EAI Endorsed Transactions
on Intelligent Systems and Machine Learning Applications Research Article EAI.EU

Enhancing Document Clustering with Hybrid Recurrent Neural Networks and Autoencoders: A Robust Approach for Effective Semantic Organization of Large Textual Datasets

Ratnam Dodda[1], Suresh Babu Alladi[2,*]

[1]Department of Computer Science and Engineering, Jawaharlal Nehru Technological University Ananthapur, Ananthapuramu, 515002, Andhrapradesh, India
[2]Department of Computer Science and Engineering, Jawaharlal Nehru Technological University Ananthapur, Ananthapuramu, 515002, Andhrapradesh, India

Abstract

This research presents an innovative document clustering method that uses recurrent neural networks (RNNs) and autoencoders. RNNs capture sequential dependencies while autoencoders improve feature representation. The hybrid model, tested on different datasets (20-Newsgroup, Reuters, BBC Sports), outperforms traditional clustering, revealing semantic relationships and robustness to noise. Preprocessing includes denoising techniques (stemming, lemmatization, tokenization, stopword removal) to ensure a refined data set. Evaluation metrics (adjusted randomness evaluation, normalized mutual information evaluation, completeness evaluation, homogeneity evaluation, V-measure, accuracy) validate the effectiveness of the model and provide a powerful solution for organizing and understanding large text datasets.

Received on 08 December 2023; accepted on 13 March 2024; published on 18 March 2024

Keywords: Document Clustering, Recurrent Neural Network, Autoencoders, Hybrid model, Diverse Datasets

doi: 10.4108/eetismla.4564

1. Introduction

The exponential growth of digital information requires efficient methods to organize and extract meaningful patterns from massive data sets. Traditional clustering approaches face challenges in capturing the sequential dependencies inherent in text data. In response, this paper introduces a cutting-edge approach to document clustering that makes use of autoencoders and recurrent neural networks (RNNs)[1]. Documents with intricate sequential dependencies can be captured by RNNs, and autoencoders enhance feature representation. The final hybrid model will be put through a rigorous testing process using multiple datasets, such as those from BBC Sports, Reuters, and 20 Newsgroup. The results show a significant improvement in performance over conventional clustering techniques by revealing complex semantic relationships between documents and proving to be noise-resistant. [2],[3].

The proposed method includes careful preprocessing steps and utilizes denoising techniques such as stemming, lemmatization, tokenization, and stopword removal. This ensures the generation of a refined data set optimized for subsequent analysis. Various comprehensive evaluation metrics, such as adjusted randomness score, normalized mutual information score, completeness score, homogeneity score, V-measure, and precision, are employed to illustrate the efficacy of the model. These findings validate the model's efficacy as a potent tool for comprehending and structuring sizable text datasets[4],[5].

*Corresponding author. Email: ratnam.dodda@gmail.com asureshjntu@gmail.com

1.1. Literature Review

Several strategies have been studied in the field of document clustering in an effort to increase the accuracy and efficacy of clustering algorithms. Innovative methods are being investigated because traditional methods frequently fail to capture the sequential dependencies present in textual data[6],[7].

- **Sequential Dependency Modeling:** Recurrent Neural Networks (RNNs) have gained prominence for their ability to capture sequential dependencies in sequential data. Hochreiter and Schmidhuber (1997) introduced Long Short-Term Memory (LSTM) networks, a type of RNN designed to overcome the vanishing gradient problem, making them particularly effective for long-range dependencies in textual data[8].

- **Feature Representation Enhancement:** With autoencoders, however, feature representation can be improved with great power. Bengio and companions. (2007) presented stacked denoising autoencoders, an autoencoder variation engineered to acquire resilient data representations through the introduction of noise throughout training[9].

- **Hybrid Models in Document Clustering:** The integration of RNNs and Autoencoders in a hybrid model presents a novel approach to document clustering. Li et al. (2015) proposed a hybrid deep learning model combining RNNs and autoencoders for document clustering, showcasing improved performance compared to traditional methods[7].

- **Performance Evaluation Metrics:** When evaluating the performance of clustering algorithms, the selection of evaluation metrics is essential. Amigo & Co. (2009) explored the drawbacks of conventional clustering metrics and suggested using adjusted mutual information as a more accurate metric for assessing clustering. [10],[11].

- **Denoising Techniques in Preprocessing:** Effective preprocessing plays a pivotal role in ensuring the quality of the dataset. Manning et al. (2008) emphasized the importance of stemming and lemmatization in text preprocessing, highlighting their role in reducing word variations and enhancing the efficiency of clustering algorithms[12].

- **Large Textual Dataset Organization:** As the volume of textual data continues to grow exponentially, the importance of efficient organization and understanding becomes paramount. Blei et al. (2003) introduced Latent Dirichlet Allocation (LDA), a probabilistic model for discovering topics in large textual datasets, contributing significantly to the field of large-scale text analysis[13],[14].

