

Research Article

An Improved Deep Learning Model for Word Embeddings Based Clustering for Large Text Datasets

Vijay Kumar Sutrakar^{*†} , Nikhil Mogre[†] 

Aeronautical Development Establishment, Defence Research and Development Organisation, Bangalore, India

Abstract

In the rapid growth of textual data in various domains has increased the need for efficient clustering techniques capable of handling large-scale datasets. Traditional clustering methods often fail to capture semantic relationships and struggle with high-dimensional, sparse data. The present study shows an improved document clustering technique, i.e., WEClustering++, which enhances the existing WEClustering framework by integrating fine-tuned BERT based word embeddings. The proposed model incorporates advanced dimensionality reduction techniques and optimized clustering algorithms to improve clustering accuracy. In the present work, the BERT-large model, fine-tuned on domain-specific datasets is utilized. Seven benchmark datasets spanning various domains and sizes are considered. These datasets include collections of research articles, news articles, and other domain-specific texts. Experimental evaluations on multiple benchmark datasets demonstrate significant performance improvements in clustering metrics, including silhouette score, purity, and ARI. Results show a 45% and 67% increase in median silhouette scores for WEClustering_K++ (K-means-based) and WEClustering_A++ (Agglomerative-based) models, respectively. Result also shows an increase of median purity metrics of 0.4% and 0.8% is obtained for proposed WEClustering_K++ and WEClustering_A++ compared to the state of art model. Also, an increase of median ARI metrics of 7% and 11% is obtained for proposed WEClustering_K++ and WEClustering_A++ compared to the state of art model. These findings highlight the potential of fine-tuned word embeddings in bridging the gap between statistical clustering robustness and semantic understanding. The proposed approach is expected to contribute to advancements in large-scale text mining applications, including document organization, topic modelling, and information retrieval.

Keywords

Deep Learning, Word Embedding, Large Text Data, Silhouette Score, Clustering Technique

1. Introduction

Recently, the explosion for textual data in digital form posed significant challenges for efficient information retrieval and processing. For example, Elsevier's repository contains over 37,000 articles on COVID-19 alone, and the number of articles published in English-language journals continues to

grow [1, 2]. A basic data grouping approach, clustering is frequently utilized in applications including document organization, social news clustering, and web search result clustering. It is also used as an initial step to tasks like sentiment analysis, topic extraction, and multi-document sum-

*Corresponding author: vks.ade@gov.in (Vijay Kumar Sutrakar)

† Nikhil Mogre and Vijay Kumar Sutrakar are co-first authors.

Received: 14 March 2025; Accepted: 31 March 2025; Published: 29 April 2025



Copyright: © The Author(s), 2025. Published by Science Publishing Group. This is an **Open Access** article, distributed under the terms of the Creative Commons Attribution 4.0 License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution and reproduction in any medium, provided the original work is properly cited.

marization. Bag of words (BOW) model [3] that utilizes scoring schemes like TF or TF-IDF [4] to numerically represent documents is a basis of traditional document approaches for clustering. However, such approaches have major limitations. It fails to capture semantic relationships, struggle with polysemy and synonymy, and face the curse of dimensionality, particularly for large datasets with high sparsity [5-8]. Ontology-based solutions like WordNet [9, 10] partially address semantic issues; however, it is limited by language coverage and design constraints. To tackle these challenges, word embeddings such as Word2Vec [11], GloVe [12], and FastText [13] provide dense, distributed representations of words. However, these embeddings are static and do not account for context-specific meanings. Recently, BERT [14] has emerged as a breakthrough model, generating contextual embeddings that adapt based on input context. Fine-tuning BERT on domain-specific datasets further enhances its ability to capture nuanced semantics, making it a robust tool for clustering and other text mining tasks.

Recently, WEClustering [15] technique is proposed for document clustering that integrates word embeddings [16-19] and clustering methods to improve clustering performance. It

begins with pre-processing, where documents are converted to lowercase and split into sentences, followed by tokenization, stop-word removal, and punctuation filtering. BERT is used to extract word embeddings, creating 1024-dimensional vectors for each word. These embeddings are clustered using Mini-Batch K-Means to form word clusters. Next, a concept-document (CD) matrix is generated by mapping TFIDF scores of words to their respective cluster centers. The resulting CD matrix is normalized and clustered using algorithms like Agglomerative Clustering or K-Means to produce document clusters [20].

In this paper, an improved document clustering technique i.e. WEClustering++ that leverages word embeddings derived from a fine-tuned BERT model is proposed. The proposed model addresses the issue of dimensionality, incorporates contextual semantics, and demonstrates high accuracy. The effectiveness of the proposed approach is demonstrated using multiple datasets and performance metrics, highlighting its suitability for large-scale text datasets. The details of proposed model are provided in Section 2. Section 3 discussed results and discussions followed by concluding remarks in Section 4.

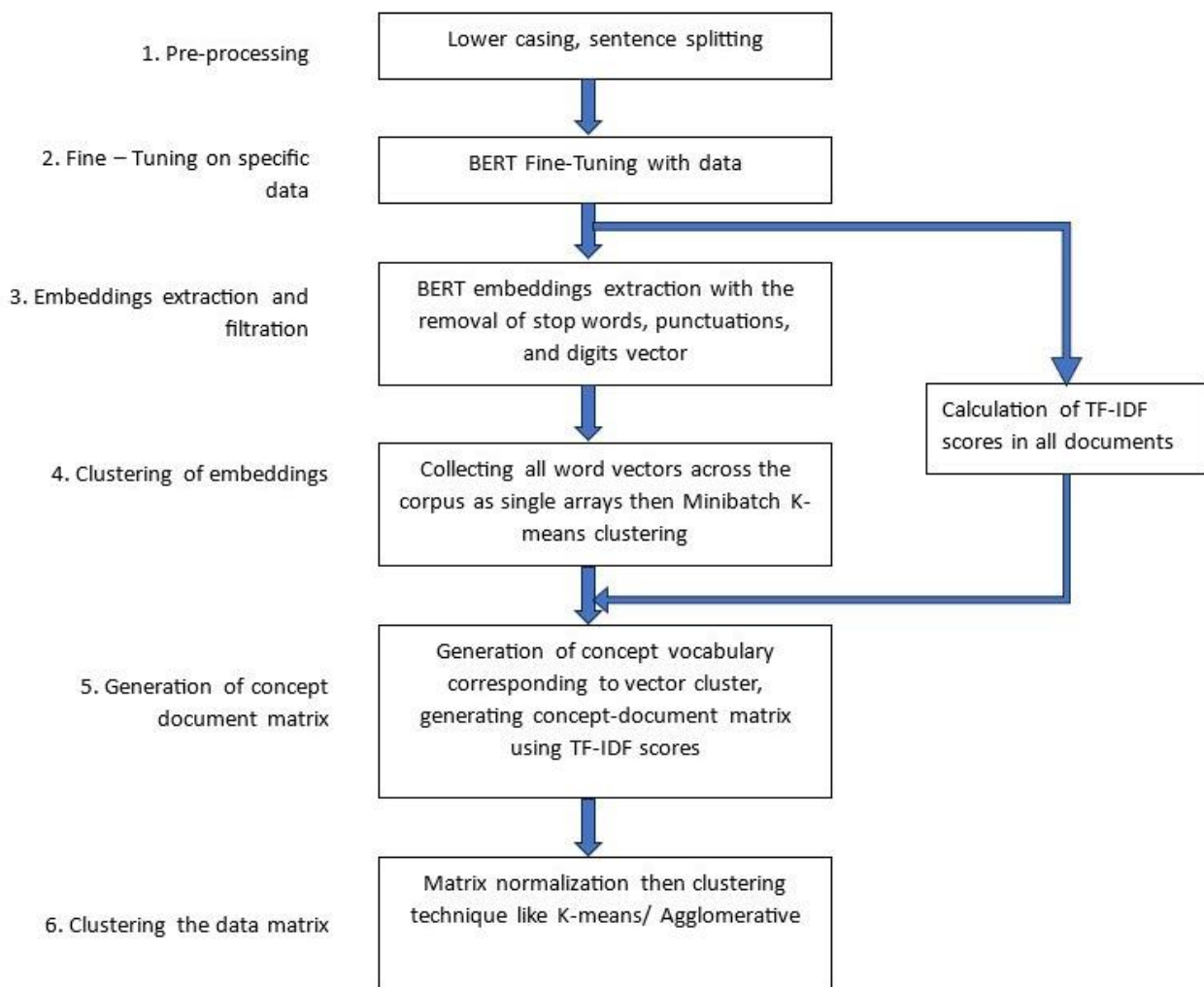


Figure 1. Flow chart of improved deep learning model for word embedding based clustering for large text datasets.

