Доверительный искусственный интеллект: честные языковые модели и где они обитают

Беликова Ю.А.

BIO

- Бакалавр ИУ9 МГТУ им. Баумана
- Магистр МФТИ по программе Методы и технологии ИИ (МТИИ)
- Исследователь в области обработки естественного языка Sber AI Lab

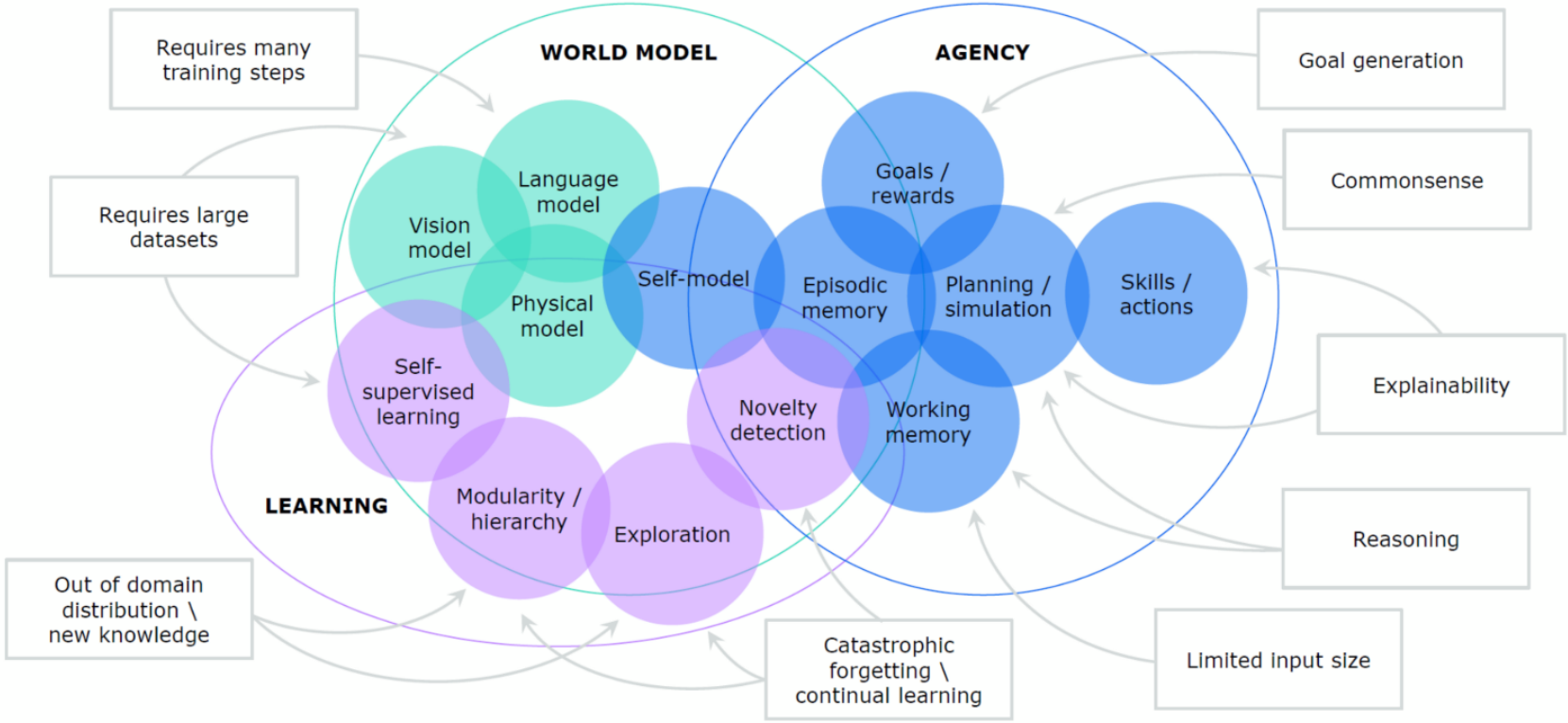
Почему именно эта магистратура?

