Journal on Big Data
DOI:10.32604/jbd.2020.010090
Article

A Novel Framework for Biomedical Text Mining

Janyl Jumadinova[1], Oliver Bonham-Carter[1], Hanzhong Zheng[1,2,*], Michael Camara[1] and Dejie Shi[3]

[1]Department of Computer Science, Allegheny College, Meadville, PA 16335, USA
[2]Department of Computer Science, University of Pittsburgh, Pittsburgh, PA 15213, USA
[3]School of Computer and Information Engineering, Hunan University of Technology and Business, Changsha, 410205, China
[*]Corresponding Author: Hanzhong Zheng. Email: zpy_s7@163.com
Received: 15 June 2020; Accepted: 25 October 2020

Abstract: Text mining has emerged as an effective method of handling and extracting useful information from the exponentially growing biomedical literature and biomedical databases. We developed a novel biomedical text mining model implemented by a multi-agent system and distributed computing mechanism. Our distributed system, TextMed, comprises of several software agents, where each agent uses a reinforcement learning method to update the sentiment of relevant text from a particular set of research articles related to specific keywords. TextMed can also operate on different physical machines to expedite its knowledge extraction by utilizing a clustering technique. We collected the biomedical textual data from PubMed and then assigned to a multi-agent biomedical text mining system, where each agent directly communicates with each other collaboratively to determine the relevant information inside the textual data. Our experimental results indicate that TexMed parallels and distributes the learning process into individual agents and appropriately learn the sentiment score of specific keywords, and efficiently find connections in biomedical information through text mining paradigm.

Keywords: Biomedical text mining; reinforcement learning; multi-agent; distributed text mining; cluster

1 Introduction

The growth of published scientific articles has been expanding at an increasing rate over the last several years. As of 2018, the PubMed database contains over 28 million records, and between 1986 and 2010 the total number of citations in PubMed has been growing annually at a 4% rate [1]. Largescale and diverse scientific data comes with its advantages and disadvantages for the scientific community. On the one hand, big data is opening new unique research opportunities for building multidisciplinary research connections, for example in biological and physical sciences, engineering and business [2]. On the other hand, due to the explosive growth of biomedical publications, it has become difficult to find and keep track of relevant research discoveries and methods within biomedical research. Researchers within a specific subfield may not be aware of new discoveries relevant to their work in another related subfield. Additionally, poor communication creates a disconnection between highly specialized fields and subfields [3], and knowledge sharing between the disciplines can often be a challenge to require the background understanding. Thus, important connections between individual elements of biomedical knowledge, that could lead toward practical use in the forms of diagnosis, prevention and treatment, may be overlooked or not found. To address this challenge, we developed an intelligent and automated knowledge extraction system, TextMed, to efficiently mine articles and find connections in biomedical information. Due to a large amount of unstructured, high-dimensional information exists in the biomedical domain, biomedical

researchers have started using different techniques such as text mining techniques in nlp [4] to extract knowledge from research articles, case reports, Electronic Health Records (EHRs), etc. Biomedical text mining summarizes the textual data which allows researchers to identify key information much faster and discover relationships obscured by the large volume of available information that exists in the literature.

Our system is built on a multi-agent based framework with learning that processes textual information in a distributed way. Multi-agent systems usually utilize reinforcement learning algorithm [5] to let each agent make action, which is a reliable machine learning technique when lacking the existing labeled data. We initially tested our framework on the PubMed research articles related to muscular atrophy, Alzheimer's disease, and diabetes, which are commonly and risky diseases in the U.S. Our preliminary system [6] demonstrated that the developed multi-agent text mining framework is able to appropriately learn the sentiment score related to specific keywords by parallel analysis of the documents by multiple software agents. It operates on various machines (cluster nodes) and we tested its generalizability on a more diverse textual data. Our experimental result indicates that the multi-agent system utilizes the sentiment information as the learning process and can analyze assigned textual data from the given dataset. The further evaluation shows that our distributed platform provides with significant computation time reduction when software agents run on different clusters. Our results also demonstrate the learning capability of the technique on various textual biomedical data.

2 Related Work

Text mining is becoming an important approach that enables researchers and practitioners to extract information from large volumes of texts, articles and other written content. Previous text mining tools have concentrated on extracting information for human Textual mining has always been gaining the popularity among the researchers in the domain of the text summarization, information retrieval, topic modeling and key phrases identification. Accessing the document similarity is also a potential research domain that gaining the attention such text summarization, document retrieval, assessing document similarity (document clustering, keyphrase identification), and extracting structured information (entity extraction, information clustering) [7–8]. The increasing complexity and quantity of biomedical information during the recent years have triggered a need for text mining tools in biomedical research.

