Expert Systems With Applications 137 (2019) 157–166

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

ADC: Advanced document clustering using contextualized representations

Jinuk Park, Chanhee Park, Jeongwoo Kim, Minsoo Cho, Sanghyun Park*

Department of Computer Science, Yonsei University, 50 Yonsei-ro, Seodaemun-Gu, Seoul 03722, Republic of Korea

ARTICLE INFO

Article history:
Received 5 March 2019
Revised 28 June 2019
Accepted 28 June 2019
Available online 29 June 2019

Keywords:
Natural language processing
Document clustering
Contextualized representations
Cosine similarity
Deep clustering

ABSTRACT

Document representation is central to modern natural language processing systems including document clustering. Empirical experiments in recent studies provide strong evidence that unsupervised language models can learn context-aware representations in the given documents and advance several NLP benchmark results. However, existing clustering approaches focus on the dimensionality reduction and do not exploit these informative representations. In this paper, we propose a conceptually simple but experimentally effective clustering framework called Advanced Document Clustering (ADC). In contrast to previous clustering methods, ADC is designed to leverage syntactically and semantically meaningful features through feature-extraction and clustering modules in the framework. We first extract features from pretrained language models and initialize cluster centroids to spread out uniformly. In the clustering module of ADC, the semantic similarity can be measured using the cosine similarity and centroids update while assigning centroids to a mini-batch input. Also, we utilize cross entropy loss partially, as the self-training scheme can be biased when parameters in the model are inaccurate. As a result, ADC can take advantages of contextualized representations while mitigating the limitations introduced by high-dimensional vectors. In numerous experiments with four datasets, the proposed ADC outperforms other existing approaches. In particular, experiments on categorizing news corpus with fake news demonstrated the effectiveness of our method for contextualized representations.

1. Introduction

Unsupervised clustering is a fundamental research topic that groups similar data patterns in images, documents, or any objects into one cluster based on their similarities. Therefore, the type of similarity used—also called the distance measure—is the most important factor in the clustering algorithm and depends on the feature representation spaces. The *k*-means algorithm, a traditional clustering algorithm, defines it as the Euclidean distance in the feature space, which generally has a low dimensionality for the sake of computational efficiency and the reliability of the measure. In the case of a high-dimensional feature space, dimension reduction techniques such as principal component analysis (PCA) or hashing (Song, Gao, Liu, Zhu & Sebe, 2018) can be used, but the preserved power of representations after reducing based on these shallow models is limited (Guo, Gao, Liu & Yin, 2017; Hsu & Lin, 2018).

In recent works, clustering methods comprising the use of deep neural networks (DNNs), referred to as *deep clustering*, have been widely studied because neural networks can generate low-dimensional latent features (Huang, Huang, Wang & Wang, 2014; Xie, Girshick & Farhadi, 2016; Yang, Fu, Sidiropoulos & Hong, 2017). These proposed methods incorporate autoencoder architectures that use TF-IDF vectors into clustering algorithms. They first represent documents as TF-IDF vectors, and then achieve non-linear mapping to low-dimensional latent feature spaces via autoencoders. However, when compared to domain-specific neural networks including state-of-the-art models for natural language processing (NLP), TF-IDF vectors suffer from relatively ineffective word representations. As tokens are not considered in a sequential manner in TF-IDF, the dependencies and relationships between tokens cannot be reflected in TF-IDF vectors, which results in less informative representations.

In NLP, distributed representations of words—also known as word embeddings—have facilitated significant improvements in several NLP benchmarks including part-of-speech tagging (Collobert et al., 2011), sentiment analysis (Socher et al., 2013), and question answering (Rajpurkar, Zhang, Lopyrev & Liang, 2016). For instance, feature-based word vectors extracted using neural methods such as Word2Vec (Mikolov, Chen, Corrado & Dean, 2013) and GloVe (Pennington, Socher & Manning, 2014) have become a stan-

* Corresponding author.
E-mail addresses: parkju536@yonsei.ac.kr (J. Park), channy_12@yonsei.ac.kr (C. Park), jwkim2013@yonsei.ac.kr (J. Kim), minsoo0104@yonsei.ac.kr (M. Cho), sanghyun@yonsei.ac.kr (S. Park).

https://doi.org/10.1016/j.eswa.2019.06.068
0957-4174/

dard part of NLP models as the initialization for word tokens. Generally, these neural networks encode word tokens into latent feature representations to model the semantical relationship between each token in the given sequence using a co-occurrence frequency. As a result, these models are more efficient in capturing semantically meaningful and rich embeddings than TF-IDF vectors.

A recent trend in modeling distributed representations for embeddings is to pre-train unsupervised language models (LMs) in a large corpus, and these LM-based features have obtained state-of-the-art results in many NLP tasks (Howard & Ruder, 2018; Peters et al., 2018; Radford, Narasimhan, Salimans & Sutskever, 2018). One of the most successful LM pre-trained models is BERT (Devlin, Chang, Lee & Toutanova, 2018), which is designed to learn deep bidirectional representations of documents. As the role of language modeling is to understand natural language and model linguistics, LM-based pre-trained models can produce rich and contextualized language representations that can enhance traditional word embeddings.

As the performance of language understanding systems—which includes document clustering—relies on the quality of feature representations, integrating contextual word embeddings naturally facilitates NLP systems and boosts their performances. However, the major limitation is the dimensionality of contextualized embeddings. To represent the diverse meanings of the same words in different situations, it is necessary to use high dimensional features for embeddings (768 dimensions for one word token in BERT); however, very-high-dimensional features can cause notorious issues in the computation of the similarity measure and unstable training in deep clustering networks (DCNs).

To address these limitations, we present a novel deep clustering framework for document clustering called Advanced Document Clustering (ADC). The main idea of the proposed method is to leverage the advantages of context-aware representations while employing efficient similarity between documents and minimizing the difficulties of networks training. In practice, performing conventional clustering with state-of-the-art representations can be troublesome due to improper distance measurements of clustering methods and the high dimensionality of vectors. We tackle this problem by proposing a novel clustering network. Our clustering module utilizes LM-based contextualized features with cosine similarity. In this way, our model performs clustering by measuring the semantic similarity between documents since cosine similarity computes the direction of document vectors and not the magnitude. Additionally, our model updates parameters based on mini-batch training, which helps avoid the tremendous computations needed to simultaneously calculate similarities. Furthermore, the proposed method exploits the top-k_c centroids update rule as well as partial cross entropy loss to prevent an extreme distortion in each update and to stabilize the training process in the case of large-scale documents.

The main contributions of our study are as follows:

- We propose the first deep clustering framework that is specialized in document clustering and fully leverages state-of-the-art contextualized document vectors. Furthermore, we tackle the high-dimensional features issue using cosine-similarity-based clustering and the mini-batch centroids update rule.
- We show that language understanding is an integral part of document clustering in various experiments. The proposed framework also outperforms existing clustering methods.
- The proposed framework can be easily combined with any other pre-trained language representation models. Furthermore, we conduct experiments using several types of representations and highlight our findings.

The rest of paper is organized as follows. Section 2 reviews several related works in document clustering and language representations. Section 3 describes the proposed ADC framework including feature extraction module, optimization and centroids update rule in clustering module. Section 4 presents datasets and evaluation metrics used for experiments. In Section 5, experimental results and discussions are demonstrated. Finally, Section 6 provides the conclusion and future works.