2. Methodology

2.1. Model Details

The proposed model builds upon the WEClustering framework [15] by incorporating fine-tuning of the BERT model, enabling the system to specialize for specific domains or applications. This enhancement allows the model to capture domain-specific nuances in text, making the clustering process more precise and relevant. However, the foundational steps of WEClustering [15] are retained. The details of fine-tune of embeddings are shown in Figure 1.

The first phase, preprocessing, involves preparing the text data for processing by the fine-tuned BERT model. This begins with lowercasing all text to ensure uniformity, followed by sentence splitting, as BERT operates at the sentence level to preserve contextual relationships. In the second phase, the pre-processed text is then ready for the fine-tuned BERT model, which generates embeddings that are contextually rich and domain-specific, aligning with the semantics of the targeted application. The third phase, embedding extraction and filtration, involves passing the pre-processed text through the finetuned BERT model. Unlike the original WEClustering [15], which uses a general pre-trained BERT model, the proposed model employs a domain-specific fine-tuned version. This enables better capture of word meanings and relationships relevant to the domain. The filtration step removes noise by excluding embeddings associated with stop words, punctuation, and digits, resulting in a refined set of embeddings that retain both statistical and semantic relevance.

The fourth phase focuses on clustering word embeddings. Mini-Batch K-Means [21] is used to group embeddings into clusters, each representing a concept or theme derived from the text. Elbow Method [22] is used for obtaining optimal cluster number, ensuring that the generated concepts are meaningful and representative of the corpus. By leveraging fine-tuned embeddings, these concepts become more cohesive and domain-relevant compared to those in the original WEClustering framework. This step also reduces the vocabulary size, transforming it from tens of thousands of words to a concise set of concepts. In the fifth phase, a Concept-Document (CD) matrix is constructed to represent documents in terms of the identified concepts rather than individual words. A scoring mechanism combining TF-IDF with the frequency of words from each concept calculates the relevance of concepts to each document. The fine-tuned embeddings ensure that these concepts are semantically aligned

with the documents, creating an accurate and meaningful matrix representation.

In the final step, the documents are clustered using algorithms like K-Means [20] or Agglomerative Clustering [23], based on the CD matrix. The dimensionality of the input is substantially reduced compared to traditional word-based approaches, leading to more cohesive and well-separated clusters. Fine-tuned embeddings enhance the cohesion and representational accuracy of the clusters, ultimately resulting in well-defined and meaningful document groupings. The improved models developed in the present work are called as WEClustering_K++ (based on K-means) and WEClustering_A++ (based on Agglomerative clustering) in rest of the document. This comprehensive framework demonstrates the advantages of integrating domain-specific fine-tuning into the existing WEClustering process [15].

2.2. Model Parameters

In general, two BERT models are available for generating embedding vectors in the embedding extraction phase. The BERT-small model [24] generates embeddings of size 768 and is available in both case-sensitive and case-insensitive versions. The BERT-large model generates embeddings of size 1024 and is available only as a case-sensitive version. In the present work, the BERT-large model, fine-tuned on domain-specific datasets is utilized. The fine-tuning ensures that the generated embeddings are not only rich in contextual meaning but also more aligned with domain-specific semantics, enhancing the clustering process. The use of higher-dimensional embeddings further improves the representation of word semantics.

In the word embeddings phase, the Mini-Batch K-Means algorithm [21] is employed for clustering. The two key parameters in this phase are: (i) the number of word clusters (K_{voc}) and (ii) the batch size (b). The value of K_{voc} is determined using the Elbow method [22], which ensures optimal clustering. For example, for the Articles-253 dataset, the value of K_{voc} is determined as 35 using [21] and the present work the K_{voc} for fine tuning is obtained 30, as shown in Figure 2. The remaining model's K_{voc} values are shown in Figure 3. The incorporation of fine-tuned embeddings often leads to smaller, more cohesive clusters, reflecting the improved semantic alignment of the words. The batch size parameter, b is chosen to balance execution efficiency and accuracy (taken from [21]).

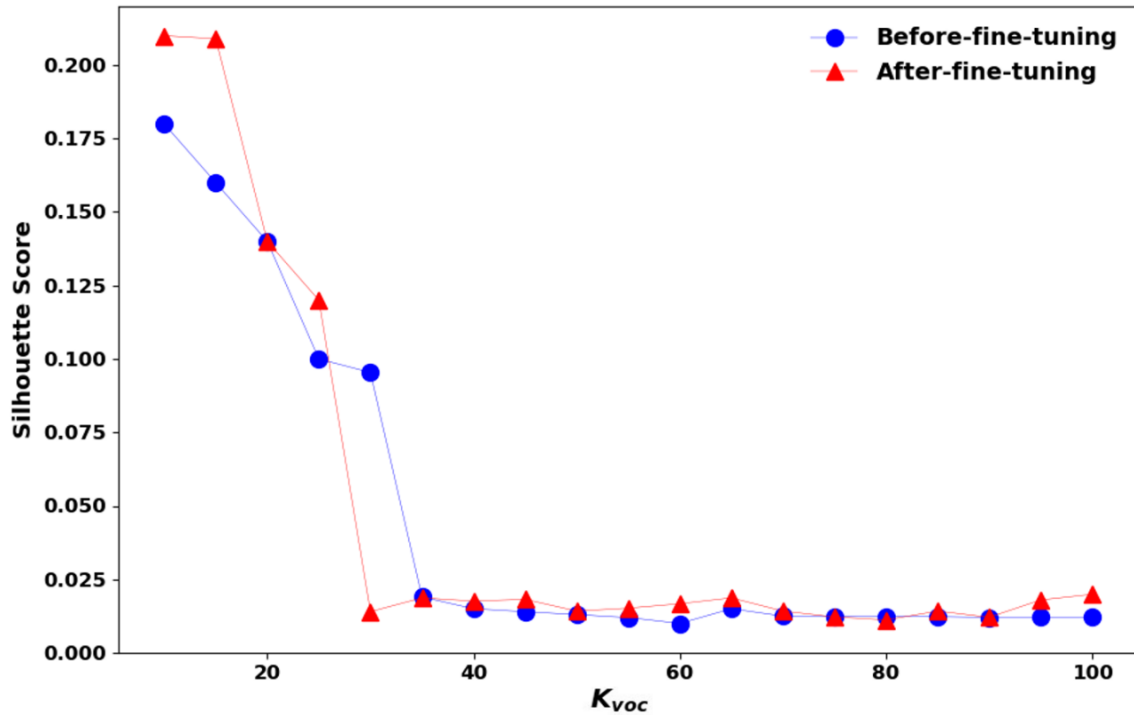


Figure 2. An example of Articles-253 dataset for finding K_{voc} using Elbow method [22].

