NTTMU-SCHEMA BeCalm API in BioCreative V.5

Hong-Jie Dai*[1,2], Mira Anne C. dela Rosa[3], Ding-You Zhang[1],
Chung-Lin Chen[1], Chen-Kai Wang[3]

[1]Department of Computer Science and Information Engineering, National Taitung University, Taitung, Taiwan, R.O.C, [2]Interdisciplinary Program of Green and Information Technology, National Taitung University, Taitung, Taiwan, R.O.C, [3]Institute of Chemistry, Academia Sinica, Taipei, Taiwan, [4]Graduate Institute of Biomedical Informatics, Taipei Medical University, Taipei, Taiwan, R.O.C.

*hjdai@nttu.edu.tw; thegreatseer@gmail.com;
fogdingding@gmail.com;
u10411237@ms104.nttu.edu.tw;
dennisckwang@gmail.com

Abstract. With the emerging of new experimental techniques, there has been a remarkable increase in the amount of available biomedical data. Processing and mining large volumes of data in chemistry has now presented a challenging issue. In order to deal with the challenge, we developed SCHEMA (Spark-based CHEMicAl entity recognizer), a robust and efficient chemical entity recognition system on top of Apache Spark. SCHEMA is developed by following the asynchronous queue design pattern, which has been employed in service-oriented architecture for providing scalable and resilient services. SCHEMA that can retrieve patents in a form of unstructured free text from different websites and recognize chemical named entities described in them. To programmatically interact with SCHEMA, a restful Web application programming interface is provided. By using the custom request tests of the BeCalm (Biomedical annotation metaserver) platform, the test results illustrated that SCHEMA can process 5,000 patients within 5 minutes, indicating an average of only 0.06 second for processing one patent including the data fetch and analysis time.

Keywords. Chemical named entity recognition; Spark; parallel processing

1 Introduction

The emergence of new experimental techniques such as high throughput screening along with the development of automatic data mining algorithms bring forth large quantities of chemistry data. In addition to experimental data in publicly available databases such as PubChem [1], chemical patents represent one of the rich resources for chemical information. There is an increasing demand to efficiently mine the large scale of data in chemistry for the future development of chemical, pharmaceutical, agrochemical, biotechnological and fragrance industries [2].

In order to promote the effective access and integration of multiple text mining systems for processing unstructured document collections, BioCreative has had introduced the idea of the meta-services for biomedical information extraction since 2008 [3]. Despite the relevance of these previous efforts, some crucial aspects have been insufficiently or only partially addressed including continuous evaluation, extraction of textual content from heterogeneous sources, harmonization of multiple biomedical text annotations and visualization and comparative assessment of automatic and manual annotations. The BeCalm (Biomedical annotation meta-server) platform [4] in BioCreative V.5 provides the first solution to address the above mentioned issues.

As one of the participants in the BioCreative V.5 TIPS (Technical interoperability and performance of annotation servers) task, the NTTMU team developed SCHEMA[1], a Spark-based CHEMicAl entity recognition system, and implemented a REST (Representational State Transfer) application programming interface (API) to continuously listen and respond to the requests from BeCalm and other end users.

2 Method

We followed the messaging pattern in service-oriented architecture to develop SCHEMA, which enables the core of SCHEMA can be interacted with other services or applications through a loosely coupled and asynchronous message-based communication model. Figure 1 shows the detail workflow. The core of SCHEMA is a program runs on Apache Spark[2]. The SCHEMA core itself runs as a background process in an operation system and monitors messages sent to the request queue. When a processing request is posted to the request queue, the SCEHMA core retrieves that message and removes it from the queue. The request is then processed by several text mining workers implemented on Apache Spark

[1] SCHEMA is available at http://210.240.162.49/SCHEMA/

[2] Apache Spark (http://spark.apache.org) is a general engine for large-scale data processing.

engine. Through the above asynchronous queuing design pattern, SCHEMA can be deployed in many distributed environments and provide asynchronous, scalable and resilient services.