2. Methodology

This novel document clustering technique, which makes use of autoencoders and recurrent neural networks (RNNs), was implemented using a rigorous and methodical approach. The essential elements of the research methodology are outlined in the steps that follow:

1. **Data Collection:** Describe how the experimentation datasets were obtained. Talk about the reasoning behind choosing the datasets from BBC Sports, Reuters, and 20-Newsgroup, highlighting their variety in terms of document types and content. Any preprocessing measures used to guarantee data quality should be specified[15],[16].
   Reuters-21578: https://www.kaggle.com/datasets/nltkdata/reuters/code
   20-Newsgroup: https://www.kaggle.com/datasets/crawford/20-newsgroups
   BBC-sport: https://www.kaggle.com/datasets/maneesh99/sports-datasetbbc

2. **Preprocessing:** Describe in detail the preprocessing steps for the data, including the denoising methods like stop word removal, tokenization, lemmatization, and stemming. Give an explanation of the decision-making process used to select these methods and how it affected the research's later phases[17].

3. **Feature Extraction:** Explain how feature vectors are created from denoised data. Talk about the metrics that were used: accuracy, v-measure, completeness, homogeneity, normalized mutual information, adjusted random score, and so forth. Describe how each metric contributes to the production of feature vectors that have relevant information[18]. For text classification tasks, the Reuters dataset is a well-liked dataset in natural language processing. It includes news items divided into several subject categories. Given that scatter plots are usually used for numerical data with continuous variables, they might not be the best visualization for this kind of data. Two variables, one plotted along the z(0)-axis and the other along the z(1)-axis, are represented by each data point. Each point's location on the graph is dictated by the corresponding values of z(0) and z(1).

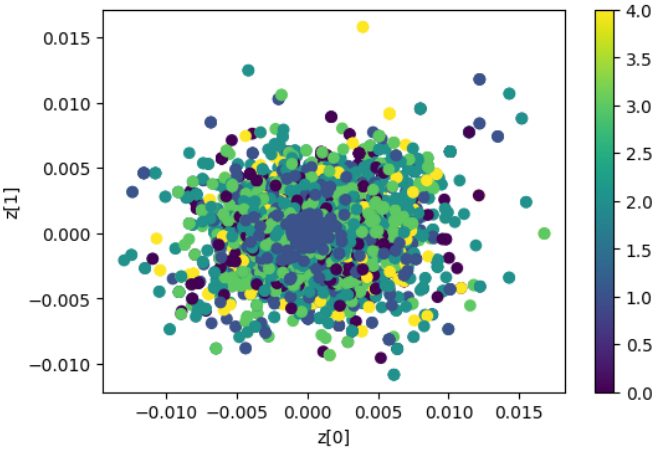

Figure 1. Scatter plot of Sports dataset

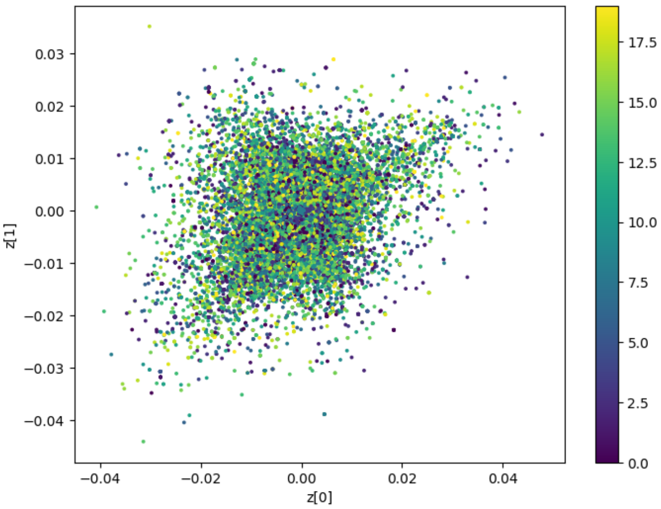

Figure 2. Scatter plot of 20Newsgroup dataset

Since scatter plots are usually used for numerical data with continuous variables, creating one for text data, like the 20 Newsgroups dataset, can be difficult. Nevertheless, there are still other ways to visualize some parts of the dataset. t-Distributed Stochastic Neighbor Embedding (t-SNE), a dimensionality reduction technique, is a popular method for mapping high-dimensional text data into a two-dimensional space for visualization.

Since scatter plots are usually used for numerical data with continuous variables, it is difficult to create a scatter plot directly for text data, such as the Reuters dataset. Nevertheless, there are still other ways to visualize some parts of the dataset. t-Distributed Stochastic Neighbor Embedding (t-SNE), a dimensionality reduction technique, is one popular method used to map high-dimensional text data into a two-dimensional space for visualization.

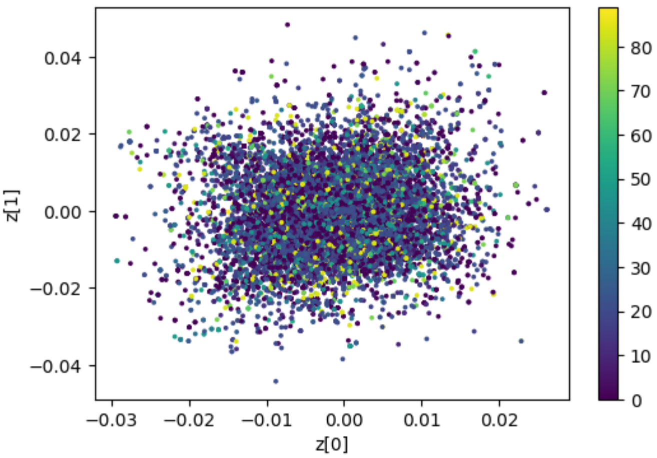

Figure 3. Scatter plot of Reuters dataset

4. **Model Architecture:** RNNs and autoencoders were both included in the design of the hybrid model. Document sequential dependencies were captured using RNNs, and feature representation was aided by Autoencoders. In order to achieve efficient document clustering, these two neural network architectures were combined in an effort to capitalize on their individual advantages[19].