Biomedical text mining researches focus most on several directions. Biomedical name entity recognition (NER), within a collection of text, assigning name attributes to certain instances [3,9]. Other research using the discriminate model to classify the text documents into certain categories for specific interests [10]. Entity occurrence study is also another direction [11]. For example, in reference [12], the authors introduce a system called, TarMiner, that automatically extracts verified miRNA-to-gene interaction from the text contents of scientific publications. They empirically studied their text mining tool using real data and evaluated TarMiner using precision, recall and F-measure metrics. Compared to other automatic text mining tools that are used to extract miRNA targets, TarMiner can traverse the entire text, have a higher accuracy and support miRNA and gene name recognition for a large range of species. In another work, Kankar et al. [13] present a system named, MedSummarizer, that assigns biological meaning to a cluster of genes using information from biomedical literature. MedSummarizer is able to create a ranked list of important biological terms or concepts, which describe the gene cluster, and compute the conceptual similarities between each pair of genes and display this similarity in the visual graphical form. In reference [14], TextPresso, a web tool that allows users to search literature and annotate a scientific article. TextPresso provides a comprehensive literature search and annotation platform with customized features for optimal use. The authors have demonstrated the usefulness of the system by providing example searches and a case study to show how their tool can aid in a biological database curation. Similar to the overarching theme of these tools, our framework is able to extract information from scientific publications, find similarities and build relationships between keywords in the literature. However, our tool can be more generally applied to biomedical literature. It also utilizes a machine learning technique to improve its results as it performs sentiment analysis (SA) of the literature.

Text mining for biomedical applications can be complicated due to a continual growth of this type of

data, making the storage of this data very costly. Additionally, running any text mining algorithm on huge amounts of data may reduce the performance of the text mining method. To offset these problems, researchers either seek to perform text mining in a distributed computing fashion or store those large volumes of data into distributed databases. Balkir et al. [15] developed a method that incorporated the MapReduce programming model to efficiently access to database on parallel computers. Their method offers a multi-layered "loop-up" architecture that has been optimized for statistical parameter estimation model and allows to rapidly access model parameters. Through various experiments, it is shown that their approach performs favourable in terms of run-time, average latency per request, and scalability. Using a multi-agent system for text mining is also becoming more common among researchers. Chaimontree et al. [16] developed a multi-agent textual information mining framework that can deal with large amounts text data and provide a feasible secure environment. In their framework, each age is able to interact with each other and exchange the information. In their JADE (Java Agent Development Environment), five different types of agents have their own distinct purpose. The best clustering algorithm that uses the F-measure has been identified. Chao et al. [17] developed a decision making toll to help physicians make reliable diagnosis. Their multi-agent learning paradigm is Dia-MAS, which utilizes heterogeneous intelligent mining agents for finishing a specific and independent task. Our presented tool is built on a multi-agent framework, where agents run on separate clusters, thus utilizing the distributed platform to reduce computational time. To summarize, we present a multi-agent system using distributed computing for mining biomedical literature. Compared with previous work that uses multi-agent system framework for textual data mining, we explored the importance of keywords and their context information in the documents. Another contribution is that we also adopted the reinforcement learning algorithm to allow each agent better explore the appropriate relations through sentiment score calculation. We also built a distributed architecture that allows the system to finish information extraction on the extremely large data set within a comparatively short period of time. Finally, we present a completely automated workflow, starting with biomedical textural data collection, storage, analysis and final visual output.

3 Text Mining Distributed System

We designed and implemented an automated workflow as shown in Fig. 1. The workflow begins with data collection, data cleaning and initial data processing by the Lister tool [18], it then continues with learning and analysis of data from Lister by our multi-agent system. Our workflow finishes with the output generation in terms of visual graphics and data stored in a shared database. The agent learns the related emotions from the list of MeSH and the main keywords used by Lister. Multi-agent parallel and distributed analysis documents.

