

Report on Implementation of a text search engine

Liu Likun
2IE19337P
liu.likun.578@s.kyushu-u.ac.jp

Abstract—In this work, we've implemented a fully functional text search engine with input of keyword and output of relevant results with scores. The backend of the system was built with Python and GETA while the front-end utilized Vue framework and element UI. Two datasets in English and Japanese respectively were utilized in the system. English dataset is financial news in the United States during Jan 2018. The Japanese dataset is 250k Japanese Wikipedia data in plain text. The system can be accessed on: <http://13.231.178.152:8233/> (Inside Campus) or <http://dist.ecoresystems.cn/> (Globally available) and source code is available on: https://github.com/ecoresystems/distributed_system_course_project/.

Keywords—Search Engine, Python, Vue, Element UI, GETA

I. INTRODUCTION

In this work, we've implemented a fully functional text search engine with input of keyword and output of relevant results with scores. The backend of the system was built with Python and GETA while the front-end utilized Vue framework and element UI. Two datasets in English and Japanese respectively were utilized in the system. English dataset is financial news in the United States during Jan 2018. The Japanese dataset is 250k Japanese Wikipedia data in plain text. The original dataset was in json format and txt format. Pre-processing was utilized before the indexing procedure in conducted. In section II, we will give an overlook of our system architecture based on the data flow. Section III will give an introduction about the data pre-processing procedure as well as a brief introduction on the data itself. Section IV describes the indexing procedure, which explain the engine and method we use in detail. Section V will introduce the internal APIs and the query process. As we've built a user interface and an API for public access, section VI will describe the technology behind the user interface. In addition, a detail description of the API endpoint will also be provided in this section. In the deployment section (Section VII), we will provide a description of the deployment process along with the issues as of the time this report is written. Additionally, the thoughts about this course and acknowledgment will present at the last section. Finally, the screenshots of the system, the running log on the demo and some fix on the course's material will be provided in the Appendix.

II. SYSTEM OVERVIEW

A. System Architecture

The system implements the User Interface as a web service using HTML 5 and Vue framework. All communications between user front-end and server back-end was sent though RESTful API. The server back-end utilized a web micro framework called Flask. All search requests are sent to the server asynchronously. Requests received via RESTful API are parsed and the parameters passed to corresponding search engine. The search results generated by the search engine are sent to the Flask and encoded into JSON format as response data. The full architecture of the system is shown in Fig. 1.

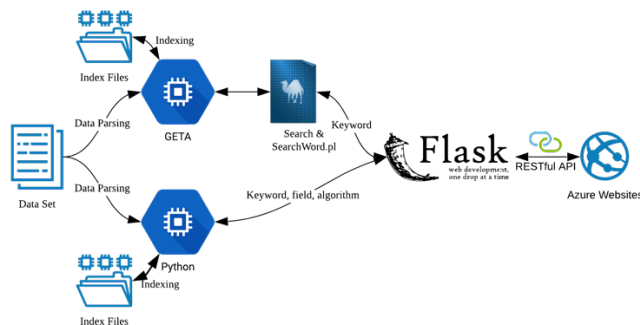


Fig. 1. System Architecture

III. DATA PRE-PROCESSING

The data used in this work are the US Financial News Articles in January 2018 and 250,000 rows of Japanese Wikipedia data. Such huge amount of data would require pre-processing before it can be indexed by a search engine. In aim of this, we've injected the data into a MySQL DBMS (Database Management System) for rapid query. Also, for the convince of reverse locating, each row of data was assigned with a UUID served as primary in the database.

