This may be the author’s version of a work that was submitted/accepted for publication in the following source:

**Mohotti, Wathsala & Nayak, Richi**
(2020)
Deep hierarchical non-negative matrix factorization for clustering short text.
In Yang, Haiqin, Pasupa, Kitsuchart, Leung, Andrew Chi-Sing, Kwok, James T., Chan, Jonathan H., & King, Irwin (Eds.) *Neural Information Processing: 27th International Conference, ICONIP 2020, Bangkok, Thailand, November 23–27, 2020, Proceedings, Part II.*
Springer, Cham, pp. 270-282.

This file was downloaded from: https://eprints.qut.edu.au/206906/

© Springer Nature Switzerland AG 2020

This work is covered by copyright. Unless the document is being made available under a Creative Commons Licence, you must assume that re-use is limited to personal use and that permission from the copyright owner must be obtained for all other uses. If the document is available under a Creative Commons License (or other specified license) then refer to the Licence for details of permitted re-use. It is a condition of access that users recognise and abide by the legal requirements associated with these rights. If you believe that this work infringes copyright please provide details by email to qut.copyright@qut.edu.au

**License:** Creative Commons: Attribution-Noncommercial 4.0

**Notice:** Please note that this document may not be the Version of Record (i.e. published version) of the work. Author manuscript versions (as Submitted for peer review or as Accepted for publication after peer review) can be identified by an absence of publisher branding and/or typeset appearance. If there is any doubt, please refer to the published source.

https://doi.org/10.1007/978-3-030-63833-7_23
Deep Hierarchical Non-negative Matrix Factorization for Clustering Short Text

Wathsala Anupama Mohotti and Richi Nayak
School of Computer Science, Queensland University of Technology, Brisbane, Australia, \{wathsalaanupama.mohotti@hdr., r.nayak@\}qut.edu.au

Abstract. This paper proposes a deep hierarchical Non-negative Matrix Factorization (NMF) method with Skip-Gram with Negative sampling (SGNS) to learn semantic relationships in short text data. The proposed unsupervised method learns a dense lower-order text presentation by minimizing the encoding and decoding error of factor matrices. Semantically-enriched dense text representation is constructed using the factor matrices where clusters are identified. We empirically evaluate the effectiveness of the method against the state-of-the-art short text clustering methods and deep neural embedding based methods.

Keywords: Deep learning · NMF · SGNS

1 Introduction

Social media platforms are a popular networking mechanism that allow users to disseminate information and assemble social views based on short-text communication [6]. A short text data faces sparsity and low word co-occurrences that create challenges for unsupervised text mining to identify groups or topics or concepts within the data. Recently, supervised deep learning methods based on shallow auto-encoder to deep auto-encoders using Recurrent Neural Networks (RNN) [10] and Convolutional Neural Network (CNN) [10] have been used in learning deep feature representation [24]. However, discovering a dense representation for short text in a fully unsupervised manner is essential in many applications to identify the clusters or concepts or topics.

Non-negative Matrix factorization (NMF) [12], which maps the high-dimensional text representation to a lower-dimensional representation, has become popular in text clustering due to its capability to learn part-based lower-order representation where groups can be identified accurately [1, 14]. Though the decomposed factor matrices are considerably dense in traditional text data and can be used to identify clusters, extreme-spareness in short text challenges them in identifying dense factor matrices for the short text data.

In this paper, we present a novel method of deep Hierarchical NMF in which input data undergoes a special normalization which results in an effect similar to word embedding. This allows NMF to identify dense factor matrices incorporating contexts in the short text data. This technique is similar to Skip-Gram
modelling with negative sampling (SGNS) when used with NMF. To the best of our knowledge, the proposed method, named as SG-DHNMF, is the first method that aims to (1) capture the semantic relationship in the short text data by analysing the pairwise documents similarity aligning with SGNS modeling and (2) progressively identify lower-rank factor matrices with each layer of hierarchical NMF that encodes the information of sparse document representation in each iteration. We conjecture that hierarchical factorization of document × document matrix into lower order can embed the geometrical structure of sparse data by combining the nearest neighbor (NN) information in each projection step. Empirical analysis with several Twitter datasets reveals that SG-DHNMF can handle sparsity in short text and outperforms the state-of-the-art short text clustering methods in finding accurate clusters.

