Report on CE706 - Information Retrieval

Assignment 2: Elasticsearch & Evaluation

Professor: Dr. Udo Kruschwitz

Student Names':

Atiwat Onsuwan 1802514

Yaowei Lyu 1802697

1. Introduction

First, after we investigated an instruction of this assignment, we decided to build the search engine system using Python that users can search for the information from the given dataset (*Signal Media One Million News Articles Dataset*) also we made an evaluation for a searching case. The two main objectives for this system are:

1. Search for relevant documents – *We have created a Python system that allows users to search for information.*

2. Evaluation searching method – *Every searching the system will calculate Precision and Recall at particular timestamp and also the average.*

Our system is divided into two sections as follow:

- **Upload data set** - (folder) Search_Engine/upload.py
  The data set can be downloaded at:
  https://research.signal-ai.com/newsir16/signal-dataset.html
- **Search and Evaluate** - (folder) Search_Engine/search.py
  https://www.elastic.co/downloads/elasticsearch

To run our code: 1.Install json_lines (Python library)

2.Install json (Python library)

3.Install Elasticsearch (Python library)

4. Put sample-1M.jsonl in Search_Engine folder

(same directory of code)

5. Run Elasticsearch batch file

2. Description of Implementation

2.1 Indexing – For the first step we uploaded the small amount of data set which are 3,000 documents for our experiment and assigned name of data set as 'news_article' and decoded the id index of the documents to normal numerical using **Python** to connect to **Elasticsearch server**. (Figure 1)

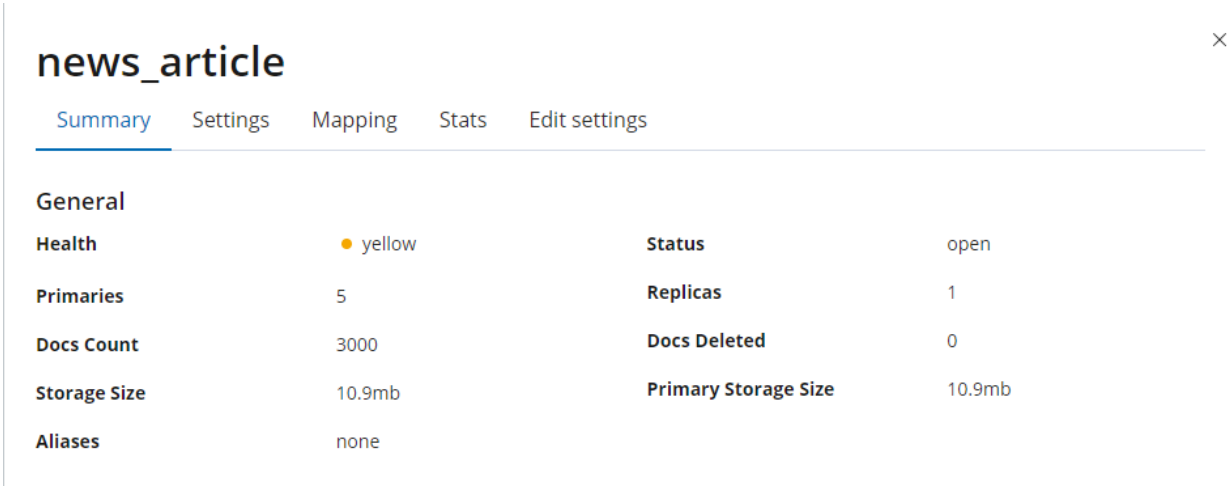

Figure 1: Uploaded document (Kibana GUI)

Next, after we have uploaded the dataset to Elasticsearch, the index mapping is defined by default (Figure 2), and we also explored the indexes and learned that all documents have these indexes in common as follows:

- **content** – contains content of the article
- **id** – id of the article after decoding to number format

- **media-type** – the type of the media either News or Blog
- **published** – the publication date and time
- **source** – tells that where did a particular article from
- **title**- the title name of the article

Figure 2: Index Mapping (Kibana GUI)

2.2 Searching – In this section after we have explored the index mapping so we could make use of it by using it for searching the relevant documents. Also, we have tried 4 examples of searching on vary field indexes using Kibana GUI as follows:

*Letter in **Blue** are index field*

*Letter in **Green** are keyword*

- First, we searched for documents that media-type is news and Google in the title which published between September 1st, 2010 and December 1st, 2015. As you can see from figure 3 below we found 8 relevant documents from 3,000 documents.

