Array 13 (2022) 100124

Contents lists available at ScienceDirect

Array

journal homepage: www.sciencedirect.com/journal/array

Deep autoencoder-based fuzzy c-means for topic detection

Hendri Murfi *, Natasha Rosaline, Nora Hariadi

Department of Mathematics, Universitas Indonesia, Depok, 16424, Indonesia

ARTICLE INFO

Keywords:
Topic detection
Clustering
Deep learning
Autoencoder
Fuzzy c-means

ABSTRACT

Topic detection is a process for determining topics from a collection of textual data. One of the topic detection methods is clustering based, which assumes that the centroids are topics. The clustering method has the advantage that it can process data with negative representations. Therefore, the clustering method allows a combination with a broader-representation learning method. In this paper, we adopt deep learning for topic detection by using a deep autoencoder and fuzzy c-means called "deep autoencoder-based fuzzy c-means". The encoder of the autoencoder performs a lower-dimensional representation learning. Fuzzy c-means groups the lower-dimensional representation to identify the centroids. The autoencoder's decoder transforms the centroids back into the original representation to be interpreted as the topics. Our simulation shows that deep autoencoder-based fuzzy c-means improves the coherence score of eigenspace-based fuzzy c-means and is comparable to the leading standard methods, i.e., nonnegative matrix factorization or latent Dirichlet allocation.

1. Introduction

Topic detection is a process used to analyze words in a collection of textual data to determine the topics in the collection, how they relate to one another, and how they change over time. The topics are usually represented by a set of words. The coherence of the words usually measures the topic's interpretability. The standard topic detection methods are *nonnegative matrix factorization* (NMF) [1], *clustering* [2], and *latent Dirichlet allocation* (LDA) [3].

In the clustering method, the cluster centers or centroids are interpreted as a topic. In other words, the clustering method groups the textual data based on their topic similarity. Unlike the other two methods, clustering can process data with negative representations. Therefore, the clustering method allows a combination with broader representation learning or dimension reduction methods. Nur'aini et al. combines *k-means* and *latent semantic analysis* (LSA) for topic detection [4]. Firstly, the textual data are transformed into a lower-dimensional eigenspace using *singular value decomposition* (SVD). Next, *k*-means is performed on the eigenspace to extract the topics that are then transformed back to the nonnegative subspace of the original space.

The *k*-means method splits the textual data into *k* clusters in which each textual datum belongs to the nearest centroid. This means that the *k*-means method assumes that each textual datum contains only one topic. This assumption is relatively weak and also differs from the standard NMF and LDA, considering that textual data may have many topics. Therefore, soft clustering is examined to be an alternative clustering method for topic detection. *Fuzzy c-means* (FCM) is one of the famous soft clustering methods [5]. Using FCM, the textual data may belong to more than one cluster and may have more than one topic. The combination of FCM and LSA called *eigenspace-based fuzzy c-means* (EFCM) is proposed for topic detection [6]. In general, some simulations show that EFCM gives coherence scores between those of LDA and NMF [7–10].

Currently, deep learning is the primary machine learning method for unstructured data such as images and text [11,12]. Deep learning has been extensively studied to extract a good representation of data by neural networks [13]. In this paper, we adopt deep learning to improve the performance of EFCM for topic prediction problems by using *deep autoencoder* (DAE) for the representation learning process. We call this topic detection method *deep autoencoder-based fuzzy c-means* (DFCM). First, the encoder of DAE performs a lower-dimensional representation learning. Next, FCM groups the lower-dimensional representation to identify the centroids. Finally, the decoder of DAE transforms the centroids back into the original representation to provide the topics. Our simulation shows that DFCM improves the coherence score of EFCM and is comparable to the leading standard methods, i.e., NMF or LDA.

This paper's outline follows: in Sections 2 and 3, we describe the related works and the methods, i.e., FCM, DAE, and DFCM. Section 4 describes the results and the discussion of our simulations. Finally, a general conclusion for the results is presented in Section 5.

* Corresponding author.
E-mail addresses: hendri@ui.ac.id (H. Murfi), natasha.rosaline@sci.ui.ac.id (N. Rosaline), nora.hariadi@sci.ui.ac.id (N. Hariadi).

https://doi.org/10.1016/j.array.2021.100124
Received 16 December 2020; Received in revised form 30 August 2021; Accepted 20 December 2021
Available online 23 December 2021

2. Related works

Topic detection methods are algorithms for discovering the topics or themes from an unstructured collection of documents. Some recent publications show the growing use of topic detection for researchers in library and information science to find the theme that they are interested in, and then examine the related documents [14–16].

The standard topic detection methods are NMF [1,17,18], clustering [2,19], and LDA [3,20–22]. Unlike the other two methods, clustering is a general method of grouping data. Furthermore, this method can also process positive and negative data representation. Thus, the clustering method is more flexible to be combined with representation learning or dimension reduction.

Fuzzy clustering is one of the most widely used clustering methods because it is soft and flexible in grouping data to the cluster [23]. Bezdek developed FCM by extending the fuzzifier value m to m > 1 [5]. This extension makes FCM a generalization from *k*-means, which is hard clustering. FCM is more suitable for the topic detection method because it allows adaptation to a document's condition with one or more topics, namely by finding the optimal fuzzifier value m.

In the era of big data, the existence of high-dimensional data is a big challenge for FCM [24]. By finding a new representation of the original data, two approaches are already used to reduce the difficulty of FCM in high-dimensional data. The first approach uses kernel methods to implicitly get more expressive features by formulating the data into the feature space constructed by some kernel functions [25,26]. The second approach is an explicit transformation of the original data. In addition to the specified nonlinear data transformations [27], random projection [28] is commonly used to obtain low-dimensional data. The second approach is more suitable because the excellent design of kernel space for clustering is complicated. The ability of kernel methods to handle large-scale data is also a concern. In several empirical studies, the combination of FCM with the data transformation approach [7,9,10] provides better performance than the random projection approach [29] for topic detection problems.

Currently, deep learning is the primary machine learning method for unstructured data such as images and text [11,12]. Deep learning has been extensively studied to extract a good representation of data by neural networks [13]. Combining the deep neural network and an unsupervised clustering method has also become an active research field [30]. In general, there are several approaches to combining deep learning and clustering. The first approach combines representation learning and clustering in two steps: using a DAE for representation learning and a clustering method for the next stage [30,31]. The second approach combines a DAE and a clustering method simultaneously [32,33]. The following approach combines clustering with a pre-trained encoder, such as Bidirectional Encoder Representations from Transformers proposed by Google [34]. However, most of the clustering methods used in these approaches are hard clustering. Few studies work on the improvement of feature quality by deep learning for fuzzy clustering. This study aims to find a good deep representation for fuzzy clustering, i.e., FCM. This research uses the first and second approaches to combine representation learning and clustering. In this approach, representation learning and clustering are carried out separately, not simultaneously, as in the second approach. This approach still requires the decoder part to transform the data back into the original representation. In addition, our approach does not use a pre-trained model because it is still challenging to determine the most important words to represent the resulting topics.

3. Methods

Let *A* be a word by document matrix and *c* be the number of topics. Given *A* and *c*, the topic detection problem is how to recover *c* topics from *A*. In the clustering-based topic detection method, the clustering centers or centroids are interpreted as topics. In this section, we describe DFCM for topic detection. First, we review the core methods, i.e., FCM and DAE.

