East Asian Math. J.
Vol. 40 (2024), No. 3, pp. 307–318
http://dx.doi.org/10.7858/eamj.2024.020

KOREAN TOPIC MODELING USING MATRIX DECOMPOSITION

June-Ho Lee and Hyun-Min Kim*

Abstract. This paper explores the application of matrix factorization, specifically CUR decomposition, in the clustering of Korean language documents by topic. It addresses the unique challenges of Natural Language Processing (NLP) in dealing with the Korean language's distinctive features, such as agglutinative words and morphological ambiguity. The study compares the effectiveness of Latent Semantic Analysis (LSA) using CUR decomposition with the classical Singular Value Decomposition (SVD) method in the context of Korean text. Experiments are conducted using Korean Wikipedia documents and newspaper data, providing insight into the accuracy and efficiency of these techniques. The findings demonstrate the potential of CUR decomposition to improve the accuracy of document clustering in Korean, offering a valuable approach to text mining and information retrieval in agglutinative languages.

1. Introduction

Natural language is the language that humans use on a daily basis. And natural language processing(NLP) is an area of research and application that explores how computers can be used to understand and manipulate natural language text or speech to do useful things. NLP researchers aim to gather knowledge on how human beings understand and use language so that appropriate tools and techniques can be developed to make computer systems understand and manipulate natural languages to perform desired tasks [4].

During several decades, NLP evolved significantly with the advances in computing technology and the changes to define linguistic regularity with statistics. NLP is used in fields such as information retrieval, information extraction, question answering, summarizing, machine translation, and dialogue systems [11].

As reducing computation time and increasing accuracy, the biggest interest in NLP is the pre-trained language model(PLM). PLM is a method that uses large

Received January 29, 2024; Accepted April 23, 2024.
2020 *Mathematics Subject Classification.* 65F55,68T50.
Key words and phrases. Singular value decomposition, Cur decomposition, Natural language processing.
* Corresponding author.

(pISSN 1226-6973, eISSN 2287-2833)

text data to train deep learning algorithms before solving language problems. Typical PLM models are Elmo, Bert, Roberta, and T5 [16, 6, 12, 17].

In [7], S. Gururangan et al. find that the use of data focused on a specific area is better than using broad-coverage data for reducing computational time and increase accuracy. For example, when solving financial problems, it is better to train a model with only financial-related documents than to use the entire document. This method is called domain adaptive pre-training(DAPT) [7]. Many applications are using DAPT. For instance, there are the economic analysis model FinBERT [2], SciBERT [3] based on natural science academic data, BioBERT [9] based on biomedical data, Clinical BERT [1] based on medical clinical data. In Korea, there are KB-ALBERT [8], a Korean financial model created by KB KOOKMIN Bank, and SKT-KoBERT [18] created by SKT. Both models are utilized for counseling chat bot.

Now our problem is how to select documents in a specific area from the entire document. In this paper, we propose a method to extract topics from specific fields from Korean language documents in order to use the DAPT method.

One of the classical methods of clustering documents by topic is Latent Semantic Analysis(LSA) in NLP. LSA is a basic statistically-based method to consider the frequency of terms in the document and the relation among terms. The process involves creating a matrix from the documents that contain the topic we want to cluster, the terms in all the documents, and the frequency of these terms. This matrix is called a term-document matrix(TDM). Using singular value decomposition(SVD), we apply dimensionality reduction to identify latent meaningful structures [5]. Furthermore, we propose to use CUR decomposition(CUR) as a dimensionality reduction method for TDM in LSA. CUR decomposition extracts C and R from the actual data matrix and uses U to approximate the original matrix accurately.

