Module 3 Short Problem Set

Max Robinson

1 Problem 1

Question: Qualitatively explain the impact of using stemming on each of the following:
(a) vocabulary size;
(b) total number of postings in an inverted file;
(c) average posting list length?
The format of a good answer would be something like: With stemming XXXX {increases, decreases, doesnt change} by {a lot, a little, at all, roughly zz%} because of YYYY. I want to see a statement about the effect, it's magnitude, and a clear rationale.

a) Vocabulary size decreases by roughly 50% because of of the reduction in similar ending terms, or small variations to words. For example, run vs. running might both be mapped to the term "run". This helps cut down on the total number of terms in our vocabulary.

b) Total number of postings in an inverted file decreases by roughly 10-20% because of the reuse of different variations of a word in the same document. For example, if a document is about "running" it is likely that the words "running" and "run" both appear in the document. Prior to stemming, these would have been two different postings, but with stemming they are the same posting, and only change the count of the term for that document.

c) Average posting list length increases by a little because more tenses of a term correspond to the same term. This means that a tense of a word that might have been its own term and have it's own postings list is now in the same postings list as all other tenses. For example, "running" and "run" are now group in the same postings list. Any document that "running" was in that "run" wasn't in increases the size of the posting list for run. This increase happens for all terms, and thus the average posting list length increases a little bit.

2 Problem 2

Question: Express the numbers {8, 14, and 513} three ways: using a 12-bit binary representation, and the gamma and delta codes. You must follow the method for computing gamma/delta described in the text and presented in the lecture materials. I strongly recommend learning to do this by hand, but you may write (and provide) a short computer program if you prefer but do not use a program that you did not write yourself.

Note: the '+' in the following calculations means "concatenate".
Binary:

```
  8: 1000
 14: 1110
513: 1000000001
```

Gamma:
8:
$unary(floor(log8))$ + binary($8 - 2^{floor(log8)}$) in 3 bits
$unary(3)$ + binary($8 - 8$)
1110 000
= 1110000

14:
$unary(floor(log14))$ + binary($14 - 2^{floor(log15)}$) in 3 bits
$unary(3)$ + binary($14 - 8$)
1110 110
= 1110110

513:
$unary(floor(log513))$ + binary($513 - 2^{floor(log513)}$) in 9 bits
$unary(9)$ + binary($513 - 512$)
1111111110 000000001 = 1111111110000000001

Delta
8:
$gamma(floor(log8))$ + binary($8 - 2^{floor(log8)}$) in 3 bits
$gamma(3)$ + 000
$unary(floor(log3))$ + binary($3 - 2^1$) + 000
$unary(1)$ + binary(1) in 1 bit + 000
10 1 + 000
delta(8) = 101000

14:
$gamma(floor(log14))$ + binary($14 - 2^3$) in 3 bits
$gamma(3)$ + binary($14 - 8$) in 3 bits
$unary(floor(log3))$ + binary($3 - 2^1$) in 1 bit + 110
$unary(1)$ + binary(1) in 1 bit + 110
10 1 + 110

delta(14)= 101110

513:
$gamma(floor(log 513))$ + binary($513 - 2^9$) in 9 bits
$gamma(9)$ + binary($513 - 512$) in 9 bits
$unary(floor(log 9))$ + binary($9 - 2^3$) in 3 bit + 000000001
$unary(3)$ + binary(1) in 3 bit + 000000001
1110 001 + 000000001
delta(513)= 1110001000000001

3 Problem 3

Question: Below is a bit sequence for a gamma encoded gap list (as described in Chapter 5 of IIR and the lecture materials). Decode the gap list and reconstruct the corresponding list of docids. Spaces are added for ease of reading – the final part only has two bits. Hint: there are four docids.
1111 1100 0100 0111 1111 0010 0000 1111 1010 1011 1111 0100 00

1111 1100 0100 0|111 1111 0010 0000| 1111 1010 101|1 1111 0100 00
Gaps:
6 digits, 001000 → $2^6 + 001000 = 1001000 = 72$
7 digits, 0100000 → $2^7 + 0100000 = 10100000 = 160$
5 digits, 10101 → $2^5 + 10101 = 110101 = 53$
5 digits, 10000 → $2^5 + 10000 = 110000 = 48$

DocIDs:
first = 0 + 72 = 72
second = 72 + 160 = 232
third = 232 + 53 = 285
fourth = 285 + 48 = 333

4 Problem 4

Question: True or False – Any bit sequence (i.e., any combination of zeros and ones) can be interpreted as a valid gamma encoded list of integers? Explain why this is true, or give an example showing that it is not.

False. Example: 1101. The number of bits before the 0 does not equal the number of bits after the 0, and thus cannot be a gamma encoding. Also, the number of bits is even, which is impossible for a gamma encoding since the number of bits required for a gamma encoding is 2 * number of bits for the unary encoding, minus the zero + 1 (for the zero in the unary). This sum is always odd.

5 Problem 5

Question: Below is a bit sequence for a set of gaps encoding using Variable Byte encoding as described in Chapter 5 of IIR. Decode the list of gaps and reconstruct the corresponding list of docids. Hint: there are three docids.
1100 0001 0000 0011 1011 0011 0000 0100 0001 1111 1000 0011

11000001|0000001110110011|000001000001111110000011|
Note: the '+' in the following calculations means "concatenate".
Gaps:
first: ~~0~~100 0001 → 100 0001 = 1000 001 = 65
second: ~~0~~000 0011 + ~~1~~011 0011 → 000 0011 + 011 0011 = 0000 0110 1100 11 = 435
third: ~~0~~000 0100 + ~~0~~001 1111 + ~~1~~000 0011 → 000 0100 + 001 1111 + 000 0011 = 0000 1000 0111 1100 0001 1 = 69507

DocIDs:
first = 0 + 65 = 65
second = 65 + 435 = 500
third = 500 + 69507 = 70007