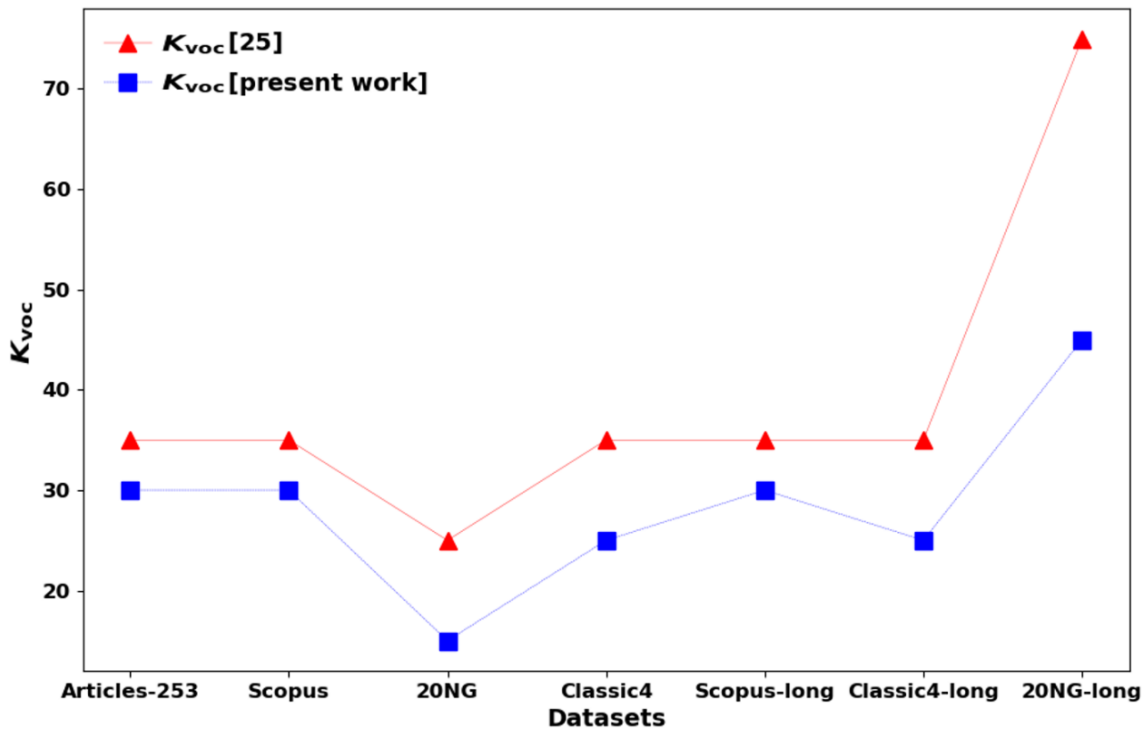


Figure 3. K_{voc} considered for different datasets.

In the final phase, document clustering is performed using algorithms such as K-means or agglomerative clustering. For agglomerative clustering, ward linkage is used as the linkage criterion. In the K-means clustering, the number of clusters (c) corresponds to the number of categories in the dataset. To improve the initialization of centroids, the K-means++

method is utilized [25], and the algorithm is executed 20 times to ensure the best clustering results.

The integration of fine-tuning in WEClustering significantly impacts the value of K_{voc} . Finetuned embeddings provide better semantic alignment and reduce redundancy, resulting in more meaningful and cohesive word clusters.

Consequently, K_{voc} decrease compared to the original WE-Clustering approach, as the finer embeddings better represent the underlying concepts in the text.

2.3. Datasets

The evaluation of an improved deep learning model for word embedding based clustering for large text dataset is conducted using seven benchmark datasets spanning various domains and sizes (refer [26] and [27] and Figure 4 for further details). These datasets include collections of research articles, news articles, and other domain-specific texts. The details of each dataset are given next.

1. Articles-253: A collection of 253 research articles across five categories, such as "Mobile Computing," "Political Science," and "Weather Review," including titles, abstracts, and references.
2. Scopus: Contains 500 articles, evenly distributed into

five categories like "Concrete," "Hyperactivity," and "Tectonic Plates," with titles and abstracts sourced from the Scopus database.

3. 20NG: A subset of the 20 Newsgroups dataset with 700 documents from four categories: "Atheism," "Religion," "Graphics," and "Space."
4. Classic4: Comprises 800 research articles across three main domains—"Aerodynamics," "Medical," and "Algorithms."
5. Scopus-long: A larger version of the Scopus dataset with 2800 articles spread across seven categories, including "Neural Networks," "Protons," and "Photosynthesis."
6. Classic4-long: An expanded version of Classic4 with 3891 documents.
7. 20NG-long: A larger subset of 20 Newsgroups with 8131 documents spanning nine categories like "Motorcycles," "Politics," "Hockey," and "Electronics."

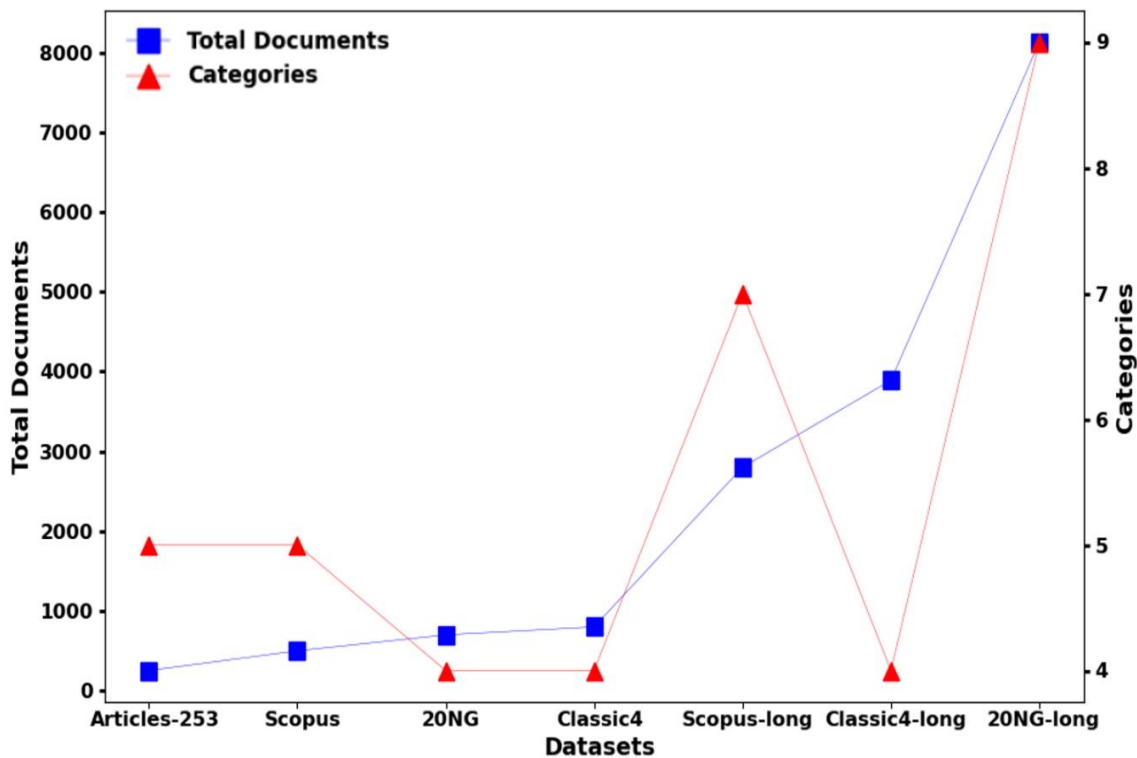


Figure 4. Total documents and categories considered for different data sets (taken from [26] and [27]).