This approach has the advantage of allowing for interpretable data analysis that reflects the original matrix. And we compare the accuracy of the two methods through experiments [14]. This paper is organized as follows. In section 2, we introduce the mathematical definitions and characteristics of SVD and CUR decomposition used in document clustering models. We also explain the characteristics of the Korean language in natural language processing and how we create the TDM for LSA. In section 3, we explains the steps that take to classify documents using the methods discussed in section 2. In section 4, we discusses the characteristics of SVD and CUR decomposition and presents the outcomes of two experiments. One experiment involves a comparison of the characteristics of each method using artificially created data. The second experiment focuses on LSA conducted with real text data.

2. Preliminaries

2.1. Matrix Decomposition

In this part, we introduce the SVD. SVD is a kind of matrix decomposition. In addition, it is very important in data science in that it can represent high dimensions in low dimensions [20].

Definition 2.1. [20](Singular value decomposition)
Let m, n be arbitrary. Given $A \in \mathbb{C}^{m\times n}$ a singular value decomposition (SVD) of A is factorization

$$A = U\Sigma V^\top$$

where $U \in \mathbb{C}^{m\times m}$, $V \in \mathbb{C}^{n\times n}$ are unitary, $\Sigma \in \mathbb{R}^{m\times n}$ is diagonal matrix. In addition, it is assumed that the diagonal entries σ_i of Σ are nonnegative and in nonincreasing order; that is, $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_p \geq 0$, where $p = \min(m, n)$.

Theorem 2.2. [20]*(Low-Rank Approximation)*
For any v with $0 \leq v \leq r$, define

$$A_v = \sum_{j=1}^{v} \sigma_j u_j v_j^\top$$

where, $u_j(v_j)$ are the left(right) singular vectors. if $v = p = \min\{m, n\}$, define $\sigma_{v+1} = 0$. Then

$$\|A - A_v\|_2 = \inf_{\substack{B\in\mathbb{C}^{m\times n}\\ \operatorname{rank}(B)\leq v}} \|A - B\|_2 = \sigma_{v+1}.$$

Theorem 2.3. [20] *For any v with $0 \leq v \leq r$, the matrix A_v of Theorem 2.2 also satisfies*

$$\|A - A_v\|_F = \inf_{\substack{B\in\mathbb{C}^{m\times n}\\ \operatorname{rank}(B)\leq v}} \|A - B\|_F = \sqrt{\sigma_{v+1}^2 + \cdots + \sigma_r^2}.$$

In this part, we introduce the CUR decomposition. Like SVD, CUR decompositions are low-rank matrix decompositions that are explicitly expressed in terms of a small number of actual columns or actual rows of the original matrix [14].

Definition 2.4. [15] (CUR Decomposition)
Let A be $m \times n$ matrix. In addition, let C be an $m \times c$ matrix whose columns consist of a small number r of columns of the original matrix A, and let R be an $c \times n$ matrix whose rows consist of a small number r of rows of the original matrix A, a CUR decomposition approximation to A if may be explicitly written as

$$\tilde{A} = CUR.$$

Definition 2.5. [19] (Discrete Empirical Interpolation Method(DEIM)) Given a full rank matrix $V \in \mathbb{R}^{m\times k}$ and a set of distinct indices $p \in \mathbb{N}^k$, the interpolatory projector for p onto $\mathrm{Ran}(V)$ is

$$\mathcal{P} \equiv V(P^{\mathsf{T}}V)^{-1}P^{\mathsf{T}},$$

where $P = I(:,p) \in \mathbb{R}^{m\times k}$, provided $P^{\mathsf{T}}V$ is invertible.

Generally $\mathcal{P}$ is an oblique projector, and it has an important property not generally setted by orthogonal projector: for any $x \in \mathbb{R}^m$,

$$(\mathcal{P}x)(p) = P^{\mathsf{T}}\mathcal{P}x = P^{\mathsf{T}}V(P^{\mathsf{T}}V)^{-1}P^{\mathsf{T}}x = P^{\mathsf{T}}x = x(p),$$

so the projected vector $\mathcal{P}x$ matches x in the p entries, justifying the name “interpolatory projector”.

