Разработка метода гибридного информационного поиска для RAG систем

Аушев И.А.

Центр когнитивного моделирования МФТИ

2024 – Научно-исследовательский семинар

RAG

Несмотря на впечатляющие результаты качества генерации LLM, эти модели сталкиваются с ограничениями, связанными с их склонностью к «галлюцинациям» – генерации правдоподобного, но фактически неверного текста.

Retrieval Augmented Generation (RAG) – мощный инструмент для улучшения вопросно-ответных навыков LLM.

RAG сильно влияет на качество работы LLM, предоставляя им доступ к релевантной информации из внешних источников знаний

RAG

Архитектура RAG базируется на двух фундаментальных компонентах: информационный поиск (Information Retrieval) и доверительная генерация (Trust Generation).

- Информационный поиск отвечает за выбор релевантной информации из внешнего источника знаний
- Модель для генерации (LLM), использует извлеченную информацию, полученную для конкретного запроса, и генерирует окончательный ответ.

RAG

Из описанной схемы работы RAG становится ясно, что результат генерации всецело зависит от результатов информационного поиска, поэтому важно правильного формировать контексты для запросов.

Для многих прикладных задач где используется RAG необходимо отвечать на запросы не только правильно, но и максимально быстро.

Hybrid Information Retrieval

На практике часто приходится использовать больше одной модели поиска нужной информации, тем самым улучшая качество получаемых контекстов для запроса.

Подход использующий композицию моделей (ретриверов) называется гибридным информационным поиском (Hybrid Information Retrieval)

Постановка задачи

Пусть Q – множество запросов (например, вопросы от пользователей). Запрос $q \in Q$ есть последовательность слов

$$q = \{w_1, w_2, ..., w_n\}$$

Имеется база знаний D, элементы D будем называть документами или контекстами.

$R(q, D)$ – функция поиска релевантных документов для $q \in Q$
$R(q, d) = \{d_1, d_2, .., d_k\}$ упорядоченный по релевантности набор документов.

$M_1(Q, D, R)$ – метрика качество поиска

Постановка задачи

Качество генерации оценивается метрикой $M_2(Q, D, R)$.

Задача RAG сводится к следующей оптимизационной задаче:

$$\arg\min_{R,G} M_2(Q, D, R)$$

На практике чаще всего сначала находят оптимальную функцию R в смысле метрики M_1, а затем функцию G

Постановка задачи и цель

Функция R в задаче может быть произвольной, это может быть один ретривер или композиция нескольких. На практике гибридная композиция моделей показывает результаты лучше.

Целью данной работы является улучшение и развитие методов гибридного информационного поиска.

Виды моделей

Ретриверы можно глобально разделить на два класса:

- Sparse ретриверы (TF-IDF, BM25, SPLADE)
- Dense ретриверы (NV-Embed, bge-en-icl)

Эти методы отличаются тем как они представляют запросы и документы в векторном виде.

TF-IDF

Пусть $V = \{v_1, v_2, \dots v_N\}$ упорядоченный словарь всех слов, присутствующих в базе знаний

TF-IDF[1] (Term Frequency-Inverse Document Frequency) – мера, используемая для оценки важности слова в документе относительно коллекции документов. Пусть v_i - слово, d_j - документ, D - база знаний

$$\text{TF-IDF}(v_i, d_j, D) = \text{TF}(v_i, d_j) \cdot \text{IDF}(v_i, D)$$

$$\text{TF}(v_i, d_j) = \frac{n_i}{\sum_{k=1}^{N} n_k}$$

где n_i – число вхождений слова v_i в документ d_j

$$\text{IDF}(v_i, D) = \ln \frac{|D|}{|\{d_j \in D | v_i \in d_j)\}|}$$

[1] Gerard Salton и Michael McGill. «Introduction to Modern Information Retrieval». В: 1983. URL: https://api.semanticscholar.org/CorpusID:43685115.

TF-IDF

Для получения векторных представлений для каждого слова q_i из запроса q вычисляется TF-IDF(q_i, q, D)

Полученные векторы оказываются сильно разреженными.
Преимущества подхода:

- Скорость
- Интерпретируемость

Недостатки:

- Семантическая ограниченность
- Лексическая ограниченность

BM25

BM25[2]– усовершенствованный вариант TF-IDF, который учитывает длину документа. Итоговое значение релевантности между запросов и документом у BM25:

$$score(q, d) = \sum \text{IDF}(q_i, D) \cdot \frac{\mathbf{f}(q_i, d) \cdot (k_1 + 1)}{\mathbf{f}(q_i, d) + k_1 \cdot (1 - b + b \cdot \frac{|D|}{A})}$$

где $\mathbf{f}(q_i, d)$ – число вхождений слова v_i в документ, A – средняя длина документов, d, k_1, b, - гиперпараметры

BM25 имеет те же преимущества и недостатки, что и TF-IDF, но на практике работает лучше для ряда задач.