2 Related Work

Document Expansion These methods typically expand the feature vectors by adding relevant terms to deal with the sparsity in data [2, 7, 9]. A common approach is to use external knowledge sources for document expansion such as Wikipedia [2], WordNet [7], and ontologies [9]. Currently, word embedding-based pre-trained models such as word2vec [15], doc2vec [11], Glove [19] and Skip-Gram [17] have been used by exploiting semantic relationships in the data. However, short text in social media enriched using these static external sources provide inadequate information due to semantic incoherence [18]. Social media data includes unstructured phrases that result in a huge variance to traditional text vocabulary. Self corpus-based expansion is proposed as an alternative semantically aligned method in which concepts are identified in the collection for augmentation using clustering [8] or topics based on term frequency probabilities [18]. However, all these methods face challenges in dealing with the fewer word co-occurrences and the unstructured nature of the micro-blogging data [11].

Supervised deep feature learning Deep neural networks have been successfully used in feature learning. Deep auto-encoder [4] was one of the first models used in the text representation. Recent research uses advanced versions of NNs to reconstruct text representation [10]. Recurrent and recursive NNs [16, 22] have been used with word-embedding to improve the representation learning process by including information of previous nodes for better semantic analysis in the dataset. Recent research applies convolution filters to capture the features similarity and achieve dense feature representation with CNNs [24, 10]. CNN-based methods produce promising results in linguistic tasks among all supervised methods due to their ability to detect patterns in the data [3]. However, these methods rely on the label data (i.e. the ground-truths) for feature learning and cannot be applied to tasks where finding labelled data is scarce.

Unsupervised deep feature learning This emerging research area includes two families of methods, word-embedding-based and matrix factorization. The word embedding methods include language modeling and feature learning techniques where words are mapped to vectors of real numbers using a vocabulary
Word2vec [15], a feed-forward NN based method that efficiently estimates word representations in vector space, is a popular model. Doc2Vec [11] is an extension of Word2vec producing document level embedding using a word vector generated for each word and a document vector generated for each document. Glove [19] is a non-neural network based vector space representation model. It considers a global word×word co-occurrence count matrix and uses the statistics in representing documents. The applicability of these pre-trained word-embedding models to short text data is limited due to vocabulary mismatch that shows a huge variance to the general text, a fewer number of word co-occurrence in short text data and, the noisy nature of micro-blogging data.

Matrix factorization methods are the leading unsupervised text representation methods. NMF has been used in clustering multi-view data by learning latent features embedded in multiple views [14]. It uses assistance provided by many views in identifying the final set of features. In comparison to the one-step dimensional reduction in traditional NMF, the use of progressive dimensional reduction with multiple iterations [5, 25] is a recent approach. This is used in [5] with deep learning similar to autoencoder network considering encoding error, trained by a non-negativity constraint algorithm to learn features that show a part-based representation of data with matrix factorization. A handful of methods has been existed that use this type of hierarchical feature learning with NMF in step-by-step fashion [23] for document data to discover feature hierarchies in concepts. However, this stacking of NMF in leaning feature hierarchies considers geometric relationships between features within each iteration and encodes data to factor matrices that could approximate the input matrix more accurately. This method captures latent features precisely that could be neglected by one step dimensional reduction process. In [25], encoding as well as decoding of a factor matrix is considered in the optimization process. It shows that consideration of both the information as in deep auto-encoders is successful in identifying communities through user×user matrix in network data.

The proposed SG-DHNMF performs a progressive factorization on a symmetric document co-relation matrix that encodes normalized neighbourhood information via overlapping terms that results in SGNS through factorization. It considers encoding as well as decoding errors in each layer of the deep NMF process to accurately capture the geometric information in the data in resulting factor matrices. There exist only a very few similar works. In [21], each document in a short text corpus is considered as a window and a term-correlation matrix is factorized to boost the performance of short text-based topic modelling. It models words based on their context through word-correlations to overcome the sparsity. It has been shown that applying factorization on a normalized word-correlation matrix is similar to SGNS that encodes the relationship between the word and its context [13].