2. Related works

Document clustering, which is a typical problem in the use of unsupervised learning techniques without prior information such as labels, is the process of categorizing documents based on the similarity measure between the existing text in the documents. The automatic clustering algorithm for various domains such as image or text has been extensively studied. Traditional algorithms for clustering include *k*-means (MacQueen, 1967), Gaussian mixture models (GMMs) (Bishop, 2006), and spectral clustering (Von Luxburg, 2007). In these methods, data points are grouped according to the inherent characteristics, which are generally applicable to different types of problems. However, when the dimensionality of an input feature is very high, traditional clustering methods tend to have a poor performance because of the inefficient strategies for computing distance measures (Steinbach, Ertöz & Kumar, 2004). Therefore, dimensionality reduction and feature reconstructing approaches—such as PCA—and non-linear transform—such as kernel—methods (Hofmann, Schölkopf & Smola, 2008) have been proposed for transforming inputs into a low-dimensional feature space. Succeeding studies of the *k*-means algorithm have proposed that this issue of the high-dimensionality of the input feature be addressed by jointly reducing the dimensionality and performing clustering (De la Torre & Kanade, 2006), but they are still restricted to linear embedding.

In recent years, with the development of deep learning, deep-learning-based approaches have become extremely influential in a variety of fields, such as image processing, NLP, and information retrieval. Even in the clustering problem, frameworks that utilize DNNs to perform deep clustering have been used to map the input into the low-dimensional feature spaces, which performs non-linear transformation. In most studies, they use TF-IDF methods to represent documents and apply DNNs to transform the vectors into the latent feature spaces. Deep clustering methods can be categorized into three major categories from the viewpoint of architecture: (i) autoencoder-based architecture for training a non-linear mapping function, (ii) DNN-based architecture trained with the clustering loss and without the reconstruction loss of the autoencoder, (iii) other architectures based on a generative adversarial network (GAN) and variational autoencoder (VAE).

The first category of methods comprises the use of an autoencoder, which comprises an encoder and decoder for generating an output similar to the input, which results in unsupervised representation learning. For example, DCN Yang et al. (2017) uses pre-trained autoencoders for dimensionality reduction and then jointly performs clustering via the *k*-means algorithm. Similarly, Tian, Gao, Cui, Chen and Liu (2014) proposed the graph clustering method, wherein a stacked autoencoder was adopted, for embedding of the graph in addition to the *k*-means clustering algorithm. Deep continuous clustering (DCC) (Shah & Koltun, 2018) also optimizes the feature embedding with an autoencoder, but there is a difference, in that, the DCC does not depend on the prior knowledge of the clusters number by a global continuous objective.

The second category of methods comprises the use of only the clustering loss for training, such as the *k*-means loss or locality-preserving loss. Deep embedding clustering (DEC) (Xie et al., 2016), one of the most representative methods in this field, performs

deep clustering using fine-tuning of the network based on an autoencoder. They first calculate TF-IDF document vectors and then acquire latent features via an autoencoder. Moreover, it iteratively refines the clusters by optimizing the KL-divergence. However, as there is a possibility that the feature space will become distorted in the process of the fine-tuning, Guo et al. (2017) kept the decoder part intact for the preservation of the local data structure and added the clustering loss to the feature space. In the case of large-scale image datasets, Hsu and Lin (2018) presented a CNN-based clustering method that incorporates the mini-batch *k*-means algorithm for addressing the issue of computation and memory complexity. It first randomly picks *k* samples and then initializes the cluster centroids with the extracted features based on AlexNet (Krizhevsky, Sutskever & Hinton, 2012), while updating sample assignments and cluster centroids via the mini-batch *k*-means algorithm.

The third category of methods comprised other architectures based on GAN or VAE, which are generative models based approaches. Since categorizing unlabeled data is associated with clustering, clustering can be considered as a special case of using GAN-based models. For example, Chen et al. (2016) and Springenberg (2015) evaluated GAN-based models in the aspect of clustering, and Jiang, Zheng, Tan, Tang and Zhou (2017) presented a model based on VAE.

While incorporating preserved information into clustering algorithms after reducing is the main focus of those deep clustering methods, recent studies in NLP show that much informative representations can be compressed by neural methods. Recently, feature-based word embeddings, i.e., the vectorization of words into multi-dimensional spaces by utilizing the co-occurrences of the words in contexts, have become a prevalent approach for language representations through non-neural (Blitzer, McDonald & Pereira, 2006) or neural (Mikolov et al., 2013) methods. These pre-trained word embeddings have been regarded as an essential element of language understanding models, and there have been several recent attempts to train word embeddings with a coupled language model (e.g., ELMo) (Peters et al., 2018) for context-sensitive representations.

Furthermore, the fine-tuning-based approach—wherein parameters of the pre-trained models are adjusted precisely for a task-specific architecture—is an active research area of language representations and exhibits a surprisingly powerful performance (Radford et al., 2018). Also, experiments using BERT (Devlin et al., 2018) have demonstrated that pre-trained representations with bidirectional language models are remarkably effective and intelligent language representation methods and outperform a variety of NLP tasks. In particularly, it adopts a masked language model, which predicts randomly masked tokens based on its context as well as a task for predicting the next sentence in order to enable pre-trained bidirectional representations, thus overcoming the limitation of unidirectional language models. According to the powerful representative ability of BERT, the projected vectors have relatively large dimensionalities compared to the dimensions of autoencoders in deep clustering studies. For instance, DEC model represents a document with a 2000-dimensional vector using TF-IDF and employs autoencoders to transform it into a 10-dimensional latent vector for a document. In BERT, there is a 768-dimensional vector for a one word-level token. Our method differs from previous studies in that our clustering method directly leverages those high dimensional vectors to minimize to the alteration of the rich embeddings.

3. Proposed method

In this section, we introduce the architecture and optimization for the proposed document clustering framework. The overview of the proposed framework is shown in Fig. 1. The framework can be divided into two modules: feature extraction module and iterative clustering module. We first cover the model architecture and construction of the document representations for clustering from unsupervised learning. We then describe the centroids update rules in the clustering phase in addition to the initialization of the centroids. Finally, we analyze the proposed method in terms of computational complexity.

3.1. Feature extraction from unsupervised learning

The previous deep clustering works are mainly focused on dimension reduction and the use of autoencoders to transform the input data into a smaller feature space (Guo et al., 2017; Xie et al., 2016). In their methods, documents are represented with TF-IDF vectors which consider the relevance of terms, and then the transformation is performed using autoencoders. TF-IDF vectors have relatively insufficient information when compared to other embedding approaches in NLP. For instance, the relationships between each token in a sequence cannot be considered in TF-IDF method since it counts the frequencies of terms and documents but does not consider the order of tokens. Moreover, owing to the limited capacity of autoencoders, information can be partially encoded and some features of the input can remain uncaptured (Goodfellow, Bengio & Courville, 2016). Therefore, we focus on fully leveraging informative language representations from pre-trained models.