A. Data Description

1) *US Financial News Articles*: Each news article is in a separate json file which contains various of informations including uuid, url, title, text, published date, etc. The preview of the json file is shown in Fig. 1.

```
{
  "root": {
    "organizations": [
      {
        "uid": "f86b78623d7a442c76c1d88e091184e2c29969",
        "thread": {
          "author": {
            "url": "https://www.cnbc.com/2018/01/03/emerging-markets-are-set-for-an-even-bigger-rally-in-2018-says-one-technician",
            "ord_in_thread": 0,
            "title": "Emerging markets are set for an even bigger rally in 2018, says one technician",
            "locations": [
              {
                "entities": [
                  {
                    "highlightText": "language: english",
                    "persons": [
                      {
                        "text": "17 Hours Ago | 02:56",
                        "text": "Emerging markets soared more than 33 percent in 2017, and Todd Gordon of TradingAnalysis.com says the rally won't A big part of the rally in emerging markets, tracked by the emerging market ETF EEM, was a weak dollar. And given 'We have a falling U.S. dollar, which will support international and emerging market currencies and will give those As for how high the latter could go, Gordon says EEM has broken 'resistance' at around $45, which was the ETF's 20 To play for a move higher, Gordon suggested buying the February 48/50 call spread for 72 cents, or $72 per options But if EEM were to close below $48, then Gordon would lose the $72 he paid for the trade. As a result, Gordon want 'If the 72 cent premium we just laid out gets cut in half to about 36 cents, let's cut the trade and move on,' he EEM started the year off strong, rallying more than 1 percent on Tuesday.",
                        "external_links": [
                          {
                            "published": "2018-01-03T13:00:00+00:00",
                            "crawled": "2018-01-03T13:34:36.006+00:00",
                            "highlightTitle": ""
                          }
                        ]
                      }
                    ]
                  }
                ]
              }
            ]
          }
        }
      }
    ]
  }
}
```

Fig. 2. Preview of US Financial Articles Data

2) *Japanese Wikipedia*: The total number of articles is 1,132,813 and the number of unique words is 2,420,073. Each line contains each article, and sentences are separated by tabs. The preview of the Japanese data is shown in Fig. 2.

リクトリア (小惑星) リクトリア (1107 Lictoria) は小惑星帯の小惑星である。イタリヤのピーノ・トリネーゼでルイージ・ヴォルタが発見した。軌道はヒギエア族に似ているが、大きさやスペクトルからそのメンバーではないと考えられる。古代ローマにおいて、ファスクスを携えて要人警護にあたったリクトルに因んで名付けられた。2008年2月に福島県で掩蔽が観測された。小林剛 (プロ雀士) 小林剛 (こばやし じょう、1976年2月12日-) は、競技麻雀のプロ雀士。東京都八王子市出身。麻将連合 - ム - 所属。東京理科大学 中退

Fig. 3. Preview of Japanese Wikipedia Data

B. Data Pre-processing

1) *US Financial News*: The data was already well formatted but for tens of thousands small files, it is a big challenge for I/O system to process. Therefore, we've injected the data into a DBMS. Further, to shrink the data size, we've selected UUID, URL, TITLE, CONTENT and PUBLISHED fields as our data. Fig. 3 gives a preview of the data in the database.

