Information Retrieval, Extraction and Integration

Profile-based Retrieval

POLITÉCNICA

Julio Martínez Bastida
Carlos Sánchez Velázquez
Rafael Sojo García

03/04/2023

Contents

1 Introduction

Information retrieval is a field of study that focuses on extracting relevant information from large datasets. With the growth of digital data, the need for efficient ways to manage and access information has become increasingly important. Information retrieval systems aim to provide users with easy access to relevant information by analyzing and indexing large collections of data whose majority of information is irrelevant to the user. Information retrieval can be used for various types of data, including text, images, audio, and video. However, in this particular project, we will focus on text-based information

The process of information retrieval typically involves two main steps: indexing and searching. In the indexing stage, the system analyzes and stores information about the documents in the collection. This information can include the document's text, metadata, and other relevant features. In the searching stage, users input queries, and the system retrieves and ranks documents based on their relevance to the query.

Information retrieval is essential for various domains, including healthcare, finance, education, and e-commerce, among others. In these domains, effective information retrieval systems can improve decision-making, save time and resources, and enhance user satisfaction. However, with the vast amounts of data being generated, there is a need for sophisticated techniques to ensure that only relevant information is retrieved.

The workflow of information retrieval involves three main steps: querying, indexing, and the matching/ranking. In the querying step, users input queries, which are used to search the index for relevant documents, these queries try to represent an existent need of information, while in the indexing step, the system analyzes and stores information about the documents in the collection, and indexes them in a way that queries and documents are represented in the same way. In the matching/ranking step, the system ranks the retrieved documents based on their relevance to the query.

Once the system has generated the search results, it can be useful to analyze and refine them further. This process can include examining the top search results to see if they are relevant to the user's needs and adjusting the system's ranking algorithm to improve the accuracy of future searches. This feedback loop is essential for ensuring that the system continues to provide accurate and relevant results over time.

There are several types of information retrieval systems, each with its strengths and weaknesses. Boolean models use logical operators such as "and," "or," and "not" to retrieve only documents that satisfy the conditions, these usually have the problem of being to strict. Vector models represent documents and queries as vectors in high-dimensional space, allowing the system to calculate similarity between the two. Probabilistic models use statistical methods to estimate the probability of a document being relevant to a query.

In this case, our source of interest is the creation of a Profile-based information retrieval system, a technique that leverages user's interests and preferences to generate more personalized search results. To achieve this, the user's profile is created by collecting relevant data such as their search history and browsing behavior. This profile is then used to filter and rank search results. By doing so, the system is able to provide search results that are more relevant to the user's needs, which can lead to a better user experience.

Profile-based information retrieval is particularly useful in situations where the user's query is ambiguous or broad, and their preferences are known. For example, in an e-commerce setting, a user searching for "shoes" may have different preferences based on factors such as brand, color, or price. By using the user's profile, the system can provide more personalized and relevant search results.

2 Problem Description

The Internet is, without a doubt, the greatest source of information in history. Therefore, it is necessary to use systems to filter the information obtained. Particularly in the field of research, the number of

papers and new developments is abysmal, so these systems are not only necessary but mandatory. Our use case is to try to generate an information retrieval system for COVID-19 papers, which has numerous lines of research.

We have selected a dataset from Kaggle [1], in which both documents and queries, and the respective relevant judgements can be found. This dataset is a reduction of TREC-COVID dataset, which is a collection of scientific articles and their metadata related to the covid pandemic. The dataset was created as part of the Text Retrieval Conference (TREC) COVID-19 Challenge, which aimed to facilitate research on information retrieval and text mining techniques for COVID-19 literature.

More specifically, in this case, the dataset belongs to the Round 3 of the TREC-COVID Challenge. This dataset is composed of several files, however for the development of this project only 3 of them were necessary, which are:

- **metadata.csv:** This dataset contains the information of 128492 medical papers, several of them related with COVID-19. The dataset contains main textual information about the paper content such as the title and the abstract, and other information less relevant to this problem, such as author, doi, license, journal or publishing date.