3.1. Fuzzy c-means

Given a dataset in the form of a word by document matrix $A = [\boldsymbol{a}_1\ \boldsymbol{a}_2 \ldots\ \boldsymbol{a}_n]$ and the number of centroids *c*, the goal of FCM can be formulated as the following constrained optimization:

$$\min_{m_{ik},\boldsymbol{q}_i} J = \sum_{i=1}^{c}\sum_{k=1}^{n} m_{ik}^{f}\|\mathbf{a}_k - \mathbf{q}_i\|^2 \tag{1}$$

$$s.t.\quad \sum_{i=1}^{c} m_{ik} = 1,\ \forall k$$

$$0 < \sum_{k=1}^{n} m_{ik} < n,\ \forall i$$

$$m_{ik} \in [0,1],\ \forall i,k$$

where $\boldsymbol{q}_i$ are centroids, m_{ik} is the membership of data point ak in cluster i, $f > 1$ is the fuzzification constant, and $\|.\|$ is any norm. The first constraint ensures that every data point has total membership in all clusters where each membership is in [0,1]. The second constraint guarantees that all clusters are nonempty [5].

The problem of the constrained optimization in Eq. (1) is to find m_{ik} and $\boldsymbol{q}_i$ that minimize the objective function *J*. The standard method to solve the constrained optimization is alternating optimization. First, we choose some initial values for $\boldsymbol{q}_i$. Then we minimize *J* concerning m_{ik}, keeping $\boldsymbol{q}_i$ fixed, giving the following:

$$m_{ik} = \left[\sum_{j=1}^{c}\left(\frac{\|\mathbf{a}_k - \mathbf{q}_i\|}{\|\mathbf{a}_k - \mathbf{q}_j\|}\right)^{2/f-1}\right]^{-1}, \forall i,k \tag{2}$$

Next, we minimize *J* on $\boldsymbol{q}_i$, keeping m_{ik} fixed, giving the following:

$$\mathbf{q}_i = \frac{\sum_{k=1}^{n}\left((m_{ik})^{f}\mathbf{a}_k\right)}{\sum_{k=1}^{n}(m_{ik})^{f}}, \forall i \tag{3}$$

This two-step optimization is iterated until a stopping criterion is fulfilled, e.g., the maximum number of iterations, insignificant changes in the objective function *J*, the membership m_{ik}, or the centroids $\boldsymbol{q}_i$ [35]. The FCM algorithm is described in more detail in Algorithm 1.

According to Eq. (2), the memberships m_{ik} tend to *0* or *1* when the fuzzification constant *f* approaches 1. The bigger the *f*, the fuzzier the memberships m_{ik}. Therefore, the setting of the fuzzification constant is quite intuitive. A small *f* means that each textual datum may contain a small number of topics. Additionally, a bigger *f* implies that each textual datum may have more topics.

Algorithm 1. FCM

```
Input: $A$, $c$, f, max iteration $(T)$, threshold $(\varepsilon)$ Output: $\boldsymbol{q}_i$
1. Set t= 0
2. Initialize $\boldsymbol{q}_i$
3. Update t= t + 1
4. Calculate $m_{ik} = \left[\sum_{j=1}^{c}\left(\frac{\|\mathbf{a}_k - \mathbf{q}_{i2}\|}{\|\mathbf{a}_k - \mathbf{q}_{j2}\|}\right)^{2/f-1}\right]^{-1}, \forall i,k$
5. Calculate $\boldsymbol{q}_i = \frac{\sum_{k=1}^{n}((m_{ik})^{f}\boldsymbol{a}_k)}{\sum_{k=1}^{n}(m_{ik})^{f}}, \forall i$
6. If a stopping, i.e., $t > T$ or $\|M^t - M^{t-1}\|_F < \varepsilon$, is fulfilled then stop, else go back to
   step 3
```

3.2. Deep autoencoder

A DAE is a deep neural network for unsupervised learning problems. This unsupervised problem is solved using a supervised learning

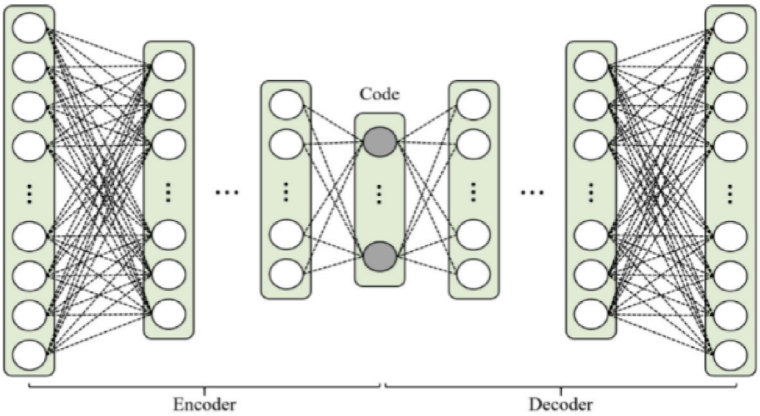

Fig. 1. Deep autoencoders.

approach, where target labels are constructed from input features. This DAE architecture has the same output layer as the input layer, and the standard supervised learning can be applied.

The architecture of DAE for representation learning can be explained in three parts: encoder, code, and decode (Fig. 1). The encoder part consists of fully connected layers used to transform data input to the code part, a new data representation. The decoder part is used to transform the new data representation back to the original representation. The decoder part consists of fully connected layers having a symmetric structure with the encoder part. The reason is if an encoder requires a certain complexity (number of layers (depth), the number of neurons in each layer (units)) to represent data to the new representation, then a decoder with the same complexity is needed to transform the new data representation back to the original data representation. For the dimension reduction problem, the number of neurons in the code part is set less than the number of neurons in the input layer [36].

A DAE can be built layer by layer using greedy layer-wise pretraining, where each layer is built by a denoising autoencoder [37]. The denoising autoencoder is an autoencoder that reconstructs the input from a corrupted version to force the hidden layer to discover a more stable and robust representation.

Given textual dataset $X = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$ with $\mathbf{x}_i \in \mathscr{R}^{D_x}$, $\forall i = 1, 2, \dots, N$, the denoising autoencoder consists of two layers as follows:

$$\widetilde{\boldsymbol{x}}_i \sim dropout_1(\boldsymbol{x}_i) \tag{4}$$

$$\boldsymbol{h}_i = g_1(\widetilde{\boldsymbol{x}}_i, \boldsymbol{w}_1) \tag{5}$$

$$\widetilde{\boldsymbol{h}}_i \sim dropout_2(\boldsymbol{h}_i) \tag{6}$$

$$\boldsymbol{y}_i = g_2(\widetilde{\boldsymbol{h}}_i, \boldsymbol{w}_2) \tag{7}$$

where $dropout_1()$ and $dropout_2()$ are methods to ignore some number of neuron outputs during training randomly, $g_1()$ and $g_2()$ are activation functions, and $\boldsymbol{w}_1$ and $\boldsymbol{w}_1$ are weights. The fitting is performed to minimize the loss $\mathscr{L}(\boldsymbol{x}, \boldsymbol{w})$, i.e., errors between $\boldsymbol{x}_i$ and $\boldsymbol{y}_i$. Next, $\boldsymbol{h}_i$ becomes a new representation for the input data of the next layer. After training on each denoising autoencoder, the denoising autoencoder's weights become the corresponding weights of the autoencoder. Furthermore, the autoencoder is retrained to minimize a reconstruction loss for all layers. This DAE algorithm is described in more detail in Algorithm 2.

