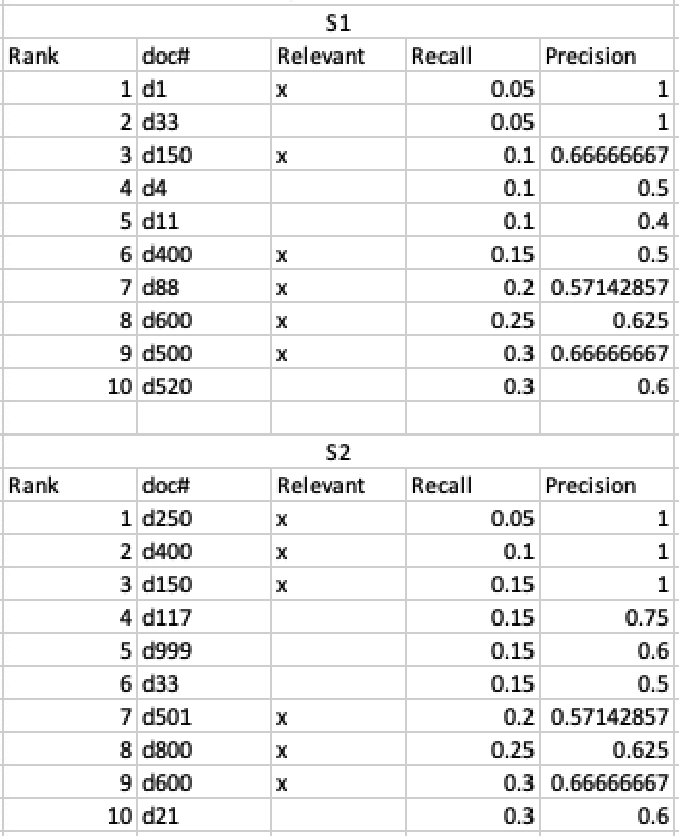
Assignment 1
Name: Serena Yang

I submitted the code with Jupyter Notebook and excel form. Here are only shows the final results for each question.

1. [25 pts] IR Evaluation

a.

