Детекция галлюцинаций больших языковых моделей

Левыкин Александр Михайлович
Научный руководитель: Воронцов Константин Вячеславович

МГУ им. М.В.Ломоносова

22 ноября 2024 г.

Понятие галлюцинации и их классификация

- Галлюцинация LLM - ответ (или его часть) модели, не соответствующий входным данным (промпту) либо сгенерированному ранее контексту, либо противоречащий общеизвестным фактам о мире.
- Галлюцинации делятся на две части:
  - Intrinsic галлюцинации (внутренние галлюцинации) - генерируемый текст не соответствует входным данным.
  - Extrinsic галлюцинации (внешние галлюцинации) -
    - генерируемый текст не соответствует фактам реального мира. (в задачах, где нет референсного документа)
    - генерируемый текст не может быть подтвержден информацией, содержащейся во входных данных. (в задачах, где есть референсный документ)
- В данной работе - фокус на внешние галлюцинациях

Классификация задач детекции галлюцинаций

Подходы детекции галлюцинаций бывают:

- reference-based - когда есть документ-источник (суммаризация, переводчики, image caption и т.д.)
- reference-free (question answering - наша задача)

Подходы детекции галлюцинаций бывают:

- online - при классификации i-го токена можно смотреть только на $1, .., i$-е токены.
- offline - можно смотреть на весь текст.

Подходы детекции галлюцинаций бывают:

- document-level - классифицируем весь текст одной меткой.
- token-level - классифицируем каждый токен по-отдельности.

В данной работе - фокус на token-level reference-free задаче.

Постановка задачи

- Пусть заданы:
  - Запрос пользователя (query, промпт, задача) - последовательность токенов $Q = \{q_1, ..., q_m\}$, где m - длина последовательности.
  - Ответ модели - последовательность токенов $X = \{x_1, x_2, ..., x_n\}$, где n - длина последовательности.
- Задача: поиск всех пар вида (i_k, j_k), где i_k и j_k, такие что $i_k \leqslant j_k$ - индексы начала и конца текстового фрагмента $\overline{x_{i_k} x_{i_k+1} ... x_{j_k}}$, являющегося галлюцинацией.
  Предполагается отсутствие вложенных сущностей, поэтому постановка выше эквивалентна следующей:
- Задача: классификация каждого токена ответа $X = \{x_1, x_2, \ldots, x_n\}$ на один из 3-х классов $y_i \in \{B, I, O\}$, где B - начало галлюцинации, I — продолжение, а O — токен, не являющийся частью галлюцинации.

Предложенный метод 1. Instruction-based подход

Метод опирается на “prompt engineering”, и “emergent behavior”.
Данные в задаче - наборы пар: запрос пользователя $Q = \{q_1, q_2, \ldots, q_m\}$ и ответ модели $X = \{x_1, x_2, \ldots, x_n\}$.
Задача: создать промпт P для генеративной LLM, на котором достигается максмум по метрикам качества на валидационной выборке.
Преимущества:

