Deep Representation Learning for Effective Clustering of Short Persian Texts

Mahdi Molaei
Computerized Intelligence Systems Laboratory, Department of Computer Engineering, University of Tabriz, Iran
m.molaei@tabrizu.ac.ir

Mohammad-Reza Feizi-Derakhshi
Computerized Intelligence Systems Laboratory, Department of Computer Engineering, University of Tabriz, Iran | College of Engineering, Uruk University, Baghdad, Iraq
mfeizi@tabrizu.ac.ir (corresponding author)

Ali-Akbar Rasooly
Software Development Laboratory, Pazhouhesh Afzar Farda Company (PAFCO), Tehran, Iran
rasooly@pafcoerp.com

Cina Motamed
Laboratoire PRISME, University of Orleans, France
cina.motamed@univ-orleans.fr

Received: 16 March 2025 | Revised: 7 April 2025 | Accepted: 12 April 2025

Licensed under a CC-BY 4.0 license | Copyright (c) by the authors | *DOI: https://doi.org/10.48084/etasr.10984*

ABSTRACT

Short text clustering poses several challenges because of the limited contextual information available, especially for low-resource languages such as Persian. This study proposes a novel deep clustering architecture that consists of an RNN-based autoencoder to learn the latent representation of the text in preserving the rich structural features. This architecture involves a second network, the Representation network, to maximize the existing distance between clusters, minimize the overall cluster overlapping, and improve clustering in the latent space. The two-phase training approach first involved training using autoencoder reconstruction loss and then jointly optimizing for improved cluster separation. Experiments with different embedding types were carried out, and the evaluation results showed that the proposed method outperformed previous approaches. The proposed model provides an impactful advancement in representation learning and training for the short-text domain.

Keywords-short text clustering; RNN-based autoencoder; deep representation learning

I. INTRODUCTION

Clustering is an essential unsupervised learning task that involves grouping similar objects. It is a crucial area of research that has received significant attention, with various aspects such as distance metrics, feature selection, and grouping methods being extensively studied. Among the numerous clustering algorithms, K-Means [1], DBSCAN [2], and GMM [3] are very popular. These algorithms typically use distance-based similarity measures, focusing primarily on low-level features. However, when dealing with high-dimensional data, distance metrics become insufficient as distances between points become relatively uniform [4]. As a result, these algorithms are not well-suited for clustering high-dimensional data.

The success of deep learning models has led to many recent studies that focused on representing data well in both supervised and unsupervised settings. However, clustering, an unsupervised task, poses a challenge because no guidance, such as class labels, is available to guide the learning process. As a result, neural networks lack constraints to preserve the desired properties in the representations they learn. Most deep clustering methods rely on autoencoders that consist of two components: an encoder and a decoder that work together to reconstruct input data samples after encoding them in a latent space. The autoencoders aim to learn the key factors of similarity in the embedded space in relation to data semantics. All autoencoders share a reconstruction loss function, but they differ in the encoding operation. As mentioned above, the main challenge in unsupervised learning is the lack of an obvious

objective function that can cluster data points according to their semantic meaning. DEC [5] was a pioneer in addressing this challenge by introducing a self-training clustering loss scheme. This approach directly modifies the representation space to enhance its suitability for clustering. Later, many approaches were proposed to improve its structure, using methods such as preserving locality, different approaches to optimize the neural network with clustering loss, different embedding variants, various loss functions, and network structures [6-15]. This study aimed to design a simple and powerful model, leveraging autoencoders to learn an embedded space suitable for clustering. The focus lies specifically on short text data in Persian, which is considered a low-resource language. The aim was to create a model to efficiently separate data clusters, thereby improving data clusterability and accuracy.

The proposed clustering model utilizes an RNN-based autoencoder to learn data representations, which are then fed into a deep neural network, called the representation network, to create a clustering-friendly embedded space. This network is trained to minimize the cross-entropy between the pairwise similarity of points in the autoencoder's latent space and the embedded space. This is achieved by learning a mapping from the low-dimensional representation Z of the autoencoder to the embedding space E. The model is trained from end to end, simultaneously optimizing the encoder, decoder, and representation network parameters. Finally, K-Means clustering is applied to the embedding space to obtain the clusters. This work aimed to detect topics from short texts in Persian. However, extracting embeddings and clustering is extremely challenging for texts with fewer than 10 tokens. Texts of this short length cannot be encoded into embeddings with sufficient features to discern topics. Therefore, the texts were filtered to include only those between 10 and 30 tokens in length. This range was chosen to focus on short texts while still allowing topic detection through embeddings and clustering. The limitations of very short texts under 10 tokens prevented effective topic detection, but by focusing on texts with 10-30 tokens, more meaningful embeddings were produced and successfully clustered into topics. The main contributions of this paper are as follows.

- Proposes a simple yet effective RNN-based autoencoder that simultaneously learns both a valid feature space F and an embedding space suitable for clustering E. In addition, in contrast to simpler Multi-Layer Perceptron (MLP) models, employing an RNN-based autoencoder architecture enables learning superior embeddings for sequential data, due to its inherent strength in modeling sequential dependencies across extended contexts.
- To improve clustering performance, a secondary feedforward network is used to increase the distance between data points in the latent space of the autoencoder. This amplification of inter-cluster margins facilitates better separation. However, excessive distortion of the latent space may result in latent vectors that no longer represent accurately the original input characteristics. To address this, a two-phase training procedure is proposed. First, the autoencoder parameters are pre-trained to initialize a robust latent space. Subsequently, the encoder, decoder, and representation network modules are jointly optimized. This joint training procedure allows for enhancing cluster separation in the latent space while retaining the autoencoder's reconstruction fidelity.
- To enhance the clustering results, a technique was employed to filter the texts to ensure they contained a specific range of tokens, specifically between 10 and 30. This filtering approach was implemented to improve the effectiveness of the clustering process.