	cord_uid	sha	source_x	title	doi	pmcid	pubmed_id	license
0	ug7v899j	d1aafb70c066a2068b02786f8929fd9c900897fb	PMC	Clinical features of culture-proven Mycoplasma...	10.1186/1471-2334-1-6	PMC35282	11472636.0	no-cc
1	02tnwd4m	6b0567729c2143a66d737eb0a2f63f2dce2e5a7d	PMC	Nitric oxide: a pro-inflammatory mediator in l...	10.1186/rr14	PMC59543	11667967.0	no-cc
2	ejv2xln0	06ced00a5fc04215949aa72528f2eeaae1d58927	PMC	Surfactant protein-D and pulmonary host defense	10.1186/rr19	PMC59549	11667972.0	no-cc
3	2b73a28n	348055649b6b8cf2b9a376498df9bf41f7123605	PMC	Role of endothelin-1 in lung disease	10.1186/rr44	PMC59574	11686871.0	no-cc
4	9785vg6d	5f48792a5fa08bed9f56016f4981ae2ca6031b32	PMC	Gene expression in epithelial cells in respons...	10.1186/rr61	PMC59580	11686888.0	no-cc
...	...	...	...	...	...	...	...	...
128487	a11jyui1	NaN	Elsevier; PMC	Impact of BSE on the biotechnology industry — ...	10.1016/s0958-2118(04)00238-1	PMC7148827	NaN	els-covid
128488	9bma9y0q	44959e4505dcc59575f4989de8abf3fcc7e48415	Medline; PMC	Evolving Gene Targets and Technology in Influe...	10.1007/s40291-013-0040-9	PMC7100497	23686537.0	no-cc
128489	50xhia4r	8922380f61b6fe160ab0550a674816f1852a843d	Medline; PMC; WHO	How to train health personnel to protect thems...	10.3352/jeehp.2020.17.10	PMC7162995	32150796.0	cc-by
128490	97h1fz34	ea20a6b97f98ce4bdddfa6408da5ddf3dd06835e	Medline; PMC	Epidemic spreading in complex networks	10.1007/s11467-008-0027-x	PMC7111544	32288753.0	no-cc
128491	clmtwq4v	4c779b2a7682e10c421d6df6175226115a703997	Elsevier; Medline; PMC	The determinants of the 1999 and 2007 Chinese ...	10.1016/j.tourman.2009.10.003	PMC7127154	32287734.0	no-cc

Figure 1: Part of the documents DataFrame

- **topics-rnd3.csv:** This file contains information about several topics or queries of interest for Covid lines of research that were generated by the TREC-COVID organization. More specifically, it contains 40 different queries or topics specified in three different levels of detail, the query column just specifies the topic in less than 10 words, the question column specifies the topic as the question being tries to answer in that topic, and the narrative column specifies the topic in a detailed way. In our case, we decided to use just the title of the topic, because it is the one that fits more to the requirements of a profile-based information retrieval system, with a simple list of key words.

	topic-id	query	question	narrative
0	1	coronavirus origin	what is the origin of COVID-19	seeking range of information about the SARS-Co...
1	2	coronavirus response to weather changes	how does the coronavirus respond to changes in...	seeking range of information about the SARS-Co...
2	3	coronavirus immunity	will SARS-CoV2 infected people develop immunit...	seeking studies of immunity developed due to i...
3	4	how do people die from the coronavirus	what causes death from Covid-19?	Studies looking at mechanisms of death from Co...
4	5	animal models of COVID-19	what drugs have been active against SARS-CoV o...	Papers that describe the results of testing d...
5	6	coronavirus test rapid testing	what types of rapid testing for Covid-19 have ...	Looking for studies identifying ways to diagno...
6	7	serological tests for coronavirus	are there serological tests that detect antibo...	Looking for assays that measure immune respons...
7	8	coronavirus under reporting	how has lack of testing availability led to un...	Looking for studies answering questions of imp...
8	9	coronavirus in Canada	how has COVID-19 affected Canada	seeking data related to infections (confirm, s...
9	10	coronavirus social distancing impact	has social distancing had an impact on slowing...	seeking specific information on studies that h...
10	11	coronavirus hospital rationing	what are the guidelines for triaging patients ...	Seeking information on any guidelines for prio...

Figure 2: Topics DataFrame

- **qrels.csv:** This last file, is the one that contains the relevant judgments for several document and query pair needed to evaluate the Information Retrieval system. This file does not contain a relevant judgement for every document and query pair, in fact it only contains these judgements for around 600-700 documents for each query, wich would be a factor to be taken into account in the processing of the data. The relevant of the documents is specified in a 3 levels scale: 0 meaning not relevant, 1 meaning partially relevant, and 2 being completely relevant.

	topic-id	iteration	cord-id	judgement
0	1	0.5	010vptx3	2
1	1	1.0	02f0opkr	1
2	1	1.0	04ftw7k9	0
3	1	1.0	05qglt1f	0
4	1	1.0	0604jed8	0
...	...	...	...	...
20723	35	2.0	zp4oddrt	2
20724	35	2.0	zppc6p20	0
20725	35	2.0	zrobzakn	0
20726	35	2.0	zwjvvio0	0
20727	35	2.0	zzmfhr2s	0

Figure 3: Relevance judgments DataFrame

3 Pre-processing

In order to address the problem, we have two main parts to focus on during the pre-processing. One that is more focus on dealing with preliminary issues, and another one that focuses in the TF-IDF message preparation pipeline.

