Архтектурные и программные методы исключения галлюционаций больших языковых моделей (LLM)

Докладчик: Ваганов Дмитрий Валерьевич,

инженер компании «Сириус-Софт», преподаватель технических дисциплин

(ТОГУ, Хабаровский институт инфокоммункаций)

Краткое содержание

Почему современные LLM, такие как ChatGPT, DeepSeek и другие, иногда выдают убедительные, но ложные ответы?

Рассмотрим природу галлюцинаций, классифицируем и предложим 3 взаимодополняющих метода их устранения:

1. **RAG (доступ к данным)**
2. **Prolog (логический вывод)**
3. **SNN (спайковые нейросети).**

Таблица галлюцинаций LLM — причины и методы устранения

Тип галлюцинации	Причина	Метод устранения
Статистическая	LLM предсказывает слова, основываясь на статистике, а не по логике	Prolog (логический вывод вместо предсказания)
Контекстная	Нет внутренней динамики, контекст — плоский список	SNN (временное кодирование, память)
Логическая	Нет встроенного понятия времени и причинно-следственных связей на основе формальной логики	Prolog + SNN (трасса логического вывода + временные шаблоны)
Данных пациента	LLM не имеет доступа к ЭМК конкретного пациента	RAG Retrieval-Augmented Generation — генерация, дополненная (связь с базой данных — медицинской картой пациента)
Актуальности	LLM обучена на статических данных (до определенной даты)	RAG (обновляемые протоколы и рекомендации по новым методикам лечения)
Уверенности	LLM обязана дать ответ, не может сказать «не знаю»	Prolog (может отказаться от ответа)
Объяснимости	Нет трассы логического вывода — только имитация объяснения	Prolog (трасса логического вывода)

Что такое LLM и почему она галлюцинирует

Суть проблемы:

LLM — статистический прибор, а не логический. Это генератор правдоподобного текста.Большие языковые модели (LLM) обучены на миллиардах текстов. Они предсказывают следующее слово на основе статистических закономерностей.

Как думают LLM:

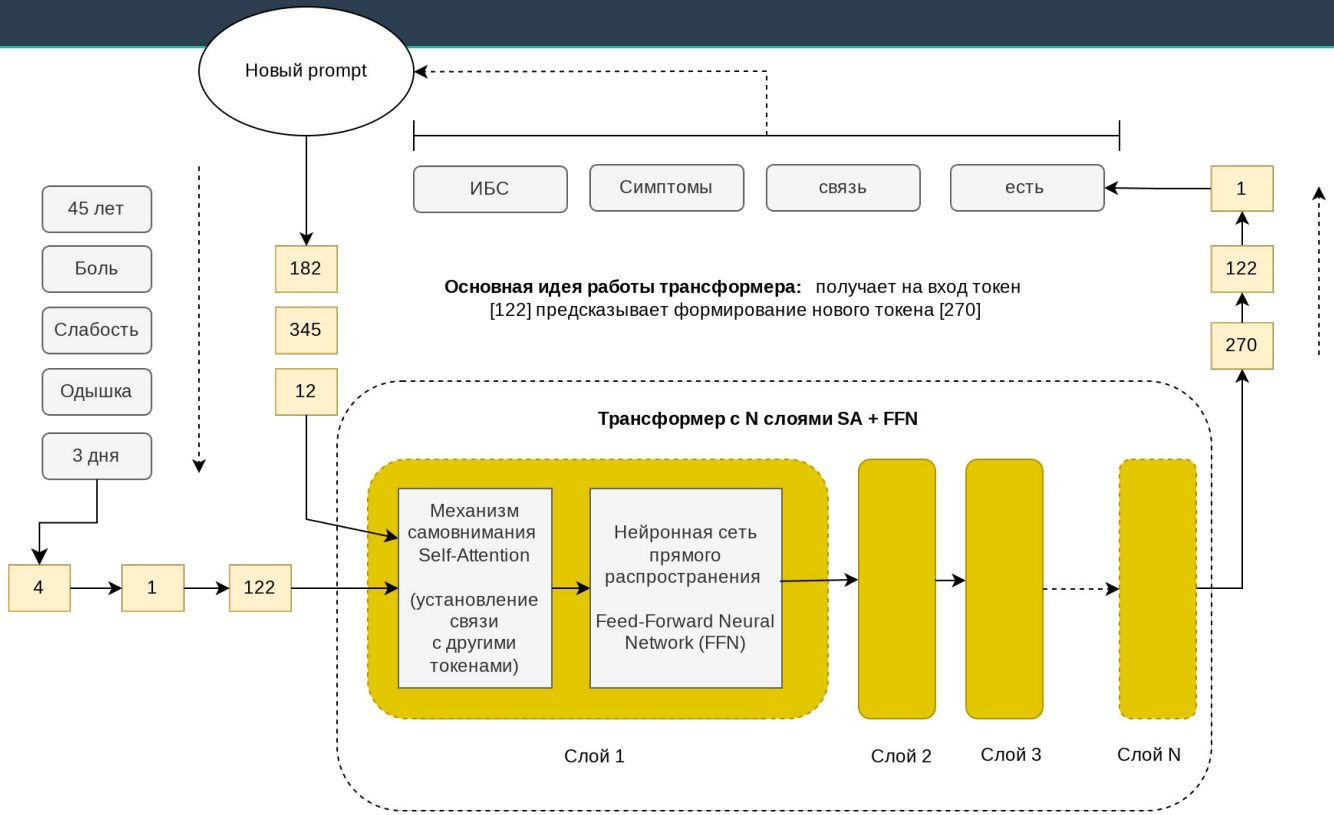
Видит фразу: «Пациент с болью в груди и одышкой...» LLM Вспоминает: в 70% текстов после этих слов шла «пневмония», в 20% — «ТЭЛА», в 10% — «перикардит». Выбирает самый частый вариант: «пневмония»