3 Deep Hierarchical NMF with SGNS-based embedding

Figure 1 illustrates the overall process of SG-DHNMF for identifying clusters in the short text data. Let $D = \{d_1, d_2, \ldots, d_n\}$ be the dataset that contain a set
of $m$ unique terms after standard prepossessing steps such as lemmatizing and stop word removal. Let $A_1 \in \mathbb{R}^{n \times n}$ represent the document × document matrix where each cell models the number of common terms between a document pair. We propose to model $A_1$ with SGNS that becomes input $A$ to NMF.

$$A_1 \in \mathbb{R}^{n \times n}$$

**Fig. 1. Overview of SG-DHNMF**

SG-DHNMF progressively decomposes $A$ into factor matrices $U_g \in \mathbb{R}^{n \times k}$ and $V_g \in \mathbb{R}^{k \times n}$ where $g \in p$ and $p$ is the number of layers (or the level of depth) in hierarchical decomposition. It does so by reducing encoding error $\|A - U_1U_2...U_pV_p\|$ and decoding error $\|V_p - U_p^TU_p^T...U_1^TA\|$ in each iteration and optimizes the process to converge for a given lower rank $k$. The sequential dimensional reduction process allows encoding geometric relationship with nearest neighbour documents to obtain dense representation in each iteration. Finally, SG-DHNMF reconstructs $A'$ (document × document) by multiplying dense factor matrices. A clustering method can be applied to this dense representation to identify cluster assignments. In SG-DHNMF we apply one step NMF to the reconstructed $A'$ with hard assignment policy by setting $k$ as the cluster number.

### 3.1 Semantic Document Representation Learning with SGNS

SG-DHNMF aims to capture the closeness within documents in $A$ with the SGNS modeling. SGNS has been used to highlight the word embedding in text data when representing them in NNs [17]. It can capture the context of a word in a corpus that the simple bag-of-word model fails [13]. The concept of negative (word, context) sampling is used with the Skip-Gram model to maximize the probability of an observed pair while minimizing the probability of unobserved pairs in distributed word representation [21]. SGNS has been proved to be equivalent to factorizing a word correlation matrix whose cells are the point-wise mutual information of the respective word and context pairs [13]. Specifically, factorizing a word correlation matrix with SGNS can model the closely related words with higher coefficients near to 1 while producing lower coefficients for loosely related words.
Distinct from the previous work, we use SGNS in SG-DHNMF to effectively encode the sparse text data by capturing the documents similarity and represent the input matrix to the NMF process. It maximizes the weight for the document pairs that show closer semantic similarity (i.e., share more common terms) in comparison to the others while minimizing the weight of document pairs that show fewer similarity. We conjecture that representing the input matrix with neighborhood information will capture the geometric structure inherent in the collection and utilise the relatedness while projecting the high-order dimensional data to low-rank data. The low-rank data will exhibit similar documents in closer space and non-similar documents in distant space. Hence, the low-rank representation obtained will improve the accuracy of a clustering solution.

Let $d_i, d_j$ be a document pair in $D$. we model the closeness between them based on their shared terms with respect to rest of the documents in the collection as in Eq. (1).

$$A_{(d_i, d_j)} = \log \left( \frac{c_{(d_i, d_j)} \times T}{\sum_{d_a \in D} c_{(d_a, d_i)} \times \sum_{d_a \in D} c_{(d_a, d_j)}} \right) \text{ where } c_{(d_i, d_j)} > 0 \quad (1)$$

where $T$ is the total number of terms shared by the all the document pairs in $D$. Eq. (1) calculates a ratio of the number of terms shared between a document pair with the number of terms that are shared by each of these documents with others in the collection. Let $c_{(d_i, d_j)}$ be the original cell value that represents the terms shared between $d_i$ and $d_j$. It is divided by the sum of the values in the $d_i$ row and $d_j$ column. Document pairs that do not share any terms, the SGNS value for them is set to 0. The cell values $A_{d_i, d_j}$ whose arguments of log are less than 0 are converted to 0 to minimize the probability of document pairs that show less similarity [13]. This step ensures that the input to NMF remains positive and will improve group identification.