3.1 Dataset preparation

First of all, we read the csv files and generated a DataFrame for each of them using pandas library. As we already mentioned, the document which contains the information about the documents has many columns that are mainly irrelevant to solve our task, such as the date of publication, the journal or doi, and others that are not even used as identifiers against other files. So we decided to remove them with the exception of *cord_uid* which is used to identify the documents also in the relevance judgments dataset, *title* and *abstract* which contain information of the document content.

Afterwards we checked for the null values, the documents dataset contained a total of 26881 documents which did not have value for the abstract, and several of them that also had Unknown as a value. Since the

size of the dataset is really big we just decided to drop all those rows containing either null or Unknown values, which makes a dataset of 101491 and 3 columns.

Even after removing all rows with missing values, the dataset stills pretty big which makes the processing of every word at each document computationally expensive, so it was decided to reduce the dataset.

To prepare the sample, at first we first we thought of doing a random sample, however, the queries do not have relevance judgments for all documents, and if we do so the number of relevance judgments for each query available in the sample would be too short. Thus, we finally we decided to sample the number of queries and select only the first 10 of them. Afterwards, the documents dataset is filtered in a way that we select all the documents with relevance judgments for all those selected queries. This sample contains a total of 4089 selected documents.

However, after selecting the sample, we found that several documents were not written in English, which is something that our system do not contemplate because it is thought to be used for English vocabulary. Therefore, we decided to remove all texts not written in English from the sample dataset.

To do so, we used the python library *langdetect* a NLP library which has the function *detect*, that given a string indicates the language in which the string is written, based on the detection of several linguistic characteristic, such as the frequency of appearance of letters words and n-grams. Once we apply this function with the title of all documents, we generate a column that indicates the document language, and remove all those not written in English from both documents and judgments. The total number of documents in the sample that were not written in English was 123.

	cord_uid	title	abstract	lan
737	1rzcrkmt	Outdoor environments and human pathogens in air	Are pathogens in outdoor air a health issue at...	nl
1265	r8lv1zzx	Border Screening for SARS	With the rapid international spread of severe ...	da
5100	9c0zrp7p	Local risk perception enhances epidemic control	As infectious disease outbreaks emerge, public...	es
5724	i5jg7mgo	Schweres akutes respiratorisches Syndrom (SARS)	The sudden appearance of the severe acute resp...	de
5769	sqxmt699	Die Bedeutung von Coronaviren: Das Beispiel SARS	The emergence of the new SARS coronavirus has ...	de
...	...	...	...	...
125920	eeon1do2	Les professionnels de santé face à la pandémie...	RÉSUMÉ Objectifs: La pandémie de la maladie à ...	fr
126606	hi3fjne4	Coronaviren als Ursache respiratorischer Infek...	BACKGROUND: There are six human pathogenic cor...	de
127047	mhdmy8kq	DECLARACIÓN DE CONSENSO EN MEDICINA CRÍTICA PA...	Resumen El comportamiento de la infección por ...	pt
127500	mb1y5ort	La sécurité sanitaire mondiale à l'heure de Na...	SUMMMARY The International Health Regulations ...	fr
128106	up929iw0	COVID-19 y enfermedad cardiovascular	Resumen En diciembre de 2019 en Wuhan en la pr...	es

123 rows × 4 columns

Figure 4: Documents not written in english

Finally, after applying all these processing steps, the final documents DataFrame contains 3961 different documents. The final structure of this DataFrame is the next.

	cord_uid	title	abstract	lan
15	le0ogx1s	A new recruit for the army of the men of death	The army of the men of death, in John Bunyan's...	en
20	0qaoam29	A double epidemic model for the SARS propagation	BACKGROUND: An epidemic of a Severe Acute Resp...	en
41	fpj5urao	Moderate mutation rate in the SARS coronavirus...	BACKGROUND: The outbreak of severe acute respi...	en
45	8zwsi4nk	Date of origin of the SARS coronavirus strains	BACKGROUND: A new respiratory infectious epide...	en
70	jh9e85c0	Molecular mechanisms of severe acute respirato...	Severe acute respiratory syndrome (SARS) is a ...	en
...	...	...	...	...
128446	0iburamm	Contingency management strategies of the Nursi...	Abstract Objectives This article aims to summa...	en
128447	dxbvfwqu	Non-animal replacement methods for human vacci...	Abstract NICEATM and ICCVAM convened an intern...	en
128460	el4jgfoe	Diagnostic and therapeutic strategies of lung ...	With the increasing number of cases and wideni...	en
128467	z68j0c63	Long-term trends in seasonality of mortality i...	Background: Seasonal patterns of mortality hav...	en
128484	faec051u	Early epidemiological analysis of the coronavi...	BACKGROUND: As the outbreak of coronavirus dis...	en

Figure 5: Documents DataFrame after sample selection

3.2 Message preparation

For the message preparation we have used an specialized python package called *Natural Language Toolkit*, that contains specific natural language processing functions.