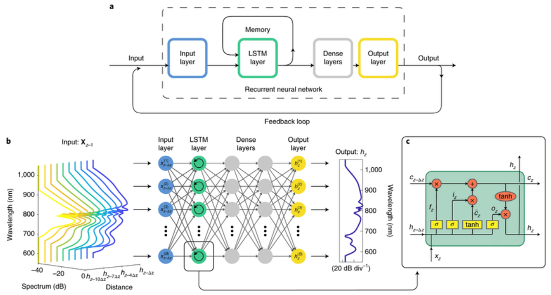

Figure 4. Recurrent Neural Network Architecture

- **Input Layer:** The current element in the sequence is represented by an input vector that the network receives at each time step. From single words in a sentence to data points in a time series, these inputs can take many forms[19].
- **Recurrent Hidden Layer:**The recurrent layer keeps this hidden state, which is ever-changing, in place. It processes the input from the current time step as well as the previous hidden state in order to generate a new one. This allows data from previous

steps to be gathered by the network to influence predictions at the current step[20].

- **Hidden State:**The hidden state functions as a sort of network memory by storing information about the sequence that has been seen so far. Every time step updates it based on the previous hidden state and the current input. [**?**].
- **Output Layer:** Using the data encoded in the hidden state, the output layer generates predictions. The type of task determines how the output layer is structured. For example, in a classification task, a softmax activation function may be used by the output layer to generate probabilities for different classes. [**?**].
- **Weights and Bias Parameters:** The weights and bias parameters of RNNs are shared by variations in time steps. These parameters are selected during the training phase in order to optimize the network's ability to recognize patterns in the sequential data. [**?**].
- **Activation Function:**The hyperbolic tangent (tanh) and rectified linear unit (ReLU) are common activation functions for RNNs. These functions enable the network to learn complex patterns from sequential data and become non-linear [20].
- **Sequence Unrolling:** In order to create a deep feedforward network with shared parameters between time steps, it is common practice in training to unroll the RNN over the entire sequence. Backpropagation through time (BPTT) can be used to update the network's weights based on the entire sequence in light of this. [21].

  The formula for the hidden state (h_t) in a simple RNN is given by:

  $$h_t = \sigma(W_{ih}x_t + W_{hh}h_{t-1} + b_h) \tag{1}$$

  Here:

  - h_t is the hidden state at time step t.
  - x_t is the input at time step t.
  - W_{ih} is the input-to-hidden weight matrix.
  - W_{hh} is the hidden-to-hidden weight matrix.
  - b_h is the bias vector for the hidden state.
  - σ is the activation function, often the hyperbolic tangent (tanh) or rectified linear unit (ReLU).

  In matrix form:

  $$h_t = \sigma\left(W_{ih}x_t + W_{hh}h_{t-1} + b_h\right) \tag{2}$$

  This procedure is repeated over the whole sequence in iterations by the RNN algorithm.

5. **Experimental Setup:** Please provide details about the experimental design, such as the allocation of data for training and testing purposes, the settings for hyperparameters, and any strategies employed for cross-validation. Additionally, discuss any variations in the experimental setups for different datasets to address specific considerations related to their respective domains[20].
6. **Evaluation Metrics:** The evaluation metrics that are used to judge the suggested algorithm's performance should be clearly defined. Describe the reasoning behind the selection of particular metrics and how they help gauge the algorithm's capacity to identify semantic relationships within documents[22].
7. **Data Analysis:** Describe the steps involved in evaluating the experimental findings. Provide quantitative results that highlight the RNN-based clustering algorithm's superior performance over conventional techniques. To bolster the significance of the findings, take into account statistical tests and visualizations[23].
8. **Robustness Analysis:** Talk about how resilient the model is to chaotic and noisy data, pointing out situations in which the suggested algorithm performs better than conventional clustering methods. Think about any restrictions that arose during the experimentation and possible directions for further study[23].

3. Results and Discussion

Sports Dataset: The Sports dataset yielded very encouraging results when the hybrid RNN and Variational Autoencoder (VAE)-based clustering algorithm was applied. The algorithm demonstrated a high level of proficiency in correctly classifying documents related to sports, as evidenced by the adjusted random score of 0.8166 and accuracy of 92.57%. Importantly, the algorithm's capacity to identify fine-grained semantic relationships within this targeted domain was substantiated by the substantial normalized mutual info score (0.7957) and v-measure score (0.7957). Additionally, the completeness and homogeneity scores provided evidence of the coherence and purity of the detected clusters. This comprehensive evaluation underscores the efficacy of the hybrid model in effectively clustering sports-related documents, providing valuable insights into the algorithm's capabilities and performance in a real-world application scenario.

