INTERSPEECH 2019
September 15–19, 2019, Graz, Austria

Neural Text Clustering with Document-level Attention based on Dynamic Soft Labels

Zhi Chen, Wu Guo, Lirong Dai, Zhenhua Ling, Jun Du

National Engineering Laboratory for Speech and Language Information Processing University of Science and Technology of China, Hefei, China

zchen17@mail.ustc.edu.cn, {guowu,lrdai,zhling,jundu}@ustc.edu.cn

Abstract

In this paper, the deep learning framework is applied in text clustering, an unsupervised task in natural language processing (NLP). Since there are no predefined labels available for text clustering, the deep neural network is trained in a pseudo-supervised fashion with labels generated from pre-clustering step. To address the wrong labelling problem from pre-clustering step, we adopt soft pseudo-labels instead of hard one-hot ones, and these labels are dynamically updated during training. Besides, we build a document-level attention over multiple documents based on dynamic soft pseudo-labels to further reduce the impact of the wrong labels. Experimental results on three public databases show that our model outperforms the state-of-the-art systems.

Index Terms: text clustering, pseudo-supervise, dynamic soft labels, document-level attention

1. Introduction

Text clustering, which plays an important role in natural language processing and information retrieval, aims to group similar documents into clusters. In general, the documents are first represented as fixed-dimensional feature vectors, and clustering algorithms are subsequently performed to partition the documents into groups. The most naive but common approach for document representation is bag-of-words, such as the Term Frequency-Inverse Document Frequency (TF-IDF) [1]. However, this simple representation cannot capture the semantics and suffers from curse of dimensionality.

To overcome the drawbacks of TF-IDF, two classes of techniques for document representations, i.e., the unsupervised technique and the supervised technique, are proposed. The unsupervised technique includes several latent semantic analysis methods. Typical methods include Probabilistic Latent Semantic Analysis (PLSA) [2] and Latent Dirichlet Allocation (LDA) [3]. Moreover, neural network-based methods, such as Neural Autoregressive Density Estimators (DocNADE) [4] and the Over-Replicated Softmax [5], have also been investigated for document representations. An alternative approach is to use autoencoder (AE) to learn text representation. For example, Alireza Makhzani proposed K-sparse autoencoder (KSAE) [6] which explicitly enforces sparsity by only keeping the K highest activities in the feedforward phase and Yu Chen proposed K-competitive Autoencoder (KATE) [7] which relies on competitive learning among the autoencoding neurons.

Compared with the unsupervised models, the supervised ones usually generate more discriminative hidden topic features by incorporating label information into the training objective. In recent years, with the help of word embedding, neural network models have brought a new solution to representation of text, such as Recursive Neural Network (RecNN) [8] and Recurrent Neural Network (RNN) [9, 10, 11]. More recently, Convolution Neural Network (CNN), as the most popular neural network model and applying convolutional filters to capture local features, has achieved an obvious performance improvement in terms of constructing text representation [12, 13, 14]. In this paper, we aim to explore the power of CNN and use densely connected CNN (DCCNN) on our text clustering task [14].

The NN mentioned above is primarily designed for specific tasks, such as text classification where predefined labels are provided for the model training. Since there is no training corpus in text-clustering tasks, the issue of how to train a supervised NN model is a dilemma. In this paper, we propose document-level attention based on dynamic soft labels (ADSL) method for text clustering. First, we obtain soft pseudo-labels by pre-clustering. In contrast to hard one-hot pseudo-labels, soft pseudo-labels are used and dynamically updated in NN training to address wrong labels problem. Second, we built document-level attention over multiple documents based on soft pseudo-labels, and the loss function is weighted by the document-level attention. The D-CCNN is trained in a supervised fashion with ADSL and our representation is extracted from the last pooling layer. After obtaining the learned features, agglomerative hierarchical clustering (AHC) algorithm is employed on them to cluster texts into topic categories [15]. The experimental results on three public databases show that the proposed approach significantly boosts the performance of text clustering.

The rest of this paper is organized as follows. In section 2, we describe the proposed pseudo-supervised approach in detail. In section 3, we present the experimental setup and results. Finally, the studys conclusions are presented in section 4.

2. Proposed Method

Although text clustering is unsupervised clustering in nature, the proposed method can tailor the supervised neural network training with this task and generate coarse-to-fine document embedding step by step. The overall architecture is depicted in Figure 1. The method can be divided into three modules:

(A).The soft labels generating module, where the soft pseudo-labels for each document are obtained through pre-clustering;

(B).The soft labels adjusting module, where the soft pseudo-labels for each document are updated;

(C).The DCCNN training module, where the DCCNN is trained with soft pseudo-labels, and document-level attention is applied to the DCCNN's loss function. In the following sections, we will describe them in detail.