Figure 3: Search Result 1

- Second, we searched for documents that media-type is blog and sports in the content which published between January 1st, 2015 and December 31st, 2015. As you can see from figure 4 below we found 26 relevant documents form 3,000 documents.

Figure 4: Search Result 2

- Third, we tried with more specific searching information which media-type is news, sports car in the content, and from Batley source which published between January 1st, 2015 and December 31st, 2015. As you can see from figure 5 below we found only 1 relevant document from 3,000 documents.

Figure 5: Search Result 3

- Last, we tried with all indexes searching field which media-type is news, Lincoln in the title, sports car in the content, and from car source which published between January 1st, 2015 and December 31st, 2015. As you can see from the figure 6 below, again, we found only 1 relevant document from 3,000 documents even though the keyword sports car is not in content index field, but the operator we use was able to consider this article was relevant.

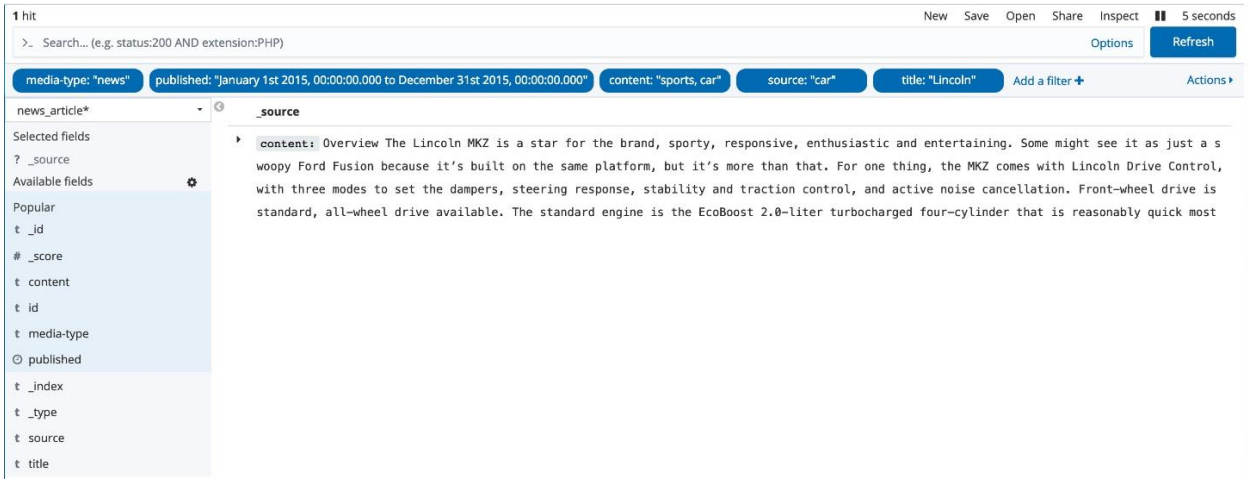

Figure 6: Search Result 4

2.3 Building a Test Collection – In this section, first, we have defined 10 possible events that a particular user might search for the information they need as well as the queries with different search setting, and the expected results in table 1 below.

Test Collection		
Events	**Queries**	**Expected Results**
A user might be looking for documents that contain a specific keyword in a title in one particular media-type. Setting: Bool search using AND operator and prefix match.	{ "query": { "bool": { "must": [{ "match": { "media-type": "news" } }, { "match_phrase_prefix": { "content": "google" } }] } } }	Documents which both news and title that contain google keyword inside.
Some user might be looking for information in a specific media-type that published within a particular range of time. Setting: Bool search using AND operator, prefix keyword match, and range of date.	{ "query": { "bool": { "must": [{ "match_phrase_prefix": { "title": } }, { "range": { "published": { "gte": "2015/01/20"	Documents type news during 2015/01/20 and 2015/10/20.

