Journal of information and communication convergence engineering
http://jicce.org

J. Inf. Commun. Converg. Eng. 21(3): 244-251, Sep. 2023 Regular paper

Analysis of Impact Between Data Analysis Performance and Database

Kyoungju Min, Jeongyun Cho, Manho Jung, and Hyangbae Lee*

Department of Sino-Korean Literature, Chungnam National University, Daejeon 34134, Korea

Abstract

Engineering or humanities data are stored in databases and are often used for search services. While the latest deep-learning technologies, such like BART and BERT, are utilized for data analysis, humanities data still rely on traditional databases. Representative analysis methods include n-gram and lexical statistical extraction. However, when using a database, performance limitation is often imposed on the result calculations. This study presents an experimental process using MariaDB on a PC, which is easily accessible in a laboratory, to analyze the impact of the database on data analysis performance. The findings highlight the fact that the database becomes a bottleneck when analyzing large-scale text data, particularly over hundreds of thousands of records. To address this issue, a method was proposed to provide real-time humanities data analysis web services by leveraging the open source database, with a focus on the Seungjeongwon-Ilgy, one of the largest datasets in the humanities fields.

Index Terms: Data Analysis, Database Performance, Data Statistics, Data Visualization, Digital Humanities

I. INTRODUCTION

Recently, rapid process has been achieved in the analysis of humanities data using computers [1]. However, effective communication between humanities and computer science researchers is challenging; thus, computer engineers cannot easily understand minute differences unique to the humanities. Analysis of numerous ancient documents based on solely on the knowledge of humanities researchers is practically impossible, and relying solely on human memory has limitations. To address this, various institutions in South Korea, such as the Institute Translation of Korean Classics (ITKC), the Kyujanggak Institute of Korean Studies at Seoul National University (Kyujanggak, 奎章閣), the National Institute of Korean History(NIKH), and the Academy of Korean Studies(AKS), have digitize original texts, images, translations in modern Korean of state-compiled and personal records [2,3,4], including the Joseonwangjo-Silrok(Veritable Records of the Joseon Dynasty, 朝鮮王朝實錄)[5]. These digitized resources are available as search-oriented services.

Unfortunately, most of these services help only find a list of search terms entered by the user in the database. Thus, humanities researchers can only manually find and use the research materials. For example, currently, determining the individuals that appear most frequently in Joseonwangjo-Silrok or identifying the most commonly used vocabulary is impossible. This limitation arises from the fact that humanities researchers have not provides a means to actively utilize the data.

To extract diverse and meaningful statistical information from humanities or historical data [6], accurately analyzing the data is crucial, and the performance of such an analysis is vital for delivering it as an actual service. To date, real-time statistical services for extracting statistics from human-

Received 24 February 2023, **Revised** 8 June 2023, **Accepted** 26 June 2023
***Corresponding Author** Hyangbae Lee (E-mail: 5386yhbbb@hanmail.net, Tel: +82-42-821-5386)
Department of Sino-Korean Literature, Chungnam National University, Daejeon 34134, Korea

Open Access https://doi.org/10.56977/jicce.2023.21.3.244 print ISSN: 2234-8255 online ISSN: 2234-8883

ities data have not been employed owing to the unresolved issue of long database computation time. In this study, we crawled and built a database using the ITKC's Collection of Korean Classics Literature [7], Kyujanggak's Ilseongrok [8], and NIKH's Seungjeongwon-Ilgy (Daily Records of Royal Secretariat of Joseon Dynasty)[4]. Then, we analyzed and visualized the data using MariaDB on a standard PC, enabling humanities researchers to utilize it effectively.

Through experiments, this study demonstrates that various algorithms can become bottlenecks depending on the amount of data stored in the database, data size, and the number of accesses. We analyze the data using the database and developed a data-processing method to enable real-time services.

By replacing the parts where a database bottleneck occurs with memory operations, achieving real-time statistical extraction from large-scale datasets consisting of over 100 K entries, such as Seungjeongwon-Ilgy, is possible. This can contribute to the analysis of almost all historical documents in real-time using an open source database like MariaDB.

II. RELATED WORKS

A. N-gram Analysis

N-gram breaks down words, such as "NGRAM" into 'N', 'G', 'R', 'A', 'M', 'NG', 'GR', 'RA', 'AM', 'NGR', 'GRA', 'RAM', 'NGRA', 'GRAM', and 'NGRAM', and analyzes them. Although this is computationally intensive and inefficient as a method of extracting meaningful vocabulary by dividing it into syllables, it is one of the most reliable methods for lexical analysis. This is because it calculates all possible combinations. Compared to Hangeul, which is a phonetic alphabet, creating a vocabulary dictionary for ideograms is challenging, n-gram is a useful approach for creating a Chinese character dictionary. Research is currently being conducted to visually extract the lexical distribution of literary works written in Chinese characters using this method.