For interacting with the SCHEMA core through, we developed a REST web service, the SCHEMA Web Server in Figure 1, which continuously listens requests from BeCalm or authorized end users. The authorized request is then placed in a message queue waiting for processing. When the SCHEMA core completes the processing request, the results are placed in the response queue which will be later consumed by the thread of the SCHEMA web server and delivered to the requester in the format of BeCalm JSON.

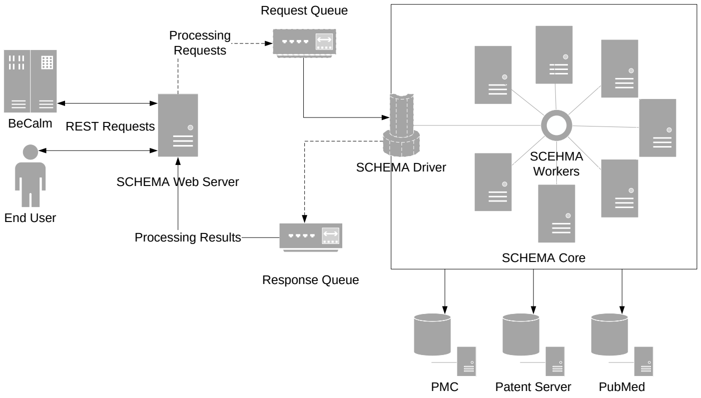


Figure 1. SCHEMA workflow.

The SCHEMA core is a Spark application consists of

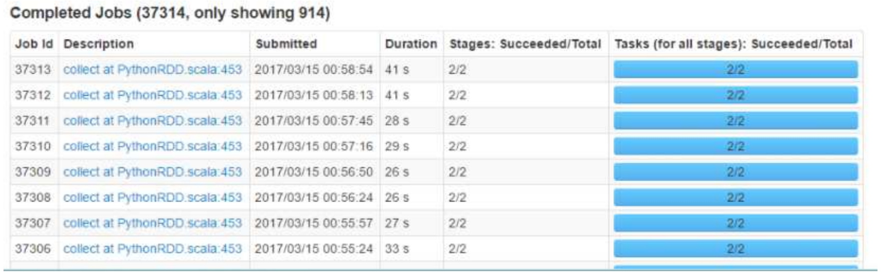
1. SCHEMA Driver: A Spark driver program that monitors the message queues and launches parallel text mining operations on the SCHEMA cluster. In the current implementation, the driver defines the distributed datasets based on the processing request and then asks the SCHEMA works to execute various operations to the datasets.
2. SCHEMA workers: The distributed agents that execute the text mining tasks. The tasks in current implementation includes: 1) download patterns from remote servers including PMC, PubMed and the patent server provided by the TIPS organizers, 2) sentence splitting, 3) tokenization, 4) part-of-speech tagging, 5)

chemical named entity recognition based on our previous work [5], and 6) recognition refinement.

In the current implement, the SCHEMA core was configured with 24 cores run on three virtual machines.

3 Results and Discussion

Figure 2 shows the web console of SCHEMA core. The results illustrated the processing time for each BeCalm request in the TIPS evaluation phase 1. In the phase, BeCalm requests for processing one document per request to validate the implementation of our annotation server within variable time intervals. As one can see that the SCHEMA core spends around 40 seconds to process one patent including the data fetch time from remote databases and the text mining processing time.