The cleaning process has the following steps:

1. *Removal of punctuation characters.* Here, we make use of the *String* built-in python package to remove all the punctuation characters from the text.

	rem_punc
15	The army of the men of death in John Bunyans m...
20	BACKGROUND An epidemic of a Severe Acute Respi...
41	BACKGROUND The outbreak of severe acute respir...
45	BACKGROUND A new respiratory infectious epidem...
70	Severe acute respiratory syndrome SARS is a ne...
...	...
128446	Abstract Objectives This article aims to summa...
128447	Abstract NICEATM and ICCVAM convened an intern...
128460	With the increasing number of cases and wideni...
128467	Background Seasonal patterns of mortality have...
128484	BACKGROUND As the outbreak of coronavirus dise...

Figure 6: Removal of punctuation characters

2. *Removal of no alphanumeric symbols.* After the punctuation removal, there are some no alphanumeric symbols left that could not be detected. These symbols can be treated by regular expressions.

3. *Tokenize.* This step refers to the split and transformation into lowercase of the words in the text. Future parts of the message pre-processing requires to have the text as a list of string to be treated separately. For some cases, it is interesting to split the text by sentences to get the *Part-of-Speech* tag, in our case, we have splitted by words.

	tokenize
15	[the, army, of, the, men, of, death, in, john,...
20	[background, an, epidemic, of, a, severe, acut...
41	[background, the, outbreak, of, severe, acute,...
45	[background, a, new, respiratory, infectious, ...
70	[severe, acute, respiratory, syndrome, sars, i...
...	...
128446	[abstract, objectives, this, article, aims, to...
128447	[abstract, niceatm, and, iccvam, convened, an,...
128460	[with, the, increasing, number, of, cases, and...
128467	[background, seasonal, patterns, of, mortality...
128484	[background, as, the, outbreak, of, coronaviru...

Figure 7: Tokenize

4. *Removal of small words.* Within the resulting list, there are some words, usually the shorter ones, that are not very informative. Therefore, we have removed words with less than 3 characters.

5. *Removal of stopwords.* Likewise, in the previous step the stopwords does not need to be consider since they are repeated very often without an important meaning for profile. These are just words that provide cohesion and grammatical structure to the text, and are not needed for the model.

	rem_stopwords
15	[army, death, john, bunyans, memorable, phrase...
20	[background, epidemic, severe, acute, respirat...
41	[background, outbreak, severe, acute, respirat...
45	[background, respiratory, infectious, epidemic...
70	[severe, acute, respiratory, syndrome, sars, i...
...	...
128446	[abstract, objectives, article, aims, summariz...
128447	[abstract, niceatm, iccvam, convened, internat...
128460	[increasing, number, cases, widening, geograph...
128467	[background, seasonal, patterns, mortality, id...
128484	[background, outbreak, coronavirus, disease, 2...

Figure 8: Removal of stopwords

6. *Lemmatization.* Finally, we transformed the list of words to its base dictionary form. We decided to apply this step instead of stemming although it is a similar process, since it does not limit itself to merely truncate the suffixes.

Then, the list of lemmatized words is returned back to the string format resulting in a clean text such as Figure 9.

	clean_text
15	army death john bunyan memorable phrase recrui...
20	background epidemic severe acute respiratory s...
41	background outbreak severe acute respiratory s...
45	background respiratory infectious epidemic sev...
70	severe acute respiratory syndrome sars infecti...
...	...
128446	abstract objective article aim summarize serie...
128447	abstract niceatm iccvam convened international...
128460	increasing number case widening geographical s...
128467	background seasonal pattern mortality identifi...
128484	background outbreak coronavirus disease 2019 c...

Figure 9: Clean text

3.3 Queries Synonyms

The previous steps are applied one after the other in sequential order for both documents and queries. However, for the queries we have made an extra treatment. This treatment consist on the expansion of the query including synonyms of each keywords.

A common problem in Information Retrieval is that the same concepts can be represented in the documents with different terms, which makes difficult to match certain documents that in fact are relevant

to the topic. This problem is typically solved by the use of *Controlled Vocabularies*, which are lists of synonyms or related terms usually called *Thesaurus*.

To do so, we have taken the not repeating lemmas synonyms returned by the Thesaurus provided by *wordnet* function from *nltk* package, see Figure 10. This way, we end up with a richer vocabulary to be consider during the cosine distance calculation in future sections.