UUID	URL	TITLE	CONTENT	PUBLISHED
00017cde4df49b3...	https://www.reute...	Burst of snow hits the South, p...	Burst of snow hits t...	2018-01-17T22:...
0003a0ac216af0d...	https://www.wsj.c...	Dutch Skepticism About the E...	The Netherlands w...	2018-01-28T22:...
0003cb2a0212e23...	https://www.reute...	BRIEF-Aspen Aerogels Says C...	16 AM / in a few se...	2018-01-26T13:...
0004648ae4072b0...	https://www.reute...	- Chinese angels keep Wanda'...	HONG KONG Chin...	2018-01-30T15:...
0004a4fb845e3ef4...	https://uk.reuters...	Controversial sheriff pardoned...	Controversial sherif...	2018-01-09T20:...
00067570c43a75...	https://www.cnbc...	AriseBank™ Announces First...	DALLAS, AriseBan...	2018-01-18T17:...
0006cdbe2168df...	https://www.cnbc...	Fiat Chrysler CEO: Something...	Fiat Chrysler CEO:...	2018-01-16T16:...
0009c75aae3f77...	https://www.wsj.c...	With Workplace Suicides Risin...	As suicide rates ha...	2018-01-17T19:...
000a8223d05ba64...	https://uk.reuters...	Tennis-Raonic slumps to early...	January 16, 2018 /...	2018-01-16T09:...
000af862b74f2524...	https://www.reute...	UK opposition party grassroots...	January 4, 2018 /...	2018-01-04T02:...
000b5236ba392dd...	https://www.cnbc...	UPDATE 1-U.S. crude stocks...	(Adds details, price...	2018-01-18T18:...
000c0150f279dc9f...	https://www.cnbc...	Zapp360 Appoints New Leader...	NEW YORK, Jan. 2...	2018-01-25T16:...
000c311c22ace4...	https://uk.reuters...	El Salvador eyes work scheme...	January 17, 2018 /...	2018-01-17T04:...
000e01843dc868d...	https://www.cnbc...	ForgeRock Announces Key Ex...	SAN FRANCISCO...	2018-01-23T17:...
000f0ad483dec0b...	https://www.reute...	BRIEF-Pfizer CEO Says No Pr...	Jan 30 (Reuters) -...	2018-01-31T00:...
0011b2423d0200...	https://uk.reuters...	Motor racing-Back with old spo...	January 18, 2018 /...	2018-01-18T21:...
0012bc725b24b3e...	https://www.cnbc...	US towns that offer financial in...	SHARES College...	2018-01-04T16:...
0012e2fc57c2911...	https://www.cnbc...	CriticalPoint Capital Acquires t...	LOS ANGELES, Ja...	2018-01-08T16:...
0012e9f22d5029e...	https://www.cnbc...	QuoteWizard Acquires Bantam...	Deal enhances Qu...	2018-01-11T16:...
00134f934a44fec...	https://www.reute...	BRIEF-Groupe Ldrc Q3 Reven...	Jan 25 (Reuters) -...	2018-01-26T01:...

Fig. 4. US Financial News Data in a Database

2) *Japanese Wikipedia*: Since the data is plain text and did not contain other information, to make it identifiable in the database, we've created a random UUID served as primary key in database assigned to each row of data. Fig. 4 shows partial of the Japanese Wikipedia data in the database.

UUID	CONTENT
00007b3041b0...	ブジェンジカ ブジェンジカ (Brzezinka) は、ポーランド南部、クラクフから約60キロメートルにある広大...
0001336f41ab1...	田中真田中 真 (たなかまな) は、日本の法学者。東京大学社会科学部教授。専門は商法、法...
00014dc041af1...	上杉 実 (かみね じつ) は、1982年10月6日生まれ。日本の女優。岐阜県出身。
0001c4c641aa...	スタンリー・クラーク (スタンリー・クラーク) は、アメリカ合衆国のジャズ・バンドリーダー。...
0001d0c641ad...	マンガレイ・ベイ・トラム マンダレイ・ベイ・トラム (英:) は、ネパールの首都にあるラスベガス・ス...
0001e3e41ae1...	ジャック・ドズル ジャック・ドズル (Jacques Donzelot, 1943年 -) は、フランスの歴史社会学者、都市...
0001f3e41ae1...	酒井 直次 (陸軍軍人) 酒井 直次 (さかい なおじ、1891年 (明治24年) 3月26日 - 1942年 (昭和17年) ...
0002089041b2...	フェリット フェリット (Felitto) は、人口1,390人のイタリア共和国カンパニア州サレルノ県のコムーネの...
000210c841e9...	渡辺 寿 (名望家) ひさしは、幕末から明治時代の、山梨県の名望家、政治家、蔵書家。字を権...
00022a841b3...	シュルツェーナー (小惑星) シュルツェーナー (768 Struveana) は小惑星帯の小惑星である。クリミア半島...
0002659041b0...	村山 三男 村山 三男 (むらやま みつお、1920年4月11日 - 1979年7月29日) は、日本の映画監督、新演...
000272e941b1...	李 鴻章の乱 李鴻章の乱 (りしあいのらん) は、1467年 (享徳12年) に成吉思汗の成吉思汗の乱で起...
0002a8841ac...	チェルヴェーレ チェルヴェーレ () は、イタリア共和国ピエモンテ州クーネオ県にある、人口約2,300人...
00040c6841b3...	カミュー・ビダン カミュー・ビダン (Kamille Bidan) は、アニメ『機動戦士ガンダム』に登場する架空の...
000434041ab1...	PQ2 船団 PQ2 船団 は、第二次世界大戦中にイギリスからソ連へ支援物資を送るために運航された3...
0004591841af...	モンフォルテ・ダルバ モンフォルテ・ダルバ () は、イタリア共和国ピエモンテ州クーネオ県にある、人口...
0004eae421b2...	サブライド サブライド サブライド サブライド サブライド サブライド サブライド サブライド サブライド サブライド
0004f0441aa1...	三屋 鎮 (金堂 鎮) 三屋 鎮 (さんせい ちん) は中華人民共和国四川省成都市金堂鎮の鎮。三屋 鎮は以下の...
0005026441ad...	社会民主党 (東ティモール) 社会民主党 (しゃかいみんしやとう、略称 PSD) は、東ティモールの極端派中...
000514de41ab...	エンリクス・ゴンザレス・ゴンザレス は、1984年 (昭和59年) 7月から1990年 (平成2年) 6月まで、...
0005305c41ae...	ラロンデ・ゴードン ラロンデ・ゴードン (Laronde Kelda Gordon、1988年11月25日 -) は、トリニダード...
00054f0c41a91...	証証 制度 共同 事務局 証証 制度 共同 事務局 (にんしやうせいどうきょうどうじききく、英語表記: Secretari...

