CE706 – SU - Information Retrieval 2022

Assignment 2

2112102

Test collection (Task 1)
Include here the selected information needs and how they will be represented as a query.

Information need	Query

Finding out how state football team has performed?	"query":
	{
	"match_phrase":
	{
	"title": "state football team"
	}
	}

Finding out when regional parliament election is held?	"query":
	{
	"match_phrase":
	{
	"title": "regional parliament election"
	}
	}

What are the new achievements of Indian Space Research Organisation?	"query":
	{
	"match_phrase":
	{
	"title": "Indian Space Research Organisation"
	}
	}

IR systems (Task 2)

There are one million articles in the given database, These pieces drew their information from a wide variety of sources, ranging from massive establishments like Reuters to more intimate venues like blogs and regional news websites. The dataset contains 734,488 news articles and 265,512 blog posts, each of which has an average word count of 405 (I assume), and the number of words in the posts ranges from one to ten thousand. I have used elasticsearch and Kibana to perform the given tasks.
The System 1 I have used here is a based on the system I have used in first assignment, but I have made some changes to create a system 2. Where first system is better than the second system as second system, I implemented porter stemmer means it will find stem of the word and then it will use that to do indexing, it will definitely make more errors than system 1. System 1 used to split the document into tokens, but system 2 splits the only indexed words. System 1 converts texts into lowercase while preprocessing the data, but what system 2 does is, it doesn’t converts the text into lowercase if it is in the uppercase then it will index it as it is. In system 1 document will also be indexed along with the token of processed document. But in system 2 it indexes only tokens which are generated in preprocessing step. Hence, I think system 1 must perform better than system 2.

Pool method (Task 3)

For each system here I have retrieved the top 10 documents I got as a result. So, I have retrieved 20 documents for each query.

	Query 1		***Query 2***		***Query 3***	
Rank	***System 1***	***System 2***	***System 1***	***System 2***	***System 1***	***System 2***
1	*6d899e17-59a0-4113-a86c-438ec499ac92*	*328d30b2-9995-45e0-b5d8-69c3f40b9e1a*	*131d4d64-b51e-489d-babc-5c98fbae9bda*	*2f5d7402-9835-4f53-b498-1463a6d8b47e*	*bc87eab5-11dc-4e13-b546-3af18b369a86*	*8b2dd51c-bbb2-4698-bbd4-d946eed6b537*
2	*ed237dc5-99d0-4df8-971e-a45bc5b38a8d*	*3da4d5fa-19cd-4fef-8f49-c9757b2cce42*	*4e645131-df44-4a07-9c80-12098585d8f1*	*c4d744f6-be2d-4b72-8f30-203e27811c14*	*083b9f46-c810-43ea-a750-6e399a80dce2*	*bba79558-f331-4a83-8030-6de77b816bbe*
3	*26fb026f-89c4-4399-a0d0-097c94dec169*	*66bc39e4-5c79-4926-a7a3-b8ba098c1567*	*c09a8f40-7d4d-4a72-834b-649f90f91e5a*	*e2f4b972-3fdd-42c1-8b1c-2aee2d442ef7*	*9332bf2c-aca2-493c-9760-2f030d38a248*	*829048ff-eaac-48d9-88f3-842116d7d9e0*
4	*1e5dc25c-0464-4960-ab1d-3e8b3332b23f*	*794c0a91-c240-497f-8e24-32ce0a740875*	*b1075450-f352-47bd-ac0a-caef497c7efe*	*4e645131-df44-4a07-9c80-12098585d8f1*	*8a1accf6-cea3-408e-a559-3123b51fd5f7*	*ad9d1bdb-b230-42dc-bceb-09f63e18b8ff*
5	*129dec80-2dc8-48e7-bef5-76cf384f6f77*	*ed237dc5-99d0-4df8-971e-a45bc5b38a8d*	*fb4f9da4-971a-4fd0-9ac6-80c5615bb276*	*131d4d64-b51e-489d-babc-5c98fbae9bda*	*5e276735-6637-4c1e-a99e-72dc7425bf99*	*3cac7a47-d047-4cfb-b102-4362954a435b*
6	*66bc39e4-5c79-4926-a7a3-b8ba098c1567*	*a577f46c-b154-4bdf-ad24-0efbbd28b545*	*97395b86-5f7f-46d8-9288-16561d370754*	*1f0fb31d-947d-4a64-8775-9650718989a9*	*60218eee-385b-4003-9a86-17516a631a2c*	*8a1accf6-cea3-408e-a559-3123b51fd5f7*
7	*794c0a91-c240-497f-8e24-32ce0a740875*	*6d67adbf-0c47-46c0-89ab-bde33abd6df7*	*1f0fb31d-947d-4a64-8775-9650718989a9*	*bbd19509-cccf-4b8d-acb8-7a802d5295e3*	*bdf181ab-073d-47c2-a898-d6dc07d0d6d6*	*96e175cc-0c03-439a-884b-13ac6234bee4*
8	*9447e337-8bb1-429e-bdae-879ebe21285f*	*e0ee19d3-75e8-41d5-b0da-2f9aba35c231*	*9281c574-1c2e-41c1-9063-38a86e6e0389*	*634cd5c8-d7a2-44ce-87f5-1e419c767ae0*	*ad9d1bdb-b230-42dc-bceb-09f63e18b8ff*	*8b4c315d-e8a6-4e78-981e-586fdd5eefa6*
9	*95da056e-9949-458a-bae5-76efeeb49bb1*	*129dec80-2dc8-48e7-bef5-76cf384f6f77*	*6d160a5a-2ea1-4bf7-af04-6d65c011a06b*	*fb4f9da4-971a-4fd0-9ac6-80c5615bb276*	*3cac7a47-d047-4cfb-b102-4362954a435b*	*441b63ad-ea70-478d-a691-9d89b785df1b*

