COMP4650 Document Analysis Assignment1

Yutong Sun
U7409788

I acknowledge that some ideas are gathered from the 3 4650 lectures for IR

Question 1

map: 0.13176310670592079
Rprec: 0.15703823953823953
recip_rank: 0.28034818710146364
P_5: 0.15333333333333335
P_10: 0.12000000000000002
P_15: 0.08888888888888889

Question 2

map: 0.181245828157282
Rprec: 0.17239538239538243
recip_rank: 0.44355440362883936
P_5: 0.20666666666666667
P_10: 0.13666666666666666
P_15: 0.11111111111111113

Question 3

First Modification:

I use Lemmatization

Lemmatization is an important technic in the initial stage of text pre-processing. Since tokenization has been down, one candidate to continue to pre-process is Lemmatization. Lemmatization could reduce the influential term, and it use morphological analysis. Since the government report may use a large number of words, many of them may have been used with different forms, I think lemmatization could unify a single word with different forms, so they can be treaded as a single word.

The new performance is:

map: 0.1464180630065137
Rprec: 0.17013347763347764
recip_rank: 0.3253595055230332
P_5: 0.17333333333333334
P_10: 0.13
P_15: 0.0977777777777778

We can see it outperform the original performance, that is reported in question 1, with a slight increase in all the 5 metrics. I think it is working, because it is common that the document uses different forms of the same word. The possible success cases are making will be transformed to make, which will unify the word successfully. The possible failure cases may be that given the same word, stripes, and strip, with only the difference of ‘es’ It should be the same, however, if the lemmatizer treat stripes in the context of verb, it will return strip, and if the lemmatizer treat strip in the context of noun, it will return stripe. Therefore, it led to different outcomes due to different analysis context, which will cause error.

Second Modification
I use converting to lower case for all tokens and the use Lemmatization

The reason for using lemmatization is the same. And the intuition of using lowercase is that it is common that in government document, there are many words starting with a capital letter, so there may be many capitals letter for some words and using lower case will make words starting with capital letters or not become the same after pre-processing.

The new performance is:
map: 0.2517203926410078
Rprec: 0.25072871572871575
recip_rank: 0.462542339160985
P_5: 0.24000000000000002
P_10: 0.16999999999999998
P_15: 0.14000000000000004

We can the performance increase significantly compared to all previous results. The reason that it is working includes the reasons for second modification, and it is also because that there are many words in the document that starting with capitals, so using lowercase will make every word to lowercase, making the lemmatization has better effect. The possible cases also include the cases in the first modification, while for the lowercase step, a possible successful case is that Cinema and cinema will all becomes the same word, cinema. The possible failure cases may be that given that FUTURE is a company name, while future is just means the word future, then the lowercase will all convert them to future, which is erroneous, as they do not have the same meaning (one stands for company, and one stands for the future periods)

Third Modification:
I use converting to lower case for all tokens and the use Stemming

The reason for using lowercase is the same. Stemming is another important technic in the initial stage of text pre-processing. Since tokenization has been down, one candidate to continue to pre-process is Stemming. Since the government report may use a large number of words, many of them may have been used with different forms, as stemming removes last few characters of words, it may assist to unify words with different forms to a single word.

The new performance is:
map: 0.23429804534046902
Rprec: 0.20953823953823958
recip_rank: 0.41350964222154485
P_5: 0.21333333333333335
P_10: 0.16000000000000003
P_15: 0.12666666666666665

It is also good performance; however, it is slightly worse than the second modifications. It means the stemming works well for stemming different words into the single word correctly. One possible successful scenario is that eaten, and eating will both be stemmed to eat, which process these two words successfully as they have the same meaning, and the final form are the same. One possible failure scenario will be that for caring and care, after stemming, caring will become car, but care is still care, so they become different words, although they have the same meaning.