Inspired by the successful transfer of pre-trained CNNs (He, Zhang, Ren & Sun, 2016; Krizhevsky et al., 2012; Simonyan & Zisserman, 2014) into image clustering (Caron, Bojanowski, Joulin & Douze, 2018; Hsu & Lin, 2018), we focused on using the advantages of pre-trained embeddings or models in other NLP tasks, especially language models. Similar to the role of pre-trained CNNs, which is to extract important features for clustering, the pre-trained embeddings or models can retain salient features from the given sentence. As language models can acquire different semantic meanings of words depending on the context, we use the contextualized representations of words for document clustering. We conduct experiments using both the embedding and language modeling approaches for comparison.

Consider a document set $\mathbb{D} = \{D_1, D_2, \ldots, D_n\}$ and each document $D_i = \{w_{i,t}\}_{t=1}^{m_i}$, where $w_{i,t}$ denotes token t in document i and n denotes the total number of documents. The goal of document clustering is to assign $\mathbb{D}$ into k cluster centroids, $\{\mu_1,\mu_2,\ldots,\mu_k\}$. The proposed framework first converts word tokens into a token-level distributed representation $e_{i,t}$ using a pre-trained model g (pre-trained embeddings in embedding approach), $e_{i,t} = g(w_{i,t})$. A contextualized document representation d_i can then be generated using any encoder strategies with those token-level representations.

$$d_i = \text{Encoder}\left(\{e_{i,t}\}_{t=1}^{m_i}\right) \tag{1}$$

The recurrent neural networks (RNN) approach can be applied by taking the last hidden states of the tokens. The most common approach is to average all the token-level representations. We take the latter method to achieve simplicity in the framework as each token-level representation has sufficient high-dimensional vectors (e.g., 768 dimensions in BERT embeddings) for obtaining the complex context of the document while averaging representations. In this manner, we achieve a contextualized document representation d_i, $i = 1, \ldots, n$.

3.2. Optimization and centroids update rule

To effectively initialize k cluster centroids $\{\mu_j\}_{j=1}^{k}$, we follow the seed strategy proposed by Arthur and Vassilvitskii (2007). The

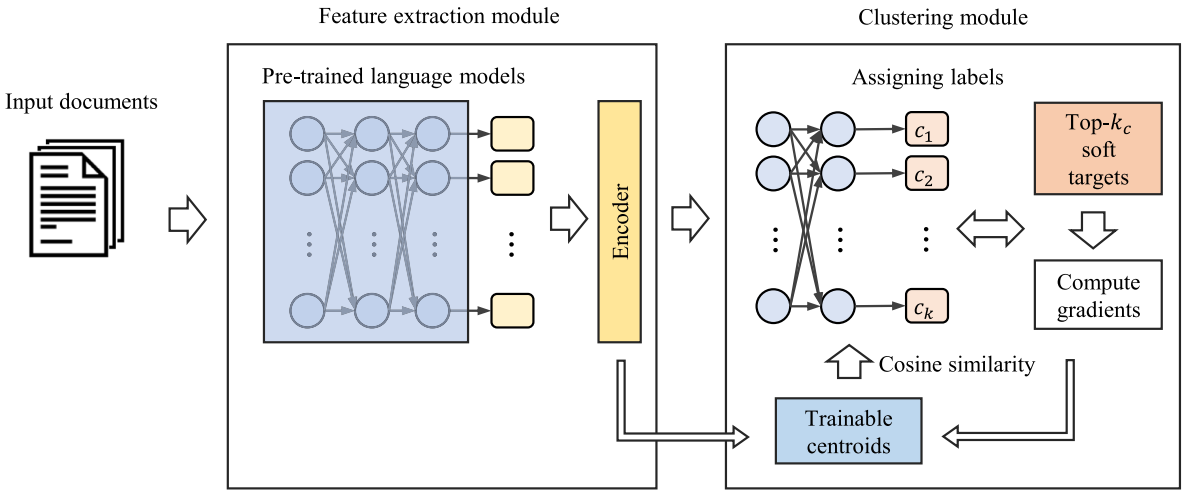

Fig. 1. Block diagram of the proposed ADC framework, which has two modules: (1) feature extraction module, and (2) clustering module. In the feature extraction module, pre-trained language models can be easily changed to other language representation models. We use BERT as the language models and average token-level representations in the encoder.

main idea of the initialization algorithm is to choose the farthest point from the nearest one among the previously selected cluster centroids. We use a cosine similarity in the initialization algorithm as it measures the differences in the direction of documents and not the magnitude, and hence has the advantages of semantic similarity.

Optimization in the parameters of the model and updating centroids is performed using the stochastic gradient descent (SGD) process. As targets are not defined in the unsupervised clustering, the previous works use KL-divergence loss calculated with the similarity distribution and an auxiliary distribution based on the similarity. This training method is a form of self-training approach. However, we focus on that a self-training approach can be severely biased when the targets are unreliable.

To resolve this issue, we propose an optimization process using cross entropy loss as an objective function with top-k_c soft targets in a mini-batch size B. We denote the clustering model by f_θ, where θ is a set of trainable variables including centroids. In order to define the soft targets corresponding to the input $\{d_i\}_{i=1}^{B}$, we compute the cosine similarity s_{ij} between the input and the given centroids and take softmax function, which can be formulated as follows:

$$s_{ij} = \frac{d_i \cdot \mu_j}{\| d_i \| \| \mu_j \|}, \tag{2}$$

$$p_i = \text{Softmax}(s_i) \tag{3}$$

where $\cdot$ denotes the dot product and $\|\cdot\|$ denotes the magnitudes of the vectors; $\text{Softmax}(x_i) = \frac{\exp(x_i)}{\sum_i \exp(x_i)}$. The cosine similarity score has range in $(-1, 1)$. It has 1, 0, and -1, respectively, when the two vectors are identical, independent, and have perfectly opposite meanings. As we have mature representations from pre-trained models in the first phase, p_{ij} can be interpreted as the probability of assigning document d_i to the centroid μ_j. The soft targets $\tilde{y}_i$ can then be derived by taking the centroid that has the highest similarity score in a self-training manner:

$$\tilde{y}_i = \arg\max_j \left(p_{ij}\right) \tag{4}$$

Our objective function is defined as follows:

$$\min_\theta \frac{1}{N} \sum_{i=1}^{N} NLL(p_i,\ \tilde{y}_i), \tag{5}$$

where NLL denotes the negative log-likelihood function. Training is performed by minimizing the loss function using SGD.

Since the above self-training based on soft targets can be severely contaminated when the supervision from soft targets is inaccurate, we adopt the top-k_c-based parameter update from Hsu and Lin (2018). We only reflect the computed loss from the top-k_c samples that have the highest similarity score with respect to the corresponding centroids in the mini-batch B. Note that, when k_c is much smaller than B, the model updates the parameters very slowly. Nonetheless, we set k_c such that it is the same as the number of clusters in the experiments, as dramatic changes in the centroids may interrupt the finding of optimum centroids for clustering. The proposed method is summarized in Algorithm 1.

Algorithm 1 Proposed clustering algorithm with high-dimensional vectors.