2.4. Performance Metrics

Clustering quality can be evaluated using various metrics that are broadly classified into external and internal categories, depending on the availability of ground truth labels. External metrics are used when true labels are available, while internal metrics are applicable when true labels are not known. In the present work, three metrics are used to assess the quality of clustering, i.e. (a) silhouette coefficient, (b) ARI, and (c) pu-

rity.

The silhouette coefficient is an internal metric that evaluates how well-separated and compact the clusters are. It is particularly useful when true labels are unavailable. Its values range from -1 to +1, where higher values indicate that clusters are dense and well-separated, while values closer to -1 suggest poor clustering [28]. The ARI is an external metric used to assess clustering quality when true labels are present. ARI measures the similarity between the clustering results and the true labels while adjusting for random chance. A score near +1

indicates strong agreement with the true labels, whereas a score near -1 signifies that the clustering is random or completely dissimilar to the actual labels [29]. Purity, another external metric, measures clustering quality by determining how well each cluster corresponds to a single class. Higher purity values indicate that the clusters align well with the true class labels [30]. Together, these metrics provide a comprehensive evaluation of clustering performance, enabling the assessment of both internal structure and alignment with external benchmarks when true labels are available.

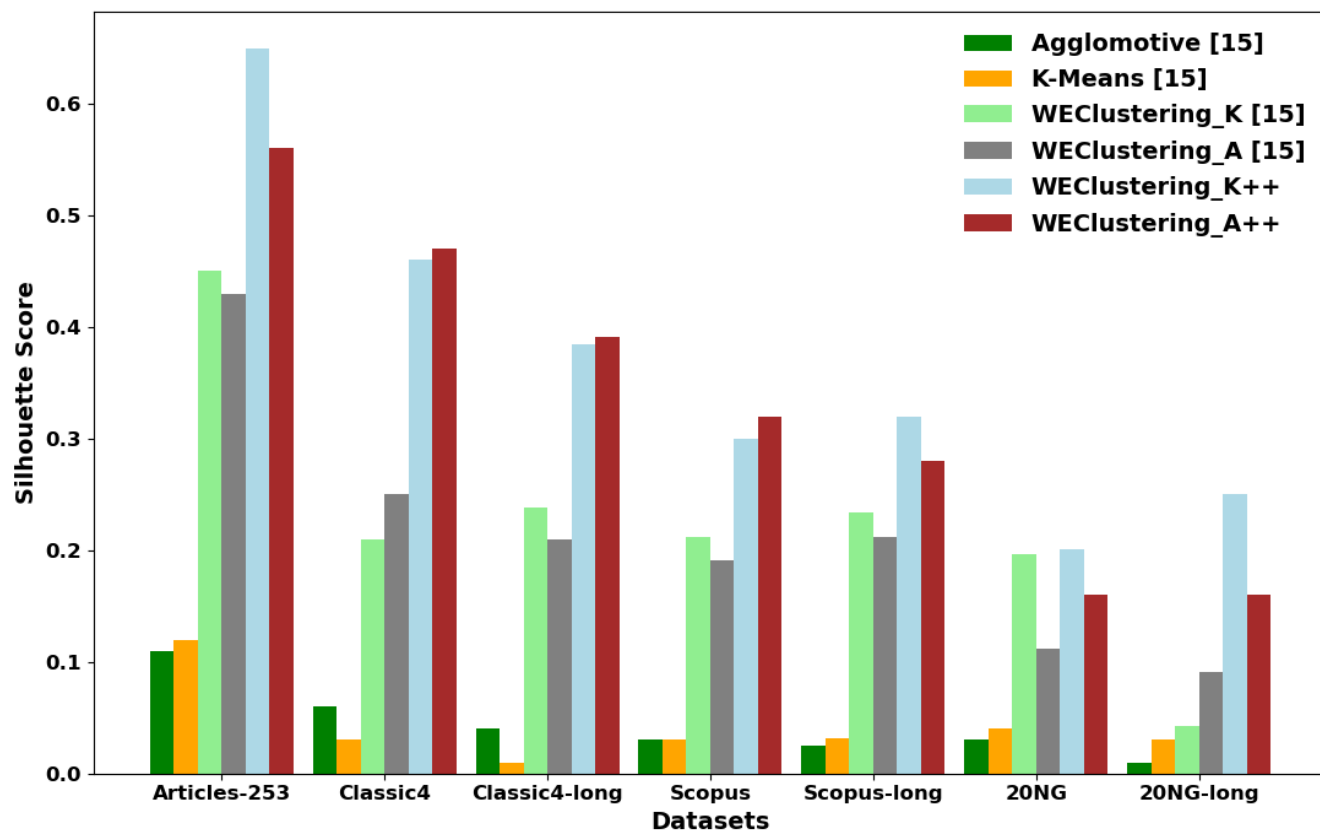
3. Results and Discussions

The WEClustering_K++ method exhibits notable enhancements in silhouette coefficient over WEClustering_K [15] across all datasets, as shown in Figure 5(a) and Table 1. In the Articles-253 dataset, the proposed model achieves a silhouette score of 0.65, a significant 44.4% increase over WEClustering_K [15]. Classic4 also benefits, with the score rising from 0.21 to 0.46, reflecting a 119% improvement. Classic4-long experiences a boost from 0.238 to 0.385, equating to a 61.7% increase. In the Scopus dataset, the improvement stands at 41.5%, while Scopus-long sees a 36.8% rise. The most striking gain is in the 20NG-long dataset, where the silhouette score surges from 0.043 to 0.25, an impressive 481% increase. Even in the 20NG dataset, a smaller but noticeable improvement of 2% is recorded. Similarly, WEClustering_A++ surpasses WEClustering_A [15], showing a 53.5% gain (0.43 to 0.56) in Articles-253, an 88% increase in Classic4, and an 86% rise in Classic4-long. Scopus and Scopus-long also benefit, with improvements of 67.5% and 32%, respectively. The 20NG dataset improves by 42.8%, underscoring the superior performance of the proposed approach. Also, an increase of median silhouette score metrics of 45% and 67% is obtained for proposed WEClustering_K++ and WEClustering_A++ compared to the state of art model, i.e. WEClustering_K [15] and WEClustering_A [15], respectively. In the case of WEClustering_K++, minimum increase of silhouette score metrics is +2% for 20NG dataset, while the maximum increase of silhouette score metrics is +481% for 20NG-long dataset. In the case of WEClustering_A++, minimum increase of silhouette score metrics is +32% for Scopus-long dataset, while the maximum increase of silhouette score metrics is +88% for Classic4 dataset, as shown in Figure 6(a).