Проблема:

LLM не знает, что ТЭЛА опаснее пневмонии. Она не «рассуждает» — она подбирает статистически правдоподобное продолжение. Это и есть корень галлюцинаций.

LLM выдает ответы, которые ТОЛЬКО ВЫГЛЯДЯТ как работа эксперта. Это создает иллюзию надежности.

Схема генерации ответа LLM



Язык программирования логики Prolog

«PROgrammation en LOGique» (программирование в логике)

Его основными разработчиками являются:

Ален Кольмероэ (Alain Colmerauer) и его группа в Университете Марселя (Франция) — 1972 г.
Роберт Ковальский (Robert Kowalski) из Эдинбургского университета (Шотландия), который разработал теоретическую интерпретацию логики предикатов как языка программирования, являющейся основой Prolog.

База знаний: семейные отношения

Факты — то, что известно о семье:

```
% === ФАКТЫ (База данных) ===

% Родители: родитель(Кто, Кому)
родитель(иван, анна).
родитель(иван, петр).
родитель(мария, анна).
родитель(мария, петр).

% Пол: мужчина(Имя), женщина(Имя)
мужчина(иван).
мужчина(петр).
женщина(мария).
женщина(анна).
```

```
% === ПРАВИЛА ===

% Сестра: X является сестрой Y, если у них общий родитель, X — женщина и X не является самим Y.

сестра(X, Y) :-
    родитель(Z, X),
    родитель(Z, Y),
    женщина(X),
    X \= Y. % Проверка, чтобы не быть сестрой самому себе
```

Трасса логического вывода — показывает цепочку рассуждений.
Prolog быстрее LLM в 500 – 2000 раз при решении простых логических задач на малых и средних базах фактов

Шаг	Действие	Состояние перменных	Резульат	Что использовалось? Факт или правило?
1	Вызов сестра(X, петр)	Y = петр	Переход к телу правила	Правило сестра(X, Y)
2	Поиск родителя Z для петр	Y = петр, Z = иван	Успех	Факт родитель(иван, петр)
3	Поиск X, где родитель(иван, X)	Z = иван, X = анна	Успех (1-й вариант)	Факт родитель(иван, анна)
4	Проверка женщина(анна)	X = анна	Успех	Факт женщина(анна)
5	Проверка анна \= петр		Успех	Встроенный предикат
6	Фиксация решения	X = анна	Решение №1	Конец ветки
7	Поиск другого X для иван	Z = иван, X = петр	Найдено	Факт родитель(иван, петр)
8	Проверка женщина(петр)	X = петр	Отказ	Нет факта женщина(петр)
9	Смена родителя Z для петр	Y = петр, Z = мария	Успех	Факт родитель(мария, петр)
10	Поиск X, где родитель(мария, X)	Z = мария, X = анна	Успех	Факт родитель(мария, анна)
11	Повтор проверок для анна	X = анна	Успех	См. шаги 4-5
12	Фиксация решения	X = анна	Решение №2	Конец ветки

Prolog: логический вывод в медицине

Факты: опрос врача + сведения из медицинской базы

```
% Симптомы
симптом(боль_в_груди, есть).
симптом(одышка, есть).

% Факторы риска
фактор_риска(недавняя_операция).
фактор_риска(варикоз).

% Возраст
возраст(65).

% Аллергия (из ЭМК пациента)
аллергия(гепарин).
```