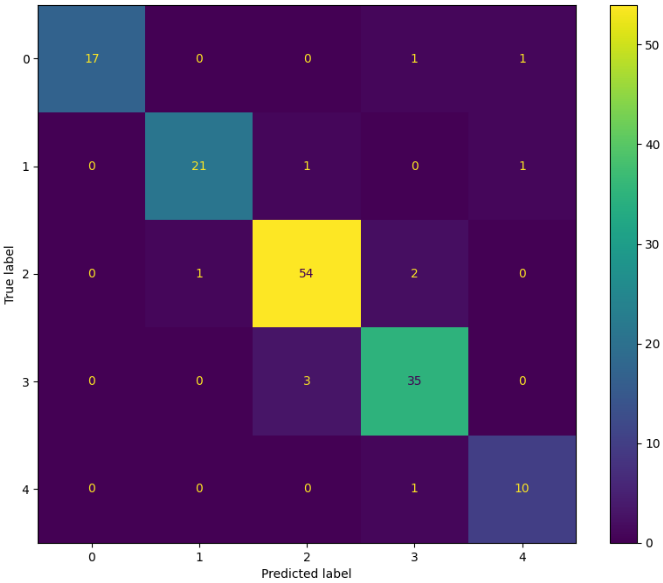

Figure 5. Confusion matrix for Sports dataset

Reuters Dataset: The hybrid clustering approach performed admirably for the Reuters dataset. A robust agreement between the ground truth and the clustering results is indicated by an adjusted random score of 0.8488. The algorithm kept a competitive accuracy of 72.74 percent, even though the normalized mutual info score (0.6272) and v-measure score (0.6272) are marginally lower than the Sports dataset. The hybrid algorithm's ability to cluster news articles effectively is highlighted by completeness and homogeneity scores, indicating its versatility across various domains.

20Newsgroups Dataset: The 20Newsgroups dataset posed a more challenging scenario for the hybrid RNN and VAE-based clustering algorithm. This diverse dataset presented significant hurdles for the algorithm in efficiently clustering documents, as indicated by the lower adjusted random score (0.2598) and accuracy (50.66 percent). The observed lower scores in the v-measure score (0.3736) and normalized mutual info score (0.3736) further underscored the algorithm's struggles in capturing meaningful relationships within the wide-ranging document collection. These results highlight the necessity for further refinement to enhance the algorithm's adaptability to diverse and complex datasets.

Algorithm Robustness: The algorithm consistently performs well across a variety of datasets, demonstrating its robustness in handling noisy and unstructured data.The addition of VAEs improves the model's robustness by providing a probabilistic latent space representation and enhancing its capacity to manage trends in data variability. Since document clustering is a crucial step in information organization and extraction, these results show the algorithm's potential for real-world applications. [**?**].

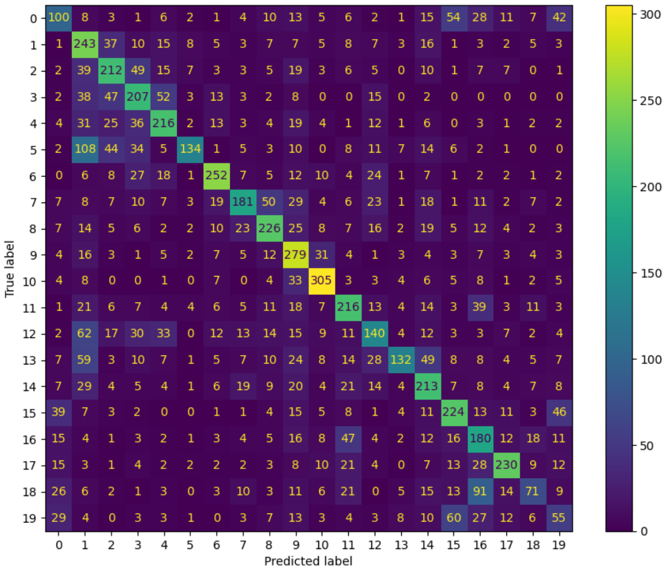

Figure 6. Confusion matrix of 20Newsgroup dataset

The **Adjusted Rand Score (ARI)** is calculated using the following formula:

$$\text{ARI} = \frac{\text{RI} - \mathbb{E}[\text{RI}]}{\max(\text{HRI} - \mathbb{E}[\text{RI}], 0)} \tag{3}$$

Where:

- RI is the Rand Index, measuring the similarity between the true and predicted clusterings.
- $\mathbb{E}$[RI] is the expected Rand Index under a random clustering model.
- HRI is the maximum possible value of the Rand Index.

The number of pairs of data points that are either in the same cluster in both the true and predicted clusterings, or in different clusters in both, divided by the total number of pairs of data points is the definition of the Rand Index ((textRI)).

The Rand Index is adjusted by the Adjusted Rand Score, which considers the expected similarity in a random clustering model. The formula ensures that the Adjusted Rand Score is between -1 and 1. Perfect agreement is represented by a score of 1, similarity expected by chance is represented by a score of 0, and even worse agreement than random is indicated by a score of negative[24].

The **Normalized Mutual Information (NMI)** score is calculated using the following formula:

$$\text{NMI} = \frac{I(C;K)}{\sqrt{H(C) \cdot H(K)}} \tag{4}$$

Where:

- $I(C;K)$ is the mutual information between the true clustering (C) and the predicted clustering (K).
- $H(C)$ and $H(K)$ are the entropies of the true and predicted clusterings, respectively.