The DEIM algorithm processes the columns of

$$V = [v_1 \quad v_2 \quad \cdots \quad v_k]$$

one at a time, starting from the leading singular vector v_1. Each step processes the next singular vector to produce the next index. The first index p_1 corresponds to the largest magnitude entry in v_1:

$$|v_1(p_1)| = \|v_1\|_\infty.$$

Now define $\boldsymbol{p_1} \equiv [p_1]$, and let

$$\mathcal{P}_\infty \equiv v_1(P_1^T v_1)^{-1}P_1^{\mathsf{T}}$$

denote the interpolatory projector for p_1 onto $\mathrm{Ran}(v_1)$. The second index p_2 corresponds to the largest entry in v_2, after the interpolatory projection in the v_1 direction has been removed:

$$r_2 \equiv v_2 - \mathcal{P}_1 v_2$$
$$|r_2(p2)| = \|r_2\|_\infty.$$

Notice that $r_2(p_1) = 0$, since $\mathcal{P}_1 v_2$ matches v_2 in the p_1 position, a consequence of interpolatory projection. This property ensures the process will never produce duplicate indices.

Now suppose we have $j-1$ indices, with

$$p_{j-1} = \begin{bmatrix} p_1 \\ \vdots \\ p_{j-1} \end{bmatrix},$$
$$P_{j-1} \equiv I(:, p_{j-1}),$$
$$V_{j-1} \equiv [v_1 \cdots v_{j-1}],$$
$$\mathcal{P}_{j-1} \equiv V_{j-1}(P_{j-1}^{\mathsf{T}}V_{j-1})^{-1}P_{j-1}^{\mathsf{T}}.$$

To select p_j, remove from v_j its interpolatory projection onto indices $\boldsymbol{p_{j-1}}$ and take the largest remaining entry:

$$r_j \equiv v_j - \mathcal{P}_{j-1}v_j$$

$$|r_j(p_j)| = \|r_j\|_\infty.$$

Assume that we picked

$$P = I(:,p) = [e_{p_1}, ..., e_{p_k}] \in R^{m\times k},\ Q = I(:,q) = [e_{q_1}, ..., e_{q_k}] \in R^{n\times k}$$

for A (not necessary to be rank k). Let

$$C = A(:,q) = AQ \in R^{m\times k} \textit{ and } R = A(p,:) = PA \in R^{k\times n}$$

and fatorize

$$A \approx C\mathrm{U}R.$$

Note that $U := ((C^TC)^{-1}C^T)A(R^T(RR^T)^{-1})$ is the best U.

we use a method DEIM to figure out C and R using U and $V^\top$ obtained from the original SVD. This method has the advantage of giving us the same result as the SVD. It's also better to understand because it take the rows and columns of a original matrix.

2.2. Natural Language Processing

In the field of linguistics and computational language processing, understanding the structural and functional aspects of language is of paramount importance. Therefore, we introduce key linguistic concepts that are foundational in analyzing and processing natural languages. These concepts collectively provide a comprehensive framework for linguistic analysis and computational processing, which are pivotal in the advancement of both theoretical and applied linguistics. We focused on the following characteristics to understand Korean. Morphology, a branch of grammar, deals with the morphological changes in words. It focuses on analyzing morphemes, which are the smallest language units that have meaning. In agglutinating languages, such as Korean, the form of the root of a word remains unchanged, and morphemes with their own meaning are connected in parallel to form phrases and sentences. Additionally, the study of syntax involves examining the structure, operation of sentences, and the elements that constitute sentences.

Definition 2.6. *Vocabulary* is the set of all unique terms across a collection of documents.

Definition 2.7. *The Term-Document Matrix (TDM)* is a matrix in which each row corresponds to a term from the vocabulary, each column corresponds to a document in the collection, and each (i, j) entry indicates the frequency of the i-th term in the j-th document .

Example 2.8. Consider a collection of two documents:

- Document 1: "apple orange"
- Document 2: "apple banana"

First, we build the vocabulary from the documents, which consists of the unique terms: {"apple", "orange", "banana"}.