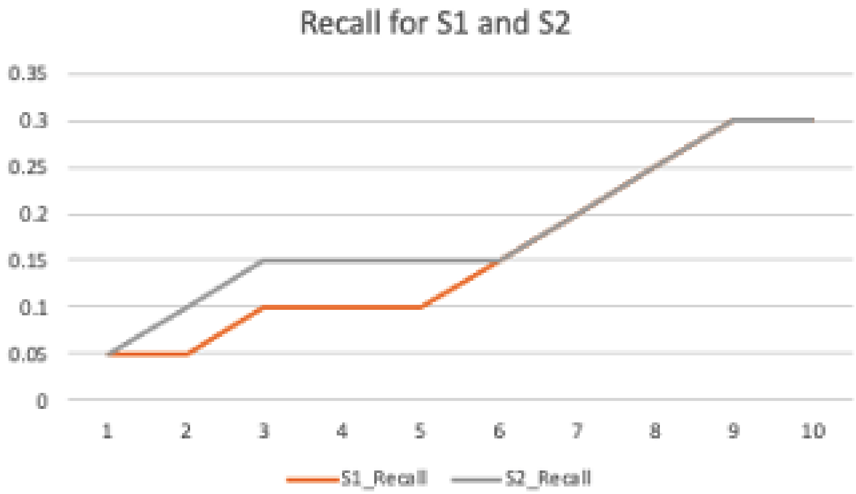


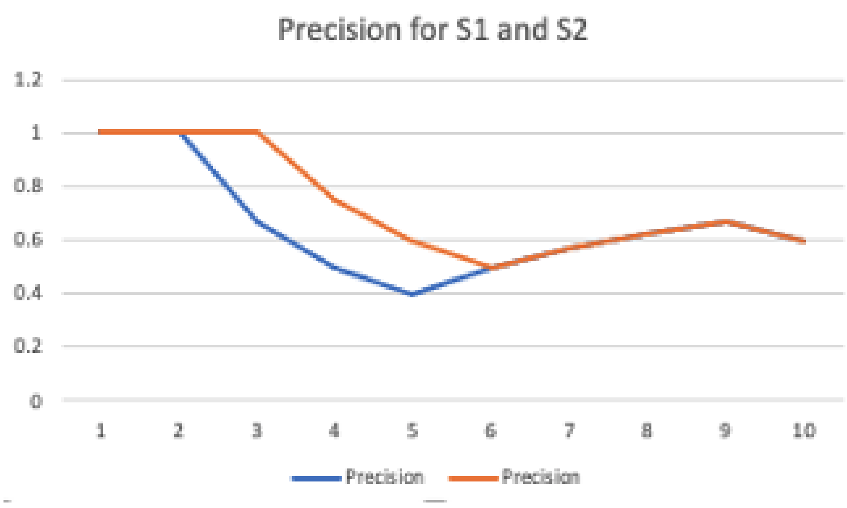
S1

Rank	doc#	Relevant	Recall	Precision
1	d1	x	0.05	1
2	d33		0.05	1
3	d150	x	0.1	0.66666667
4	d4		0.1	0.5
5	d11		0.1	0.4
6	d400	x	0.15	0.5
7	d88	x	0.2	0.57142857
8	d600	x	0.25	0.625
9	d500	x	0.3	0.66666667
10	d520		0.3	0.6

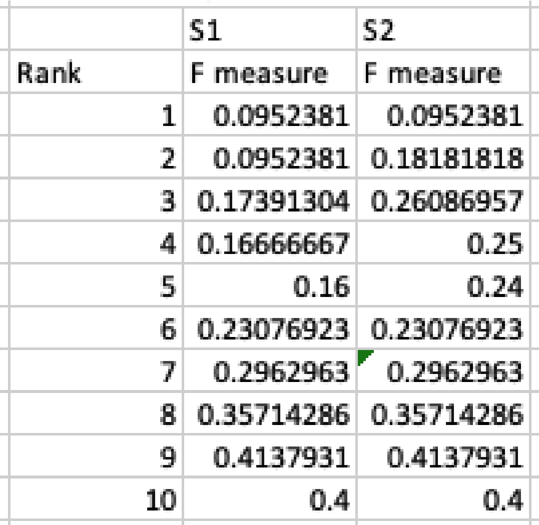
S2

Rank	doc#	Relevant	Recall	Precision
1	d250	x	0.05	1
2	d400	x	0.1	1
3	d150	x	0.15	1
4	d117		0.15	0.75
5	d999		0.15	0.6
6	d33		0.15	0.5
7	d501	x	0.2	0.57142857
8	d800	x	0.25	0.625
9	d600	x	0.3	0.66666667
10	d21		0.3	0.6

