ИСПОЛЬЗОВАНИЕ B+ TREE ДЛЯ ИНВЕРТИРОВАННЫХ ФАЙЛОВ

ГУБИН М.В.
КОНСОРЦИУМ КОДЕКС

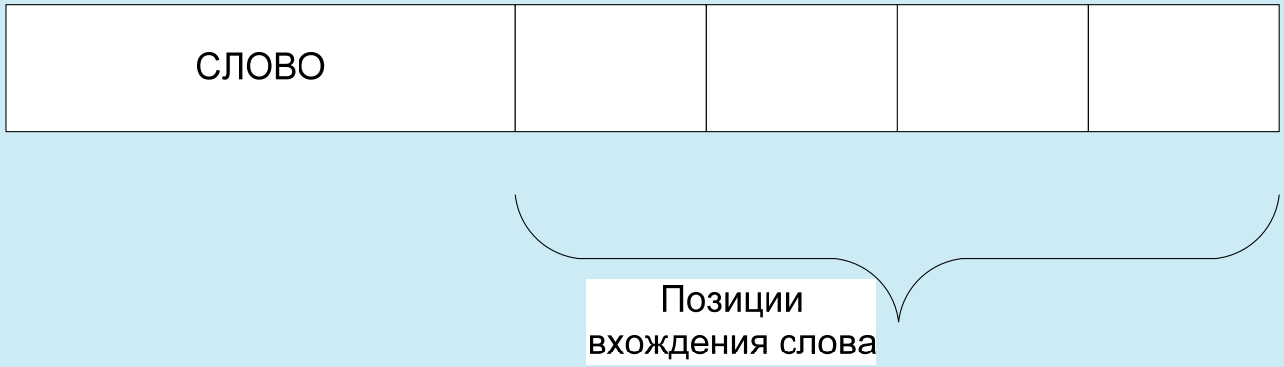
B+Tree как основа инв. файла
плюсы

- Существуют готовые реализации
- Управление буферами, совместное использование хорошо исследовано
- Можно быстро обратиться к конкретному вхождению слова при поиске и модификации

B+Tree как основа инв. файла
МИНУСЫ

- Не компактные
- Медленно обрабатывают длинные листы вхождения

Организация хранения



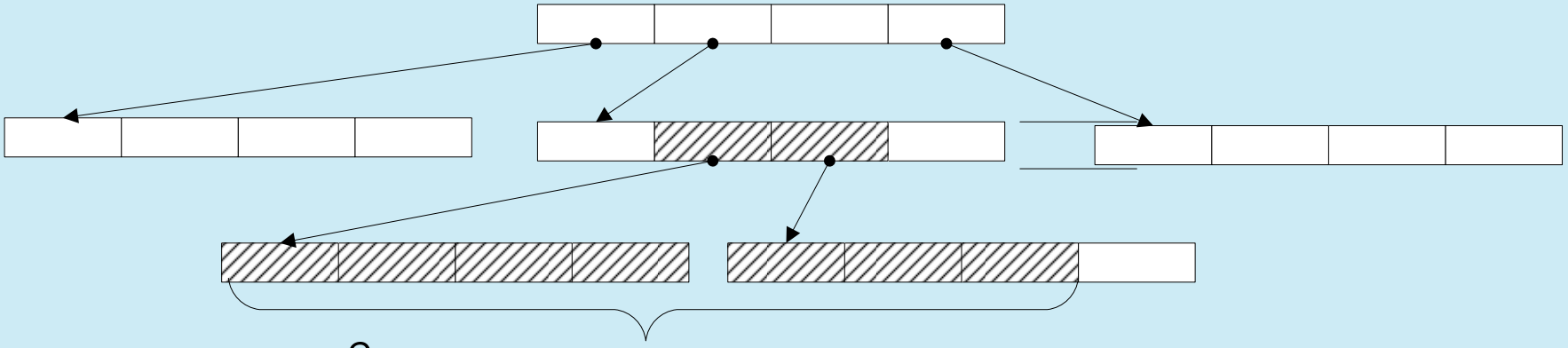
- Каждый слот – слово + кусочек листа вхождения
- Длинные листы – несколько слотов
- Переменный размер слотов, деление страниц по размеру занимаемых данных

Сжатие индекса (двухуровневое)

- Внутри слота – слово BOCU-I, фрагмент листа – дельта+байтовое кодирование
- Внутри листа – сжатие всех данных листа с помощью deflate