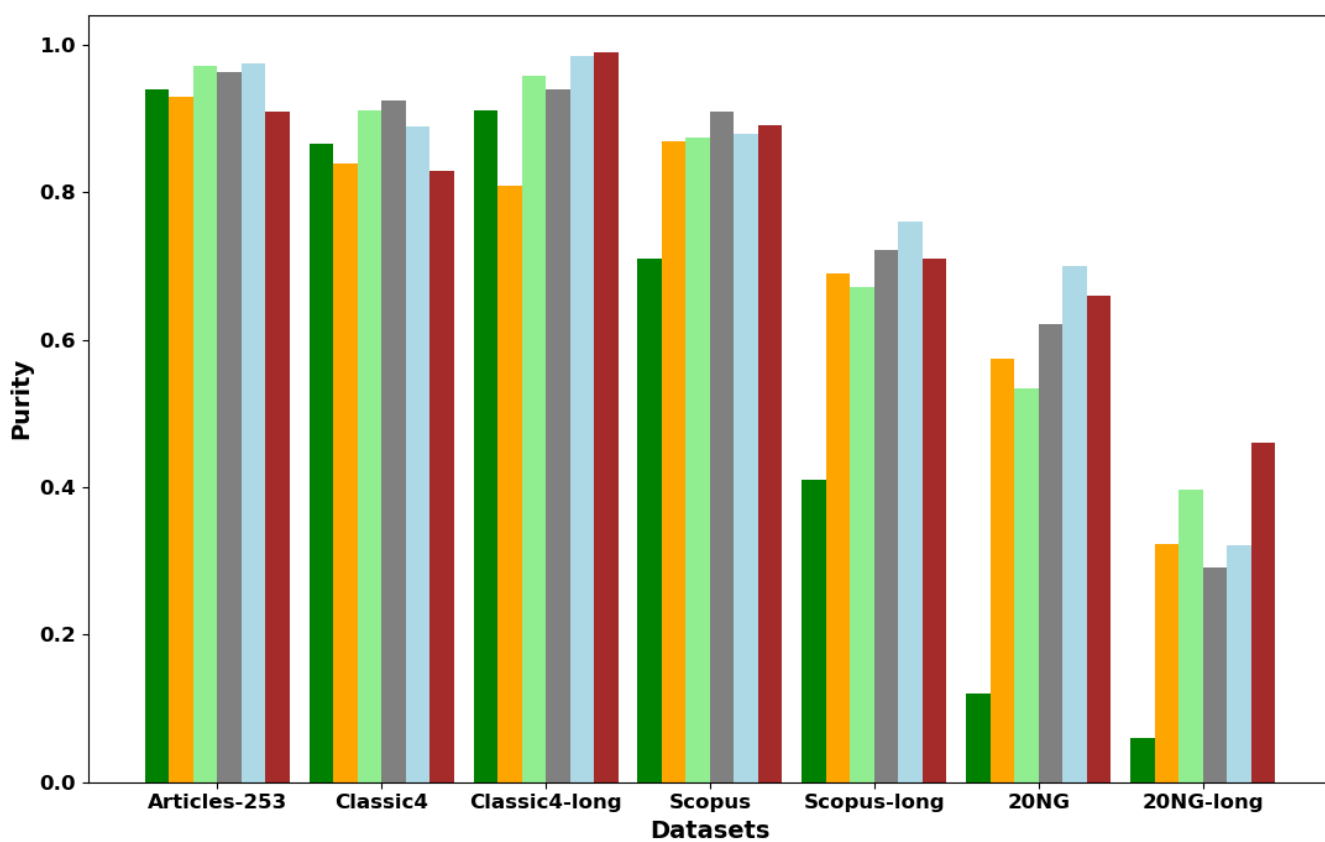
In terms of purity scores, the proposed models outperform WEClustering, as shown in Figure 5(b) and Table 2. WEClustering_K++ achieves a purity score of 0.975 in the Articles-253 dataset, slightly surpassing WEClustering_K's 0.971. However, in Classic4, WEClustering_K++ records a slightly lower score (0.89) than WEClustering_K's 0.911. Classic4-long shows a significant increase, with purity rising from 0.958 to 0.985. Scopus sees a reduction from 0.975 to 0.88, while Scopus-long experiences an increase from 0.722 to 0.76. The 20NG dataset

demonstrates a notable improvement, with the purity score increasing from 0.534 to 0.701, while 20NG-long records a change from 0.397 to 0.321. WEClustering_A++ also shows consistent improvements, with Articles-253 increasing from 0.963 to 0.971, Classic4 experiencing an increase from 0.925 to 0.912, and Classic4-long showing a remarkable jump from 0.94 to 0.991. Scopus-long remains stable at 0.722, while the 20NG dataset sees a 6.1% rise. The 20NG-long dataset experiences the most substantial gain, improving from 0.291 to 0.46, reinforcing the efficacy of the proposed clustering technique. Also, an increase of median purity metrics of 0.4% and 0.8% is obtained for proposed WEClustering_K++ and WEClustering_A++ compared to the state of art model, i.e. WEClustering_K [15] and WEClustering_A [15], respectively. In the case of WEClustering_K++, minimum change of purity score is -19.1% for 20NG-long dataset, while the maximum increase of purity score is +31.3% for 20NG dataset. In the case of WEClustering_A++, minimum change of purity score is -1.41% for Classic4 dataset, while the maximum increase of purity score is +58% for the 20NG-long, as shown in Figure 6(b).

The WEClustering_K++ and WEClustering_A++ models also demonstrate significant advantages in terms of ARI, as shown in Figure 5(c) and Table 3. In the Articles-253 dataset, WEClustering_K++ attains an ARI of 0.985, a slight increase over WEClustering_K's 0.971. The Classic4 dataset records a rise from 0.932 to 0.96, whereas Classic4-long sees a slight drop from 0.96 to 0.915. The Scopus dataset showcases an improvement, with ARI increasing from 0.925 to 0.967, while Scopus-long advances from 0.71 to 0.76. The most prominent improvement is observed in the 20NG dataset, where the ARI score jumps from 0.434 to 0.601. Similarly, the 20NG-long dataset exhibits a sharp increase from 0.202 to 0.521. WEClustering_A++ also delivers strong performance gains, with Articles-253 experiencing a slight drop from 0.989 to 0.967. The Classic4 dataset shows only a minor decline, while Classic4-long sees an increase from 0.847 to 0.931. Scopus-long progresses from 0.672 to 0.8, while 20NG rises from 0.302 to 0.56. The highest improvement is observed in the 20NG-long dataset, where the ARI score jumps from 0.191 to 0.6, demonstrating the effectiveness of the proposed clustering method. Also, an increase of median ARI metrics of 7% and 11% is obtained for proposed WEClustering_K++ and WEClustering_A++ compared to the state of art model, i.e. WEClustering_K [15] and WEClustering_A [15], respectively. In the case of WEClustering_K++, minimum change of ARI score is -4.69% for Classic4-long dataset, while the maximum increase of ARI score is +157.92% for 20NG-long dataset. In the case of WEClustering_A++, minimum change of ARI score is -2.23% for Articles-253 dataset, while the maximum increase of ARI score is +214.14% for the 20NG-long dataset, as shown in Figure 6(c). Overall, the WEClustering_K++ and WEClustering_A++ models consistently deliver superior results compared to their baseline counterparts across all three-evaluation metrics.



(a)



(b)

