Chapter 5

Big Data Framework for Biomedical Literature Mining

The massive size of available biomedical literature requires researchers to utilize novel big data technologies in data storage and analysis. Among them is cloud computing which has become the most popular solution for big data applications in industry. However, many bioinformaticians still rely on expensive and inefficient in-house infrastructure to discover knowledge from biomedical literature. Although some cloud-based solutions were constructed recently, they failed to sufficiently address a few key issues including scalability, flexibility, and reusability. Moreover, no study has taken computational cost into consideration. To fill the gap, we proposed a cloud-based big data framework that enables researchers to perform reproducible and scalable large-scale biomedical literature mining in an efficient and cost-effective way. Additionally, a cloud agnostic platform was constructed and then evaluated on two open access corpora with millions of full-text biomedical articles. The results indicate that our framework supports scalable and efficient large-scale biomedical literature mining.

This work was originally published as: Shen, Z., Wang, X., & Spruit, M. (2019, June). Big Data Framework for Scalable and Efficient Biomedical Literature Mining in the Cloud. *In Proceedings of the 2019 3rd International Conference on Natural Language Processing and Information Retrieval* (pp. 80-86). https://doi.org/10.1145/3342827.3342843

5.1 Introduction

A huge amount of biomedical literature has been produced over the decades, and the number is still growing on a daily basis. Furthermore, with the advocacy of open science, an increasing number of publications become openly accessible. For example, as the leading biomedical literature database, PubMed Central (PMC) has archived over 5.3 million research papers, of which around 2.4 million full-text articles are easily accessible at the PMC Open Access Subset (PMC OA Subset) (Pubmed Central, 2019; Pubmed Open Access, 2019). The size of the latest update of full-text articles in the PMC OA Subset is around 100 GB, so storing, processing, and analyzing such massive data is not exactly trivial.

The difficulty of extracting hidden useful knowledge from literature increases as the number of literature arise. Analyzing today's massive biomedical literature, especially full-text articles, is a challenging task. The huge amount of available full-text articles results in very high computational and storage requirements. However, existing biomedical text mining tools that run on single commodity hardware cannot meet the requirements. For instance, a full run of BioContext on 20 million MEDLINE abstracts and 234 thousand PMC full-texts took 2-3 months (Gerner et al., 2012). To tackle this infrastructure barrier, cloud-based solutions that integrate multiple off-the-shelf big data technologies including parallel computing and cloud computing were introduced (Luo et al., 2016). Parallel computing is a fundamental infrastructure which enables the execution of data analysis tasks simultaneously on a cluster of machines or supercomputers (Luo et al., 2016). A prominent example is Hadoop, an open-source MapReduce package for distributed data management (Dean and Ghemawat, 2004). Cloud computing is a new paradigm for sharing computational resources across the Internet. It offers a scalable and cost-effective solution for big data analysis (Assunção et al., 2015). Amazon and Google are today's two biggest cloud infrastructure providers (Rajdho and Biba, 2013).

Nevertheless, the development of cloud-based solutions for large-scale literature mining is complicated, requiring IT experts to integrate a number of specific technologies, such as massive parallel processing, distributed databases, scalable storage systems, and a variety of third-party text mining libraries (Assunção et al., 2015; Yang et al., 2017b). Additionally, the configuration and management of cloud infrastructure are notoriously difficult and resource-intensive (Hendrickson et al., 2016). Since bioinformaticians specialized in text mining or natural language processing often lack the experience of managing cloud infrastructures, it is of great necessity that there is an integrated cloud-based platform which relieves them from the burden of technical details, such as cloud setups and configurations. On the other hand, faced with various text mining needs, biomedical scientists need a great deal of freedom when it comes to selecting natural language processing (NLP) and text mining techniques (Lamurias and Couto, 2019).

In this work, we first proposed a big data framework named SELM to facilitate large-scale biomedical literature mining by utilizing cloud computing (Bahrami and Singhal, 2015) and Apache Spark (Zaharia et al., 2016). In specific, SELM incorporates a storage layer that supports cost-effective and scalable data storage in the cloud, and a management layer which manages both cloud infrastructure and text mining applications. An implementation of the framework was then presented as a cloud-based computational platform. The main characteristics of the platform are as follows: (1) scalable, (2) flexible, (3) efficient, (4) affordable, (5) reusable analysis to improve research reproducibility. Furthermore, the code of our implementation is open source and available on GitHub (Shen and Wang, 2019).