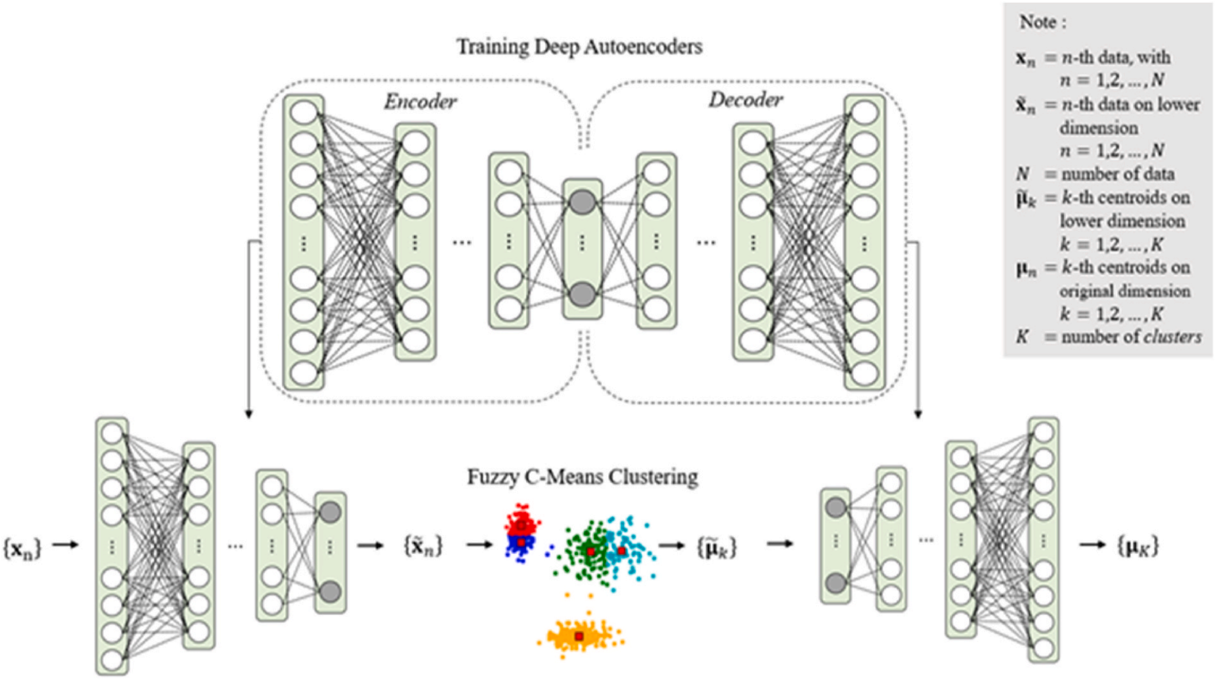
Training Deep Autoencoders
Encoder
Decoder
Fuzzy C-Means Clustering
$\{\mathbf{x}_n\}$
$\{\tilde{\mathbf{x}}_n\}$
$\{\tilde{\boldsymbol{\mu}}_k\}$
$\{\boldsymbol{\mu}_K\}$
Note :
$\mathbf{x}_n$ = n-th data, with $n = 1,2,\dots,N$
$\tilde{\mathbf{x}}_n$ = n-th data on lower dimension $n = 1,2,\dots,N$
N = number of data
$\tilde{\boldsymbol{\mu}}_k$ = k-th centroids on lower dimension $k = 1,2,\dots,K$
$\boldsymbol{\mu}_n$ = k-th centroids on original dimension $k = 1,2,\dots,K$
K = number of *clusters*

Fig. 2. Deep autoencoder-based fuzzy c-means.

3.3. DAE-based fuzzy c-means

The DFCM is a proposed topic detection method that combines DAE for representation learning and FCM for fuzzy clustering. A FCM works well for low-dimensional textual data and generates only one topic for high dimensional textual data. We can set the *f* of FCM with a small value to push FCM to produce more than one topic. However, this small *f* assumes that each textual datum contains few topics and only one topic when *f* approaches one. Therefore, we use DAE to transform the data to lower-dimensional representation and keep *f* adaptable for textual data with multi-topics. Fig. 2 provides a general process of DFCM.

Algorithm 2. DAE

```
Input: $X$, the size of code p
Output: encoder($\mathbf{w}$), decoder($\mathbf{w}$)
1. Initialize autoencoder(p)
2. $\boldsymbol{h}_n^{(0)} = \boldsymbol{x}_n$
3. Let m be the number of layers of the autoencoder
4. FOR i = 1 TO m
5.    Fitting the denoising autoencoder for the $i$-th layer: $deAutoencoder\ (\boldsymbol{w}^{(i)})$, $\boldsymbol{w}^{(i)} = \min_{\boldsymbol{w}} \mathscr{L}(\boldsymbol{h}_n^{(i-1)}, \boldsymbol{w}),\ \forall n$
6.    $\boldsymbol{h}_n^{(i)} = deEncoder(\boldsymbol{h}_n^{(i-1)}),\ \forall n$
7. Initialize weight of autoencoder with the corresponding weight of the denoising
   autoencoder: autoencoder $(\boldsymbol{w}^{(i)}), \forall i$
8. Fitting the autoencoder: autoencoder $(\boldsymbol{w})$, $\boldsymbol{w} = \min_{\boldsymbol{w}} \mathscr{L}(\boldsymbol{x}_n, \boldsymbol{w}),\ \forall n$
```

Given a textual dataset $X = \{\mathbf{x}_1,\ \mathbf{x}_2,\ \ldots,\ \mathbf{x}_N\}$ with $\mathbf{x}_i \in \mathscr{R}^{D_x}$, $\forall i\ =\ 1, 2,\ \ldots,\ N$, the dimension of new data representation *p*, and the number of topic c. First, the textual data are transformed into a lower-dimensional representation using an encoder. We denote this transformation as follows:

$$\widetilde{X} = encoder(X, p) \tag{8}$$

where $\widetilde{\mathbf{x}}_i \in \mathscr{R}^{D_{\widetilde{x}}}$, $\forall i\ = 1, 2,\ \ldots,\ N$. Next, we perform FCM on the dataset $\widetilde{X}$ with the lower-dimensional representation. In this step, centroids $\widetilde{\boldsymbol{\mu}}_i \in \mathscr{R}^{D_{\widetilde{x}}}$, $\forall i = 1, 2,\ \ldots,\ c$ are extracted from all *c*-given clusters as follows:

$$\widetilde{\boldsymbol{\mu}}_i = FCM(\widetilde{X}, c, f, T, \varepsilon) \tag{9}$$

These centroids $\widetilde{\boldsymbol{\mu}}_i$ are interpreted as the topics in the lower-dimensional representation. However, the topics have no meaning and will be meaningful if they are transformed back to the original representation. Therefore, it is necessary to transform the extracted topics back to the original representation as follows:

$$\boldsymbol{\mu}_i = \max(0, decoder(\widetilde{\boldsymbol{\mu}}_i)) \tag{10}$$

where $\boldsymbol{\mu}_i \in \mathscr{R}^{D_x}$, $\forall i\ = 1, 2,\ \ldots,\ c$, and max () is a function that gives a maximum between 0 and each element of $decoder(\widetilde{\boldsymbol{\mu}}_i)$. This DFCM algorithm is described in more detail in Algorithm 3.

