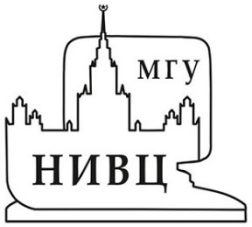
Международная научная конференция «Математика в созвездии наук»
Секция "Суперкомпьютерные вычисления"



Суперкомпьютеры как необходимая основа для развития больших языковых моделей

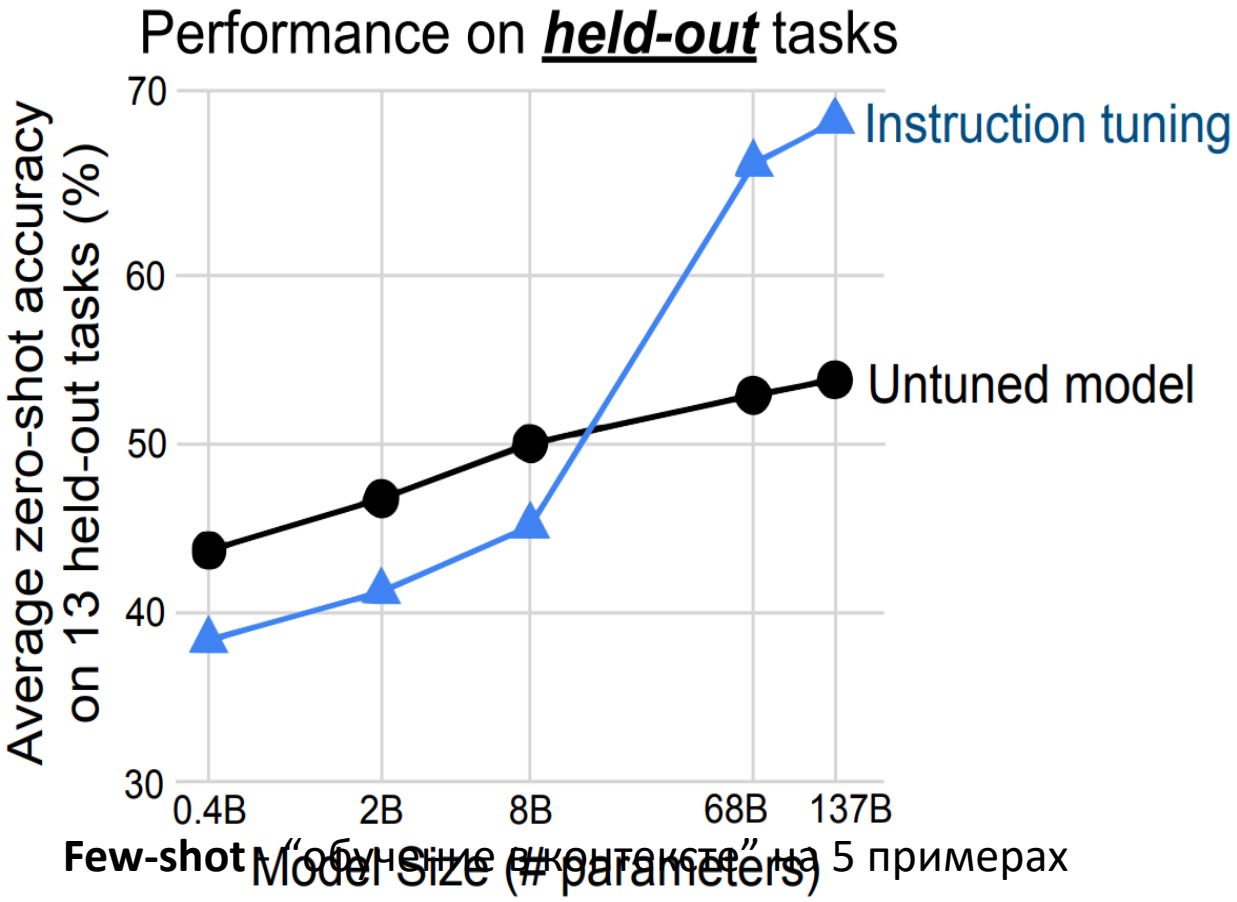
Тихомиров Михаил Михайлович, к.ф-м.н., научный сотрудник НИВЦ МГУ
Чернышев Д.И., аспирант, ВМК МГУ
Лукашевич Н.В., д.т.н., ведущий научный сотрудник НИВЦ МГУ
Добров Б.В., к.ф.-м.н., зав. лаб. НИВЦ МГУ