	, "lte": "2015/10/20" , "format": "yyyy/MM/dd||yyyy" } } }] } } }	

Users might be searching for information in a specific media-type that contains their keyword in content that published during a period.
 	
Setting: Bool search using AND operator, match keywords only, and range of date.
	{
	"query": {
	"bool": {
	"must": [
	{
	"match": {
	"media-type":
	"news"
	}
	},
	{
	"match": {
	"content":
	"facebook"
	}
	},
	{
	"range": {
	"published": {
	"gte":
	"2015/01/20",
	"lte":
	"2015/10/20",
	"format":
	"yyyy/MM/dd||yyyy"
	}
		Documents type news which include Facebook in content and published during 2015/01/20 and 2015/10/20.

	} }] } } }	

Some users will search for articles that content contains two different keywords while it published date during a specific period.
 	
Setting: Bool search using AND operator, match + prefix match with different keywords in the same field in the range of date.
	{
	"query": {
	"bool": {
	"must": [
	{
	"match": {
	"content": "news"
	}
	},
	{
	"match_phrase_prefix": {
	"content": "google"
	}
	},
	{
	"range": {
	"published": {
	"gte": "2015/01/13" ,
	"lte": "2015/09/20" ,
	"format": "yyyy/MM/dd||yyyy"
	}
	}
	}
	]
	}
	}
	}
		Documents that content contains both sports and cars published during 2015/01/13 and 2015/09/20.

It is possible users will find the document that content contains the keyword from a specific source which title matches their keywords.
 	
Setting: Bool search using AND operator, match + prefix match with different keywords in the diverse field.
	{
	"query": {
	"bool": {
	"must": [
	{
	"match": {
	"content": "sports cars"
	}
	},
	{
	"match": {
	"title": "SVR"
	}
	},
	{
	"match_phrase_prefix": {
	"source": "car"
	}
	}
	]
	}
	}
	}
		Documents which its content includes sports cars and its title includes SVR from source cars.

Some users might find some documents that its content and title contains a specific keyword but give the weight of importance in title field more than content.
 	
Setting: Boosting Search
	{
	"query": {
	"multi_match" : {
	"query" : "sports car",
	"fields": ["title^3", "content"]
	}
	
	}
	}
		Documents that either its title or content contains sports car keyword while the documents with title sports car will come first.

It is possible that the user will be seeking for some documents that	{
	"query": {
	"wildcard" : {
	"source" : "a*"
		All documents which its source begins with A or a letter.

its source begin with a specific letter or word Setting: Wildcard Search (similar to Regx but less complex)	} } }	

The users might not be sure with the spelling of the keyword they are using to find the documents that its content and title contains a specific keyword.
 	
Setting: Fuzzy Search with Multi match.
	{
	"query": {
	"multi_match" : {
	"query" : "casr",
	"fields": ["title", "content"],
	"fuzziness": "AUTO"
	}
	}
	}
		Documents which its content or title contains word casr, cars ,csar, casr, acsr, or acrs.

The user could find documents with the same keyword for different filed from content or title.
 	
Setting: Bool Search using the OR operator.
	{
	"query": {
	"bool": {
	"should": [
	{
	"match": {
	"content": "car"
	}
	},
	{
	"match": {
	"source": "car"
	}
	}
	]
	}
	}
	}
		Documents with the car keyword for both content and title.

Some user might be interested in searching for documents that its title match the exact their keyword
 	
Setting: Term Search
	{
	"query": {
	"term" : {
	"title": "kill"
	}
	}
	}
		Documents that its title has exact kill keyword in there.

Table 1: Events and Queries

2.4 Evaluation – In this section, after we have defined test collections and the events in the previous section before, so we came up with the evaluation of some examples searching setting using Python code we have developed to calculate *Precision* and *Recall* and the documents score. In this task, Python will be a bridge to connect to Elasticsearch server as well as our GUI.