Input:
Number of clusters: k
Maximum iterations: *MaxIter*
Feature extraction model: g
Clustering model: f_θ
Mini-batch size: B
Parameter for computing loss: $k_c (k_c < B)$
Let $\boldsymbol{\mu} = \{\mu_j\}_{j=1}^{k}$ be the initial centroids.

for $iter = 1$ **to** *MaxIter* **do**:
  $\{D_i\}_{i=1}^{B} \leftarrow B$ documents randomly picked from $\mathbb{D}$
  $\boldsymbol{d} = \{\}$
  for $i = 1$ **to** B **do**:
    $\{w_{i,t}\}_{t=1}^{m_i} \leftarrow \text{tokenize}(D_i)$
    $\{e_{i,t}\}_{t=1}^{m_i} \leftarrow g(\{w_{i,t}\}_{t=1}^{m_i})$
    $d_i = 1/m_i \sum_t e_{i,t}$
    // extract and encode contextualized representations
    $\boldsymbol{d} \cup d_i$
  end for
  $\boldsymbol{s},\ \boldsymbol{p} = f_\theta(\boldsymbol{d}, \boldsymbol{\mu})$
  // compute cosine similarity $\boldsymbol{s}$, probability $\boldsymbol{p}$
  $\boldsymbol{p}', \tilde{\boldsymbol{y}}' \leftarrow 1 - \text{NN}(s, k_c)$
  // assign top-k_c highest samples to soft targests
  update parameters in f_θ using $\text{NLL}(\boldsymbol{p}', \tilde{\boldsymbol{y}}')$
end for

3.3. Analysis

We present a theoretical analysis for our method to compare our scheme with conventional techniques. Since our method has

Table 1
Dataset statistics for experiments. For SQuAD 1.1 dataset, we construct samples from the corpus using the 10 largest topics. We also use documents that have more than 500 tokens in the Yahoo Answers dataset.

Dataset	Number of examples	Number of classes
SQuAD 1.1	1094 (10 largest topics)	10
Yahoo answers	38,368 (over 500 tokens)	10
REUTERS (full)	685,071	4
REUTERS (10 K)	10,000	4
FakeNewAMT	480	12

two modules, feature extraction and clustering module, we analyze the complexity of each module separately.

First, we discuss the feature extraction module which can be incorporated with various pre-trained models. The computational complexity of feature extraction thus varies depending on the models used with the module. Considering sequential operations for each token, the pre-trained embeddings (e.g., GloVe) have $O(L)$ and the RNN-based language models have $O(LH^2)$, where L denotes the maximum sequence length of tokens in documents and H denotes the maximum number of neurons in hidden layers (Graves, 2012; Hochreiter & Schmidhuber, 1997). In the case of BERT or Transformer (Vaswani et al., 2017), the complexity is $O(L^2H)$ due to the attention mechanism in Transformer called self-attention. BERT is more efficient than RNN-based language models in terms of computational complexity as the sequence length L is typically much smaller than the hidden size H.

In the clustering module, we compute the similarity between centroids and documents and select the most similar top-k_c samples for partial loss. If the document representations have D dimensional vectors from the feature extraction, the complexity of the clustering module is $O(nDk)$, where n denotes the total number of documents and k denotes the number of centroids. Therefore, if we use BERT for feature extraction, the computational complexity of the proposed framework is $O(nL^2H + nDk)$. Additionally, $k \leq L$ and $D \leq H$ holds in the general case of BERT, the complexity is $O(nL^2H)$.

Since neural models in the feature extraction module can have high computational complexity, conventional algorithms that do not require neural models often have relatively low complexity. For instance, k-means algorithm has $O(n^{Dk+1})$, which can be reduced to $O(nDki)$, given D dimensional document representations and the maximum iteration i (Manning, Raghavan & Schutze, 2008). On the other hand, DEC has a complexity of $O(nH^2)$, which is similar to ours as it uses neural models (autoencoders) to encode documents with TF-IDF vectors. Generally, $H \leq L^2$ holds in BERT, thus DEC is more efficient than ours with respect to computational complexity.

4. Experiments

4.1. Datasets

We evaluate our proposed method on four document datasets and compare it to other algorithms. In order to obtain large-scale corpora without human-annotation, we use a variety of datasets such as question answering, document classification and categorization. We evaluate the performance using these datasets as a task of clustering into n classes, respectively. A summary of the dataset statistics is shown in Table 1.

- **SQuAD 1.1**: The Stanford question answer dataset (SQuAD) is a dataset for reading comprehension that comprises questions and corresponding answers for reading passages of Wikipedia articles (Rajpurkar et al., 2016). We chose only the 10 largest articles from among 536 articles and collected 1094 passages for the experiments.
- **Yahoo Answers**: We obtained the Yahoo Answers dataset (Zhang, Zhao & LeCun, 2015), which comprised questions, question titles and best answers that are categorized into main classes: Society & Culture, Politics & Government, Science & Mathematics, Business & Finance, Health, Education & Reference, Computers & Internet, Sports, Entertainment & Music, and Family & Relationships. We selected samples of over 500 tokens for only using documents of length similar to real world data.
- **REUTERS**: Reuters dataset consists of approximately 810,000 English news stories labeled based on a category tree (Lewis, Yang, Rose & Li, 2004). Following the precedent research (Xie et al., 2016), we used four root categories: corporate/industrial, government/social, markets, and economics as labels and excluded all documents labeled with multiple root categories, which results in the REUTERS (full) dataset of 685,071 articles. As the computational resources, a subset comprising random 10,000 examples from the full dataset is used for REUTERS 10 K.
- **FakeNewsAMT**: FakeNewsAMT dataset (Pérez-Rosas, Kleinberg, Lefevre & Mihalcea, 2018) comprises legitimate news articles belonging to six different domains (sports, business, entertainment, politics, technology, and education) from a variety of mainstream news websites. Also, it contains fake news collected through crowdsourcing via Amazon Mechanical Turk (AMT) for each corresponding domain. The number of documents is doubled in the case for taking fake news (240 documents for legitimate news and 240 for fake news). In the dataset, labels include the six domains for legitimate news and six additional categories for fake news. In Section 5.3, we variated the number of categories for further investigation.

4.2. Evaluation metrics

We adopt two standard unsupervised evaluation metrics that are widely used in deep clustering studies to compare our proposed method to other algorithms. For all the algorithms, the number of clusters are set as the number of ground-truth categories of each dataset, and we evaluate the clustering performance using the unsupervised clustering accuracy (ACC):

$$\mathrm{ACC} = \max_m \frac{\sum_{i=1}^{n} 1\{y_i = m(c_i)\}}{n} \tag{6}$$

where y_i is the ground-truth label, c_i is the cluster assignment that is created by the clustering algorithm, and m is a mapping function between the clustering assignments and labels, ranging over all possible one-to-one mappings. This metric finds the optimal matching between the ground-truth label and the cluster assignment from an unsupervised clustering algorithm. The most effective mapping function can be computed by Hungarian algorithm in a linear assignment problem (Xu, Liu & Gong, 2003).

Another metric is the normalized mutual information (NMI) (Arthur & Vassilvitskii, 2007):

$$\mathrm{NMI}(Y, C) = \frac{I(Y, C)}{\frac{1}{2}[H(Y) + H(C)]} \tag{7}$$

where Y denotes the ground-truth labels, C denotes the cluster labels, I denotes the mutual information of two discrete random variables, and H denotes the entropy, which is considered as a measure of uncertainty about a random variable.

5. Results and discussion

We evaluate the proposed ADC on several documents corpus and compare the proposed method against other unsupervised

Table 2
Samples from each dataset. Note that samples from the REUTERS are composed of a list of tokens without the sequential property of natural language.

