Performance Analysis and Parallelization of CosineSimilarity of Documents

Bandi Harshavardhan Reddy[1], Gopa laasya lalitha priya[2], Kodakanti Prashanth[3], L Mohana Sundari[4]

[123]UG Student, School of Computer Science and Engineering, Vellore Institute of Technology, India.
[4]Assistant Prof Senior, School of Computer Science and Engineering, Vellore Institute of Technology, India.

---***---

***Abstract* - Currently, the Internet contains an extensive col- lection of documents, and search engines utilize web crawlers to retrieve content for queries. The retrieved pages are then ranked based on their relevance to the query using page rank algorithms. Typically, the cosine similarity algorithm is employedto determine the similarity between the retrieved content and the query. However, a challenge arises when dealing with a large set of retrieved documents. Applying the conventional cosine similarity algorithm to rank pages becomes difficult in such cases. To address this issue, we propose an optimized algorithm that utilizes parallelization to calculate the cosine similarity of documents in large sets. By parallelizing the procedure, we can enhance efficiency and reduce latency by processing a greater number of documents in less time.**

Keywords-Web Crawlers, Cosine Similarity, Parallelizing, Effi-ciency, Optimized

I. INTRODUCTION

The primary goal of the project is to utilize parallel computing to determine document similarity, thereby simplifying the task and achieving similarity results with reduced computational power. By implementing cosine similarity algorithms in parallel computing, we aim to enhance the speed and efficiencyof document search. Unlike other algorithms, cosine similarity can effectively address certain problems. When processing a query, numerous relevant documents are retrieved and sub- sequently require page ranking. However, the existing page ranking algorithm proves inefficient when dealing with a large number of retrieved documents. To overcome this challenge, we have devised a more powerful and efficient algorithm that leverages parallelization to process extensive document sets insignificantly less time compared to the page ranking algorithm.

II. LITERATURE REVIEW

[1] The rapid global expansion of the Internet has resulted in a massive amount of data being stored on servers. The amount of data produced in the last two years alone surpasses the cumulative data generated in previous years, primarily attributed to the extensive adoption of Internet of Things (IoT) devices. This data has emerged as a valuable resource for con- ducting predictive analysis of forthcoming events. However, the increasing diversity of data types and the speed at which it is being generated has posed a challenge for data analysis technology. The objective of this study is to examine theinteraction between big data files and a range of data mining algorithms, including Na¨ıve Bayes, Support Vector Machines, Linear Discriminant Analysis Algorithm, Artificial Neural Networks, C4.5, C5.0, and K-Nearest Neighbor. Specifically,

Twitter comments are analyzed as the input data for these algorithms. [2]This research paper examines the use of Sharpened Cosine Similarity (SCS) as an alternative to convolutional layers in image classification. While previous studies have reported promising results, a comprehensive empirical analysis of neural network performance using SCS is lacking. The researchers investigate the parameter behavior and potential of SCS as an alternative to convolutions in different CNN architectures, employing CIFAR-10 as a benchmark dataset. The findings indicate that while SCS may not significantly im-prove accuracy, it has the potential to learn more interpretable representations. Moreover, in specific situations, SCS might provide a marginal improvement in adversarial robustness.

[3] A parallelization approach to enhance the performance of the summarization process. By integrating the Non-dominated Sorting Genetic Algorithm II (NSGA-II) with MapReduce, the parallelization approach achieves enhanced efficiency and quicker extraction of text summaries from multiple documents. To evaluate its performance, the proposed method is comparedto a nonparallelized version of