### 3.2 Feature Learning with Hierarchical NMF

The matrix $A$ modelled with SGNS becomes input to the deep factorization process. The progressive factorization in NMF enables SG-DHNMF to learn document context relationship accurately as it captures higher-abstract features with every iteration of projection in comparison to learning latent features based on one-step lower dimensional projection. The hierarchical representation of NMF enforces lower to higher level feature learning with each progression. Generally, NMF factorizes a given symmetric input matrix $A \in R^{n \times n}$ into two factor matrices $U \in R^{n \times k}$ and $V \in R^{k \times n}$ as in Eq. (2), where $k$ is the lower rank that generally be the required cluster number.

$$\min_{U, V \geq 0} \|A - UV\|_F \quad s.t \ U \geq 0, V \geq 0 \quad (2)$$

The SG-DHNMF factorization process with $p$ layers starts factorizing the input matrix $A$ into two non-negative factor matrix pairs $(U_1, V_1)$ and proceeds with factorizing each $V_g$ at each layer $g+1 \in p$ hierarchically. $p$ is an empirical parameter that represents the number of layers on which NMF is applied progressively.
In the short text data, the hierarchical representation learning through factorizing a matrix model with SGNS concept allows the data to promote document co-occurrences to deal with sparsity.

\[ A \approx U_1 U_2 \ldots U_p V_p \]  

where \( V_p \in R^{k \times n} \), \( U_g \in R^{r_{g-1} \times r_g} \) where we set \( n = r_0 \geq r_1 \geq \ldots \geq r_p = k \) and \( 1 \leq g < p \).

In the short text data where a fewer number of co-occurrences in terms exist, it becomes difficult to accurately identify the factors only considering encoding of input information. It is important to validate identified factors with decoding that inversely track factors and approximate them through input data. SG-DHNMF attempts to minimise the total approximation error by using both encoding and decoding to obtain an optimum lower-order dense representation. The decoding component, given in Eq. (4), is included in the objective function of SG-DHNMF as follows.

\[
V_p \approx U_p^T U_{p-1}^T \ldots U_1^T A
\]

\[
\min_{V_g, V_g \geq 0} ||A - U_1 U_2 \ldots U_p V_p||_F^2 + \min_{V_g, V_g \geq 0} ||V_p - U_p^T U_{p-1}^T \ldots U_1^T A||_F^2
\]

s.t \( U_g \geq 0, V_g \geq 0 \)

(5)

The objective function of SG-DHNMF as in Eq. (5) calculates the total encoding error within \( p \) layer with the first component (Eq. (3)) and the total decoding error within \( p \) layer with the second component (Eq. (4)) in each iteration. The total error is attempted to minimize over the iterations in obtaining accurate dense factors for short text. This use of reconstruction loss ensures to capture the geometric information accurately in factor matrices. Fig. 2 illustrates the impact of hierarchical NMF in comparison to a single NMF using a toy dataset. The multi-layer progressive factorization \( (k = 20, 10, 5) \) can achieve the denser and accurate lower-order representation in comparison to the one-step projection \( (k = 5) \).

**Update Rules for SG-DHNMF** We initially pre-train each layer to have initial approximation of factor matrices \( U_g \) and \( V_g \) by simply decomposing the input matrix for each layer as follows. This pre-training process starts by decomposing \( A \) as \( A \approx U_1 V_1 \) by minimizing \( ||A - U_1 V_1||_F^2 + ||V_1 - U_1^T A||_F^2 \) where \( U_1 \in R^{n \times r_1} \) and \( V_1 \in R^{r_1 \times n} \). Matrix \( V_1 \) is then decomposed as \( V_1 \approx U_2 V_2 \) by minimizing \( ||V_1 - U_2 V_2||_F^2 + ||V_2 - U_2^T V_1||_F^2 \) where \( U_2 \in R^{r_1 \times r_2} \) and \( V_2 \in R^{r_2 \times n} \).
This process is continued until all the $p$ layers are pre-trained. This type of pre-training has been found effective and efficient [25]. It has greatly reduced the training time of the model as it gives better initialization for the model.

