1.

I do not have the time to read the whole paper and write a 1 page summary on it, so I will take the -10 points for this question.

4.

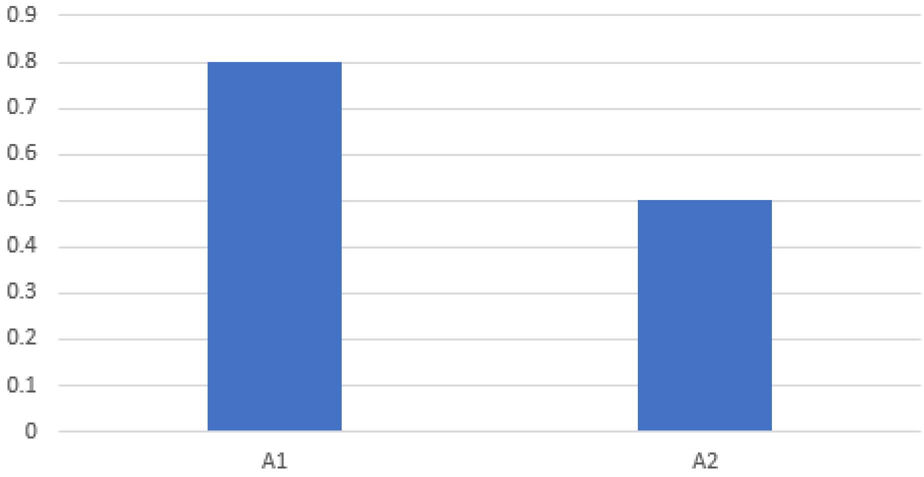
a.

P@5 = 0.6

NDCG @ 5 = 0.6813

b.

The system worked much better for the “espresso” query than for the “turkish coffee” query. I believe this is because if you are looking for just “espresso”, if you take how many times it is mentioned it will most likely be about espresso. When searching for “turkish coffee” you will get results that include “coffee” a lot, but not necessarily “turkish”.



6. The main difference between Byte-Pair Encoding and WordPiece is that WordPiece uses a formula to decide when to add to the vocabulary instead of frequency, which is what Byte-Pair Encoding uses.

WordPiece works like this: Go through all of the letters, the first letter of each word is tokenized, the rest are individually tokenized with a “##” in front of them. You then go through the text and take the unique pairs and calculate the scores for all the pairs. The formula is: frequency of pair/(freq of first element * freq of second element). You take the pair with the highest score, merge the pairs, and add it to the vocabulary. You recursively calculate and merge until the vocabulary is satisfactory.

(Round 1)

low	->	l, ##o, ##w
low	->	l, ##o, ##w
low	->	l, ##o, ##w
low	->	l, ##o, ##w
low	->	l, ##o, ##w
lowest	->	l, ##o, ##w, ##e, ##s, ##t
lowest	->	l, ##o, ##w, ##e, ##s, ##t
newer	->	n, ##e, ##w, ##e, ##r
newer	->	n, ##e, ##w, ##e, ##r
newer	->	n, ##e, ##w, ##e, ##r
newer	->	n, ##e, ##w, ##e, ##r
newer	->	n, ##e, ##w, ##e, ##r
newer	->	n, ##e, ##w, ##e, ##r
wider	->	w, ##i, ##d, ##e, ##r
wider	->	w, ##i, ##d, ##e, ##r
wider	->	w, ##i, ##d, ##e, ##r
new	->	n, ##e, ##w
new	->	n, ##e, ##w

Vocab (Round 1):

l, ##o, ##w, ##e, ##s, ##t, n, ##r, w, ##i, ##d

Pairs scores (Round 1):

l, ##o	=	1/2
##o, ##w	=	7 / 22
##w, ##e	=	4/17
##e, ##s	=	2 / 21
##s, ##t	=	1/2
n, ##e	=	8 / 27
##e, ##w	=	2 / 7
##e, ##r	=	9 / 28
w, ##i	=	1 / 2
##i, ##d	=	1 / 2
##d, ##e	=	3 / 22

(Round 2)

low	->	lo, ##w
low	->	lo, ##w
low	->	lo, ##w
low	->	lo, ##w
low	->	lo, ##w
lowest	->	lo, ##w, ##e, ##s, ##t
lowest	->	lo, ##w, ##e, ##s, ##t
newer	->	n, ##e, ##w, ##e, ##r
newer	->	n, ##e, ##w, ##e, ##r
newer	->	n, ##e, ##w, ##e, ##r
newer	->	n, ##e, ##w, ##e, ##r
newer	->	n, ##e, ##w, ##e, ##r
newer	->	n, ##e, ##w, ##e, ##r
wider	->	w, ##i, ##d, ##e, ##r
wider	->	w, ##i, ##d, ##e, ##r
wider	->	w, ##i, ##d, ##e, ##r
new	->	n, ##e, ##w
new	->	n, ##e, ##w

Vocab (Round 2):

l, ##o, ##w, ##e, ##s, ##t, n, ##r, w, ##i, ##d, lo

Pairs scores (Round 2):

lo, ##w	=	7 / 22
##w, ##e	=	. . .
##e, ##s	=	. . .
##s, ##t	=	. . .
n, ##e	=	. . .
##e, ##w	=	. . .
##e, ##r	=	. . .
w, ##i	=	. . .
##i, ##d	=	. . .
##d, ##e	=	. . .

Do this process recursively until your vocabulary is satisfactory.