B. Google Books N-gram Viewer [9]

Using n-gram method, Google Books N-gram Viewer, a web service, visually represents the frequency of vocabulary found in books printed in English since 1800. Since the 2000s, its coverage has expanded beyond English to include books published in French, Russian, Chinese, German, Italian, and other languages. The users' desired vocabularies are visually displayed using a Google chart, making the observation of the usage trends of the respective vocabularies convenient.

C. Search Query Performance

The research [10] presents an analysis of the search query performance of open-source databases such as MySQL based on different types of datasets. To accurately compare the performance of search queries, the study distinguished between singleton, multi-condition, range, and join queries and compared their performance based on whether indexes were used. Join queries exhibited exponentially increasing search times with an increase in the number of data tuples, whereas the other methods show linearly increasing search times. The study also highlights the benefits of using indexes that affect the performance of database searches for query performance improvement.

However, this method used very small datasets and conducts measurements on datasets ranging 200 to half million entries, and the data size did not exceed 13.5 MB. A significant difference exists between this study and present research in terms of characteristics of the data, as this study did not address large datasets exceeding several GB with over 100 K entries.

Various methods have been studied to improve or compare the query performance of databases. Studies comparable to this study are briefly summarized in Table 1.

Table 1. Comparative analysis of related studies and this paper

	RDBMS	NoSQL	Hardware	# of Records	Data Size/Record	memory	Remarks
[10]	MariaDB, MySQL,..	N/A	N/A	200-50 K	Very Small	N/A	Performance comparison based on Index usage
[11]	Oracle, MySQL,..	Mongo, Redis,..	N/A	10-100 K	Unknown	N/A	NoSQL is more effective in large amounts of data
[12]	Derby, HSQL, H2		N/A	100K	Unknown	Yes	Comparison of Open-Source In-Memory DB
[13]	N/A	RocksDB	SSD, Flash	200M	1K	N/A	Storage technique control
This paper	MariaDB	N/A	SSD Compare	15K	Large	Partial	Minimization of DB operations and memory operation for performance

III. DATABASE IMPACT EXPERIMENTS

A. Experiment Environment

The experimental setup for analyzing the impact of the database on the data analysis is summarized in Table 2.

Table 2. Experimental environment

	Contents	Remarks
Language	PHP 7.4.27	
DB	Maria DB 10.4.22	XAMPP 7.4.1
Web Server	Apache 2.4.41	
CPU	AMD Ryzen 95900X	12-Core, 3.7GHz
RAM	64 GB	DDR4 3200MHz
OS	Windows 10	Enterprise Ed.
Storage	Samsung SSD 870 SanDisk Ultra 3D	NVMe for disk speed effect

The data used in the analysis experiment were obtained by crawling the original text data of Seungjeongwon-Ilgy, which was written in Chinese characters and provided by NIKH [4]. However, the first part of Seungjeongwon-Ilgy was lost owing to several wars. The remaining text extensively describes a total of 100 K daily government official records spanning 281 years, from 1623 to 1910.

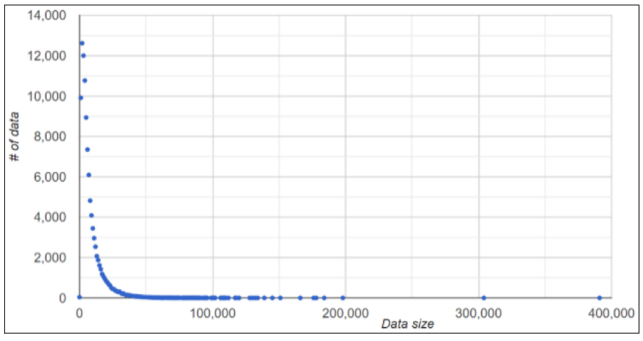

Fig. 1. Distribution of data size.

Fig. 1 shows the size distribution of the crawled data used in our experiment. The title of the book is Seungjeongwon-Ilgy (diary), and the average data size recorded per day is 7.4 KB, with the maximum size reaching 390 KB. A significant amount of data ranged from 29 to 40 KB. Note that designing a database schema, data exceeding 65 KB cannot be processed as a BLOB. This dataset was utilized to test the impact of MEDIUMBLOB or MEDIUMTEXT large-capacity data on the database performance.