5.2 Related literature

Over the last few years, a number of platforms or systems have been produced to perform text mining tasks on large-scale literature corpora. For instance, Labropoulou et al. devised the OpenMinTed platform to support text mining of open access scholarly content (Labropoulou et al., 2018). Users can conduct text mining on self-defined corpora with graphic user interfaces. Textpresso Central is a similar tool crafted specifically for the biomedical domain (Müller et al., 2018). Although both tools are proven to be effective and efficient for researchers to discover useful insights from literature, they do not address big data challenges at all. To fill the gap, studies on large-scale biomedical text mining introduced software with the capacity for handling big data. SparkText, a text mining framework based on Apache Spark, was created and evaluated in (Ye et al., 2016). Tafti et al. proposed their big data analytics system to identify adverse drug events (ADEs) from literature and social media posts (Tafti et al., 2017). Besides, high performance computers were also employed in large-scale biomedical text mining (Ide et al., 2018; Xing et al., 2018). Nonetheless, given that supercomputers, such as Tianhe-2 (Liao et al., 2014), XSEDE (Towns et al., 2014), are not reachable to most researchers, the supercomputer dependent frameworks lack practical implications.

Table 5.1 summarizes the main characteristics of the existing platforms or systems, including open source, cloud support, scalability, flexibility, and reproducibility.

- **Open source**: Releasing the code of a software under open source licenses becomes a popular practice in software industry and academia. It enables people to freely reuse the software for their own purposes (McKiernan et al., 2016). Whether a platform is open source is usually explicitly described.

- **Cloud support**: As discussed previously, cloud-based solutions are scalable and cost-effective. We examined the existing tools on whether they support large-scale text mining in the cloud.

- **Scalability**: It refers to how well a platform can scale up to large-scale literature mining. The size of corpora in the evaluation reflects the scalability of a system. According to the evaluation results stated in (Labropoulou et al., 2018; Müller et al., 2018; Ye et al., 2016; Tafti et al., 2017; Ide et al., 2018; Xing et al., 2018), we divided into high, medium and low.

- **Flexibility**: Given that a complex collection of text mining techniques are required for biomedical literature mining tasks (Lamurias and Couto, 2019), platforms or systems are not able to include all text mining techniques for users. Therefore, a certain degree of freedom should be granted to users in customizing text mining methods according to their specific needs. Flexibility reflects the level of freedom.

- **Reproducibility**: Reproducing computational experiments with a high degree of certainty is becoming a very important factor in assessing the quality of one's research (Korolev and Joshi, 2014). How well a platform supports the development of reproducible text mining pipelines determines the level of its reproducibility.

As shown in Table 5.1, not all existing platforms have cloud support. But there is a consensus among researchers about the benefits of utilizing cloud-based solutions for big data (Alharthi et al., 2017). Healthcare system developed on the cloud has shown its capacity to collect, store, and analyze big health data (Zhang et al., 2017). Geospatial scientists also utilized cloud computing to tackle big geospatial data challenges (Yang et al., 2017a).

Paper/Platforms	Description	Open source	Cloud support	Scalability	Flexibility	Reproducibility
OpenMinTed Labropoulou et al. (2018)	A platform facilitating text mining of scholarly content	Yes	Yes	Low	Medium	High
Textpresso Central Müller et al. (2018)	A customizable platform for searching, text mining, viewing, and curating biomedical literature	No	Yes	Low	Medium	Medium
bigNN Tafti et al. (2017)	A scalable framework to analyze ADEs from large-scale biomedical text	No	No	Medium	Low	Low
SparkText Ye et al. (2016)	an efficient text mining framework built on big data infrastructure and a Cassandra NoSQL database	No	No	Medium	Medium	Low
LAPPS Grid Ide et al. (2018); Towns et al. (2014)	An open, interoperable web service platform for natural language processing (NLP) research and development	Yes	Yes	Medium	Medium	High
ParaBTM Xing et al. (2018); Liao et al. (2014)	A runnable framework that enables parallel text mining on the Tianhe-2 supercomputer	Yes	No	High	Low	Low

Table 5.1: Summary of Existing Tools for Large-scale Biomedical Literature Mining

5.3 Framework

In this section, we present a framework which is composed of three main components for storing, analyzing massive biomedical literature and a underlying cloud infrastructure layer.

As displayed in Figure 5.1, the framework provides a high-level representation of our large-scale biomedical literature mining solution which is cloud-based and built upon Infrastructure-as-a-Service (IaaS). Instead of being limited to IaaS from a specific provider, it is compatible with all major public cloud infrastructures.