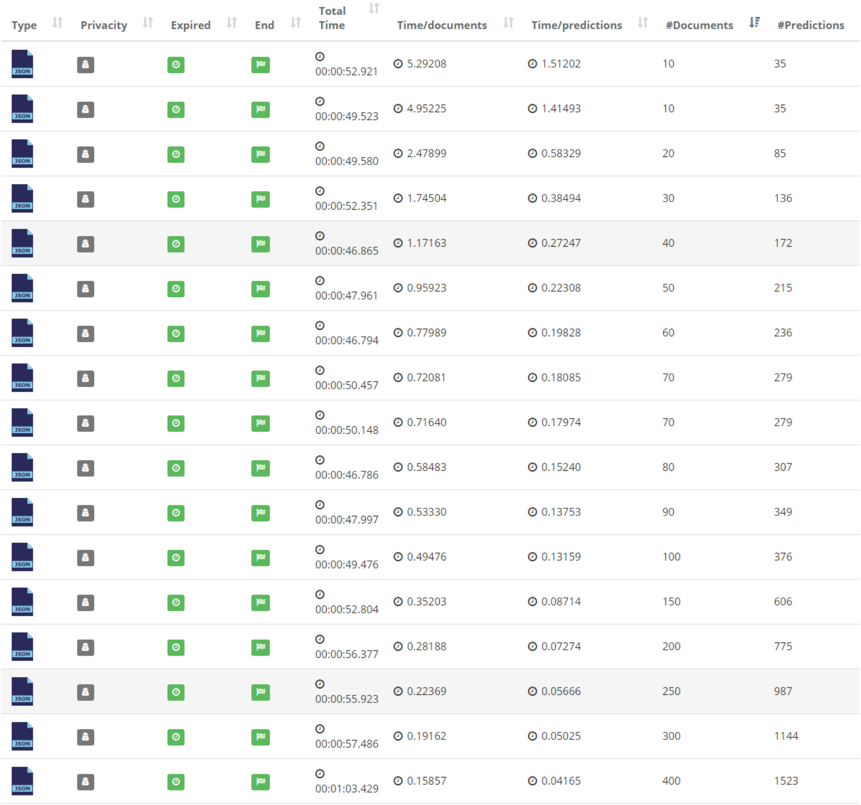
Completed Jobs (37314, only showing 914)

Job Id	Description	Submitted	Duration	Stages: Succeeded/Total	Tasks (for all stages): Succeeded/Total
37313	collect at PythonRDD.scala:453	2017/03/15 00:58:54	41 s	2/2	2/2
37312	collect at PythonRDD.scala:453	2017/03/15 00:58:13	41 s	2/2	2/2
37311	collect at PythonRDD.scala:453	2017/03/15 00:57:45	28 s	2/2	2/2
37310	collect at PythonRDD.scala:453	2017/03/15 00:57:16	29 s	2/2	2/2
37309	collect at PythonRDD.scala:453	2017/03/15 00:56:50	26 s	2/2	2/2
37308	collect at PythonRDD.scala:453	2017/03/15 00:56:24	26 s	2/2	2/2
37307	collect at PythonRDD.scala:453	2017/03/15 00:55:57	27 s	2/2	2/2
37306	collect at PythonRDD.scala:453	2017/03/15 00:55:24	33 s	2/2	2/2

Figure 2. SCHEMA core web console for processed jobs.

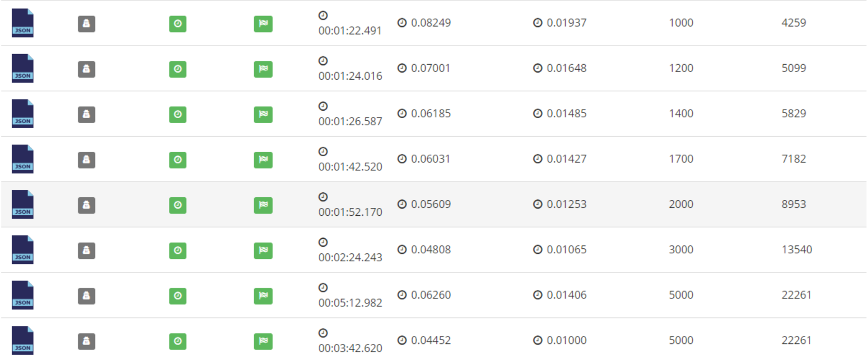
Figure 3 shows the evaluation interface provided by BeCalm platform. Here we used the custom request function to adjust the number of documents in a request to examine the processing time of SCHEMA. As one can see that SCHEMA spends around 50 seconds to process 10 patents. With the numbers of patents per request increasing from 10 to 300, we did see significant increase of processing time of SCHEMA. We can also observe that the time for processing one patent was decreased from ~5 seconds to 0.192 seconds. The time for generating one predication was also reduced from 1.512 seconds to 0.050 seconds.