Магистратура МТИИ [1] дает системный взгляд на разные области ИИ.

- **Обработка естественного языка (NLP):** *Как научить машину понимать и генерировать человеческий язык?*
  - Задачи: машинный перевод, создание чат-ботов, анализ тональности, суммаризация.
- **Компьютерное зрение (CV):** *Как научить машину "видеть" и интерпретировать визуальный мир?*
  - Задачи: распознавание объектов, сегментация изображений, системы для беспилотного транспорта.
- **Обучение с подкреплением (RL):** *Как научить агента принимать оптимальные решения для достижения цели?*
  - Задачи: обучение игровых ботов (AlphaGo), управление роботами, оптимизация логистики, дообучение языковых моделей.

[1] https://wiki.cogmodel.mipt.ru/s/mtai/doc/kursy-xbuoD9Zxcs

Системный взгляд на ИИ



Особенности исследовательского трека

Ключевые преимущества:

- **Погружение в науку:** Основной фокус программы — развитие навыков самостоятельного исследования.
- **Междисциплинарность:** Возможность изучать и применять методы из разных областей ИИ.
- **Публикационная активность:** Ожидается участие в международных конференциях, подготовка научных статей, работа в коллаборациях.

Вызовы и особенности:

- **Высокая самостоятельность:** Необходимость самому формулировать гипотезы, планировать эксперименты и анализировать результаты.
- **Плотная научная среда:** Высокая конкуренция, регулярные конференции.
- **Сложность совмещения с индустрией:** Программа требует полного погружения; совмещать с full-time работой вне R&D крайне сложно.

Советы

Как начать:

- Изучите или повторите основы (линейная алгебра, оптимизация и др.)
- Читайте современные статьи (конференции, журналы, arXiv)
- Участвуйте в соревнованиях (Kaggle, Codabench)
- Воспроизводите результаты известных работ

Исследовательские навыки:

- Критическое мышление и скептицизм
- Умение формулировать гипотезы
- Навыки экспериментального дизайна

Soft skills:

- Коммуникация: презентации, статьи, обсуждения
- Сотрудничество в команде
- Управление временем и проектами

Большие языковые модели

Определение

Большая языковая модель (Large Language Model, LLM) — параметрическая нейросетевая модель для генерации текста, построенная на архитектуре трансформера и отличающаяся большим числом параметров.

Ключевые компоненты:

- **Токенизация**: текст → последовательность токенов
- **Эмбеддинги**: токены → векторы фиксированной размерности
- **Блоки трансформера**: многоголовое внимание + feed-forward сети
- **Автогрессивная генерация**: предсказание следующего токена

Токенизация: от текста к числам

Пример токенизации (BPE/SentencePiece):

”Привет, как дела?”

↓

”При”, ”вет”, ”,”, ”как”, ”дела”, ”?”

↓

1234, 5678, 15, 891, 2345, 63

Ключевые принципы:

- **Разбиение на подстроки**: токены
- **Словарь**: фиксированный набор токенов (обычно 30K-100K)
- **Специальные токены**: [CLS], [SEP], [PAD], [UNK]
- **Обработка OOV**: незнакомые слова → подтокены

Эмбеддинги: превращение токенов в векторы

Формула эмбеддинга:

$$\mathbf{e}_i = \mathbf{E}[\text{token\_id}_i, :] \in \mathbb{R}^{d_{\text{model}}}$$

$\mathbf{E} \in \mathbb{R}^{|V| \times d_{\text{model}}}$ — обучаемая матрица эмбеддингов