In recent years, the use of deep neural networks for clustering tasks, especially in image processing, has garnered substantial research attention due to their exceptional capacity in learning complex nonlinear relationships within data. Most deep clustering approaches focus on learning an effective high-level representational abstraction of the data to better equip the subsequent clustering process. A discussion of related works follows, focusing on recent approaches that use autoencoder-based architectures for text clustering. In summary, deep clustering techniques are well-positioned to leverage the representational learning strengths of deep neural networks, typically by focusing on objectives that ultimately improve clustering performance. For a comprehensive overview of deep clustering methods and other categories of these algorithms, the study in [16] provides an excellent survey of the latest techniques and developments. Additionally, a comprehensive survey on deep clustering was presented in [17], discussing taxonomic categorizations, current challenges, and future directions.

Several recent autoencoder-based deep clustering approaches are based on the seminal DEC model [5], which uses an autoencoder framework to learn embedded feature representations and subsequently performs clustering by minimizing KL divergence loss. Enhancements to DEC include IDEC [6], which highlights the importance of maintaining data structure integrity and includes the term of the lower-dimensional feature representation's reconstruction loss during fine-tuning tasks. In essence, IDEC aims to jointly optimize the weighted clustering loss and the autoencoder's reconstruction loss. DEC-DA [18] augments training data to improve embedding learning in DEC. In [19], Discriminatively Boosted image Clustering (DBC) was introduced to address the challenges of image representation learning and clustering. DBC shares a similar pipeline to DEC, but its learning process is self-paced [20], which means that it starts by selecting the easiest instances and gradually adds more complex samples. Density-based clustering has also been integrated with deep feature learning, as exemplified by DDIC [21]. DDIC leverages a denoising autoencoder to obtain feature representations and then clusters based on density without needing to pre-specify cluster numbers. In [15], a novel clustering approach used deep neural networks to simultaneously learn feature representations and embeddings suitable for clustering. This approach encourages the separation of natural clusters in the embedding space, thus enhancing the efficacy of the clustering process.

More recent DAE-based efforts focus on improving learned representations and clustering performance by combining deep embedding learning with traditional methods. DSCDAE [22] and SC-EDAE [23] specifically aimed to incorporate spectral

clustering objectives within optimized autoencoder architectures for clustering tasks. In [13], two novel methods, Structural Text Network Graph Autoencoder (STN-GAE) and Soft Cluster Assignment Autoencoder (SCA-AE), were proposed to improve short text clustering by advancing representation learning. STN-GAE utilizes a Graph Convolutional Network (GCN) within an autoencoder framework to capture structural relationships between texts, where texts are nodes in a K-nearest neighbor graph based on semantic similarity scores. Additionally, SCA-AE incorporates a soft cluster assignment approach that uses Student's t-distribution and KL-divergence loss to help embeddings align more closely with cluster centers and make them better suited for clustering. The study showed that BERT-based pre-trained representations performed significantly better than traditional embeddings such as Word2vec.

In the domain of Persian text clustering, an innovative method was proposed in [24] to cluster unlabeled Persian text data, addressing a significant need in Persian Natural Language Processing (NLP). This approach involves three main steps. First, it utilizes ParsBERT [25], a monolingual language model specifically designed for Persian and built on the BERT architecture. ParsBERT overcomes limitations in multilingual models, which often underperform with low-resource languages, such as Persian, by providing language-specific contextual representations that capture the unique semantic and syntactic nuances of Persian text. To develop ParsBERT, a large and diverse dataset was compiled from numerous Persian sources, including Persian Wikipedia, news websites, and other public text corpora. This comprehensive dataset enables ParsBERT to achieve high accuracy across several NLP tasks, including sentiment analysis, text classification, and Named Entity Recognition (NER). Such a dedicated model greatly enhances feature representation quality for Persian data, providing insights that general multilingual models lack. Using ParsBERT, Persian text data was transformed into high-quality numerical features optimized for clustering. Following this, a stacked autoencoder further refined these features by reducing their dimensionality, improving their suitability for clustering. Finally, the k-means algorithm organized these refined features into coherent clusters, allowing an effective clustering of semantically similar content. An innovative aspect of this work was the combined use of ParsBERT and the stacked autoencoder, which together support the extraction of rich and distinctive features crucial for accurate clustering. Furthermore, BERTopic modeling was applied to interpret each cluster more precisely by analyzing the underlying text context, providing valuable insights into the topic structure. The method demonstrated strong performance, achieving a high Silhouette score of 0.60, and consistently outperformed other clustering techniques within Persian NLP, highlighting its robustness, efficiency, and practical utility for Persian text clustering tasks.

Recently, a deep clustering framework built on the Neural Network Meaningful Learning (NNMeL) theory was introduced [26]. This model, known as DCSS, first employed a Contextualized Sentence Encoder (CSE) that used an attention mechanism along with LSTM units to transform raw text into a contextualized latent space. This was followed by a Sentence Similarity Classifier (SSC), which trained the network by taking pairs of sentences and learning their semantic similarity. The resulting contextualized representations were then passed through a Vector Separator Network (VSN) that mapped these vectors into a space where they were more separable for clustering. In contrast to traditional autoencoder-based methods, where reconstruction loss from denoising is used to learn latent representations, DCSS focused explicitly on training for sentence similarity before applying clustering.

Autoencoders are popular networks in unsupervised tasks such as text clustering, offering advantages by learning compact encoded representations that help reduce noise and dimensionality, making clustering more efficient. However, their effectiveness largely depends on architectural choices and the quality of the generated features. Although autoencoders hold great potential, they have limitations: Selecting the right architecture can be challenging, and poor choices can affect performance. Furthermore, if the encoded features are of low quality or irrelevant, clustering quality can be reduced.

This study extends the model in [15], which addresses these issues by combining an autoencoder with a representation network to create better-separated clusters. Unlike most autoencoder-based approaches that rely on basic architectures such as MLP, which are less effective for text data due to their inability to model sequential patterns, this approach is specifically tailored for text. Clustering is further optimized by applying clustering loss to a separate neural network, rather than directly to the encoder, creating a dedicated embedding space that is more effective for clustering Persian text data.

II. PROPOSED METHOD

A. Data Preparation

Social media posts often contain various types of noise, including typos, emojis, mentions, hashtags, and URLs. This study employed a customized tokenizer to parse Persian social media data. After tokenization, all nonvocabulary tokens, such as stop words, numeric, emojis, hashtags, and mentions, were removed [27]. This filtering process reduced noise while retaining the core textual content.