First, We have tried to search for media-type = news the relevant documents in the Elasticsearch database of 3,000 documents in there by using keyword computer science to find in content which published in range 2014/01/01 to 2016/01/01. With this search case (figure 7)., we found 4 relevant documents and calculate the *precision* and *recall* (figure 8). Base on 4 relevant documents we found, we decided to focus on document number 1478 because this case it has the maximum search score and its search score benchmark for the next search case (figure 10). For this time see the precision and recall at K1478 at figure 9 below and search score for this document is **35.598**.

```
Media-type:news
Content keyword:computer science
Published start date (yyyy/mm/dd):2014/01/01
Published end date (yyyy/mm/dd):2016/01/01
```

Figure 7: Search Case 1

```
Precision AVG. => 0.001386928145491996
Recall AVG => 0.0008388888877777735

Number of document in database: 3000
Found : 4
```

Figure 8: Search Result 1

```
@K 1478 | P= 0.00270636 | R= 0.00133333 Found document id: 1478
```

Figure 9: P&R @K 1478

```
Document ID: 1478
Search score: 35.598343
Media type: News
Title: Microsoft expands global YouthSpark initiative to focus on computer science
From source: MoneyShow.com
Published: 2015-09-17T01:00:00Z
Content: Microsoft Corp. announced on Wednesday a new commitment of $70 million in community investments over the next three years to increase access to computer science educati

Invests $70 million in community programs to increase access to computer science education for all youth and build greater diversity into the tech talent pipeline.

SAN FRANCISCO , Sept. 16, 2015 /PRNewswire/ -- Microsoft Corp. announced on Wednesday a new commitment of $70 million in community investments over the next three years to incre

"If we are going to solve tomorrow's global challenges, we must come together today to inspire young people everywhere with the promise of technology," said Microsoft CEO Satya

Over the next three years, Microsoft will deliver on this commitment through cash grants and nonprofit partnerships as well as unique program and content offerings to increase a

Nadella reinforced the company's commitment to computer science education today during the annual Dreamforce conference hosted by Salesforce where he called upon thousands of te

"Computer science is a foundational subject – like algebra, chemistry or physics – for learning how the world works, yet it's offered in less than 25 percent of American high sc

There are three additional key elements of Microsoft's global commitment to increasing access for all youth to the full range of computing skills, from digital literacy to compu
Global philanthropic investments with nonprofits in 80 countries, including the Center for Digital Inclusion in Latin America , Silatech in the Middle East and Africa , CoderDoj
Since 2012, Microsoft YouthSpark has created new opportunities for more than 300 million youth around the world, offering technology skills training and connections to employmen

More information about YouthSpark and access to tools and resources can be found at http://YouthSparkHub.com and http://imagine.microsoft.com .

Those wanting more information on the TEALS program and to learn more about how they can get involved should visit http://TEALSK12.org .

Microsoft (Nasdaq "MSFT" @microsoft) is the leading platform and productivity company for the mobile-first, cloud-first world, and its mission is to empower every person and eve

Logo - http://photos.prnewswire.com/prnh/20000822/MSFTLOGO

To view the original version on PR Newswire, visit: http://www.prnewswire.com/news-releases/microsoft-expands-global-youthspark-initiative-to-focus-on-computer-science-300144592

SOURCE  Microsoft Corp.
```

Figure 10: Measurement 1

Next, we tried with the title search in the same range of publication time (figure 11); we used the keyword as computer science as the first search case to get the same document number 1478 as a result (its title has computer science) and to see the difference of search score.

```
Title-name (Pre-fix):computer science
Published start date (yyyy/mm/dd):2014/01/01
Published end date (yyyy/mm/dd):2016/01/01
```

Figure 11: Search Case 2

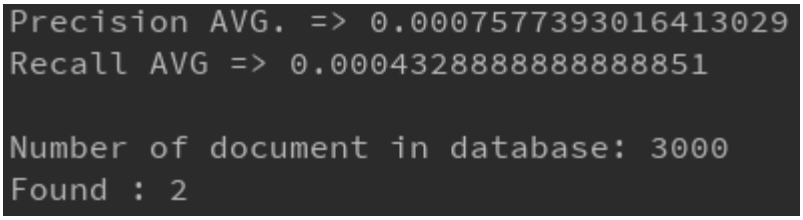

Figure 12: Search Result 2

As you can see from the search result above (figure 12) we found 2 relevant documents because the title contains less text to match our keyword, but we have lower precision while got higher recall at K= 1478 (figure 13) because document number 1478 did not get the maximum score as the first search case(figure 14), we got the document number 678 that was the most relevant instead(figure 15).