Матрица эмбеддингов
$\mathbf{E}$
$|V| \times d_{\text{model}}$

$\xrightarrow{\text{lookup}}$

token_id
1234

$\Downarrow$

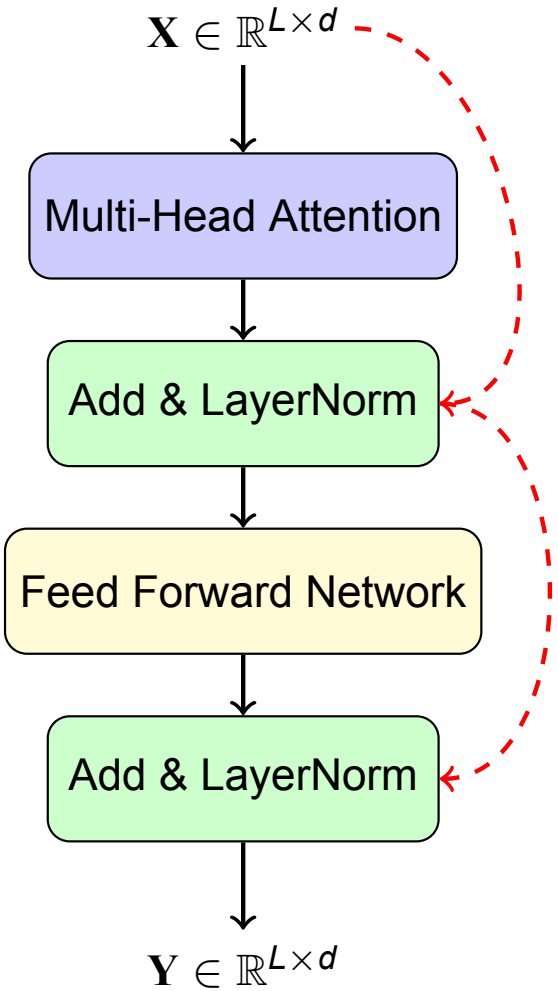
Вектор эмбеддинга
$[0.2, -0.1, 0.8, \ldots, 0.3]$
d_{model} чисел

Позиционное кодирование:

$$PE_{(pos,2i)} = \sin\left(\frac{pos}{10000^{2i/d_{\text{model}}}}\right) \quad PE_{(pos,2i+1)} = \cos\left(\frac{pos}{10000^{2i/d_{\text{model}}}}\right)$$

Итоговый вход: $\mathbf{x}_i = \mathbf{e}_i + \mathbf{pe}_i$

Архитектура блока трансформера



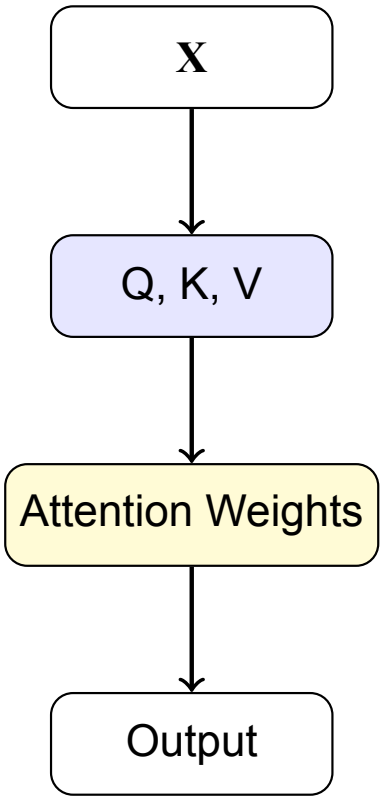
Ключевые операции:

- Multi-Head Attention: $\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V$
- FFN: $\max(0, \mathbf{x}\mathbf{W}_1 + \mathbf{b}_1)\mathbf{W}_2 + \mathbf{b}_2$
- LayerNorm: $\gamma \frac{\mathbf{x}-\mu}{\sigma} + \beta$

Особенности:

- Residual connections
- Нормализация после каждого слоя
- Параллельная обработка последовательности

Механизм внимания: как модель ”смотрит”