Algorithm 3. DFCM

```
Input: $X$, the size of code $p$, the number of topics $c$, max number of iterations T,
   threshold $\varepsilon$
Output: $\boldsymbol{\mu}_i$
1. Build autoencoder: $encoder$, decoder = DAE ($X$, p)
2. Transform X: $\widetilde{X} = encoder(X)$
3. Perform FCM: $\widetilde{\boldsymbol{\mu}}_i = FCM(\widetilde{X}, c, T, \varepsilon),\ i = 1, 2,\ \ldots,\ c$
4. Calculate the topics: $\boldsymbol{\mu}_i = \max(0, decoder(\widetilde{\boldsymbol{\mu}}_i)),\ i = 1, 2,\ \ldots,\ c$
```

4. Results

To examine the performance of DFCM, we apply the method to extract topics on two datasets: English email and Indonesian news. To prepare the textual data for the topic detection methods, we executed two main processes: cleaning and vectorizing. Firstly, we converted all words into lowercase, erased words containing domains such as www.* or https://*, and words containing @username, and erased # in #words. To standardize words with non-standard spelling, we replaced two or more repeating letters with only two occurrences. Stopwords and words with low-frequency terms occurring in fewer than *t* of the total *m* documents were also excluded, where the *t* threshold was set to max (10; *m*/1000). Finally, we used the term frequency-inverse document frequency for weighting.

Given a tweet collection, the topic detection methods produce topics represented by their top 10 most frequent words. The standard quantitative method to measure the interpretability of the topics is topic coherence. In our simulations, we use one of the topic coherence measures called TC-W2V [38]. Suppose a topic $\boldsymbol{t}$ consists of *n* words that are $\{t_1, t_2,\ \ldots, t_n\}$, the TC-W2V of topic $\boldsymbol{t}$ is as follows:

$$TC - W2V(\boldsymbol{t}) = \frac{1}{\binom{n}{2}} \sum_{j=2}^{n} \sum_{i=1}^{j-1} \text{similarity}\left(wv_j, wv_i\right) \tag{11}$$

where wv_j and wv_i are vectors of word t_j and t_i constructed by a *word2vec* model.

The simulations are conducted in a Windows-based Python environment. For EFCM and DFCM parameters, we set $f = 1.1$, the maximum number of iteration $T\ =\ 1000$, and threshold $\varepsilon\ =\ 0.005$. As mentioned before, setting of *f* is quite intuitive. A small *f* means that each tweet may contain a small number of topics. However, a large *f* implies that each textual datum may contain more topics. We initialize the centroids of FCM for both EFCM and DFCM using the best of the 10-run *k*-means clustering. In DFCM, we use commonly used parameters for DAEs and avoid dataset-specific tuning as much as possible. Specifically, we set symmetrical network dimensions to d-500-500-2000-p for all datasets, where *d* is the data-space dimension and *p* is the lower dimension of 10 or 5, as previously implemented [33,39]. We implement this representation learning using Python-based Keras.[1] On the other hand, we use truncatedSVD implementation of scikit-learn for dimension reduction in EFCM [40]. Finally, we need to tune the parameters, i.e., the lower dimension *p* and the number of topics *c*, for EFCM and DFCM.

This simulation also compares DFCM with two standard topic detection methods: LDA and NMF. We use the LDA and NMF implementations provided by scikit-learn [40]. The LDA algorithm uses the batch variational Bayes method for training LDA. Two parameters are usually optimized for this training method: α and η. The α controls the mixture of topics for a specific document; a smaller α means the document will likely have less of a mixture of topics. The η controls the distribution of words per topic; a larger η means the topic will likely have more words. To optimize both parameters, we use a hyperparameter grid and run an algorithm for each combination [0.01, 0.1, 0.25, 0.5, 0.75, 1].

For NMF, the data vectors are normalized to unit length. The implementation of NMF uses a coordinate descent algorithm. There is no parameter we optimize for this algorithm. To reduce the instability of random initialization, the NNDSVD initialization is performed.

4.1. Enron – an English email dataset

The first dataset is Enron consisting of approximately 500,000 emails generated by employees of the Enron Corporation.[2] The Federal Energy Regulatory Commission obtained it during its investigation of Enron's collapse. To calculate the TC-W2V of the extracted topics, we use a pretrained *word2vec* model trained on the Google News dataset for the English email dataset. The model contains 300-dimensional vectors for 3

[1] https://keras.io.

[2] https://www.cs.cmu.edu/~./enron/.

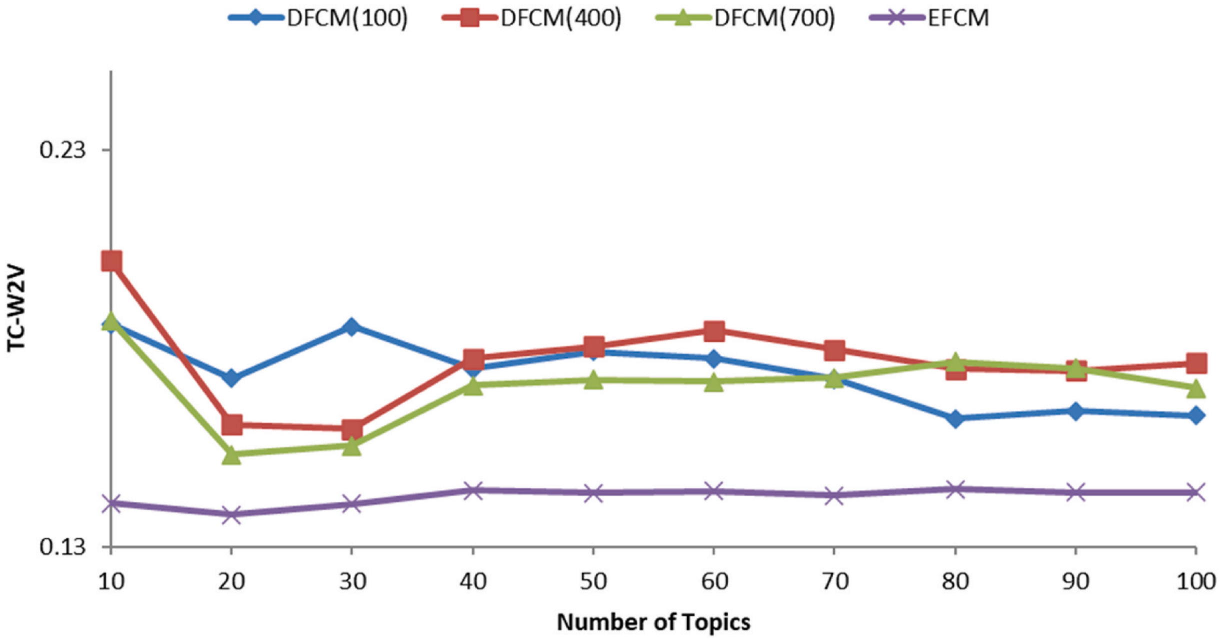

Fig. 3. Coherence scores in terms of TC-W2V for the Enron dataset on the number of topics 10, 20, …, 100 when the lower-dimensional representation is set to five. DFCM(100), DFCM(400), and DFCM(700) mean the number of epochs of deep autoencoders is set to 100, 400, and 700, respectively.