Fig. 5 Japanese Wikipedia Data in a Database

IV. DATA PARSING AND INDEXING

In this section, we've parsed and indexed the data in two different systems. The first system was implemented by Python called whoosh (whooshjp for Japanese package). The second system was GETA (Generic Engine for Transposable Association). As the data format required by different system as well as the nature of the data itself differs in 2 languages, we've parsed the data accordingly.

A. Python

The index process in python was straight forward. For English data (US Financial News Articles), we've set the UUID as ID and indexed other fields including title, content, URL and published date. For Japanese data (Japanese Wikipedia), since there wasn't enough field provide by the data, we've only indexed the content.

B. GETA

The GETA runs on Linux OS and only accept a plain text file with a special format to build the index. The sample file provides tools that written in Perl to create frequency file for both English and Japanese data. As we mainly built the system

on Python, we've modified the tools to accept arguments in aim of make it callable from the system.

1) *US Financial News Articles*: To provide as much agilability of searching as possible, we've provide two search field for the data: the title and the content. However, as the nature of the GETA, the two fields has to be treated as two different datasets using UUID to link together. We have added two flags to the original tool "mkfreq.pl": the "i" flags which stands for initial ID, and the "u" flag which will accept the UUID of the current content. The parsing process will produce a large frequency file. The sample of the frequency file is shown in Fig. 5. Note that this sample is also the same format as the text field as well as the format of the Japanese Wikipedia data (data format required by GETA). Then we used GETA to generate the corresponding wams data for indexing.

@75def43c-4f40-4e93-8366-fa111348cdd3 (UUID)

1 z
1 i:1
1 ...
2 ...

Fig. 6. Sample of the frequency file

2) *Japanese Wikipedia*: Similar to the process done to the US Financial News Articles, the Japanese Wikipedia data firstly us Mecab to divide sentence into words and then generated a frequency file. From there, the GETA engine was utilized to generate the wams file for the Japanese Wikipeda data.

V. INTERFACE AND QUERY

Subsequently, the next process is to provide a query interface for the applications to access. While the query interface provided by the system is a generic web API (RESTful API), there were also two sub-APIs for python to call.

A. Interfaces

1) *The Python Interface*: The python interface accepts serval parameters including: "fields" which distinguish the corresponding search target, "query_str" which indicates the corresponding search keyword string, "item_count" was the maximum number of the returned reslt, "data_source" was the selection of the data source, a selector to identify US Financial News Data and Japanese Data. "weighting_alg": we've provided 3 algorithms for the search process: Frequency, TF-IDF and BM25F.