The definition is when it found 2 same relevant which has the equal number of keyword detected from difference document it will consider the first document it finds as the most relevant. The search score for document 1478 this time is **11.647** means that when we reduced the field for searching from 3 to 2 which we took media-type out from our query it made our system get more work it had to go through all document in that publication time either media-type is News or Blog.

Figure 13: P&R @K 1478

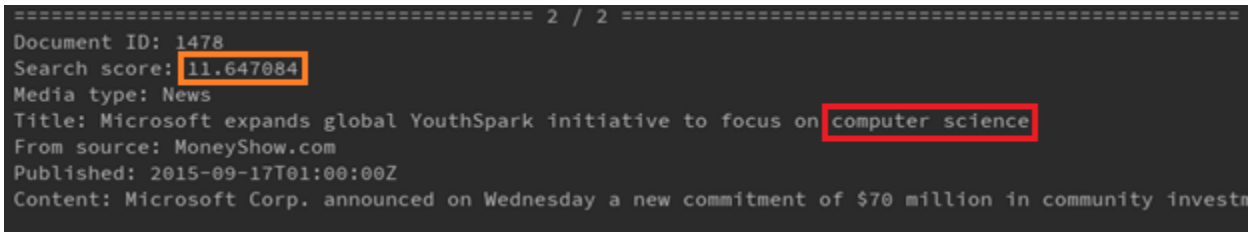

Figure 14: Measurement 2

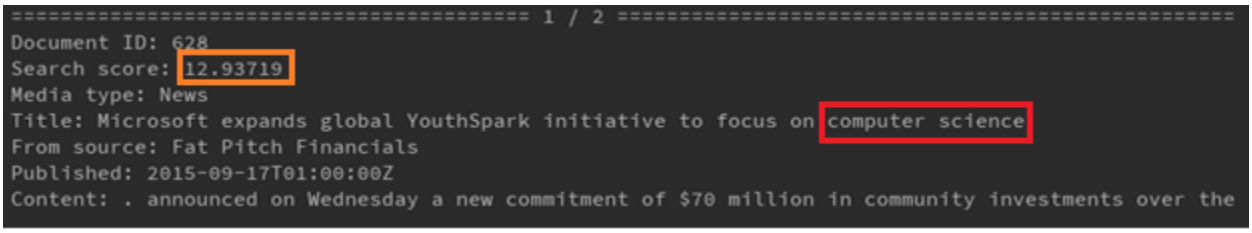

Figure 15: Measurement 3

Last, We tested another search case (figure 16) with only the 1 field index by searching MoneyShow on the source field just to get the document number 1478 as before. After we searched using this search case we found 12 relevant documents with the average of precision and recall (Figure 17).

Figure 16: Search Case 3

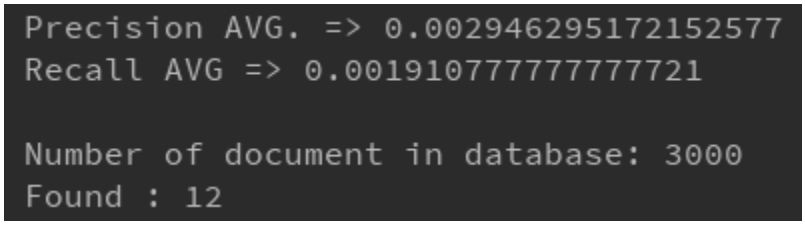

Figure 17: Search Result 3

Figure 18: P&R @K 1478

Now, let's look at the precision and recall in figure 18 above, at K 1478 you might notice that P&R and this search case are the highest among 3 search cases we have tested because we have 12 relevant documents and there are some relevant documents before this attempt, so the average P&R at this point is high.