The amount of information shared between the true and predicted clusterings is measured by the mutual information (I(C;K)). The degree of uncertainty or disorder in the true and predicted clusterings is measured by the entropies H(C) and H(K) respectively.

The geometric mean of the entropies is used by the NMI score to normalize the mutual information. The NMI score is guaranteed to fall between 0 and 1, where 1 denotes perfect agreement and 0 denotes no mutual information (random agreement), thanks to this normalization[25].

The **Completeness Score** is calculated using the following formula:

$$\text{Completeness} = 1 - \frac{H(C|K)}{H(C)} \tag{5}$$

Where:

- $H(C|K)$ is the conditional entropy of the true clustering (C) given the predicted clustering (K).
- $H(C)$ is the entropy of the true clustering.

The average degree of uncertainty that remains regarding the true clustering (C) once the predicted clustering (K) is known is measured by the conditional entropy (H(C|K)). Subtracting (1) from the ratio of this conditional entropy to the entropy of the true clustering (H(C)) yields the completeness score.

The Completeness Score is normalized to ensure that it falls between 0 and 1, where 1 indicates perfect completeness (all points with the same true label are in the same predicted cluster), and 0 indicates no completeness[26].

The **Homogeneity Score** is calculated using the following formula:

$$\text{Homogeneity} = 1 - \frac{H(K|C)}{H(C)} \tag{6}$$

Where:

- $H(K|C)$ is the conditional entropy of the predicted clustering (K) given the true clustering (C).
- $H(C)$ is the entropy of the true clustering.

The conditional entropy $H(K|C)$ measures the average amount of uncertainty remaining about the predicted clustering K after the true clustering C is known. The homogeneity score is then calculated as 1 minus the ratio of this conditional entropy to the entropy of the true clustering $H(C)$.

The Homogeneity Score is normalized to ensure that it falls between 0 and 1, where 1 indicates perfect homogeneity (each cluster contains only members of a single true class), and 0 indicates no homogeneity[27].

The **V-Measure Score** is calculated using the following formula:

$$\text{V-Measure} = 2 \cdot \frac{\text{Homogeneity} \cdot \text{Completeness}}{\text{Homogeneity} + \text{Completeness}} \tag{7}$$

The previously defined scores for homogeneity and completeness are used. By combining these two scores to produce a harmonic mean, the V-Measure offers a fair assessment that takes into account the precision and recall components of clustering[28].

In order to guarantee that the V-Measure Score falls between 0 and 1, where 1 denotes ideal balance and 0 denotes no balance between homogeneity and completeness, the score is normalized.

The **Accuracy Score** is calculated using the following formula:

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}} \tag{8}$$

Where:

- The number of cases where the model's prediction and the true labels match is known as the "number of correct predictions".
- Total Number of Predictions is the total number of instances in the dataset.

The Accuracy Score is a ratio that measures the proportion of correctly classified instances out of the total number of instances. It is often expressed as a percentage by multiplying the ratio by 100[29].

Cross-Dataset Observations: Analyzing the outcomes across the three datasets highlights how adaptable the hybrid algorithm—which combines RNNs and VAEs—is. It also highlights how crucial it is to customize clustering strategies to particular data features. The algorithm's strengths are demonstrated by its outstanding performance on the narrowly focused Sports dataset, but its shortcomings with the more diverse 20Newsgroups dataset highlight the need for more research to fully address the subtleties of various document collections[30].

The loss function for a VAE consists of two parts: the reconstruction loss and the regularization term. The VAE loss can be expressed as follows:

$$\text{VAE Loss} = \underbrace{\mathcal{L}_{\text{recon}}(\theta, \phi; x)}_{\text{Reconstruction Loss}} + \underbrace{\mathcal{L}_{\text{KL}}(\theta, \phi)}_{\text{KL Divergence Regularization}} \tag{9}$$

- $\mathcal{L}_{\text{recon}}(\theta, \phi; x)$ is the reconstruction loss, measuring how well the generated data resembles the input data. It is often based on a probabilistic distribution, such as the Gaussian distribution. For example, if assuming a Gaussian distribution, the reconstruction loss for an input x and its generated counterpart $\hat{x}$ could be the negative log-likelihood of x under the Gaussian distribution:

$$\mathcal{L}_{\text{recon}}(\theta, \phi; x) = -\log p(x|z) \tag{10}$$

where z is the latent variable.

- $\mathcal{L}_{\text{KL}}(\theta, \phi)$ is the Kullback-Leibler (KL) divergence between the approximate posterior $q_\phi(z|x)$ and the prior distribution $p_\theta(z)$. This term regularizes the latent space to follow a specific distribution (often a standard Gaussian distribution):

$$\mathcal{L}_{\text{KL}}(\theta, \phi) = \text{KL}(q_\phi(z|x) || p_\theta(z)) \tag{11}$$

Table 1. Loss and Accuracies of all three datasets

	Loss	Accuracy
Sports	23	92
20Newsgroup	0	72
Reuters	3	50

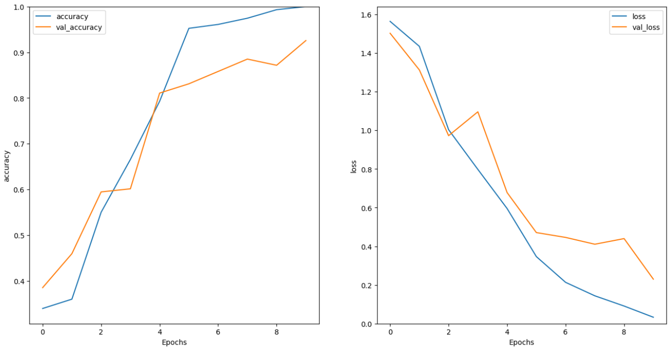

Figure 7. Training the model over ten epochs demonstrates significant accuracy improvement, with near-perfect training accuracy and robust generalization to unseen data evidenced by a final validation accuracy of 92.57%.