Datasets	Categories	Samples
SQuAD 1.1	American_Idol	American Idol was nominated for the Emmy's Outstanding Reality Competition Program for nine years but never won. Director Bruce Gower won a Primetime Emmy Award for ...
Yahoo Answers	Sports	Formula1 car, What makes a Formula1 car a fast car? Let me break it down Engine- 3 liter V10. Approximately 900+ hp, 19,000 rpm. The engine weighs about 200 lbs...
RETUERS	government/social	socc colomb beat chil world cup qualif ...
	corporate/industrial	open cent sunday rule survey ...
FakeNewsAMT	Business (legitimate)	Brexit talks will fail without compromise: José Manuel BarrosoBrexit negotiations are on course to fail unless both Britain and the European Union ditch their winner-takes-all approach to the...
	Business (fake)	Toshiba's Westinghouse creating thriving job market for US citizens With Trump promising to bring jobs back to the US, Toshiba's Westinghouse is showing ...

Table 3
Comparison of the proposed framework and the baseline methods in the case of topic clustering datasets. Since we report published results for DEC on REUTERS (both full and 10 K), NMI scores are not available in the original paper, hence computed additionally.

Models		SQuAD 1.1	Yahoo answers	REUTERS (full)	REUTERS 10 K
k-means	ACC	0.8418	0.3660	0.5329	0.5242
	NMI	0.8568	0.2822	0.4014	0.3149
GMM	ACC	0.7322	0.4507	0.6013	0.5342
	NMI	0.7915	0.3427	0.3447	0.3170
BIRCH	ACC	0.7267	0.4678	0.5574	0.5171
	NMI	0.6993	0.3233	0.3781	0.3639
DEC	ACC	0.8082	0.5352	**0.7563**	**0.7217**
	NMI	0.8625	0.6009	**0.4812**	**0.4258**
Proposed	ACC	**0.8511**	**0.6442**	0.6711	0.6440
	NMI	**0.8836**	**0.6837**	0.3969	0.3813

Table 4
Comparison of clustering performance using distributed representations from BERT.

Models (with BERT vectors)		SQuAD 1.1	Yahoo answers	REUTERS 10K
k-mean	ACC	0.6856	0.3733	0.3751
	NMI	0.6478	0.2384	0.0493
GMM	ACC	0.7011	0.3938	0.3746
	NMI	0.6903	0.2665	0.0487
BIRCH	ACC	0.6563	0.3635	0.3988
	NMI	0.6683	0.2783	0.0628
DEC	ACC	0.7240	0.5706	0.5753
	NMI	0.7835	0.6342	**0.3895**
Proposed	ACC	**0.8458**	**0.6444**	**0.6416**
	NMI	**0.8856**	**0.6844**	0.3758

clustering methods. Furthermore, we explore the effect of proposed clustering module and semantic understanding. Also, we compare distance measures in the initialization method and other embedding approaches in the feature extraction module.

5.1. Main results

To investigate the effectiveness of language understanding using contextualized document representations, we conduct experiments on both the document corpus with natural language and the corpus symbolized with term frequencies. The document corpus written in natural language includes SQuAD 1.1, Yahoo Answers, and FakeNewsAMT datasets. In these datasets, a document consists of consecutive sentences, and a sentence is composed of natural language tokens, which can be fully human recognizable. However, the REUTERS dataset is symbolized using keyword token terms and represented with a simple list of tokens. Apparently, there is no sequential characteristic in the dataset. Table 2 shows samples from each dataset.

We compare the proposed document clustering framework with three commonly used clustering approaches and one autoencoder-based clustering method. Our baselines are the GMM, *k*-means++, BIRCH (Zhang, Ramakrishnan & Livny, 1996), and DEC (Xie et al., 2016). Also, we use BERT as the feature extraction module in the proposed framework. We set the number of clusters as the same as the number of labels in each dataset and use TF-IDF features for the baseline methods. Further comparison for types of input features will be discussed in the following section.

Table 3 reports the results of document clustering on the above-mentioned datasets. The ADC significantly outperforms the baseline methods in YAHOO answers and SQuAD 1.1 datasets based on the accuracy measure. This indicates that contextualized document representations can be helpful for document understanding, and the proposed clustering algorithm is optimized for utilizing those high-dimensional distributed features. Surprisingly, the simple k-means algorithm with TF-IDF features achieves comparable accuracy to our method and DEC on SQuAD 1.1 dataset. As we used the 10 largest topics, each class includes approximately 100 documents with specific words which distinguish the corresponding topic easily. This results in the second highest performance for *k*-means algorithm. In the case of the REUTERS datasets, we observed that DEC shows the highest performance among all the clustering methods. It should be noted that the REUTERS dataset is composed of a list of repeated tokens without the sequential property of natural language. Therefore, we can conclude that the proposed ADC is more effective in the case of documents in the natural language form rather than in the case of documents with listed tokens, which is not natural in the real world.

5.2. Impact of clustering module

We conduct an additional experiment to demonstrate the effectiveness of optimization and centroids updates in our proposed clustering module. To investigate the impact of the clustering module, we compare the baselines against our system with the same input features. As our method needs distributed representations for documents, we incorporate dense vectors from BERT into both the baselines and ours.

Table 4 presents the results for the clustering algorithms using distributed vectors achieved by BERT. We expected improved performances in the baselines, as we fed highly informative vectors than TF-IDF, except for the REUTERS dataset. In the case of the REUTERS dataset, as it contains only the simple lists of tokens, it is acceptable to result in lower performances. However, the existing algorithms show significantly lower accuracies for all datasets comparing to the results in Table 3, in which they used TF-IDF input features. The comparison of clustering performance between vectors from BERT and TF-IDF can show different aspects depending on the characteristics of corpus.

Nevertheless, it is obvious that baseline methods do not take full advantages from contextualized representations as our method outperforms the existing algorithms with a large margin. Since we compare algorithms with the same input features, we conclude that the proposed clustering method intensely contributes

Table 5
Comparison of the proposed framework and the baseline methods (TF-IDF) on FakeNewsAMT dataset.

Models		FakeNewsAMT		
		w/o fake news	w/ fake news	w/ fake news (sep.)
k-means	ACC	0.6083	0.4188	0.3375
	NMI	0.4055	0.2472	0.4017
GMM	ACC	0.5417	0.3042	0.3063
	NMI	0.3650	0.1931	0.3326
BIRCH	ACC	0.4708	0.3708	0.2687
	NMI	0.3288	0.1786	0.2942
DEC	ACC	0.5521	0.4160	0.3438
	NMI	0.4771	0.3725	0.4473
Proposed	ACC	**0.8708**	**0.6437**	**0.7583**
	NMI	**0.8687**	**0.6480**	**0.8623**

Table 6
Comparison of clustering performance on FakeNewsAMT dataset using distributed representations from BERT.