Правила: из клинических рекомендаций

```
% Диагноз: ТЭЛА
диагноз(ТЭЛА) :-
    симптом(боль_в_груди, есть),
    симптом(одышка, есть),
    (фактор_риска(тромбоз);
фактор_риска(варикоз)),
    возраст(Возраст), Возраст > 60.
```

Запрос:

```
?- диагноз(пациент, Что).
```

Трасса вывода (что видит врач):

1. Проверяем: боль в груди? → ДА
2. Проверяем: одышка? → ДА
3. Проверяем: фактор риска тромбоза? → ДА (недавняя операция)
4. Проверяем: возраст > 60? → 65 > 60 → ДА
5. → ДИАГНОЗ: ТЭЛА
6. Дополнительно: аллергия на гепарин → антикоагулянты: НЕ гепарин

Реальные медицинские системы на Prolog

Prolog — это не теория, а рабочий инструмент. 40 лет назад он диагностировал лейкемию, сегодня — синдром Гийена-Барре.

Система	Область	Результат
Диагностика лейкемии (1987)	Онкология	Протестирована на 400 случаях, выдает объяснение
Диагностика синдрома Гийена-Барре (2023)	Неврология	Точность 83%
Видео-ЭЭГ мониторинг (2025)	Реанимация	Выявление ишемии в многодневных ЭЭГ-сигналах

Слабые стороны Prolog (почему нужна LLM ?): Prolog не понимает человеческий язык

У Prolog есть фундаментальная проблема для реальной медицины: он не понимает синонимы и требует строгой формы ввода. Чтобы Prolog понял «Анна — женщина», мы ввели факт женщина(анна). Но что, если врач скажет: «Анна — женского пола» «Анна — девушка» «Анна — женского рода» «Пациентка Анна».

Проблема: Prolog не поймет ни одного из этих вариантов, если мы заранее не пропишем все возможные синонимы. Пример из медицинской диагностики: Чтобы Prolog понял «боль в груди», нужно прописать все варианты: «болит за грудиной»,«дискомфорт в торакальной области»,«колет в груди»>«давит в грудной клетке» «жжение за грудиной» ... и так миллионы вариантов! **Это невозможно!**

Решение проблемы: комбинация Prolog и LLM: Prolog — идеальный логик, но он «глухой и слепой». LLM — идеальный «переводчик» с человеческого языка. Они дополняют друг друга.

Что не умеет Prolog	Что умеет LLM
Понимать синонимы	Распознает: «колет в груди» = «боль в груди»
Понимать свободную речь	Врач говорит обычным языком
Извлекать структуру из текста	Превращает рассказ в структурированные факты

RAG (Retrieval-Augmented Generation) — технология, которая позволяет LLM получать актуальные данные перед генерацией ответа

Три шага RAG:
1. Поиск — LLM обращается к внешней базе данных (ЭМК пациента, клинические рекомендации, протоколы больницы)
2. Дополнение — найденные данные добавляются в контекст запроса
3. Генерация — LLM отвечает с учетом этих данных

Пример из практики:

Без RAG:
LLM: «Рекомендован гепарин» (стандартный ответ)

При использовании RAG (из ЭМК пациента):
RAG находит: «У пациента аллергия на гепарин»
LLM: «Рекомендованы прямые пероральные антикоагулянты (апиксабан, ривароксабан) — у пациента аллергия на гепарин»

Спайковые нейросети (SNN) — возвращение к принципам природы

Спайковые нейронные сети (SNN) — архитектура искусственных нейросетей, которая физически повторяет работу биологического нейрона. Нейроны активны только когда нужно (1–2%), как в мозге. Это называется событийной обработкой — вычисления происходят только при поступлении сигнала (спайка)

Принцип работы SNN

Параметр	LLM (трансформер)	SNN	Мозг
Активность нейронов	Все активны постоянно	1–2% активны в момент	1–2% активны в момент
Период рефрактерности	Нет	Есть (как диастола)	Есть
Энергопотребление	Мегаватты	Микроватты — милливатты	20 ватт

Как решают спайковые сети проблемы с галлюцинациями?