As noted previously, texts were filtered to include only those with 10 to 30 tokens. This range was selected because texts with fewer than 10 tokens lack sufficient information to discern their appropriate cluster. Consequently, the latent vectors generated for these ultra-short texts via the autoencoder are generally uninformative, reducing clustering performance. An upper bound of 30 tokens was imposed to improve short-text clustering. Longer texts intrinsically convey more information, potentially simplifying clustering. Thus, to focus only on short texts, they were excluded. To conclude the preprocessing phase, several methods, such as TF-IDF, SIF [10], Glove [28], and FastText [29], were employed to transform the textual data into vector representations. Subsequently, these embeddings were normalized using min-max normalization to minimize variance. This normalization accelerates training and speeds up convergence by scaling the data to a common range. These embeddings are provided as input to the model.

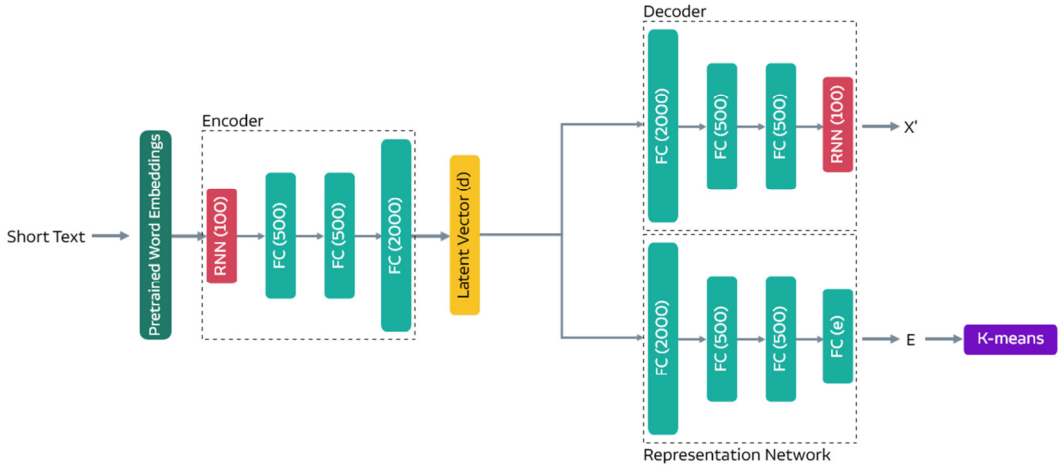

Fig. 1. Structure of the proposed method.

B. *Autoencoder*

The autoencoder is used to reduce the dimensionality of the features and compresses the high-dimensional input X into a low-dimensional latent representation Z while preserving most of the information content. Employing these compressed embeddings can reduce computational complexity and runtime. To minimize reconstruction error and achieve reconstructed outputs $\hat{X}$ closer to the original inputs X, Mean Squared Error (MSE) is commonly used as the loss function to quantify the deviation between $\hat{X}$ and X.

$$L\left(X,\hat{X}\right) = ||X - \hat{X}||^2 \quad (1)$$

Additionally, an autoencoder variant known as the Denoising Auto-Encoder (DAE) [30] accepts intentionally corrupted input data and is optimized to recover the original undistorted samples as outputs. This allows learning feature representations robust to partial corruption. Rather than directly reconstructing the input X, DAE is trained to reconstruct the clean input X from a corrupted version $\hat{X}$. This approach employed an RNN-based DAE (RNN-DAE) with cross-entropy reconstruction loss. The utilization of RNN layers is justified because of their ability to extract significant and informative features from textual data. For this RNN-DAE, LSTM, GRU, Bi-LSTM, and Bi-GRU layers were utilized.

C. *Representation Network*

Autoencoders, constrained by the smaller latent space Z, model data distributions concentrated near low-dimensional manifolds. They aim to learn representations implicitly capturing the local coordinate frame of these manifolds in the latent space Z. However, this yields compactly positioned representations with severe overlap between natural clusters. Therefore, clustering in the latent space Z is ineffective [15]. To allow for more effective clustering, the encoder must be adapted to sufficiently separate natural clusters in the representation space.

A fully-connected neural network, termed the representation network, is employed to overcome this limitation [15], connected to the encoder portion of the framework. This network is optimized to learn the mapping from the latent space Z to the embedded space E by learning weights φ that preserve local structure while promoting greater separation between natural clusters in E. This generates embeddings of well-separated clusters, facilitating more effective clustering compared to the original representation space.

To capture local data structure, pairwise distances in the latent and embedded spaces are converted into Student's t-distribution-based similarity probabilities p_{ij} and q_{ij}, respectively, as defined in (2) and (3). Here, p_{ij} defines the probability of latent representations z_i and z_j, with α set to $2d$, where d is the latent dimension. In contrast, q_{ij} uses $\alpha = 1$ to induce greater cluster separation in the embedded space E [15].

$$p_{ij} = \frac{(1+|| f(x_i) - f(x_j)||^2 / \alpha)^{-\frac{\alpha+1}{2}}}{\sum_{k \neq l}(1+|| f(x_k) - f(x_l) ||^2 / \alpha)^{-\frac{\alpha+1}{2}}} \quad (2)$$

$$q_{ij} = \frac{(1+|| h(z_i) - h(z_j)||^2)^{-1}}{\sum_{k \neq l}(1+|| h(z_k) - h(z_l) ||^2)^{-1}} \quad (3)$$

Finally, the representation network's weights φ are learned by minimizing the cross-entropy between distributions p and q as formalized in (4). Minimizing this cross-entropy loss is equivalent to minimizing the entropy of distribution p as well as the KL divergence between p and q and results in pushing clusters apart and reducing overlap, allowing more accurate and efficient clustering.

$$L_E = -\sum_i \sum_j p_{ij} \log(q_{ij}) = H(p) + KL(p||q) \quad (4)$$

D. *Training*

The optimization method consists of two phases:

- Initialization and Latent Space Creation: The goal of the initialization phase is to pretrain the autoencoder to learn how to effectively compress the input data into a lower-dimensional latent vector space. The autoencoder is trained in an unsupervised manner solely to minimize the reconstruction loss. This trains the autoencoder how to encode the most salient features of the data in the latent bottleneck layer. An RNN-based autoencoder is used, which is better suited for sequence data as it can learn

temporal dynamics. The key is training the autoencoder on in-domain data to learn a latent space tailored to the characteristics of the specific dataset. Proper initialization is crucial to ensure that the autoencoder provides informative latent representations before the representation network is added.