B. Performance According to Query Types

The largest dataset, Seungjeongwon-Ilgy, contains records spanning more than 281 years and over 3,300 months, out of a total of over, 100 K data entries. If calculating the frequency of vocabulary occurrence by year, month, or day is necessary, many queries in a database are essential. In addition, determining a desirable method by first measuring the performance according to the query method and number of queries is essential. To this end, the performance was measured using four simple methods. The algorithm used is the same as that in Algorithm 1.

Algorithm 1. Test cases for query performance

```
// Case 1. Single Query
$sql = "select * from diary order by idx asc";
while($data = db_fetch($sql)) { doNothing( ); }

// Case 2. Multiple Query with Key
for( $i=1; $i<=$totalCount; $i++ )
    $sql = "select * from diary where idx='$i'";
    while($data = db_fetch($sql)) { doNothing( ); }

// Case 3. Multiple Query with LIMIT
for( $i=1; $i<=$totalCount; $i++ )
    $sql = "select * from diary order by idx limit $i, 1";
    while($data = db_fetch($sql)) { doNothing( ); }

// Case 4. Multiple Query with Paging Index
for( $i=1; $i<=$totalCount; $i++ )
    $sql = "select * from ( SUBQUERY limit $i, 1) ";
    while($data = db_fetch($sql)) { doNothing( ); }

drawPerformanceChart( );
```

The performance, as shown in Fig. 2, was measured for the same four test cases as Algorithm 1; Case 3 shows the time required to perform the operation in units. The experi-

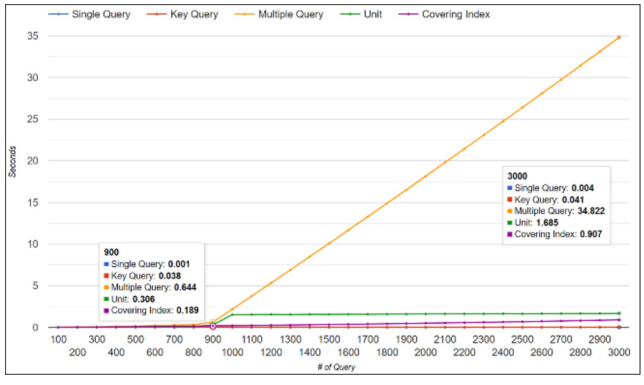

Fig. 2. Performance for Algorithm 1.

ments were conducted using 3 K pieces of data. Fig. 2 indicates the time measured for every 100 operations to enable an accurate performance evaluation.

As shown in Fig. 2, when measuring a query with the LIMIT option, a slight difference exists depending on the system situation; however, the operation time rapidly increases after a certain number of iterations. This phenomenon is commonly observed in bulletin boards containing a large number of posts, where data calculation processing at the backend significantly affects the overall performance when paging through content. According to the graph, algorithms such as Case 3 should be excluded from the time-series data statistical extraction for data distributions such as Seungjeongwon-Ilgy. As shown in Fig. 2, which presents the performance analysis for up to 3 K queries, the use of covering index [14] showed a remarkable improvement compared to Case 2. However, it is worth noting that it still requires 0.907 seconds for 3 K queries. Fig. 3 illustrates the accurate performance results obtained by applying the same algorithm to the entire dataset for case 1, 2 and 4 as the number of queries increases.

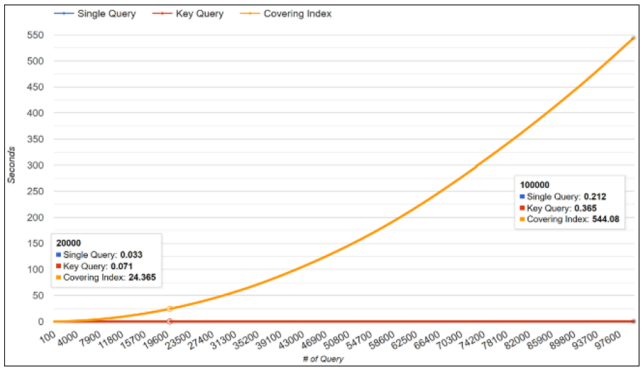

Fig. 3. Query performance across the full range.

As the number of queries increases, the measurement time using the covering index increases exponentially, taking 544 seconds (approximately 10 minutes) for 100 K queries in daily statistics. Although this method demonstrates performance improvement, it is not suitable for real-time time-series data analysis and thus should be excluded.