In each iteration of the optimization process of Eq. (5), entries of the factor matrices for each layer are updated sequentially following multiplicative update rule principles. Following update rules for $U_g$ and $V_g$ have been derived based on the objective function using a derivative process similar to [25] for minimizing the total error.

$$U_g \leftarrow U_g \odot \frac{2\psi_T^{g-1} A V^T_p \Phi^{g+1}}{\psi_T^{g-1} \psi_{g-1} U_1 \Phi_{g+1} V_p V^T_p \Phi^{g+1} + \psi_T^{g-1} A A^T \psi^{g+1} U_g \Phi^{g+1} \Phi_T + 1}$$

(6)

where $\psi_{g-1} = U_1 U_2 \ldots U_{g-1}$ and $\Phi_{g+1} = U_g+1 \ldots U_{p-1} U_p$. When $g=1$ and $g=p$, we set $\psi_0 = I$ and $\Phi_{p+1} = I$ respectively.

We start updating each matrix $U_g$ as in Eq. (6), and then update $V_g$ for lower rank $k$ within each iteration as follows:

$$V_g \leftarrow V_g \odot \frac{2\psi_T^g A}{\psi_T^g \psi_g V_g + V_g}$$

(7)

This process used in SG-DHNMF is illustrated in Algorithm 1.

---

**Algorithm 1: The SG-DHNMF algorithm**

**Input**: The Document-Document matrix model with SGNS weighting $A$, Number of layers $p$ with the configuration, Number of Clusters $k$

**Output**: The final Document-Cluster matrix $C$

**while** Convergence of Eq. (5) where number of iterations $\leq 100$ **do**

**foreach** $g=1: p$ **do**

- Compute $U_g$ using Eq. (6)
- Compute $V_g$ using Eq. (7)

**end**

**end**

$A' = V_p U_p$

$C \leftarrow$ Apply NMF with hard clustering policy on $A'$ to assign to $k$ clusters

---

4 Empirical Analysis

**Datasets**: Two publicly available tweets [26] and stack overflow [24] with their ground-truth labels, used in prior short text clustering research, have also been used. Two additional twitter datasets from Trisma (https://trisma.org/) spanning across discussions on Cancer types and University Education have been used. The DS1:cancer dataset consists of 8 cancer types and the DS2:Edu dataset consists of 7 subject streams. These subgroups are considered as clusters. Stop words were removed. Terms with $>90\%$ frequency and $<3$ were removed. Table 1 reports the details of the datasets as well as the depth of the hierarchical-NMF model and the layer configuration at each depth. The layer configurations have been set based on experiments to systematically reduce the input data matrix.
**Baselines:** SG-DHNMF is compared with traditional unsupervised clustering methods including traditional NMF, Latent Dirichlet allocation (LDA) and k-means [1]. SG-DHNMF is also compared with the state-of-the-art unsupervised methods proposed to address the sparseness in short text, (1) Gibbs Sampling algorithm for the Dirichlet Multinomial Mixture model for short text clustering (GSDMM) [26], (2) Short-text topic modeling via NMF [21] (SeaNMF) that combines SGNS and NMF, (3) Deep autoencoder-like NMF [25] for community detection with sparse data that combines deep learning and NMF, and (4) k-means clustering with the document-level embedding, Doc2Vec [11] that can be used as an alternative way to obtain dense representation. Doc2Vec which creates a numeric representation that includes positive and negative numbers for a document limits us to use Doc2Vec with NMF.

Additionally, we compare unsupervised SG-DHNMF against the commonly used dense representation learning supervised methods to show how well SG-DHNMF can learn deep features without the guidance. State-of-the-art single-layer shallow auto-encoder [10], RNN with Gated Recurrent Unit (RNN-GRU) [10] and CNN [10], that rely on ground-truth labels, are used. Finally, we compare the use of SGNS concept in learning the dense text representation accurately.