[2] Stephen E. Robertson и др. «Okapi at TREC-3». В: *Proceedings of The Third Text REtrieval Conference, TREC 1994, Gaithersburg, Maryland, USA, November 2-4, 1994*. Под ред. Donna K. Harman. NIST Special Publication. 1994.

SPLADE

SPLADE[3](SParse Lexical AnD Expansion Model) – подход на основе архитектуры BERT и задачи Masked Language Modeling (MLM).

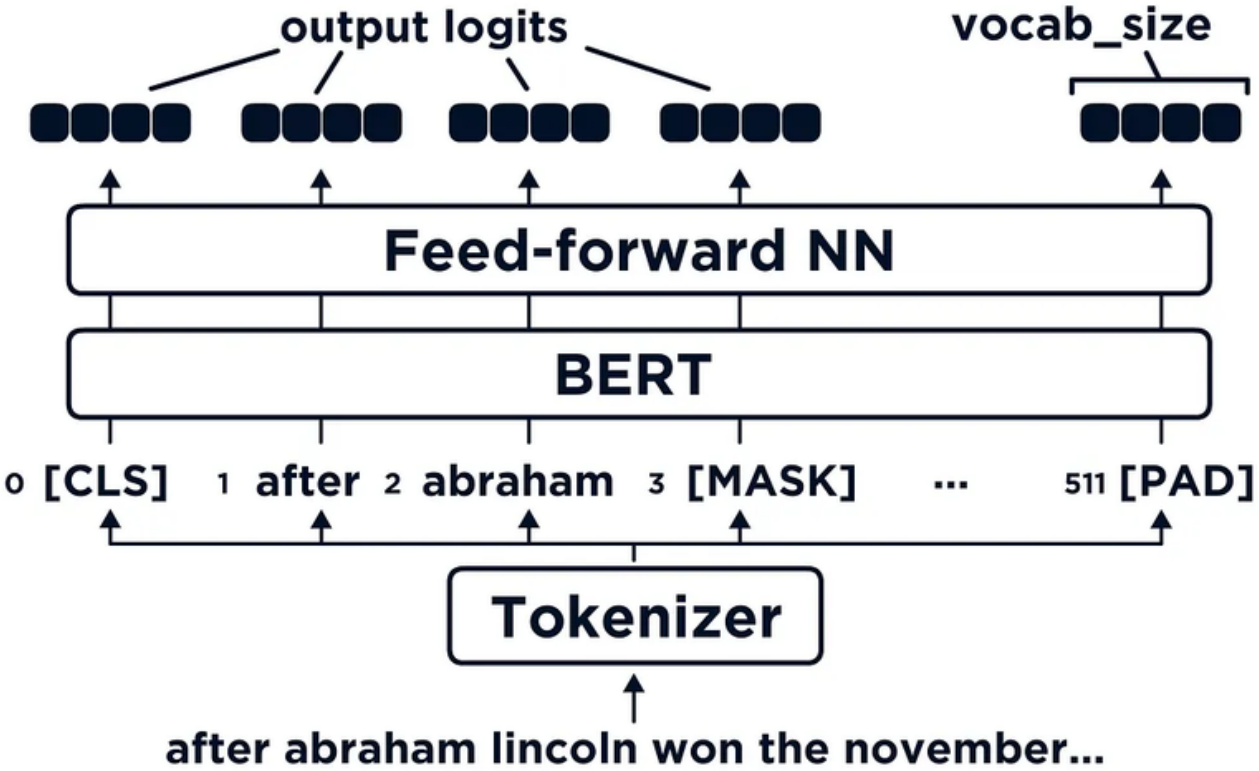

Рис 1. Архитектура BERT для задачи MLM

[3] Thibault Formal, Benjamin Piwowarski и Stéphane Clinchant. «SPLADE: Sparse Lexical and Expansion Model for First Stage Ranking». B: Association for Computing Machinery, 2021. ISBN: 9781450380379.

SPLADE

Пусть векторы $h_1, \dots h_n$ – эмбеддинги токенов после BERT и линейной проекции Далее вычисляем вектор w:

$$w = \sum_{i=1}^{n} \ln(1 + \text{ReLU}(\hat{h}_i))$$

Вектор w содержит в себе информацию о каждом слове из исходного запроса или документа

Используя полученные векторные представления, производится поиск наиболее релевантных документов.

SPLADE

Преимущества подхода:

- Не является семантически ограниченным
- Интерпретируемость

Недостатки:

- Слабая разреженность получаемых представлений
- Менее эффективная модель

NV-Embed[4]

Подход основывается на использовании decoder-only LLM архитектуры для получения векторных представлений.