2) *The GETA Interface*: The GETA interface was written in Perl, the Perl code used in the system has some modifications to the output statement to make the output easy for python to process. First, we've canceled the result serial number. Second, we've disabled the formatting strings in print statement e.g. %2d was reduced to %d. The out put of search.pl and searchWord.pl is shown in Fig. 7.

```

fe42d1dd84ecede8c67cbddd18dc3e255ffc161d
45cd9c911807bf41a7590e82f8c8bf33bba391a
3efdf9d2c1a989e4b2ef60ffc863a5236f86bf5
cb1e5f4527921a3041bfcd8c3d8183d0f1ab9147
c6f4000f0e8002cccf8c762fc7bbe4617d25450
564 41.759466 technology
382 1.485458 brief
253 0.699451 to
114 0.459746 of|
101 3.837625 net

```

Fig. 7. Output of the Perl Script

B. Query Process

1) *The Python Process:* When receiving a python query with parameters, the system will firstly determine the data source and open the corresponding index folder. Then the field and the key word string will be passed into the engine for querying. For the US Financial News Articles data, a result with title, score, url, content and uuid will be returned for each hit. Also, the runtime of the search and the number of hits is also returned. For the Japanese Wikipedia data, a result with title, score, content and uuid with runtime and total hist were returned.

2) *The GETA Query:* We've modified the ci.conf file to provide flexibility for the GETA to query. When receiving a request, the system will call "search.pl" and "searchWord.pl" to get the result and the corresponding score. Next, the system will query the MySQL database with the uuids of the searched result to get the content. We've tried to make the response as consistant as possible. As a result, a similar data response with score, url, title, content and uuid along with runtime and total hits was returned. A preview of the response data is shown in Fig. 8.

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```

Fig. 8 Response Preview

VI. THE USER INTERFACE

To provide relatively optimum user experience, we've built the user interface as a web application. The Front-end of the systems is written in HTML 5 and JavaScript based on Vue Framework and Element UI. Interactions between the front-end and the back end was done asynchronously via AJAX.

We've provided a total of 6 options:

Key word: The search word.

Data Source: switch between US Financial News Articles

Fields: switch between title and text

Engine: switch between Python and GETA

Algorithm: algorithm used for searching (Python engine only)

Limits: Maximum hits

The result is listed in a table with 4 fields: title, score, url and uuid. We've also added some additional features to the system. As the content might be too long to display, we will only display the title in the table. However, the content is available via a single click on the title. For the US Financial News Articles data, we've also preserved the original URL to the article itself for easy access. In addition, a popup window will be displayed upon each successful search indicating the total hits and the runtime of the search.

VII. DEPLOYMENT

Since the search engine is a functional system, we've deployed the system on two different servers: Kyushu University's QUEENS (HTTP Endpoint: <http://13.231.178.152:8233/> OS: Amazon Linux, Inside Kyushu University Only) and Microsoft Azure Cloud (HTTP Endpoint: <https://dist.ecoresystems.cn/> OS: Ubuntu Server 18.04 LTS, Globally Available). We've installed apache2 and MySQL DBMS on both system and installed the wsgi module to allow flask to serve the pages and provide the relative API (API is available on endpoint "/doc_search"). The API will accept XXX parameters. A complete parameters description is shown in Table 1.

Parameter	Type	Value	Description
data_source	String	US Financial News, Japanese Wiki	Specify which data to search.
search_engine	String	Python, GETA	Specify the search engine
key_words	String	User specific	Specify the search word
limit	Integer	User specific	Specify the max number of results
search_algorithm	String	TF-IDF, Frequency, BM25F	Specify the algorithm for the search
fields	String	Title, Text	Specify the fields for the search

Table 1. Required Request Parameters for the API

However, as of the time this report is written, there were still some issues during the deployment:

1) Due to the limited storage on the QUEENS (8.4GB total) and the huge data size from two datasets as well as index file for the both engine, it is confirmed that memory error is likely to occur during the query of Japanese Wikipedia data. (Insufficient swap memory)

2) As the flask is running on the root directory, the calls for Perl from Python is likely to fail and therefore affects the return results from GETA.