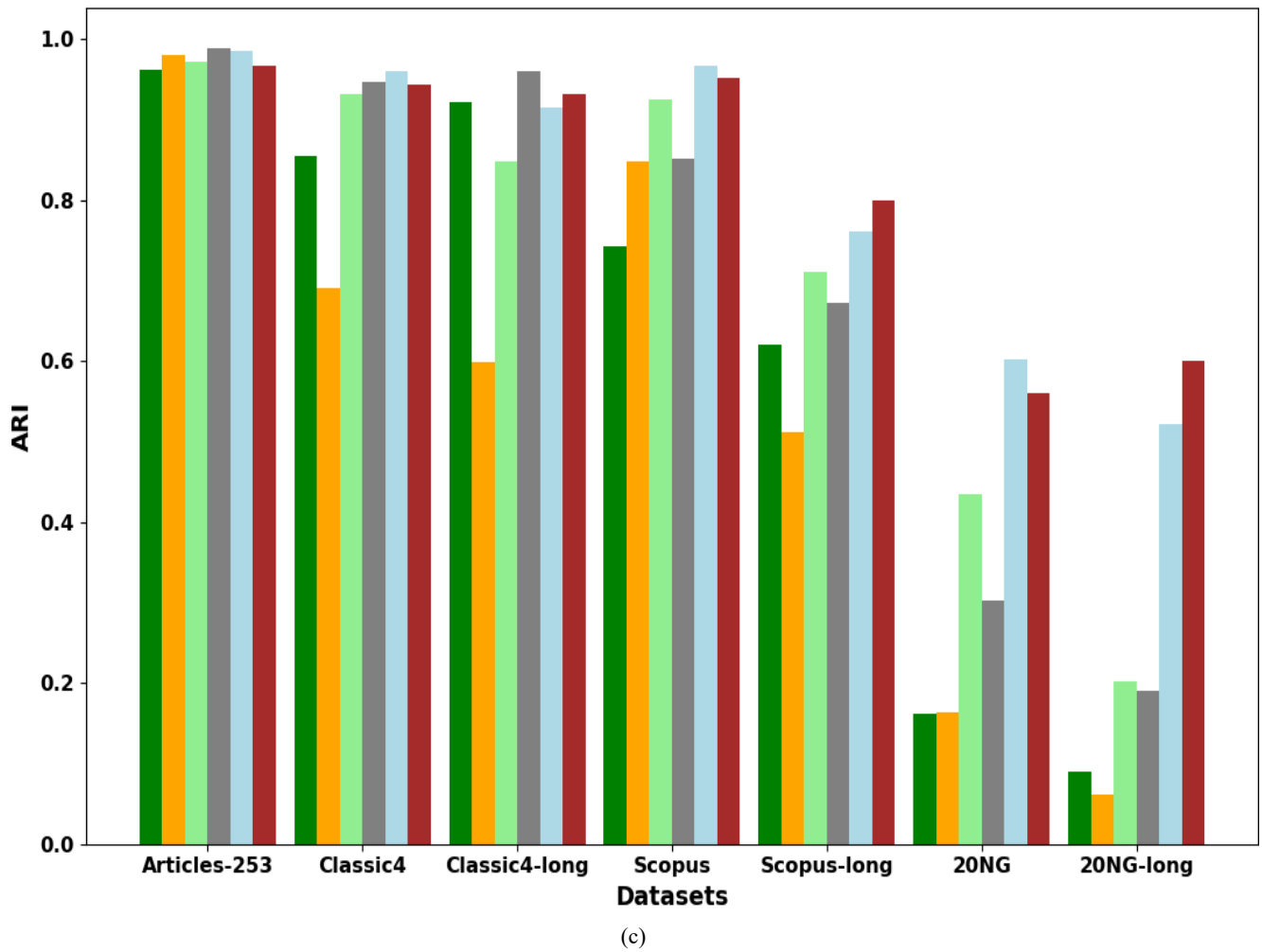


Figure 5. Variation of (a) Silhouette coefficient, (b) purity, and (c) ARI for different dataset and for different clustering techniques.

Table 1. Silhouette coefficient of state-of-the-art models versus proposed model for different datasets.

Data	Agglomerative [15]	K_means [15]	WEClustering_K [15]	WEClustering_A [15]	WEClustering_K++	WEClustering_A++
Articles-253	0.11	0.12	0.45	0.43	0.65	0.56
Classic4	0.06	0.03	0.21	0.25	0.46	0.47
Classic4-long	0.04	0.01	0.238	0.21	0.385	0.391
Scopus	0.03	0.03	0.212	0.191	0.3	0.32
Scopus-long	0.025	0.032	0.234	0.212	0.32	0.28
20NG	0.031	0.04	0.197	0.112	0.201	0.16
20NG-long	0.01	0.03	0.043	0.091	0.25	0.16

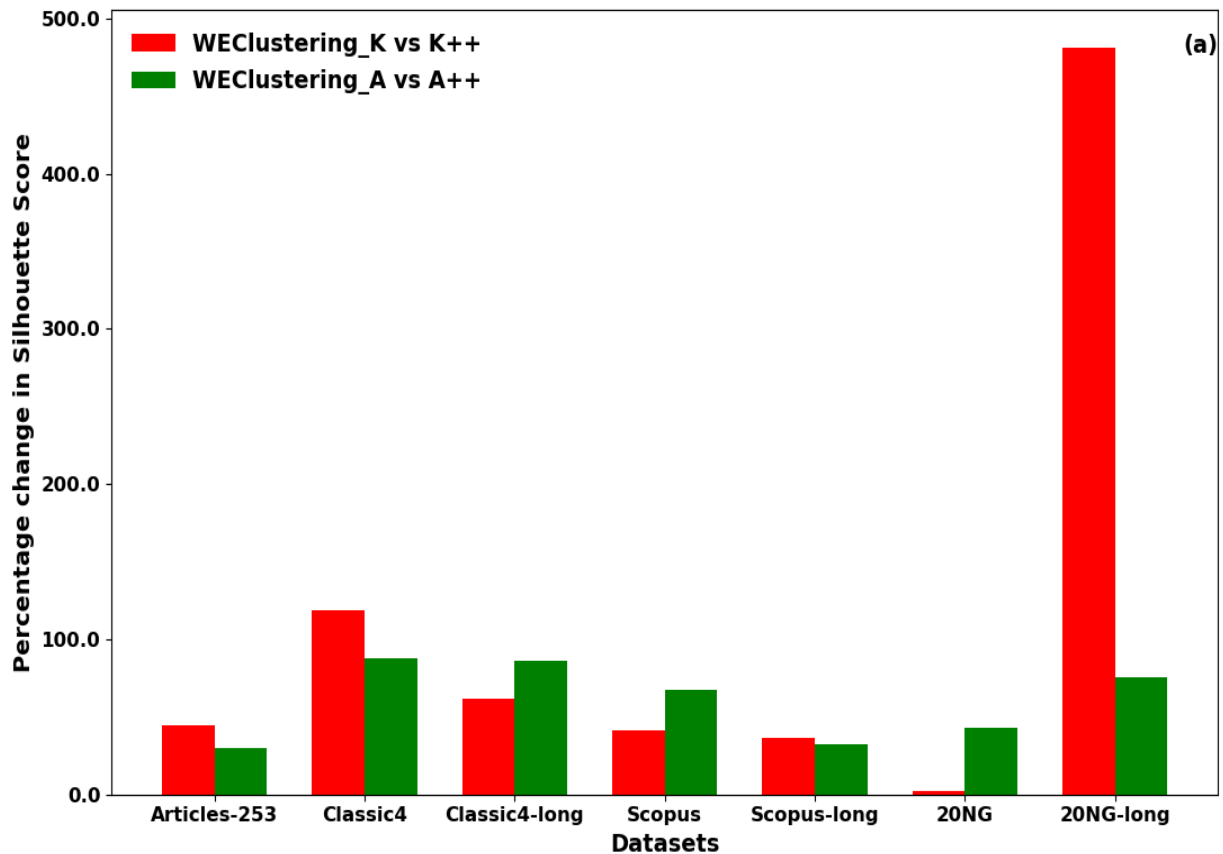
Table 2. Purity values of state-of-the-art models versus proposed model for different datasets.

Data	Agglomerative [15]	K_means [15]	WEClustering_K [15]	WEClustering_A [15]	WEClustering_K++	WEClustering_A++
Articles-253	0.94	0.93	0.971	0.963	0.975	0.971

Data	Agglomerative [15]	K_means [15]	WEClustering_K [15]	WEClustering_A [15]	WEClustering_K++	WEClustering_A++
Classic4	0.866	0.84	0.911	0.925	0.89	0.912
Classic4-long	0.911	0.81	0.958	0.94	0.985	0.991
Scopus	0.71	0.87	0.975	0.91	0.88	0.892
Scopus-long	0.41	0.69	0.722	0.722	0.76	0.71
20NG	0.12	0.574	0.534	0.622	0.701	0.66
20NG-long	0.94	0.93	0.971	0.963	0.975	0.971

Table 3. ARI values of state-of-the-art models versus proposed model for different datasets.