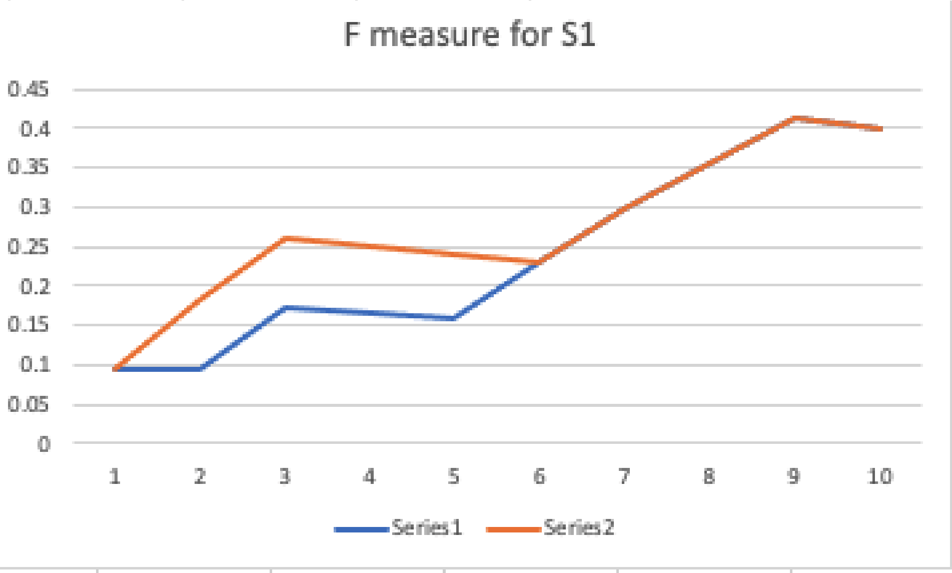




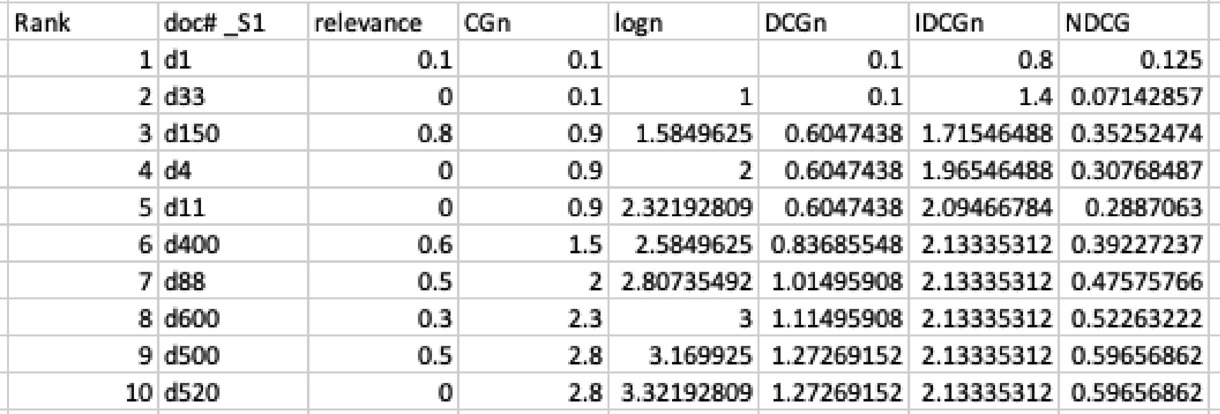
b.



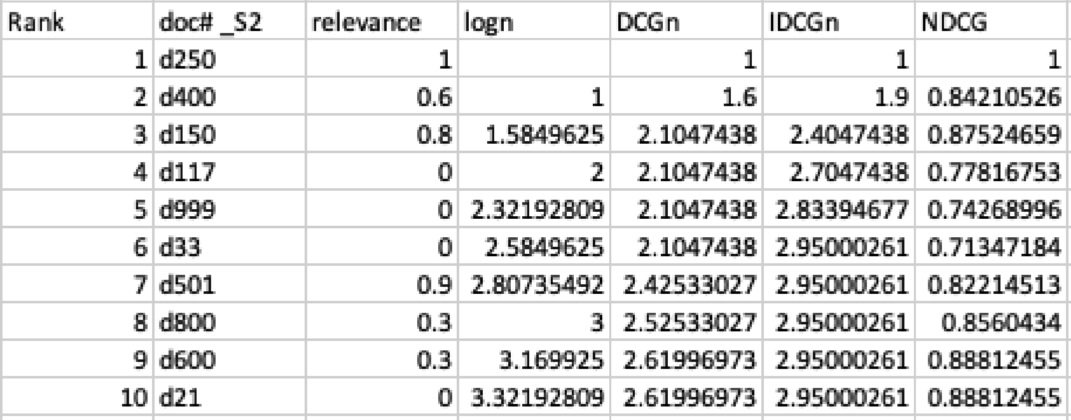
	S1	S2
Rank	F measure	F measure
1	0.0952381	0.0952381
2	0.0952381	0.18181818
3	0.17391304	0.26086957
4	0.16666667	0.25
5	0.16	0.24
6	0.23076923	0.23076923
7	0.2962963	0.2962963
8	0.35714286	0.35714286
9	0.4137931	0.4137931
10	0.4	0.4



