CE706 - Information Retrieval 2021

Assignment 2

Student ID : 2004458

Test collection (Task 1)

Test collection in Information Retrieval systems consists of a collection of documents, a sample of queries(a query is nothing, but search string formed by the user in order to obtain the information needed by him/her from a collection of documents) and a list of relevant documents for each query.

To evaluate the efficiency of any Information retrieval system, we need a test collection. To create the test collection we need the documents, which contains the information we are searching for. Going through each document manually is time consuming process, hence I chose to use Kibana to form the test collection. For Information retrieval system1 and system2, 1000 documents are indexed using Elastic Search. The following 3 queries are searched using Kibana and the top ten relevant documents(based on relevancy score, higher the score more relevant is the document to query we are searching for) are collected for each query. **This forms the test collection(query, document ids in which this query is present).**

Information need	**Query**
What is the role of environment in virus transmission?	Effect of environmental factors on virus
What are the steps implemented in airports and aircrafts to reduce the risk of covid-19?	Covid-19 air travel safety
How is the virus spreading in community?	Transmission of virus in community

A csv file is created with this test collection. This is how the csv file looks like, it has a query and corresponding document 'pmcid' in which the query is present.

Effect of environmental factors on virus	PMC32945	PMC32654	PMC27975	PMC28217	PMC17797	PMC28372	PMC35851	PMC29093	PMC27701	PMC1351169
Covid-19 air travel safety	PMC3577(	PMC29502	PMC28132	PMC29398	PMC2797(	PMC27654	PMC30327	PMC33147	PMC2781(	PMC2823611
Transmission of virus in community	PMC28932	PMC28515	PMC28761	PMC32661	PMC33243	PMC3227(	PMC34841	PMC3447(	PMC3222(	PMC2804000

IR systems (Task 2)

1000 documents were selected from the whole collection, title and abstract column were used to create keywords. Both these columns are combined and pre-processed to create the keywords. These keywords are added as separate column and documents are indexed using ElasticSearch.

```
{
    "publish_time" => "2013-03-13",
           "path" => "C:/Users/jhans/OneDrive/Documents/docs2_IR2.csv",
            "sha" => "df783d511b145a10e7f609a87392eb50799d2b2b",
       "abstract" => "NOA36/ZNF330 is an evolutionarily well-preserved protein present in the nucleolus and mitochondria of mammalian cells. We have previously reported that the pro-apoptotic activity of this
 protein is mediated by a characteristic cysteine-rich domain. We now demonstrate that the nucleolar localization of NOA36 is due to a highly-conserved nucleolar localization signal (NoLS) present in residues 1?
33. This NoLS is a sequence containing three clusters of two or three basic amino acids. We fused the amino terminal of NOA36 to eGFP in order to characterize this putative NoLS. We show that a cluster of three
lysine residues at positions 3 to 5 within this sequence is critical for the nucleolar localization. We also demonstrate that the sequence as found in human is capable of directing eGFP to the nucleolus in sever
al mammal, fish and insect cells. Moreover, this NoLS is capable of specifically directing the cytosolic yeast enzyme polyphosphatase to the target of the nucleolus of HeLa cells, wherein its enzymatic activity
was detected. This NoLS could therefore serve as a very useful tool as a nucleolar marker and for directing particular proteins to the nucleolus in distant animal species.",
        "journal" => "PLoS One",
  "pdf_json_files" => "document_parses/pdf_json/df783d511b145a10e7f609a87392eb50799d2b2b.json",
       "@version" => "1",
        "license" => "cc-by",
  "pmc_json_files" => "document_parses/pmc_json/PMC3596294.xml.json",
            "doi" => "10.1371/journal.pone.0059065",
       "source_x" => "PMC",
          "pmcid" => "PMC3596294",
      "pubmed_id" => "23516598",
        "authors" => "de Melo, Ivan S.; Jimenez-Nuñez, Maria D.; Iglesias, Concepción; Campos-Caro, Antonio; Moreno-Sanchez, David; Ruiz, Felix A.; Bolívar, Jorge",
            "url" => "https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3596294/",
          "title" => "NOA36 Protein Contains a Highly Conserved Nucleolar Localization Signal Capable of Directing Functional Proteins to the Nucleolus, in Mammalian Cells",
     "@timestamp" => 2021-04-09T06:32:46.254Z,
       "cord_uid" => "0jx6mwiw",
   "text_stemmed" => "noa protein contain highli conserv nucleolar local signal capabl direct function protein nucleolu mammalian cellsnoa znf evolutionarili well preserv protein present nucleolu mitochondria
 mammalian cell previous report pro apoptot activ protein mediat characterist cystein rich domain demonstr nucleolar local noa due highli conserv nucleolar local signal nol present residu nol sequenc contain thr
ee cluster two three basic amino acid fuse amino termin noa egfp order character put nol show cluster three lysin residu posit within sequenc critic nucleolar local also demonstr sequenc found human capabl direc
t egfp nucleolu sever mammal fish insect cell moreov nol capabl specif direct cytosol yeast enzym polyphosphatas target nucleolu hela cell wherein enzymat activ detect nol could therefor serv use tool nucleolar
marker direct particular protein nucleolu distant anim speci",
           "host" => "LAPTOP-BHI84IJF",
```