10	*ade98406-d404-404c-88e3-a943dbd54f9d*	*061c1f06-68a6-4c18-ac4c-3ac9cf2d204f*	*55f48fe7-80a0-4260-a5a7-1b4335644e47*	*b1075450-f352-47bd-ac0a-caef497c7efe*	*4b6f42b4-1d35-46da-a8aa-3494f485f01e*	*4fb249fd-da0e-4a51-97af-6fc30f1273eb*
Different documents	*12*		*12*		*14*	

Relevance assessments (Task 4)

Relevance criteria:
Here every relevant documents are being collected from the results of system 1 and system 2. The documents which give the perfect information that gives the answer to our question or query.

Fill the following table with the ID of the relevant documents

	ID of relevant documents
Query 1	*794c0a91-c240-497f-8e24-32ce0a740875* *ed237dc5-99d0-4df8-971e-a45bc5b38a8d* *66bc39e4-5c79-4926-a7a3-b8ba098c1567* *129dec80-2dc8-48e7-bef5-76cf384f6f77* *6d899e17-59a0-4113-a86c-438ec499ac92* *95da056e-9949-458a-bae5-76efeeb49bb1*
Query 2	*131d4d64-b51e-489d-babc-5c98fbae9bda* *1f0fb31d-947d-4a64-8775-9650718989a9* *fb4f9da4-971a-4fd0-9ac6-80c5615bb276* *b1075450-f352-47bd-ac0a-caef497c7efe* *6d160a5a-2ea1-4bf7-af04-6d65c011a06b* *634cd5c8-d7a2-44ce-87f5-1e419c767ae0* *55f48fe7-80a0-4260-a5a7-1b4335644e47* *4e645131-df44-4a07-9c80-12098585d8f1*
Query 3	*ad9d1bdb-b230-42dc-bceb-09f63e18b8ff* *3cac7a47-d047-4cfb-b102-4362954a435b* *8a1accf6-cea3-408e-a559-3123b51fd5f7* *8b4c315d-e8a6-4e78-981e-586fdd5eefa6* *441b63ad-ea70-478d-a691-9d89b785df1b* *bdf181ab-073d-47c2-a898-d6dc07d0d6d6* *60218eee-385b-4003-9a86-17516a631a2c* *bc87eab5-11dc-4e13-b546-3af18b369a86* *083b9f46-c810-43ea-a750-6e399a80dce2* *4b6f42b4-1d35-46da-a8aa-3494f485f01e* *4fb249fd-da0e-4a51-97af-6fc30f1273eb*

Evaluation (Task 5)