2 апреля 2024 г.

(Бурная) история развития больших нейросетевых языковых моделей (LLM)

- 2012 --- Сверточные сети в обработке изображений
- 2014 --- Word2Vec (кодирование контекстов употребления слов в пространстве «эмбеддингов»
- 2017 --- “Attention is all you need” – концепт «внимания» - нейросетевое представление взаимной информации
- 2018 --- Transformer – компактное представление «самовнимания»
- 2019 --- GPT2, BERT – возможность самообучения по текстам
- 2022 --- ChatGPT = GPT3.5 – инструктивное обучение – интеграция входных данных и формулировок задач их обработки
- 2023 --- мультимодальные LLM
- 2024 --- человекоботы Figure 01, Tesla Optimus,….

Только большие модели понимают инструкции



Few-shot – “обучение в контексте” на 5 примерах

GPT vs FLOPS: сколько стоит GPT

Model	Total train compute (PF-days)	Total train compute (flops)	Params (M)	Training tokens (billions)	Flops per param per token	Mult for bwd pass
BERT-Base	1.89E+00	1.64E+20	109	250	6	3
BERT-Large	6.16E+00	5.33E+20	355	250	6	3
RoBERTa-Base	1.74E+01	1.50E+21	125	2,000	6	3
RoBERTa-Large	4.93E+01	4.26E+21	355	2,000	6	3
GPT-3 Small	2.60E+00	2.25E+20	125	300	6	3
GPT-3 Medium	7.42E+00	6.41E+20	356	300	6	3
GPT-3 Large	1.58E+01	1.37E+21	760	300	6	3
GPT-3 XL	2.75E+01	2.38E+21	1,320	300	6	3
GPT-3 2.7B	5.52E+01	4.77E+21	2,650	300	6	3
GPT-3 6.7B	1.39E+02	1.20E+22	6,660	300	6	3
GPT-3 13B	2.68E+02	2.31E+22	12,850	300	6	3
GPT-3 175B	3.64E+03	3.14E+23	174,600	300	6	3

Для обучения GPT-3 175B (**3640 PF-days,** $4.6M-$12M) потребовалось бы **7 месяцев** обучения на **512 V100**, или **43 дня** на **512 A100** (₽70M и 112 месяцев на Volta-1).

Стоимость обучения InstructGPT: 4.9 PF-days для SFT и 60 PF-days для PPO-ptx.

Основные задачи использования LLM и варианты действий на практике

- Бизнес-задачи:
  - Обучение ЛОКАЛЬНОЙ версии LLM (конфиденциальность)
  - Настройка на предметную область (тексты, инструкции, промпты)
- Технически:
  - Основной сценарий – ДООБУЧЕНИЕ базовых моделей (foundation models)
  - Проблемы вычислительных ресурсов, ограниченные возможности по размеру моделей
  - Набор оптимизаций (увеличение возможностей малых моделей)

Требования к вычислительным ресурсам в зависимости от размеров локальной LLM

Основная задача – использовать «локальную» модель, использование которой безопасно с точки зрения утечки корпоративных секретов

Размер модели (B = млрд.)	Минимальное	Рекомендуемое
7B	RTX 4090	1 A100
13B	1 A100	1 A100
33B	1 A100	2 A100
70B	2 A100	4 A100