The storage layer utilizes the distributed file system from the cloud providers. It

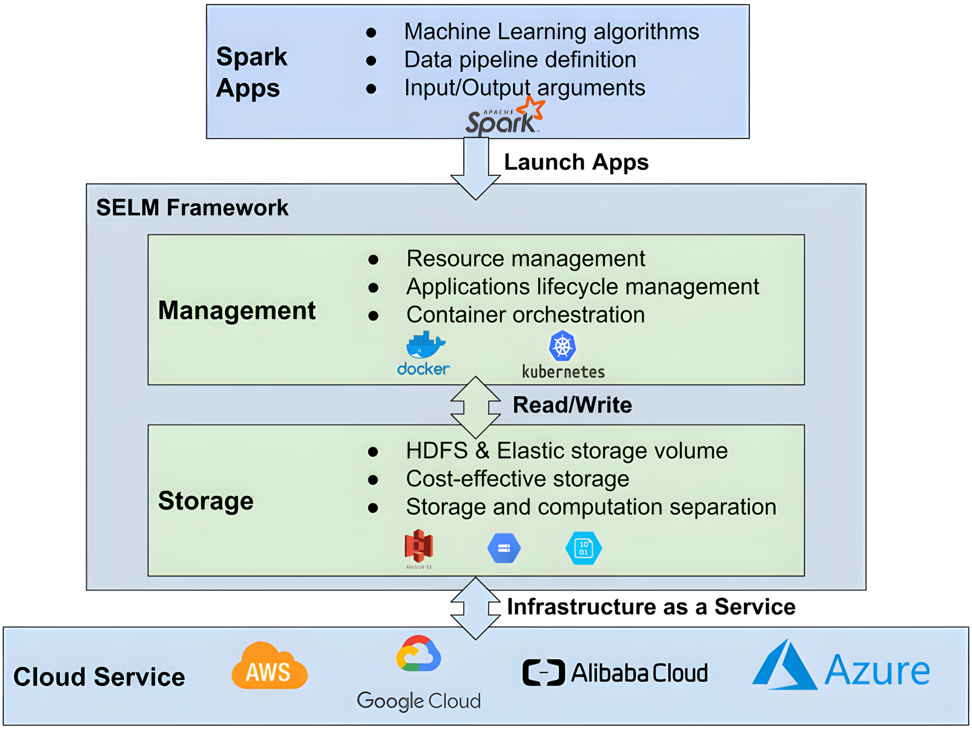

Figure 5.1: SELM: the Big Data Framework for Scalable and Efficient Biomedical Literature Mining in the Cloud

is extremely cost-effective and scalable due to the separation of storage and computation. High availability and data loss prevention are also secured at the same time. In addition, the accessibility of the storage layer is further extended to various use cases that are not bounded with particular machines or networks. In practice, Amazon Web Services (AWS) S3, Google cloud storage and Azure blob storage are employed for the reason that they are scalable for unstructured data with the vendor-variant Hadoop Distributed File System (HDFS) (Shvachko et al., 2010).

In the management layer, Kubernetes and Docker are employed to manage computational and storage resource, and Spark applications at scale. Spark applications written by bioinformaticians in the Spark Apps layer are built as Docker images which is an executable package. Firstly, it allows users to easily manage the lifecycles of Spark applications with simple configuration files. Secondly, it facilitates the computation resources to interact with the storage layer in an orderly fashion which is based on user-defined requirements. To scale up to run Spark applications on a large cluster, Kubernetes is introduced to orchestrate the Docker containerized Spark applications.

As the external component of the framework, the Spark apps layer allows biomedical researchers to easily created reusable text mining modules with Apache Spark based on their specific needs. To begin with, biomedical researchers can create a Spark application with some widely used programming languages, including Python, Java, Scala, and R. Moreover, the built-in libraries in Apache Spark and third-party NLP libraries make it simpler. Before running a Spark application, Kubernetes deployment configurations should be added so that the application could be scaled up to running on a large number of servers. As mentioned before, Spark applications are assembled as Docker images in this layer which could be easily distributed and deployed across platforms (Boettiger, 2014). Therefore, biomedical literature mining tasks created for the framework are reusable.

5.4 Evaluation

To evaluate the performance of the framework, a cloud platform was constructed and evaluated on two large-scale full-text articles datasets. This section briefly discusses the datasets, the details of the experiment setup and evaluation metrics.