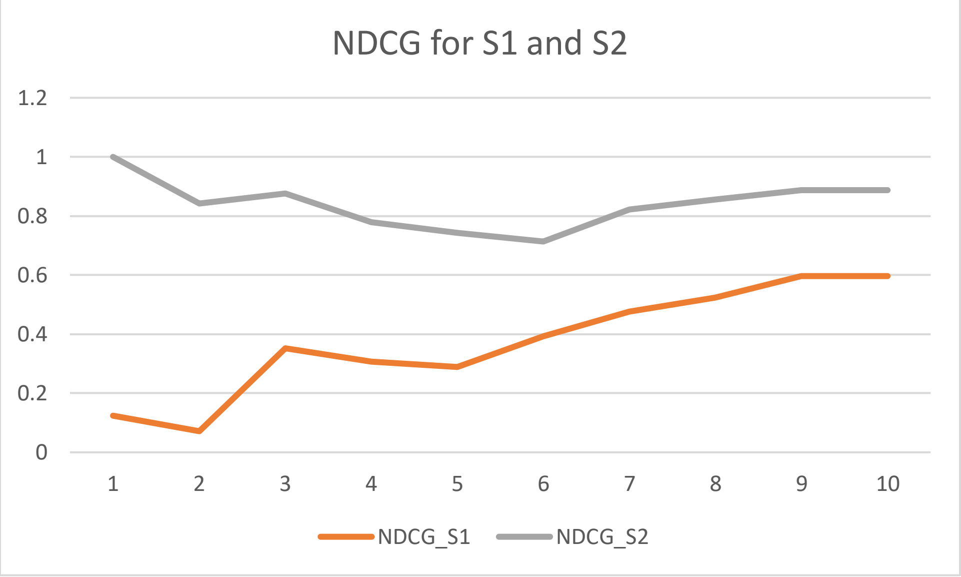
c.



Rank	doc# _S1	relevance	CGn	logn	DCGn	IDCGn	NDCG
1	d1	0.1	0.1		0.1	0.8	0.125
2	d33	0	0.1	1	0.1	1.4	0.07142857
3	d150	0.8	0.9	1.5849625	0.6047438	1.71546488	0.35252474
4	d4	0	0.9	2	0.6047438	1.96546488	0.30768487
5	d11	0	0.9	2.32192809	0.6047438	2.09466784	0.2887063
6	d400	0.6	1.5	2.5849625	0.83685548	2.13335312	0.39227237
7	d88	0.5	2	2.80735492	1.01495908	2.13335312	0.47575766
8	d600	0.3	2.3	3	1.11495908	2.13335312	0.52263222
9	d500	0.5	2.8	3.169925	1.27269152	2.13335312	0.59656862
10	d520	0	2.8	3.32192809	1.27269152	2.13335312	0.59656862

