - D,$$

$$D_{ii} := 0 \quad \forall\, i \in \{1, \ldots, n\}.$$

Полученная матрица D интерпретируется как взвешенная матрица смежности неориентированного графа G, в котором вершины соответствуют токенам, а веса рёбер выражают расстояние между токенами. Чем выше исходный вес между двумя токенами, тем меньше расстояние между ними в графе, что соответствует более тесной связи между шагами доказательства.

Далее внимание между токенами внутри одного блока (то есть внутри каждого из P, φ, $t_1, \ldots, t_k$) зануляется. Это позволяет исключить из рассмотрения внутреннюю структуру блоков и сфокусироваться на связях между ними. Формально:

$$D_{ij} := 0 \quad \text{если } i, j \in B, \quad B \in \{P, \varphi, t_1, \ldots, t_k\}.$$

На полученном графе рассчитывается признак, представляющий собой упрощенную версию Manifold Topology Divergence (MTD) (Manifold Topology Divergence: a Framework for Comparing Data Manifolds.). В исходной формулировке MTD вычисляется как средняя сумма длин кросс-баркодов гомологий первого порядка по нескольким случайным подвыборкам:

$$\text{MTD} = \frac{1}{m} \sum_{j=1}^{m} mtd_j,$$

где mtd_j — сумма длин кросс-баркодов для jй подвыборки.

В данной работе используется модифицированная версия метода: без сэмплирования подмножеств и с суммой длин кросс-баркодов для нулевой гомологии. После совершенных упрощений, новый метод становится эквивалентен рассчету суммы длин ребер минимального остовного дерева, построенного на графе D. Значение метода пропорционально количеству ненулевых ребер в минимальном остовном дереве, что равно инкременту количества тактик в доказательстве, поэтому полученное значение необходимо нормировать на количество тактик. Пусть такой метод будет обозначен MTD_0.

$$\text{MTD}_0(D) = \frac{1}{k} \sum_{(i,j) \in T_{\min}} D_{ij},$$

где $T_{\min}$ — множество рёбер, входящих в минимальное остовное дерево графа.

4.0.2 Внимание токенов конца блоков на себя (CBT)

Второй метод оценки связности доказательства рассматривает самовнимание токенов, завершающих блок (CBT). Для фиксированных слоя ℓ и головы q, из матрицы внимания $A \in R^{n \times n}$ извлекаются диагональные элементы $\left(A_{c_P c_P},\ A_{c_\varphi c_\varphi},\ A_{c_{t_1} c_{t_1}},\ \ldots,\ A_{c_{t_k} c_{t_k}}\right)$, соответствующий самовниманию токенов конца блока. Эти значения интерпретируются как мера независимости токена от контекста:

$$A_{ii} = 1 - \sum_{j \neq i} A_{ij},$$

что следует из свойства нормировки attention-весов:

$$\sum_{j=1}^{n} A_{ij} = 1.$$

Таким образом, чем выше A_{ii}, тем слабее токен i взаимодействует с остальными токенами в последовательности. В контексте доказательств это может трактоваться как слабая включённость соответствующей тактики в общий контекст. Итого:

$$= \frac{1}{k+2}\left(A_{c_P c_P} + A_{c_\varphi c_\varphi} + \sum_{r=1}^{k} A_{c_{t_r} c_{t_r}}\right).$$

4.0.3 Построение классификаторов на основе MTD_0 и CBT

Оба признака, MTD_0 и CBT, представляют собой скалярные величины, вычисляемые для каждой пары слоя ℓ и головы запроса q БЯМ. Для построения бинарного классификатора на их основе необходимо выбрать одну пару (ℓ^*, h^*), и подобрать оптимальный порог бинаризации скалярного признака.