1. Q, K, V:

$$\mathbf{Q} = \mathbf{X}\mathbf{W}^Q$$

$$\mathbf{K} = \mathbf{X}\mathbf{W}^K$$

$$\mathbf{V} = \mathbf{X}\mathbf{W}^V$$

2. Веса внимания:

$$\mathbf{A} = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_k}}\right)$$

3. Итог:

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \mathbf{A}\mathbf{V}$$

Маскирование:

$$A_{i,j} = \begin{cases} -\infty, & j > i \\ \text{score}_{i,j}, & j \leq i \end{cases}$$

(для автогрессии)

Автогрессивная генерация: шаг за шагом

Процесс генерации:

Шаг 1	Шаг 2	Шаг 3	Шаг 4	Шаг 5
”Сегодня”	”погода”	”очень”	”хорошая”	”?”

$$P(\text{”погода”}|\text{”Сегодня”})$$
$$\downarrow$$
$$P(\text{”очень”}|\text{”Сегодня погода”})$$
$$\downarrow$$
$$P(\text{”хорошая”}|\text{”Сегодня погода очень”})$$
$$\downarrow$$
$$P(\text{”?”}|\text{”Сегодня погода очень хорошая”})$$

Математически:

$$P(y_1, ..., y_T|x) = \prod_{t=1}^{T} P(y_t|x, y_1, ..., y_{t-1})$$

На каждом шаге: подаем всю последовательность → получаем распределение → выбираем токен → повторяем

Стратегии декодирования

1. **Greedy Beam Search:**
   - $y_t = \arg\max_{w \in V} P(w|x, y_{<t})$
2. **Beam Search:**
   - Сохраняем k наиболее вероятных последовательностей
   - На каждом шаге расширяем каждую из них
   - Выбираем лучшую итоговую последовательность
3. **Sampling:**
   - **Temperature sampling:** $P'(w) = \frac{\exp(\text{logit}_w / T)}{\sum_v \exp(\text{logit}_v / T)}$
   - **Top-k sampling:** выбираем из k наиболее вероятных токенов
   - **Top-p sampling:** выбираем из токенов с суммарной вероятностью p

Стадии обучения языковой модели

1. **Предобучение (Pre-training):**
   - **Цель:** Выучить общие закономерности языка на терабайтах текста.
   - **Задача:** Предсказание следующего слова.

$$\mathcal{L}_{\text{pretrain}} = -\sum_{t=1}^{\ell} \log P_{\Theta}(y_t | x, y_{<t})$$

Стадии обучения языковой модели

1. **Предобучение (Pre-training):**
   - **Цель:** Выучить общие закономерности языка на терабайтах текста.
   - **Задача:** Предсказание следующего слова.

$$\mathcal{L}_{\text{pretrain}} = -\sum_{t=1}^{\ell} \log P_{\Theta}(y_t | x, y_{<t})$$

2. **Инструктивное дообучение (SFT):**
   - **Цель:** Научить модель следовать инструкциям на парах ”инструкция -> ответ”.

Стадии обучения языковой модели

1. **Предобучение (Pre-training):**
   - **Цель:** Выучить общие закономерности языка на терабайтах текста.
   - **Задача:** Предсказание следующего слова.

$$\mathcal{L}_{\text{pretrain}} = -\sum_{t=1}^{\ell} \log P_\Theta(y_t|x, y_{<t})$$

2. **Инструктивное дообучение (SFT):**
   - **Цель:** Научить модель следовать инструкциям на парах ”инструкция -> ответ”.
3. **Выравнивание (Alignment) через DPO:**
   - **Цель:** Сделать ответы полезными, честными и безвредными.
   - **Метод:** Direct Preference Optimization (DPO) учит модель предпочитать ”хорошие” ответы (y_+) ”плохим” (y_-).

$$\mathcal{L}_{\text{DPO}} \propto -\log\sigma\left(\beta\log\frac{\pi_\theta(y_+|x)}{\pi_{\text{ref}}(y_+|x)} - \beta\log\frac{\pi_\theta(y_-|x)}{\pi_{\text{ref}}(y_-|x)}\right)$$

Галлюцинации языковых моделей

Определение

Галлюцинации – это случаи, когда языковая модель генерирует правдоподобный, но **фактически неверный** или несуществующий факт.