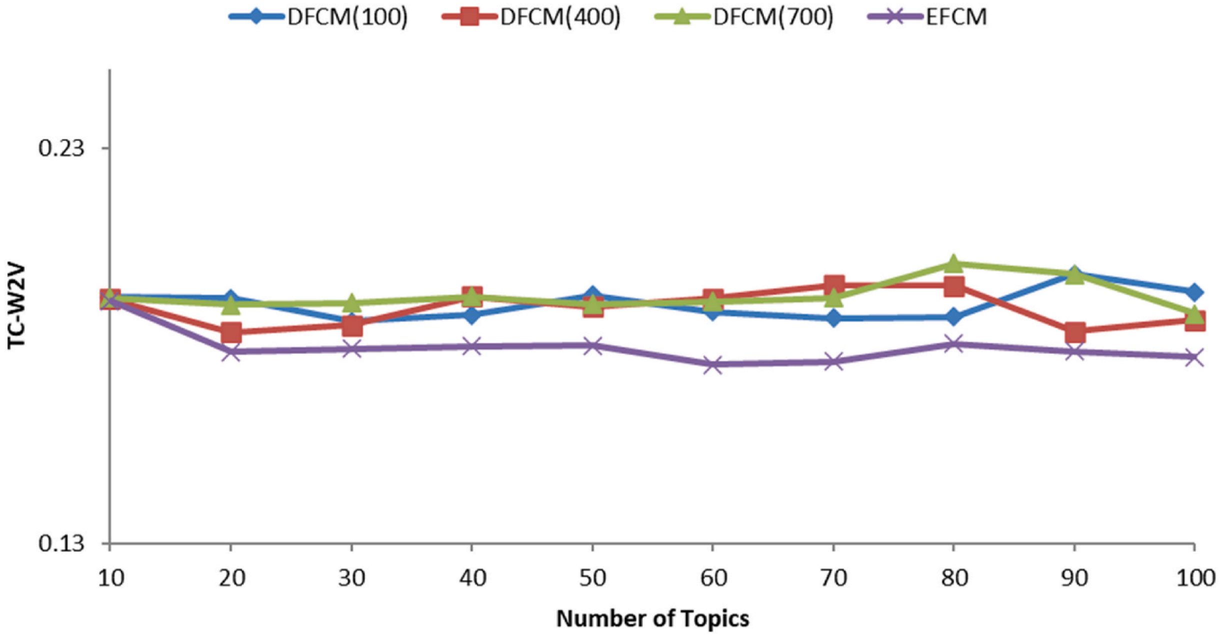

Fig. 4. Coherence scores in terms of TC-W2V for the Enron dataset on the number of topics 10, 20, …, 100 when the lower-dimensional representation is set to 10. DFCM(100), DFCM(400), and DFCM(700) mean the number of epochs of deep autoencoders is set to 100, 400, and 700, respectively.

million words and phrases.[3]

First, we analyze the effect of the number of DAE training epochs on the coherence scores of DFCM. Using a batch size of 256, the coherence scores for several epoch sizes are given in Fig. 3. First, the number of epochs is set to 100. Then, this number of epochs is increased to 400 and 1000. The average coherence score of DFCM fluctuates and increases when the number of epochs is increased from 100 to 400. However, the average coherence score tends to decrease when the number of epochs is increased to 700. If we choose 400 as the number of epochs, then DFCM gives the mean coherence scores of 0.1771, 22% better than EFCM.

Fig. 4 provides simulation results similar to Fig. 3, but for 10-dimensional representation. Compared to the five-dimensional representation, the mean coherence scores of DFCM fluctuate only slightly as the number of epochs increases from 100 to 400 and 700. The DFCM gives the mean coherence scores of 0.1896 for 400 epochs. Like the five-dimensional data representation, DFCM still provides a better mean coherence score than EFCM, which is about 5% better.

Figs. 3 and 4 show that the 10-dimensional representation of data is more suitable for both the DFCM and EFCM methods. The DFCM with 10-dimensional representation gives mean coherence scores 7% better than for DFCM with the five-dimensional representation. Meanwhile,

[3] https://code.google.com/archive/p/word2vec/.

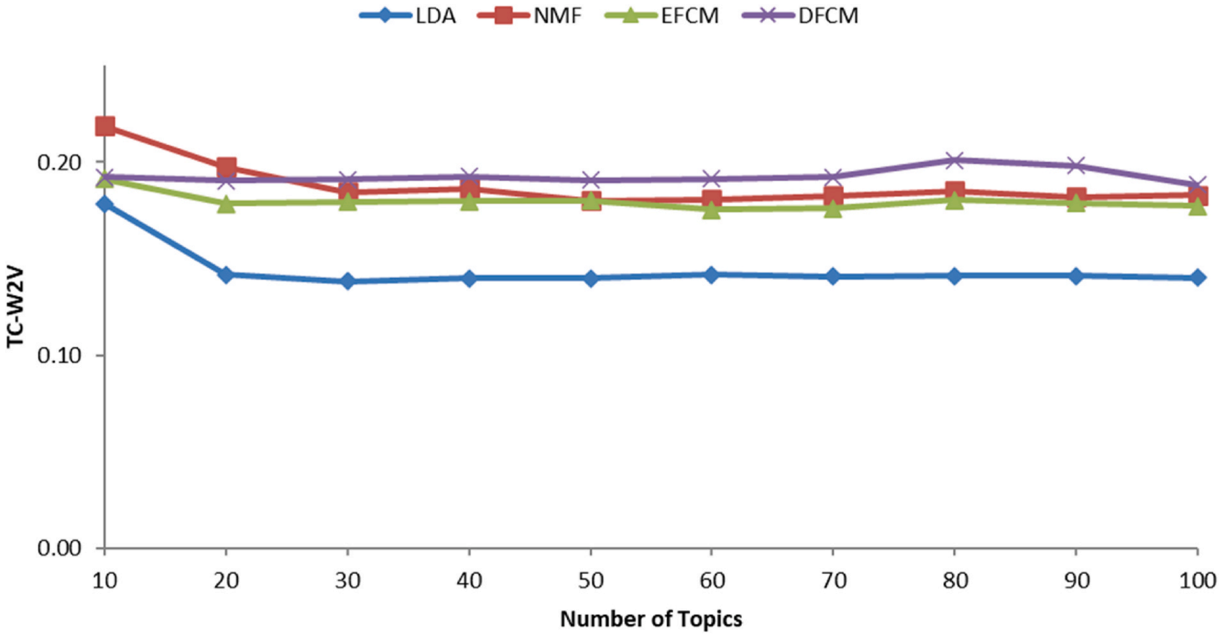

Fig. 5. Comparison of coherence scores in terms of TC-W2V for LDA, NMF, EFCM, and DFCM on the number of topics 10, 20, …, 100 for the Enron dataset.