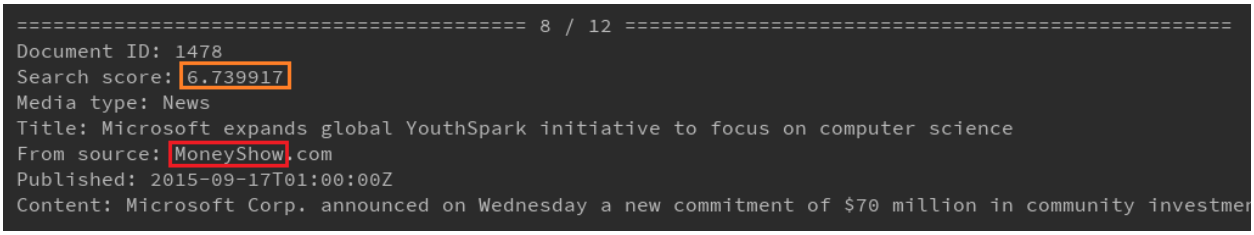

Figure 19: Measurement 3

When we check at the search score of this search case (Figure 19), we will see that it has the lowest rating at **6.739** because we decreased the index field searching to 1 field,

so it had to go through every document in the database and check if the particular document is from source (prefix) MoneyShow or not.

Field searching	Number of the document found	Ranking score of a focus document (1478)	Average Precision	Average Recall
Media-type Content Keyword Range of Publication	4 Relevant Documents	35.598	0.00138	0.00083
Title Keyword Range of Publication	2 relevant documents	11.647	0.000757	0.000432
Source	12 relevant documents	6.739	0.002946	0.001910

Table 2: Comparison of Searching Case

In conclusion of this task, Table 2 above shows the comparison of 3 searching case we have tried. When we looked into the comparison closely, we noticed that the more relevant document we found, the higher the average of precision and recall. On the other hand, the more searching index field (specific search), the higher the score we get. Moreover, we got this type of evaluation motivation from the class and lab exercise we assume that if we tried with another search setting such as *Boosting* we would see the difference of ranking score.

2.5 Complete search engine – As we have done the previous task so far, it led us to the complete search engine which 7 selection menu in the system GUI shows in figure 20 below.

```
Selection searching for document menu base on...:
1.'Media-type' and 'Content' keyword in range of 'Published' date
2.'Title' name in range of 'Published' date
3.Specific 'Source' of all time
4.Specific keyword in 'Content' of all time
5.Exact keyword in 'Title'
6.Begin letter of 'Source'
7.Keyword for 'Title' or 'Content' but title will be more important than content

Enter Choice:
```

Figure 20: Selection Menu

Now, we will show how our system works by using menu 2 as an example to search for the title contains prefix keyword data in the publication date range 2015/01/01 to 2016/01/01. The result of searching show in figure 21 below.

```
Selection searching for document menu base on...:
1.'Media-type' and 'Content' keyword in range of 'Published' date
2.'Title' name in range of 'Published' date
3.Specific 'Source' of all time
4.Specific keyword in 'Content' of all time
5.Exact keyword in 'Title'
6.Begin letter of 'Source'
7.Keyword for 'Title' or 'Content' but title will be more important than content

Enter Choice:2
Title-name (Pre-fix):data
Published start date (yyyy/mm/dd):2014/01/01
Published end date (yyyy/mm/dd):2016/01/01

Number of document in database: 3000
Found : 20

========================================= 1 / 20 ================================================
Document ID: 2217
Search score: 6.7316504
Media type: News
Title: US stocks dip on weak China data
From source: Yahoo! UK and Ireland
Published: 2015-09-23T15:49:31Z
Content: US stocks dipped early Wednesday following disappointing Chinese factory data and signs of s

About 40 minutes into trade, the Dow Jones Industrial Average was at 16,297.16, down 33.31 points (0.

The broad-based S&P 500 slipped 1.51 (0.08 percent) to 1,941.23, while the tech-rich Nasdaq Composite

China's Purchasing Managers' Index (PMI) for factory activity in September fell to its lowest level s

A PMI reading for the eurozone dipped to 53.9 points in September from 54.3 points in August. Despite

Stocks fell sharply Tuesday on global growth fears. Analysts expected light trading volumes on Wednes

Software   (Xetra: 330400 - news ) and cloud computing company Citrix Systems  (NasdaqGS: CTXS - news

Heron Therapeutics  (NasdaqCM: HRTX - news )  surged 19.8 after releasing positive clinical results f
```

Figure 21: Example Search Result

Also, our system shows the evaluation of particular searching as figure 22 and 23 below.