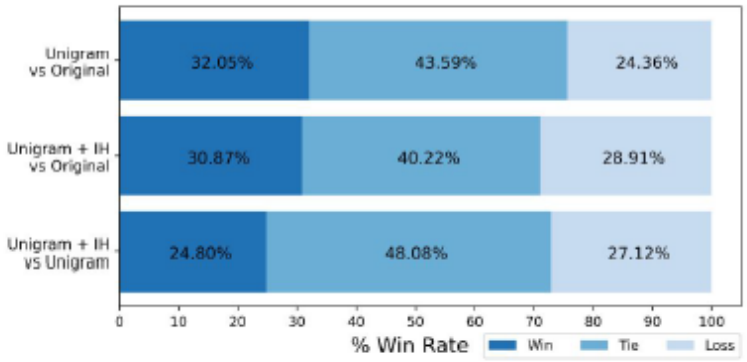
- Данные приведены для графических карт NVideo.
- Видеочипы H100, которые имеют столько же памяти, что и A100 (80 ГБ), однако имеют в 3 раза большую пропускную способность при этом аренда этих чипов на рынке лишь в 2 раза дороже, поэтому аренда H100 в 1.5 раза выгоднее A100.

Адаптация LLM путем настройки токенизации на русский язык

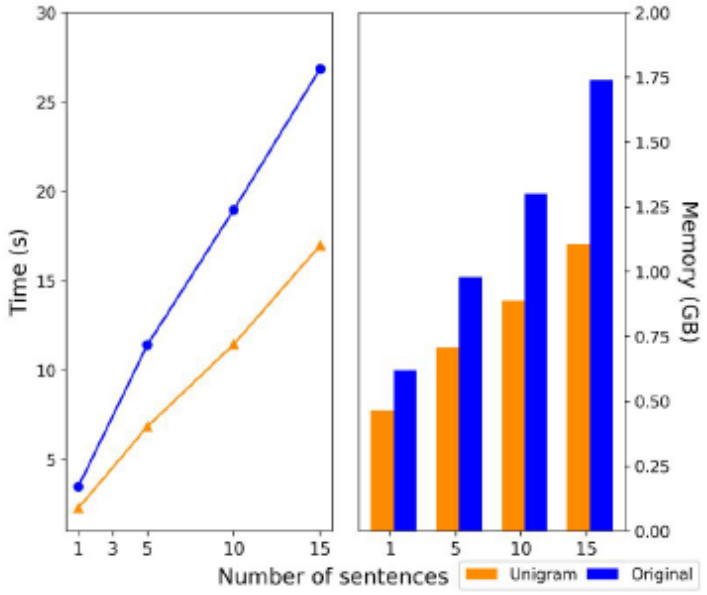
- Новый токенизатор обучался на русскоязычных текстовых данных, из-за чего любое слово представляется в среднем меньшим количеством токенов. Так, например, слово ‘**интеллект’** в новом токенизаторе представляется одним токеном, а в оригинальном разбивалось на три: [**'инте', 'лле', 'кт']**

Сравнение путем выбора лучшей генерации из двух (side-by-side).

Было подготовлено 78 вопросов для моделей,15 аннотаторов.

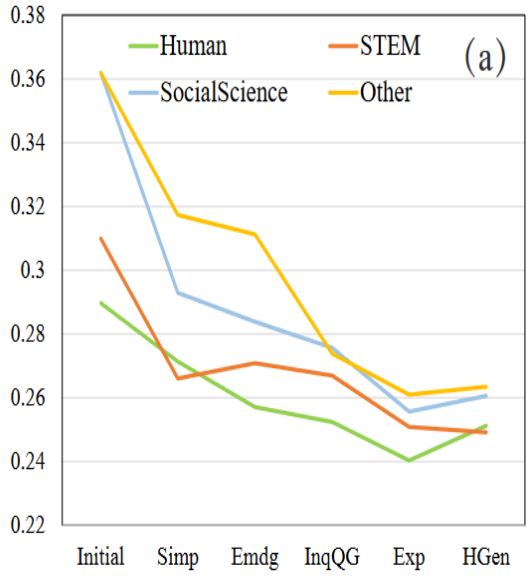


До **60%** прироста в скорости **при генерации** и до **35%** прироста в скорости **при обучении**.

