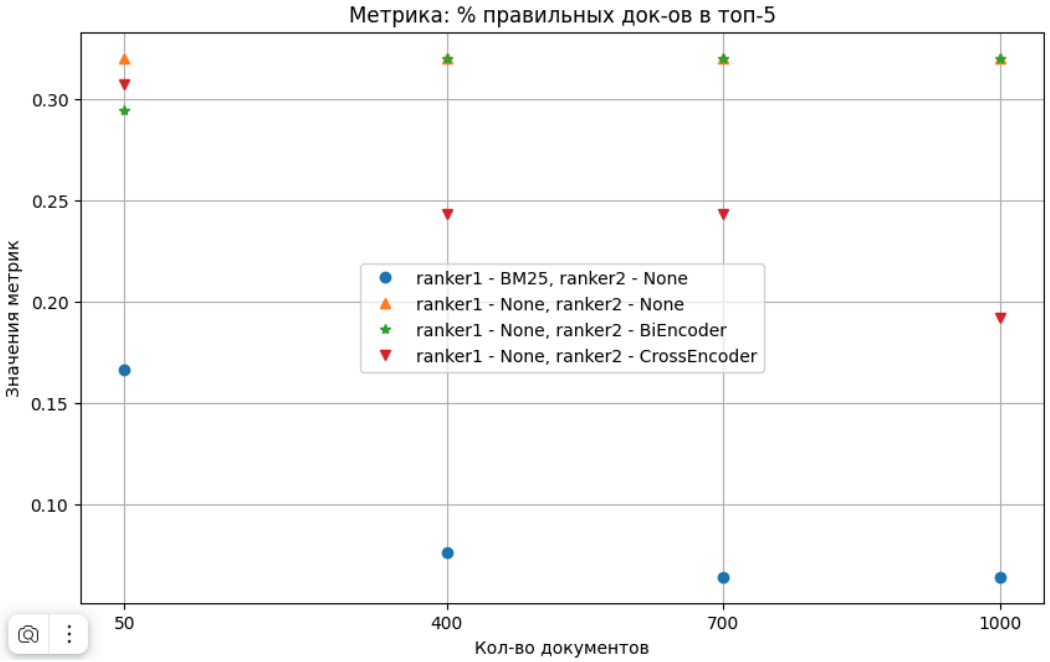
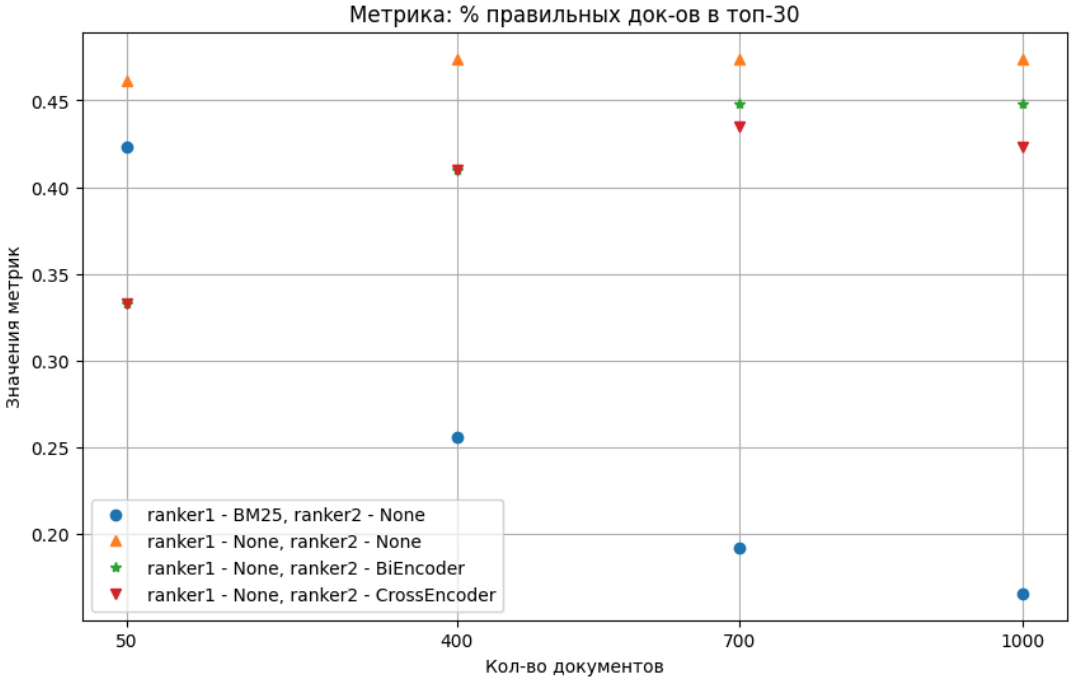
Схема поисковика

1) В базе данных хранится 744954 документа.
   Под документом понимается каждая часть страницы википедии, выделения в содержании.
   Например: https://ru.wikipedia.org/wiki/Россия#Животный_мир и
   https://ru.wikipedia.org/wiki/Россия#Растительность - это два отдельных документа
2) Поисковик работает на одном узле, и все компоненты запущены в рамках монолитного приложения.