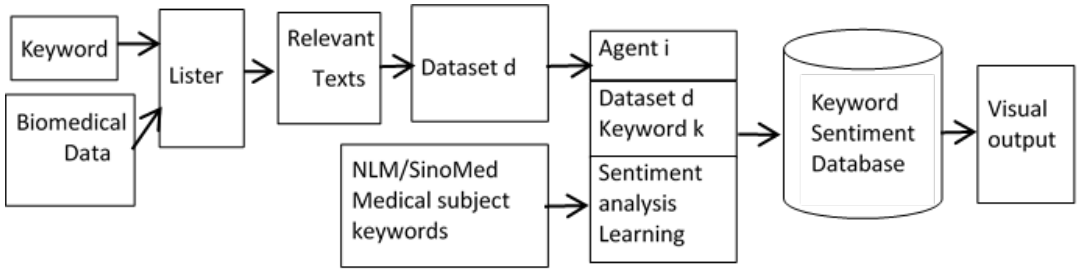


Figure 1: The overview of multi-agent learning system workflow

Here, the medical subject word list (MeSH) is created and maintained by the national library of medicine (NLM). NLM is the largest medical library in the world. It provides a variety of rich scientific and technological materials related to medicine for free. It is a rare and advantageous tool for scientific research. SinoMed is developed by the institute of medical information/library of the Chinese academy of medical sciences. It has abundant resources and can reflect the new progress of biomedical research at home and abroad.

3.1 Data Collection and Preprocessing

We utilize Lister tool automatically to download, parse and clean every article available through the PubMed database [19]. Lister uses a bag-of-words approach to parse articles with a given keywords, recording the abstract and article PMID number in which the keyword has been found. Then, Lister performs preprocessing and cleaning of the produced text by performing the standard text preprocessing pipeline including removing punctuations, whitespaces, stop words etc. This process can be repeated indefinitely for any number of unique keywords for a corpus of any size, generate relevant abstractions after executions.

At the end of the data collection and preprocessing by Lister, the data set D obtained from Lister is divided into a subset of data sets D_i. Each data subset is then assigned to and processed by one software agent i. Lister uses MeSH keywords as the primary keywords and associates the sentiment, which is secondary keyword. A list of MeSH keywords was created to conform to the MeSH (Medical Subject Headings) descriptors containing over 27,000 unique entries relevant specifically to the biomedical field in 2015 from U.S national Library of Medicine.

3.2 Learning and Analysis by Agents

After the subset of abstracts were created by the Lister, TextMed starts to create different software agents, which the workflow of each agent is illustrated in Fig. 2 and each agent follows the same process. Using the NLM Mesh keyword list, each agent starts to parse the assigned dataset. Each agent first determines whether a given keyword, or its pluralized form, appears in the given text at least once. We declared a variable named proximity that is used for keyword match. For example, the agent starts the sentiment analysis on the condition that whether the keyword appears within the text. If it does not appear in the text, then the agent will match the next keyword, then vice versa. The main purpose of the proximity is that the number of words in the context of the keywords will be included in the sentiment analysis because we assume that the certain left and right words around the keywords all play an important roles in understanding the keywords in the text. For example, if the keyword “diabetes” is found in the sentence fragment “connection between type 2 diabetes and Alzheimer disease,” then a proximity value of 1 will truncate the fragment to “type 2 diabetes and Alzheimer”, while a proximity value of 4 or greater will include the entire fragment. We iterate through all possible values of proximity, from 1 through the total word count of the abstract, performing sentiment analysis for each match at each possible proximity. The workflow of Agent i is shown in Fig. 2.

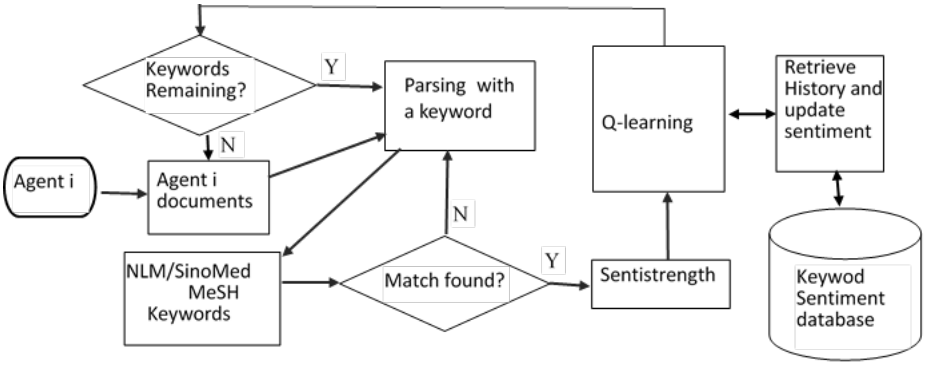


Figure 2: The workflow of an agent *i* in TextMed tool

Sentiment analysis can be expressed as SentiStrength. The input of SentiStrength is a text and the output is a positive/negative emotion with score values. The higher value indicates the stronger in positive/negative emotion. Similar to TaoBao, Twitter and YouTube comments, but we have adopted its capabilities, to create a composite score by combining the positive and negative scores obtained from SentiStrength.