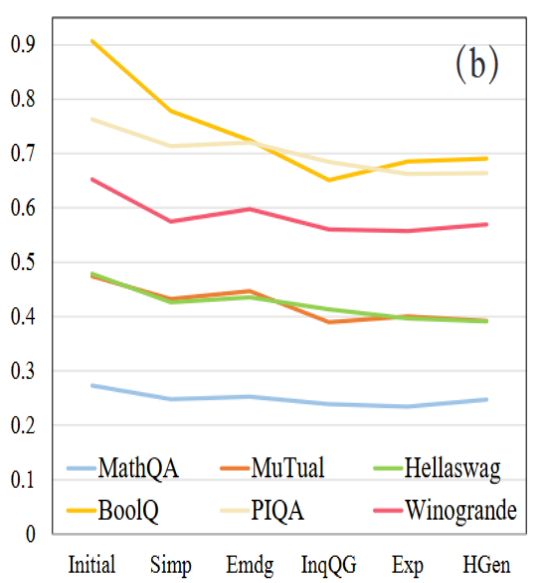


Проблема: дообучение LLM приводит к потере знаний

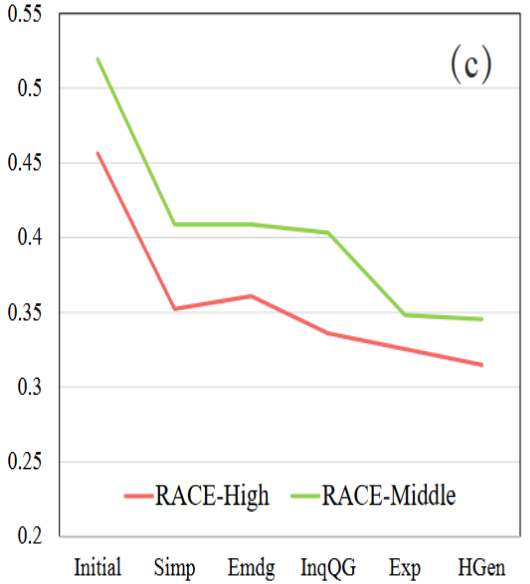
Ответы на общие вопросы



Рассуждения по тексту



Понимание текста



Больше эпох обучения

*Simp, Emdg, InqQG, Exp, HGen – части инструктивного датасета, на котором шло дообучение

*Дообучение шло по схеме Initial (исходная модель)->Simp->Embg->InqQG->Exp->HGen

Luo Y. et al. "An empirical study of catastrophic forgetting in large language models during continual fine-tuning" //arXiv preprint arXiv:2308.08747. – 2023.

Решение: пост-коррекция обучения

Традиционная схема обучения:

$$W_{trained} = W_o + \Delta W$$
$$\Delta W = -\nabla L(W_o, X)$$

где L – функция потерь, $W \in \mathbb{R}^{n \times m}$

Схема LoRA:

$$\Delta W \approx W_{LoRA} = AB$$
$$A = -\nabla L(\lambda A_o, X)$$
$$B = -\nabla L(\lambda B_o, A(X))$$

где $A_o \in \mathbb{R}^{n \times r}, B_o \in \mathbb{R}^{r \times m}, \lambda = \frac{\alpha}{r}$

