

# COMP4650 Assignment 1 Answers

Jieli Zheng

u6579712

## Q1 & Q2

TF scores:

Mean Average Precision: 0.16925271708009806

R-Precision: 0.19803030303030303

Reciprocal rank: 0.37175925925925923

Precision at 5: 0.19999999999999998

Precision at 10: 0.14666666666666667

Precision at 15: 0.09777777777777778

TF-IDF scores:

Mean Average Precision: 0.28087837129801413

R-Precision: 0.29668109668109666

Reciprocal rank: 0.5202777777777777

Precision at 5: 0.30000000000000004

Precision at 10: 0.21666666666666665

Precision at 15: 0.14444444444444446

## Q3

The system performs better with TF-IDF instead of TF similarity.

I think the Reciprocal rank is the most useful measure for evaluating this government system.

Firstly I assume that this information retrieval system is designed for government office to search for the previous official government documents. Officer usually has clearly targets so that only the exact document will be needed. And in this case it's a precision-critical task and there is only one relevant document in the system. As a result, the Reciprocal rank should be the most important measure depending on our demand.

### First setting:

PorterStemmer (Default), WordPunctTokenizer

Scores:

Mean Average Precision: 0.28525094870140955

R-Precision: 0.32620444072056975

Reciprocal rank (chosen): 0.56226318484383

Precision at 5: 0.3096774193548387

Precision at 10: 0.21612903225806449

Precision at 15: 0.1440860215053764

Firstly, we only change the Tokenizer to WordPunctTokenizer, which tokenize texts by using punctuations(';', ':', ',', '.', '!', '?'). Compared to the default setting in Q2, we find that Reciprocal rank of the first setting is better than the default one in Q2. The main reason for this performance is that splitting by punctuations is more powerful than only splitting by newline. Rarely will people write documents without using punctuations.

### Second setting:

PorterStemmer, TreebankWordTokenizer:

Scores:

Mean Average Precision: 0.29262703852584804

R-Precision: 0.3112445887445887

Reciprocal rank (chosen): 0.5533333333333333

Precision at 5: 0.32

Precision at 10: 0.22333333333333333

Precision at 15: 0.14888888888888889

In the second setting, the tokenizer has been changed to TreebankWordTokenizer. In this case, the performance(Reciprocal rank) of the second setting is slightly better than the default one. The main reason is that TreebankWordTokenizer splits text by not only Whitespaces, but also punctuations, standard contractions, and treating most punctuation characters as separate tokens.

**Third setting:**

SnowballStemmer with language set to English, WhitespaceTokenizer(Default)

Scores:

Mean Average Precision: 0.28767069934034223

R-Precision: 0.3176334776334776

Reciprocal rank (chosen): 0.5527777777777777

Precision at 5: 0.3266666666666666

Precision at 10: 0.21999999999999995

Precision at 15: 0.14666666666666667

In the third setting, we use SnowballStemmer in English and default tokenizer. And we find the Reciprocal rank is slightly higher than the default setting. SnowballStemmer is designed by the suffix stripping algorithm.

**Forth setting:**

ISRISemmer, WhitespaceTokenizer(Default)

Scores:

Mean Average Precision: 0.15132227032227033

R-Precision: 0.1723953823953824

Reciprocal rank (chosen): 0.4384656084656084

Precision at 5: 0.19333333333333336

Precision at 10: 0.14000000000000004

Precision at 15: 0.09333333333333335

In the forth setting, we use ISRISemmer and default tokenizer. ISRISemmer is based on Arabic Stemming without a root dictionary Algorithm and the Reciprocal rank of this setting is much worse than the default setting. The main reason is that ISRISemmer sometimes converts words to wrong word stems that changes the vector form of the document.

## Q4

In this section I firstly use: PorterStemmer (default), WhitespaceTokenizer (Default), BM25 similarity. Mean Average Precision: 0.393447515117158

R-Precision: 0.40248917748917745

Reciprocal rank (chosen): 0.6072222222222222

Precision at 5: 0.4266666666666667

Precision at 10: 0.2766666666666667

Precision at 15: 0.18444444444444444