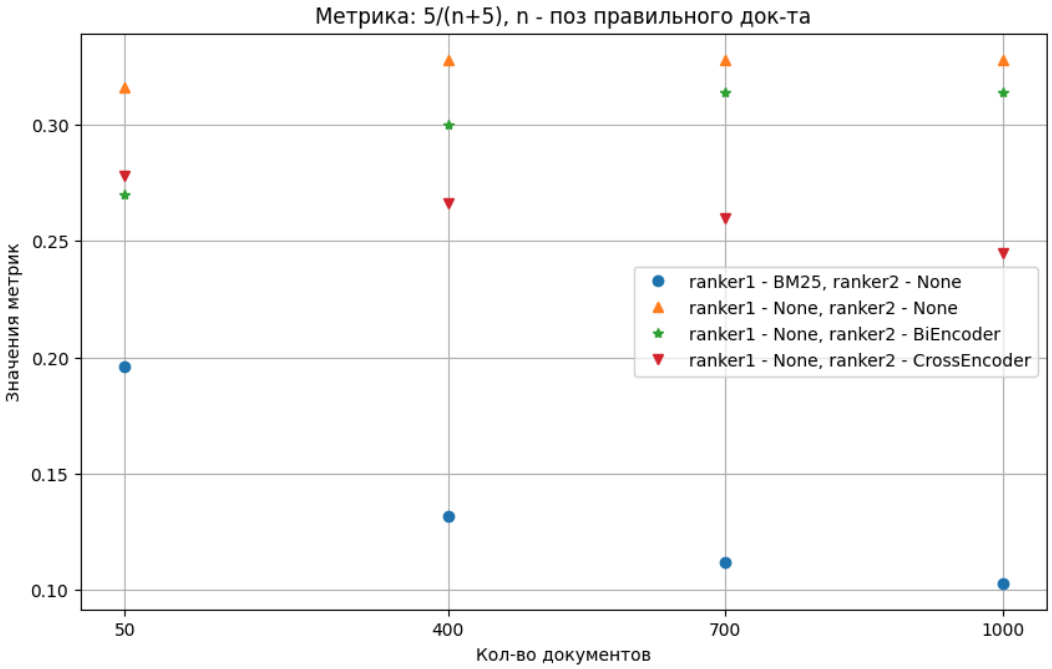
Метрики качества



Лучшей оказалась конфигурация, когда после получения индексов документов из векторной БД, они никак больше не переранжируются.
Вне зависимости от количества искомых документов в БД, в среднем более чем в 30% случаев документ (единственный), соответствующий запросу, оказывается в топ-5 выдачи.



При расширении проверки выдачи, конфигурация, описанная выше, только усилила своё превосходство.
Вне зависимости от количества искомых документов в БД, в среднем более чем в 45% случаев документ (единственный), соответствующий запросу, оказывается в топ-30 выдачи.



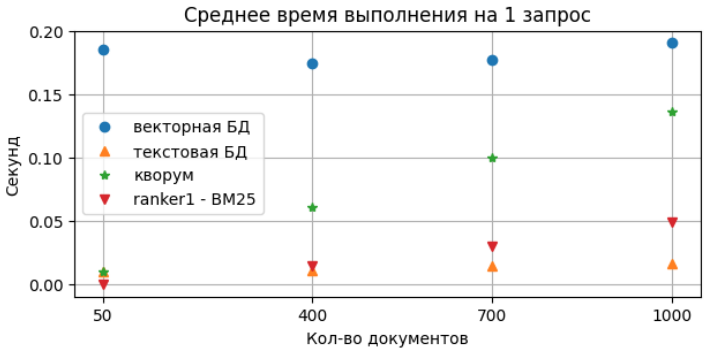
Данная метрика также подтверждает лучшее качество конфигурации без дальнейшего ранжирования документов, полученных векторным поиском в БД faiss с индексом FlatL2.