The pre-processing includes:

1. Removal of punctuations and numbers
2. Removing stop words
3. Converting to lower case
4. Removing words that are less than length 2
5. Tokenizing the sentences
6. **Lemmatization(IR1) and Stemming (IR2)**

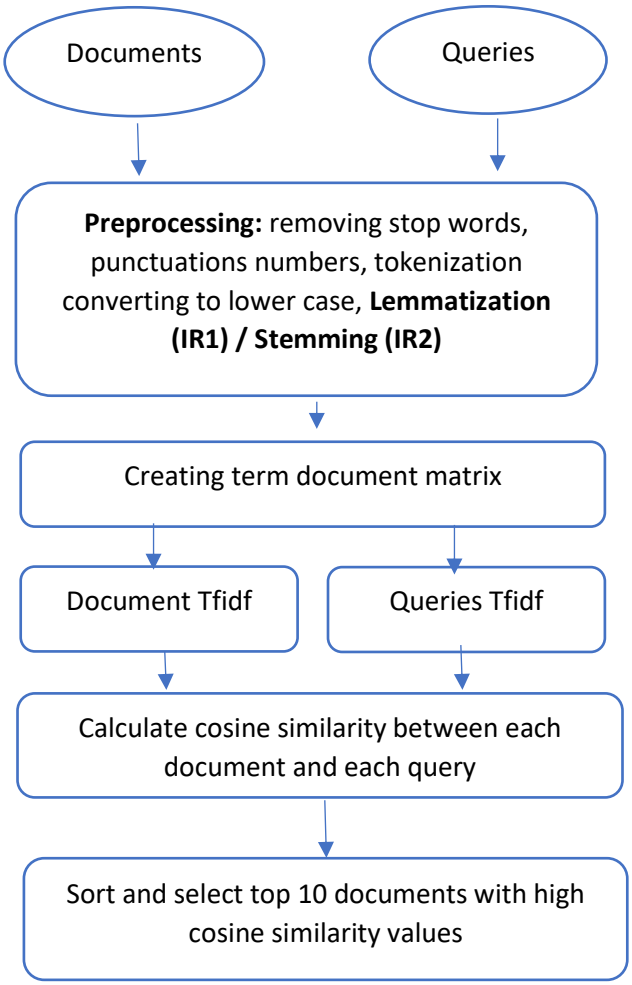
For constructing Information Retrieval system 1(IR 1), Lemmatization was used extract the base forms of words, whereas for IR2 stemming is used as normalization technique. The following snippet shows the pre-processing steps followed for IR1 and IR2:

```
def preprocess_IR1(text):
    text = re.sub('[^a-zA-Z]',' ',text) # removing numbers and punctuations
    text = str(text).lower()            # convert all characters into lowercase
    text = word_tokenize(text)          # tokenization
    text = [item for item in text if item not in stop_words] # removing stopwords
    text = [lemma.lemmatize(word=w,pos='v') for w in text]   # lemmatization
    text = [i for i in text if len(i) > 2]   # removing token of length <=2
    text = ' '.join(text)                    # joining the tokens with space in between to form sentence

    return text
def preprocess_IR2(text):
    text = re.sub('[^a-zA-Z]',' ', text)
    text = str(text).lower()
    text = word_tokenize(text)
    text = [item for item in text if item not in stop_words]
    text = [stemmer.stem(token) for token in text]          # Stemming
    text = [i for i in text if len(i) > 2]
    text = ' '.join(text)

    return text
```

Vector Space Model (IR1 and IR2)



The above diagram shows the steps implemented in creating IR1 and IR2 (vector space models).

VSM for Information retrieval represents documents and queries are vectors of weights.

In the vector space model(VSM) is an algebraic model, which involves two steps.

- In the first step each document in the corpus is broken down into words, by applying pre-processing steps, after this each document is represented as vector of words.
- In the second step the created word vectors are transformed into numerical format as term document matrix using CountVectorizer from sklearn library. In the term document matrix, each row represents term vectors across all the documents and columns represent document vectors across all the terms(vocabulary).
- Now, we calculate the weights for each term in the matrix across all the documents. The weights are calculated using tf-idf, the document the rare words get higher score.