the NSGA-II algorithm. [4] Within this study, we present two novel similarity metrics that are tailored for comparing sense-level representations. By harnessing the characteristics of sense embeddings, we effectively enhance existing strategies and achieve a more robust correlation with human similarity ratings. Furthermore, we advocate for the inclusion of a task that involves identifying the senses that underlie the similarity rating, alongside semantic similar- ity. Empirical results validate the advantages of the proposed metrics for both semantic similarity and sense identification tasks. Additionally, we provide a comprehensive explanation of the implementation of these similarity metrics using six crucial sets of sense embeddings. [5] The proposed system has three main components: pre-processing, feature extraction, and evaluation. In the preprocessing phase, the system removes stop words, stemming, and performs tokenization to convert the input into a set of words. Finally, the evaluation phase uses the cosine similarity technique to evaluate the similarity between the reference answer and the student's answer. [6] A neuromemristive competitive learning system, which is a type of artificial neural network that incorporates memristors as synaptic weights. The system is designed to learn from data inan unsupervised manner, without the need for labeled training data. In this paper, the MNIST dataset is utilized, comprising of 60,000 training images and 10,000 testing images showcasing handwritten digits. The system effectively learns the image features by mapping them onto a high-dimensional space and subsequently clustering them using cosine similarity. [7] Proposed a parallel algorithm to construct a nearest neighbor graph using the cosine similarity measure. The algorithm is based on the idea of dividing the data set into multiple partitions, and then While the proposed algorithm is effective in constructing a nearest neighbor graph using the cosine similarity measure in a parallel computing environment, there are some limitations to this research. Firstly, the 3 processing each partition in parallel. To calculate the similarity between every pair of data points, the cosine similarity measure isemployed. Subsequently, the algorithm creates a graph where each data point is represented as a node, and the edges connecting the nodes are determined by the cosine similarity measure. [8] This research paper focuses on the application of Natural Language Processing (NLP) techniques for measuring semantic text similarity in documents related to safety-critical systems. The objective is to assess the level of semantic equivalence among multi-word sentences found in railway safety documents. The study involves preprocessing and cleaningunstructured data of different formats, followed by the application of NLP toolkits and similarity metrics such as Jaccard and Cosine. The results indicate the feasibility of automating the identification of equivalent rules and procedures, as well as measuring similarity across diverse safety-critical documents, using NLP and similarity measurement techniques. [9] The paper introduces a methodological approach for evaluating the performance of discovery algorithms. The evaluation encompasses several metrics, including discovery estimation errors, complexity, range-based RSS technique, and criteria such as success ratio, residual energy, accuracy, and root-mean- square error (RMSE). To assess the performance, two distinct discovery studies, namely Hamming and Cosine, are compared with a reference RMSE. The paper concludes by presenting an overview of the discovery algorithm improvement cycle, which shows a 21 percent decrease in discovery error and improved RMSE accuracy, along with a 29 percent reduction in com- plexity through Euclidean distance analysis. [10] This research paper focuses on the growing relevance of bioinformatics,genomics, and biological computations due to the COVID-19 pandemic. It explores the representation of virus genomes as character strings based on nucleobases and applies document similarity metrics to measure their similarities. The study then applies clustering algorithms to the similarity results to group genomes together. In this paper, the utilization of P systems, also known as membrane systems, is introduced as computation models inspired by the information flow observed in membrane cells. These P systems are applied specifically for the purpose of data clustering. It investigates a novel and versatile clustering method for genomes using membrane clustering models and document similarity metrics, an area that has not been extensively explored in the context of membrane clustering models. [11] This article discusses the optimization and acceleration of machine learning using instruction set-based accelerators on Field Programmable Gate Arrays (FPGAs) for processing large-scale data. This article introduces a flexible accelerator design that incorporates out-of- order automatic parallelization, specifically designed for data- intensive applications. The accelerator is capable of supporting four essential applications: clustering algorithms, deep neural networks, genome sequencing, and collaborative filtering. To optimize instruction-level parallelism, the accelerator utilizes an out-of-order scheduling method for parallel dataflow computation. [12] This paper addresses the challenges faced by China Telecom in managing the increasing number and scope of technical projects. The process of examination, evaluation, and management becomes more complex, leading to repeated declaration and approval of projects. The paper focuses onthe semantic similarity calculation method for project con- tents. This method enables the convenient, efficient, and accurate identification of similar projects. By reducing redundant project construction, it enhances the utilization efficiency of research funds and improves the scientific research management level of the enterprise. The proposed approach aims to streamline project management, enhance resource allocation, and improve overall research management practices within China Telecom. [13] This article addresses the challenge of fast and efficient depth estimation from stereo images for real- time applications such as robotics and autonomous vehicles. While deep neural networks (DNN) have shown promising results, their