C. Performance of LIKE Option

For a time-series visualization of the frequency of, for example, the Admiral Lee Soon-shin or Turtle Ship on a graph, a search system or bulletin board attempts to search a database using a LIKE query. The performance of the database was measured by performing a search using time-series data.

Algorithm 2. Search performance with LIKE option

```
$keywordList = "Lee Soon-sin, Jung Yak-yong,……";
$keys = explode(",", $keywordList);

for($i = 1; $i <= $NUM; $i++)
    $sql = "select * from diary where
                    (content like '%$keys[1]%')
                    or (content like '%$keys[2]%')
                    or …… ";
    while($data = db_fetch($sql)) { doNothing( ); }

drawPerformanceChart( );
```

A measurement algorithm, similar to Algorithm 2, was utilized and actual search terms were historical figures. Fig. 4 illustrates the performance of a single query based on the number of search terms using the LIKE query option for extensive content a large dataset.

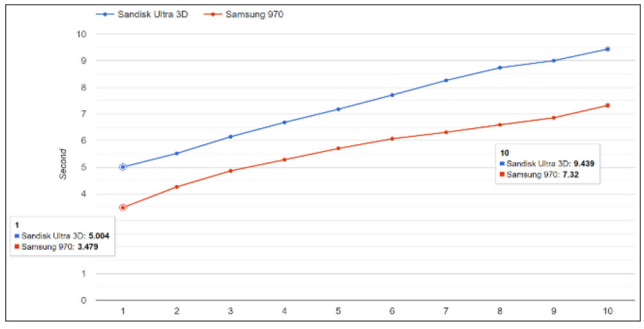

Fig. 4. Performance according to the number of keywords using the LIKE option.

According to the result of the lower graph in Fig. 4, the time required for a single query increases linearly with the number of search terms. Even for single search term, it takes more than 3 seconds to find a result. This indicates that the use of the LIKE query option leads to a full-scan issue, making it unsuitable for real-time service development.

D. Effect of Disk Speed

A database is a special purpose file system designed to store information on a disk. When analyzing large amounts of data, it is important to consider the impact of disk speed. Fig. 4 presents a graph comparing the performance of two disks with different speeds, conducted in the same environment, to examine this effect. The experiment was conducted using different Samsung and Sandisk NVMe disks. The results in Fig. 4 shows that a disk has a significant influence on performance. Table 3 provides hardware information from both manufacturers, including disk speed.

Table 3. Product speed comparison

Model \ Speed	SequentialRead	SequentialWrite
Samsung 970 (1 TB)	3400 MB/s	2500 MB/s
Sandisk Ultra 3D (1 TB)	2400 MB/s	1950 MB/s
Rate (%)	42	28

A search query involves reading data from disk. According to Table 3, the performance for sequential reading was 42% better than that of Ultra 3D. Additionally, the results from Fig. 4 show an average performance difference of 30.2% among the 10 experimental trials, even when considering the experimental errors. This indicates that seek times can be reduced proportionally by using a faster disk.

Based on the preliminary experimental results tailored to the data characteristics, several meaningful conclusions can be drawn. First, using the LIMIT option when querying the database leads to a deterioration in performance. Second, the covering index significantly improves the performance with the LIMIT option, but its effectiveness decreases when running multiple queries repeatedly. Third, LIKE searches negatively impact performance. Fourth, when analyzing large amounts of data, it is advisable to reduce the number of queries or directly access key values. In other words, improper database access becomes a bottleneck.

IV. APPLYING TO DATA ANALYSIS

A. Analysis of Person and Places

About 400 K people are mentioned 7.3 million times in the Seungjeongwon-Ilgy, which was collected through crawling [15]. They appear in over 50 K places and are mentioned 1.3 million times as well. In the crawled data, person and place information is tagged in XML, and the collected information is divided into field using sharps (#) and organized by date. For example, “Lee Soon-sin#Song Si-yeol#Lee Hwang”. The algorithms presented below represent the cases where the characteristics of the database discussed in Chapter 3, are considered and the cases where they are not considered.

Logically, in Case 1 of Algorithm 3, if (name, total) exists in the PERSONS table, it iteratively compares this table to determine whether to insert or update. On the other hand, Case 2 computes an associative array, personsArray, from all the data in memory, and then inserts it into a database or converts it to JSON (JavaScript Object Notation) for later use. In the latter case, the query is executed only once for retrieval. The performance graph for extracting 30 K data using Case 1 is labeled as ‘Measurement’ in Fig. 5.

