Домашнее задание 4

В данном домашнем задании Вам предстоит реализовать поиск с помощью инвертированного индекса.

1. Индексация

Реализуйте построение инвертированного индекса в памяти для коллекции из домашней работы номер 3. В каждом постинглисте также сохраните значение term-frequency.

Сохраните полученный индекс на диске в бинарном формате. Размер сохраненного индекса в байтах должен быть порядка 8*(сумму длин всех постинг листов).

Отдельно сохраните на диск дополнительные данные о коллекции, которые пригодятся для поиска, например, названия статей или среднюю длину документа. Размер дополнительных данных, должен быть пропорционален количеству документов коллекции.

2. Поиск

Для простоты реализации поиска, не требуется делать чтение постинглистов с диска по запросу - достаточно считать их с диска в память целиком. Также загрузите с диска дополнительные данные о коллекции.

Реализуйте поиск документов с ранжированием BM25 на основе инвертированного индекса в парадигме document-at-time. Функция поиска должна принимать число - ограничение на количество документов, возвращаемое поиском. Используемое количество дополнительной памяти должно быть пропорционально этому ограничению и никак не должно зависить от размера постинглистов или размера коллекции. Результаты поиска должны быть аналогичные тем, что были в домашней работе номер 3.

Сравните качество и скорость работы нового алгоритма поиска с предыдущим.

Реализуйте static pruning до 50 элементов для каждого постинглиста. Сравните качество и скорость работы нового индекса с предыдущим.

~~Дополнительно~~

3. Сжатие индекса (+1 балл)

Реализуйте кодирование чисел алгоритмом VarInt.

Сравните эффективность разных вариантов кодирования постинглистов:

- Базовый вариант (4 байта на число)
- Какой-нибудь алгоритм сжатия общего назначения (lz4/zstd/brotli/gzip)
- VarInt
- Delta-кодирование + Какой-нибудь алгоритм сжатия общего назначения
- Delta-кодирование + VarInt