- Joint Training: In this phase, the representation network is added to the trained autoencoder, and all modules are optimized together. The representation network is trained to further refine the latent space to make it more amenable for clustering. It minimizes a cross-entropy loss, which encourages clearer separation between data points in the latent and embedded spaces. However, the autoencoder's reconstruction accuracy must be maintained during this phase. If the latent space is distorted too much, the decoded outputs will become less accurate. Therefore, the clustering loss is combined with the autoencoder reconstruction loss when training the encoder. This forces the encoder to balance the clustering-friendliness and fidelity of the latent space. The decoder and representation networks are trained simultaneously to optimize this joint latent space. Training all modules together allows for improving clustering performance while retaining the autoencoder's reconstruction capability.

Regarding loss functions, the cross-entropy (4) is utilized for the representation network. The decoder module employs a reconstruction loss, while the encoder uses both the reconstruction loss and the representation loss. The representation loss is weighted by a coefficient γ. This combined encoder loss lets us control how much the latent space Z gets distorted. The objective function is defined in (5).

$$L = L_R + \gamma L_E \quad (5)$$

III. EXPERIMENTS

The model was implemented in the Python programming language utilizing the TensorFlow framework [31].

A. Dataset

This study used the Sep_TD_Tel01 dataset [32], which is in Persian. This dataset was collected from Telegram without any limitations, such as keywords, and therefore, it fully represents a data stream nature. Several studies have used this dataset in different fields such as NER, event detection, text clustering, and topic detection [26, 33-35]. These studies show the dataset's versatility and significance in Persian language processing research. Most studies on text clustering focused on the Twitter social network. Twitter's popularity stems from its provision of free API access to data for users [36, 37]. In contrast, this study focuses on the Telegram social network because it is very popular in Iran. A message collector system was developed by the ComInSys Laboratory to access Telegram data. This dataset includes messages collected from public channels and groups, with approximately assigned topic labels to 23% of them. The dataset has 75 distinct labeled clusters, which were reduced to 44 clusters by constraining the texts to have between 10 and 30 tokens. Table I presents details of the dataset. Figures 2 and 3 illustrate the number of posts and their clusters after preprocessing and normalization.

TABLE I. DATASET SUMMARY [32]

Number of posts	10,209
Number of super-topics	2
Number of sub-topics	81
Number of labeled posts	2,365

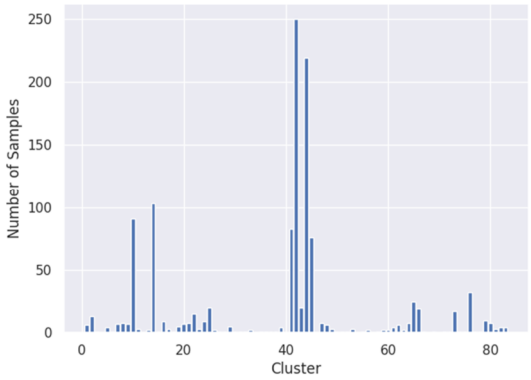

Fig. 2. Posts count and their cluster before preprocessing.

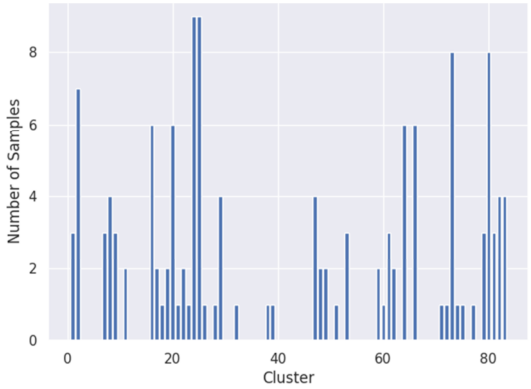

Fig. 3. Posts count and their cluster after preprocessing.

B. Experimental Setup

1) Network Architecture and Parameter Settings

Similarly to DEC, the encoder and decoder networks utilize the following architectures, respectively: D-RNN(100)-FC(500)-FC(500)-FC(2000)-FC(d) and FC(d(-FC(2000)-FC(500)-FC(500)-RNN(100)-FC(D), where "D" denotes the dimensionality of the input data and "d" indicates the dimension of the latent representation space. The representation network architecture used is the same as [15] with d-FC(2000)-FC(500)-FC(500)-FC(e), where "e" represents the embedding space dimension. In this architecture, RNN stands for using an RNN layer, and "FC" denotes fully connected neural networks. For all these layers, the Rectified Linear Unit (ReLU) [38] was used as the activation function. Stochastic Gradient Descent (SGD) with momentum [39] and Adam [40] optimizer were used for training, with a batch size of 32 samples. The learning rate was set to 0.001, while the latent dimension "d" was set to 100. Each experimental configuration was executed 10 times, and the average score across these 10 repetitions was recorded.

2) Evaluation Metrics

Unsupervised clustering accuracy (Acc) and Normalized Mutual Information (NMI) were used for evaluation.

- Unsupervised Clustering Accuracy (Acc): Given an individual text example x_i, let c_i denote the obtained cluster label and t_i indicate the label provided in the corpus. Accuracy is formally defined as:

$$Acc = \frac{\sum_{i=1}^{n} \delta(t_i, map(c_i))}{n} \quad (6)$$

where n represents the total number of texts, $\delta(x, y)$ is an indicator function being 1 if $x = y$ and 0 otherwise, and $map(c_i)$ signifies a permutation function that maps each cluster label c_i to the equivalent corpus label using the Hungarian algorithm [41].