Процедура выбора (ℓ^*, h^*) проводится на валидационной выборке. Для обоих признаков и каждой пары (ℓ, h) вычисляются значения на всех примерах валидационной выборки. Далее для каждой пары определяется абсолютная разность между средними значениями признака на положительных и отрицательных примерах и с ее помощью оценивается разность математических ожиданий:

$$\Delta_{\ell,h} = \left| E_{y=1}[s^{(\ell,h)}] - E_{y=0}[s^{(\ell,h)}] \right|,$$

где s обозначает соответствующий признак: MTD_0 или .

Оптимальной считается пара (ℓ^*, h^*), для которой достигается максимум величины $\Delta_{\ell,h}$:

$$(\ell^*, h^*) = \arg\max_{(\ell,h)} \hat{\Delta}_{\ell,h}.$$

После выбора (ℓ^*, h^*) необходимо выбрать наилучший порог классификации. Для этого на сбалансированной валидационной выборке строится ROC-кривая по значениям признака $s^{(\ell^*,h^*)}$, и определяется порог τ^*, максимизирующий Accuracy:

$$\tau^* = \arg\max_{\tau} F_1(\tau),$$

где бинаризация осуществляется правилом

$$\hat{y} = \begin{cases} 1, & s^{(\ell^*,h^*)} \geq \tau, \\ 0, & \text{иначе.} \end{cases}$$

После выбора головы и порога классификатор фиксируется и применяется на тестовой выборке без дополнительной настройки.

Глава 5

Вычислительный эксперимент

В данном разделе описываются условия эксперимента, проведённого для проверки качества предложенных методов классификации частичных доказательств, и сравниваются полученные результаты с результатами двух бейзлайнов: классификацией БЯМ в режиме запроса без обучения (Large Language Models are Zero-Shot Reasoners) и градиентный бустинг на векторе специального токена начала последовательности. Во всех экспериментах в качестве БЯМ используется deepseek-prover-v2-7b (deepseek-prover-v2-7b).

5.0.1 Обработка датасета

Эксперимент проводится на основе датасета PropL (Learn from Failure: Fine-Tuning LLMs with Trial-and-Error Data for Intuitionistic Propositional Logic Proving) с примерами доказательств утверждений из области пропозиционной логики на языке Lean 4. PropL предоставляет примеры решения задач путем проб и ошибок с откатами на предыдущие шаги для неудачных попвток. Из таких траекторий были выделены правильные последовательности тактик, завершающиеся доказательством, а также всевозможные неправильные последовательности, приводящие к неудаче. Далее были отобраны частичные доказательства длины ровно 5 по числу тактик, при этом полная длина исходных цепочек должна быть не менее 7. Такой отбор гарантирует, что пример действительно является частью доказательства, а не целым. Длина частичного доказательства в ровно 5 тактик объясняется желанием избежать эффекта масштаба, при этом рассматривая не слишком короткие доказательства. Также была проведена фильтрация доказательств, у которых полученные части правильной и неправильной траекторий совпали. После фильтрации выборка была сбалансирована по классам методом уменьшения числа примеров преобладающего класса. В результате был сформирован итоговый набор данных объемом 13290 примеров.

Далее выборка была разбита на несколько частей. Было зафиксированно 1000 примеров для теста, 200 примеров для выбора наилучшего промпта для классифи-

кации БЯМ в режиме запроса без обучения, 300 примеров для выбора наилучших пар слой, голова запроса и выбора порога классификации для методов MTD_0, CBT. Для бейзлайна на основе ГБ в обучающую выборку были включены все примеры из датасета, кроме тестовых. В данной работе важен порядок экспериментов.

5.0.2 Классификацией БЯМ в режиме запроса без обучения

В первом эксперименте каждое частичное доказательство передаётся в модель как объединение трех текстов: промпта – поясняющий текст перед доказательством, определяющий роль модели, частичное доказательство и текстовая инструкция после доказательства, задающей формат ответа. Было рассмотрено пять вариантов промпта при фиксированной формулировке инструкции 5.2. Все варианты были протестированы на валидационной выборке. В случае, если модель не соблюдала необходимый формат ответа, метка класса выбиралась случайно и равновероятно. Доля таких нерегламентированных ответов была меньше 4% для каждого варианта промпта. По значению точности был выбран наилучший промпт. Результаты приведены в таблице 5.1.