Nevertheless, the system is fully functional in localhost (local environment) and we will provide screenshots of the operational system in the appendix section. In addition, we will continue to resolve these problems and hopefully make both sites operational when this report is graded.

THOUGHTS ABOUT THE LECTURE AND ACKNOWLEDGEMENT

This lecture provides an introduction about the World Wide Web and the search engine mechanism. However, as there exist some bugs in the program as well as some minor issues in the course materials (sample program) as we will address in the appendix, the development process was severely delayed due to these issues. Also, as a conventional habit, this report uses **we** or **our** to indicate first person, but it is actually **I** since this work is one-man's work.

In addition, we would like to thank Microsoft for providing virtual machine and corresponding resources for free.

APPENDIX

A. Known Issues In Sample:

The GETA 2 program is known to have missed an INT_MAX value, originally I've resolve this with adding #define at each .c file, but there's a better solution by adding the #define INT_MAX <Max value of INT here> to the "limits.h" file located in include floder. The mkfreq.pl file will generate empty word every now and then, this will cause GETA to malfunction in some cases. Also, the end of line should be <FF> instead of "</p></div>
<div data-bbox="515 62 929 109" data-label="Image">

</div>
<div data-bbox="520 112 880 223" data-label="Table">
<table>
<tr>
<th>Title</th>
<th>Score</th>
<th>URL</th>
<th>UUID</th>
</tr>
<tr>
 BRIEF-Wuhan Thais Medical Technology to set up medical technology JV with partners | 11.255900849983028 | https://www.reuters.com/article/brief-wuhan-thais-medical-technology-to-brief-wuhan-thais-medical-technology-to-set-up-medical-technology-jv-with-partners-idUSL3N1P3SLR | 160bda82dc38c30770614d5855ce1c7462d2aa |
</tr>
<tr>
 Microbot Medical Enters into Agreement to Acquire Novel Technology to Enhance Existing Technology Platforms and Strengthen Patent Portfolio | 11.255900849983028 | http://www.cnn.com/2018/01/08/globe-news-wire-microbot-medical-enters-into-agreement-to-acquire-novel-technology-to-enhance-existing-technology-platforms-and-strengthen.html | 2181dee3c3f55deb4b085679818c20d9ec0335be |
</tr>
<tr>
 BRIEF-Urovo Technology Still Studying Potential Applications Of Blockchain Technology | 11.255900849983028 | https://www.reuters.com/article/brief-urovo-technology-still-studying-potential-applications-of-blockchain-technology-idUSL3N1P3SLR | 3efdf9d2c1a989e4b2e60f863a5236f86b5 |
</tr>
</table>
</div>
<div data-bbox="520 225 922 248" data-label="Caption">
<p>Screenshot 4. Search title field on US Financial News Articles based on TF-IDF</p>
</div>
<div data-bbox="515 259 929 369" data-label="Image">

</div>
<div data-bbox="545 393 893 405" data-label="Caption">
<p>Screenshot 5. Search on Japanese Wikipedia based on TF-IDF</p>
</div>
<div data-bbox="515 415 929 536" data-label="Image">

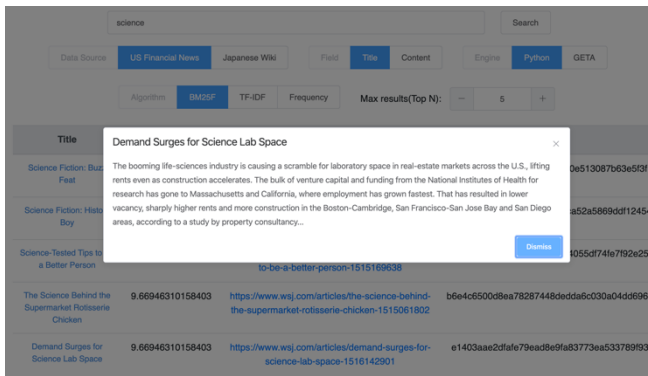
</div>
<div data-bbox="545 552 893 564" data-label="Caption">
<p>Screenshot 6. Search on Japanese Wikipedia based on BM25F</p>
</div>
<div data-bbox="71 279 480 294" data-label="Section-Header">
<h3>B. Screenshots of the system in Development environment:</h3>
</div>
<div data-bbox="71 300 475 364" data-label="Image">

