CE706 - Information Retrieval 2021

Assignment 2

2003930

Test collection (Task 1)

Information need	Query

What can we learn about coronaviruses by studying genes?	"query": {
	"match_phrase": {
	"title": "gene"
	}
	}

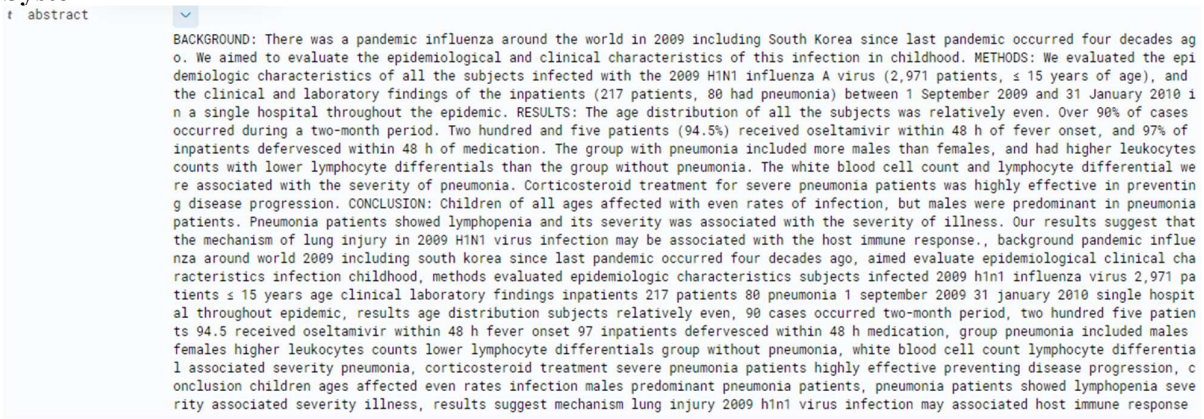
How do coronaviruses and other infectious diseases cause outbreaks?	"query": {
	"match_phrase_prefix": {
	"title": "outbreak"
	}
	}

What is the effects of coronaviruses and infectious diseases other viruses in America?	"query": {
	"match": {
	"title": "america usa"
	}
	}

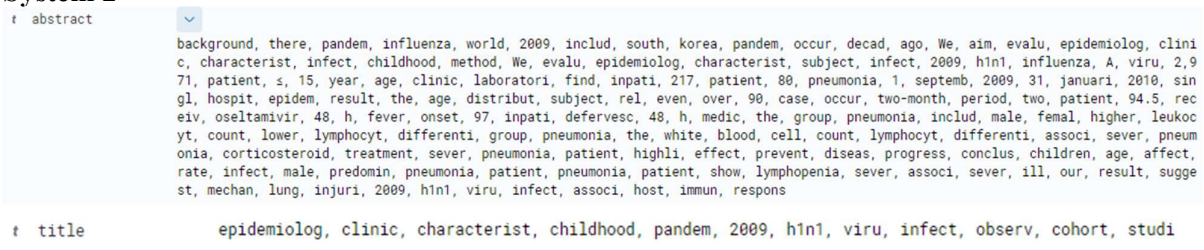
IR systems (Task 2)

The original system used is based on that used in the first assignment and then I made modifications from that system to create the second system. The two systems used have 5 differences between them. Some are much more impactful than the others. The first difference is that in system 1 I decided to implement lemmatization using the wordnet lemmatizer. In system 2 I decided to change this and implement stemming instead. This is done using a porter stemmer which is also from NLTK. This means that system 2 will look for the stem of a word and use that for indexing, this will likely make more errors than the method implemented in the first system. The second difference is that in the in system 1 each document gets split up into tokens of its sentences, whereas in system 2 the documents are split up into word tokens which are indexed. This means that system 2 will produce more tokens of a shorter length. The third difference is a minor one. This is how the stop words are generated. The first system uses the NLTK stopword corpus whereas the second uses the Gensim stopword corpus. These corpus' have a few very minor differences, so I do not expect this change to have a massive affect on the systems performance. The next change I made is that while system 1 converts all text to lowercase during the preprocessing steps, the second system does not do this. If the text in the document is uppercase, then system 2 will index it as so. The final difference between the two systems is that in system 1 the original document gets indexed alongside the tokens of the processed document. This does not happen in system 2. System 2 will only index the tokens generated from the preprocessing steps. I predict that system 1 will perform much better than the second system. Examples of the abstract and title fields from a document can be seen below to illustrate the differences.

