COMP6714 Assignment 1

Yao Yuan z5092195

Q1.

(1).

According to the question, word appears at least once in each of all N documents.

Lets Assume that posting IDs are always start from 1 to N.

For uncompressed encoding. each posting ID requires 4 bytes.

Therefore, the total encoding is $\boxed{4N \text{ bytes}}$

By applying Elias-γ encoding. and gap,

we know that the first posting ID is 1, and all the following gaps are 1 as well.

Since Elias-γ encoding for 1 is a bit '0'.

Therefore, the total encoding is $\boxed{\lceil \frac{N}{8} \rceil \text{ bytes}}$

By applying Elias-δ encoding. and gap,

Similar to Elias-γ, and Elias-δ encoding for 1 is a bit '0' as well.

Therefore, the total encoding is $\boxed{\lceil \frac{N}{8} \rceil \text{ bytes}}$

Since, $\text{compression ratio} = \frac{\text{uncompressed size}}{\text{compressed size}}$

Elias-γ compression ratio = Elias-δ compression ratio = $\frac{4N}{\lceil \frac{N}{8} \rceil}$

(2). According to the question, we are using gaps and term frequency in our postings; and term frequency for each doc is known to be 10.

For uncompressed encoding. we need 4 bytes for first posting ID and 4 bytes for frequency; and all the following gaps use 4 bytes as well.
Therefore, the total encoding is $\lceil \frac{8}{5}N \rceil$ bytes for $\frac{N}{5}$ docs.

By applying Elias-γ encoding.
Assume the id of the first doc is R, $R \in [1, N]$
Then, bits for first posting ID is $2\lfloor \log_2 k \rfloor + 1$ bits
bits for term frequency 10 is 7 bits
For the following postings, we are using gap 5,
Then bits for gap is 5 bits
Therefore, in total $2\lfloor \log_2 k \rfloor + 1 + 7 + (5+7) \cdot (\frac{N}{5} - 1)$

$$= 2\lfloor \log_2 k \rfloor + \frac{12}{5}N - 4 \text{ bits}$$

$$= \left\lceil \frac{2\lfloor \log_2 k \rfloor + \frac{12}{5}N - 4}{8} \right\rceil \text{ bytes}$$

By applying Elias-δ encoding.
Assume the id of the first doc is R, $R \in [1, N]$
Then, bits for first posting ID is $2\log_2\log_2 k + \log_2 k$ bits
bits for term frequency 10 is 8 bits
For the following postings, we are using gap 5,
Then bits for gap is 5 bits
Therefore, in total $2\log_2\log_2 k + \log_2 k + 8 + (5+8) \cdot (\frac{N}{5} - 1)$

$$= \left\lceil \frac{2\log_2\log_2 k + \log_2 k + \frac{13}{5}N - 5}{8} \right\rceil \text{ bytes}$$

Since, compression ratio $= \dfrac{\text{uncompressed size}}{\text{compressed size}}$

$$\text{Elias-}\gamma \text{ compression ratio} = \frac{\left\lceil \frac{8}{5}N \right\rceil \text{ bytes}}{\left\lceil \frac{2\lfloor \log_2 k \rfloor + \frac{12}{5}M - 4}{8} \right\rceil \text{ bytes}}$$

$$\text{Elias-}\delta \text{ compression ratio} = \frac{\left\lceil \frac{8}{5}N \right\rceil \text{ bytes}}{\left\lceil \frac{2\log_2\log_2 k + \log_2 k + \frac{13}{5}N - 5}{8} \right\rceil \text{ bytes}}$$

Q2.

(1). Assume after all logarithmic merges have finished, there will be n number of sub-indexes formed in disk.

Therefore, we can have $2^n = t$.

Because for each merge operation, we merge 2 into 1.

And there are t sub-indexes in total.

Since $2^n = t$,

$n = \lceil \log_2 t \rceil$ for n to be an integer.

For special case $t = 1$, we say that $n \leq \lceil \log_2 t \rceil + 1$

(2). From the proof of (1), we know that for t sub-indexes, there will be at most $\log_2 t$ sub-indexes in disk.

Therefore, for each sub-index in t, it will be called at most $\log_2 t$ times for merge operation.

Since for each sub-index, it has M pages. The read/write cost for it would be M. for each merge operation.

There are t sub-indexes in total.

Therefore the I/O cost for the whole merge operation will be bounded by $2 \cdot t \cdot M \cdot \log_2 t$ for read and write.

∴ the total I/O cost of the logarithmic merge is $O(t \cdot M \cdot \log_2 t)$.

Q3.

(1). The formula for recall is $\frac{TP}{TP+FN}$.

By using stemming, the original TP will not reduce because the stemming results must include the original correct words.

The original FN may reduce because we may include actual positives

For example.

When the query term is "automate". and when using Boolean query retrival, docs having "automates" will not be considered; But it should consider.

By applying stemming, both "automate" and "automates" are "automa". Therefore, "automates" docs are considered.

Therefore FN decreases.

Therefore, we can conclude that stemming will not hurt recall.

(2). The formula for F_1 is

$$F_1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} = \frac{TP}{TP + \frac{1}{2}(FP + FN)}$$

We know that when FN decrease by x, TP would increase by x.
when FP increase by y, TN would decrease by y.

From (1), we know that stemming will always decrease FN by an amount. Assume it is x, and $x \geq 0$.
And we know that stemming will always increase FP by an amount. Assume it is y, and $y \geq 0$.

$$\therefore \text{new } F_1 = \frac{TP + x}{TP + x + \frac{1}{2}(FP + y + FN - x)} = \frac{TP + x}{TP + \frac{1}{2}(FP + FN) + \frac{1}{2}(x + y)}$$

Let $TP = a$, $TP + \frac{1}{2}(FP + FN) = b$, $b \geq a$

$$\therefore F_1 = \frac{a}{b}, \text{ new } F_1 = \frac{a + x}{b + \frac{1}{2}(x + y)}$$

If $F_1 = \text{new } F_1$,

Then. $\frac{a}{b} = \frac{a + x}{b + \frac{1}{2}(x + y)}$

$$\frac{a}{2}(x + y) = bx$$

$$ax + ay = 2bx$$

$$ay = (2b - a)x$$

$$\frac{y}{x} = \frac{2b - a}{a}$$

Therefore. when $\frac{y}{x} > \frac{2b-a}{a}$, F_1 will decrease after stemming

when $\frac{y}{x} < \frac{2b-a}{a}$, F_1 will increase after stemming.

In conclusion, it is wrong that stemming always helps or hurts the F_1 score.

Intuitively, when the stemming word includes a lot of words that are not "same" as query word, F_1 score will be dominant by increasing in FP and will hurt F_1 score.
In opposite, if the stemming word include a lot of words that are "same" as query word, F_1 score will be dominant by decreasing in FN and will help F_1 score.