Once agent *i* calculates the local sentiment of keyword *k* in the document, the global sentiment value, gs_k associated with the keyword *k* for all agents correspondingly will be adjusted. At the initial stage, as an agent found a specific keyword at the first time in one document, set $gs_k - ls_{k,d}$, However, if other agents have been already discovered the keywords before, then, the agent uses reinforcement learning algorithm to generate its new learned gs_k .

In multi-agent system, agents are action-based and receive a scaler reward from reinforcement learning after taking action. In our experiment, the global sentiment score for each keyword is the exploring environment, where agents interact with each other. The agents required to optimize reward by considering the short term and long term goals. The reward that agent received from each action is the historical state map. We selected reinforcement learning as a suitable learning paradigm for biomedical text mining because there is no sentimental analysis on keywords scope in biomedical research. Therefore, the reinforcement learning can validate the sentiment score under lacking of reference information.

3.3 Agent Uses Q-Learning Algorithm to Correlate Information

Q-learning is a model free learning paradigm that commonly uses in the design of a multi-agent system. The equation of the reward calculation is illustrated in Eq. (1), where reward r_k can be calculated according to Eq. (1), N is the number of files processed.

$$r_k = \sum_{i=d}^{N} \frac{|gs_k - ls_{k,d}|}{N} \quad (1)$$

This gives us the advances of taking short-term goal and long-term goal both into the consideration. For each keyword, we maintain a Q-table, a matrix, to represent each state-action pair as historical rewards. The initial Q-table set each index to be the largest, least favorable reward. With the local sentiment, the algorithm goes through all the positions of the table and search for the smallest, most favorable reward, and returns action required to take to transition to next state. The Q-table is also dynamic which means it will gradually update with the agents continue to making decisions. In essence, the whole process can be interpreted as a Markova decision process, in which the decision making is partially random, and partial under control.

Each agent objectively receives the optimal reward and the action should also be the optimal action by comparing all the reward values stored into the Q-table. This greedy strategy aims to force agents always to make the optimal decision by maximizing the total reward it received. However, one of the well-known challenges in reinforcement learning is the tradeoff between the exploration and exploitation, which agent tries to explore more information as possible. To address it, we introduce a stochastic mechanism that adds variations in the optimal action taking in range of [–1, 1] followed by Gaussian distribution. This gives better states to be research and recorded. The global leaned sentiment of this keyword is updated by summing the local sentiment with the current 'optimal' action. The updating procedure of the Q-table for the keyword is illustrated using the formation in Eq. (2). Here, $\alpha \in (0,1)$ is a learning rate and $\gamma \in (0,1)$ is a discount factor. The agent's local sentiment feedback and other global sentiment from other agents all incorporate into the calculation of the reward during the reward-based learning process for TextMed.

$$Q(ls, gs) = Q(ls, gs) + \alpha \cdot [r_k + \gamma \cdot \max Q(ls, gs) - Q(ls, gs)] \quad (2)$$

The concept of the utility U_i is the average reward when all keywords are found in a specific document retrieved by the agent after finishing parsing the document. The figation of U_i is calculated as Eq. (3), which the smaller utility values represent that more optimal rewards.

$$U_{i,d} = \sum_{k=1}^{numkeywords} r_{k,d} \quad (3)$$

Eventually, when all agents have completed parsing their assigned documents, the data collected from each keyword is exported to an SQL database. The data is divided into five different tables: primary keywords, secondary keywords, documentation, matching, and learning. This gives us more flexibility to identify correlations between different keywords across multiple documents.

3.4 Agents Computing on Distributed Clusters

We designed a cluster computing architecture in order to improve the processing and learning speed of software agents. The traditional approach of biomedical text mining requires all processes to occur on a single machine, which can easily generate high computational cost and system overhead. As for a multi-agent system satisfies the concept of distribution but the computation throughput is still limited to the resources of physical device such CPU and GPU. Distributed computing, including a cluster of computers, applies the high throughput technique that directly increases the overall system performance. TexMed utilizes a cluster that was built on a NFS (Network File System), allowing the resource sharing among machines in the system and operations such as "read" and "write" to the files that are stored on the NFS. Our cluster allows for each agent to be run on different machines or for a subset of agents to run on a single machine. This way, a larger amount of data can be processed by independently performing the text mining process on individual machines. Fig. 3 demonstrates our cluster set up and the details of its implementation are discussed below.

