Fatih Çakır

CS 533 INFORMATION RETRIEVAL
HOMEWORK #6

Question 2)

Lexicon (corpus)

bilmek, bilgiç, bilgin, bilgisayar, bilim, bilinç, bilişim, bilişsel

bilgin

Prefix	**Successor Variety**	**Letters**
b	1	İ
bi	1	l
bil	3	m, g, i
bilg	1	i
bilgi	3	ç, n ,s
bilgin		

When there is more than one peak, we choose the latter one since the initial peak will make it difficult to find the stem. Hence we chose '**bilgi**' as the stem of the word 'bilgin'.

Question 4)

Let F be the length of the signature and n be the number of objects (i.e. documents). Then in our case $F = 192$ and $n = 10{,}240$.

a) For Sequential Signatures (SS) the signature file size is

$$Signature\ File\ Size = F \times n = 192 \times 10{,}240 = 1966080\ bits$$

$$1966080 \times 1\frac{byte}{8\ bits} = 245760\ bytes$$

b) For Bit-sliced Signatures (BS) ,which basically is the traverse of the SS method, the signature file size is the same

$$Signature\ File\ Size = F \times n = 192 \times 10{,}240 = 1966080\ bits$$

$$1966080 \times 1\frac{byte}{8\ bits} = 245760\ bytes$$

Question 6)

Let op denote on-bit density where

$$op = \frac{\#\ of 1^{'}\ \sin a\ signature}{signature\ length}$$

and let fd denote the false-drop resolution where

$$fd = \frac{\#\ of\ matches}{\#\ of\ signatures}$$

We know that after processing i amount of bits we have the equality

$$fd_i = op^i$$

a) From the question we know that $op = \left(\frac{1}{2}\right)^4$, $F = 16000\ bits$ and the number of signatures is 2^{30}.

   a. After processing 1 bit of the query

$$[\left(\frac{1}{2}\right)^4] \times 2^{30} = 2^{26}$$

   b. After processing 2 bits of the query

$$\left[\left(\frac{1}{2}\right)^4\right]^2 \times 2^{30} = 2^{22}$$

   c. After processing 3 bits of the query

$$\left[\left(\frac{1}{2}\right)^4\right]^3 \times 2^{30} = 2^{18}$$

   d. After processing 4 bits of the query

$$\left[\left(\frac{1}{2}\right)^4\right]^4 \times 2^{30} = 2^{12}$$

   number of documents has to be accessed, respectively.

b) After a certain point the cost of false drop resolution processing becomes lower than bit processing if there are very long bit slices in our structure. Hence in order to improve the performance of bit sliced signature files we may only partially evaluate a query signature but at the same time maintain the contribution of each query term to the query evaluation process. The subset of the query signature is selected by considering submission probabilities of the queries with different numbers of terms.

Question 8)

a) EPP stands Extended Prefix Partitioning, *'where the key length is chosen to be the shortest prefix which contains a predefined number of zeros described by z'*. [1]

For $z = 2$ we can partition the signatures as

Partition No.	Signatures
P_1	**00** 111100 **00** 001111
P_2	**010** 01011
P_3	**0110** 1100
P_4	**1010** 0011
P_5	**10110** 100

b) FKP stands for Floating Key Partitioning method where it *'examines each of the consecutive nonoverlapping k-substrings of a signature and selects the leftmost substring that has the least amount of 1s'*[1].

For $k = 2$ we can partition the signatures as

Partition No.	Signatures
P_1	**00** 111100 **00** 001111
P_2	01 **00** 1011
P_3	1010 **00** 11
P_4	011011 **00** 101101 **00**

c) For signature query Q1: 0000 1111 since the prefix of the query is '0000' will have to access all partitions for both EPP and FKP.

For signature query Q2: 1111 0000 since the prefix of the query is 1111 there are no suitable partitions to be accessed for both EPP and FKP.

For signature query Q3: 1001 1010 because of the prefix we can access only P_4 and P_5 of EPP, but we do not access any partitions of FKP.

For signature query Q4: 1110 0001 because of the prefix there are not suitable partitions to be accessed for EPP, but we do access all partitions of FKP.

[1] Signature files: an integrated access method for formatted and unformatted databases, by D. Aktug, F. Can

Question 10)

In the paper the authors adopt four different stemmer methods:

1) ***No Stemming****:* In this method no stemming is used, thus all the available words are indexed. This method was considered as baseline method.
2) ***Fixed-Prefix Stemming****:* In this method only the first n characters of the words are employed as stems. n was selected in the range of $[3,7]$. For words that are less than n characters, the word itself is considered as a stem.
3) ***Successor Variety (SV):*** This method determines the roots of a word according to the number of distinct succeeding characters for each prefix of the word to be stemmed. For a certain prefix of a word the stem is chosen where the frequency of the succeeding characters is the highest, i.e. it is based on the intuition that the stem of a word would be the prefix at which the maximum successor variety is observed.
4) ***Lemmatizer-based Stemming****:* This method is a morphological analyzer that examines word forms and returns their base or dictionary forms. It involves determining the part of speech of a term, and applying several normalization schemes for each part. The part of speech is detected before attempting to find the root since, the stemming rules change depending on a word's part of speech[2].

The authors adapted the successor variety approach to the Turkish language by considering the letter (character) transformations that may occur in Turkish, e.g. 'agaç' can become 'agac -i' or 'burun' can become 'burnu'. Such transformation happens for certain characters in the Turkish alphabet and the authors algorithm can manage such transformations by computing the probability of existence of a specific transformation. The probabilities are calculated by using the distribution of corpus terms that are related to transformations. If the probability is greater than a specific threshold, prefix under consideration contributes to the SV count of the corresponding non-transformed prefix[3].

[2] http://en.wikipedia.org/wiki/Stemming#Lemmatisation_Algorithms

[3] "Information retrieval on Turkish texts." by Can, Kocberber, Balcik, Kaynak, Ocalan., Vursavas, *Journal of the American Society for Information Science and Technology*. 59(3): 407-421