Method	map
original	0.13176310670592079
Mod. 1	0.1464180630065137
Mod. 2	0.2517203926410078
Mod. 3	0.23429804534046902

The metric used is map, that is mean average precision, which is fair and comprehensive to evaluate the performance of different modifications. We can all modifications outperform original method, which means our pre-processing works to some extent. Mod.2 outperform Mod. 1 suggest that lowercase is a useful tool when dealing with such documents. Mod.2 outperform Mod.3 suggests that sometimes lemmatization will outperform stemming, due to potential reasons that for such documents, the abovementioned successful scenario case is more likely to occur, and stemming may make more mistakes compared to lemmatization (such as the abovementioned example caring and care). However, their performance difference gap is not huge, and during the testing, I found that the time lemmatization used is much longer than stemming. Therefore, if the computation cost for lemmatization is too high or not manageable, stemming will also be a sufficient alternative method.

Question 4

Index length: 906290
Welcoming
./gov/documents/31/G00-31-2565694
./gov/documents/42/G00-42-4180551
./gov/documents/85/G00-85-0255215
./gov/documents/86/G00-86-2161870
./gov/documents/86/G00-86-4087434
./gov/documents/97/G00-97-2878104
./gov/documents/98/G00-98-1962568
Australasia OR logistic
./gov/documents/07/G00-07-3154026
./gov/documents/21/G00-21-0639911
./gov/documents/30/G00-30-2255236
./gov/documents/32/G00-32-0004755
./gov/documents/33/G00-33-2724973
./gov/documents/65/G00-65-1638589
./gov/documents/83/G00-83-3561112
./gov/documents/86/G00-86-0945012
heart AND warm
./gov/documents/13/G00-13-0161657

./gov/documents/19/G00-19-2662921
./gov/documents/19/G00-19-3991415
./gov/documents/28/G00-28-2840017
./gov/documents/42/G00-42-1673788
./gov/documents/47/G00-47-1858869
./gov/documents/74/G00-74-1894735
./gov/documents/83/G00-83-0224491
global AND space AND wildlife
./gov/documents/43/G00-43-1356582
./gov/documents/56/G00-56-3884873
./gov/documents/60/G00-60-1787423
./gov/documents/68/G00-68-4089689
./gov/documents/84/G00-84-0274223
./gov/documents/91/G00-91-4181375
./gov/documents/92/G00-92-2187116
engine OR origin AND record AND wireless
./gov/documents/49/G00-49-1102992
./gov/documents/71/G00-71-2668944
./gov/documents/98/G00-98-3148091
placement AND sensor OR max AND speed
./gov/documents/65/G00-65-3743536
./gov/documents/67/G00-67-4173730
./gov/documents/69/G00-69-4043921
./gov/documents/78/G00-78-0236359
./gov/documents/90/G00-90-2842409

Question 5

(a)

The data I would need is a collection of the documents,
The Query I would need is A test suite of information needs, which include all the Boolean operators designed in my IR system
The ground truth I would need is a set of relevance judgments, which can be used to make binary assessment of either relevant or irrelevant for each (query, document) pair

The challenges I would have to face include:
Huge data size: the data may be too large, and the implementation of the IR systems can be extremely slow
Skewness of data: some terms may occur too frequently, such as stop words, and some terms may be too rare, which will hinder the IR system performance
Faulty data: data may be nonsense or unstructured due to human errors

Metrics appropriate are precision, recall, accuracy, and F-measure, because for the Boolean query system, that is not ranked, the ranks of retrieved documents are not important, and we only need to compare the results relevant to the number of Retrieved documents and not retrieved documents. And these 4 metrics just do such jobs: they help to reflect how correct the system is and which part of the system goes wrong (from false positive and false negative)

(b)
The data I would need is a collection of the documents,
The Query I would need is A test suite of information needs
The ground truth I would need is a set of relevance judgments. However, since it is a ranked system, it needs additional function evaluate the score for retrieved document, based on the binary assessment

The challenges I would have to face include:
Huge data size: the data may be too large, and the implementation of the IR systems can be extremely slow
Skewness of data: some terms may occur too frequently, such as stop words, and some terms may be too rare, which will hinder the IR system performance
Faulty data: data may be nonsense or unstructured due to human errors

Metrics appropriate are Precision-recall curve, interpolated precision, as well as MAP and MRR. Precision-recall curve denotes the Precision and recall of the top-k retrieved documents and setting k to 10 just perfectly meet our needs. Interpolated precision makes it easier to interpret as it calculates the maximum precision at a given recall level. Map is the mean average precision and MRR is the Mean Reciprocal Rank, which provide information about the ranks of retrieved documents.