**Evaluation metrics:** Two measures used to evaluate the accuracy of the short text clustering are standard pairwise F1-score (F1) which calculates the harmonic average of precision and recall, and Normalized Mutual Information (NMI) which measures the purity against the number of clusters [18].

| Dataset  | # Docs | # Terms | #Clusters | Sparsity | Layers Configuration |
|----------|--------|---------|-----------|----------|----------------------|
| DS1:Cancer | 20568  | 8851    | 8         | 0.9913   | 800-160-32-8         |
| DS2:Uq    | 7504   | 5522    | 7         | 0.9974   | 700-140-28-7         |
| DS3:tweet | 2472   | 5077    | 89        | 0.9422   | 356-89               |
| DS4:Stackoverflow | 16407  | 2302    | 20        | 0.9413   | 2000-400-80-20       |

**Comparison with traditional clustering methods:** Fig. 3 shows the comparative results of SG-DHNMF with traditional shallow clustering methods when the input matrix is document × document representation encoded without SGNS and reconstruction loss. It also shows the results on traditional methods when the input matrix is used in classical way of term × document matrix. Results show that the document × document matrix representation is able to produce better results for traditional clustering methods. Additionally, it depicts that SG-DHNMF is superior to traditional shallow clustering methods as it uses step-wise deep learning to identify dense representation for text. This learning process shows a significant impact on large sparse datasets such as DS1.

**Comparison with unsupervised short text mining methods:** Table 2 details the performance of SG-DHNMF with state-of-the-art unsupervised methods that have been designed to handle sparsity in the data. The comparison with SeaNMF shows the superiority of using the SGNS concept in SG-DHNMF and with DANMF shows the superiority of using hierarchical learning.
in SG-DHNMF. The Doc2Vec encoding with k-means that extends the word-embedding concept to document embedding shows the least performance as there exists low word co-occurrences among short documents that this method can be benefited on. GSDMM which uses probability calculation face challenges due to less word co-occurrences and is unable to capture probabilities. DANMF uses reduced representation of network data for community detection balancing encoding and decoding, using a very large user to user network. However, DANMF does not perform well with extremely sparse short text data. SeaNMF uses the term×term relationship to learn word-context relationship with NMF and is able to achieve the best results among the baselines. Datasets DS3 and DS4 show generally good results with other methods such as SeaNMF, DANMF and Doc2Vec in comparison to DS1 and DS2 which are more denser. In highly sparse datasets SG-DHNMF produce superior results.

**Table 2.** Performance comparison with state-of-the-art unsupervised methods

| Dataset | SG-DHNMF NMI | GSDMM NMI | SeaNMF NMI | DANMF NMI | Doc2Vec NMI | RNN F1 | CNN F1 |
|---------|--------------|-----------|------------|-----------|--------------|-------|-------|
| DS1     | 0.94         | 0.93      | 0.06       | 0.17      | 0.11         | 0.32  | 0.03  |
| DS2     | 0.31         | 0.45      | 0.01       | 0.22      | 0.08         | 0.81  | 0.17  |
| DS3     | 0.86         | 0.74      | 0.8        | 0.57      | 0.87         | 0.75  | 0.75  |
| DS4     | 0.65         | 0.63      | 0.39       | 0.31      | 0.6          | 0.55  | 0.56  |
| **Avg** | **0.69**     | **0.69**  | **0.32**   | **0.42**  | **0.44**     | **0.61** | **0.38** |

**Comparison with supervised deep learning methods:** Deep learning methods have been commonly developed to learning dense representation with using ground-truth labels. They have been rarely used in unsupervised setting, except only a handful such as DANMF [25], benchmarked in Table 2. Results in Table 3 show the performance of state-of-the-art supervised methods in categorising documents to respective clusters as classes. RNN and CNN based methods trained with ground-truth data only perform 4.35% and 7.25% better than SG-DHNMF in NMI that is not trained with ground-truth data. In spite of training, the shallow auto-encoder method shows inferior performance. It highlights the importance of deep learning embedded with hierarchical NMF in SG-DHNMF without supervision and shows that it even outperforms supervised shallow methods.
Table 3. Comparison of SG-DNMF with Supervised deep learning methods

| Dataset | SG-DHNMF | Autoencoder-shallow | RNN-GRU | CNN |
|---------|----------|---------------------|---------|-----|
|         | NMI      | F1-score            | NMI     | F1-score |
| DS1     | 0.94     | 0.93                | 0.01    | 0.25  |
| DS2     | 0.31     | 0.45                | 0.01    | 0.84  |
| DS3     | 0.86     | 0.74                | 0.05    | 0.13  |
| DS4     | 0.65     | 0.63                | 0.01    | 0.1   |
| Avg     | 0.69     | 0.69                | 0.02    | 0.33  |

Fig. 4. With and Without Skip-Gram With Negative Sampling

**Impact of SGNS:** The SGNS concept has been used in learning the word-context relationship in text mining [13]. In SG-DHNMF, we have proposed to use SGNS in learning the document-context relationship. Fig. 4 shows that the document-context relationship can be learnt accurately with the SGNS modelling and factorizing the document correlation matrix. DS1 and DS2 which are extremely sparse show higher boost with SGNS modelling by capturing the document-correlations in modelling.