In addition, Figure 4 shows the processing time of requests containing more than 1000 patents. SCHEMA can process 2000 documents per request within two minutes. We can also observe that the average processing time per document did not increase when the number of documents per request increase. All of the above results illustrate the reliability and the power of parallel processing of SCHEMA core.



Type	Privacity	Expired	End	Total Time	Time/documents	Time/predictions	#Documents	#Predictions
JSON	🔒	⊙	⚑	00:00:52.921	5.29208	1.51202	10	35
JSON	🔒	⊙	⚑	00:00:49.523	4.95225	1.41493	10	35
JSON	🔒	⊙	⚑	00:00:49.580	2.47899	0.58329	20	85
JSON	🔒	⊙	⚑	00:00:52.351	1.74504	0.38494	30	136
JSON	🔒	⊙	⚑	00:00:46.865	1.17163	0.27247	40	172
JSON	🔒	⊙	⚑	00:00:47.961	0.95923	0.22308	50	215
JSON	🔒	⊙	⚑	00:00:46.794	0.77989	0.19828	60	236
JSON	🔒	⊙	⚑	00:00:50.457	0.72081	0.18085	70	279
JSON	🔒	⊙	⚑	00:00:50.148	0.71640	0.17974	70	279
JSON	🔒	⊙	⚑	00:00:46.786	0.58483	0.15240	80	307
JSON	🔒	⊙	⚑	00:00:47.997	0.53330	0.13753	90	349
JSON	🔒	⊙	⚑	00:00:49.476	0.49476	0.13159	100	376
JSON	🔒	⊙	⚑	00:00:52.804	0.35203	0.08714	150	606
JSON	🔒	⊙	⚑	00:00:56.377	0.28188	0.07274	200	775
JSON	🔒	⊙	⚑	00:00:55.923	0.22369	0.05666	250	987
JSON	🔒	⊙	⚑	00:00:57.486	0.19162	0.05025	300	1144
JSON	🔒	⊙	⚑	00:01:03.429	0.15857	0.04165	400	1523

Figure 3. BeCalm evaluation interface for processing 10 to 400 patents.



JSON	🔒	⊙	⚑	⊙ 00:01:22.491	⊙ 0.08249	⊙ 0.01937	1000	4259
JSON	🔒	⊙	⚑	⊙ 00:01:24.016	⊙ 0.07001	⊙ 0.01648	1200	5099
JSON	🔒	⊙	⚑	⊙ 00:01:26.587	⊙ 0.06185	⊙ 0.01485	1400	5829
JSON	🔒	⊙	⚑	⊙ 00:01:42.520	⊙ 0.06031	⊙ 0.01427	1700	7182
JSON	🔒	⊙	⚑	⊙ 00:01:52.170	⊙ 0.05609	⊙ 0.01253	2000	8953
JSON	🔒	⊙	⚑	⊙ 00:02:24.243	⊙ 0.04808	⊙ 0.01065	3000	13540
JSON	🔒	⊙	⚑	⊙ 00:05:12.982	⊙ 0.06260	⊙ 0.01406	5000	22261
JSON	🔒	⊙	⚑	⊙ 00:03:42.620	⊙ 0.04452	⊙ 0.01000	5000	22261

Figure 4. BeCalm evaluation interface for processing requests with more than 1000 patents.