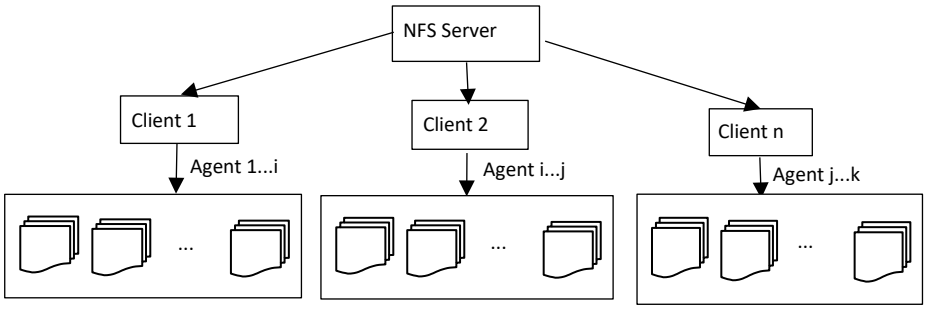


Figure 3: The cluster-based architecture for agents

The cluster was implemented by using the model of multi-clients (nodes) and a server. Clients can be joined to the registration group by adding their IP addresses to the group and establishing a TCP connection with the server using the IP addresses. A thread is then created on the server-side to manage each TCP connection between server and client. Using this approach, the server is able to communicate with all clients without any communication blocks. In our cluster set up, each client is assigned a set number of agents, that in turn are assigned a set number of documents to process and analyze, as discussed above. The NFS server assigns jobs to individual clients based on agents associated with clients and the total number of biomedical files stored in specific directory in NFS. When each client receives its job from the server, it immediately performs file fetching from NFS and simultaneously but independently runs software agent text mining on those files. Text mining processes are distributed among all cluster machines. Once agents for each client finish their text mining process, the data is automatically saved into the NFS.

4 Experiment Results

We tested the efficiency and accuracy of our system in an experimental study by recording and comparing the running time, the utility and reward values under different settings. We also conducted the hyper-parameter tuning with our system as the proximity parameter and number of agents, and we comprehensively evaluated the results concerting the output of the reinforcement learning algorithm including the correlation between the sentiment values and the reward values.

4.1 Data Sets

The dataset consists of six different sizes, shown in Tab. 1, consisting of research article abstracts that were obtained from the PubMed database. All six data sets were obtained by using the Lister tool, which first downloads and decompresses into NXML files from all the articles that are in PubMed data from March 2018. Lister then collects abstracts containing one of the keywords (primary keyword) from the downloaded articles. The keywords that we decided to use are muscular atrophy, Alzheimer's, aspirin,

diabetes, fever, and obesity separately to obtain abstracts containing these words, which were further converted into plain text format by Lister [20]. These keywords are commonly appeared into the PubMed dataset. Our system then ran on each of these data sets by specifying the number of cluster nodes and the number of agents to implement during execution. The TextMed stores the result into an SQL database, and then will be extracted and parsed into visual graphs and tables result to better visualization and easy for analysis. In particular, TexMed produces tables with running times, reward and utility graphs, as well as heatmap graphs. A heatmap has been commonly used in medical, clinical fields to illustrate the hierarchical relationships represented in two-dimensional colorful space. The numerical values have been clustered across the top and the side.

The heatmap conveys three distinct parameters as represented by the x-axis, y-axis, and the hue or intensity varies the color representation, respectively. Heatmaps are extremely useful to visualize the complex data and easy for interpreting the meaning.

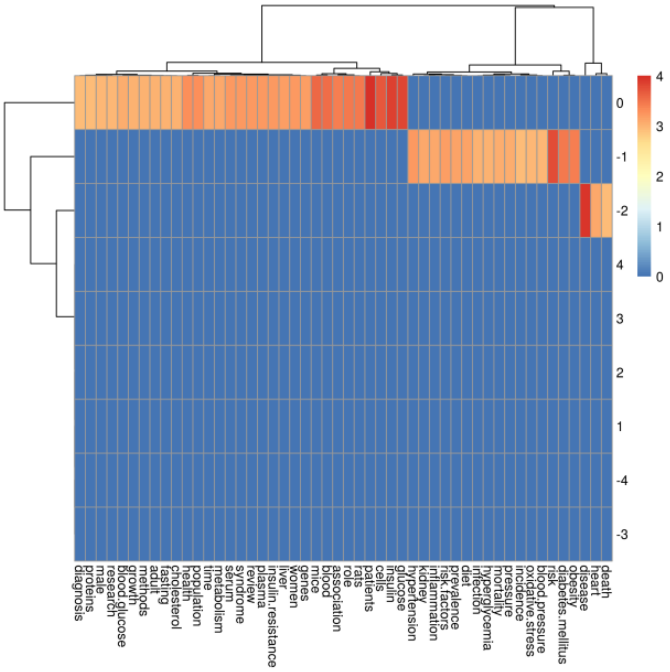
Table 1: The data sets used in our experiments