**Sensitivity Analysis** We evaluate the depth of the layers used in NMF for deep learning the low-order features. Results in Table 4 show that this parameter depends on the nature and size of the dataset. This is similar to hyper-parameters tuning in neural network-based methods. The best coefficients for factor matrices are identified in iterative fashion reducing the encoding and decoding error. We measure the total error for 100 iterations and reported the normalized total error as in Fig. 5 (a). It depicts that SG-DHNMF converges within this 100 iterations for all the datasets. Figure. 5(b) shows how performance varies with sparsity in datasets. We have chosen a subset of cancer dataset (DS1) with different cluster numbers to form varying sparse datasets. Results show that performance increases with the sparsity in datasets. Fig. 5(c) shows time taken for hierarchical NMF-based model training against the data size considering the subsets of Cancer dataset. SG-DHNMF shows a trend close to quadratic efficiency when double the sample size, similar to a NMF based method.

**Complexity Analysis** The computational complexity of SG-DHNMF is higher than any methods that use term×document matrix as an input. It includes an additional step calculating the document×document matrix by measuring pair-wise similarity in the document set. Excluding this additional pair-wise comparison, computational complexity of SG-DHNMF is $O(p(n^2 r + nr^2))$ where $n$ is the number of documents, $p$ is the number of layers in HNMF with $(n >> p)$,
Table 4. Performance comparison using different number of layers

| Dataset | 1 layer | 2 layer | 3 layer | 4 layer | 5 layer |
|---------|---------|---------|---------|---------|---------|
|         | NMI     | F1      | NMI     | F1      | NMI     | F1      | NMI     | F1      | NMI     | F1      |
| DS1     | 0.66    | 0.65    | 0.81    | 0.74    | 0.74    | 0.94    | 0.93    | 0.65    | 0.64    |
| DS2     | 0.19    | 0.38    | 0.3     | 0.45    | 0.22    | 0.39    | 0.31    | 0.45    | 0.25    | 0.39    |
| DS3     | 0.83    | 0.69    | 0.86    | 0.74    | 0.83    | 0.7     | -       | -       | -       |
| DS4     | 0.62    | 0.59    | 0.64    | 0.62    | 0.64    | 0.62    | 0.65    | 0.63    | -       | -       |

Fig. 5. Convergence and Performance with sparsity and scalability

and $r$ is the maximum layer size in layer configuration out of all layers. This is similar to complexity of DANMF. However, seaNMF has $O(n^2)$ complexity [21] while GSDMM [26] and Doc2vec [11] have $O(n \log(v))$ and $O(knl)$ complexity respectively where $v$ is the size of the vocabulary, $l$ is the average document length and $k$ is the number of groups/clusters. In contrast, deep NN models have higher computation complexity to SG-DHNMF.

5 Conclusion

We present a novel unsupervised method for short text clustering by learning a feature representation with deep NMF. The short text data shows extreme sparseness and fewer co-occurrences and creates additional challenges for a clustering algorithm to learn categories. This paper develops a feature learning method with the progressive use of NMF similar to deep NNs to explore the document-context relationships and encoding neighbour information within each step. Empirical analysis shows the superiority of SG-DHNMF.