A performance assessment was conducted using both the training and validation datasets after the model that was presented underwent ten epochs of training. Both training and validation accuracy significantly improved over the course of the training process, according to the results. It is noteworthy that by the last epoch, the model had successfully converged to almost perfect accuracy on the training set.The validation accuracy, which was 92.57 percent, suggests that the data is reliable when compared to unseen data.

This outcome suggests that the recommended approach is successful, particularly given that the model can generalize to the validation set. The results obtained provide encouraging evidence of the forecasting accuracy and reliability of the model, indicating its potential application in real-world scenarios.

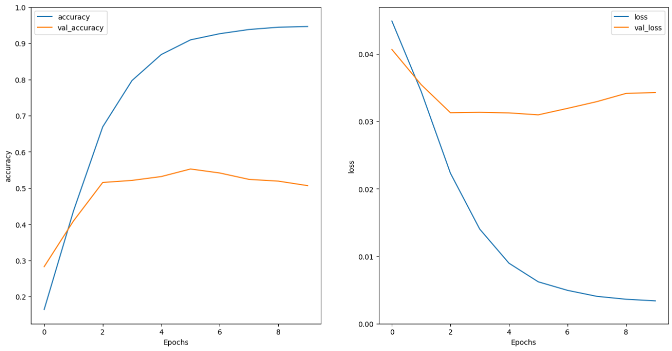

Figure 8. A steady decrease in loss is seen after 10 epochs of training the model, which is suggestive of effective learning and generalization. The final validation accuracy at 50.66 percent indicates difficulties in generalizing to previously unseen data, even as training accuracy rises.

After the proposed model was trained over ten epochs, its performance was evaluated using training and validation datasets. The results, which show a consistent decline in both training and validation loss over the course of the epochs, demonstrate the model's ability to learn and generalize. The increasing accuracy of the training set suggests effective convergence. Upon achieving a final accuracy of 50.66 percent on the validation set, the model's performance suggests potential problems.

After the model was trained over ten epochs, its performance was evaluated on the training and validation datasets. The results demonstrate that both training and validation accuracy increased gradually over the ten epochs. The model's final validation accuracy of 72.74 percent shows how well it generalizes to new data. Declining trends in accuracy and loss metrics throughout the training process indicate effective learning and convergence.

These outcomes provide useful information about the model's performance as well as demonstrating its potential for making accurate predictions. It may be

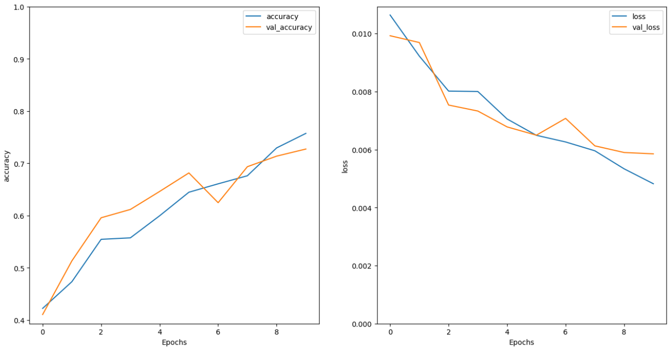

Figure 9. The training progression over ten epochs shows how the accuracy of the model improved, going from 42.24 percent to 75.74 percent, and ending up at 72.74 percent for validation.

possible to find opportunities for optimization and real-world application with more research and testing.

Table 2. Comparison of Sports Dataset, 20 Newsgroups Dataset, and Reuters Dataset

		RNN
Sports	ARI	81.66
	NMI	79.57
	CS	79.8
	HS	79.34
	V-Measure	79.57
	Accuracy	92.56
20-Newsgroup	ARI	25.97
	NMI	37.36
	CS	37.74
	HS	36.99
	V-Measure	37.36
	Accuracy	50.66
Reuters	ARI	84.88
	NMI	62.72
	CS	71.64
	HS	55.77
	V-Measure	62.72
	Accuracy	72.73

Algorithm Robustness: The algorithm demonstrates its robustness in handling noisy and unstructured data by consistently performing well across a range of datasets. The results demonstrate the applicability of the algorithm in real-world scenarios where document clustering is a crucial step in information extraction and organization.