Data	Agglomerative [15]	K_means [15]	WEClustering_K [15]	WEClustering_A [15]	WEClustering_K++	WEClustering_A++
Articles-253	0.961	0.98	0.971	0.989	0.985	0.967
Classic4	0.854	0.69	0.932	0.947	0.96	0.943
Classic4-long	0.921	0.598	0.847	0.96	0.915	0.931
Scopus	0.743	0.847	0.925	0.851	0.967	0.952
Scopus-long	0.621	0.511	0.71	0.672	0.76	0.8
20NG	0.162	0.164	0.434	0.302	0.601	0.56
20NG-long	0.09	0.062	0.202	0.191	0.521	0.6



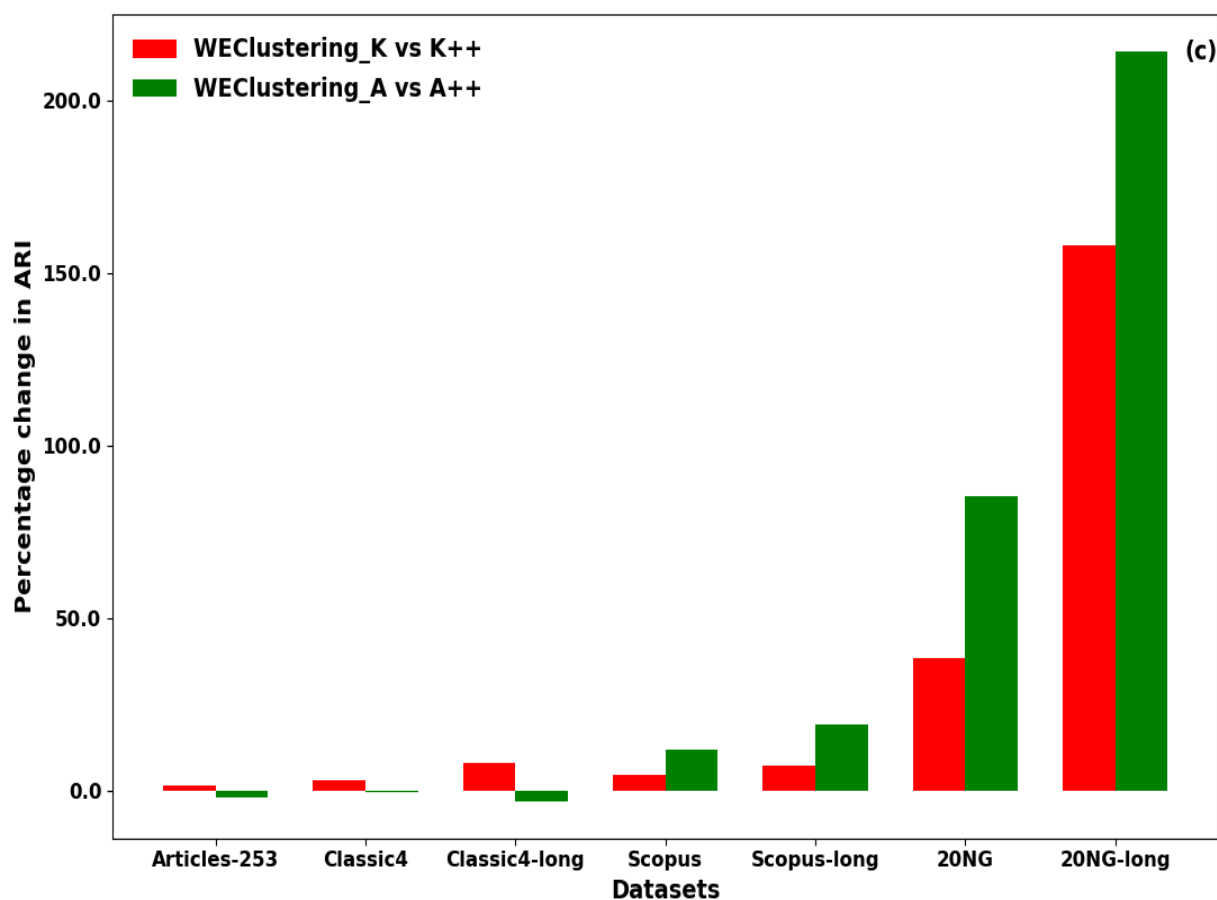
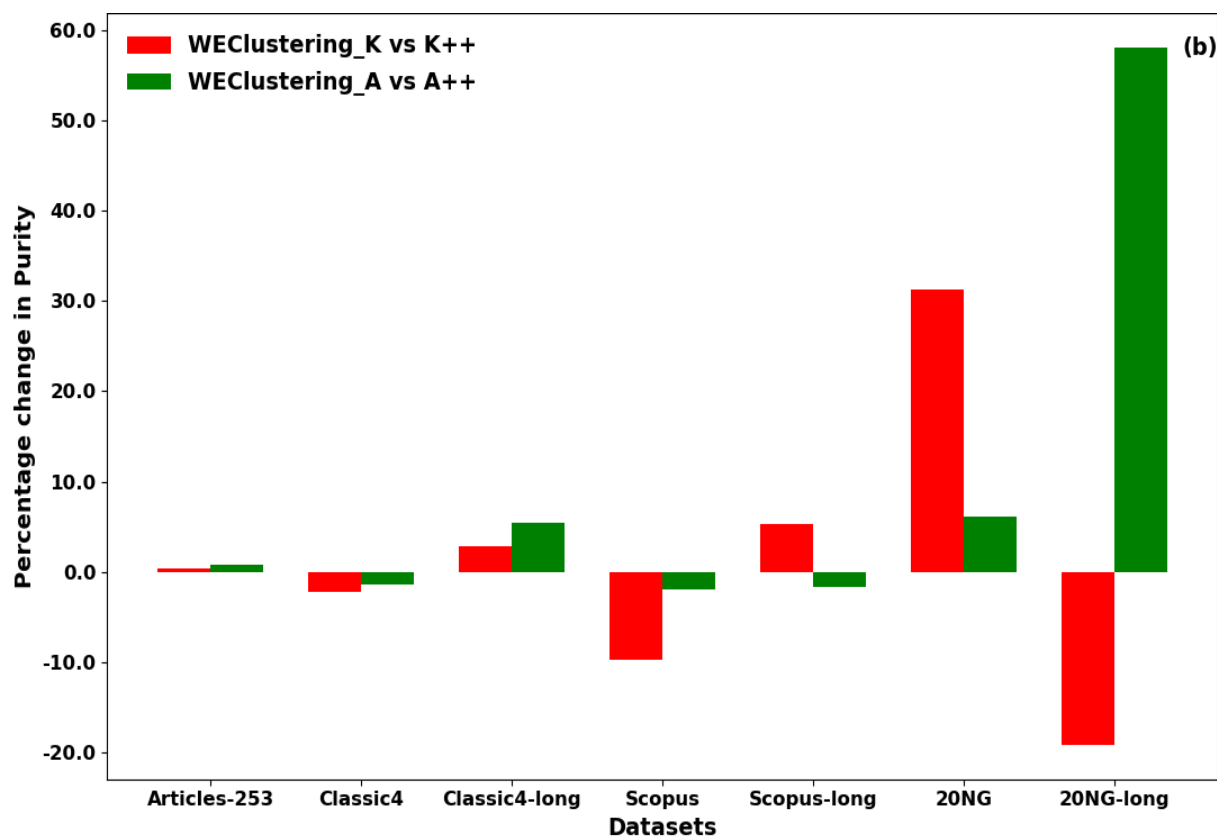


Figure 6. Percentage variations in (a) silhouette score, (b) purity, and (c) ARI of WEClustering_K [15] vs WEClustering_K++ and WEClustering_A [15] vs WEClustering_A++.