Another possibility to do this that could led us to a wider vocabulary is extracting commonly related words by the use of statistics with big amounts of texts, but this is a way more complex process than just using an already defined *Thesaurus*

	clean_query	synonims
0	coronavirus origin	coronavirus origin beginning root rootage sour...
1	coronavirus response weather change	coronavirus response reaction answer reply rec...
2	coronavirus immunity	coronavirus immunity unsusceptibility resistan...
3	people coronavirus	people citizenry multitude masses mass coronav...
4	animal model covid19	animal beast brute creature fauna carnal flesh...
5	coronavirus test rapid testing	coronavirus test trial tryout examination exam...
6	serological test coronavirus	serological serologic test trial tryout examin...
7	coronavirus reporting	coronavirus reporting coverage reportage repor...
8	coronavirus canada	coronavirus canada Canada
9	coronavirus social distancing impact	coronavirus social sociable mixer societal dis...

Figure 10: Clean queries and their corresponding synonyms

3.4 TF-IDF preparation and generation

Once all the text has been properly processed for both queries and documents, we have to encode the information in a way that it could be processed. Thus, we must transform our data from strings to numeric vectors maintaining as possible the meaning of the text, so that similar documents could be place near to their respective queries in the vectorial space. To do so, we have use the *TfidfVectorizer* function from *Scikit-Learn*, and a bag-of-words containing the vocabulary from both documents and queries.

In order to encode the information, the structure is not needed, instead, we use have to weight each word according to the frequency of every word for each document. However, we must also consider the document frequency and the rareness of the word, which is also a very informative metric. These two metrics, Term Frequency and Inverse Document Frequency, are multiplied together leading to the TF-IDF score.

For the TF-IDF vector of the documents, we have considered the cleaned abstract, whereas for the queries, the cleaned query with its synonyms. As it was mentioned, we have included the common vocabulary, or bag-of-words. Thus, we end with two matrices of 3963 documents and 10 queries, with 28829 words each.

4 Evaluation

4.1 Model Evaluation

For the model evaluation we need two main things:

- **Cosine similarity matrix**. Once we obtain the TF-IDF vectors for both queries and documents as explained previously, the cosine similarity between them is calculated. For this step we have use the *cosine_similarity* function from *Scikit-Learn*. See Figure 11. The cosine similarity is used to compare the angles generated between the vectors, allowing us to obtain the relevance of each document to the query.

	doc_id	coronavirus origin	coronavirus response to weather changes	coronavirus immunity	how do people die from the coronavirus	animal models of COVID-19	coronavirus test rapid testing	serological tests for coronavirus	coronavirus under reporting	coronavirus in Canada	coronavirus social distancing impact
0	le0ogx1s	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
1	0qaoam29	0.002485	0.017757	0.005190	0.004661	0.053044	0.003079	0.003217	0.048772	0.005190	0.002720
2	fpj5urao	0.002099	0.001783	0.004383	0.003936	0.000000	0.018702	0.019546	0.003342	0.004383	0.002297
3	8zwsi4nk	0.030182	0.017500	0.024102	0.021643	0.005480	0.014296	0.014941	0.018375	0.024102	0.012630
4	jh9e85c0	0.003208	0.002725	0.006700	0.006016	0.000000	0.003974	0.004153	0.005108	0.006700	0.003511
...	...	...	...	...	...	...	...	...	...	...	...
3966	0iburamm	0.002914	0.002476	0.006086	0.005465	0.006311	0.025968	0.027139	0.034593	0.006086	0.003189
3967	dxbvfwqu	0.000000	0.000000	0.000000	0.000000	0.023236	0.089928	0.040235	0.032030	0.000000	0.000000
3968	el4jgfoe	0.002932	0.002491	0.006123	0.005498	0.025394	0.003632	0.003796	0.004668	0.006123	0.003209
3969	z68j0c63	0.000000	0.071988	0.000000	0.000000	0.040793	0.008691	0.009083	0.000000	0.000000	0.000000
3970	faec051u	0.019425	0.001375	0.003380	0.003035	0.035042	0.002005	0.002095	0.099909	0.003380	0.044949