http://dx.doi.org/10.21437/Interspeech.2019-1417

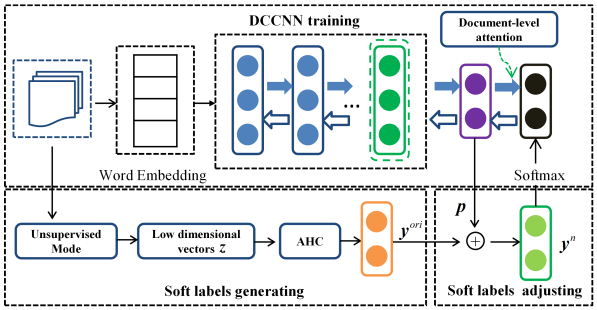


Figure 1: *Architecture of the proposed model.*

2.1. Soft labels generating

All documents $[d_1,d_2,...,d_n]$ are first converted into fixed dimensional vectors $[\boldsymbol{z}_1,\boldsymbol{z}_2,...,\boldsymbol{z}_n]\in\mathbb{R}^l$, which are l-dimensional vectors calculated by the unsupervised model, such as AE or LDA.

Once the document vectors $\boldsymbol{z}$ are generated, the AHC clustering algorithm is applied to partition the text vectors into r clusters, where r is a hyperparameter. Clustering centroids can be obtained, and $[\boldsymbol{u}_1,\boldsymbol{u}_2,...,\boldsymbol{u}_r]\in\mathbb{R}^l$ represent all clustering centroids. The Euclidean distance s_{ij} between document d_i and clustering centroid j can be calculated as:

$$s_{ij} = ||\boldsymbol{z}_i - \boldsymbol{u}_j||_2^2 \tag{1}$$

The choice of target distribution for each document is crucial for the subsequent DCCNN training. A natural approach would be to set the label for each document to a one-hot distribution, which would only take the closest clustering centroid into consideration. However, this approach is inevitably accompanied by the wrong labelling problem. To address this problem, dynamic soft labels are used to replace the one-hot labels. The original soft label y_i^{ori} of document d_i can be calculated using following equations.

$$q_{ij} = \frac{min([s_{i1}, s_{i2}, ..., s_{ir}])}{s_{ij}} \tag{2}$$

$$y_{ij}^{ori} = \frac{(q_{ij})^{\alpha}}{\sum_{k=1}^{r}(q_{ik})^{\alpha}} \tag{3}$$

where α is a hyperparameter and y_{ij}^{ori} can be interpreted as the probability of assigning document d_i to cluster j.

2.2. Soft labels adjusting

The soft pseudo-labels for documents are dynamically updated every T epochs in the NN training procedure, and T is set to 6 in this paper. The pseudo-label $\boldsymbol{y}_i^{n+1}$ of the $(n+1)^{th}$ epoch for document d_i is updated as follows.

$$\boldsymbol{y}_i^{n+1} = \begin{cases} \boldsymbol{y}_i^{ori} + \delta^{(N-n)/T}\boldsymbol{p}_i^n, & \text{if } (n\%T) = 0 \\ \boldsymbol{y}_i^n, & \text{if } (n\%T)! = 0 \end{cases} \tag{4}$$

$$\boldsymbol{y}_i^{n+1} = \text{softmax}(\boldsymbol{y}_i^{n+1}) \tag{5}$$

where the adaptation coefficient $\delta^{(N-n)/T}$ (δ=0.85) controls the balance between the original soft label $\boldsymbol{y}_i^{ori}$ and the posterior-gram $\boldsymbol{p}_i^n$ produced by DCCNN for d_i in the n^{th} epoch. As the epoch n(N is total epoch) increases, $\boldsymbol{p}_i^n$ is more reliable and has more impact on the pseudo-label $\boldsymbol{y}_i^{n+1}$.

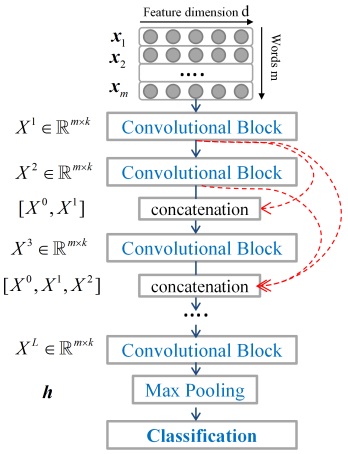


Figure 2: *DCCNN.*

2.3. DCCNN training and clustering

After obtaining the soft pseudo-labels of the documents, the DCCNN is trained. The architecture of the DCCNN for text clustering is depicted in Figure 2.

The input of the DCCNN is the word embedding matrix of a document $\boldsymbol{X}^0$. Let $\boldsymbol{x}_i\in\mathbb{R}^d$ be the d-dimensional pretrained word vector of the i-th word in a document, and the whole document can be represented as an $m\times d$ dimensional matrix:$\boldsymbol{X}^0=[\boldsymbol{x}_1,\boldsymbol{x}_2,...,\boldsymbol{x}_m]$. The DCCNN comprises L layers, each of which implements a nonlinear transformation $f_l(w_l,(.))$, where l indexes the layer and w_l is window size. We denote the output of the l^{th} layer as $\boldsymbol{X}^l$, and $\boldsymbol{X}^l\in\mathbb{R}^{m\times k}$ receives the feature-maps of all preceding layers, $\boldsymbol{X}^1,...,\boldsymbol{X}^{l-1}$, as input [16]:

$$\boldsymbol{X}^l = f_l(w_l, [\boldsymbol{X}^1, ..., \boldsymbol{X}^{l-1}]) \tag{6}$$

where $[\boldsymbol{X}^1,...,\boldsymbol{X}^{l-1}]$ refers to the concatenation of the feature maps produced by layers 1,...,l-1. Lastly, we use a max-pooling operation on $\boldsymbol{X}^L$ and obtain a feature vector $\boldsymbol{h}\in\mathbb{R}^k$.