computational complexity makes them unsuitable for real-world deployment. The proposed system, called StereoBit, aims to achieve real-time disparity map generation with close to state-of-the-art accuracy. It introduces a binary neural network for efficient similarity calculations using weighted Hamming distance. The network is derived from a well-trained network with cosine similarity using a novel approximation approach. An optimization framework is also presented to maximize the computing power of StereoBit while minimizing memory usage. Experimental evaluations demonstrate that StereoBit achieves 60 frames per second on an NVIDIA TITAN Xp GPU with a 3.56 percent non-occluded stereo error on the KITTI 2012 benchmark. The proposed system offers a fast and lightweight solution for real-time stereo estimation with high accuracy. [14] In this study, the authors address the scalability challenge associated with spectral clustering, a clustering technique that offers various advantages over k-means but suffers from high computational complexity and memory requirements. To overcome these limitations,the authors propose a GPU-based solution. They introduce optimized algorithms specifically designed to construct the similarity matrix in Compressed Sparse Row (CSR) format directly on the GPU. Furthermore, they leverage the spectral graph partitioning API provided by the GPU-accelerated nv- GRAPH library for efficient eigenvector extraction and other necessary computations. The authors conducted experiments using synthetic and real-world large datasets, showcasing the exceptional performance and scalability of their GPU implementation for spectral clustering. The proposed solution effectively enables efficient and scalable spectral clustering for large datasets, making it a valuable tool for data analysis and clustering tasks. [15] This project aims to address the challenge of finding desired rooms and compatible roommates by matching them according to their expectations. In situations like relocating for a job transfer, individuals often struggle to find suitable rooms, roommates, and locations that minimize commuting distances. Budget considerations are also taken into account, ensuring that rooms within the preferred price range are displayed. Unlike existing systems that overlook user preferences, this project considers both room and roommate preferences. The system utilizes filters based on amenities, gender, and price, displaying results based on a match score that reflects compatibility. Rooms and potential roommates are sorted based on the highest likelihood of being a good fit, offering users a personalized and efficient solution for finding their perfect room or preferred roommates. [16] This paper introduces a novel extractive approach for multi-document textsummarization. The approach begins by consolidating the con- tent from multiple documents into a single text file, eliminating redundancy. To ensure comprehensive content coverage while avoiding redundancy, the study employs a hybrid similarity approach based on the Word Mover Distance (WMD) and Modified Normalized Google Distance (M-NGD) (WM) Hy- brid Weight Method. The feature weights are optimized using Dolphin swarm optimization (DSO), a metaheuristic approach. The proposed methodology is implemented in Python using the multiling 2013 dataset, and its performance is assessed using ROUGE and AutoSummENG metrics. The experimental results validate the effectiveness of the proposed technique for multi-document text summarization. [17] This research focuses on enhancing performance in hydrodynamic simula- tion codes through various techniques. One such technique is Adaptive Mesh Refinement (AMR), which improves mem- ory optimization in mesh-based simulations. To address the challenges of integrating AMR into existing applications, a new branch called Phantom-Cell AMR is introduced, enabling a smooth transition to optimization while reducing devel- oper effort. The Phantom-Cell AMR scheme is tested on different architectures and parallel frameworks to showcase its optimizations. The research also investigates efficient data structures that optimize memory layout for cache performance, aiming for performant and portable codes across CPU and GPU architectures. Parallel performance and portability are prioritized to meet the requirements of high-performance computing systems. [18] In this research, we present a novel approach utilizing multiband on-off keying (MB-OOK) mod- ulation with a noncoherent receiver. We evaluate the perfor- mance of a differential MB-OOK transceiver employing a THz channel model. Our findings showcase a remarkable data rate of 54.24 Gbps with a bit error rate (BER) of 10 power -3 for distances less than 35 meters between the transmit- ter and receiver. Moreover, our proposed system exhibits a favorable balance between throughput, power consumption, and complexity. These advantageous attributes position it as a promising solution for THz applications that demand high- speed data transmission. [19] In this research, we introduce HyperOMS, a novel approach that leverages hyperdimensional computing to tackle these challenges. Unlike existing algorithms that utilize floating-point numbers to represent spectral data, HyperOMS encodes them as high-dimensional binary vectors. This encoding enables efficient OMS operations in a high-dimensional space. The parallelism and simplicity of boolean operations make HyperOMS suitable for implementation on parallel computing platforms. Experimental results demonstrate that HyperOMS, when executed on a GPU, achieves significantly faster performance (up to 17 times) and greater energy efficiency (6.4 times) compared to state-of-the-art GPU-based OMS tools. Furthermore, HyperOMS delivers comparable search quality to competing search tools. These findings highlight the potential of HyperOMS in accelerating protein identification while maintaining high search accuracy, thereby advancing the field of mass spectrometry- based proteomics. [20] Within this study, we present a novel similarity function designed for feature pattern clustering and high-dimensional text classification. Our proposed function facilitates supervised learning-based dimensionality reduction while preserving the underlying word distribution. Notably, our approach ensures consistency in word distribution both pre and post dimensionality reduction.