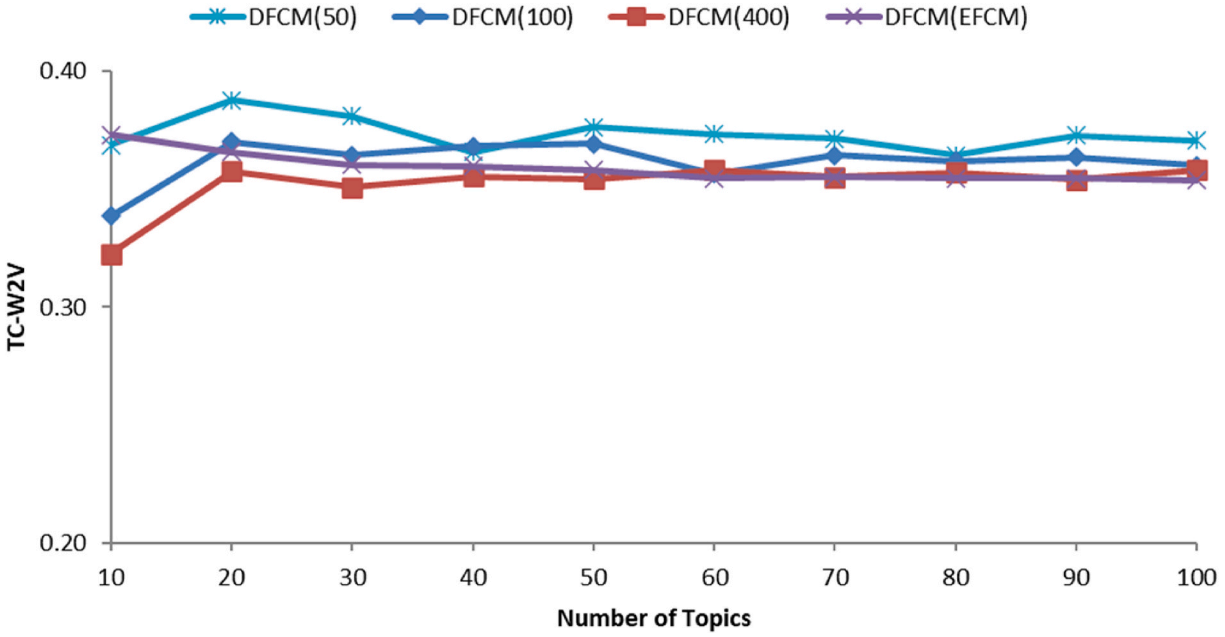

Fig. 6. Coherence scores in terms of TC-W2V for the Berita dataset on the number of topics 10, 20, …, 100 when the lower-dimensional representation is set to five. DFCM(50), DFCM(100), and DFCM(400) mean the numbers of epoch of deep autoencoders are set to 50, 100, and 400, respectively.

EFCM with 10-dimensional representation provides an average coherence score 26% better than EFCM with five-dimensional representation.

Furthermore, we also provide a comparison between the DFCM and two other standard methods: NMF and LDA. Fig. 5 includes coherence scores of four topic detection methods for the number of topics 10, 20, …, 100. First, Fig. 5 confirms the previous simulation results that EFCM provides coherence scores between those of NMF and LDA. The DFCM reaches a coherence score that is slightly better than for NMF in almost all numbers of topics (Fig. 5). Only for the topic number of 10, does NMF give a significantly better coherence score. For all number of topics, DFCM provides better mean coherence scores of 3%, 7%, and 34% compared to NMF, EFCM, and LDA, respectively.

4.2. *Berita – an Indonesian news dataset*

The second dataset is Berita, consisting of 50,304 digital Indonesian news articles shared online through Twitter by nine Indonesian news portals widely known in Indonesia. These are Antara (antaranews.com), Detik (detik.com), Inilah (inilah.com), Kompas (kompas.com), Okezone (okezone.com), Republika (republika.co.id), Rakyat Merdeka (rmol.co), Tempo (tempo.co), and Viva (viva.co.id). The news articles contain published dates, titles, and some first sentences of contents. We construct the *word2vec* model using a corpus consisting of 750,000 Indonesian documents from wiki, news, and tweets to measure the TC-W2V of the extracted topics. Unlike the *word2vec* model for the first English dataset, we train the Berita dataset to this *word2vec* model.

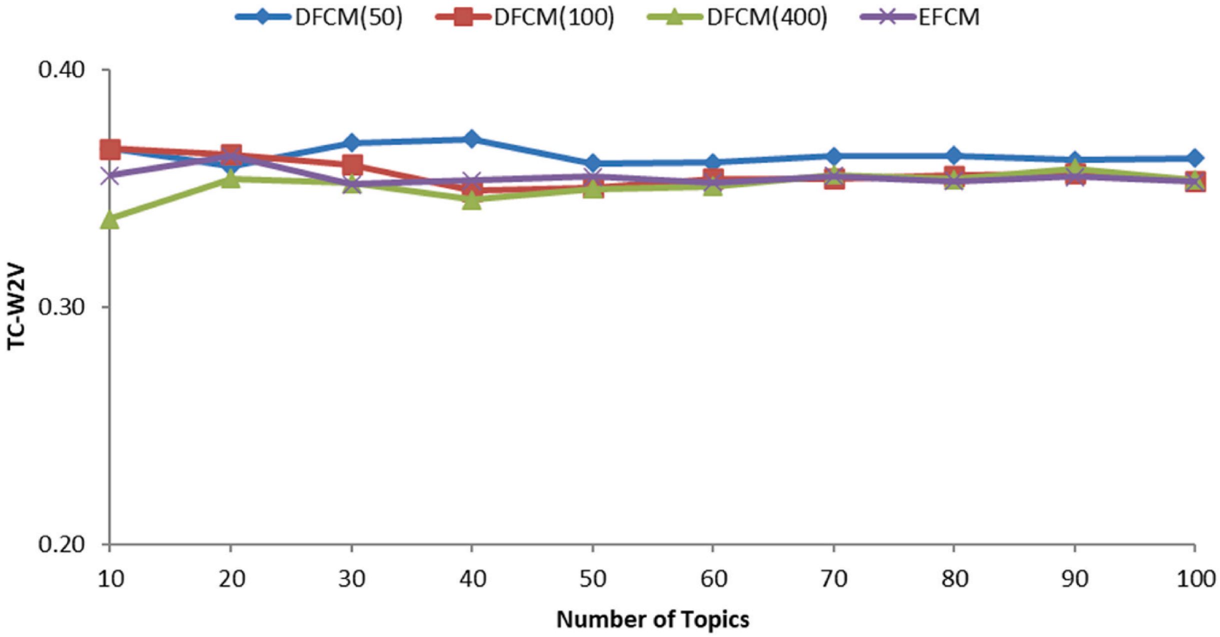

Fig. 7. Coherence scores in terms of TC-W2V for the Berita dataset on the number of topics 10, 20, …, 100 when the lower-dimensional representation is set to 10. DFCM(50), DFCM(100), and DFCM(400) mean the numbers of epoch of DAE are set to 50, 100, and 400, respectively.