For calculating the precision and recall I have developed the functions using pseudocode. It takes ids which are in relevant documents and the next result generated by our system.

```
For P@K

pred_docs = the first k docs in prediction list
actual_docs = appropriate documents
correct_doc_values = docs in pred_docs AND actual
Pk = correct_doc_values/k
return Pk

for R@K:

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

Query 1:

Rank	System 1	P@5	R@5	System 2	P@5	R@5
1	***6d899e17-59a0-4113-a86c-438ec499ac92***	=1/1	=1/6	*328d30b2-9995-45e0-b5d8-69c3f40b9e1a*	'=0/1	'=0/6
2	***ed237dc5-99d0-4df8-971e-a45bc5b38a8d***	=2/2	=2/6	*3da4d5fa-19cd-4fef-8f49-c9757b2cce42*	'=0/2	'=0/6
3	*26fb026f-89c4-4399-a0d0-097c94dec169*	=2/3	=2/6	***66bc39e4-5c79-4926-a7a3-b8ba098c1567***	=1/3	=1/6
4	*1e5dc25c-0464-4960-ab1d-3e8b3332b23f*	=2/4	=2/6	***794c0a91-c240-497f-8e24-32ce0a740875***	=2/4	=2/6
5	**129dec80-2dc8-48e7-bef5-76cf384f6f77**	**=3/5**	**=3/6**	***ed237dc5-99d0-4df8-971e-a45bc5b38a8d***	**=3/5**	**=3/6**

Query 2:

Rank	System 1	P@5	R@5	System 2	P@5	R@5
1	***131d4d64-b51e-489d-babc-5c98fbae9bda***	=1/1	=1/8	*2f5d7402-9835-4f53-b498-1463a6d8b47e*	=0/1	=0/8
2	***4e645131-df44-4a07-9c80-12098585d8f1***	=2/2	=2/8	*c4d744f6-be2d-4b72-8f30-203e27811c14*	=0/2	=0/8
3	*c09a8f40-7d4d-4a72-834b-649f90f91e5a*	=2/3	=2/8	*e2f4b972-3fdd-42c1-8b1c-2aee2d442ef7*	=0/3	=0/8
4	***b1075450-f352-47bd-ac0a-caef497c7efe***	=3/4	=3/8	***4e645131-df44-4a07-9c80-12098585d8f1***	=1/4	=1/8
5	***fb4f9da4-971a-4fd0-9ac6-80c5615bb276***	**=4/5**	**=4/8**	***131d4d64-b51e-489d-babc-5c98fbae9bda***	**=2/5**	**=2/8**

Query 3:

Rank	**System 1**	**P@5**	**R@5**	**System 2**	**P@5**	**R@5**
1	***bc87eab5-11dc-4e13-b546-3af18b369a86***	=1/1	=1/11	*8b2dd51c-bbb2-4698-bbd4-d946eed6b537*	=0/1	=0/11
2	***083b9f46-c810-43ea-a750-6e399a80dce2***	=2/2	=2/11	*bba79558-f331-4a83-8030-6de77b816bbe*	=0/2	=0/11
3	*9332bf2c-aca2-493c-9760-2f030d38a248*	=2/3	=2/11	*829048ff-eaac-48d9-88f3-842116d7d9e0*	=0/3	=0/11
4	***8a1accf6-cea3-408e-a559-3123b51fd5f7***	=3/4	=3/11	***ad9d1bdb-b230-42dc-bceb-09f63e18b8ff***	=1/4	=1/11
5	*5e276735-6637-4c1e-a99e-72dc7425bf99*	**=3/5**	**=3/11**	***3cac7a47-d047-4cfb-b102-4362954a435b***	**'=2/5**	**'=2/11**

Results:

	System 1		**System 2**	
	P@5	**R@5**	**P@5**	**R@5**
Q1	0.6	0.5	0.6	0.5
Q2	0.8	0.5	0.4	0.25
Q3	0.6	0.272727	0.4	0.18181818

Web search (Task 6)

Here below I have explained the differences between both the systems.

System 1	System 2
Implemented lemmatization (wordnet lemmatizer)	Implemented stemming (porter stemmer)
Each sentence of document gets splitted	Only indexed sentences splitted into tokens
It uses the stopword corpus (NLTK)	It uses stop word corpus (Gensim)
During preprocessing It converts text into lowercase.	It will index uppercase document as it is.
The original document will be indexed with the processed document	It will index only tokens which are generated during preprocessing step.

By doing all these changes and by seeing the results of system 1 and system 2, as well as P@5 and R@5 score, I think that system 1 is working better than the system 2. So, I would recommend system 1 for web search.