Models (with BERT vectors)		FakeNewsAMT		
		w/o fake news	w/ fake news	w/ fake news (sep.)
k-means	ACC	0.6042	0.4021	0.3729
	NMI	0.4674	0.2713	0.4060
GMM	ACC	0.5583	0.4125	0.3667
	NMI	0.4713	0.2965	0.3774
BIRCH	ACC	0.5792	0.4042	0.3625
	NMI	0.4781	0.2547	0.4149
DEC	ACC	0.6319	0.4883	0.5703
	NMI	0.5911	0.4492	0.7113
Proposed	ACC	**0.8708**	**0.6437**	**0.7583**
	NMI	**0.8687**	**0.6480**	**0.8623**

to performance improvement. We utilize the cosine similarity with partial updates in centroids whereas baseline methods take other metrics (e.g., Euclidean distance or divergence in distribution). To measuring semantic similarity between those contextualized representations, the cosine similarity plays an important role in our novel clustering method. Using partial updates to keep centroids from being distorted by inaccurate loss in self-supervised training also maximizes the performance.

5.3. Impact of semantic understanding

We conduct further experiments to study the effect of contextualized representations. We compare the ADC to baseline methods on FakeNewAMT corpus, which needs in-depth language understanding to perform clustering (see Section 4.1 for details). Similar to Section 5.1, we use TF-IDF vectors for baseline methods and BERT as feature extraction module in ours. Also, we test those clustering algorithms utilizing the same input features from BERT in order to support our results.

In the FakeNewsAMT dataset, we compare the clustering performances for the absence or presence of fake news. In the first case, we set six topics as labels when we exclude fake news (w/o fake news). Similarly, we use seven topics with one auxiliary label which indicates a fake news category for the second case which is comprising fake news (w/ fake news). In the original dataset, the fake news has six domains corresponding to the legitimate domains (e.g., business, education, and politics). To consider these as separate categories, we set six labels for legitimate news and six additional labels for fake news, with a total of twelve labels. We refer this third setting to "w/ fake news (sep.)" in Table 5. Also, the number of documents is 240 and 480, respectively for the case of excluding fake news and including them.

Table 5 presents the performance of baselines and the proposed method on above-mentioned settings. As shown in the table, our document clustering framework outperforms the baselines consistently across all settings. In the first setting, though there is no fake news, the baselines exhibit poor performance with a large margin comparing to ours. When we included fake news in the corpus, we observed a significant performance drop in all methods. However, the proposed method still shows higher accuracy and NMI score than the baselines. In addition, if we separate the fake news label into six domains in the third case, there is persistent degradation of performance in all the clustering methods except for ours. Surprisingly, we observe that our method demonstrates better performance with twelve separated labels than with one integrated label for fake news. This implies that our model has the ability to measure the similarity between fake and legitimate news as well as detail categories in fake news itself, and hence has enhanced language and semantic understanding in comparison with the baselines.

Since we have used contextualized representations in feature extraction module using BERT for ours, we conduct an additional experiment similar to Section 5.2 in order to prove the ability of proposed method beyond the informative vectors. To allow the same starting line, we apply the rich representations attained using BERT to baseline methods.

Table 6 shows the results of clustering algorithms using semantically rich document representations from BERT. We observe that the contextualized vectors affect the performance improvements in overall settings on FakeNewsAMT dataset. When we cluster the corpus without fake news (w/o fake news), all baselines methods show improved performances in overall compared to the results using TF-IDF vectors. Also, results in other settings appear the similar advances by enriched vectors. However, except for DEC, the baselines show the consistent debasement as we add fake news (w/ fake news) and separate the labels (w/ fake news sep.), which is similar aspect to Table 5. When we divide the auxiliary label for fake news into additional six domains, DEC and our method shows higher performances. This suggests that the contextualized vectors have enough representative power to distinguish domains of fake news itself, as well as the legitimate news and fake news. Nevertheless, DEC short of ability to retrieve the representative strength in the vectors compared to our method.

On top of the experiment for FakeNewsAMT dataset, we questioned the different aspect in comparison of two vectors from TF-IDF and BERT. In SQuAD 1.1 and Yahoo Answers datasets in Table 4, we observed clustering results utilizing TF-IDF outperforms those using BERT vectors. In contrast, we found the degraded or similar results for TF-IDF vectors in FakeNewsAMT dataset. To study this we inspect the corpus and TF-IDF weight scheme. As shown in the Table 7, we observe that some class-related words are frequent in SQuAD 1.1, whereas there is no frequent appearance of words related to the classes in FakeNewsAMT. TF-IDF methods do not matter only the frequencies of words, but the methods do count the inverse document frequencies to measure the amount of information the word has. For instance, 'the' is the most frequent word in all four classes, but second and third ranked for TF-IDF scores in SQuAD 1.1 classes. Also, class-related words are top ranked for TF-IDF weights in SQuAD 1.1 classes. However, the high TF-IDF score ranked words are not well constructed in FakeNewsAMT dataset due to the small size of the corpus and broad subjects in the class. Despite these circumstances, the contextualized representations still offer characteristics to distinguish documents, hence can result in better clustering performances.

5.4. Comparison of initialization

To study the contribution of distance measures in the initialization method, we compare cosine similarity and Euclidean dis-

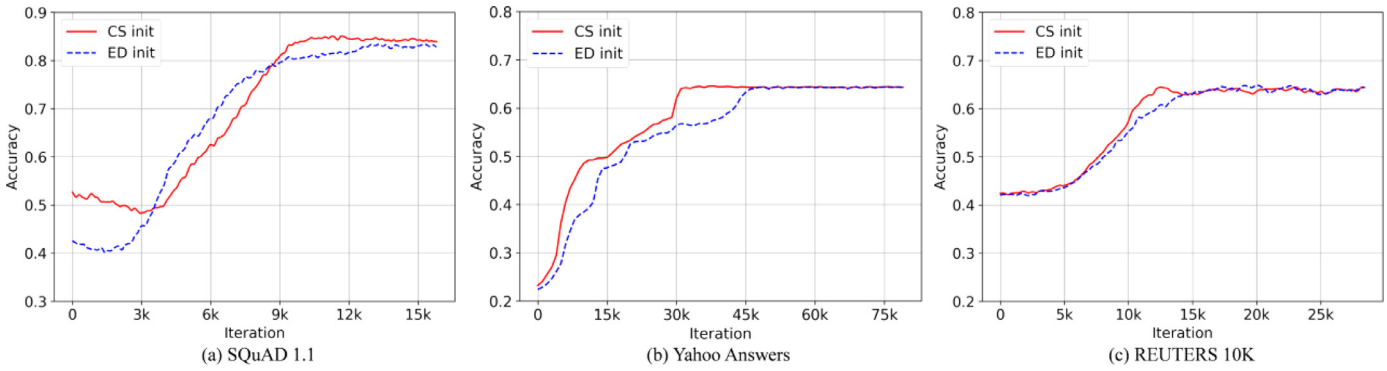

Fig. 2. Accuracies during training process on (a) SQuAD 1.1, (b) Yahoo Answers, and (c) REUTERS 10 K datasets.

Table 7
The Top-10 frequent terms and highest TF-IDF terms for SQuAD and FakeNewsAMT datasets. The frequencies and TF-IDF weights are represented in the bracket, respectively beside frequent terms and high TF-IDF terms.