The TDM would look like this:

Term/Document	Document 1	Document 2
apple	1	1
orange	1	0
banana	0	1

3. Method

3.1. LSA using SVD

Let A be a TDM and rank is r, then there exists U, V such that $A = U\Sigma V^\top$, where Σ is a diagonal matrix with singular value entries by SVD. The first r columns of the orthogonal matrix V contain r orthonormal eigenvectors associated with r nonzero eigenvalues of $A^\top A$. Since $A = U\Sigma V^\top$,

$$\begin{aligned} A^\top A &= (U\Sigma V^\top)^\top (U\Sigma V^\top) \\ &= (V\Sigma^\top U^\top)(U\Sigma V^\top) \\ &= V\Sigma^\top \Sigma V^\top. \end{aligned}$$

$V\Sigma^\top \Sigma V^\top$ is a square matrix representing the similarity between documents. Also, V represents the documents in the r-dimensional factor space as vectors. The (i, j) component in $V\Sigma^\top \Sigma V^\top$ is the similarity between the i-th document and the j-th document. The similarity is calculated by how many terms included in each document occur at the same time.

$$TDM = \underset{\text{Term}}{\overset{\text{Topic}}{\begin{bmatrix} & & \\ & U & \\ & & \end{bmatrix}}} \; \underset{\text{Topic}}{\overset{\text{Topic}}{\begin{bmatrix} & \Sigma & \end{bmatrix}}} \; \underset{\text{Topic}}{\overset{\text{Document}}{\begin{bmatrix} & V^\top & \end{bmatrix}}}$$

To identify clusters between topics and documents, we multiply Σ_k matrix and V_k^T matrix, where k represents the desired number of topics for classification, thus truncating the rank to k and making k the rank r. Σ is a diagonal matrix with the singular values of A on its diagonal. These values represent the importance of each latent semantic space, which is an abstract space where terms and documents are represented in a reduced dimensional form. Larger values indicate that the corresponding axis explains more of the total variance in the data. Specifically, the diagonal elements of Σ reflect the significance and variability of the latent topics. This shows how important each topic is in the relationship between documents and terms, making Σ crucial for LSA to uncover the latent semantic structure. At this time, the created matrix is D. The columns of D can represent the vectors in $\mathbb{R}^k$. Also, we project to the nearest axis through the cosine similarity of the vectors and the axis. Repeating the same process for the columns, we can group documents with the same axis into the same topic.

$$\text{Topic}\overset{\text{Document}}{\begin{bmatrix} & D & \end{bmatrix}} = \text{Topic}\overset{\text{Topic}}{\begin{bmatrix} & \Sigma & \end{bmatrix}}\ \text{Topic}\overset{\text{Document}}{\begin{bmatrix} & V^\top & \end{bmatrix}}$$

3.2. LSA using CUR Decomposition

Using the SVD results from the TDM generated in the previous steps, we will apply LSA analysis using CUR. CUR forms matrices C and R by sampling important columns and rows from the original matrix, and U is constructed to provide an approximation of A. CUR has the advantage of effectively explaining data analysis results since it samples actual data. Additionally, it can analyze mixed topics without the orthogonality constraint present in SVD. For document classification, we focus on the columns of the decomposed matrix V. To avoid confusion between $U\Sigma V^\top$ and CUR, we change U in $U\Sigma V^\top$ to W. As shown in the algorithm below, we can get k-index vectors p_k. Then, the unit vector Q that fits the vector P_k is multiplied by A to create $C = AQ$. At this time, we choose the column of the first k of C. Finally, the cosine similarity is calculated by multiplying each column of C by the original matrix A.