Figure 11: Cosine similarity Matrix

- **Relevance Judgements Matrix**. As it was introduced, each document comes with its corresponding relevance judgement from 0 to 2 for a given query, however, not all the documents are labeled for all the queries. For that reason, we made a function to label these documents as -1 in the matrix construction. These negatively labeled documents, will be removed during the evaluation of each query. On the other hand, every value higher than 0, will be consider as relevant. See Figure 12

	cord_uid	coronavirus origin	coronavirus response to weather changes	coronavirus immunity	how do people die from the coronavirus	animal models of COVID-19	coronavirus test rapid testing	serological tests for coronavirus	coronavirus under reporting	coronavirus in Canada	coronavirus social distancing impact
15	le0ogx1s	0	-1	-1	-1	-1	-1	-1	-1	-1	-1
20	0qaoam29	0	0	1	-1	-1	-1	-1	-1	-1	-1
41	fpj5urao	-1	-1	-1	-1	-1	0	-1	0	-1	-1
45	8zwsi4nk	-1	-1	-1	-1	-1	-1	-1	0	0	-1
70	jh9e85c0	1	0	1	-1	0	-1	-1	0	0	-1
...	...	...	...	...	...	...	...	...	...	...	...
128446	0iburamm	-1	-1	-1	-1	-1	0	-1	-1	-1	-1
128447	dxbvfwqu	-1	-1	-1	-1	-1	-1	0	-1	-1	-1
128460	el4jgfoe	-1	-1	-1	-1	-1	-1	-1	-1	0	-1
128467	z68j0c63	-1	-1	-1	0	-1	-1	-1	-1	-1	-1
128484	faec051u	-1	-1	-1	-1	-1	-1	0	1	0	-1

Figure 12: Relevance Judgements Matrix

Once we have the matrices, we select each query with its labeled documents and sort them according to the cosine similarity.

4.1.1 Results

For the quality study of the model, we have made use of the functions from the script evaltools.py that was left in moodle, which has been slightly modified to suit our needs. The ROC and Precision-Recall curves have been used, as well as the R-precision, the Mean Average Precision (MAP) and the Area Under the ROC-Curve (AUC).

Query	AP	MAP	AUC
coronavirus origin	**0.74**	**0.76**	**0.6**
coronavirus response to weather changes	0.58	0.6	0.5
coronavirus inmunuity	0.65	0.71	0.55
how do people die from the coronavirus	0.58	0.58	0.5
animal models of COVID-19	0.66	0.66	0.46
coronavirus test rapid testing	**0.82**	**0.87**	**0.66**
serological tests for coronavirus	**0.59**	**0.55**	**0.65**
coronavirus under reporting	0.61	0.63	0.49
coronavirus in Canada	0.56	0.65	0.53
coronavirus social distancing impact	**0.79**	**0.91**	**0.70**

Table 1: model metrics for each query.

As we can see in the table above, there are some acceptable results, such as *coronavirus origin*, *coronavirus test rapid testing*, *serological tests for coronavirus* and *coronavirus social distancing impact*, whose AUC-scores are at least 0.6. However, other queries have given quite bad results, such as the *animal models of COVID-19* query, whose AUC is less than 0.5. This may be due to the fact that some of the queries words are not very frequent in the dataset.

Finally, we can see in the following graphs the ROC curves and the precision-recall curves. In the former Figure 13 we can see how some of the curves are convex, which makes the area under their curve less than 0.5, i.e., they predict worse than a random model. On the other hand, the precision-recall plots Figure 14 show large flat areas, which allows us to increase the recall a lot without affecting the precision of the model.