Наша схема:

$$W_{trained} = W_o + \phi W_{LoRA}$$

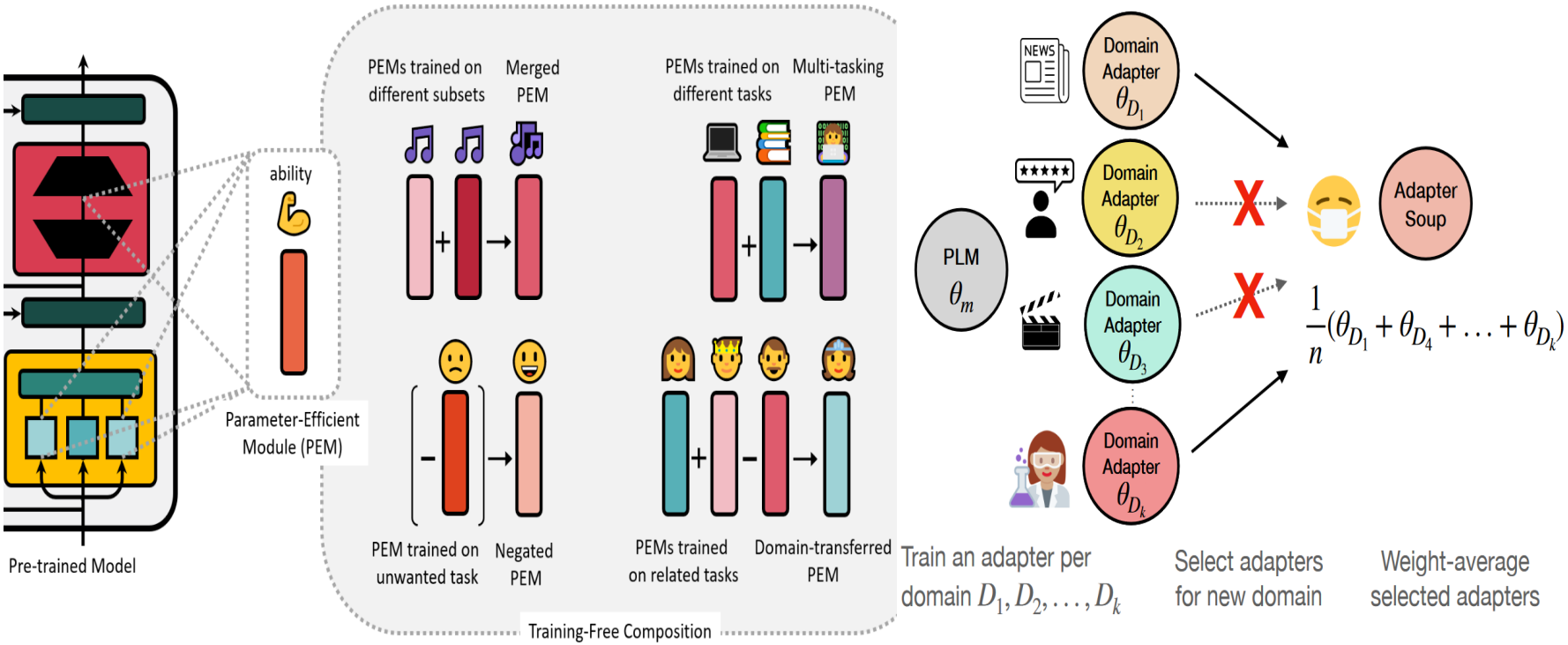
где ϕ – коэффициент коррекции (merging factor), определяется после обучения на отложенной выборке

	MMLU-RU		
	Translated	MERA	RSG
SOLAR 10.7B	**57.9**	0.708	0.794
Two-Stage			
+ Vocabulary optimization			
Projection init	54.5	0.690	0.791
Mean init	54.6	0.706	0.790
+ Continued pre-training			
Embedding-only	55.6	0.684	0.792
LoRA($\alpha = 1024$)	54.4	0.690	0.796
+ merging factor $\frac{1}{2}$	56.8	0.719	**0.805**
LoRA($\alpha = 512$)	55.3	0.693	0.798
+ merging factor $\frac{2}{3}$	57.0	**0.722**	0.800
End-to-End			
LoRA($\alpha = 512$)	54.9	0.695	0.787
+ merging factor $\frac{1}{2}$	56.5	0.714	0.794

Tikhomirov M. and Chernyshev D., "Improving LLM Russian Adaptation with Preliminary Vocabulary Optimization" // under review in Lobachevski 2024.