Algorithm 1 DEIM point selection algorithm

Input: $\mathbf{V}$, an $n \times k$ matrix $(n \geq k)$
Output: q, an integer vector with k distinct entries in $\{1, \dots, n\}$
$\mathbf{v} = \mathbf{V}(:, 1)$
$[\sim, q_1] = \max(|\mathbf{v}|)$
$\mathbf{q} = [q_1]$
for $j = 2, 3, \dots, k$ **do**
  $\mathbf{v} = \mathbf{V}(:, j)$
  $\mathbf{c} = \mathbf{V}(\mathbf{q}, 1 : j-1)^{-1}\mathbf{v}(\mathbf{q})$
  $\mathbf{r} = \mathbf{v} - \mathbf{V}(:, 1 : j-1)\mathbf{c}$
  $[\sim, q_j] = \max(|\mathbf{r}|)$
  $\mathbf{q} = [\mathbf{q}; q_j]$
end for
return q

$$Q = I(:, q) = [e_{q_1}, ..., e_{q_k}] \in R^{n\times k}$$

for A (not necessary to be rank k). Let

$$C = A(:, q) = AQ \in R^{m\times k}$$

Then we can get following CUR decomposition:

$$TDM = \text{Term}\overset{\text{Topic}}{\begin{bmatrix} \\ C \\ \\ \end{bmatrix}} \text{Topic}\overset{\text{Topic}}{\begin{bmatrix} U \end{bmatrix}} \text{Topic}\overset{\text{Document}}{\begin{bmatrix} R \end{bmatrix}}$$

Since we are interested in classifying documents about topics, we focus on C in the CUR decomposition. The result of computing the cosine similarity of each column of C and A

$$\text{sim}_{\text{cosine}}\left(A(:,q_k),A\right) \in R^{m\times k}$$

By selecting a higher similarity value from the resulting rows, we get the result of LSA using CUR.

4. Experiments and Conclusion

In this section, we compare SVD and CUR through several classification examples. In section 3.1, we provide a synthetic example. And section 3.2, we use a comparison using the Korean Wikipedia dataset, considering the difficulties of the Korean language in NLP. Wikipedia data set is useful for NLP because it generally provides objective facts and information. The experiment is conducted using MATLAB and Python.

4.1. Synthetic Example

In this section, we compare clustering using SVD and CUR decomposition through a synthetic example.

Example 4.1. Suppose that there are 500 points in each of two different tilted cylinders. Both cylinders have a center origin, a radius of 1, and a height of 20. The first cylinder is tilted -20 degrees in the y-axis direction and 30 degrees in the z-axis direction. The second cylinder is tilted 75 degrees in the x-axis direction and 15 degrees in the z-axis direction. Let the points in the first cylinder be A, and the points in the second cylinder be B. Let $X \in \mathbb{R}^{3\times 1000}$, such that let $X(:,1:500) = A$ and $X(:,501:1000) = B$.

The results of clustering the columns of X into two clusters using LSA with SVD is Figure 1. And the results of clustering the columns of X into two clusters using LSA with CUR decomposition is Figure 2

In this experiment, we examine how matrix factorizations classify documents with mixed topics. In practical datasets, topics often overlap to some extent. The orthogonality of SVD usually leads to poor classification performance for mixed-topic data. However, the CUR decomposition method, which samples directly from the original data, is more flexible in classifying documents with multiple topics. This study examines the results of using LSA with SVD and CUR on datasets with two related topics.

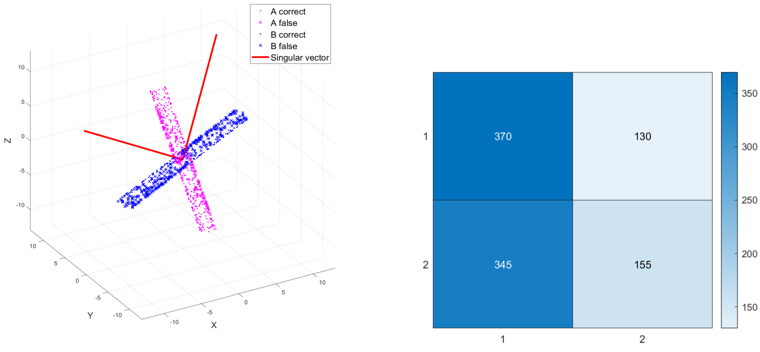

FIGURE 1. Classified points in two cylinder of SVD(left) and heat map(right) by using SVD.