Empirical results validate the effectiveness of our method, as it successfully reduces dimensionality while preserving word distribution, leading to enhanced classification accuracies in comparison to alternative measures. This research contributes to the advancement of text document classification and clustering by addressing the limitations of traditional approaches and emphasizing the significance of word distribution.

III. PROPOSED METHODOLOGY

The proposed methology of our project goes as follows:

1) Download the documents from the web.

2) Parallelize the procedure of calculation of the TF for allthe terms in all the documents.

3) To address the issue of varying term frequencies in documents of different sizes, it is important to normalize the TF (Term Frequency). This normalization process becomes particularly crucial in large documents where term frequencies can be significantly higher compared to smaller documents. By normalizing the TF based on the document's size, this problem can be mitigated effectively. Notably, the process of TF normalization is parallelized for improved efficiency.

4) Normalized TF = (No.of times the term occuring in the document) / (Total no.of terms in the document)

5) Certain terms that occur too frequently have very little power, so they have to be weighed down whereas terms occurring less in the document may be more relevant so ithas to be weighed up.

6) IDF (Inverted Term Frequency) = 1+log2 (Total no.of documents) / (No.of documents with the word in it) After calculating the TF and IDF we have to calculate the similarity for documents.

Here,d1 is document and d2 is query

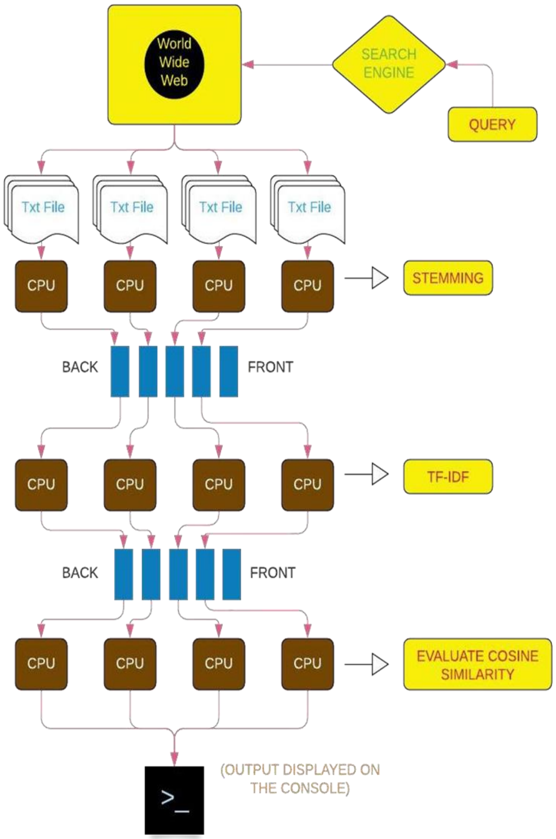
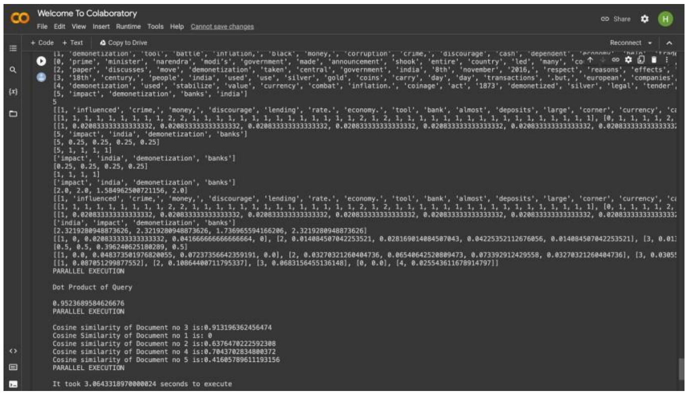