Примеры:

- Модель уверенно придумывает несуществующие научные термины, цитаты, имена, даты.
- Генерация ссылок на статьи или книги, которых не существует.
- Ответы на вопросы, на которые нет информации в параметрах модели, но она всё равно ”выдумывает” ответ.

Решение – Генерация с дополненной выборкой

Определение

Генерация с дополненной выборкой (Retrieval Augmented Generation, RAG) — подход, комбинирующий LLM с внешней системой извлечения знаний.

Архитектура RAG:

1. **Retrieval**: поиск релевантных документов $D = \{d_i\}$
2. **Augmentation**: расширение контекста вопроса q документами D
3. **Generation**: генерация ответа a на основе (q, D)

Решение – Генерация с дополненной выборкой

Определение

Генерация с дополненной выборкой (Retrieval Augmented Generation, RAG) — подход, комбинирующий LLM с внешней системой извлечения знаний.

Архитектура RAG:

1. **Retrieval**: поиск релевантных документов $D = \{d_i\}$
2. **Augmentation**: расширение контекста вопроса q документами D
3. **Generation**: генерация ответа a на основе (q, D)

Но... RAG не является панацеей. Модели все равно галлюцинируют, даже имея перед глазами правильный ответ.

Примеры контекстуальных галлюцинаций

QUESTION: how to prepare to get an ultrasound?

REFERENCE: Passage 1: 1 If you are just interested in checking the gender, you can go as early as 17 weeks. 2 Any earlier, and the results may not be accurate, leading to an extra coat of paint for your bedroom. 3 To get good pictures, 3D ultrasounds are best performed between 20 and 32 weeks, and the best pictures are between 24 and 30 weeks.

...

RESPONSE: Based on the given passages, here's how to prepare for an ultrasound:

* For a gender reveal ultrasound, wait until at least 17 weeks of pregnancy to ensure accurate results.

* For 3D ultrasounds, schedule the appointment between 20 and 32 weeks of pregnancy for the best pictures ...

a

- **Evident Conflict**: прямое противоречие контексту
- **Evident Introduction**: добавление необоснованной информации
- **Subtle Conflict**: изменение смысла контекста
- **Subtle Introduction**: домысливание деталей

[a] Wu и др., *RAGTruth: A Hallucination Corpus for Developing Trustworthy Retrieval-Augmented Language Models*, 2023.

Вопрос для размышления

Если LLM обучены предсказывать следующий токен на основе контекста, почему они генерируют информацию, противоречащую этому контексту?

Вопрос для размышления

Если LLM обучены предсказывать следующий токен на основе контекста, почему они генерируют информацию, противоречащую этому контексту?

Возможные гипотезы:

- Конфликт между параметрическими знаниями и контекстом
- Ошибки в механизме внимания
- Проблемы с пониманием инструкций
- Переобучение на некорректных данных
- Фундаментальные ограничения архитектуры

Ключевой вопрос

Как можно препятствовать контекстуальным галлюцинациям?

Почему LLM галлюцинируют?