Альтернативное решение: ансамбли моделей

- **Метод LoRA** явным образом выделяет компоненты весов моделей ΔW_{task}, отвечающие за поведение в конкретных задачах (обозначены как PEM)
- **Арифметические операции** над этими компонентами позволяют изменять их содержимое и таким образом управлять поведением конечной модели



Zhang J. et al. "Composing Parameter-Efficient Modules with Arithmetic Operation" // Advances in Neural Information Processing Systems. – 2024. – T. 36

Решение: UDLM

Динамическое ансамблирование:

$$\Delta W(h) = \sum_{i=1}^{N} hG(h)_i \Delta W_{task}^i$$

$$G(h) = softmax\big(TopK(h \cdot W_G)\big)$$

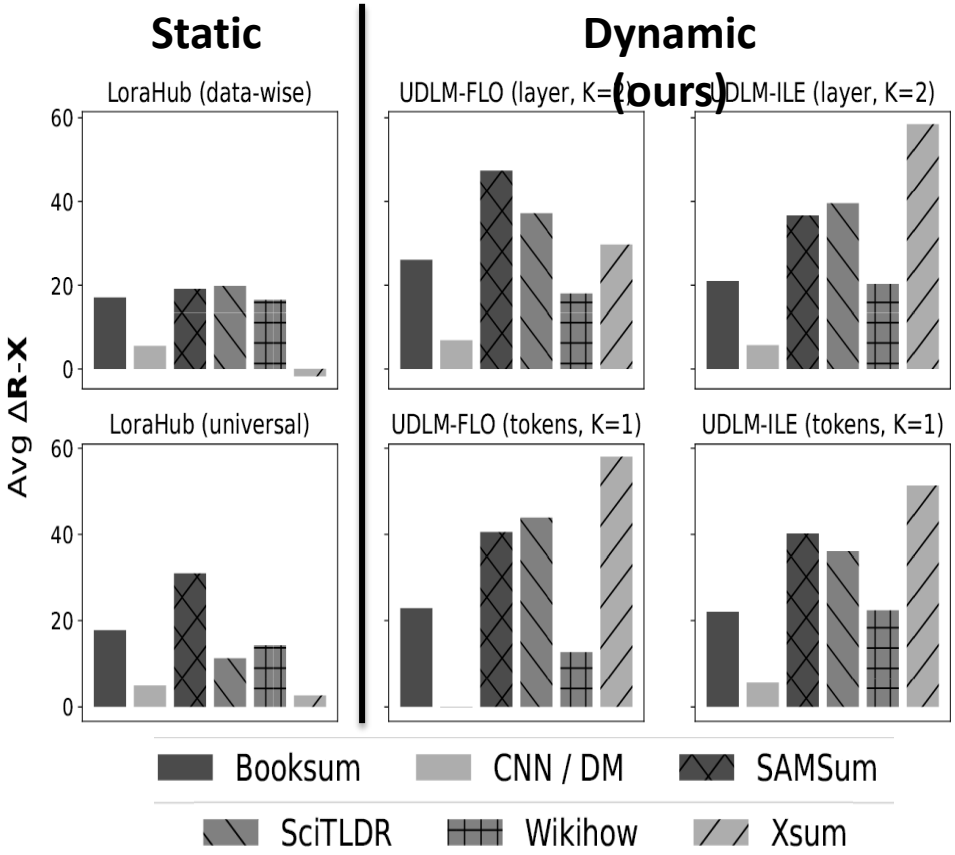
где h – текущий вход модели

Как настроить W_G без обучения?
использовать статистику активаций слоев модели:

$$W_G^i = \frac{\widehat{W}_G^i}{\|\widehat{W}_G^i\|}$$

$$\widehat{W}_G^i = AGG_{batch}\big(AGG_{seq}(L_{<l}(X))\big)$$

где $L_{<l}(X)$ – выходы слоя l модели для подвыборки данных X



*Сравнение методов ансамблирования при переносе знаний на другие домены в задаче аннотирования