Fig. 1. Proposed Model Architecture

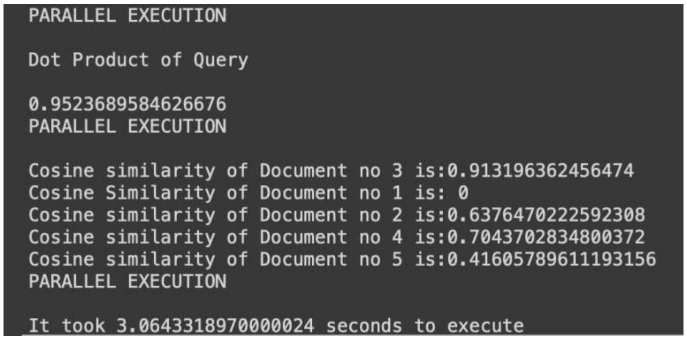
IV. RESULTS

The following snippets describe the output of the program and the difference of time interval between parallel execution and serial execution



```
[1, 'demonetization', 'tool', 'battle', 'inflation,', 'black', 'money,', 'corruption', 'crime,', 'discourage', 'cash', 'dependent', 'economy', 'help', 'trad
[0, 'prime', 'minister', 'narendra', 'modi's', 'government', 'made', 'announcement', 'shook', 'entire', 'country', 'led', 'many', 'co
[2, 'paper', 'discusses', 'move', 'demonetization', 'taken', 'central', 'government', 'india', '8th', 'november', '2016,', 'respect', 'reasons', 'effects', '
[3, '18th', 'century,', 'people', 'india', 'used', 'use', 'silver', 'gold', 'coins', 'carry', 'day', 'day', 'transactions', '.but,', 'european', 'companies',
[4, 'demonetization', 'used', 'stabilize', 'value', 'currency', 'combat', 'inflation.', 'coinage', 'act', '1873', 'demonetized', 'silver', 'legal', 'tender',
[5, 'impact', 'demonetization', 'banks', 'india']
5
[[1, 'influenced', 'crime,', 'money,', 'discourage', 'lending', 'rate.', 'economy.', 'tool', 'bank', 'almost', 'deposits', 'large', 'corner', 'currency', 'ca
[[1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 2, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1], [0, 1, 1, 1, 1, 2,
[[1, 0.020833333333333332, 0.020833333333333332, 0.020833333333333332, 0.020833333333333332, 0.020833333333333332, 0.020833333333333332, 0.020833333333333332, 0.02083333333333333
[5, 'impact', 'india', 'demonetization', 'banks']
[5, 0.25, 0.25, 0.25, 0.25]
[5, 1, 1, 1, 1]
['impact', 'india', 'demonetization', 'banks']
[0.25, 0.25, 0.25, 0.25]
[1, 1, 1, 1]
['impact', 'india', 'demonetization', 'banks']
[2.0, 2.0, 1.584962500721156, 2.0]
[[1, 'influenced', 'crime,', 'money,', 'discourage', 'lending', 'rate.', 'economy.', 'tool', 'bank', 'almost', 'deposits', 'large', 'corner', 'currency', 'ca
[[1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 2, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1], [0, 1, 1, 1, 1, 2,
[[1, 0.020833333333333332, 0.020833333333333332, 0.020833333333333332, 0.020833333333333332, 0.020833333333333332, 0.020833333333333332, 0.020833333333333332, 0.02083333333333333
['india', 'impact', 'demonetization', 'banks']
[2.3219280948873626, 2.3219280948873626, 1.736965594166206, 2.3219280948873626]
[[1, 0, 0.020833333333333332, 0.041666666666666664, 0], [2, 0.014084507042253521, 0.028169014084507043, 0.04225352112676056, 0.014084507042253521], [3, 0.01
[0.5, 0.5, 0.396240625180289, 0.5]
[[1, 0.0, 0.048373501976820055, 0.07237356642359191, 0.0], [2, 0.03270321260404736, 0.06540642520809473, 0.073392912429558, 0.03270321260404736], [3, 0.0305
[[1, 0.087051299877552], [2, 0.10864400711795337], [3, 0.0683156455136148], [0, 0.0], [4, 0.025543611678914797]]
PARALLEL EXECUTION

Dot Product of Query

0.9523689584626676
PARALLEL EXECUTION

Cosine similarity of Document no 3 is:0.913196362456474
Cosine Similarity of Document no 1 is: 0
Cosine similarity of Document no 2 is:0.6376470222592308
Cosine similarity of Document no 4 is:0.7043702834800372
Cosine similarity of Document no 5 is:0.41605789611193156
PARALLEL EXECUTION

It took 3.0643318970000024 seconds to execute
```