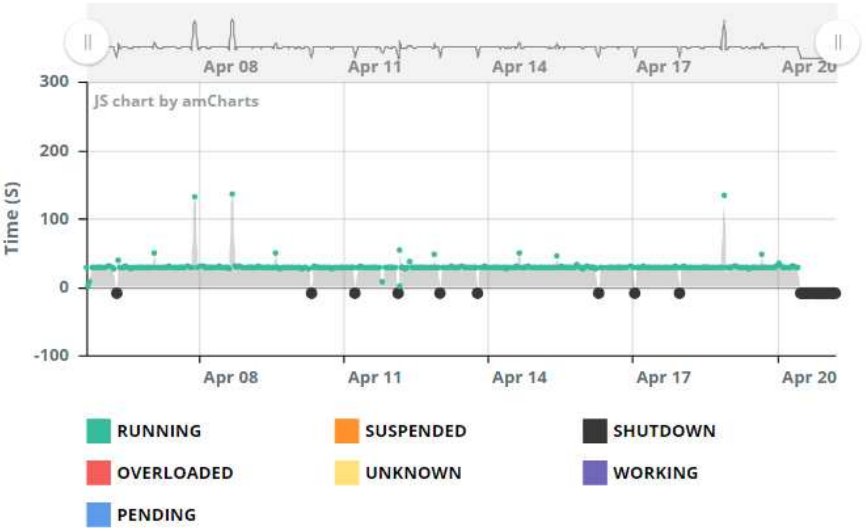
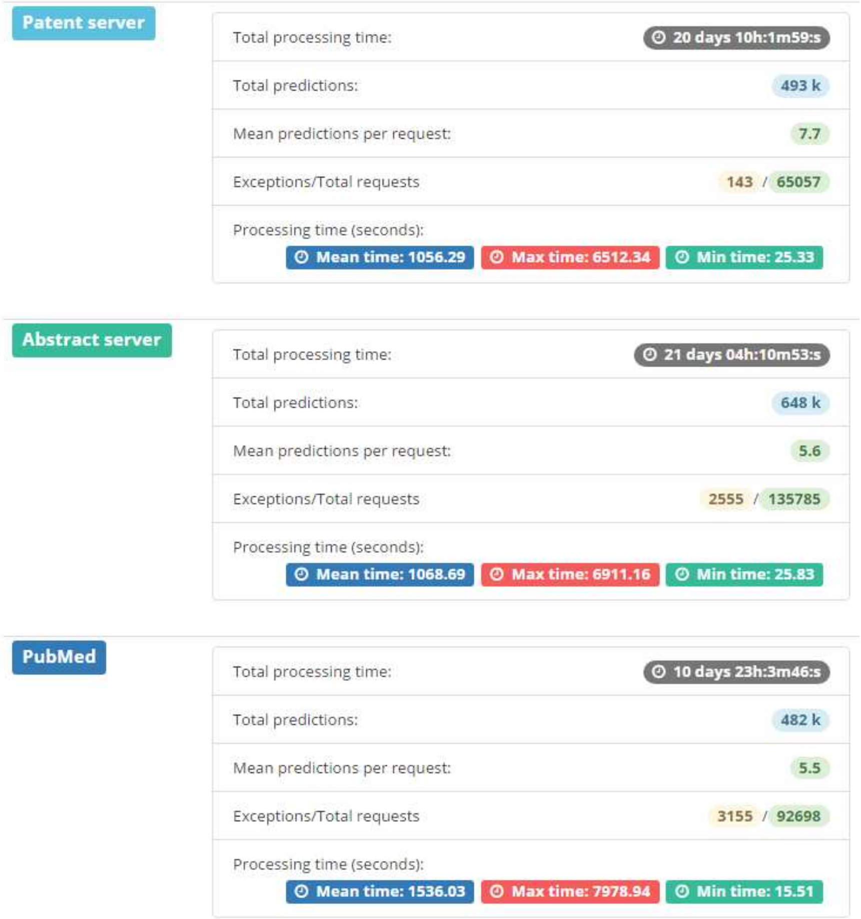


Figure 5. The official historical server response timeout record.