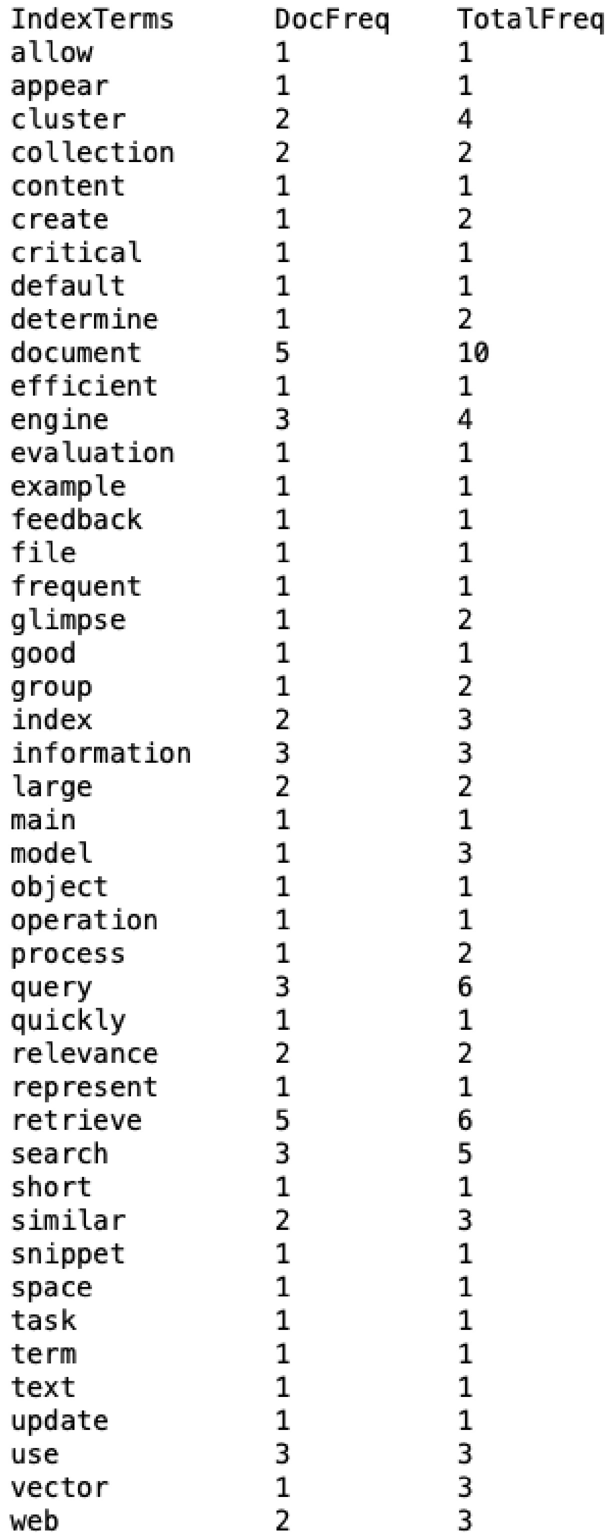


Rank	doc# _S2	relevance	logn	DCGn	IDCGn	NDCG
1	d250	1		1	1	1
2	d400	0.6	1	1.6	1.9	0.84210526
3	d150	0.8	1.5849625	2.1047438	2.4047438	0.87524659
4	d117	0	2	2.1047438	2.7047438	0.77816753
5	d999	0	2.32192809	2.1047438	2.83394677	0.74268996
6	d33	0	2.5849625	2.1047438	2.95000261	0.71347184
7	d501	0.9	2.80735492	2.42533027	2.95000261	0.82214513
8	d800	0.3	3	2.52533027	2.95000261	0.8560434
9	d600	0.3	3.169925	2.61996973	2.95000261	0.88812455
10	d21	0	3.32192809	2.61996973	2.95000261	0.88812455



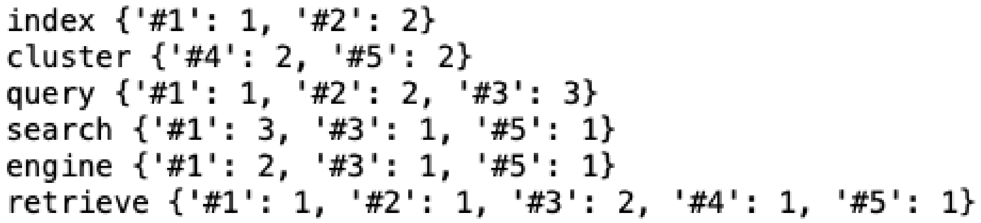
2. [40 pts] Inverted Indexes

a.



```
IndexTerms      DocFreq    TotalFreq
allow           1          1
appear          1          1
cluster         2          4
collection      2          2
content         1          1
create          1          2
critical        1          1
default         1          1
determine       1          2
document        5          10
efficient       1          1
engine          3          4
evaluation      1          1
example         1          1
feedback        1          1
file            1          1
frequent        1          1
glimpse         1          2
good            1          1
group           1          2
index           2          3
information     3          3
large           2          2
main            1          1
model           1          3
object          1          1
operation       1          1
process         1          2
query           3          6
quickly         1          1
relevance       2          2
represent       1          1
retrieve        5          6
search          3          5
short           1          1
similar         2          3
snippet         1          1
space           1          1
task            1          1
term            1          1
text            1          1
update          1          1
use             3          3
vector          1          3
web             2          3
```

b.



```
index {'#1': 1, '#2': 2}
cluster {'#4': 2, '#5': 2}
query {'#1': 1, '#2': 2, '#3': 3}
search {'#1': 3, '#3': 1, '#5': 1}
engine {'#1': 2, '#3': 1, '#5': 1}
retrieve {'#1': 1, '#2': 1, '#3': 2, '#4': 1, '#5': 1}
```