- Normalized Mutual Information (NMI): For label set T and cluster set C, NMI is defined as:

$$NMI(T, C) = \frac{MI(T, C)}{\sqrt{H(T)H(C)}} \quad (7)$$

where $MI(T, C)$ represents mutual information between T and C, and $H(T)$ and $H(C)$ denote the entropies of T and C, respectively. NMI assesses clustering accuracy by measuring the correlation between the predicted clusters and ground truth labels.

C. Results

The scarcity of previous work on the Persian dataset challenges the comparison of the results with other works. Based on baseline works [5, 15], the aim is to extend them and address the unique challenges of Persian. Experiments were carried out both without and following the constraint of text lengths to 10-30 tokens. Initially, the dataset contained 75 clusters, but after applying the mentioned constraints, the clusters were reduced to 44. For both settings, various high-quality embedding types, including TF-IDF, SIF, GloVe, FastText, and ParsBert, were evaluated. Table II presents the results before and after the token constraint. In addition, this method is compared to some baseline and recent deep clustering models, mainly autoencoder-based architectures that have been shown to yield strong performance on textual data. As shown in Table II, adding the token number constraints greatly improved evaluation metrics. This is because texts shorter than 10 tokens cannot capture the key information or cluster structure well. The results in Table II clearly demonstrate that the proposed method outperforms prior works in terms of Acc and NMI across various embedding types, particularly after applying constraints. Table II provides a comprehensive comparison of clustering performance using different embedding types before and after applying token constraints, offering a clear numerical representation of the improvements achieved. Table III presents the results of the proposed approach using different types of RNN layers to compare their performance using TF-IDF embeddings. Since TF-IDF embeddings demonstrated superior performance over other embedding techniques, Table IV provides a detailed comparison of the proposed approach with previous works using TF-IDF.

As seen in Table III, the proposed approach achieved better accuracy and NMI across all RNN layer variants. This performance boost can be attributed to the ability of RNNs to effectively model sequential information, especially within short texts. Among these models, the best result was achieved using Bi-LSTM as the RNN layer of the AE module. Table IV presents a performance comparison between different deep clustering techniques and the proposed method using optimal configurations. Both Acc and NMI metrics are reported on the basis of the same dataset. The proposed approach demonstrates a significant improvement in clustering performance, achieving an Acc of 0.63285 and an NMI of 0.82321, substantially outperforming previous methods that achieved at most 0.61314 Acc and 0.78276 NMI. This validates that the proposed approach can effectively learn semantic features that capture the inherent structures and clusters within short texts. The recurrent autoencoder architecture and two-phase training procedure enable robust representation learning from these limited contexts.

TABLE II. COMPARISON OF ACC AND NMI OF CLUSTERING METHODS BASED ON THREE TYPES OF EMBEDDING BEFORE AND AFTER TOKEN CONSTRAINTS

Method	**Network**	**Embedding**	**Acc (%)**		**NMI (%)**	
			Before constraints	**After constraints**	**Before constraints**	**After constraints**
K-means	-	TF-IDF	22.071	54.517	16.651	76.650
DEC [5]	FCN	TF-IDF	22.467	60.162	16.879	77.949
IDEC [6]	FCN		24.987	61.013	18.025	78.537
AE+RN+K-means [15]	FCN		28.889	61.314	17.134	78.276
This approach	RNN		**32.517**	**63.285**	22.076	**82.321**
DEC	FCN	Glove	15.283	52.981	31.677	75.793
IDEC	FCN		17.879	55.087	34.320	76.112
AE+RN+K-means	FCN		16.425	53.285	31.981	77.097
This approach	RNN		17.361	60.814	29.521	76.982
DEC	FCN	FastText	17.214	46.257	32.011	69.514
IDEC	FCN		19.026	49.351	33.041	71.671
AE+RN+K-means	FCN		18.902	48.905	31.985	70.013
This approach	RNN		17.982	60.549	31.957	76.142
DEC	FCN	SIF	19.172	54.257	32.995	76.514
IDEC	FCN		19.783	56.863	**33.581**	78.291
AE+RN+K-means	FCN		20.034	57.001	19.527	76.012
This approach	RNN		20.819	61.475	21.547	77.025
Stacked AE+K-means [24]	FCN	ParsBERT	30.328	61.995	23.014	76.361
This approach	RNN		31.851	62.789	26.490	78.987

TABLE III. COMPARISON OF ACC AND NMI OF THE PROPOSED METHOD BASED ON DIFFERENT TYPES OF RNN LAYERS

Method	Type of RNN layer used in AE	Optimizer	Acc	NMI
This approach	Dense	Adam	0.58175	0.79744
		SGD-mom	0.58540	0.80221
	LSTM	Adam	0.53284	0.77755
		SGD-mom	0.63139	0.81736
	Bi-LSTM	Adam	0.52920	0.77245
		SGD-mom	**0.63285**	**0.82321**
	GRU	Adam	0.51241	0.76281
		SGD-mom	0.61898	0.80980
	Bi-GRU	Adam	0.50218	0.76260
		SGD-mom	0.61605	0.81652

TABLE IV. ACC AND NMI COMPARISON OF DIFFERENT METHODS AND THE PROPOSED APPROACH WITH OPTIMAL SETTINGS.

Method	AE Network	ACC (%)	NMI (%)
K-means	-	54.517	76.650
DEC	FCN	60.162	77.949
IDEC	FCN	61.013	78.537
AE+RN+K-means	FCN	61.314	78.276
Stacked AE+K-means	FCN	61.995	76.361
This approach	**RNN**	**0.63285**	**0.82321**

Analyzing the component-level results provides further insights. The configuration utilizing a Bi-LSTM recurrent layer coupled with the SGD-Mom optimizer obtained the strongest performance among the variants examined. This aligns with the motivation to use RNN architectures that are suitable for the sequential modeling of short texts. The Bi-LSTM-based model surpassed even the best method's performance by a significant margin, demonstrating the benefits of the proposed approach's innovations. The training process, as discussed above, involved two phases: Phase 1 to initialize the autoencoder and Phase 2 to jointly train with the representation network and clustering. Five different settings for the proposed method were implemented and evaluated.