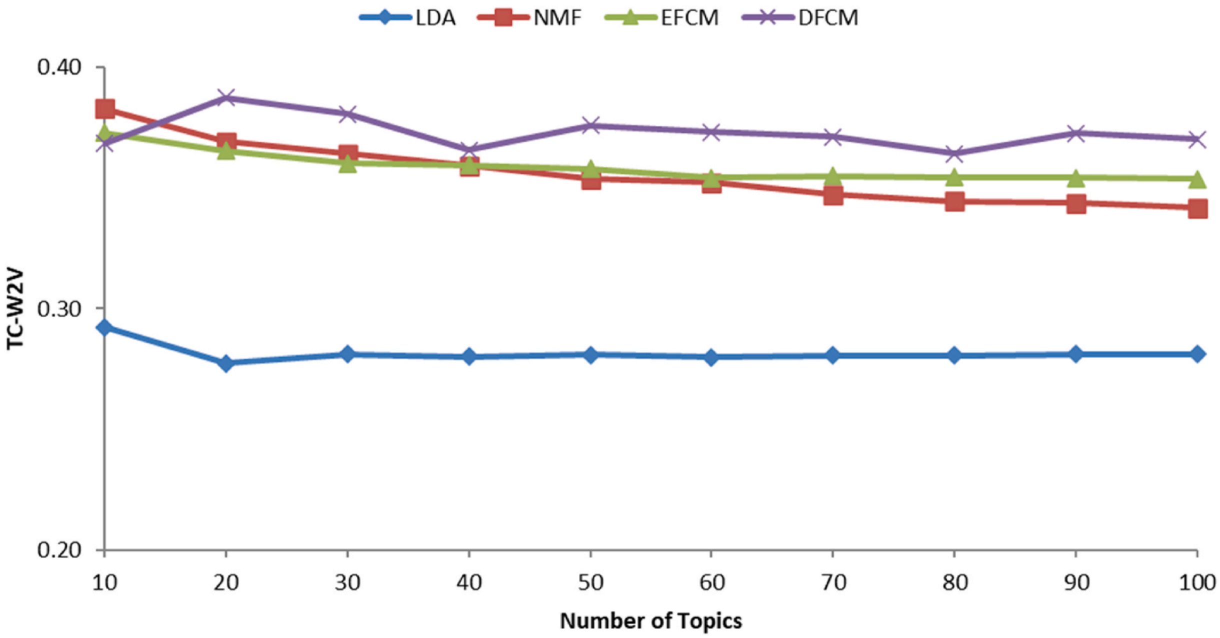

Fig. 8. Comparison of coherence scores in terms of TC-W2V for LDA, NMF, EFCM, and DFCM on the number of topics 10, 20, …, 100 for the Berita dataset.

Therefore, all vocabularies of the Berita dataset exist in the *word2vec* model.

The simulations for the Berita dataset are given in Figs. 6–8. Fig. 6 is a simulation to see the effect of the number of epochs in the DAE learning on coherence scores of DFCM for the five-dimensional representation. Fig. 7 is a simulation to see the effect of the number of epochs in the DAE learning on coherence scores of DFCM for the 10-dimensional representation. The initial number of epochs is 50 and is increased to 100 and then 400. In general, increasing the number of epochs lowers the coherence score for most topics. For the epoch number of 400, DFCM even provides a coherence score below the EFCM in almost all numbers of topics. The same conditions are seen for the 10-dimensional representation in Fig. 7.

If we use 50 epochs for DAE learning, then DFCM gives an average coherence score of 0.3730 for the five-dimensional representation. Meanwhile, EFCM provides an average coherence score of 0.3589. This means that DFCM achieves a slightly higher average coherence score than EFCM, which is about 4% better. A similar result is shown for the 10-dimensional representation where the DFCM gives an average coherence score of about 3%, slightly higher than the EFCM. Figs. 6 and 7 show that the five-dimensional representation provides a slightly better coherence score for both DFCM and EFCM.

Fig. 8 provides a comparison of DFCM on the Berita dataset with two other standard topic detection methods: NMF and LDA. In this comparison, both DFCM and EFCM use the five-dimensional representation. Fig. 8 shows that DFCM, EFCM, NMF, and LDA provide mean coherence scores of 0.3730, 0.3588, 0.3560, and 0.2815, respectively. These results indicate that DFCM achieves a better average coherence score than EFCM, NMF, and LDA. However, NMF still gives a better coherence score for the smallest number of topics, i.e., 10. In this Berita dataset, NMF and EFCM provide almost the same mean coherence scores.

5. Discussion

In the previous sub-chapter, the simulations show that DFCM achieves better mean coherence scores than EFCM. For the Enron dataset, DFCM provides mean coherence scores that are 7% better than the EFCM and 4% more than the EFCM for the news dataset. The main difference between these two methods is the lower-dimensional representation learning process for which DFCM uses DAE, while EFCM uses truncatedSVD. The DAE and truncatedSVD produce different lower-dimensional representations. The truncatedSVD creates a lower-dimensional representation with orthogonal dimensions or features, but DAE produces lower-dimensional representations with dimensions or features that are not orthogonal. Topics generally consist of words that are not necessarily orthogonal, especially in their meaning. Also, DAE implements denoising processes implicitly to produce these lower-dimensional representations. Thus, each of these lower-dimensional characteristics will more or less affect the resulting mean coherence scores.

Compared to NMF, DFCM also provides a higher average coherence score. The DFCM achieved a 3% better average coherence score for the Enron dataset and a 5% better average coherence score for the news dataset. In contrast to EFCM, DFCM and NMF provide lower-dimensional representations with non-orthogonal dimensions or features. The NMF carried out a topic extraction process in the original space, which consisted of words. Thus, the resulting topics can be directly interpreted, and their coherence scores calculated. Meanwhile, DFCM extracts the topics in the lower-dimensional space and must be transformed back to the original space so that the extracted topics can be interpreted and coherence scores calculated. However, DFCM makes it possible to process a better representation for textual data that generally have a lot of noise and variation. Thus, success of DFCM in achieving better coherence scores is mainly because DFCM processes textual data with better representations. The same condition for LDA, where LDA performs the topic extraction process in the original space, consists of words.

Deep learning is currently a popular supervised learning approach, especially for unstructured data such as images and text. Deep learning integrates the feature extraction process with classification or regression processes. In the context of unsupervised learning, DAE is a popular deep learning method for representation learning. This method allows performing the denoising process while reducing dimensions to produce better lower-dimensional representation. However, integration of representation learning methods with an unsupervised learning problem such as topic detection is still an opportunity to be developed.

6. Conclusions

DFCM is a topic detection method that combines DAE for representation learning and fuzzy c-means for topic extraction. Therefore, DFCM allows processing a better representation for textual data, which generally have a lot of noise and variation. Unlike EFCM, DFCM extracts topics from lower-dimensional representations with dimensions or features that are not orthogonal. This representation is more realistic to represent the topics. Our simulation shows that DFCM gives a higher accuracy in terms of the coherence score than EFCM and the two standard methods of NMF and LDA.

Author statement

Hendri Murfi: Conceptualization, Methodology, Writing- Original draft preparation. Natasha Rosaline: Data curation, Visualization. Nora Hariadi: Writing- Reviewing and Editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This paper was supported by Universitas Indonesia, Indonesia under a PDUPT 2020 grant. Any opinions, findings, conclusions, and recommendations are the authors' and do not necessarily reflect those of the sponsor.

References

[1] Lee DD, Seung HS. Learning the parts of objects by nonnegative matrix factorization. Nature 1999;401:788–91.

[2] Allan J. Topic detection and tracking: event-based information organization. Kluwer; 2002.

[3] Blei DM, Ng AY, Jordan MI, Edu BB, Ng AY, Edu AS, Edu JB. Latent dirichlet allocation. J Mach Learn Res 2003;3:993–1022. https://doi.org/10.1162/jmlr.2003.3.4-5.993.

[4] Nur'aini K, Najahaty I, Hidayati L, Murfi H, Nurrohmah S. Combination of singular value decomposition and K-means clustering methods for topic detection on Twitter. In: Nal conference on advanced computer science and information systems (ICACSIS) 2015 internatio. IEEE; 2015. p. 123–8. https://doi.org/10.1109/ICACSIS.2015.7415168.

[5] Bezdek JC, Ehrlich R, Full W. FCM: the fuzzy c-means clustering algorithm. Comput Geosci 1984;10(2):191–203. https://doi.org/10.1016/0098-3004(84)90020-7.

[6] Muliawati T, Murfi H. Eigenspace-based fuzzy c-means for sensing trending topics in Twitter. In: 1862 AIP conference proceedings §; 2017. https://doi.org/10.1063/1.4991244. Depok.