References

1. Aggarwal, C.C., Zhai, C.: Mining text data. Springer Science & Business Media (2012)
2. Banerjee, S., Ramanathan, K., Gupta, A.: Clustering short texts using wikipedia. In: SIGIR. pp. 787–788. ACM (2007)
3. He, T., Huang, W., Qiao, Y., Yao, J.: Text-attentional convolutional neural network for scene text detection. T IMAGE PROCESS 25(6), 2529–2541 (2016)
4. Hinton, G.E., Salakhutdinov, R.R.: Reducing the dimensionality of data with neural networks. science 313(5786), 504–507 (2006)
5. Hosseini-Asl, E., Zurada, J.M., Nasraoui, O.: Deep learning of part-based represen-
tation of data using sparse autoencoders with nonnegativity constraints. T NEUR
NET LEAR 27(12), 2486–2498 (2015)
6. Hu, X., Liu, H.: Text analytics in social media. In: Mining text data, pp. 385–414.
Springer (2012)
7. Hu, X., Sun, N., Zhang, C., Chua, T.S.: Exploiting internal and external semantics
for the clustering of short texts using world knowledge. In: CIKM. pp. 919–928.
ACM (2009)
8. Jia, C., Carson, M.B., Wang, X., Yu, J.: Concept decompositions for short text clus-
tering by identifying word communities. Pattern Recognition 76, 691–703 (2018)
9. Jin, O., Liu, N.N., Zhao, K., Yu, Y., Yang, Q.: Transferring topical knowledge from
auxiliary long texts for short text clustering. In: CIKM. pp. 775–784. ACM (2011)
10. Kowsari, K., Jafari Meimandi, K., Heidarysafa, M., Mendu, S., Barnes, L., Brown,
D.: Text classification algorithms: A survey. Information 10(4), 150 (2019)
11. Lau, J.H., Baldwin, T.: An empirical evaluation of doc2vec with practical insights
into document embedding generation. arXiv preprint arXiv:1607.05368 (2016)
12. Lee, D.D., Seung, H.S.: Learning the parts of objects by non-negative matrix fac-
torization. Nature 401(6755), 788–791 (1999)
13. Levy, O., Goldberg, Y.: Neural word embedding as implicit matrix factorization.
In: NIPS. pp. 2177–2185 (2014)
14. Luong, K., Balasubramaniam, T., Nayak, R.: A novel technique of using coupled
matrix and greedy coordinate descent for multi-view data representation. In:
WISE. pp. 285–300. Springer (2018)
15. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word represen-
tations in vector space. arXiv preprint arXiv:1301.3781 (2013)
16. Mikolov, T., Kombrink, S., Burget, L., Černocký, J., Khudanpur, S.: Extensions of
recurrent neural network language model. In: ICASSP. pp. 5528–5531. IEEE
(2011)
17. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed represen-
tations of words and phrases and their compositionality. In: NIPS. pp. 3111–3119
(2013)
18. Mohotti, W.A., Nayak, R.: Corpus-based augmented media posts with density-
based clustering for community detection. In: ICTAI. pp. 379–386. IEEE (2018)
19. Pennington, J., Socher, R., Manning, C.D.: Glove: Global vectors for word represen-
tation. In: 2014 EMNLP. pp. 1532–1543 (2014)
20. Roy, A., Park, Y., Pan, S.: Learning domain-specific word embeddings from sparse
cybersecurity texts. arXiv preprint arXiv:1709.07470 (2017)
21. Shi, T., Kang, K., Choo, J., Reddy, C.K.: Short-text topic modeling via non-
negative matrix factorization enriched with local word-context correlations. In:
2018 WWW. pp. 1105–1114 (2018)
22. Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C.D., Ng, A.Y., Potts, C.: Recursive deep models for semantic compositionality over a sentiment treebank.
In: 2013 EMNLP. pp. 1631–1642 (2013)
23. Song, H.A., Lee, S.Y.: Hierarchical representation using nmf. In: ICONIP. pp. 466–
473. Springer (2013)
24. Xu, J., Xu, B., Wang, P., Zheng, S., Tian, G., Zhao, J.: Self-taught convolutional
neural networks for short text clustering. Neural Networks 88, 22–31 (2017)
25. Ye, F., Chen, C., Zheng, Z.: Deep autoencoder-like nonnegative matrix factoriza-
tion for community detection. In: CIKM. pp. 1393–1402 (2018)
26. Yin, J., Wang, J.: A dirichlet multinomial mixture model-based approach for short
text clustering. In: SIGKDD. pp. 233–242 (2014)