Tf-idf formula

$$w_{i,j} = tf_{i,j} * \log(\frac{N}{df_i})$$

$$w_{i,j} = \text{tf-idf weight for token } i \text{ in document } j$$

$$tf_{i,} = \text{number of occurences of token } i \text{ in document } j$$

$$df_i = \text{number of documents that contain token } i$$

$$N = \text{total number of documents}$$

- The same pre-processing steps are applied on queries, then each query is converted into a vector of weights by transforming on the instances initialized for fitting documents.
- Then to measure the similarity between the document and query, cosine similarity is used as similarity metric. Cosine similarity measures the cosine angle between two vectors projected in a multidimensional space. To find relevant document to query, the similarity score between each document vector and query term vector is calculated by applying cosine similarity. Lower the angle between the vector, higher the document is relevant to query.

$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}},$$

Pool method (Task 3)

In the picture below **'A' represents IR system 1,** the columns in the 'A' are the 3 queries passed to IR1 and for each query top 10 documents are collected. Similarly, **'B' represents the IR system2** for which the same 3 queries are passed and top 10 documents for each query are collected.

	A			B		
	Effect of environmental factors on virus	Covid-19 air travel safety	Transmission of virus in community	Effect of environmental factors on virus	Covid-19 air travel safety	Transmission of virus in community
0	PMC2837245	PMC3577649	PMC2851561	PMC2837245	PMC3577649	PMC2851561
1	PMC3265445	PMC3032737	PMC2893203	PMC3265445	PMC3032737	PMC3541974
2	PMC3294595	PMC2950238	PMC3227662	PMC3294595	PMC2950238	PMC3324376
3	PMC2909313	PMC3314701	PMC3509329	PMC2909313	PMC3314701	PMC2893203
4	PMC2770169	PMC2939898	PMC3324376	PMC2821766	PMC1764036	PMC3227662
5	PMC2821766	PMC2813231	PMC3266138	PMC3585141	PMC2912811	PMC3509329
6	PMC3585141	PMC1764036	PMC3484124	PMC2770169	PMC2939898	PMC2204055
7	PMC2797517	PMC2781002	PMC2206439	PMC1351169	PMC2813231	PMC3057078
8	PMC3339311	PMC3223866	PMC3086881	PMC2797517	PMC2823611	PMC3222642
9	PMC1351169	PMC2796493	PMC1876810	PMC2981509	PMC2796493	PMC3266138

The table below shows the number of document that are did not match for each query when searched using IR1 and IR2.

Query	# different documents
Effect of environmental factors on virus	*3*
Covid-19 air travel safety	*4*
Transmission of virus in community	*8*

Relevance assessments (Task 4)

Every document in the pool is accessed whether it is relevant to the query the user has searched for or not. This is a binary relevance judgement; the document is considered relevant if it is present in the actual test collection else it is considered not relevant. All 60 document documents are accessed whether they are relevant or not.

The figure below shows the judgement for each document in the pool for one query searched using IR1, similarly all the documents six pools were judged.

```
Binary Relevance Judgement for each predicted document in the pool:

predicted_doc PMC2837245 is relevant
predicted_doc PMC3265445 is relevant
predicted_doc PMC3294595 is relevant
predicted_doc PMC2909313 is relevant
predicted_doc PMC2770169 is relevant
predicted_doc PMC2821766 is relevant
predicted_doc PMC3585141 is relevant
predicted_doc PMC2797517 is relevant
predicted_doc PMC3339311 is not-relevant
predicted_doc PMC1351169 is relevant
```

Query	ID of relevant documents
Effect of environmental factors on virus	'PMC1351169', 'PMC2770169', 'PMC2797517', 'PMC2821766', 'PMC2837245', 'PMC2909313', 'PMC2981509', 'PMC3265445', 'PMC3294595', 'PMC3585141'
Covid-19 air travel safety	'PMC2781002', 'PMC2813231', 'PMC2823611', 'PMC2939898', 'PMC2950238', 'PMC3032737', 'PMC3223866', 'PMC3314701', 'PMC3577649'
Transmission of virus in community	'PMC2851561', 'PMC2893203', 'PMC3222642', 'PMC3227662', 'PMC3266138', 'PMC3324376', 'PMC3484124'

Evaluation (Task 5)

Precision and recall are used to evaluate the IR systems constructed:

- **Precision formula:**

$$P@k = \frac{\#\ of\ retrieved\ documents\ that\ are\ relevant\ @k}{\#\ of\ retrieved\ documents\ at\ k}$$

- **Recall formula:**

$$R@k = \frac{\#\ of\ retrieved\ documents\ that\ are\ relevant\ @k}{total\ \#\ of\ relevant\ documents}$$