Таблица 5.1: Классификацией БЯМ в режиме запроса без обучения: метрики для разных промптов на валидационной выборке

Промпт	Accuracy	Precision	Recall	F1
1	0.52	0.56	0.19	0.28
2	0.57	0.64	0.29	0.40
3	0.62	0.65	0.49	0.56
4	0.60	0.69	0.37	0.48
5	0.61	0.67	0.42	0.52

5.0.3 Градиентный бустинг

Catboost (CatBoost: unbiased boosting with categorical features) над вектором специального токена начала последовательности обучается со стандартными параметрами обучения. Используется готовая реализация из библиотеки catboost (https://catboost.ai/do версии 1.2.8. Затем модель используется для получения предсказаний на тестовой выборке.

5.0.4 MTD_0 и CBT

Для каждого слоя и головы запроса рассчитывается значение предложенных методов на ваидационной выборке. Тепловая карта абсолютной разницы среднего зна-

Таблица 5.2: Тексты промптов и инструкции формата ответа

№	Текст промпта
1	You are a Lean 4 proof assistant. Your task is to determine whether the given sequence of Lean 4 tactics represents a semantically correct beginning of a proof of the initial goal.
2	You are a formal proof expert specialized in the Lean 4 theorem prover. Your job is to evaluate whether a given partial sequence of tactics correctly begins a proof of the stated goal in Lean 4.
3	Simulate the behavior of the Lean 4 proof engine. Given an initial goal and a partial proof script, decide whether the script represents a valid and logically consistent start toward proving the goal.
4	You are reviewing partial Lean 4 proofs. Your task is to check whether the given sequence of tactics reflects a correct start — that is, whether it stays on a path that could logically lead to proving the goal, without any semantic errors.
5	Pretend you are running an internal semantic check in the Lean 4 proof assistant. Based on the goal and the partial tactic script provided, determine whether the proof starts correctly and logically — as it would be accepted by Lean's internal logic engine.
Инструкция (конец ввода)	
You must return a single JSON object, and nothing else, in the following format: `{ "verdict": "yes"}` or `{ "verdict": "no"}` Only use lowercase "yes"or "no"as the value of the field. Do not include explanations or any other fields. Do not explain your decision.	

чения методов на правильных и неправильных доказательствах представлены на рисунке 5.1.

Наилучшие пара слой, голова запроса это (10, 10) и (24, 18) для MTD_0 и CBT соответственно. Для лучших пар определим порог классификации на основе ROC-кривых 5.2. Отметим, что для метода CBT перед определением порога значения показателя были домножены на -1, поскольку более высокие значения CBT соответствуют меньшей связности доказательства, то есть более вероятному отсутствию решения.

После определения лучших пар голова, слой и порогов классификации, методы были применены к тестовой выборке.

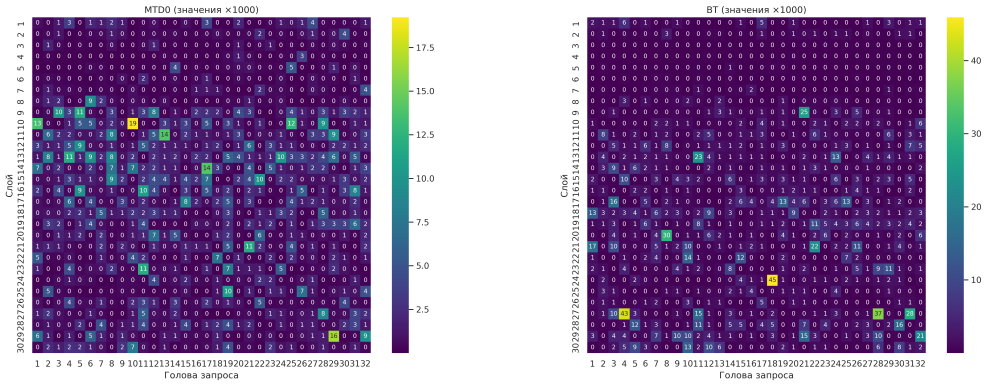

Рис. 5.1: Тепловые карты значений $\Delta_{\ell,h}\mathrm{MTD}_0$ (слева) и CBT (справа).