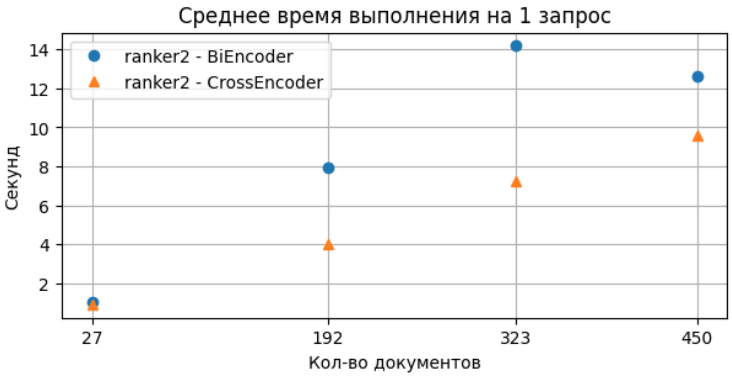
Поэтому, **конфигурация без ранжирования является базовой при работе поисковика.**

Замечания.

- BM25 и кворум использовался со стеммингом текста.
- Порог кворума был >0, где сам кворум вычислялся как len(пересечений в словах запроса и документа) / len(слов в запросе)
- Для 2-ой стадии ранжирования использовались топ-10% док-ов.

Latency компонентов

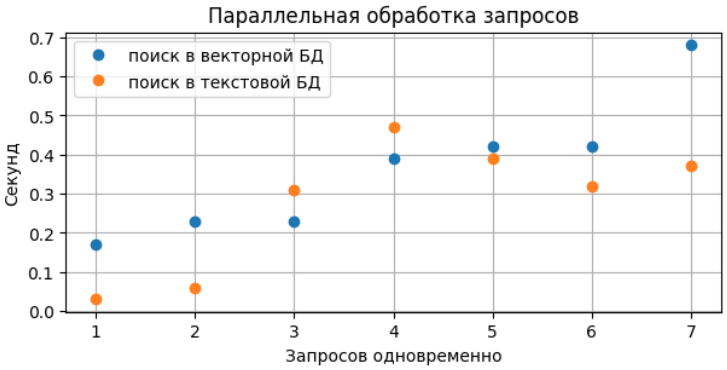


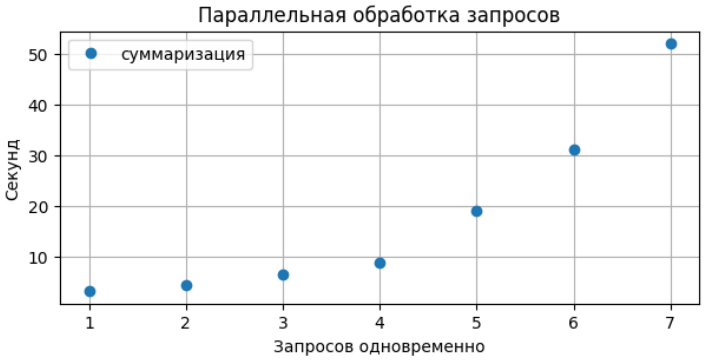


Среднее время работы модели-суммаризатора для ответа на один вопрос - 2.5 секунды.

В итоге, время обработки одного запроса базовой конфигурацией (без ранжирований) поисковика составляет ~ **0.25 сек поиск документов + 2.5 сек ранжирование = 2.75 сек.**
Причём, при изменении поиска документов от 50 до 1000 время работы сильно не меняется.
Поэтому **к базовой конфигурации добавляем ещё и поиск документов в БД - 1000**.

Пропускная способность





Можно заметить, что после перехода порога в 2 запроса одновременно базы данных начинают вести себя нестабильно, то есть перестают работать за примерно одинаковое время.
А если вынести ранжирование на несколько отдельных узлов, то обработка 2-ух запросов будет занимать столько же времени, что и один.

Не учитывая суммаризацию, можно сказать, что поисковик может обрабатывать 2 запроса одновременно сохраняя описанный выше latency. Тогда **пропускная способность равна 1/0.25 * 2 = 8 запросов в секунду.**
Для дальнейшего увеличения пропускной способности поисковика, с сохранением одинакового latency для каждого запроса, необходимо ставить поисковик не на одном узле, а на нескольких.

Выбор значения treshhold для кворума

Видим, что при увеличении treshhold больше 0.15 начинает падать качество, поэтому **выбираем treshhold = 0.15**.

Ещё раз напомню, кворум вычислялся как:
len(пересечений в словах запроса и документа) / len(слов в запросе)