[7] Murfi H. The accuracy of fuzzy C-means in lower-dimensional space for topic detection. Lecture notes in computer science (including subseries lecture notes in artificial intelligence and lecture notes in bioinformatics), vol. 11344. LNCS; 2018. https://doi.org/10.1007/978-3-030-05755-8_32.

[8] Murfi H. Monitoring trending topics of real-world events on Indonesian tweets using fuzzy C-means in lower dimensional space. In: ACM international conference proceeding series; 2019. https://doi.org/10.1145/3369114.3369127.

[9] Nugraha P, Rifky Yusdiansyah M, Murfi H. Fuzzy C-means in lower dimensional space for topics detection on Indonesian online news. Comm Comput Info Sci 2019;1071. https://doi.org/10.1007/978-981-32-9563-6_28.

[10] Nugraha P, Rifky Yusdiansyah M, Murfi H. Fuzzy c-means in lower dimensional space for topics detection on Indonesian online news. In: Tan Y, Shi Y, editors. Data mining and big data. Singapore: Springer Singapore; 2019. p. 269–76.

[11] Goodfellow I, Bengio Y, Courville A. Deep learning. MIT Press; 2016.

[12] Zhang A, Lipton ZC, Li M, Smola AJ. Dive into deep learning. 2020.

[13] Bengio Y, Courville A, Vincent P. Representation learning: a review and new perspectives. IEEE Trans Pattern Anal Mach Intell 2013;35:1798–828. https://doi.org/10.1109/TPAMI.2013.50.

[14] Battsengel G, Geetha S, Jeon J. Analysis of technological trends and technological portfolio of unmanned aerial vehicle. J. Open Innov.: Technol. Mark. Complex. 2020;6:48. https://doi.org/10.3390/joitmc6030048.

[15] Lamba M, Madhusudhan M. Mapping of topics in DESIDOC journal of library and information technology, India: a study. Scientometrics 2019;120(2):477–505. https://doi.org/10.1007/s11192-019-03137-5.

[16] Parlina A, Ramli K, Murfi H. Theme mapping and bibliometrics analysis of one decade of big data research in the Scopus database. Information 2020;11(2). https://doi.org/10.3390/info11020069.

[17] Cichocki A, Phan AH. Fast local algorithms for large scale nonnegative matrix and tensor factorizations. IEICE T. Fund. Electr. E92-A 2009;(3):708–21. https://doi.org/10.1587/transfun.E92.A.708.

[18] Févotte C, Idier J. Algorithms for nonnegative matrix factorization with the β-divergence. Neural Comput 2011;23(9):2421–56. https://doi.org/10.1162/NECO_a_00168.

[19] Petkos G, Papadopoulos S, Kompatsiaris Y. Two-level message clustering for topic detection in Twitter. CEUR Workshop Proceed. 2014;1150:49–56.

[20] Blei DM. Probabilistic topic models. Commun ACM 2012;55(4):77–84.

[21] Hoffman MD, Blei DM, Bach F. Online learning for latent Dirichlet allocation. In: Proceedings of the 23rd international conference on neural information processing systems, ume 1. USA: Curran Associates Inc.; 2010. p. 856–64.

[22] Hoffman MD, Blei DM, Wang C, Paisley J. Stochastic variational inference. J Mach Learn Res 2013;14(1):1303–47.

[23] Ruspini EH, Bezdek JC, Keller JM. Fuzzy clustering: a historical perspective. Compet Intell Mag 2019;14(1):45–55. https://doi.org/10.1109/MCI.2018.2881643.

[24] Winkler R, Klawonn F, Kruse R. Fuzzy c-means in high dimensional spaces. Int J Fuzzy Syst Appl 2011;1(March):2–4. https://doi.org/10.4018/ijfsa.2011010101.

[25] Huang H, Chuang Y, Chen C. Multiple kernel fuzzy clustering. IEEE Trans Fuzzy Syst 2012;20(1):120–34. https://doi.org/10.1109/TFUZZ.2011.2170175.

[26] Shang R, Zhang W, Li F, Jiao L, Stolkin R. Multi-objective artificial immune algorithm for fuzzy clustering based on multiple kernels. Swarm Evol. Comput. 2019;50:100485. https://doi.org/10.1016/j.swevo.2019.01.001.

[27] Zhu X, Pedrycz W, Li Z. Fuzzy clustering with nonlinearly transformed data. Appl Soft Comput 2017;61:364–76. https://doi.org/10.1016/j.asoc.2017.07.026.

[28] Rathore P, Bezdek JC, Erfani SM, Rajasegarar S, Palaniswami M. Ensemble fuzzy clustering using cumulative aggregation on random projections. IEEE Trans Fuzzy Syst 2018;26(3):1510–24. https://doi.org/10.1109/TFUZZ.2017.2729501.

[29] Yusdiansyah MR, Murfi H, Wibowo A. Random space-based fuzzy c-means for topic detection on Indonesia online news. In: Lecture notes in computer science (including subseries lecture notes in artificial intelligence and lecture notes in bioinformatics), vol. 11909. LNAI; 2019. https://doi.org/10.1007/978-3-030-33709-4_12.

[30] Song C, Huang Y, Liu F, Wang Z, Wang L. Deep auto-encoder based clustering. Intell Data Anal 2014;18(6S):S65–76.

[31] Song C, Liu F, Huang Y, Wang L, Tan T. Auto-encoder based data clustering. In: Ruiz-Shulcloper J, di Baja G, editors. Progress in pattern recognition, image analysis, computer vision, and applications. Berlin, Heidelberg: Springer Berlin Heidelberg; 2013. p. 117–24.

[32] Guo X, Gao L, Liu X, Yin J. Improved deep embedded clustering with local structure preservation. In: Proceedings of the 26th international joint conference on artificial intelligence. AAAI Press; 2017. p. 1753–9.

[33] Xie J, Girshick R, Farhadi A. Unsupervised deep embedding for clustering analysis. In: Proceedings of the 33rd international conference on international conference on machine learning, vol. 48. JMLR.org; 2016. p. 478–87.

[34] Guan R, Zhang H, Liang Y, Giunchiglia F, Huang L, Feng X. Deep feature-based text clustering and its explanation. IEEE Trans Knowl Data Eng 2020. https://doi.org/10.1109/TKDE.2020.3028943.

[35] Bezdek JC, Hathaway RJ. Convergence of alternating optimization, neural parallel. Sci Comput 2003;11(4):351–68.

[36] Hinton GE, Salakhutdinov RR. Reducing the dimensionality of data with neural networks. Science 2006;313(5786):504–7. https://doi.org/10.1126/science.1127647.

[37] Vincent P, Larochelle H, Lajoie I, Bengio Y, Manzagol P-A. Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion. J Mach Learn Res 2010;11:3371–408.

[38] O'Callaghan D, Greene D, Carthy J, Cunningham P. An analysis of the coherence of descriptors in topic modeling. Expert Syst Appl 2015;42(13):5645–57. https://doi.org/10.1016/j.eswa.2015.02.055.

[39] van der Maaten L. Learning a parametric embedding by preserving local structure. In: van Dyk D, Welling M, editors. Proceedings of the twelfth international conference on artificial intelligence and statistics, vol. 5. Clearwater Beach, Florida USA: Hilton Clearwater Beach Resort; 2009. p. 384–91. PMLR, http://proceedings.mlr.press/v5/maaten09a.html. Retrieved from.

[40] Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Duchesnay E. Scikit-learn: machine learning in Python. J Mach Learn Res 2011;12:2825–30.