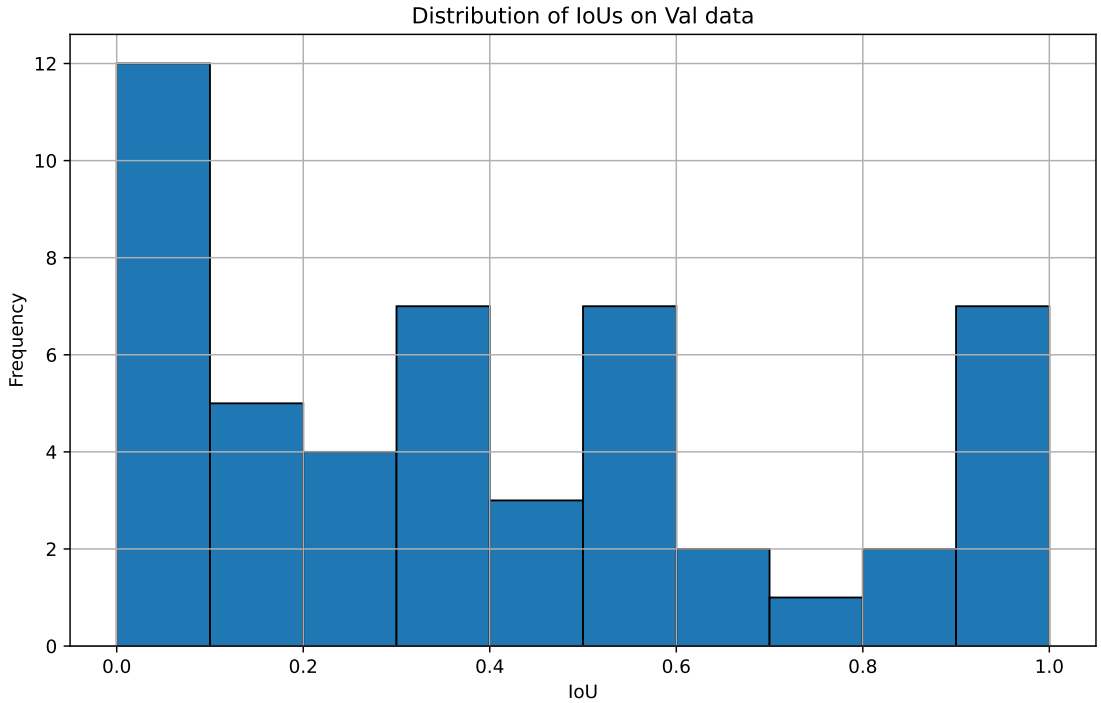
- **Адаптивность:** подбор промптов позволяет решать широкий класс задач.
- **Экономичность:** не нужно дообучать модель, что снижает затраты на дополнительные данные и вычислительные ресурсы.
- **Интерпретируемость:** промптом можно потребовать пояснение галлюцинации.

Недостатки:

- **Чувствительность к промпту**
- **Отсутствие вероятностей** $\Rightarrow$ сложно управлять степенью уверенности.
- **Зависимость от масштабов модели:** требуются очень крупные модели, что увеличивает временные затраты.

Instruction-based подход. Эксперимент

Данные: SemEval-2025 Validation Set на английском языке (50 троек: (вопрос, ответ, фрагменты_галлюцинации).
Модель: Meta-Llama-3.1-8B-Instruct (release - 2024).
Лучший результат: 39.7% по IoU.



Предложенный метод 2. Обучение LSTM

Для данных, в которых помимо сгенерированного ответа модели известны логиты каждого токена, можно обучить модель, анализирующую лишь временной ряд логитов.
Архитектура модели включает слой LSTM, после которого добавляется линейный слой, понижающий размерность до 1.
Модель обучается с использованием функции потерь BCE (Binary Cross-Entropy Loss), которая минимизируется по следующей формуле:

$$\text{BCE Loss} = -\frac{1}{N}\sum_{i=1}^{N}\left(y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)\right)$$

где:

- N — количество токенов в обучающей выборке,
- y_i — истинная метка для токена i (1 для галлюцинации, 0 для корректного токена),
- $\hat{y}_i$ — предсказанная моделью вероятность того, что токен i является галлюцинацией.