Data Set Primary Keyword	Number of Documents
Muscular Atrophy	400
Alzheimer's	2,700
Aspirin	6,000
Diabetes	10,000
Fever	30,000
Obesity	39,000

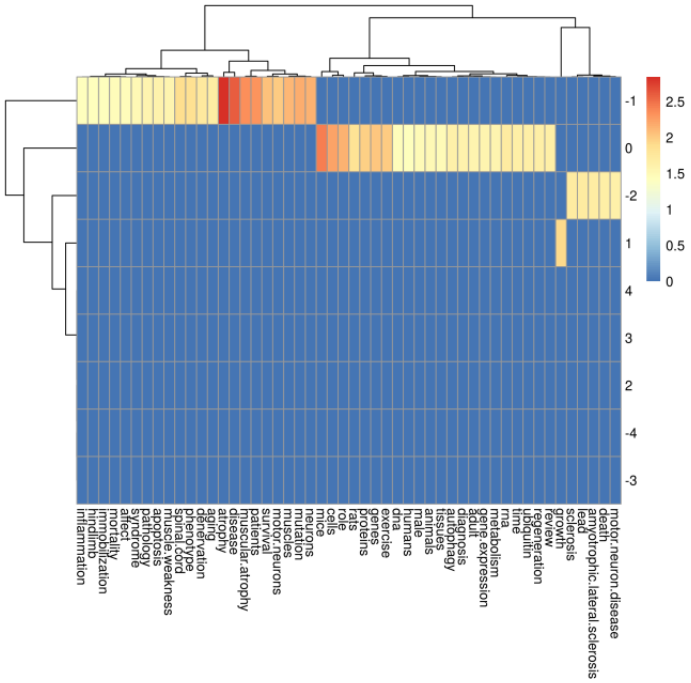
Fig. 4 shows two high-incidence search heatmaps about the Alzheimer's and Diabetes disease, which are widely common in the U.S. The development of Alzheimer's disease usually goes through three different stages: early, middle and severe. Many biomedical research institutions developed different research means of studying the progress. Another study direction of Alzheimer's depends on the PET scan or MRI. However, the number of available of PET scans or MRI are limited and usually suffers from site effect. Therefore, recently many researches have been made for studying the intensity normalization and harmonization. As for Diabetes, obesity is one of the influential factors and the U.S. is one of the countries that suffer from it. It describes each data set that we tested on with the number of 100 agents and the number of occurrences of the top 50 keywords associated with the corresponding sentiment values. The x-axis represents the top 50 keywords, indicating which of the MeSH keywords appeared most frequently in the documents that were parsed. The y-axis is the global learned sentiment of the entire multi-agent part. The scale is a log normalized representation of a global count. The warmer color in the graph represents the more keywords are matched and vice versa. Since one keyword can only be paired with one sentimental value, the dark blue color below a specific sentiment value shows that the reward is non-applicable.

As for the global count, higher counts allow the intervene of reinforcement learning to occur. Then, agents to more accurately adjust the global sentiment as more matches are found. The interactions among agents in a multi-agent system are very important for improve the overall system performance with respect to the information exchange. Thus, the heatmaps in Fig. 4 indicate an estimated reliability of the sentiment score for the keywords listed. Similarly, the warmer color in the shade cells represents that the given keyword is more reliable. In addition, they indicate that most of the frequently occurring keywords have negative or neutral sentiment values. We think this is due to the scope of that documents are collected and analyzed is limited because all of them are related to the biomedical field. When SentiStrength calculates evaluate the sentiment score of a text, words are inclined towards to bad or unfavorable could have high change of receiving large values. For example, sentimental words such as

worsen, death, deteriorated, infection. For diseases like Alzheimer's or muscular atrophy, the development progress can easily turn to be worse and can develop very fast as time goes on. It seems rational to expect low sentiment scores in general.



A. Alzheimer's disease



B. Diabetes

Figure 4: Two high-incidence search heatmaps

Sentiment score for a string of text, words that are generally considered bad or unfavourable are given high negative values. Such negative words might include pain, disease, or death, among many others. Given the propensity of these words appearing in texts concerning ailments like Alzheimer's disease or muscular atrophy, it seems rational to expect lower sentiment scores in general.