Dataset	**Year range**	**Size**	**Task**	**Model**	**Cloud Specifications**	**Evaluation metrics**
PMC OA Subset	1918-2019	1,010,787	Vilidate the scalability with large biomedical literature corpora	LDA	190G RAM, 64 cores	Execution time Estimated cost
SparkText	2009-2016	29,437	Classifying full-text articles into breast lung or prostate cancer	Naive Bayes	80G RAM, 16 cores	Execution time Estimated cost Accuracy Precision Recall

Table 5.2: Details about Datasets, Experiment Setup, and Evaluation Metrics

5.4.1 Datasets

The first dataset contains over one million full-text biomedical articles from PubMed Open Access Subset. The articles were obtained via the PMC FTP service as zip files (Pubmed Open Access, 2019). With this dataset, we intend to demonstrate the powerful scalability of our approach by conducting topic modelling on three big samples. The biggest sample contains 1010787 full-text articles, and the smallest one comprises

267471 articles. The second dataset devised by Ye et al. in (Ye et al., 2016) was a collection of labeled biomedical articles from PubMed. More specifically, it contains 29437 full-text articles which are divided into three groups: breast cancer, lung cancer, and prostate cancer. we employed 80% of the entire dataset to train a classification model while the remaining 20% was used for testing.

5.4.2 Experiment Setup

An implementation of SELM has been created to verify our design and to evaluate its performance on analyzing big biomedical literature corpora. The implementation was constructed on top of the Google Cloud Platform. In particular, the cloud service in our implementation is Google Cloud. The storage layer relied on Google Cloud Storage. An open source software, Kubernetes Operator for Apache Spark (GoogleCloudPlatform, 2019), was utilized for the management layer. These tools were integrated into a cloud platform which supports running Spark applications in a scalable and efficient manner. The specifications of the cloud infrastructure in each experiment are in Table Table 5.2.

Spark applications were developed in Scala 2.11. Each Spark application consists of two main components: NLP pipeline and data modeling. In the NLP pipeline, raw text is preprocessed and vectorized into features with the built-in Apache Spark MLlib and the external NLP library for Apache Spark, namely Spark-NLP from the Johnsnow labs. Data modeling takes into the extracted features from the NLP pipeline to train the latent Dirichlet allocation (LDA) for topic extraction and the Naive Bayes model for document classification.

5.4.3 Evaluation Metrics

The performance metrics applied in the evaluation include execution time, estimated cost, and three common prediction measures: accuracy, precision, and recall. Execution time refers to the amount of time it takes to complete the execution of a Spark application in the cloud. Long execution time would significantly slow the development of text mining solutions. In most cases, data scientists obtain their best results through numerous rounds of execution. Thus, less execution time means more efficiency and synergies in text mining. Since cloud computing shares computational resources through the Internet, execution time can be reduced by expanding the number of cloud servers. However, the bigger the cloud is, the expensive it becomes. Taking affordability into consideration, we calculated the cloud computational cost for each experiment. Since all cloud infrastructure providers support the pay-as-you-go model, costs are estimated on the basis of execution time.

Besides execution time and estimated cost, accuracy, precision, and recall are employed to measure the prediction performance of the document classification model trained on the second dataset. From the classification point of view, four possible outcomes are defined: true positive (TP), true negative (TN), false positive (FP), and false negative (FN). Based on the four outcomes, accuracy, precision and recall are calculated.

- Accuracy: it is calculated as the ratio of correctly classified document to the total number of documents. Accuracy = (TP + TN) /(TP + TN + FP + FN)
- Precision: it measures the exactness of a prediction. It refers the percent of positive predictions that are accurate. Precision = TP/(TP + FP)
- Recall: it measures the sensitivity of a classifier. It is calculated as Recall = TP/(TP + FN) To compare the performance of our approach,

we used the performance of a big data framework in (Ye et al., 2016) as the benchmark to conduct comparison.

5.5 Results

In this section, we present the results of two experiments and discuss the implications and limitation of the evaluation.

5.5.1 Topic Modelling on the PMC OA Subset

We demonstrated the scalability of SELM with a big biomedical literature corpus which contains over one million biomedical full-texts articles. To our best knowledge, there is no research on biomedical literature mining that has analyzed data of this size, even in those where big data frameworks were developed for biomedical literature mining (Ye et al., 2016; Tafti et al., 2017; Xing et al., 2018). Moreover, as a generative probabilistic model, LDA is computationally intensive. The result of this experiment also applies to text mining tasks using other common classification models, including Support Vector Machine (SVM), Logistic Regression, and Random Forest.