Фундаментальные причины:

- **Архитектурные ограничения**: автогрессивная природа генерации
- **Данные обучения**: противоречия и неточности в корпусах
- **Переобучение**: запоминание вместо понимания
- **Проблема выравнивания**: несоответствие целей обучения и использования

Особенности в RAG:

- Конфликт между параметрическими знаниями и контекстом
- Неспособность признать незнание
- ”Уверенность” в неверных фактах

Важно

Галлюцинации критичны в медицине, праве, финансах — областях, где ошибки недопустимы

Подходы к борьбе с галлюцинациями

Black-box методы:

- Внешняя проверка фактов
- SelfCheckGPT
- FactScore
- Специальные схемы запросов
- Ансамблирование моделей

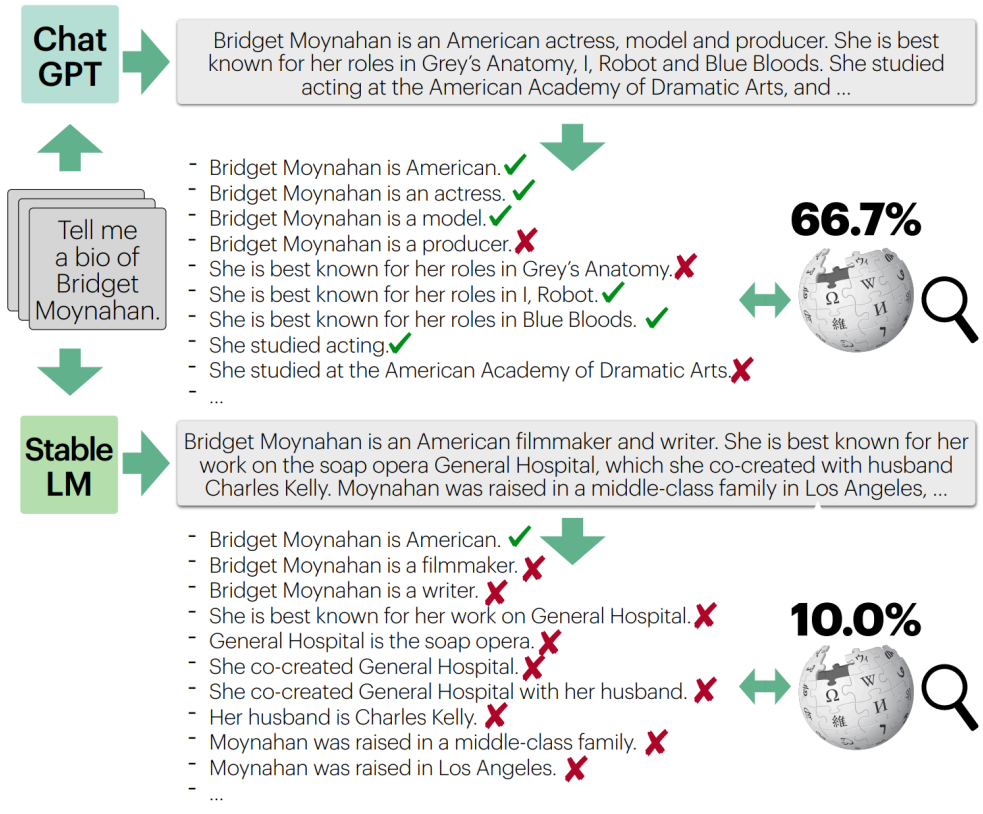
Плюсы: работают с любыми моделями
Минусы: вычислительные затраты

White-box методы:

- Анализ внутренних состояний
- Карты внимания
- Активации нейронов
- Управляемое декодирование
- Коррекция скрытых состояний