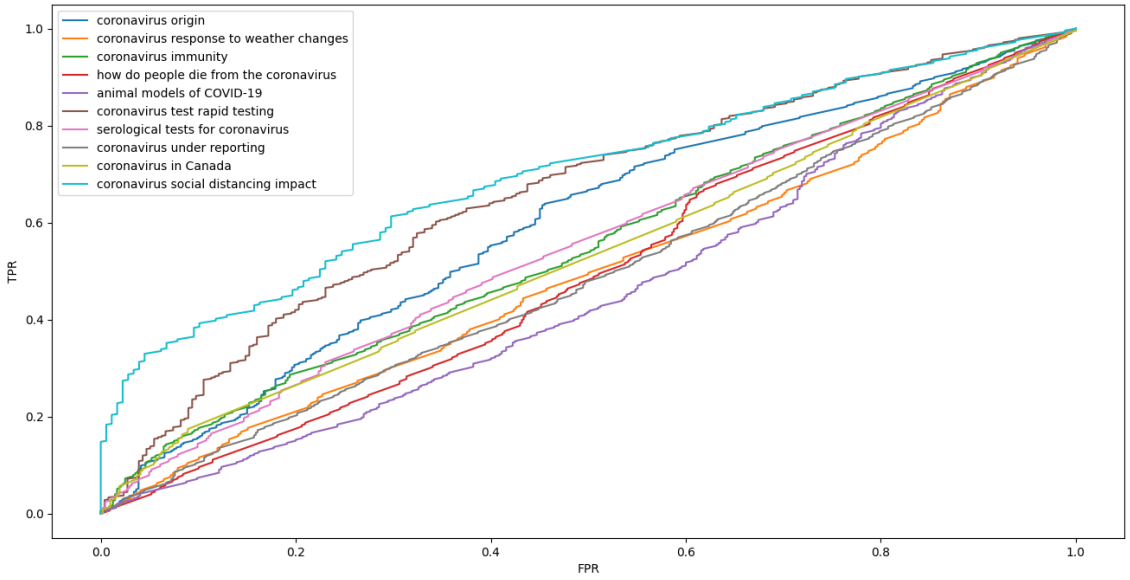

Figure 13: ROC Curves for each query

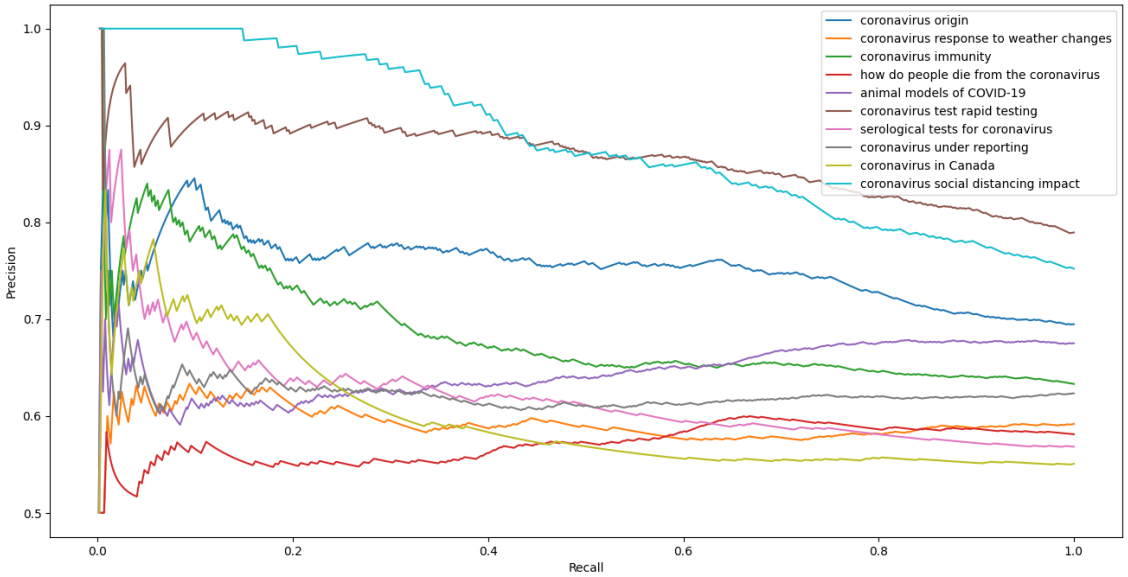

Figure 14: Precision-Recall Curves for each query

4.2 Comparison. Other use cases

Since we have not found any other group that did the evaluation in a similar way to ours, we have tried to study the behavior of our model in another context. For this purpose, we have used another dataset from Kaggle [2] that deals with movies and their synopses. This dataset contains a list of 36000 movies from all over the world for which we have the following information:

- Release Year.
- Title.
- Origin/Ethnicity.
- Director.
- Cast.
- Genre.
- Wiki Page.
- Plot.

Due to the size of the dataset, we had to do some preprocessing to reduce both rows and columns:

1. Remove all non-English speaking films.
2. Remove all movies that were not from one of the 8 most popular genres in the dataset: drama, comedy, horror, western, thriller, action, adventure and crime.
3. Reduce to approximately 1000 the drama and comedy films to balance the categories between 500 and 1000.
4. Concatenation of title, director, cast, genre and plot.

After this pre-processing, we proceeded in the same way as we did with the previous COVID-19 case to generate the TF-IDF vectors and the calculation of the cosine similarity matrix. With respect to the relevance judgements, we have defined the matrix in a slightly different way since we did not have them beforehand. Here, the relevancy is obtained from the Genre of the film, where each query refers to different genres. That said, we have defined the following user's queries:

- *drama, thriller*
- *horror, adventure*
- *comedy, crime*
- *western, action*
- *adventure, crime*

4.2.1 Results

The following table shows the results of the model applied to the film use case. In this case, it can be observed that the area under the ROC curve is much better in general terms than in the COVID-19 documents use case. This is probably due to the complexity of the context, for example, some synonyms in everyday life do not have to be synonyms when we talk with formalisms. This leads us to think that the model is very simple for the complexity of the previous use case. In addition, we can also observe a substantial improvement in both AP and MAP.

Query	**AP**	**MAP**	**AUC**
drama, thriller	0.73	0.92	0.71
horror, adventure	0.70	0.86	0.67
comedy, crime	0.69	0.92	0.67
western, action	0.68	0.92	0.67
adventure, crime	0.64	0.81	0.62

Finally, figures 15 and 16 show the ROC and Precision-Recall plots respectively.