Обработка длинных листов вхождения

- Определение длины пост-листа на основании % индекса
- Для длинных листов переход от учета всех вхождений к «локальным контекстам»



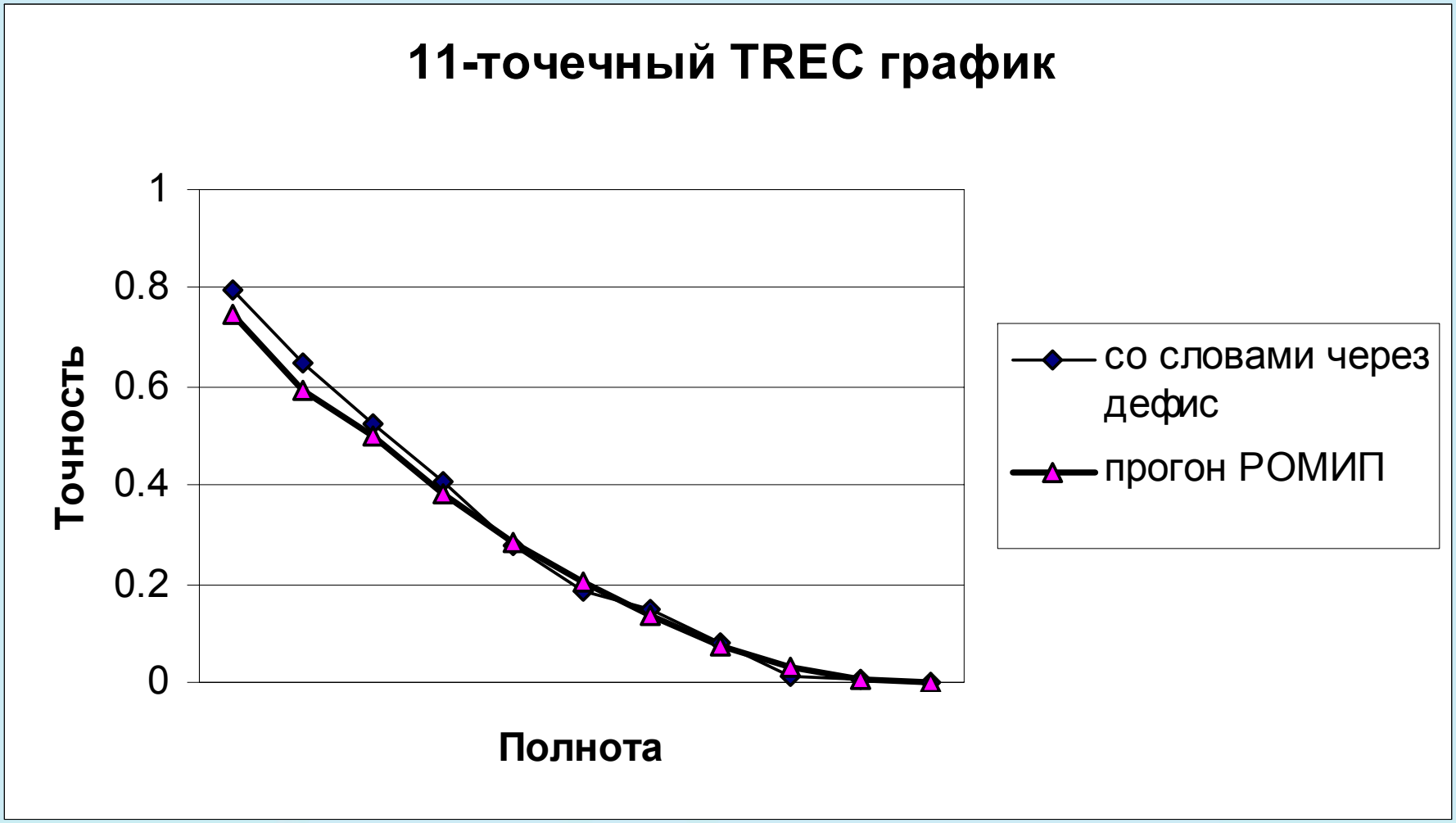
Формула для определения веса документа

$$S = \sum_{t \in Q} \frac{(K+1)\,f}{K(1-b+bL)+f} * \ln\left(\frac{(r+0.5)(N-n-R+r+0.5)}{(n-r+0.5)(R-r+0.5)}\right)$$