Тип галлюцинации	Как SNN помогает?
Контекстная (потеря нити рассуждения)	Временное кодирование спайков создает «память» о предыдущих событиях
Логическая (нарушение причинности)	Временные паттерны (сначала А, потом Б) кодируют причинно-следственные связи

Применение SNN в медицине:

Область	Результат
Сегментация опухолей мозга (глиома)	Точность 92 %
Рентген грудной клетки у детей	Точность 96%, специфичность 1,0 при пневмонии
Анализ биосигналов при боли в спине	Точность 80%, энергоэффективная обработка

SNN добавляет в ИИ то, чего у LLM нет от природы — чувство времени и причинности, работая при этом на микроваттах.

Сложности обучения SNN и аппаратные реализации

Аппаратные реализации ЕСТЬ по всему миру (включая РФ)

Чип (платформа)	Страна	Характеристики	Энергопотребление	Статус
Intel Loihi 2	США	1 млн нейронов, 128 млн синапсов, 2,3 млрд транзисторов, 31 мм²	10–100 мВт/чип	Исследовательский
BrainChip Akida	Австралия	Цифровой событийный AI-процессор, обучение на чипе	Суб-мВт — мВт	Коммерческий
Innatera Pulsar	Нидерланды	Первый массово вупускаемый нейроморфный микроконтроллер, RISC-V CPU + CNN-акселератор + SNN-акселератор	< 1 мВт (суб-милливатт)	Коммерческий (2025)
SpiNNaker2	Германия	152 ARM-ядра на чип, 48 чипов на серверную плату, симулирует 150–180 млн нейронов	~100 кВт (система целиком)	Развернут в Sandia Labs (США)
DYNAP-SE2	Китай / Швейцария	Событийный процессор для ЭЭГ, uW-класс, аналого-цифровой гибрид	Микроватты (uW)	Производство в Китае, штаб-квартира в Швейцарии
«Алтай»	Россия	Нейроморфный процессор от «Мотив» и «Лаборатории Касперского»	0,5 Вт	Прототип (нет коммерческого образца)

Сложности обучения SNN и решение на основе волновой памяти и спайковых нейросетей

Спайковые нейронные сети (SNN) — архитектура, которая физически повторяет работу биологического нейрона. Нейроны активны только когда нужно (1–2%), как в мозге. Это называется событийной обработкой — вычисления происходят только при поступлении сигнала (спайка).

Способ обучения	Описание	Связь с волновой памятью	Стоимость
ANN-to-SNN Conversion	Сначала обучается обычная нейросеть (ANN), затем конвертируется в спайковую (ANN)	НЕТ СВЯЗИ	**Высокая** (за счет использования дорого обоудования)
Прямое программирование архитектуры	Архитектура сети (пейсмекеры/детекторы) задается так, чтобы колебательные режимы формировались естественно — подобно тому как это происходит в мозге	Результат работ Кащенко Сергея Александровича, Майорова Вячеслава Владимировича в книге «**Модели волновой памяти**» **2009 г.**	**Низкая** (Вместо итеративной настройки миллионов весов метод волновой памяти предлагает архитектурное проектирование. Процесс сводится не к обучению, а к программированию сети путем настройки параметров связи между нейронами для формирования нужных колебательных режимов (паттернов)

Модель волновой памяти — ПРОРЫВ в области обучения нейросетей

Российские ученые С.А. Кащенко и В.В. Майоров предложили альтернативный подход, изложенный в монографии «Модели волновой памяти».

Идея	Обяснение
Нейрон как автогенератор	Нейроны генерируют спайки даже без внешнего сигнала, работая в автономном режиме
Пейсмекеры	Группы нейронов, которые задают ритм всей сети, синхронизируя спайки
Детекторы	Группы нейронов, которые подавляются пейсмекерами и распознают паттерны
Кольцевая структура	Однородные нейронные модули, соединенные в кольцо
Вместо обучения	Проектирование архитектуры. Информация хранится в пространственно-временной структуре спайков, а не в весах.

Стоимость: тысячи раз дешевле (не требует GPU-кластеров)

Как LLM, Prolog и SNN дополняют друг друга

Параметр	LLM	Prolog	SNN
Понимает речь врача	да	Нет	Нет
Распознает синонимы	да	Нет	Нет
Логический вывод	Нет	Да	Нет
Трасса вывода	Нет	Да	Нет
Энергоэффективность	нет (нужна отдельная электростанция)	Да (микроваты)	Да (микроваты)
Временное кодирование	Нет	Нет	Да

Будущее систем искуссвенного интеллекта: гибрид LLM, RAG, Prolog, SNN

Параметр	Роль
LLM	Переводчик: понимает речь врача, извлекает факты
Prolog	«Логик» — выводит диагноз, показывает трассу, дает гарантии
RAG	Связь с реальными данными пациентов
SNN	«Ускоритель» — энергоэффективная обработка, временное кодирование