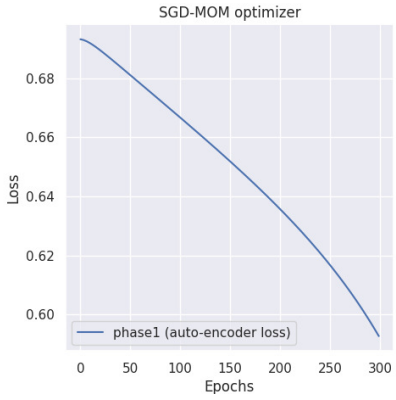

Fig. 4. Loss values of the Bi-LSTM model for Phase 1.

Figures 4 and 5 illustrate the loss values for the respective phases, while Figure 6 compares the loss trends across both phases for the best-performing setting in terms of results. The consistent decrease in loss values demonstrates the model's convergence and ability to learn discriminative representations for effective clustering, as evidenced by the successful optimization of the objective function.

Fig. 5. Loss values of the Bi-LSTM model for Phase 2.

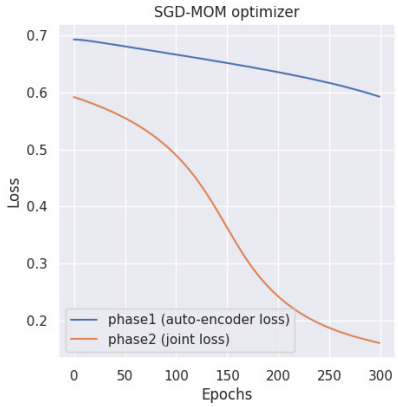

Fig. 6. Comparison of loss values for Phase 1 and Phase 2.

IV. DISCUSSION

The results demonstrate the efficacy of the proposed method in significantly improving Acc and NMI scores on the Persian short text dataset. A critical aspect of this study was learning an effective mapping from the initial text embedding space to a transformed one, where the clusters would be more separable and amenable to efficient clustering. To achieve this, an RNN-based autoencoder with a representation network was used, enabling the modeling and extraction of sequential data from the text. This approach was advantageous in capturing useful information while minimizing cluster overlap.

Although the proposed architecture outperformed other deep clustering architectures, the imbalanced nature of the dataset, as evident from Figures 2 and 3, may have influenced the results. The challenge of class imbalance was not the primary focus of this study; however, it is acknowledged that addressing this issue could potentially enhance clustering performance. Developing strategies to mitigate the impact of class imbalance on clustering algorithms could be a valuable direction for future research.

Furthermore, the performance of the proposed method on the Persian short text dataset underscores its potential applicability to languages with rich morphological structures and complex word formations. Similar challenges in extracting discriminative features for effective clustering have been noted in recent surveys on text-matching techniques [42], and experimental studies such as [43] further illustrate the potential of deep-based models to handle complex linguistic features. This study contributes to the literature by demonstrating the effectiveness of deep learning techniques in clustering tasks for morphologically rich languages, which have traditionally posed challenges for traditional deep clustering methods. In addition, the scarcity of works on the Persian language introduces a significant challenge in terms of reproducibility and comparative analysis, limiting direct comparative evaluations. Furthermore, the distinct nature of the dataset used, while having many challenges like other low-resource languages, introduces additional complexities when attempting to benchmark against methods on other languages. Due to these constraints and the lack of existing open-source implementations of previous works in this field, especially recent ones, we were unable to perform a direct comparison with more recent approaches and instead implemented baseline methods for comparison. Most studies in this field extend baselines and add some novelty to improve them. This approach allowed us to identify the weaknesses of previous works on low-resource languages such as Persian and then address them to improve language modeling, dataset quality, and clustering effectiveness. Thus, this method served a dual purpose: to improve existing works on short text clustering and contribute to the Persian NLP field. Adopting and extending existing works on the Persian language can bridge the gap between general short-text clustering approaches that focus mainly on English.

To our knowledge, the study in [26] represents one of the first attempts to apply deep clustering techniques to Persian short texts using the Sep_TD_Tel01 dataset. Similarly, the proposed method is among the first to focus on short-text clustering within this dataset. Although both approaches leverage a sentence similarity-based training strategy, several fundamental differences set them apart. The DCSS model was trained on the entire Sep_TD_Tel01 dataset without any token-length constraints, whereas this approach specifically filters texts to include only those containing 10-30 tokens. The rationale behind this decision is that clustering extremely short texts is particularly challenging due to the lack of context, and filtering helps ensure a clear short-text clustering setting. Although this step removes some longer, more informative samples, it forces the model to operate under more constrained conditions, making the competitive performance of the proposed method particularly noteworthy. Moreover, the proposed architecture differs significantly from DCSS. While DCSS employs a contextualized sentence encoder with attention and LSTM layers, this method integrates an RNN-based autoencoder combined with an additional representation network to refine the learned embeddings. This dual-stage architecture improves the representation learning process and enhances clustering performance on the filtered dataset. Although a direct numerical comparison is not feasible due to dataset differences, the results indicate that the proposed approach performs effectively under stricter conditions, suggesting that the proposed enhancements in representation learning contribute to improved clustering quality.

These distinctions underscore the tailored nature of the proposed framework for clustering Persian short texts in low-resource NLP settings. Despite differences in dataset preprocessing and experimental setups, this approach offers a strong alternative, highlighting the impact of architectural choices and dataset filtering on clustering performance. Future research could further explore these aspects by directly comparing different preprocessing strategies and extending the proposed method to more diverse datasets.

In addition, while the results are promising, it is essential to acknowledge some limitations. One potential limitation is the size and diversity of the dataset used for the evaluation. Future research could explore the performance of the proposed method on larger and more diverse datasets that encompass a wider range of topics and domains. Additionally, investigating the proposed method's robustness across different languages and text genres would further validate its generalizability and practical utility.

In conclusion, this study presents a novel and effective approach for short text clustering, particularly in languages with rich morphological structures. By addressing the challenge of class imbalance and exploring larger and more diverse datasets, future research can build on these findings and further advance the field of text clustering using deep learning techniques.