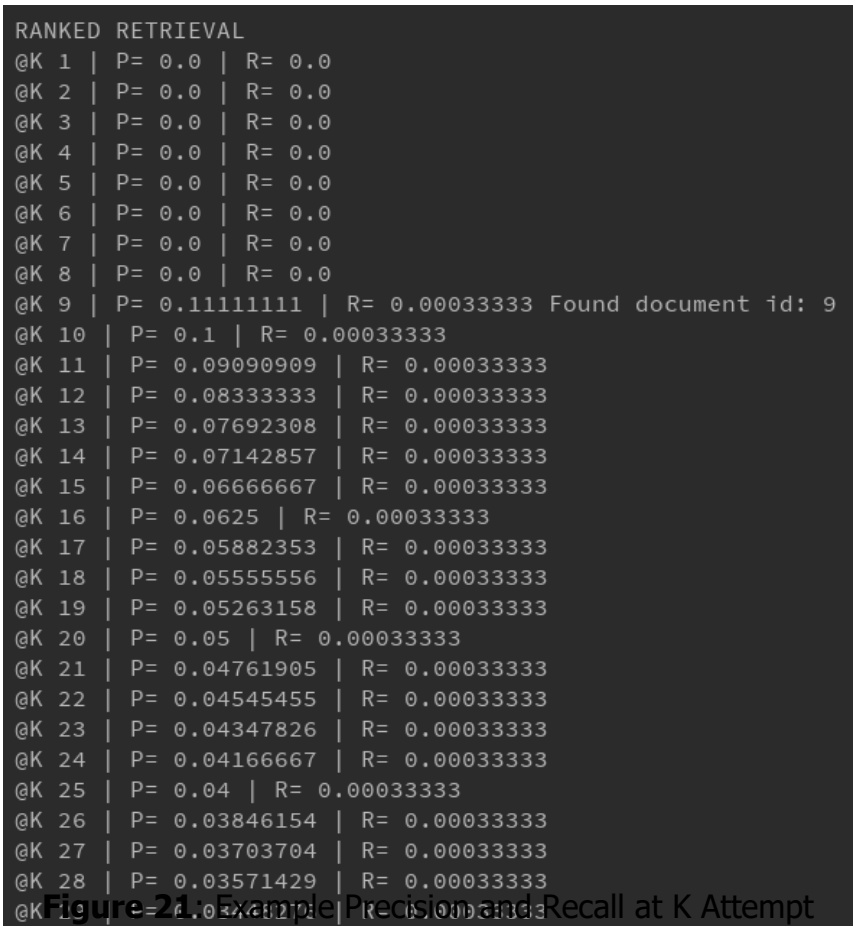

Figure 21: Example Precision and Recall at K Attempt

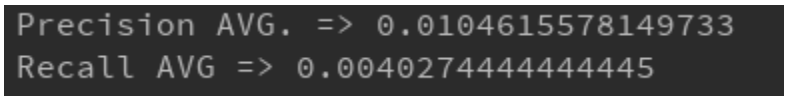

Figure 22: Example Average Precision and Recall

3. Discussion of Functionality Implementation and Possible Improvements

In summary, our system work on Python as a bridge between the Elasticsearch server, we developed the program that allows the users to upload the set of document to the Elasticsearch server, as we mentioned before in our case we uploaded 3,000 documents to the server for study purpose. Then we tried Kibana GUI to search for relevant documents in vary fields after we started to understand how searching query work, we created our search engine system using Python connects to the server. We came up with some effective searching selection menu in the GUI based on the experiments in the test collection and evaluation tasks.

Finally, in our opinion we can improve our system by setting some useful search setting such as stemmer. Also, we can improve our search engine system by making the searching menu more flexible.