4.2 Running Time and Number of Agents

In our first set of experiments we analyze the efficiency of our clustering technique. Textmed was tested on a cluster of computers using a client and server communication model. We conducted experiments with 1, 2, 10, 20, and 30 nodes, where each node corresponds to a single physical machine. Each machine used in the experiments was 64-bit Dell OptiPlex 3010 with Intel(R) Core (TM) i3-3220 CPU @ 3.30GHz and with 4GB of memory. We also vary the number of software agents to run on each node from 10,50 and 100. The average running time over 100 separate runs of our system using different number of cluster nodes for Aspirin data set with 10 agents operating on each cluster node, is shown in Tab. 2. We note that the running time using more than one node is reduced by at least 30%. With the increase in the number of nodes, the running time is reduced further, but that reduction is not directly linear. We posit that the communication costs with the increase number of nodes influence the total running time.

Table 2: Running time of different number of cluster nodes

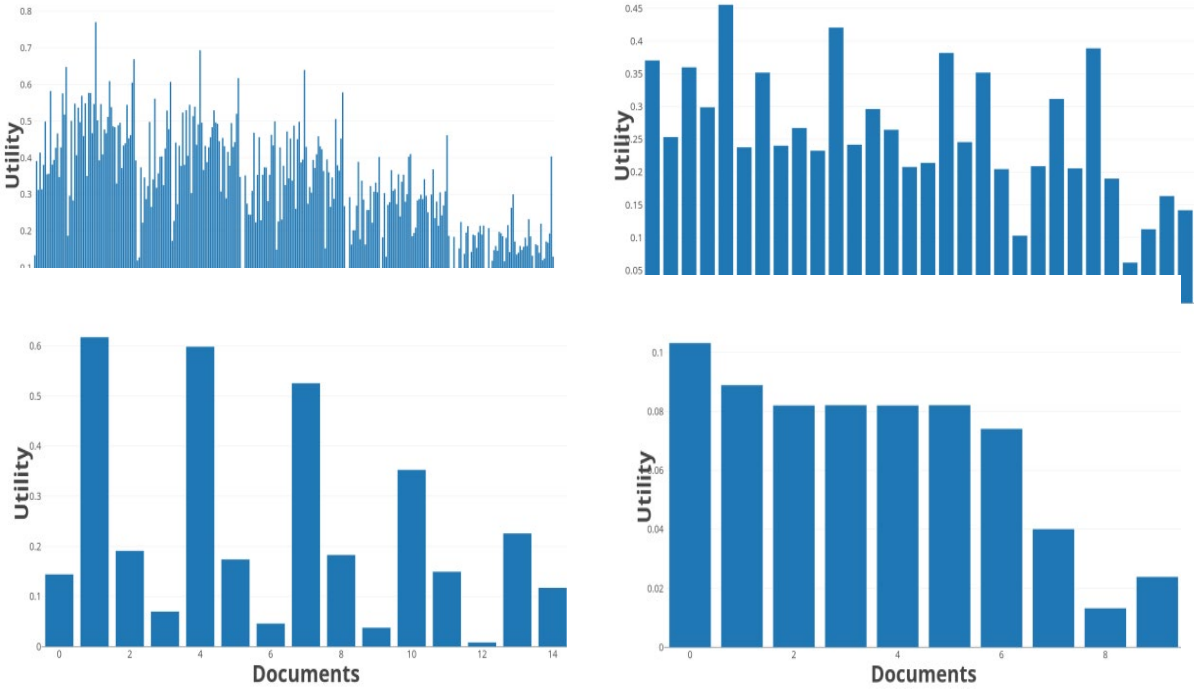
No. of nodes	Ave. Time (sec.)
1	512.8
2	340.7
10	129.8
20	80.7
30	52.8

Next, we analyzed what effect different number of agents has on the results of clustering. Fig. 5 shows the individual agent utility per document for a system with (A) 1, (B) 10, (C) 20, and (D) 30 cluster nodes with 10 agents operating on each node with proximity value of 10 for the Aspirin data set. We observed that with clusters and therefore agents, each agent being responsible for processing and analyzing fewer documents, there was less fluctuation in the agent's utility value.

5 Conclusion

We developed a novel framework, called TextMed, that consists of multiple software agents operating in a distributed platform to perform the texting and analyzation of the biomedical research articles related to the common diseases in U.S int an automatic and intelligent way. The entire dataset is divided into subset and assigned to software agents in our system. All of the agented collaborate together to learn the sentiment pertaining to specific keywords from the article abstract or common related to the diseases. We also employed a reinforcement learning technique to allow agents better correlate to calculate the sentiment score. Our experiment results indicate that the TextMed tool performs the parallel and distributed computing for textural data analysis and can Our automated workflow of data collection, cleaning, learning, analysis and visual and database output generation makes the task of text mining biomedical literature more accessible. In the future, we plan to study the different text preprocessing approaches such as using the Med7 name entity recognition tool, which can better help to identify the keywords and apply the similarity comparison for similar keyword merging or apply the deep

reinforcement learning method to continue to increase our current work. We also plan to release our developed tool to allow other researchers an opportunity to apply it and possibly expand it.