V. CONCLUSIONS

This study introduced a new deep clustering method customized for short texts. The main idea was to use an RNN autoencoder with an extra representation network to jointly learn features optimized for clustering. Several key contributions were made. An RNN autoencoder, which is better than simple fully-connected autoencoders for short and sequential data, was proposed to create a latent space that captures the rich structure of text. A representation network, trained to spread clusters more in the latent space, was also added. A two-step training approach was used to improve cluster separation while maintaining the autoencoder's reconstruction accuracy. Finally, the texts were preprocessed to be 10-30 tokens long, which is the ideal range for clustering short texts. With these innovations, this model can learn high-quality cluster-aware embeddings tailored to short texts.

The experiments showed significant improvements in clustering accuracy compared to previous methods. The proposed RNN autoencoder and two-step training allow for more effective embedded feature learning from these limited contexts. Limiting text lengths also reflects the focus solely on short texts. Specifically, this method surpassed prior clustering models in Acc and NMI, particularly when incorporating Bi-LSTM layers. Compared to baseline methods and recent deep clustering approaches, such as autoencoder-based models, this framework consistently outperformed them across multiple embedding techniques. The performance gain validates the contributions, particularly in adapting RNN architectures and a

structured training approach for clustering short texts effectively.

In summary, this work introduced a deep clustering framework that demonstrated substantial benefits for the increasingly vital short-text domain. The proposed improvements provide significant accuracy gains and more robust cluster formation, providing an impactful advance in representation learning and training for limited contexts. Future work can build on these cluster-oriented embeddings and recurrent two-phase architectures to further advance short-text clustering.

DECLARATIONS

Funding: This research received no specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Conflicts of Interest: The authors declare that they have no conflict of interest.

REFERENCES

[1] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*, 1967, vol. 5, pp. 281–298.

[2] D. Birant and A. Kut, "ST-DBSCAN: An algorithm for clustering spatial–temporal data," *Data & Knowledge Engineering*, vol. 60, no. 1, pp. 208–221, Jan. 2007, https://doi.org/10.1016/j.datak.2006.01.013.

[3] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.

[4] M. Steinbach, L. Ertöz, and V. Kumar, "The Challenges of Clustering High Dimensional Data," in *New Directions in Statistical Physics: Econophysics, Bioinformatics, and Pattern Recognition*, L. T. Wille, Ed. Springer, 2004, pp. 273–309.

[5] J. Xie, R. Girshick, and A. Farhadi, "Unsupervised Deep Embedding for Clustering Analysis," in *Proceedings of The 33rd International Conference on Machine Learning*, Jun. 2016, pp. 478–487.

[6] X. Guo, L. Gao, X. Liu, and J. Yin, "Improved deep embedded clustering with local structure preservation," in *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, Melbourne, Australia, May 2017, pp. 1753–1759.

[7] B. Yang, X. Fu, N. D. Sidiropoulos, and M. Hong, "Towards K-means-friendly Spaces: Simultaneous Deep Learning and Clustering," in *Proceedings of the 34th International Conference on Machine Learning*, Jul. 2017, pp. 3861–3870.

[8] J. Yang, D. Parikh, and D. Batra, "Joint Unsupervised Learning of Deep Representations and Image Clusters," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 5147–5156, https://doi.org/10.1109/CVPR.2016.556.

[9] J. Chang, L. Wang, G. Meng, S. Xiang, and C. Pan, "Deep Adaptive Image Clustering," in *2017 IEEE International Conference on Computer Vision (ICCV)*, Venice, Italy, Oct. 2017, pp. 5880–5888, https://doi.org/10.1109/ICCV.2017.626.

[10] A. Hadifar, L. Sterckx, T. Demeester, and C. Develder, "A Self-Training Approach for Short Text Clustering," in *Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019)*, Florence, Italy, 2019, pp. 194–199, https://doi.org/10.18653/v1/W19-4322.

[11] K. Zhang, Z. Lian, J. Li, H. Li, and X. Hu, "Short Text Clustering with a Deep Multi-embedded Self-supervised Model," in *Artificial Neural Networks and Machine Learning – ICANN 2021*, 2021, pp. 150–161, https://doi.org/10.1007/978-3-030-86383-8_12.

[12] M. Hao, W. Wang, and F. Zhou, "Joint Representations of Texts and Labels with Compositional Loss for Short Text Classification," *Journal of Web Engineering*, vol. 20, no. 3, pp. 669–688, Feb. 2021, https://doi.org/10.13052/jwe1540-9589.2035.

[13] H. Yin, X. Song, S. Yang, G. Huang, and J. Li, "Representation Learning for Short Text Clustering," in *Web Information Systems Engineering – WISE 2021*, 2021, pp. 321–335, https://doi.org/10.1007/978-3-030-91560-5_23.

[14] C. Wei, L. Zhu, and J. Shi, "Short Text Embedding Autoencoders With Attention-Based Neighborhood Preservation," *IEEE Access*, vol. 8, pp. 223156–223171, 2020, https://doi.org/10.1109/ACCESS.2020.3042778.

[15] P. Dahal, "Learning Embedding Space for Clustering From Deep Representations," in *2018 IEEE International Conference on Big Data (Big Data)*, Seattle, WA, USA, Dec. 2018, pp. 3747–3755, https://doi.org/10.1109/BigData.2018.8622629.

[16] Y. Ren *et al.*, "Deep Clustering: A Comprehensive Survey," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 36, no. 4, pp. 5858–5878, Apr. 2025, https://doi.org/10.1109/TNNLS.2024.3403155.

[17] S. Zhou *et al.*, "A Comprehensive Survey on Deep Clustering: Taxonomy, Challenges, and Future Directions," *ACM Computing Surveys*, vol. 57, no. 3, Aug. 2024, Art. no. 69, https://doi.org/10.1145/3689036.

[18] D. Chen, J. Lv, and Z. Yi, "Unsupervised Multi-Manifold Clustering by Learning Deep Representation," presented at the The AAAI-17 Workshop on Crowdsourcing, Deep Learning, and Artificial Intelligence Agents, 2017.