As seen from the results of extracting a portion of the complete person data, Case 1, which relies on the database,

Algorithm 3. People and place extraction

```
// Case 1. Extraction by DB only
$sql = “select people from diary order by idx”;
while($data = db_fetch($sql))
    $persons = explode(“#”, $data[people]);
    while($persons[$num])
        $existSql = “select if exist $persons[$num]”;
        if( existsPerson )
            updateFrequencyCount( existsPerson ); // ①
        else
            insertFrequencyByOne( $persons[$num] ); // ②
// Case 2. Extraction for performance
$sql = “select people from diary order by idx”;
while($data = db_fetch($sql))
    $persons = explode(“#”, $data[people]);
    while($persons[$num])
        if( existsInArray )
            personsArray[$persons[$num]] ++;
        else
            personsArray[$persons[$num]] = 1;
insertDBorMakeJSON(personsArray); // if needed
```

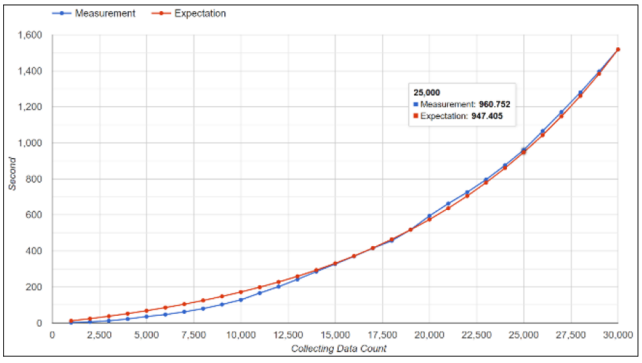

Fig. 5. Database-oriented person information extraction performance and expectation.

exponentially increases the time, making it impossible to extract the entire data. On the other hand, extracting data from memory after querying the database once requires 1.045 seconds to extract a total of 390 K person information and 0.232 seconds to create a JSON file. The size of the generated JSON file is 23.5 MB, which is significantly smaller compared to the sequential write performance in Table 2, thus not becoming a bottleneck.

Meanwhile, ‘Expectation’ in Fig. 5 represents a graph of the estimated time required to extract a total of 390 K data using Equation 1. By adjusting the value of ‘a’ in Equation ③, an experiment was conducted, resulting in almost infinite time required since the x-axis value is 390 K. In other words, it reaches an incalculable level. Note that this does not imply an exponential increase in calculation speed. Among ① and ② in Algorithm 3, the newly added speed is calculated in ② . A total of over 7 million queries are exe-

cuted, which can be improved by utilized, and it represents the outcome of the case 1 procedure in Algorithm 3.

Equation 1. Calculating time expectation

$$y = a \times (b^x - 1) \quad \cdots\cdots ①$$

$$y_{last} = a \times (b^{last} - 1) \quad \cdots\cdots ②$$

$$b^{last} = \frac{y_{last} + a}{a}$$

$$b = \left(\frac{y_{last} + a}{a}\right)^{\frac{1}{last}}$$

$$y_{expect} = a \times \left[\left\{\left(\frac{y_{last} + a}{a}\right)^{\frac{1}{last}}\right\}^x - 1\right] \quad \cdots\cdots ③$$

If the results are stored using a large-scale associative array, such as Case 2 of Algorithm 3 mentioned above, it can be resolved with only tens of MB of memory.

B. N-gram Analysis

To analyze the vocabulary for all possible cases, such as brute-force attacks, an n-gram analysis was conducted. If the analysis is performed regardless of performance, the utilization is very high, but it requires a significant amount of computation and memory. When analyzing the Seungjeongwon-Ilgy using n-gram, at least 0.8 billion comparison operations were performed, resulting in the creation of 150 million data. As discussed in Chapter 3, it is not feasible to generate results with such a large number of database queries.

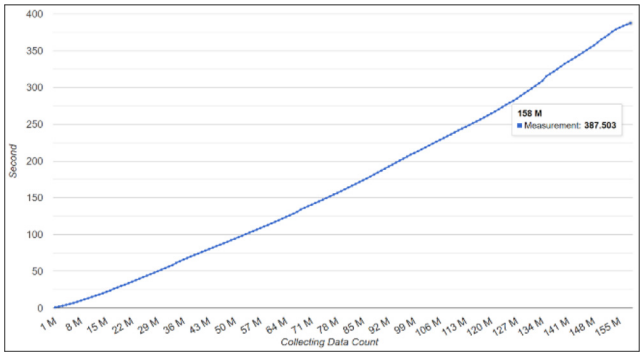

Fig. 6. Performance of data extraction by n-gram.