Есть два популярных метода получения эмбеддингов для последовательности токенов: 1) mean pooling и 2) использование эмбеддинга последнего токена <EOS>.

В работе авторы представили слой Latent Attention Layer для получения более качественно эмбеддинга последовательности токенов

[4] Chankyu Lee и др. «NV-Embed: Improved Techniques for Training LLMs as Generalist Embedding Models». В: *Submitted to The Thirteenth International Conference on Learning Representations*. under review. 2024. URL: https://openreview.net/forum?id=lgsyLSsDRe.

Latent Attention Layer

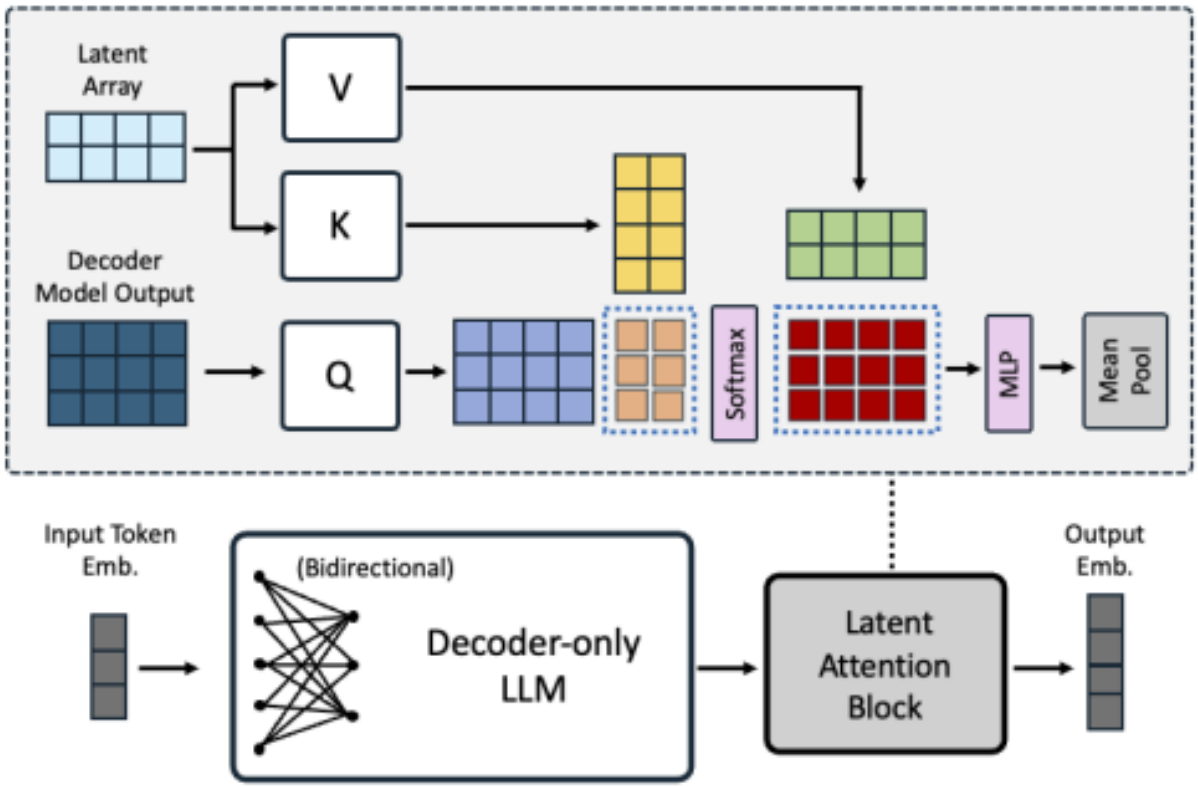

Рис 2. Latent Attention Layer

NV-Embed

Обучение модели происходит с помощью стандартного Constrastive Learning.

Авторы статьи предложили перейти от casual attention к стандартному bi-directional attention во время обучения decoder-only LLM.
Было предложено также обучать модель на смежных задачах(классификация, кластеризация) для улучшения обобщающей способности модели.

В результате обученная модель получила лучший результат на бенчмарке MTEB[5]

[5] Niklas Muennighoff и др. «MTEB: Massive Text Embedding Benchmark». В: *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*. Dubrovnik, Croatia, май 2023.

NV-Embed

Преимуществом данного подхода является качество получаемых эмбеддингов, недостатком – вычислительная сложность подхода.

BGE-en-icl[6]

Decoder-only LLM демонстрируют отличные возможности для in-context learning (ICL). В модели bge-en-icl предложили использовать эту идею для улучшения процесса создания эмбеддингов входных запросов.