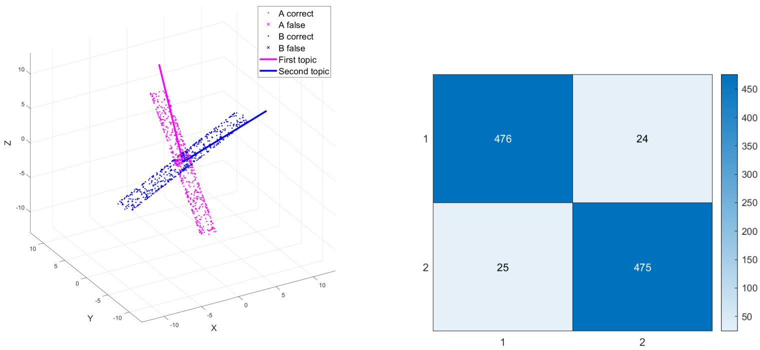

FIGURE 2. Classified points in two cylinder of CUR(left) and heat map(right) by using CUR decomposition.

4.2. Document Binary Clustering Experiment

In this section, we compare clustering using SVD and CUR decomposition through a document clustering example using text data. Data is selected from two topics from Wikipedia's documents. The reasons for using Wikipedia data are as follows. The first is the use of dictionary forms to provide consistent structured information. Second, there is no biased word use due to the collaboration of various people. Lastly, it is easy to access and can find clear targets through its own classification standard. The first topic is American politicians, and the second topic is Korean singers.

The first topic has 180 documents, and the other Korean singers topic has 135 documents. For the reasons of morphology, agglutinating language, and syntax, which are representative characteristics of Korean, TDM is constructed by extracting only nouns from text data. The data of words for each topic in the example are as follows.

	Topic1: American politicians	Topic2: Korean singers
Number of documents	180	135
Average value of terms	330	197
Standard deviation of words	685	278
Median of terms	123	113
Maximum value of terms	7175	2470
Minimum value of word	3	4

TABLE 1. Word distribution and statistical information in documents of two topics.

Because there is a big difference in the number of words contained in the document. We extracted the same number from each document and analyzed it. In this study, all comparisons with text were conducted using the built-in noun extraction feature of the Okt model, a Python library contained within the Konlpy package specialized in Korean NLP. The results of the experiment are Table1 and Table2.

Example 4.2. TDM consisted of nouns extracted from a total of 350 documents.

LSA using SVD tends to be unstable in classification tasks due to its sensitivity to changes in term frequency and its susceptibility to outliers. This sensitivity becomes particularly noticeable when there are significant fluctuations in the number of terms. On the other hand, LSA using CUR decomposition offers stability and accuracy, with exceptions primarily observed in cases characterized by extreme variations in term frequency.

5. Conclusion

In this paper, we compared LSA with SVD and LSA with CUR decomposition through experiments. We found that LSA using SVD is sensitive to term changes and LSA using SVD is strongly affected by outliers, while LSA using CUR decomposition is stable and shows better accuracy under various conditions. These findings indicate the potential of CUR decomposition for improving document clustering in Korean, offering a valuable approach for text mining and information retrieval in agglutinative languages.

# of letters	SVD (%)	CUR (%)
100	42.8	**64.7**
200	42.8	**98.0**
300	**100**	97.7
400	42.8	**98.4**
500	**100**	98.0
600	**99.6**	99.0
700	42.8	**99.0**
800	99.0	**99.6**
900	42.8	**62.8**
1000	42.8	**66.4**
Average	70.4	**88.3**

TABLE 2. Compared with the CUR decomposition SVD accuracy according to the number of letters.

6. Acknowledgments

This paper is based on the Master's thesis of the first author at Pusan National University [10].