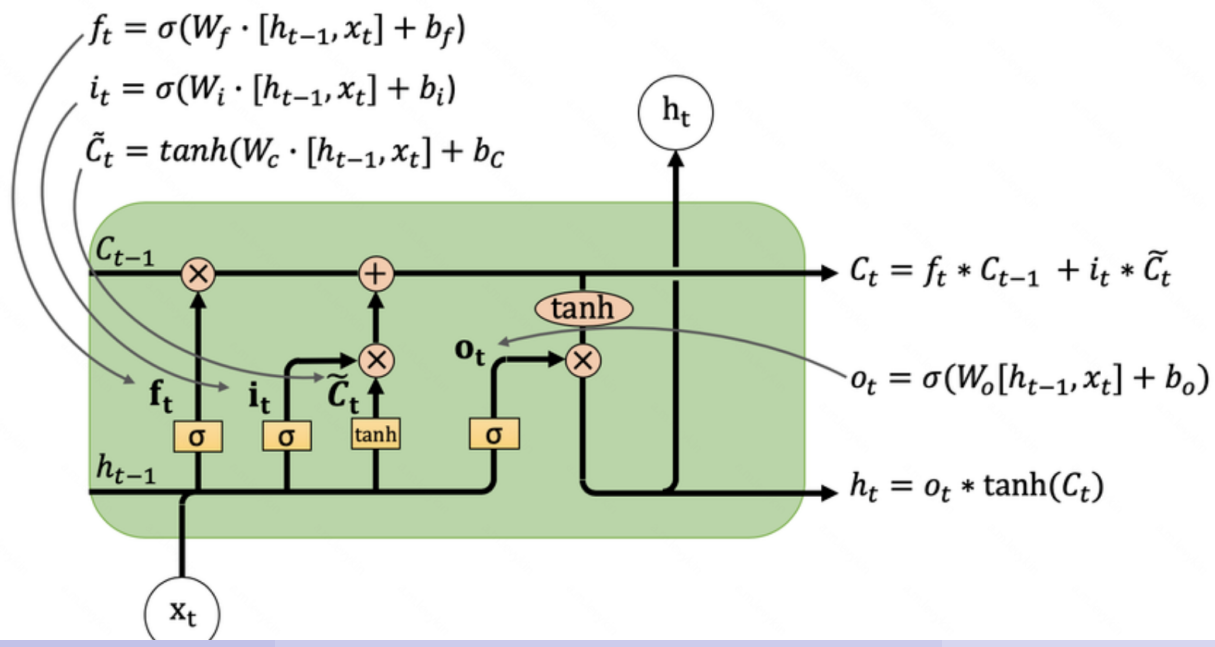
Предложенный метод 2. Обучение LSTM

Плюсы подхода:

- **Свобода от языка:** модель анализирует временные зависимости без учёта лексики ⇒ снижение влияния языковых особенностей и концентрация на статистике.

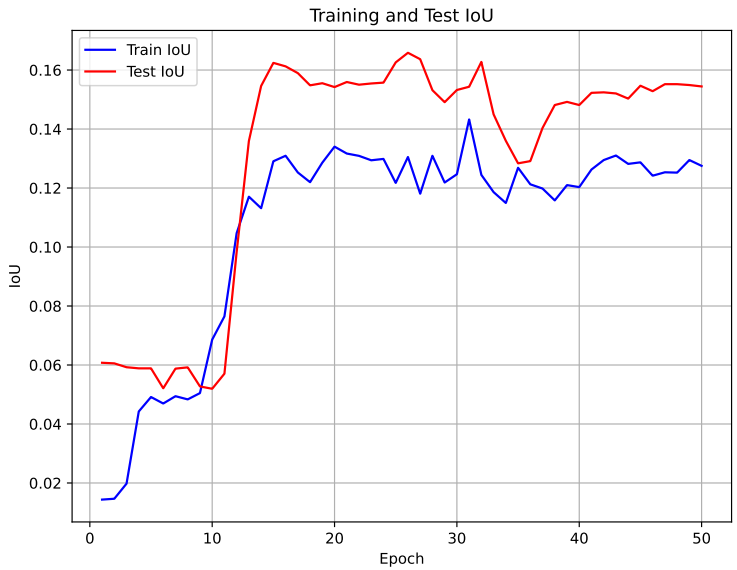
Минусы подхода:

- **Потеря информации:** снижение точности, так как модель не учитывает семантический контекст слов.
- **Слабость модели:** модель была представлена в 1997 году и уже давно не является SOTA.



Обучение LSTM. Эксперимент

Данные: SemEval-2025 Validation Set на 10 языках (500 троек: (вопрос, ответ, фрагменты_галлюцинации).
Модель: `torch.nn.LSTM` → `torch.nn.Linear`.
Setup: Train/Test = 9:1, `hidden_size` = 2048. Обучение - 50 эпох.
Лучший результат: 15.5% по IoU на тестовой выборке.
Процесс обучения представлены на графиках



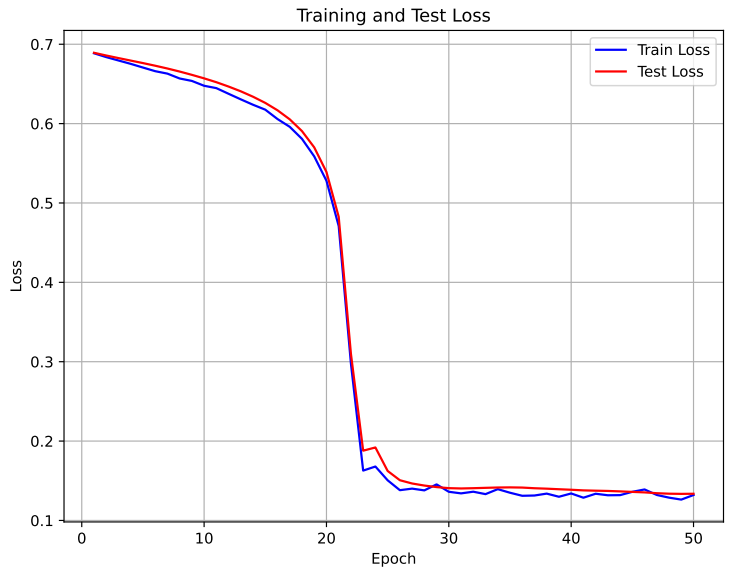


Рис.: Сравнительные графики IoU и BCE Loss на обучающей и тестовой выборках.