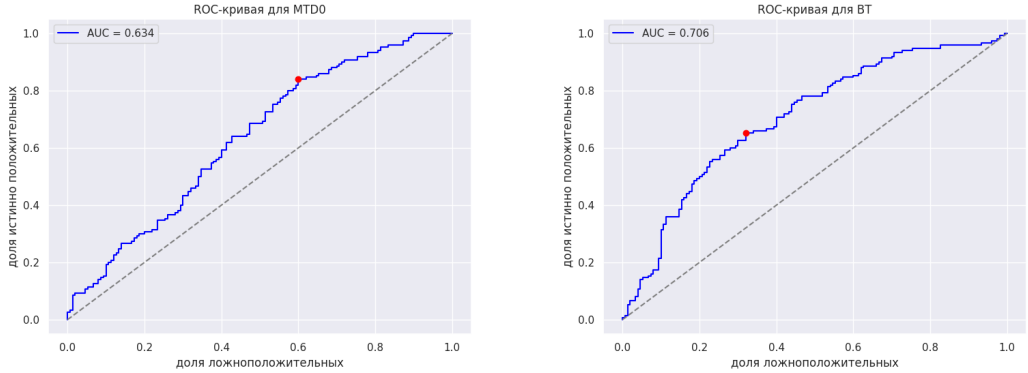

Рис. 5.2: ROC-кривые для лучших пар слой, голова для MTD_0 (слева) и CBT (справа) на валидационной выборке. Красной точкой отмечен оптимальный порог классификации.

5.0.5 Результаты

Итоговые результаты на тестовой выборке приведены в таблице 5.3. Все метрики усреднялись по классам.

Таблица 5.3: Сравнение всех методов классификации на тестовой выборке

Метод	Точность (Accuracy)	Точность (Precision)	Полнота	F1
MTD_0	0.57	0.58	0.57	0.55
CBT	0.62	0.62	0.62	0.62
ГБ	0.52	0.70	0.06	0.12
CO	0.58	0.60	0.58	0.56

Сравнение представленных в таблице 5.3 методов показывает, что наилучшие результаты демонстрирует метод CBT, он достигает максимального значения по общей точности (0.62). Метод MTD_0 имеет точность классификации выше ГБ, однако проигрывает второму бейзлайну – собственной оценке модели. Полученные результаты

подтверждают гипотезу, что предложенные методы на основе анализа матриц внимания модели DeepSeek-Prover-v2-7b позволяют классифицировать частичные доказательства. Кроме способности классифицировать, они обладают интерпретируемостью, и могут быть использованы в реальных сценариях фильтрации и оценки шагов в задачах автоматического доказательства.

Глава 6

Заключение

Итак, был получен классификатор неоконченных математических доказательств, который принимает на вход матрицы внимания фиксированных слоя и головы запроса и отражает меру логической связности доказательства. Была продемонстрирована точность классификации выше бейзлайнов. В частности, улучшение в точности классификации CBT относительно собственной оценки БЯМ составило более 3%. Направления для дальнейших исследований включают изучение обобщающей способности метода: сохранется ли выбор лучших слоя и головы запроса в задачах пропозиционной логики на естественном яыке или задач из другой области математики на Lean 4? Другим возможным продолжением исследования является интеграция предложенного оценщика в системы, где он может использоваться как внешний сигнал, например, в задачах обучения с подкреплением или в механизмах логического рассуждения с возможностью откатов.

Литература

(не менее 17-30 ссылок) Взять всю литературу из LinReview и для обсуждения полного текста работы привести полный список Совет: используйте команду TeX при выводе из файла bbl для получения полного списка