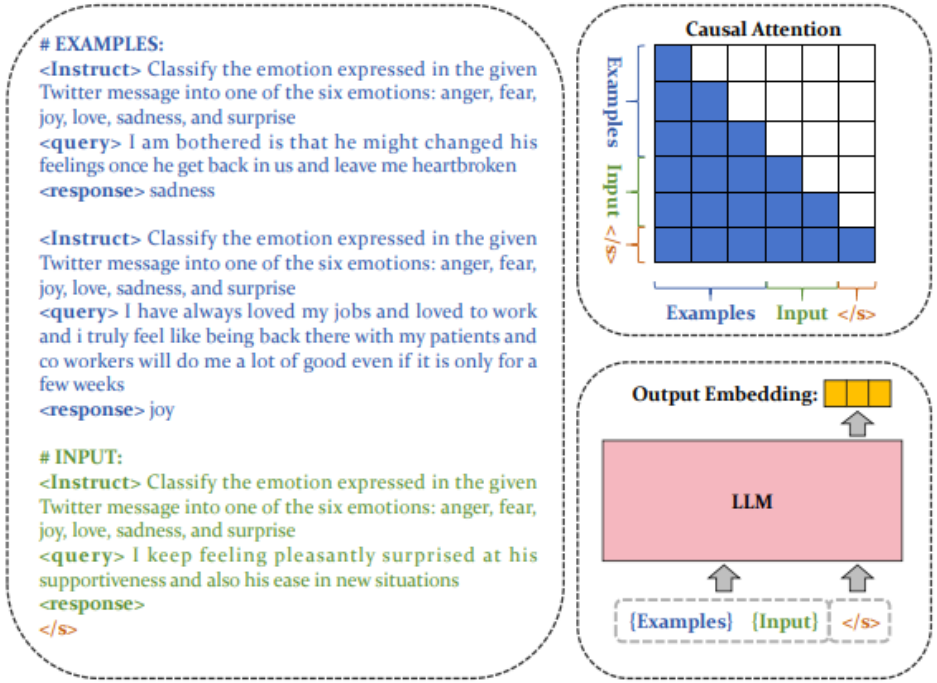

Рис 3. Архитектура bge-en-icl

[6] Chaofan Li и др. «Making Text Embedders Few-Shot Learners». В: *Submitted to The Thirteenth International Conference on Learning Representations*. under review. 2024. URL: https://openreview.net/forum?id=wfLuiDjQ0u.

bge-en-icl

Промпты для модели имеют следующий вид:

$$e_i = \text{<Instruct>}\{\text{task definition}\}\text{<query>}q_i\text{<response>}d_i$$

$$\text{prompt} = \{e_1\}\ldots\{e_n\}\text{<Instruct>}\{\text{task definition}\}\text{<query>}q < \text{response} >$$

Используя такой промпт модель обучается стандартно с помощью Constrastive Learning. Подход показал отличные результаты на бенчмарке MTEB.

bge-en-icl

Подход с помощью in-context learning помогает модели решать неизвестную для себя задачу, что является плюсом.

Однако, такой подход требует внимательного подбора промптов для конкретной задачи.

Hybrid Retrieval

Подходы к гибридному поиску разделяются на 3 стратегии: параллельные, последовательные, и их комбинация в рамках одной архитектуры.

Последовательной стратегии заключается в пошаговом применении ретриверов, основываясь на результат предыдущих.

Параллельная стратегия подразумевает прямое аггрегирование результатов разных ретриверов.

Последовательная стратегия

Последовательная стратегия часто реализуется с использованием дополнительных cross-encoder моделей, которые принимают на вход запров q и документ d, и возвращают оценку релевантности между ними.
Такой процесс называют Reranking.

Параллельная стратегия

Сейчас популярен метод Reciprocal Rank Fusion (RRF)[7], который реализует параллельную стратегию поиска. Пусть w_1 и w_2 - результаты ретрива двух разных моделей, d - документ, K - константа.

$$\text{score}(d) = \frac{1}{\text{rank d in } w_1 + K} + \frac{1}{\text{rank d in } w_2 + K}$$

[7] Gordon V. Cormack, Charles L. A. Clarke и Stefan Büttcher. «Reciprocal rank fusion outperforms condorcet and individual rank learning methods». В: *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval* (2009). URL: https://api.semanticscholar.org/CorpusID:12408211.

Результаты

Часть из рассмотренных методов были испробованы на практике в рамках Regulatory Natural Language Processing Workshop 2025. В задаче требовалось создать RAG систему для датасета финансово-нормативных документов.
В предложенном решении были реализованы две идеи:

1. Префильтрация исходных документов с помощью BM25, для получения более узкого набора.
2. Параллельный ретрив с помощью нескольких моделей(BM25, NV-Embed-v2 и bge-en-icl). Далее Reciprocal Rank Fusion между моделями всеми моделями.

Описанный подход занял 2 место и статья, в которой описано решение задачи, принята на конференцию COLING 2025.