</div>
<div data-bbox="195 368 360 381" data-label="Caption">
<p>Screenshot 1. Initial Interface</p>
</div>
<div data-bbox="71 391 475 539" data-label="Image">

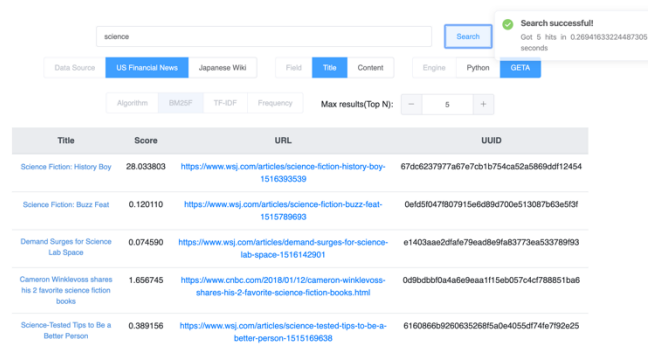
</div>
<div data-bbox="83 548 474 572" data-label="Caption">
<p>Screenshot 2. Search title field on US Financial News Articles based on Frequency</p>
</div>
<div data-bbox="71 582 475 743" data-label="Image">

</div>
<div data-bbox="83 744 474 766" data-label="Caption">
<p>Screenshot 3. Search title field on US Financial News Articles based on BM25F</p>
</div>
<div data-bbox="508 573 929 719" data-label="Image">

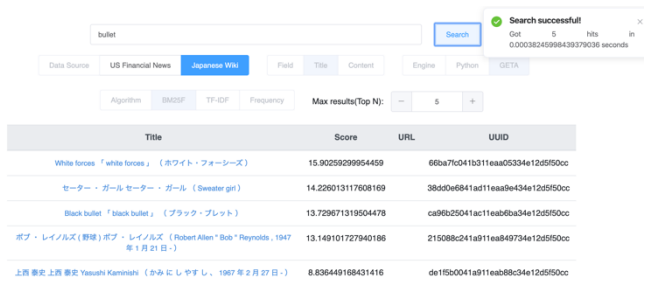
</div>
<div data-bbox="578 719 857 731" data-label="Caption">
<p>Screenshot 7. Article view of Japanese Wikipedia</p>
</div>
</div>



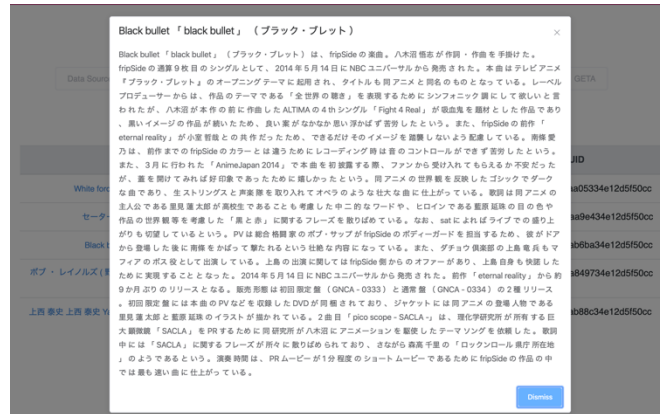
Screenshot 8. Article view of US Financial News Articles



Screenshot 9. Search on US Financial News Articles based on GETA



Screenshot 10. Search on Japanese Wikipedia data



Screenshot 11. Content view of Japanese Wikipedia data

C. Running Log of the demo system

The running log was uploaded to GitHub named debug.log and can be accessed via:
https://github.com/ecorecosystems/distributed_system_course_project/