Fig. 6 shows the n-gram extraction performance results of the entire Seungjeongwon-Ilgy dataset. A total of 150 million n-gram data can be extracted in 387 seconds. Table 4 summarizes the 1-3 syllables with the highest frequency in Seungjeongwon-Ilgy through this n-gram analysis.

However, saving 150 million data as JSON files, reducing memory calculation time, or visualizing lexical relationships in real-time are areas of future research.

Table 4. List of high-frequency syllables extracted through N-gram analysis (Original text)

1 Syllable Words		2 Syllable Words		3 Syllable Words	
之	6.5M	上曰	810K	言啓曰	251K
曰	4.1M	傳曰	772K	副承旨	161K
以	3.0M	啓曰	768K	牌不進	101K
不	2.7M	何如	571K	假注書	100K
爲	2.5M	承旨	448K	大妃殿	98K

C. Visualization of Time Series Data

Historical records are closely related to the passage of time. In particular, chronology, which involves recording events based on dates, such as Seungjeongwon-Ilgy or Joseongwangjo-Silrok, proves highly effective in deriving meaning when presented as time series [14,15] data. Although records prior to 1623 were lost due to the influence of the several wars, Seungjeongwon-Ilgy is vast record of 281 years from 1623 to 1910. By minimizing database access as previously described and utilizing an associative array, it becomes feasible to extract yearly, monthly, and daily search term statistics from this 281-year collection of records.

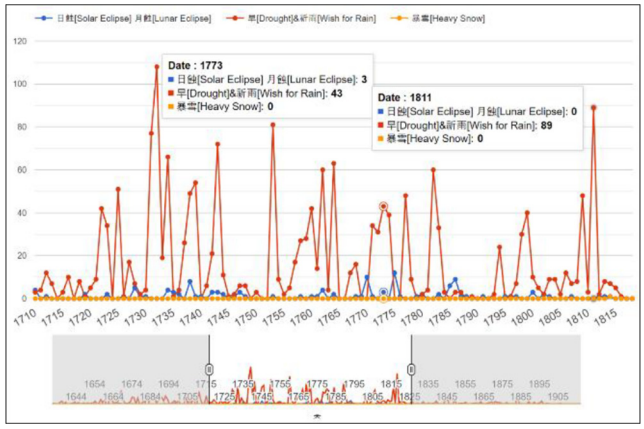

Fig. 7. Time series data statistics yearly graph.

Fig. 7 displays the frequency of vocabulary appearance by year for the search words entered by the user. Square bracket ([ ]) are symbols used for explanation and are shown only in the graph, not in the retrieval. ① Solar eclipse or lunar eclipse (OR Search), ② Drought and rain (AND Search), ③ Heavy snowfall frequency are calculated and compared simultaneously (Multiple Search, separated by commas), and they represent annual statistics. For user convenience, a section setter called controller or brush can be used at the bottom to allow the user to adjust the settings for detailed observation.

Similarly, Fig. 8 depicts the visualization of daily statistical information as time series data. If a user identifies an outlier in Fig. 7, they would require date-specific statistics

for detailed verification. However, a challenge arises when attempting to view data by date across all 100 K sections. This issue pertains not to the search time but rather to the process of data visualization. Previous studies [16,17] have indicated that problems can arise when dealing with 20 K or more data points. Therefore, users should narrow down their search range for daily statistics. The demonstrated example of statistical visualization of time series data can extract and visualize statistical information within 3 seconds under the conditions outlined in Table 2.

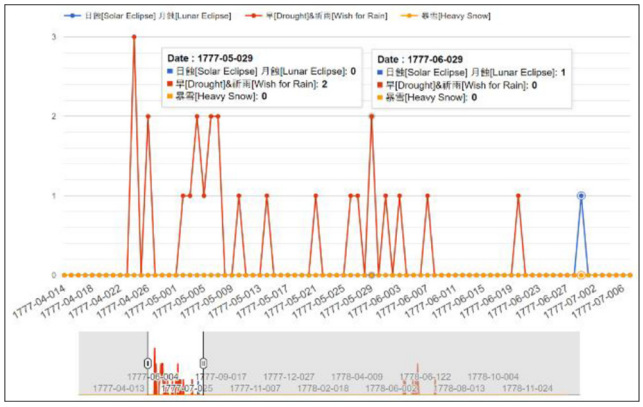

Fig. 8. Time series data statistics daily graph.