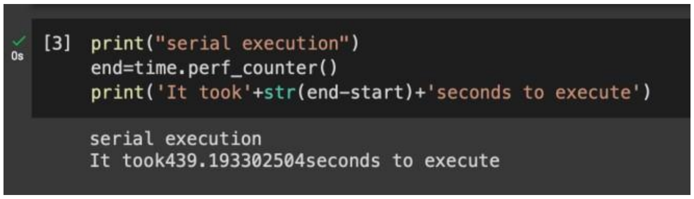
Fig. 2. Output of the program



```
PARALLEL EXECUTION

Dot Product of Query

0.9523689584626676
PARALLEL EXECUTION

Cosine similarity of Document no 3 is:0.913196362456474
Cosine Similarity of Document no 1 is: 0
Cosine similarity of Document no 2 is:0.6376470222592308
Cosine similarity of Document no 4 is:0.7043702834800372
Cosine similarity of Document no 5 is:0.41605789611193156
PARALLEL EXECUTION

It took 3.0643318970000024 seconds to execute
```

Fig. 3. Time taken for parallel execution

This shows the similarity between documents and the time it took to search the query on parallel execution.



```
[3]  print("serial execution")
     end=time.perf_counter()
     print('It took'+str(end-start)+'seconds to execute')

     serial execution
     It took439.193302504seconds to execute
```

Fig. 4. Time taken for serial execution

This shows the time the system took to search the query in the documents.

V. CONCLUSION

Based on our analysis and experimentation, it can be con- cluded that parallelizing the calculation of cosine similarity yields significant performance improvements. Our observa- tions indicate that the parallel implementation achieved a remarkable speedup of up to 140 times compared to the sequential implementation when dealing with large document collections.

Additionally, we found that selecting the appropriate number of threads for parallel execution depends on the available cores in the system and the size of the document collection. Insufficient thread utilization may occur when using too few threads, whereas excessive threads can result in overhead due to thread synchronization and contention.Furthermore, the performance of the cosine similarity calculation is influenced by the characteristics of the document collection, such as the average document length and the sparsity of document vectors. Notably, the algorithm's performance degrades as the sparsity of document vectors increases.

In summary, this project emphasizes the significance of par- allelization in efficiently computing cosine similarity for large document collections. It also highlights the importance of carefully tuning system parameters to achieve optimal perfor- mance.

The project can contribute to advancing the field of document similarity analysis, improve the efficiency of cosine similarity calculations, and enable the development of high-performance applications that rely on document matching and retrieval.

REFERENCES

[1] D. M. Amin and A. Garg, "Performance analysis of data mining algorithms," *Journal of Computational and Theoretical Nanoscience*, vol. 16, no. 9, pp. 3849–3853, 2019.

[2] S. Wu, F. Lu, E. Raff, and J. Holt, "Exploring the sharpened cosine similarity," in *I Can't Believe It's Not Better Workshop: Understanding Deep Learning Through Empirical Falsification*.