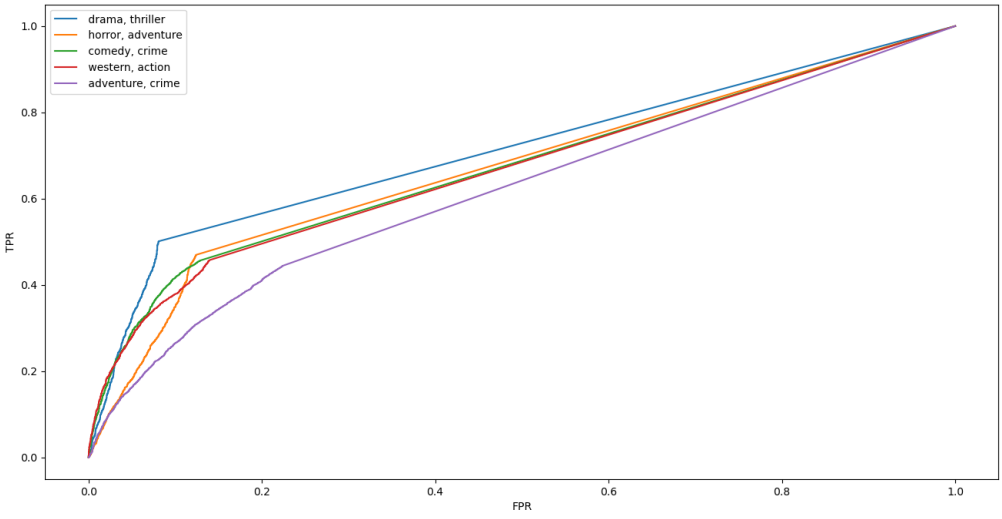

Figure 15: ROC curves for the queries in the film use case

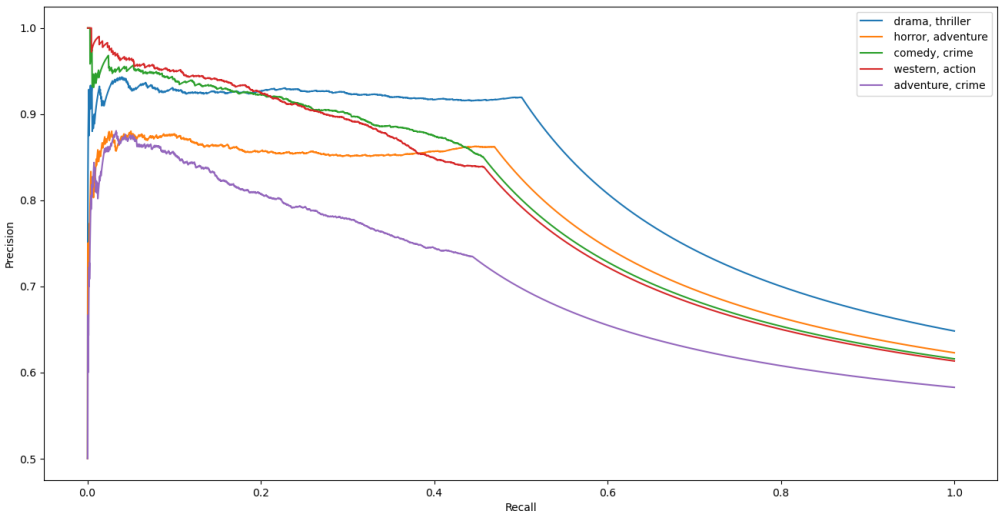

Figure 16: Precision-Recall curves for the queries in the film use case

5 Upgrades and future work

To wrap up the assignment, we have seen that the model does not work very well for some queries in the COVID-19 document set, although it does for the movie queries. As we said before, this is most likely due to the difference in context complexity of the two use cases, so we have considered the following improvements that should be made so that the model could fit this work.

- To use n-grams apart from single words so that queries and documents could be understood in more detail.
- In this paper, we have expanded the queries with the synonyms of the words that appear in it. However, after some more thought on this approach, we have considered that it is likely that unifying the words of the different documents by a particular synonym could improve the results.
- Expand the vocabulary with words from the same semantic field, thus generating more relevant information and possibly better results.

In conclusion, we believe that using only the TF-IDF matrix may be a simplistic approach to a problem as challenging as information retrieval.

References

[1] "Kaggle." https://www.kaggle.com/competitions/trec-covid-information-retrieval/data?select=qrels.csv, 2023.

[2] "Wikipedia Movie Plots — kaggle.com." https://www.kaggle.com/datasets/jrobischon/wikipedia-movie-plots. [Accessed 08-Apr-2023].