$$L = \frac{l}{L_{avg}}$$

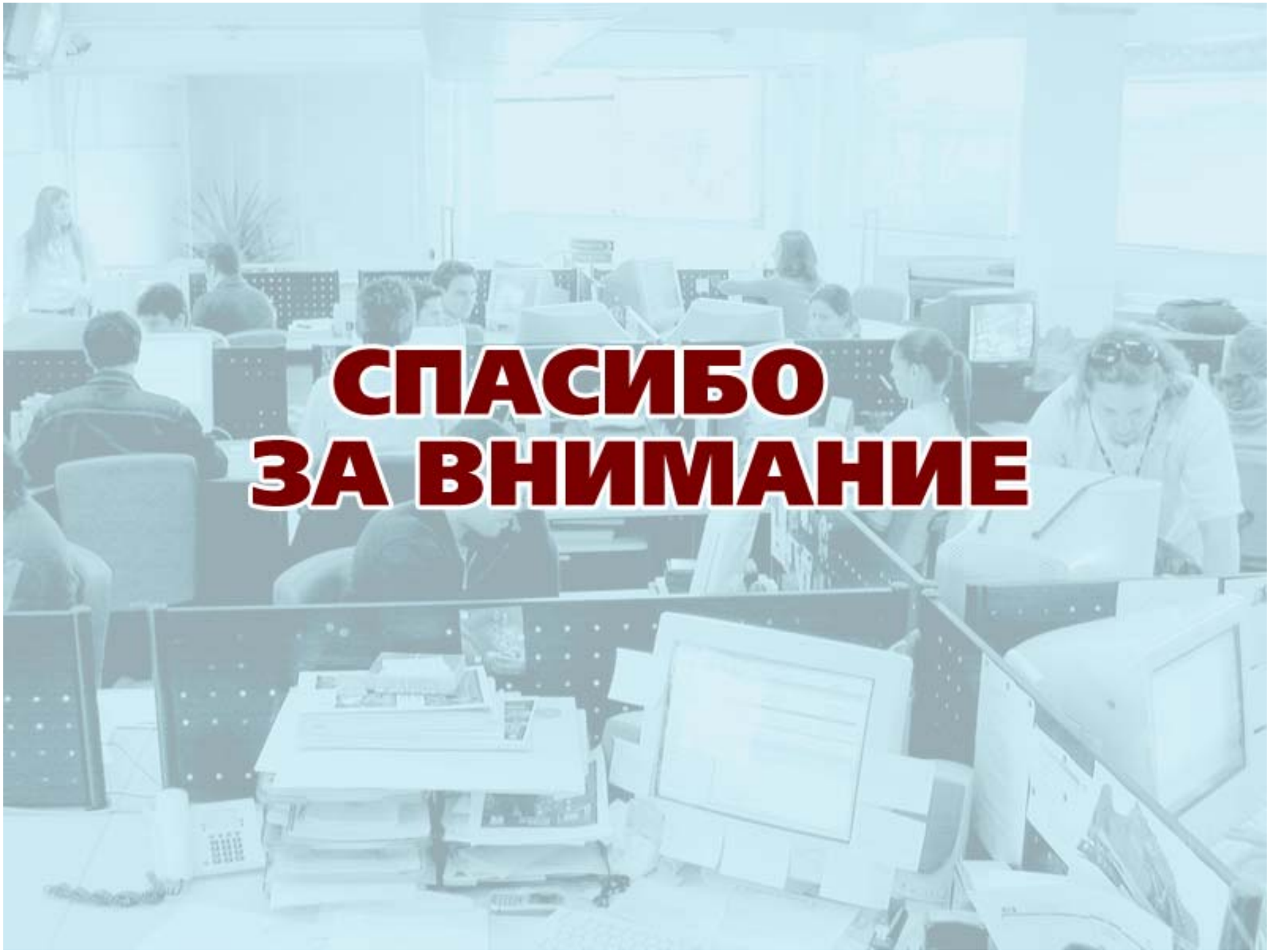
- Фразы – как отдельные термины
- Часто встречающиеся термины – только позиции рядом с другими словами
- L_{avg} – средняя длина для отобранных документов

«Работа над ошибками»



Выводы

- Компактный индекс (25-28%)
- Высокое быстродействие (0.2-0.3 сек/запрос) для коллекций РОМИП
- Сравнимое качество поиска


СПАСИБО
ЗА ВНИМАНИЕ