c. i. index AND query
{‘#1’, ‘#2’}

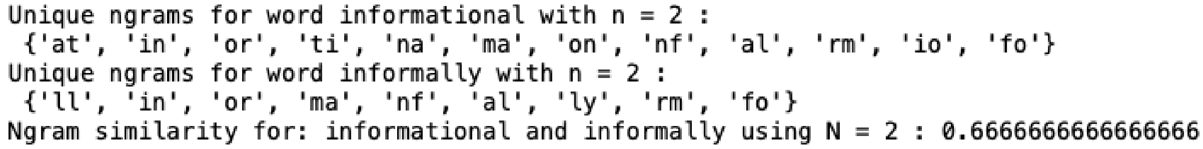
ii. index OR query
{‘#1’, ‘#2’, ‘#3’}

iii. index AND (NOT query)
{}

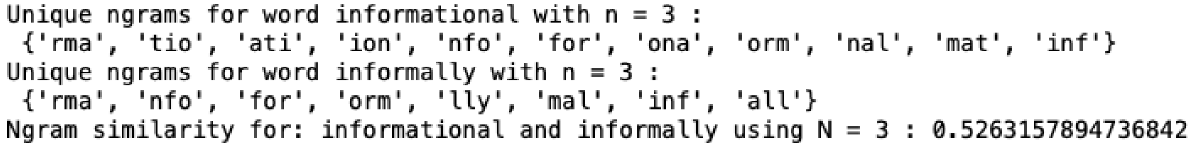
iv. (search AND query) OR (search AND retrieve)
{‘#1’,’#3’,’#5’}

v. (index OR cluster) AND (engine OR search)
{‘#1’, ‘#5’}

3. [25 pts] Character N-grams to Find Similar Terms
‘informational’ and ‘informally’

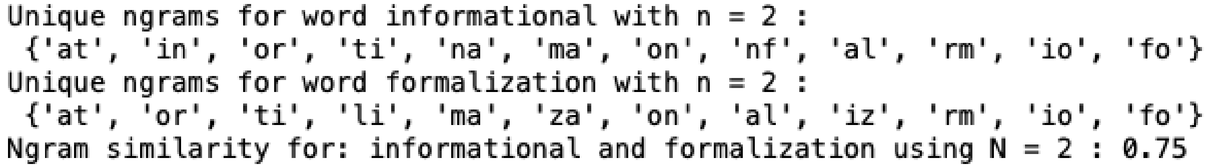


```
Unique ngrams for word informational with n = 2 :
 {'at', 'in', 'or', 'ti', 'na', 'ma', 'on', 'nf', 'al', 'rm', 'io', 'fo'}
Unique ngrams for word informally with n = 2 :
 {'ll', 'in', 'or', 'ma', 'nf', 'al', 'ly', 'rm', 'fo'}
Ngram similarity for: informational and informally using N = 2 : 0.6666666666666666
```

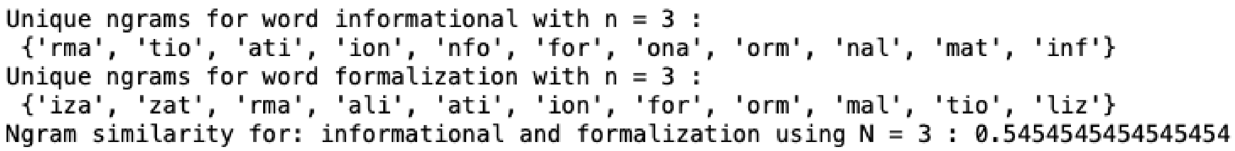


```
Unique ngrams for word informational with n = 3 :
 {'rma', 'tio', 'ati', 'ion', 'nfo', 'for', 'ona', 'orm', 'nal', 'mat', 'inf'}
Unique ngrams for word informally with n = 3 :
 {'rma', 'nfo', 'for', 'orm', 'lly', 'mal', 'inf', 'all'}
Ngram similarity for: informational and informally using N = 3 : 0.5263157894736842
```

‘informational’ and ‘formalization’

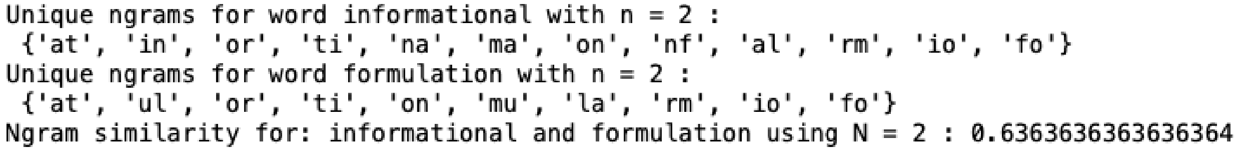


```
Unique ngrams for word informational with n = 2 :
 {'at', 'in', 'or', 'ti', 'na', 'ma', 'on', 'nf', 'al', 'rm', 'io', 'fo'}
Unique ngrams for word formalization with n = 2 :
 {'at', 'or', 'ti', 'li', 'ma', 'za', 'on', 'al', 'iz', 'rm', 'io', 'fo'}
Ngram similarity for: informational and formalization using N = 2 : 0.75
```