	System 1		System 2	
	P@5	**R@5**	**P@5**	**R@5**
Q1 (Effect of environmental factors on virus)	1.0	0.56	1.0	0.56
Q2 (Covid-19 air travel safety	1.0	0.71	0.8	0.57
Q3 (Transmission of virus in community)	0.67	0.8	0.8	0.67

Information Retrieval System 1

Query 1

k	Result	R@k	P@k
1	PMC2837245	0.11	1.0
2	PMC3265445	0.22	1.0
3	PMC3294595	0.33	1.0
4	PMC2909313	0.44	1.0
5	PMC2770169	0.56	1.0
6	PMC2821766	0.67	1.0
7	PMC3585141	0.78	1.0
8	PMC2797517	0.89	1.0
9	PMC3339311	0.89	0.89
10	PMC1351169	1.0	0.9

Query 2

k	Result	R@k	P@k
1	PMC3577649	0.14	1.0
2	PMC3032737	0.29	1.0
3	PMC2950238	0.43	1.0
4	PMC3314701	0.57	1.0
5	PMC2939898	0.71	1.0
6	PMC2813231	0.86	1.0
7	PMC1764036	0.86	0.86
8	PMC2781002	1.0	0.88
9	PMC3223866	1.0	0.78
10	PMC2796493	1.0	0.7

Query 3

k	Result	R@k	P@k
1	PMC2851561	0.17	1.0
2	PMC2893203	0.33	1.0
3	PMC3227662	0.5	1.0
4	PMC3509329	0.5	0.75
5	PMC3324376	0.67	0.8
6	PMC3266138	0.83	0.83
7	PMC3484124	1.0	0.86
8	PMC2206439	1.0	0.75
9	PMC3086881	1.0	0.67
10	PMC1876810	1.0	0.6

Information Retrieval System 2

Query 1

k	Result	R@k	P@k
1	PMC2837245	0.11	1.0
2	PMC3265445	0.22	1.0
3	PMC3294595	0.33	1.0
4	PMC2909313	0.44	1.0
5	PMC2821766	0.56	1.0
6	PMC3585141	0.67	1.0
7	PMC2770169	0.78	1.0
8	PMC1351169	0.89	1.0
9	PMC2797517	1.0	1.0
10	PMC2981509	1.0	0.9

Query 2

k	Result	R@k	P@k
1	PMC3577649	0.14	1.0
2	PMC3032737	0.29	1.0
3	PMC2950238	0.43	1.0
4	PMC3314701	0.57	1.0
5	PMC1764036	0.57	0.8
6	PMC2912811	0.57	0.67
7	PMC2939898	0.71	0.71
8	PMC2813231	0.86	0.75
9	PMC2823611	1.0	0.78
10	PMC2796493	1.0	0.7

Query 3

k	Result	R@k	P@k
1	PMC2851561	0.17	1.0
2	PMC3541974	0.17	0.5
3	PMC3324376	0.33	0.67
4	PMC2893203	0.5	0.75
5	PMC3227662	0.67	0.8
6	PMC3509329	0.67	0.67
7	PMC2204055	0.67	0.57
8	PMC3057078	0.67	0.5
9	PMC3222642	0.83	0.56
10	PMC3266138	1.0	0.6

Discussion:

Information Retrieval system 1 used lemmatization as a normalization technique to extract keywords fro m documents and queries. It performs morphological analysis of words as a result it produces correct wo rds that are present in dictionary. The size of vocabulary formed using lemmatization in IR1 is 12295 for 1000 documents.

```
------------------------------ This is IR System 1------------------------------

(1000, 12295)
['effect environmental factor virus', 'covid air travel safety', 'transmission virus community']
```

Information Retrieval system 2 uses stemming to extract keywords from documents and queries. Stemming works by chopping off the beginning or ending of the word, taking into consideration of a list of common suffixes , prefixes that can be found in inflected word. As a result, the word formed may not be actual words that can be found in English dictionary. The size of vocabulary formed using lemmatization in IR2 is 9810 for 1000 documents.

Stemming also reduced the vocabulary size considerably , also the figure below shows when the q ueries stemmed in IR2. This clearly shows stemming is not a good method to extract base forms.

```
------------------------------ This is IR System 2 --------------------

(1000, 9810)
['effect environment factor viru', 'covid air travel safeti', 'transmiss viru commun']
```

References:

1. https://www.datasciencecentral.com/profiles/blogs/information-retrieval-document-search-using-vector-space-model-in

2. https://honingds.com/blog/natural-language-processing-with-python/
3. https://stackoverflow.com/questions/56604737/i-have-two-formulas-for-calculating-cosine-similarity-whats-the-difference