[3] J. M. Sanchez-Gomez, M. A. Vega-Rodríguez, and C. J. Pérez, "Paral- lelizing a multi-objective optimization approach for extractive multi- document text summarization," *Journal of Parallel and Distributed Computing*, vol. 134, pp. 166–179, 2019.

[4] D. Colla, E. Mensa, and D. P. Radicioni, "Novel metrics for computing semantic similarity with sense embeddings," *Knowledge-Based Systems*,vol. 206, p. 106346, 2020.

[5] M. A. Thalor, “A descriptive answer evaluation system using cosine sim- ilarity technique,” in *2021 International Conference on Communication information and Computing Technology (ICCICT)*. IEEE, 2021, pp. 1–4.

[6] B. W. Ku, C. D. Schuman, M. M. Adnan, T. M. Mintz, R. Pooser,

K. E. Hamilton, G. S. Rose, and S. K. Lim, “Unsupervised digit recognition using cosine similarity in a neuromemristive competitive learning system,” *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, vol. 18, no. 2, pp. 1–20, 2022.

[7] D. C. Anastasiu and G. Karypis, “Parallel cosine nearest neighbor graph construction,” *Journal of Parallel and Distributed Computing*, vol. 129, pp. 61–82, 2019.

[8] A. W. Qurashi, V. Holmes, and A. P. Johnson, “Document processing: Methods for semantic text similarity analysis,” in *2020 International Conference on INnovations in Intelligent SysTems and Applications (INISTA)*. IEEE, 2020, pp. 1–6.

[9] O. Hayat, R. Ngah, and S. Z. Mohd Hashim, “Performance analysis of device discovery algorithms for d2d communication,” *Arabian Journal for Science and Engineering*, vol. 45, pp. 1457–1471, 2020.

[10] P. Lehotay-Kéry and A. Kiss, “Membrane clustering of coronavirus variants using document similarity,” *Genes*, vol. 13, no. 11, p. 1966, 2022.

[11] C. Wang, L. Gong, X. Li, and X. Zhou, “A ubiquitous machine learning accelerator with automatic parallelization on fpga,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 31, no. 10, pp. 2346–2359, 2020.

[12] Z. Guo, X. Li, and X. Wang, “Research on text similarity: study case in project management,” in *Third International Conference on Machine Learning and Computer Application (ICMLCA 2022)*, vol. 12636. SPIE, 2023, pp. 204–211.

[13] G. Chen, H. Meng, Y. Liang, and K. Huang, “Gpu-accelerated real-time stereo estimation with binary neural network,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 31, no. 12, pp. 2896–2907, 2020.

[14] G. He, S. Vialle, N. Sylvestre, and M. Baboulin, “Scalable algorithms using sparse storage for parallel spectral clustering on gpu,” in *IFIP In- ternational Conference on Network and Parallel Computing*. Springer, 2021, pp. 40–52.

[15] S. Rahman *et al.*, “Optimal room and roommate matching system using nearest neighbours algorithm with cosine similarity distribution,” *Available at SSRN 3869826*, 2021.

[16] A. K. Srivastava, D. Pandey, and A. Agarwal, “Extractive multi- document text summarization using dolphin swarm optimization ap- proach,” *Multimedia Tools and Applications*, vol. 80, pp. 11 273–11 290, 2021.

[17] D. Dunning *et al.*, “Parallelization and performance portability in hydrodynamics codes,” Ph.D. dissertation, 2020.

[18] M. El Ghzaoui, J. Mestoui, A. Hmamou, and S. Elaage, “Performance analysis of multiband on–off keying pulse modulation with noncoher- ent receiver for thz applications,” *Microwave and Optical Technology Letters*, vol. 64, no. 12, pp. 2130–2135, 2022.

[19] J. Kang, W. Xu, W. Bittremieux, and T. Rosing, “Massively parallel open modification spectral library searching with hyperdimensional computing,” *arXiv preprint arXiv:2211.16422*, 2022.

[20] V. K. Kotte, S. Rajavelu, and E. B. Rajsingh, “A similarity function for feature pattern clustering and high dimensional text document classification,” *Foundations of Science*, vol. 25, no. 4, pp. 1077–1094, 2020.