4. Conclusions

The proposed WEClustering++ models significantly improve document clustering by integrating finetuned BERT embeddings, optimized dimensionality reduction, and enhanced clustering techniques. The model effectively captures semantic nuances by leveraging domain-specific contextual embeddings, leading to improved clustering quality compared to traditional and existing state-of-the-art methods. Experimental results across multiple benchmark datasets demonstrate that WEClustering++ achieves higher silhouette scores, purity, and ARI, highlighting its ability to generate well separated and semantically meaningful clusters. The observed performance gains validate the importance of contextualized word embeddings and optimized clustering techniques in large-scale text mining tasks. The Proposed model may be further expanded to multilingual datasets, real-time clustering applications, and integration with adaptive learning models to further enhance scalability and efficiency.

Abbreviations

ARI	Adjusted Rand Index
BERT	Bidirectional Encoder Representations from Transformers
IDF	Inverse Document Frequency
TF	Term Frequency
WEClustering++	Word Embeddings Clustering++

Acknowledgments

Authors would like to thank Shri. Y Dilip, Director, Aeronautical Development Establishment, Mr. Manjunath S M, Technology Director and Mr. Diptiman Biswas, Group Director for their support during the research work carried out at ADE, DRDO.

Author Contributions

Vijay Kumar Sutrarakar: Conceptualization, Methodology, Supervision, Writing – review & editing

Nikhil Mogre: Conceptualization, Formal Analysis, Software, Validation, Visualization, Writing – original draft

Conflicts of Interest

The authors declare no conflicts of interest.

References

- [1] Novel coronavirus resource directory (2020) Accessed Feb 08, 2025 <https://doi.org/10.2172/1602724>
- [2] Johnson R, Watkinson A, Mabe M (2018) The stm report. An overview of scientific and scholarly publishing, 5th Ed. <https://doi.org/10.1017/9780511862489.019>
- [3] Manning CD, Schütze H (1999) Foundations of statistical natural language processing. MIT press, New York <https://doi.org/10.1162/coli.2000.26.2.277>
- [4] Robertson S (2004) Understanding inverse document frequency: on theoretical arguments for idf. *J Doc* 60(5): 503–520 <https://doi.org/10.1108/00220410410560582>
- [5] Hotho A, Staab S, Stumme G (2003) Ontologies improve text document clustering. In: Third IEEE international conference on data mining, pp 541–544. IEEE. <https://doi.org/10.1109/icdm.2003.1250972>
- [6] Mehta V, Bawa S, Singh J (2021) Stamantic clustering: combining statistical and semantic features for clustering of large text datasets. *Expert Syst Appl* 174: 114710 <https://doi.org/10.1016/j.eswa.2021.114710>
- [7] Sedding J, Kazakov D (2004) Wordnet-based text document clustering. In: proceedings of the 3rd workshop on robust methods in analysis of natural language data. Association for Computational Linguistics, pp 104–113 <https://doi.org/10.3115/1621445.1621458>
- [8] Wei T, Lu Y, Chang H, Zhou Q, Bao X (2015) A semantic approach for text clustering using wordnet and lexical chains. *Expert Syst Appl* 42(4): 2264–2275 <https://doi.org/10.1016/j.eswa.2014.10.023>
- [9] Mikolov T, Chen K, Corrado G, Dean J (2013) Efficient estimation of word representations in vector space. *arXiv preprint arXiv: 1301.3781* <https://doi.org/10.32614/cran.package.fasttext>
- [10] Miller GA (1995) Wordnet: a lexical database for English. *Commun ACM* 38(11): 39–41 <https://doi.org/10.1145/219717.219748>
- [11] Chang WC, Yu HF, Zhong K, Yang Y, Dhillon I (2019) Xbert: extreme multi-label text classification with using bidirectional encoder representations from transformers. *arXiv preprint arXiv: 1905.02331* <https://doi.org/10.1145/3394486.3403368>
- [12] Pennington J, Socher R, Manning CD (2014) Glove: global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), pp 1532–1543 <https://doi.org/10.3115/v1/d14-1162>
- [13] Bojanowski P, Grave E, Joulin A, Mikolov T (2017) Enriching wordvectors with subword information. *Trans Assoc Comput Linguist* 5: 135–146. https://doi.org/10.1162/tacl_a_00051
- [14] Devlin J, Chang MW, Lee K, Toutanova K (2018) Bert: pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv: 1810.04805* <https://doi.org/10.20944/preprints202411.2377.v1>

- [15] Mehta, V., Bawa, S., & Singh, J. (2021). WEClustering: Word embeddings based text clustering technique for large datasets. *Complex & Intelligent Systems*, 7(6), 3211–3224. <https://doi.org/10.1007/s40747-021-00512-9>.
- [16] Almeida F, Xexéo G (2019) Word embeddings: a survey. arXiv preprint arXiv: 1901.09069 <https://doi.org/10.32614/cran.package.sweater>
- [17] Bakarov A (2018) A survey of word embeddings evaluation methods. arXiv preprint arXiv: 1801.09536 <https://doi.org/10.32614/cran.package.sweater>
- [18] Camacho-Collados J, Pilehvar MT (2018) From word to sense embeddings: a survey on vector representations of meaning. *J Artif Intell Res* 63: 743–788 <https://doi.org/10.1613/jair.1.11259>
- [19] Wang S, Zhou W, Jiang C (2020) A survey of word embeddings based on deep learning. *Computing* 102(3): 717–740 <https://doi.org/10.1007/s00607-019-00768-7>
- [20] Fränti P, Sieranoja S (2018) K-means properties on six clustering benchmark datasets. *Appl Intell* 48(12): 4743–4759 <https://doi.org/10.1007/s10489-018-1238-7>
- [21] Sculley D (2010) Web-scale k-means clustering. In: *Proceedings of the 19th international conference on World wide web*, pp 1177–1178 <https://doi.org/10.1145/1772690.1772862>
- [22] Han J, Pei J, Kamber M (2011) *Data mining: concepts and techniques*. Elsevier, Amsterdam <https://doi.org/10.1016/b978-0-12-381479-1.00007-1>
- [23] Nielsen F (2016) *Hierarchical clustering. Introduction to HPC with MPI for data science*. Springer, New York, pp 195–211 https://doi.org/10.1007/978-3-319-21903-5_8
- [24] Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv preprint arXiv: 1810.04805.
- [25] Arthur D, Vassilvitskii S (2006) k-means++: the advantages of careful seeding. Technical report, Stanford <https://doi.org/10.5220/0012045000003546>
- [26] Soares VH. Downloads. Available from: <https://vhasoares.github.io/downloads.html> (accessed 18 November 2024)
- [27] Langley J. 20 Newsgroups Dataset. Available from: <http://qwone.com/~jason/20Newsgroups/> (accessed 18 November 2024)
- [28] Shahapure KRS and Nicholas C (2020) Cluster Quality Analysis Using Silhouette Score, IEEE DSAA, <https://doi.org/10.1109/dsaa49011.2020.00096>
- [29] Santos JM, Embrechts M (2009) On the use of the adjusted rand index as a metric for evaluating supervised classification. *International conference on artificial neural networks*. Springer, New York, pp 175–184 https://doi.org/10.1007/978-3-642-04277-5_18
- [30] Manning CD, Schütze H, Raghavan P (2008) *Introduction to information retrieval*. Cambridge University Press, Cambridge <https://doi.org/10.1007/s10791-009-9096-x>