

Описание алгоритма

1. Файл загружается в память с использованием mmap
2. Из файла последовательно извлекаются слова и предложения, в это время формируются три главные сущности программы:
 - a. Обратный индекс. Отображение слова в множество пар вида <номер предложения, TF> (Важно отметить, что по построению это множество отсортировано по возрастанию номеров предложений)
 - b. Таблица IDF для каждого слова в тексте (В действительности просто подсчитывается количество вхождений слова в текст. Далее используем термин "IDF" в его истинном значении)
 - c. Таблица отступов. Чтобы быстро передвигаться по тексту, составляется массив, где индекс элемента - номер предложения, а значение - отступ в байтах * sizeof(wchar_t)
3. Программа ожидает поисковый запрос.

Способ выбора сниппета

1. Входной запрос разбивается на слова и сортируется в порядке убывания IDF. Это с большой вероятностью дает нам структуру (1) вида

```
****
*****
*****
*****
```

где число звездочек - количество предложений, в которых встречается i-е слово запроса.

2. Выбираем 2 предложения из текста с максимальными весами.
Вес k-го предложения рассчитывается из формулы (i - номер термина в запросе):

$$Weight(k) = \sum_i TF(k, i) \times IDF(k, i)$$

В действительности осуществляется обход не всех предложений. При достаточном количестве слов в запросе осуществляется обход первых двух рядов структуры (1)

3. Для каждого предложения из ряда 0 смотрим, встречается ли оно в последующих рядах. Если да, то добавляем член в сумму, иначе - идем ниже.
Повторяем это максимум до 3 совпадений. Это число выбрано из соображений вероятности и экономии ресурсов: вряд ли более четырех слов запроса встретятся в одном предложении, а учитывая, что с ростом уровня IDF уменьшается, можно отбросить нижние уровни, выиграв в скорости выдачи сниппета.

Аналогичная операция проводится для уровня 1 - получаем новый набор предложений.

Из полученного набора выбираем пару с наибольшими весами и возвращаем ее.