C: 20 nodes D: 30 nodes

Figure 5: Utility for different number of cluster nodes with 10 agents on each cluster, averaged over 100 runs

Funding Statement: This research is supported by Natural Science Foundation of Hunan Province (No. 2019JJ40145), Scientific Research Key Project of Hunan Education Department (No. 19A273), and open Fund of Key Laboratory of Hunan Province (2017TP1026).

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] Z. Lu, "Pubmed and beyond: A survey of web tools for searching biomedical literature," *Database the Journal of Biological Databases and Curation*, 2011.

[2] D. A. Reed and J. Dongarra, "Exascale computing and big data," *Communications of the ACM*, vol. 58, no. 7, pp. 56–68, 2015.

[3] J. T. Chang, H. Schutze and R. B. Altman, "Finding gene and protein names one word at a time," *Bioinformatics*, vol. 20, no. 2, pp. 216–225, 2004.

[4] G. H. Gonzalez, T. Tahsin, B. C. Goodale, A. C. Greene and C. S. Greene, "Recent advances and emerging applications in text and data mining for biomedical discovery," *Briefings in Bioinformatics*, vol. 17, no. 1, pp. 33–42, 2015.

[5] S. Abdallah and V. Lesser, "Multiagent reinforcement learning and self-organization in a network of agents," 2007. [Online]. Available: https://www.researchgate.net/publication/221454715.html.

[6] M. Camara, O. Bonham-Carter and J. Jumadinova, "A multi-agent system with reinforcement learning agents for biomedical text mining," 2015. [Online]. Available: https://www.researchgate.net/publication/280385970.html.

[7] I. H. Witten, "Text mining," *Practical Candbook of Internet Computing,* Boca Raton: CRC Press, 2004.

[8] A. S. Yeh, L. Hirschman and A. A. Morgan, "Evaluation of text data mining for database curation: lessons learned from the KDD challenge cup," *Bioinformatics*, vol. 19, no. Suppl. 1, pp. 1331–1339, 2003.

[9] G. Zhou, J. Zhang, J. Su, D. Shen and C. Tan, "Recognizing names in biomedical texts: a machine learning approach," *Bioinformatics*, vol. 20, no. 7, pp. 1178–1190, 2004.

[10] F. Liu, T. K. Jenssen, V. Nygaard, J. Sack and E. Hovig, "A figure legend indexing and classification system," *Bioinformatics*, vol. 20, no. 16, pp. 2880–2882, 2004.

[11] H. Yu and E. Agichtein, "Extracting synonymous gene and protein terms from biological literature," *Bioinformatics*, vol. 19, no. suppl. 1, pp. 1340–1349, 2003.

[12] R. M. Tsoupidi, I. Kanellos, T. Vergoulis, I. S. Vlachos, A. G. Hatzigeorgiou *et al.,* "TarMiner:automatic extraction of mirna targets from literature," in *the 27th Int. Conf. on Scientific and Statistical Database Management*, 2015.

[13] P. Kankar and S. Mukherjea, "Text-based summarization and visualization of gene clusters," in *the 2005 ACM Symposium on Applied Computing*, 2005.

[14] H. M. Muller, K. M. Van Auken, Y. Li and P. Sternberg, "Textpresso central: A customizable platform for searching, text mining, viewing, and curating biomedical literature," *BMC Bioinformatics*, vol. 19, no. 1, pp. 94, 2018.

[15] A. S. Balkir, I. Foster and A. Rzhetsky, "A distributed look-up architecture for text mining applications using mapreduce," in *High Performance Computing, Networking, Storage and Analysis*, IEEE, 2011.

[16] S. Chaimontree, K. Atkinson and F. Coenen, "Multi-agent based clustering: Towards generic multi-agent data mining," in *Advances in Data Mining: Applications and Theoretical Aspects*, Springer Berlin Heidelberg, 2010.

[17] S. Chao and F. Wong, "A multi-agent learning paradigm for medical data mining diagnostic workbench," in *Data Mining and Multi-agent Integration*, Springer US, 2009.

[18] O. Bonham-Carter and D. R. Bastola, "A text mining application for linking functionally stressed-proteins to their post-translational modifications," 2015. [Online]. Available:
https://ieeexplore.ieee.org/document/7359753.html

[19] N. PubMed, 2015. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed.html.

[20] U. N. L. of Medicine, Mesh keywords, 2015. [Online]. Available: http://www.nlm.nih.gov/mesh.html.