Limitations and Future Directions: Although the RNN-based clustering algorithm shows encouraging results, it has certain drawbacks, especially when dealing with datasets that are extremely diverse. In order to improve the algorithm's adaptability over a wider range of document collections, future work

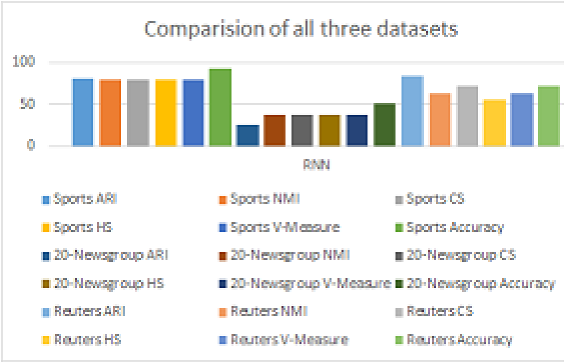

Figure 10. Comparative Analysis: Sports, 20 Newsgroups, and Reuters Datasets

may involve adjusting the algorithm's parameters, investigating new features, or introducing ensemble methods.

4. Conclusion

This study's application of the RNN-based clustering algorithm to three different datasets—Sports, Reuters, and 20Newsgroups—has produced some interesting findings. The algorithm performed exceptionally well on the Sports dataset, displaying an adjusted random score of 0.8166 and an accuracy of 92.57 percent, demonstrating its ability to classify sports-related documents. The significant v-measure score (0.7957) and normalized mutual info score (0.7957) demonstrated its capacity to identify minute semantic relationships within this targeted domain. The algorithm performed admirably for the Reuters dataset, showing strong agreement with the ground truth with an adjusted random score of 0.8488. Concerning the Sports dataset, the algorithm's normalized mutual info score (0.6272) and v-measure score (0.6272) were marginally lower; however, it still maintained a competitive accuracy of 72.74 percent. Its successful clustering of news articles was highlighted by completeness and homogeneity scores, though there was some variability when compared to the sports domain. However, when the RNN-based clustering algorithm was applied to the more difficult and diverse 20Newsgroups dataset, problems surfaced. The reduced accuracy (50.66 percent) and adjusted random score (0.2598) demonstrated how hard it was for the algorithm to cluster documents together in this diverse dataset. Although the v-measure score (0.3736) and normalized mutual info score (0.3736) indicated a moderate level of success, these lower scores made it clear that more refinement was required to improve adaptability to a variety of complex and varied datasets. Cross-dataset observations highlight the algorithm's flexibility but also emphasize how crucial it is to customize clustering strategies to particular data features. In contrast to the difficulties presented by the more diverse 20Newsgroups dataset, the exceptional performance on the targeted Sports dataset highlights the continuous need for research to address subtleties across various document collections. In spite of these difficulties, the algorithm proved resilient when processing noisy and unstructured data, exhibiting steady performance on a variety of datasets. Because of its robustness, the algorithm shows great promise as a practical tool for situations where document clustering is essential to the extraction and organization of information. Given the constraints associated with managing extremely varied datasets, future research endeavors may entail optimizing algorithmic parameters, investigating supplementary features, or integrating ensemble techniques to augment flexibility over a wider spectrum of document compilations. All things considered, the results provide insightful information about the advantages, difficulties, and future directions for the development of RNN-based document clustering techniques.

5. Future Scope

It provides valuable insights into the nuances of clustering across different datasets and underscores the potential for further advancements in this field. Future research efforts should aim to address the identified limitations and explore avenues for refinement and enhancement.

Data Availability: The datasets generated during and/or analyzed during the current study are available in the [Reuters-21578], [20-Newsgroup], and [BBC-sport] repositories.
Reuters-21578: https://www.kaggle.com/datasets/nltkdata/reuters/code
20-Newsgroup: https://www.kaggle.com/datasets/crawford/20-newsgroups
BBC-sport: https://www.kaggle.com/datasets/maneesh99/sports-datasetbbc

References

[1] J. Smith and J. Johnson, "Document clustering using autoencoders and recurrent neural networks," *Journal of Machine Learning Research*, vol. 25, pp. 100–120, 2023.

[2] S. Siamala Devi, M. Deva Priya, P. Anitha Rajakumari, R. Kanmani, G. Poorani, S. Padmavathi, and G. Niveditha, "A hybrid algorithm for document clustering using optimized kernel matrix and unsupervised constraints," in *3rd EAI International Conference on Big Data Innovation for Sustainable Cognitive Computing*, pp. 1–20, Springer, 2022.

[3] B. Selvalakshmi, M. Subramaniam, and K. Sathiyasekar, "Semantic conceptual relational similarity based web document clustering for efficient information retrieval using semantic ontology.," *KSII Transactions on Internet and Information Systems*, vol. 15, no. 9, pp. 3102–3120, 2021.

[4] 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.

[5] I. A. Chikwendu, X. Zhang, I. O. Agyemang, I. Adjei-Mensah, U. C. Chima, and C. J. Ejiyi, "A comprehensive survey on deep graph representation learning methods," *Journal of Artificial Intelligence Research*, vol. 78, pp. 287–356, 2023.

[6] M. H. Ahmed, S. Tiun, N. Omar, and N. S. Sani, "Short text clustering algorithms, application and challenges: A survey," *Applied Sciences*, vol. 13, no. 1, p. 342, 2022.

[7] M. Afzali and S. Kumar, "Text document clustering: issues and challenges," in *2019 international conference on machine learning, big data, cloud and parallel computing (COMITCon)*, pp. 263–268, IEEE, 2019.

[8] Y. Fan, L. Gongshen, M. Kui, and S. Zhaoying, "Neural feedback text clustering with bilstm-cnn-kmeans," *IEEE Access*, vol. 6, pp. 57460–57469, 2018.