Performance comparison of our approach and other systems for large-scale biomedical literature mining is shown in Table 5.3 The analysis of over one million full-text articles took around 25 minutes with a cost of around 1.5 US dollars. Our implementation only took nine minutes to analyze 267,471 full-text articles. It took six minutes to build a classification model with 29,437 full-text articles on SparkText. The execution time of running the same analysis on the same dataset in the second experiment is only one minute. Therefore, in terms of time efficiency, our framework is around six times faster than SparkText. Furthermore, when it comes to cost, our framework's is much lower than that of the other three systems. It offers an affordable means for biomedical scientists to fully utilize the available biomedical literature.

	Size	**Model**	**Time (minutes)**	**Cost ($)**
SELM	1,010,787	LDA	25	1.5
	428,152	LDA	15	0.75
	267,471	LDA	9	0.45
	29,437	Naive Bayes	1	0.02
bigNN	14,017	SVM	88.9	597*
	21,843	Naive Bayes	135.3	
SparkText	29,437	Naive Bayes	6	1100*
ParaBTM	61,078	Conditional Random Fields	240	High**

* Monthly cost estimated on server specifications provided on the papers and the prices of Google Cloud.

** The authors stated that the cost of a full run of over 1 million full-text articles from PMC is beyond their funding budget.

Table 5.3: Performance Comparison with Large Corpus

5.5.2 Cancer Articles Classification

Table 5.4 shows the comparison of execution time, estimated cost and the prediction metrics between our solution and SparkText. Like SparkText, we trained a Naive Bayes classification model on a dataset of 29,437 full-text articles. First of all, SELM is six times faster than SparkText. Secondly, the cost of execution with our framework is much less and more flexible. In comparison with SparkText our approach do not require a server running at 24/7. Cloud infrastructure only starts on demand and costs are calculated based on the amount of actual running time. Last but not least, prediction results are improved with our approach. Therefore, it is safe to claim that our framework is a better choice in handling large-scale biomedical literature mining.

	Accuracy	**Precision**	**Recall**	**Time (minutes)**	**Cost ($)**
SELM	90.28%	90.36%	90.28%	1	0.02
SparkText	86.44%	87.61%	89.12%	6	1100

Table 5.4: Performance on Cancer Articles Classification

5.6 Discussions

The execution time and estimated costs from both experiments indicate that the proposed framework is more efficient and affordable in large-scale biomedical literature mining. The fact that the framework supported the analysis of over one million full-text articles exhibits its good scalability in handling big data. No existing biomedical literature mining system has demonstrated similar capacity. Besides, the prediction performance in the second experiment suggests that SELM is capable of being scaled down to medium-sized datasets. As mentioned previously, the framework is designed as a flexible platform that supports a variety of biomedical literature mining tasks. During the evaluation, we performed two literature mining tasks on two different datasets, which ensures that our framework can be easily expanded to other biomedical literature mining tasks. Therefore, we argue that the purpose of designing a scalable, flexible, efficient and affordable framework for biomedical literature mining is fulfilled.

The target users of the platform built on the framework include any biomedical researcher who works at discovering knowledge from a huge amount of biomedical literature. It relieves biomedical researchers from the burden of technical details on configuring computational infrastructure. With the platform, biomedical researchers focus on building text mining Spark applications and distributing them as Docker images. Spark applications in the format of Docker images are reusable modules that facilitate reproducible research.

There are some noted limitations in our study. First, not all 2.4 million full-text articles in the PMC OA Subset were used in our evaluation. Although there is no doubt that the framework can scale up to analyze all 2.4 million full-text articles, it is of great interest to see all articles being analyzed. Secondly, common biomedical literature mining tasks, such as entity recognition and relation extraction, are not validated in the evaluation. However, these tasks are crucial in knowledge discovery from biomedical literature. For instance, relation extraction helps advance precision medicine by discovering the association between gene and disease (Pletscher-Frankild et al., 2015), gene and drug (Cañada et al., 2017), and other associations. Future studies should include the implementation of these tasks. At last, the framework was introduced as cloud infrastructure independent, meaning it can seamlessly integrate with all major cloud service providers, including Amazon, Google, and Microsoft. However, only Google Cloud was tested in the evaluation.

5.7 Conclusion

In this study, we presented a big data framework for large-scale biomedical literature mining in the cloud. The scalability and cost-effectiveness are guaranteed by the cloud infrastructure upon which the framework was built. Besides, the storage and management components supported by Kubernetes and Docker make developing, managing and reproducing Spark based biomedical literature mining as easy as possible. Experimental results validate that our platform offers a scalable, flexible, efficient and affordable solution for large-scale biomedical literature mining.

In future work, we plan to further evaluate the platform by inviting more biomedical researchers to evaluate it and provide feedback. Then we will improve its usability based the feedback, We also would like to investigate the performance of our platform in analyzing literature or textual data from other domains.