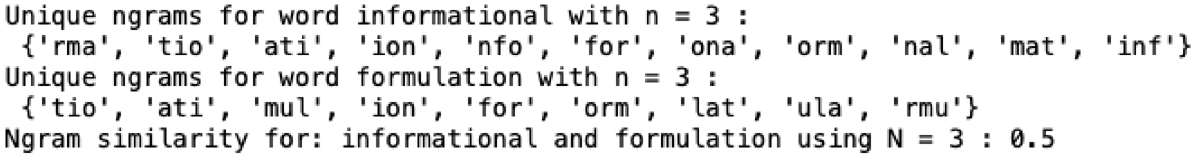


```
Unique ngrams for word informational with n = 3 :
 {'rma', 'tio', 'ati', 'ion', 'nfo', 'for', 'ona', 'orm', 'nal', 'mat', 'inf'}
Unique ngrams for word formalization with n = 3 :
 {'iza', 'zat', 'rma', 'ali', 'ati', 'ion', 'for', 'orm', 'mal', 'tio', 'liz'}
Ngram similarity for: informational and formalization using N = 3 : 0.5454545454545454
```

‘informational’ and ‘formulation’



```
Unique ngrams for word informational with n = 2 :
 {'at', 'in', 'or', 'ti', 'na', 'ma', 'on', 'nf', 'al', 'rm', 'io', 'fo'}
Unique ngrams for word formulation with n = 2 :
 {'at', 'ul', 'or', 'ti', 'on', 'mu', 'la', 'rm', 'io', 'fo'}
Ngram similarity for: informational and formulation using N = 2 : 0.6363636363636364
```



```
Unique ngrams for word informational with n = 3 :
 {'rma', 'tio', 'ati', 'ion', 'nfo', 'for', 'ona', 'orm', 'nal', 'mat', 'inf'}
Unique ngrams for word formulation with n = 3 :
 {'tio', 'ati', 'mul', 'ion', 'for', 'orm', 'lat', 'ula', 'rmu'}
Ngram similarity for: informational and formulation using N = 3 : 0.5
```

4. [10 pts] WordNet

WordNet is a word network. It contains semantic information of words compared with other dictionaries or standard thesauri. Based on the definition of each word, WordNet put the same meaning words into the same group, which is called 'synset.' Besides a word definition, WordNet also provided a brief description for each word using context. When the user searches a word, for example, 'world,' WordNet will return 'world' can be noun and adjective and return the semantic relation word and a context description.

As mentioned above, WordNet is a word-to-word relational dictionary. I think three features of WordNet distinguish it from other types of standard thesauri. Firstly, since the synset is the basic structure for WordNet, users can find more than one appropriate word to present an already known context in the same synset. Assume a WordNet is a dictionary; synset can be the key. Each value can be present as the set of same-meaning words. Secondly, WordNet is not only using synonym relation as its structure concept but also using synonymy, antonymy, hypernymy/hyponymy, meronymy, and entailment word-to-word relation. This structure concept makes the relation between words becomes easier to look up. Third, synset is not equal to a lemma in WordNet, but it is a key because one synset only includes one meaning explanation. However, one lemma could have multiple meanings and statements in a standard dictionary.

WordNet can be used in a few ways in the context of information retrieval. Firstly, it can use for synonyms. When the user has an already known concept, the program can be based on the user description to find the right words. Secondly, in different sentences, the word means different. For example, wordNet can get words' different meanings based on its hyponymy, troponymy, entailment, and meronymy. Also, WordNet can find related words that co-occurrence in a paper to get the word meaning based on the context. Another way is currently using, that combine with context window around six words, the result of using WordNet to eliminate definitions for the polyseme word is optimal. Combining linguistic context and in the similar context of co-occurrence words in WordNet, the accuracy for word sense disambiguation is highest. Semantic relationship in WordNet helps improve search results.