Figure 5 and 6 show the official evaluation results on the BeCalm platform. Started from 2017/1/23, SCHEMA processed 292,540 requests and generated around 493k, 648k, and 482k predictions for the patent server, abstract server and PubMed, respectively. At the end of March, we shutdown SCHEMA for one week because we had other computing tasks required for running on the VM server. However, we forgot to stop the SCHEMA web server, therefore it continues to accept request from BeCalm. After we restarted the SCHEMA driver, it started to process all received requests in the request queue and replied to BeCalm, which may one of the reason lead to the large max processing time.

Among all requests, SCHEMA generated 5,853 exceptions. Most of the exceptions were occurred with an error message of "Request getState not retrieve data" which occurred constantly since the beginning of April after we restart our SCHEMA server. Since then BeCalm cannot receive the correct running state of SCHEMA and throws aforementioned exceptions. We don't know the reason cased the errors.



Patent server

Total processing time:	20 days 10h:1m59:s
Total predictions:	493 k
Mean predictions per request:	7.7
Exceptions/Total requests	143 / 65057
Processing time (seconds):	Mean time: 1056.29 Max time: 6512.34 Min time: 25.33

Abstract server

Total processing time:	21 days 04h:10m53:s
Total predictions:	648 k
Mean predictions per request:	5.6
Exceptions/Total requests	2555 / 135785
Processing time (seconds):	Mean time: 1068.69 Max time: 6911.16 Min time: 25.83

PubMed

Total processing time:	10 days 23h:3m46:s
Total predictions:	482 k
Mean predictions per request:	5.5
Exceptions/Total requests	3155 / 92698
Processing time (seconds):	Mean time: 1536.03 Max time: 7978.94 Min time: 15.51

Figure 6. Statistics by document provider.

4 Conclusion

In the paper, we give a briefly introduction of the development of NTTMU team's SCHEMA. The architecture of SCHEMA is flexibly and its computing power can be easily scale up by adding more cores within the Apache Spark platform. In the future, we will review the performance report evaluated by the TIPS task and study the effect of different configurations of our SCHEMA cores.

5 Acknowledgment

The study was supported by Ministry of Science and Technology (MOST) 105-2221-E-143-003 and Ministry of Education-Talent Cultivation Program for Advanced Mobile Broadband Technology-Big Data module.

REFERENCES

1. Wang Y, Xiao J, Suzek TO, Zhang J, Wang J, Bryant SH: **PubChem: a public information system for analyzing bioactivities of small molecules**. *Nucl Acids Res* 2009, **37**(suppl_2):W623-633.
2. Tetko IV, Engkvist O, Koch U, Reymond JL, Chen H: **BIGCHEM: challenges and opportunities for Big Data analysis in chemistry**. *Molecular Informatics* 2016, **35**(11-12):615-621.
3. Leitner F, Krallinger M, Rodriguez-Penagos C, Hakenberg J, Plake C, Kuo C-J, Hsu C-N, Tsai R, Hung H-C, Lau W *et al*: **Introducing meta-services for biomedical information extraction**. *Genome Biology* 2008, **9**(Suppl 2):S6.
4. Pérez-Pérez M, Pérez-Rodríguez G, Blanco-Míguez A, Fdez-Riverola F, Valencia A, Krallinger M, Lourenco A: **Benchmarking biomedical text mining web servers at BioCreative V.5: the technical Interoperability and Performance of annotation Servers - TIPS track**. In: *Proceedings of the BioCreative V5 Challenge Evaluation Workshop*. 2017: 12-21.
5. Dai HJ, Lai PT, Chang YC, Tsai RT: **Enhancing of chemical compound and drug name recognition using representative**

tag scheme and fine-grained tokenization. *Journal of Cheminformatics* 2015, **7**(Suppl 1 Text mining for chemistry and the CHEMDNER track):S14.