[19] F. Li, H. Qiao, and B. Zhang, "Discriminatively boosted image clustering with fully convolutional auto-encoders," *Pattern Recognition*, vol. 83, pp. 161–173, Nov. 2018, https://doi.org/10.1016/j.patcog.2018.05.019.

[20] M. Kumar, B. Packer, and D. Koller, "Self-Paced Learning for Latent Variable Models," in *Advances in Neural Information Processing Systems*, 2010, vol. 23.

[21] Y. Ren, N. Wang, M. Li, and Z. Xu, "Deep density-based image clustering," *Knowledge-Based Systems*, vol. 197, Jun. 2020, Art. no. 105841, https://doi.org/10.1016/j.knosys.2020.105841.

[22] X. Yang, C. Deng, F. Zheng, J. Yan, and W. Liu, "Deep Spectral Clustering Using Dual Autoencoder Network," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA, Jun. 2019, pp. 4061–4070, https://doi.org/10.1109/CVPR.2019.00419.

[23] S. Affeldt, L. Labiod, and M. Nadif, "Spectral clustering via ensemble deep autoencoder learning (SC-EDAE)," *Pattern Recognition*, vol. 108, Dec. 2020, Art. no. 107522, https://doi.org/10.1016/j.patcog.2020.107522.

[24] S. Hosseini and Z. A. Varzaneh, "Deep text clustering using stacked AutoEncoder," *Multimedia Tools and Applications*, vol. 81, no. 8, pp. 10861–10881, Mar. 2022, https://doi.org/10.1007/s11042-022-12155-0.

[25] M. Farahani, M. Gharachorloo, M. Farahani, and M. Manthouri, "ParsBERT: Transformer-based Model for Persian Language Understanding," *Neural Processing Letters*, vol. 53, no. 6, pp. 3831–3847, Dec. 2021, https://doi.org/10.1007/s11063-021-10528-4.

[26] E. Zafarani-Moattar, M. R. Kangavari, and A. M. Rahmani, "Neural Network Meaningful Learning Theory and its Application for Deep Text Clustering," *IEEE Access*, vol. 12, pp. 42411–42422, 2024, https://doi.org/10.1109/ACCESS.2024.3375754.

[27] M. Molaei and D. Mohamadpur, "Distributed Online Pre-Processing Framework for Big Data Sentiment Analytics," *Journal of AI and Data Mining*, vol. 10, no. 2, pp. 197–205, Apr. 2022, https://doi.org/10.22044/jadm.2022.11330.2293.

[28] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532–1543.

[29] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching Word Vectors with Subword Information," *Transactions of the Association for Computational Linguistics*, vol. 5, pp. 135–146, Jun. 2017, https://doi.org/10.1162/tacl_a_00051.

[30] P. Vincent, H. Larochelle, Y. Bengio, and P. A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proceedings of the 25th international conference on Machine learning*, Apr. 2008, pp. 1096–1103, https://doi.org/10.1145/1390156.1390294.

[31] M. Abadi *et al.*, "TensorFlow: A System for Large-Scale Machine Learning," presented at the 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16), 2016, pp. 265–283. [Online]. Available: https://www.usenix.org/conference/osdi16/technical-sessions/presentation/abadi.

[32] M. Ranjbar-Khadivi, M.-R. Feizi-Derakhshi, A. Forouzandeh, P. Gholami, A.-R. Feizi-Derakhshi, and E. Zafarani-Moattar, "Sep_TD_Tel01." Mendeley, Jan. 6, 2022, https://doi.org/10.17632/372rnwf9pc.1.

[33] P. Gholami-Dastgerdi, M. R. Feizi-Derakhshi, and P. Salehpour, "SSKG: Subject stream knowledge graph, a new approach for event detection from text," *Ain Shams Engineering Journal*, vol. 15, no. 12, Dec. 2024, Art. no. 103040, https://doi.org/10.1016/j.asej.2024.103040.

[34] M. Ranjbar-Khadivi, S. Akbarpour, M. R. Feizi-Derakhshi, and B. Anari, "A Human Word Association Based Model for Topic Detection in Social Networks," *Annals of Data Science*, Jul. 2024, https://doi.org/10.1007/s40745-024-00561-0.

[35] A. Forouzandeh, M. R. Feizi-Derakhshi, and P. Gholami-Dastgerdi, "Persian Named Entity Recognition by Gray Wolf Optimization Algorithm," *Scientific Programming*, vol. 2022, no. 1, 2022, Art. no. 6368709, https://doi.org/10.1155/2022/6368709.

[36] X. Liu *et al.*, "Emotion classification for short texts: an improved multi-label method," *Humanities and Social Sciences Communications*, vol. 10, no. 1, pp. 1–9, Jun. 2023, https://doi.org/10.1057/s41599-023-01816-6.

[37] X. Liu *et al.*, "Developing Multi-Labelled Corpus of Twitter Short Texts: A Semi-Automatic Method," *Systems*, vol. 11, no. 8, Aug. 2023, Art. no. 390, https://doi.org/10.3390/systems11080390.

[38] X. Glorot, A. Bordes, and Y. Bengio, "Deep Sparse Rectifier Neural Networks," in *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, Jun. 2011, pp. 315–323.

[39] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, "On the importance of initialization and momentum in deep learning," in *Proceedings of the 30th International Conference on Machine Learning*, May 2013, pp. 1139–1147.

[40] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization." arXiv, Jan. 30, 2017, https://doi.org/10.48550/arXiv.1412.6980.

[41] H. W. Kuhn, "The Hungarian method for the assignment problem," *Naval Research Logistics Quarterly*, vol. 2, no. 1–2, pp. 83–97, 1955, https://doi.org/10.1002/nav.3800020109.

[42] A. Alqahtani, H. Alhakami, T. Alsubait, and A. Baz, "A Survey of Text Matching Techniques," *Engineering, Technology & Applied Science Research*, vol. 11, no. 1, pp. 6656–6661, Feb. 2021, https://doi.org/10.48084/etasr.3968.

[43] S. Rezaei *et al.*, "An experimental study of sentiment classification using deep-based models with various word embedding techniques," *Journal of Experimental & Theoretical Artificial Intelligence*, https://doi.org/10.1080/0952813X.2024.2384568.