References

[1] E. Alsentzer, J. R. Murphy, W. Boag, W.-H. Weng, D. Jin, T. Naumann, and M. McDermott, *Publicly available clinical BERT embeddings*, arXiv preprint arXiv:1904.03323 (2019).

[2] D. Araci, *Finbert: Financial sentiment analysis with pre-trained language models*, arXiv preprint arXiv:1908.10063 (2019).

[3] I. Beltagy, K. Lo, and A. Cohan, *SciBERT: A pretrained language model for scientific text*, arXiv preprint arXiv:1903.10676 (2019).

[4] G. G. Chowdhury, *Natural language processing*, Annu. Rev. Inf. Sci. Technol. 37 (2003), no. 1, 51–89.

[5] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, *Indexing by latent semantic analysis*, J. Am. Soc. Inf. Sci. 41 (1990), no. 6, 391–407.

[6] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, *Bert: Pre-training of deep bidirectional transformers for language understanding*, arXiv preprint arXiv:1810.04805 (2018).

[7] S. Gururangan, A. Marasović, S. Swayamdipta, K. Lo, I. Beltagy, D. Downey, and N. A. Smith, *Don't Stop Pretraining: Adapt Language Models to Domains and Tasks*, arXiv preprint arXiv:2004.10964 (2020).

[8] KB Bank AI Team, *KB-ALBERT-KO* (2020) , GitHub repository, `https://github.com/KB-Bank-AI/KB-ALBERT-KO`.

[9] J. Lee, W. Yoon, S. Kim, D. Kim, S. Kim, C. H. So, and J. Kang, *BioBERT: a pre-trained biomedical language representation model for biomedical text mining*, Bioinformatics 36 (2020), no. 4, 1234–1240.

[10] JH LEE, *Korean Document Clustering by Topic Using Matrix Factorizations*, Master's Thesis, Pusan National University (2021).

[11] E. D. Liddy, *Natural language processing* (2001), 4–6.

[12] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, *Roberta: A robustly optimized bert pretraining approach*, arXiv preprint arXiv:1907.11692 (2019).

[13] N. Ljubesic, D. Boras, N. Bakaric, and J. Njavro, *Comparing measures of semantic similarity*, ITI 2008-30th Int. Conf. Inf. Technol. Interfaces (2008), 675–682.

[14] M. W. Mahoney and P. Drineas, *CUR matrix decompositions for improved data analysis*, Proc. Natl. Acad. Sci. 106 (2009), no. 3, 697–702.

[15] M. W. Mahoney, M. Maggioni, and P. Drineas, *Tensor-CUR decompositions for tensor-based data*, SIAM J. Matrix Anal. Appl. 30 (2008), no. 3, 957–987.

[16] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, *Deep contextualized word representations*, arXiv preprint arXiv:1802.05365 (2018).

[17] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, *Exploring the limits of transfer learning with a unified text-to-text transformer*, arXiv preprint arXiv:1910.10683 (2019).

[18] SK T-Brain, *KoBERT* (2019), GitHub repository, `https://github.com/SKTBrain/KoBERT`.

[19] D. C. Sorensen and M. Embree, *A deim induced cur factorization*, SIAM J. Sci. Comput. 38 (2016), no. 3, A1454–A1482.

[20] L. N. Trefethen and D. Bau III, *Numerical linear algebra*, SIAM 50 (1997), 25-36.

June-Ho Lee
Department of Mathematics, Pusan National University,
Busan 46241, Republic of Korea,
Finance, Fishery, Manufacture Industrial Mathematics Center on Big Data,
Pusan National University, Busan 46241, Republic of Korea
Email address: `goodmath613@pusan.ac.kr`

Hyun-Min Kim
Department of Mathematics, Pusan National University,
Busan 46241, Republic of Korea,
Finance, Fishery, Manufacture Industrial Mathematics Center on Big Data,
Pusan National University, Busan 46241, Republic of Korea
Email address: `hyunmin@pusan.ac.kr`