The approaches we have examined so far differ from the conventional methods of improving search speed using techniques such as indexing, covering indexes, and query optimization. Through experiments, we confirmed that in the case of large data where the data size of a single record is significant, the database becomes a bottleneck for real-time statistical extraction. To address this, we minimize database access and perform computations in memory to enhance performance. This combines the advantages of in-memory databases with the existing database environment, enabling real-time processing of large-scale data retrieval.

V. CONCLUSIONS AND FUTURE WORKS

In Korea, as the academic world's concerns that a serious decline in the birth rate will act as a fatal blow to humanities research have become reality, it is moving in a serious direction, such as reducing the number of students. As a way to respond to this, digital humanities have emerged, and attempts are being made to overcome the categories that rely on the memories of existing researchers by using computers. However, there is a large gap between engineering and the humanities, which hinders research. In order to analyze many ancient documents by computer, database-based research should be prioritized rather than the latest technologies such as BART and BERT. However, unlike the expectation that the size and amount of humanities data will be very small compared to big data, the target of analysis is big and vast. Therefore, the desired analysis results cannot be obtained without an accurate understanding of the database.

A misunderstanding of databases has been criticized for being inefficient when using n-gram for lexical analysis. In addition, due to the nature of documents dealing with historical information, time series data analysis is required, and a representative service for studying time series vocabulary statistics is Google's N-gram Viewer.

In this study, the performance was measured for the Seungjeongwon-Ilgy composed of over 100 K and the data size of MEDUUMTEXT using algorithms and hardware characteristics that programmers can generally consider. Through this, it was shown by experiments that the number of database queries and the covering index cause performance degradation as the amount of data increases. In addition, it was shown that database access becomes a serious bottleneck depending on the characteristics of the data. In order to solve this problem, it was shown that it can be applied to time series data statistics and a method of extracting more than 150 million n-gram data using database and associative array. Through this, it can be used for similar types of data analysis including humanities data analysis in the future by showing the relationship of database's influence on data analysis. So this demonstrated that minimizing database access and increasing memory computations are necessary to extract statistics in real-time from large-scale humanities data.

Future research is needed to effectively store the data of over 150 million n-gram in JSON or database. The Seungjeongwon-Ilgy is one of the largest humanities data in Korea. It is necessary to apply the analysis method of the Seungjeongwon-Ilgy to analyze and visualize many old document written in Chinese characters in real time.

REFERENCES

[1] R. Ma, "Boundaries, extensions, and challenges of visualization for humanities data: Reflections on three cases," in *IEEE 7th Workshop on Visualization for the Digital Humanities*, Oklahoma City, USA, pp. 1-5, 2022. DOI: 10.1109/VIS4DH57440.2022.00006.

[2] Korean Classics DB, Jan. 2023, [Internet] Available: https://db.itkc.or.kr/.

[3] Kyujanggak Text Search System, Jan. 2023, [Internet] Available: https://kyudb.snu.ac.kr.

[4] The Daily Records of Royal Secretariat of Joseon Dynasty, Jan. 2023, [Internet] Available: https://sjw.history.go.kr/.

[5] The Veritable Records of the Joseon Dynasty, Feb. 2023, [Internet] Available: https://sillok.history.go.kr/.

[6] S. Kessler and C. Rothen, "Pro-amateur information space: www.bildungsgeschichte.ch," *Digital Turn und Historische Bildungsforschung*. pp. 113-125, 2022. DOI: 10.35468/5952-08.

[7] Collection of Korean Classics Literatures, Feb. 2023, [Internet] Available: https://db.itkc.or.kr/dir/item?itemId=MO#/dir/list?itemId