Предложенный метод 3. Дообучение NER модели

Модель обучается предсказывать вероятности $\hat{S}$ для каждого токена текста.

Обозначения матриц меток и предсказаний:

$S \in \{0,1\}^{N\times 3}$: бинарная матрица, содержащая истинные метки для каждого токена, где:

- $S_{i,0} = 1$, если токен x_i отмечен как **B**,
- $S_{i,1} = 1$, если токен x_i отмечен как **I**,
- $S_{i,2} = 1$, если токен x_i отмечен как **O**.

$\hat{S} \in [0,1]^{N\times 3}$: матрица предсказаний модели, где $\hat{S}_{i,t}$ обозначает вероятность того, что токен x_i относится к классу $t \in \{B, I, O\}$.

Задача обучения модели заключается в поиске параметров, доставляющих минимум функции потерь:

$$CE(S, p(S \mid \theta)) = CE(S, \hat{S}) = -\frac{1}{N}\sum_{i=1}^{N}\sum_{t\in\{B,I,O\}} S_{i,t}\log(\hat{S}_{i,t}) \longrightarrow \min_{\theta}$$

Дообучение NER модели. Эксперимент

Данные: HaluEval с переразметкой в token classification формат с помощью GPT-4 (10000 троек: (вопрос, ответ, фрагменты_галлюцинации)).
Модель: `distilbert-base-uncased`.
Setup: Train/Test = 9:1, Обучение - 3 эпохи.
Лучший результат: 21.7% по IoU на тестовой выборке.
Процесс обучения представлены на графиках

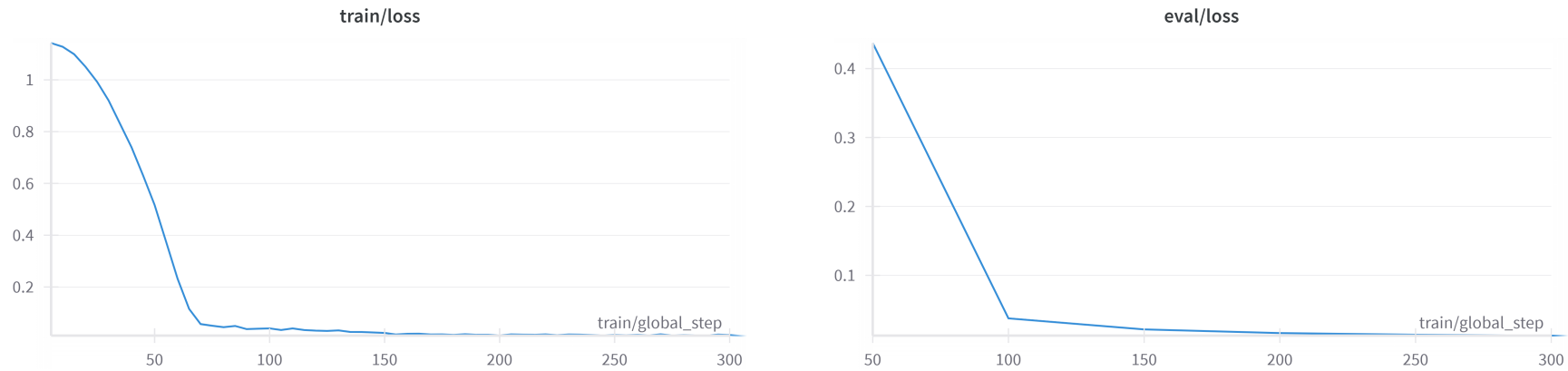


Рис.: Сравнительные графики IoU и BCE Loss на обучающей и тестовой выборках.

- Лучшее качество показала LLama-3.1-8B, однако временные затраты слишком велики.
- Дальнейшие исследования будут связаны с построением RAG систем.

Спасибо за внимание!