	SQuAD 1.1				FakeNewsAMT (w/o fake news)			
	Class 1 (American_Idol)		Class 2 (Portugal)		Class 1 (Technology)		Class 2 (Business)	
	Frequent terms	High TF-IDFs	Frequent terms	High TF-IDFs	Frequent terms	High TF-IDFs	Frequent terms	High TF-IDFs
1	the (987)	**season (4.9)**	the (897)	**portugal (4.7)**	the (207)	the (3.3)	the (309)	the (3.5)
2	in (343)	the (3.5)	of (475)	**portuguese (4.3)**	to (131)	to (2.6)	to (169)	to (2.7)
3	of (304)	**show (3.2)**	and (442)	the (3.5)	and (98)	of (2.2)	in (126)	in (2.5)
4	and (290)	**idol (3.1)**	in (325)	and (2.8)	of (96)	and (2.1)	and (114)	and (2.3)
5	to (24)	was (2.2)	to (184)	of (2.7)	in (88)	in (2.1)	of (112)	of (2.3)
6	a (213)	**contestants (2.1)**	a (159)	in (2.4)	a (81)	on (2.1)	a (106)	said (1.8)
7	**season (193)**	on (1.9)	**portuguese (120)**	to (1.8)	on (63)	**google (1.9)**	for (53)	with (1.7)
8	was (167)	**american (1.9)**	**portugal (100)**	de (1.8)	that (42)	its (1.8)	on (50)	on (1.6)
9	as (142)	in (1.8)	as (96)	**Lisbon (1.6)**	for (39)	**new (1.8)**	that (46)	by (1.6)
10	on (139)	and (1.8)	by (93)	is (1.5)	is (39)	will (1.8)	with (43)	its (1.6)

Table 8
Comparison of clustering accuracy using different initialization methods.

Models	SQuAD 1.1	Yahoo answers	REUTERS 10 K
ADC with ED init	0.8370	0.6437	0.6393
ADC with CS init	**0.8458**	**0.6444**	**0.6416**

tance for calculating the initial centroids and present the results in Table 8. We report the result using cosine similarity in initialization as "CS init" and the one using Euclidean distance as "ED init". As shown in the table, we find that our method shows slightly better performances when the centroids are initialized using cosine similarity. These results suggest that CS initialization method can help improve the clustering performance in terms of accuracy.

To further investigate the effect of the two measures, we plot the accuracies during the training process in Fig. 2. As illustrated in the figure, the accuracies show fluctuations for some iterations as we train the clustering module in a self-supervised manner. However, as we utilize partial loss to prevent inaccurate updates, the final accuracies eventually converge in both initialization methods. Furthermore, we observe that ED initialization converges slower than CS initialization for all datasets. This implies that the centroids selected by ED initialization need more manipulations since they do not suit well to the clustering module which uses cosine similarity. Apparently, using CS initialization is more natural as we update centroids and compute loss based on cosine similarity.

5.5. Comparison of embeddings

In this section, we compare several embeddings for the proposed ADC framework. For feature-based pre-trained embeddings, we choose GloVe and fastText for achieving word-level representations. Furthermore, we conduct experiments using BERT and ELMo for language-model-based feature extraction. The pre-trained embeddings and models can be downloaded from the respective official websites. The dimension for token-level representations is 300 for both GloVe and fastText, 1024 for ELMo, and 768 for BERT.

Table 9
Comparison of clustering performance on the proposed framework with different embeddings.

Embeddings		Yahoo answers	SQuAD 1.1	FakeNewsAMT w/ fake news (sep.)
GloVe	ACC	0.5971	0.8162	0.5312
	NMI	0.3949	0.6940	0.6660
fastText	ACC	0.6252	**0.8603**	0.6229
	NMI	0.6648	0.7572	0.7339
BERT	ACC	0.6442	0.8511	**0.7583**
	NMI	0.6837	**0.8488**	**0.8623**
ELMo	ACC	**0.6857**	0.7776	0.6604
	NMI	**0.7234**	0.8080	0.7946

As shown in Table 9, contextualized representations from BERT or ELMo achieves the highest accuracy on two of the three datasets. As language models generate context-aware embeddings with high-dimensional vectors, a single word token can have different values depending on the given document. Moreover, it can aid in disambiguating the document in terms of syntactic and semantic meaning and boost the performance of document clustering. In the case of SQuAD 1.1 dataset, fastText provides the highest accuracy, followed by comparable results from BERT. As shown in samples in Section 5.1, paragraphs in SQuAD 1.1 always contain the corresponding topic words. This could increase the importance of topic terms and boost clustering performance even with feature-based representations. However, contextualized document representations outperform feature-based approaches on Yahoo Answers and FakeNewsAMT datasets which need intensive language understanding to distinguish the characteristics in each class. Based

on these observations, we conclude that contextualized document representations from language models benefit document clustering in the proposed ADC.

6. Conclusion and future works

In this paper, we propose a simple but efficient deep clustering method called Advanced Document Clustering. In NLP systems, distributed representations for tokens have been widely used for significantly improving the performance. Moreover, empirical experiments in recent studies have demonstrated that contextualized representations can capture syntactic and semantic meaning in the context and advance the state-of-the-art for several NLP tasks. We focus on leveraging those contextualized features and introduce a novel document clustering framework optimized for those high-dimensional vectors. Since contextualized representations imply complex contexts in documents, we use a cosine similarity both in the initialization and clustering module. In this way, we achieve appropriate similarity measure between distributed representations as maximizing the advantages of semantic comparison based on the directions in cosine similarity. Furthermore, to stabilize the self-training process in the ADC, we introduce top-k_m-based optimization and centroids update rule.

We evaluate our method through in-depth experiments with four document corpus and compare the results with four different clustering baselines. The proposed method provides the best performance for all datasets except for REUTERS dataset which is not natural in real world data. We also prove that the proposed optimization and centroids update rule offers significant improvement in performance through comparing against baselines with the same contextualized features. Also, we explore the effect of distance measures in the centroids initialization method. Furthermore, the experimental results demonstrate the effectiveness of using contextualized embeddings in document clustering. A potential future work is to fine-tune representation models in the feature extraction and learn clustering modules simultaneously. Fine-tuning the feature extraction module can further improve the clustering performance as it can generate contextualized and domain-specific representations. Another approach is to use a projection method to high-dimensional vectors to produce smaller dimensional features in the model. Clustering with smaller vectors will guarantee computational efficiency, and an attempt to prevent the information loss in the projection method would be an interesting study.

Conflict of interest

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

Credit authorship contribution statement

Jinuk Park: Conceptualization, Formal analysis, Methodology, Writing - original draft, Writing - review & editing, Software. **Chanhee Park:** Data curation, Writing - original draft. **Jeongwoo Kim:** Investigation, Writing - review & editing. **Minsoo Cho:** Writing - original draft, Validation. **Sanghyun Park:** Conceptualization, Supervision, Validation.

Acknowledgments

This research was supported by the MSIT (Ministry of Science and ICT), Korea, under the SW Starlab support program (IITP-2017-0-00477) supervised by the IITP (Institute for Information & communications Technology Promotion).

References

Arthur, D., & Vassilvitskii, S. (2007). K-means++: The advantages of careful seeding. In *Proc ACM-SIAM symposium on discrete algorithms* (pp. 1027–1035). Society for Industrial and Applied Mathematics.

Bishop, C. M. (2006). *Pattern recognition and machine learning*. Springer.

Blitzer, J., McDonald, R., & Pereira, F. (2006). Domain adaptation with structural correspondence learning. In *Proceedings of the 2006 conference on empirical methods in natural language processing* (pp. 120–128). https://doi.org/10.3115/1610075.1610094.