=MO.
[8] Ilseong-rok Original Text and Image Search System, Feb. 2023, [Internet] Available: https://kyudb.snu.ac.kr/series/main.do?item_cd=ILS.
[9] Google Books Ngram Viewer, Feb. 2023, [Internet] Available: https://books.google.com/ngrams/.
[10] M. K. Min, "Experiments of search query performance for SQL-based open source databases," *International Journal of Internet, Broadcasting and Communication*, vol. 10, no. 2, pp. 31-38, May 2018. DOI:10.7236/IJIBC.2018.10.2.6.
[11] R. Čerešňák and M. Kvet, "Comparison of query performance in relational a non-relation databases," in *13th International Scientific Conference on Sustainable (TRANSCOM 2019)*, Novy Smokovec, Slovak Republic, pp. 170-177, 2019. DOI: 10.1016/j.trpro.2019.07.027.
[12] M. K. Park, "A Study on Application using and performance comparison of in-memory database," *2016 Master's thesis, Soongsil University*, 2016.
[13] F. Yang, K. Dou, S. Chen, M. hou, J. U Kang, and S. Y. Cho, "Optimizing NoSQL DB on flash: A case study of RocksDB," in *2015 IEEE 12th International Conference on Ubiquitous Intelligence and Computing (UIC-ATC-ScalCom)*, Beijing, China, pp. 1062-1069, 2015. DOI: 10.1109/UIC-ATC-ScalCom-CBDCom-IoP.2015.197.
[14] R. Kaushik, P. Bohannon, J. F. Naughton, and, H. F. Korth, "Covering indexes for branching path queries," in *Proceedings of the 2002 ACM SIGMOD international conference on Management of data*, Madison, USA, pp. 133-144, 2002. DOI: 10.1145/564691.564707.
[15] K. J. Min and B. C. Lee, "The analysis of Chosun dynasty poetry using 3D data visualization," *Journal of the Korea Institute of Information and Communication Engineering*, vol. 25, no. 7, pp. 861-868, Jul. 2021. DOI:10.6109/jkiice.2021.25.7.861.
[16] K. J. Min, B. C. Jin, and M. H. Jung, "Massive graph expression and shortest path search in interpersonal relationship network," *Journal of the Korea Institute of Information and Communication Engineering*, vol. 26, no. 4, pp. 624-632, Apr. 2022. DOI: 10.6109/jkiice.2022.26.4.624.
[17] K. J. Min, J. Y. Cho, M. H. Jung, and H. B. Lee, "Optimization for large-scale n-ary family tree visualization," *Journal of information and communication convergence engineering*, vol. 21, no. 1, pp. 54-61, Mar. 2023. DOI: 10.56977/jicce.2023.21.1.54.
[18] J. V. D. Donckt, J. V. D. Donckt, E. Deprost, and S. V. Hoecke, "Plotly-resampler: Effective visual analytics for large time series," in *2022 IEEE Visualization and Visual Analytics*, Oklahoma City, USA, pp. 21-25, 2022. DOI:10.6109/jkiice.2022.26.4.624.
[19] M. Aminazadeh and F. Noorbehbahani, "City intersection clustering and analysis based on traffic time series," in *12th International Conference on Computer and Knowledge Engineering*, Mashhad, Iran, pp. 274-281, 2022. DOI: 10.1109/ICCKE57176.2022.9960065.

Kyoungju Min
received a B.S. degree in 2000 and M.S. degree in 2002 from the Department of Computer Engineering in Chungnam National University, Korea. He is currently a Ph.D. student at the Department of Sino-Korean Literature of Chungnam National University. He has been a CEO of Eureka Solution since 2008. He is also active as an IT professional instructor. His current research interests include digital humanities, data analysis, data visualization, and network security.

Jeongyun Cho
received a B.S. degree in 1996 from the Department of Sino-Korean Literature in Chungnam National University, Korea, an M.S. degree in 1998 from the Department of Korean Language and Literature in Chungnam National University, and a Ph.D. in 2009 from the Department of Korean Language and Literature in Korea University, Korea. Since 2001, she worked as a lecturer in the Department of Sino-Korean Literature at Chungnam National University, where she now works as an assistant professor. Her research interests include Sino-Korean prose, Sino-Korean literature criticism, and digital humanities.

Manho Jung
received a B.S. degree in 1997, an M.S. degree in 1999 from the Department of Sino-Korean Literature in Chungnam National University, Korea, and a Ph.D. in 2005 from the Department of Sino-Korean Literature in Dankook University, Korea. Since 2000, he has worked as a lecture and research professor in the Department of Sino-Korean Literature at Chungnam National University, where he has worked as an associative professor since 2019. His research interests include Sino-Korean grammar, artificial intelligence classics translation and digital humanities.

Hyangbae Lee
received a B.S. degree in 1994 from the Department of Sino-Korean Literature of Chungnam National University, Korea, an M.S. degree in 1996 from the Department of Korean Language and Literature in Chungnam national University, and a Ph.D. in 2000 from the Department of Sino-Korean Literature in Dankook University, Korea. He has worked as professor in Department of Sino-Korean Literature at Chungnam National University since 2001. His research interests include Sino-Korean criticism and digital humanities.