System 1



System 2



Pool method (Task 3)

Query	# different documents	Id of the documents retrieve by System 1	Id of the documents retrieve by System 2

"query": {
 	
"match_phrase": {
“title”:
“gene”
}
}
	16	*bjjft7ut*
		x53t9i4k
		sbnnh2mm
		tloaa3v1
		jzwcy7dr
		k1hwh640
		nhb4o6ty
		hkc4vbmj
		dcjwfes7
		baugu1gh
			x53t9i4k
			bjjft7ut
			nhb4o6ti
			mxjtj5c0
			t6l692zu
			6d9x0xbj
			31q7ftnd
			sbnnh2mm
			dcjwfes7
			hkc4vbmj

"query": {
 	
"match_phrase_prefix": {
“title”:
“outbreak”
}
}
	11	*ttyo4z6f*
		t7004uw2
		2mfbqs8i
		31q7ftnd
		crjwej14
		5b936n3g
		fite9vs8
		h3yxymh3
		9fr0m92p
		36dhfptw
			t7004uw2
			ttyo4z6f
			31q7ftnd
			crjwej14
			2mfbqs8i
			5b936n3g
			h3yxymh3
			36dhfptw
			fite9vs8
			2ad1tu4

"query": {
"match": {
"title":
"america usa"
}
}
	4	*7vhcf929*
		hwlvk68z
		ntx35a8s
		v9k7vpi8
			7vhcf929
			hwlvk68z
			ntx35a8
			v9k7vpi8

Relevance assessments (Task 4)

Relevance criteria: Documents give some detail which can be used to answer the question set out by the information needs. Does not have to answer the question. But must outline some example of information which could be used to answer the question.

Query	**ID of relevant documents**

"query": {
"match_phrase": {
"title": "gene"
}
}
	x53t9i4k
	k1hwh640
	nhb4o6ty
	hkc4vbmj
	dcjwfes7
	baugu1gh
	
	mxjtj5c0
	t6l692zu
	31q7ftnd

"query": {
 	
"match_phrase_prefix": {
"title": "outbreak"
}
}
	ttyo4z6f
	t7004uw2
	2mfbqs8i
	31q7ftnd
	crjwej14
	5b936n3g
	fite9vs8
	h3yxymh3
	9fr0m92p
	36dhfptw

"query": {
"match": {
"title": "america usa"
}
}
	hwlvk68z
	ntx35a8s
	v9k7vpi8

Evaluation (Task 5)

To calculate the precision and recall I created functions which took a list of the ids of the relevant documents in each query and then a second argument of the results provided by a system.

Pseudocode for P@K:

k_pred = the first k values in prediction list
actual = relevant documents
correct_values = documents which appear in k_pred AND acutal
pk = correct_values/k
return pk

Pseudocode for R@K:

correct_values = documents which appear in the first k values of prediction list AND items in the relevant document list
if correct_values is 0
return 0
else
return correct_values / relevant document list length

	System 1		**System 2**	
	P@5	**R@5**	**P@5**	**R@5**
Q1	0.2	0.11	0.6	0.33
Q2	1.0	0.5	1.0	0.5
Q3	0.6	0.75	0.6	0.75

Discussion: In my initial discussion of the two systems I predicted that system 1 would perform much better than system 2. However, the results above show a slightly different story. Both query 2 and 3 provide us the exact same results. However, in the first query you can see that system 2 is more accurate than the first. However, it is worth noting that the sample size of the data used is small which is likely part of the reason for both query 2 and 3 yielding the same results across the two systems. Especially in the case of query 3 where each system returned the same 4 results – potentially the query was too specific for this and the information need needed to be wider.

In the first query I was looking to solve the information need of "What can we learn about coronaviruses by studying genes?". The simplest way to try and solve this was by looking for the phrase of gene. By using the match_phrase parameter it allowed me to include words such as "genetic" which would also be applicable. However, it also includes words such as "generally" in the scope. This is not what we want to include. The reason for the score of system 1 being so low is that it included more of these sorts of words than system 2 and therefore had less relevant documents. This is likely since system 1 used lemmatization as supposed to stemming which return different values.