Caron, M., Bojanowski, P., Joulin, A., & Douze, M. (2018). Deep clustering for unsupervised learning of visual features. In *Proceedings of the European Conference on Computer Vision (ECCV)* (pp. 139–156). https://doi.org/10.1007/978-3-030-01264-9_9.

Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I., & Abbeel, P. (2016). InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in neural information processing systems* (pp. 2172–2180). Retrieved from. http://arxiv.org/abs/1606.03657.

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., & Kuksa, P. (2011). Natural language processing (almost) from scratch. *Journal of Machine Learning Research, 12*(Aug), 2493–2537.

Devlin, J., Chang, M., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of deep bidirectional transformers for language understanding. Retrieved from http://arxiv.org/abs/1810.04805.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press.

Graves, A. (2012). A comparison of network architectures. In *Supervised sequence labelling with recurrent neural networks* (pp. 47–56). Berlin, Heidelberg: Springer. https://doi.org/10.1007/978-3-642-24797-2_5.

Guo, X., Gao, L., Liu, X., & Yin, J. (2017). Improved deep embedded clustering with local structure preservation. In *IJCAI* (pp. 1753–1759). https://doi.org/10.24963/ijcai.2017/243.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 770–778).

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation, 9*(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735.

Hofmann, T., Schölkopf, B., & Smola, A. J. (2008). Kernel methods in machine learning. *The Annals of Statistics, 36*(3), 1171–1220. https://doi.org/10.1214/009053607000000677.

Howard, J., & Ruder, S. (2018). Universal language model Fine-tuning for text classification. In *Proceedings of the 56th annual meeting of the association for computational linguistics (Volume 1: Long Papers)* (pp. 328–339).

Hsu, C. C., & Lin, C. W. (2018). CNN-Based joint clustering and representation learning with feature drift compensation for large-scale image data. *IEEE Transactions on Multimedia, 20*(2), 421–429. https://doi.org/10.1109/TMM.2017.2745702.

Huang, P., Huang, Y., Wang, W., & Wang, L. (2014). Deep embedding network for clustering. In *International conference on pattern recognition* (pp. 1532–1537). https://doi.org/10.1109/ICPR.2014.272.

Jiang, Z., Zheng, Y., Tan, H., Tang, B., & Zhou, H. (2017). Variational deep embedding: An unsupervised and generative approach to clustering. In *IJCAI international joint conference on artificial intelligence* (pp. 1965–1972). https://doi.org/10.24963/ijcai.2017/273.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (pp. 1097–1105).

Lewis, D. D., Yang, Y. M., Rose, T. G., & Li, F. (2004). RCV1: A new benchmark collection for text categorization research. *Journal of Machine Learning Research, 5*(Apr), 361–397.

Von Luxburg, U. (2007). A tutorial on spectral clustering. *Statistics and Computing, 17*(4), 395–416. https://doi.org/10.1007/s11222-007-9033-z.

MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. In *5th Berkeley symposium on mathematical statistics and probability: 1* (pp. 281–297).

Manning, C. D., Raghavan, P., & Schutze, H. (2008). *Introduction to information retrieval*. Cambridge: Cambridge University Press https://doi.org/10.1017/CBO9780511809071.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. Retrieved from http://arxiv.org/abs/1301.3781.

Pennington, J., Socher, R., & Manning, C. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on Empirical Methods in Natural Language Processing (EMNLP)* (pp. 1532–1543). https://doi.org/10.3115/v1/D14-1162.

Pérez-Rosas, V., Kleinberg, B., Lefevre, A., & Mihalcea, R. (2018). Automatic detection of fake news. In *Proceedings of the 27th international conference on computational linguistics* (pp. 3391–3401). Retrieved from. http://arxiv.org/abs/1708.07104.

Peters, M., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., et al. (2018). Deep contextualized word representations. In *proceedings of NAACL-HLT* (pp. 2227–2237). ACL. https://doi.org/10.18653/v1/N18-1202.

Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2018). *Improving language understanding with unsupervised learning* Technical report, OpenAI.

Rajpurkar, P., Zhang, J., Lopyrev, K., & Liang, P. (2016). SQuAD: 100,000+ questions for machine comprehension of text. Retrieved from http://arxiv.org/abs/1606.05250.

Shah, S. A., & Koltun, V. (2018). Deep continuous clustering. Retrieved from http://arxiv.org/abs/1803.01449.

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. Retrieved from http://arxiv.org/abs/1409.1556.

Socher, R., Perelygin, A., Wu, J. Y., Chuang, J., Christopher, D., Manning, A. Y. N., et al. (2013). Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on Empirical Methods in Natural Language Processing (EMNLP)* (pp. 1631–1642).

Song, J., Gao, L., Liu, L., Zhu, X., & Sebe, N. (2018). Quantization-based hashing: A general framework for scalable image and video retrieval. *Pattern Recognition, 75*, 175–187. https://doi.org/10.1016/j.patcog.2017.03.021.

Springenberg, J. T. (2015). Unsupervised and semi-supervised learning with categorical generative adversarial networks. Retrieved from http://arxiv.org/abs/1511.06390.

Steinbach, M., Ertöz, L., & Kumar, V. (2004). The challenges of clustering high dimensional data. In *New directions in statistical physics* (pp. 273–309). Berlin, HeidelbergBerlin Heidelberg: Springer. https://doi.org/10.1007/978-3-662-08968-2_16.

De la Torre, F., & Kanade, T. (2006). Discriminative cluster analysis. In *Proceedings of the 23rd international conference on machine learning* (pp. 241–248). ACM Press. https://doi.org/10.1145/1143844.1143875.

Tian, F., Gao, B., Cui, Q., Chen, E., & Liu, T. (2014). Learning deep representations for graph clustering. In *Proceedings of the twenty-eighth AAAI conference on artificial intelligence* (pp. 1293–1299). AAAI Press.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., et al. (2017). Attention is all you need. In *Advances in neural information processing systems* (pp. 5998–6008). Retrieved from. http://arxiv.org/abs/1706.03762.

Xie, J., Girshick, R., & Farhadi, A. (2016). Unsupervised deep embedding for clustering analysis. In *International conference on machine learning* (pp. 478–487).

Xu, W., Liu, X., & Gong, Y. (2003). Document clustering based on non-negative matrix factorization. In Proceedings of the 26th annual international ACM SIGIR conference on research and development in informaion retrieval – SIGIR '03 (pp. 267–273). ACM. https://doi.org/10.1145/860484.860485.

Yang, B., Fu, X., Sidiropoulos, N. D., & Hong, M. (2017). Towards K-means-friendly spaces: Simultaneous deep learning and clustering. In *Proceedings of the 34th international conference on machine learning-volume 70* (pp. 3861–3870).

Zhang, T., Ramakrishnan, R., & Livny, M. (1996). BIRCH: An efficient data clustering method for very large databases. In *Proceedings of ACM SIGMOD international conference on Management of data* (pp. 103–114). ACM Press. https://doi.org/10.1145/233269.233324.

Zhang, X., Zhao, J., & LeCun, Y. (2015). Character-level convolutional networks for text classification. In *Advances in neural information processing systems* (pp. 649–657).