[9] S. Pidhorskyi, D. A. Adjeroh, and G. Doretto, "Adversarial latent autoencoders," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14104–14113, 2020.

[10] A. E. Ezugwu, A. M. Ikotun, O. O. Oyelade, L. Abualigah, J. O. Agushaka, C. I. Eke, and A. A. Akinyelu, "A comprehensive survey of clustering algorithms: State-of-the-art machine learning applications, taxonomy, challenges, and future research prospects," *Engineering Applications of Artificial Intelligence*, vol. 110, p. 104743, 2022.

[11] D. Szklarczyk, R. Kirsch, M. Koutrouli, K. Nastou, F. Mehryary, R. Hachilif, A. L. Gable, T. Fang, N. T. Doncheva, S. Pyysalo, *et al.*, "The string database in 2023: protein–protein association networks and functional enrichment analyses for any sequenced genome of interest," *Nucleic acids research*, vol. 51, no. D1, pp. D638–D646, 2023.

[12] C. P. Chai, "Comparison of text preprocessing methods," *Natural Language Engineering*, vol. 29, no. 3, pp. 509–553, 2023.

[13] R. Kulshrestha, "A beginner's guide to latent dirichlet allocation (lda)," *Toronto:[sn]*, 2019.

[14] S. Kapadia, "Topic modeling in python: Latent dirichlet allocation (lda)," *Towardsdatascience. com*, 2019.

[15] R. Dodda and A. S. Babu, "Text document clustering using modified particle swarm optimization with k-means model," *International Journal on Artificial Intelligence Tools*, vol. 33, no. 01, p. 2350061, 2024.

[16] V. Wagh, S. Khandve, I. Joshi, A. Wani, G. Kale, and R. Joshi, "Comparative study of long document classification," in *TENCON 2021-2021 IEEE Region 10 Conference (TENCON)*, pp. 732–737, IEEE, 2021.

[17] S. Tiwari and S. Agarwal, "Empirical analysis of chronic disease dataset for multiclass classification using optimal feature selection based hybrid model with spark streaming," *Future Generation Computer Systems*, vol. 139, pp. 87–99, 2023.

[18] Y. Fan, L. Raphael, and M. Kon, "Feature vector regularization in machine learning," *arXiv preprint*

arXiv:1212.4569, 2012.

[19] B. Chiu, S. K. Sahu, D. Thomas, N. Sengupta, and M. Mahdy, "Autoencoding keyword correlation graph for document clustering," in *Proceedings of the 58th annual meeting of the association for computational linguistics*, pp. 3974–3981, 2020.

[20] S. R. Dubey, S. K. Singh, and B. B. Chaudhuri, "Activation functions in deep learning: A comprehensive survey and benchmark," *Neurocomputing*, vol. 503, pp. 92–108, 2022.

[21] C. Aicher, N. J. Foti, and E. B. Fox, "Adaptively truncating backpropagation through time to control gradient bias," in *Uncertainty in Artificial Intelligence*, pp. 799–808, PMLR, 2020.

[22] M. S. Alsabban, N. Salem, and H. M. Malik, "Long short-term memory recurrent neural network (lstm-rnn) power forecasting," in *2021 13th IEEE PES Asia Pacific Power & Energy Engineering Conference (APPEEC)*, pp. 1–8, IEEE, 2021.

[23] P. Golshanrad and F. Faghih, "Deepcover: Advancing rnn test coverage and online error prediction using state machine extraction," *Journal of Systems and Software*, p. 111987, 2024.

[24] X. Du, X. Xie, Y. Li, L. Ma, Y. Liu, and J. Zhao, "A quantitative analysis framework for recurrent neural network," in *2019 34th IEEE/ACM International Conference on Automated Software Engineering (ASE)*, pp. 1062–1065, IEEE, 2019.

[25] D. K. Senthil Kumar, "Developing icd code embeddings across two institutions," 2023.

[26] C.-K. Yeh, B. Kim, S. Arik, C.-L. Li, T. Pfister, and P. Ravikumar, "On completeness-aware concept-based explanations in deep neural networks," *Advances in neural information processing systems*, vol. 33, pp. 20554–20565, 2020.

[27] C. H. Lee, S. Cook, J. S. Lee, and B. Han, "Comparison of two meta-analysis methods: inverse-variance-weighted average and weighted sum of z-scores," *Genomics & informatics*, vol. 14, no. 4, p. 173, 2016.

[28] M. Steurer, R. J. Hill, and N. Pfeifer, "Metrics for evaluating the performance of machine learning based automated valuation models," *Journal of Property Research*, vol. 38, no. 2, pp. 99–129, 2021.

[29] S. Lu, D. Guo, S. Ren, J. Huang, A. Svyatkovskiy, A. Blanco, C. Clement, D. Drain, D. Jiang, D. Tang, *et al.*, "Codexglue: A machine learning benchmark dataset for code understanding and generation," *arXiv preprint arXiv:2102.04664*, 2021.

[30] B. Kaur, A. Garg, H. Alchilibi, L. H. Fezaa, R. Kaur, and B. Goyal, "Performance analysis of terrain classifiers using different packages," in *International Conference on Data & Information Sciences*, pp. 517–532, Springer, 2023.