Data

1) **Чанкинг по сымслу (семантический, структура документа) +?**
2) **Overlap между чанками >20% (минимум) +**
3) Препроцессинг, лематизация (spacy, nltk) ССЫЛКИ УБИРАТЬ +
4) Лематизация ДЛЯ BM25 +
5) IVF / HNSW / PQ (Product Quantization) + использовать батч (fiass gpu) +???
6) Кэшировать запросы?

Prompting

1) **Ролевая инструкция (“Ты эксперт…”) + добавить конкретики**
2) **Чёткая структура вывода**
3) **Контекст всегда подаётся раньше вопроса**
4) **Ансамбль промптов + переписать промпт ллм НАЙТИ ПРОМПТЫ**
5)
6) Просить оценить свой ответ перед выводом
7) Few-shot

Retrieval

1) BM25 + FIASS +
2) Ансамбль энкодеров +
3) **Двуступенчатый ретривал + использование cross-encoder (next sentence prediction) +**
4) **Дополнительная мета, как в задании на кинопоиск**
5) Maximal Marginal Relevance (есть в langchain) помогает разнообразную инфу добавлять в контекст +
6) Hierarchical retrieval + родительские куски текста
7) **Adaptive K - выбираем порог по уверенности а не по количеству +**

Reasoniong

1) **Multi-hop/ Self-ask - разбивает запрос на части + Self-RAG + ReAct**
2) Chain-of-Thought prompting

Generation

1) Разбить важные куски по краям
2) **Суммаризация контекста**
3) **Просить указывать источник каждого факта**
4) Self-consistency sampling - посемплить несколько ответов с разным сидом и температурой, разным порядком, выбрать самый часто встречающийся ллмкой
5) **Отдельная LLM проверяет соответствие ответ-контекст**

Отдельно попытаться пообучать кросс экнодер

На чем?

1) Одна фраза из чанка, остальное в чанке позитив, любой другой чанк из другого документа негатив +
2) Заголовок - запрос, содержащиеся в абзаце/параграфе этого заголовка - ответ, если есть структура
3) Через Spacy получить ключевое слово в абзаце, абзац - ответ
4) Прпросить ллм задать вопрос