Плюсы: эффективность
Минусы: требуют доступа к модели

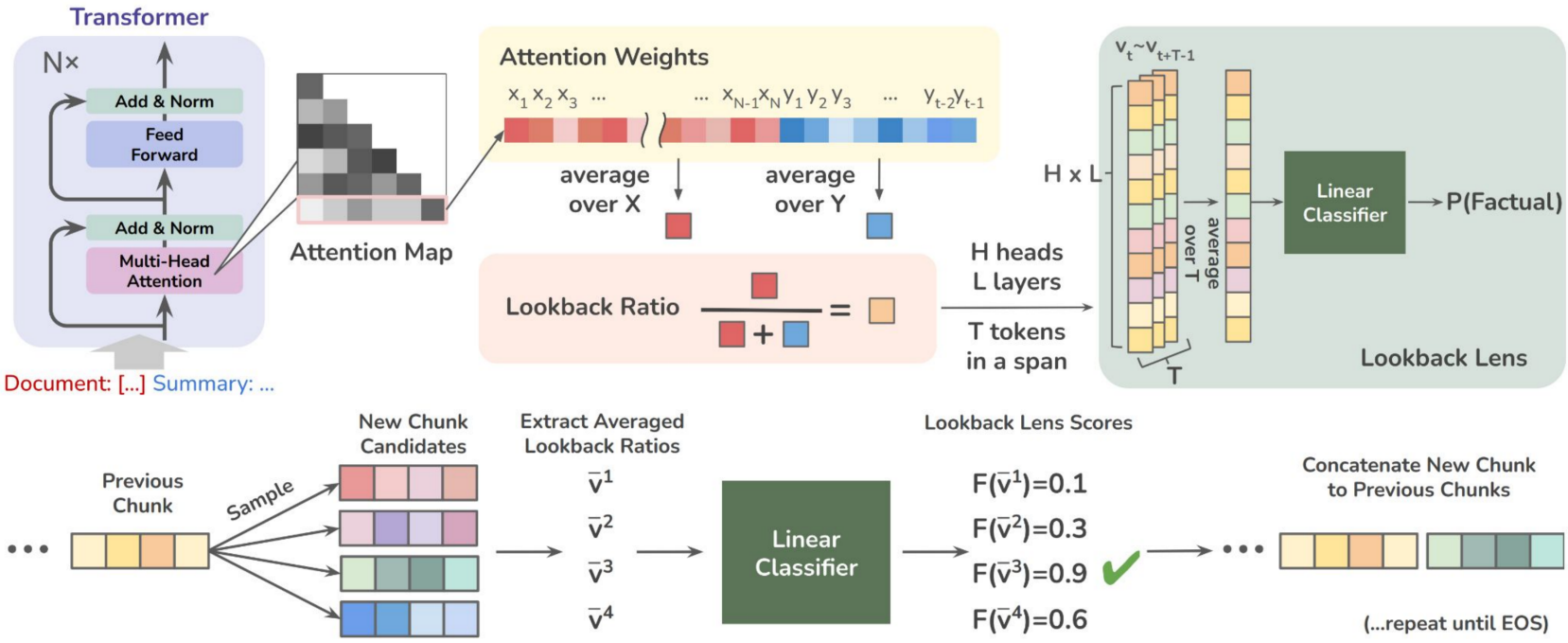
Black-box методы детекции галлюцинаций



FactScore[2]

[2] Min и др., *FActScore: Fine-grained Atomic Evaluation of Factual Precision in Long Form Text Generation*, 2023.

White-box методы детекции галлюцинаций



Lookback Lens[3]

[3] Chuang и др., *Lookback Lens: Detecting and Mitigating Contextual Hallucinations in Large Language Models Using Only Attention Maps*, 2024.

Моя магистерская работа

Цель: разработать white-box методологию для детекции и минимизации контекстуальных галлюцинаций

Методология:

1. **Детекция галлюцинаций**
   - Необучаемый подход: ансамбль топологического анализа + неопределенность
   - Обучаемый подход: пробинг внутренних состояний
2. **Минимизация галлюцинаций**
   - Управляемое декодирование с выбором кандидатов
   - Дообучение методом DPO на автоматических парах предпочтений

Метод №1: Необучаемая детекция

Как поймать ”ложь” без разметки данных? Первый предложенный мной метод не требует обучения и основан на двух сигналах изнутри модели:

1. **Топологический анализ карт внимания (MTopDiv):**
   - Строим граф, показывающий, на какие слова из контекста модель ”смотрит”. Если ответ ”топологически оторван” от контекста, это признак галлюцинации.
2. **Оценка неопределенности (Uncertainty):**
   - Измеряем ”уверенность” модели в каждом сгенерированном токене (через энтропию). Высокая неопределенность — высокий риск галлюцинации.