*ILE и FLO – способы подсчета статистики выходов

Chernyshev D., "Improving Abstractive Summarization with Unsupervised Dynamic LoRA Mixtures" // under review in Lobachevski 2024.

Выводы - 1

- **Плохой стиль решения задач**
  - Если что-то не получается, то надо взять модель большего размера, и надеяться, что она будет «умнее»
- **Правильный подход**
  - Понимание того, «как все устроено» внутри LLM
  - Улучшение при необходимости базовых компонентов LLM:
    - Оптимальная настройка на русский язык
    - Формирование нейросетей меньшего размера, которые в нужных задачах демонстрируют такие же возможности, что и оригинальные большие LLM
  - Контроль галлюцинаций и дообучения с использованием графов знаний / лингвистических онтологий

Выводы - 2

- В условия большого количества настроечных параметров (включая подбор текстов, их очистка, векторизации, модификации архитектуры, модификации обучения, и т.п.) - для достижения результатов требуются массовые прогоны

- Необходима среда исследователей:
  - Поиск разных путей -- Конкуренция
  - Стенды для студентов
  - Обучение специалистов

- Гарантии выполнения бизнес-задач – госведомства, крупные корпорации и т.п.

Адаптация LLM путем настройки токенизации на русский язык

- Новый токенизатор обучался на русскоязычных текстовых данных, из-за чего любое слово представляется в среднем меньшим количеством токенов. Так, например, слово ‘**интеллект’** в новом токенизаторе представляется одним токеном, а в оригинальном разбивалось на три: [**'инте', 'лле', 'кт']**
- Показано, что алгоритм unigram (Kudo T. , 2018) для токенизации подходит лучше для русского языка, чем повсеместно применяемый BPE (Sennrich R., Haddow B., Birch A., 2016)
- Было установлено, что для адаптации модели достаточно заменить токенизацию, слой эмбеддингов и слой предсказаний и дообучить только новые слои на русскоязычных текстах
- Разработанный подход позволил повысить среднее качество на бенчмарке Russian Super Glue (Shavrina T. et al., 2020) по сравнению с исходной моделью с 0.68 до 0.7
- Также эксперименты показали, что unigram подход к токенизации позволяет достичь более высокого качества на прикладных задачах, чем BPE

Проблема: динамические ансамбли требуют обучения

- Для комбинации существующих моделей **(plug-and-play)** существуют только статические методы
- Однако эффективность лучших статических ансамблей (**LoRAHub**) существенно хуже динамических (**MOELoRA**)
- Но для динамических нужно дополнительно обучать матрицу комбинаций W_G

Model	CMeIE	CHIP-CDN	CHIP-CDEE	CHIP-MDCFNPC
ChatGPT	0.3058	0.6069	0.2838	0.5854
Huatuo	0.1826	0.3610	0.1658	0.3487
P-Tuning	0.4552	0.8687	0.5256	0.7423
LoRA (Full)	0.5089	0.8748	0.5464	0.7780
LoRA (Single)	0.4984	0.8882	0.5528	0.7765
LoRA (Full+TP)	0.4933	0.8814	0.5450	0.7705
LoRAHub	0.4411	0.8442	0.5041	0.7177
MOELoRA	**0.5193***	**0.8928***	**0.5697***	**0.7933***

*Сравнение методов ансамблирования на задачах обработки медицинских документов

Liu Q. et al. Moelora: An moe-based parameter efficient fine-tuning method for multi-task medical applications //arXiv preprint arXiv:2310.18339. – 2023.