Результат

Объединяем эти два сигнала в единый скор. Высокий скор = вероятная галлюцинация.

Метод №2: Обучаемая детекция

Второй подход — обучить небольшой классификатор, который предсказывает галлюцинации по внутренним состояниям LLM.

- **Признаки для классификатора:**
  - *Агрегированные скрытые состояния:* Средние, максимальные значения векторов, представляющих слова.
  - *Lookback признаки:* Соотношение внимания, которое модель уделяет контексту, по сравнению с уже сгенерированным текстом.
- **Классификаторы:**
  - Использовались как классические модели (CatBoost), так и современные подходы для табличных данных (TabPFN), которые отлично работают на малых объемах данных.

Метод №3: Минимизация галлюцинаций

После детекции мы можем заставить модель быть честнее.

1. **Управляемая генерация (Ensemble Guided Decoding):**
   - Генерируем несколько вариантов ответа.
   - С помощью нашего детектора оцениваем ”уровень галлюцинаций” в каждом из них.
   - Выбираем самый ”честный” вариант.
2. **Дообучение через DPO:**
   - Используя наш детектор, автоматически создаем обучающие пары: ”хороший” (y_+) и ”плохой” (y_-) ответы.
   - Дообучаем LLM по методу DPO, чтобы она научилась генерировать ответы, похожие на y_+, и избегать ответов, похожих на y_-.

Что получилось в итоге?

Сравнение метрики Faithfulness (в %, чем выше — тем лучше) для Llama-2-7b на датасете CoQA.

Метод	Faithfulness (%)
Стандартная генерация (Greedy)	70.7
Выбор лучшего кандидата (Ensemble+)	81.9
Наша модель после дообучения (DPO)	**83.5**

Вывод: Предложенная методология значительно повышает фактическую точность (*Faithfulness*) ответов LLM.

Ключевой результат

Мы можем не только детектировать, но и эффективно исправлять галлюцинации, дообучая модель на автоматически созданных данных.

Пример: анализ галлюцинаций в ответах LLM

Контекст:

- Компания была основана в Калифорнии. В открытых источниках не указана точная дата основания. Основатели — Джон Смит и Мария Иванова.

Вопрос:

- Когда и где была основана компания, кто её основатели?

Исходный ответ модели

Компания была основана в 1985 году в Калифорнии. Её основатели — Джон Смит и Мария Иванова. Компания быстро вышла на рынок Европы.

- Явное добавление: 1985 год — не указан в контексте
- Скрытое добавление: выход на рынок Европы — не подтверждено
- Оценка ансамбля детекции: 0.89

Скорректированный ответ

Компания основана в Калифорнии. Основатели — Джон Смит и Мария Иванова.

- Ensemble-Guided Fine-tuning
- Faithfulness score: 0.97

Будущие направления исследований

Технические вызовы:

- Масштабирование на модели 10B+ параметров
- Работа с длинными контекстами (4K+ токенов)
- Динамическая коррекция во время генерации
- Real-time DPO адаптация

Методологические направления:

- Каузальное понимание механизмов галлюцинаций
- Интерпретируемость внутренних состояний
- Универсальные детекторы для разных архитектур
- Этические аспекты доверительного ИИ

Индустриальные применения:

- Валидация ИИ-систем для критических областей
- Стандарты надежности для LLM
- Регулирование использования ИИ

Ключевые выводы

- LLM совершили прорыв, но галлюцинации – серьезная проблемой
- White-box подходы показывают многообещающие результаты
- Доверительный ИИ критически важен для практических применений
- Исследования в этой области активно развиваются

Спасибо за внимание!

Вопросы?

Беликова Ю.А.
Email: belikova.iua@phystech.edu