To fit the pseudo-labels distribution, we apply a logistic operation to the output vector $\boldsymbol{h}$ as follows:

$$\boldsymbol{p} = \text{softmax}(\mathbf{W}_o\boldsymbol{h} + \boldsymbol{b}_o) \tag{7}$$

where $\mathbf{W}_o$ is the weight matrix and $\boldsymbol{b}_o$ is the bias. The output $\boldsymbol{p}\in\mathbb{R}^r$ is the probability distribution over the r cluster, and the target label is $\boldsymbol{y}$ using equation (5). In general, the cross-entropy loss L_i for document d_i in the n^{th} epoch is calculated as follows:

$$L_i = -\sum_{j=1}^{r} y_{ij}^n * logp_{ij}^n \tag{8}$$

where $\boldsymbol{y}_i^n$ and $\boldsymbol{p}_i^n$ are soft pseudo-labels and DCCNN's output of document d_i in the n^{th} epoch.

To address the wrong labelling problem, we use document-level attention to de-emphasize the noisy document. We use $argmax(\boldsymbol{y}_i^n)$ as cluster label of d_i. $\mathbb{C}^k$ stands for a sequence of documents $[d_1^k,d_2^k,...,d_{|\mathbb{C}^k|}^k]$ with the same cluster label k ($argmax(\boldsymbol{y}_1^n)$=...=$argmax(\boldsymbol{y}_{|\mathbb{C}^k|}^n)$=$k$). The loss L_i^k for document d_i^k is calculated as follows.

$$L_i^k = -\sum_{j=1}^{r} \beta_i^k * y_{ij}^n * logp_{ij}^n \tag{9}$$

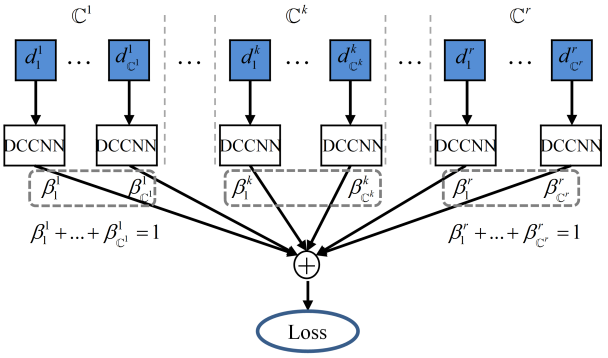


Figure 3: *Architecture of document-level attention.*

where $\boldsymbol{y}_i^n$ and $\boldsymbol{p}_i^n$ are soft pseudo-labels and DCCNN's output of d_i^k in the n^{th} epoch, and β_i^k is a document-level attention weight ($argmax(\boldsymbol{y}_i^n)$=$k$). β_i^k and it is defined as follows.

$$\beta_i^k = \frac{(v_i^k)^2}{\sum_{j \in \mathbb{C}^k} (v_j^k)^2} \tag{10}$$

where v_i^k is a learning parameter, and it is initialized to 1 for all documents when soft pseudo-labels $\boldsymbol{y}^n$ is updated. We jointly optimize v_i^k and all other DCCNN parameters using batch gradient descent in the training process. The value v_i^k of the noisy documents (incorrectly labeled documents) usually obtains a larger gradient and decreases rapidly. In the training procedure, the attention scores of the correct documents are bigger than those of the noisy documents. Through this way, we can train a robust DCCNN which emphasizes the correct documents and de-emphasizes the noisy documents. In addition, document-level attention can normalize loss contribution of each centroid to prevent large clusters which have much more documents than other clusters from distorting the hidden feature space.

The DCCNN is trained in a supervised fashion with dynamic soft labels, and document-level attention is applied to DCCNN loss function. A pooling layer is used to convert the contextual feature vectors of each word in the document produced by DCCNN to a fixed-sized vector representation $\boldsymbol{h}$. Next, we can categorize the documents into r clusters through conventional AHC algorithm.

3. Experiment

In this section, we describe the dataset used for text clustering and the performance of the proposed approach.

3.1. Database

To demonstrate the effectiveness of the proposed method, the experiments are conducted on the Reuters10k [17], StackOverflow [18] and Fisher English corpus [19].

Reuters10k: Reuters contains about 810,000 English news stories labeled with a category tree. We used the four root categories: corporate/industrial, government/social, markets, and economics as labels and further pruned all documents that are labeled by multiple root categories to get 685,071 articles. Since some algorithms do not scale to the full Reuters dataset, we also sampled a random subset of 10,000 examples, which we call REUTERS-10k, for comparison purposes.

StackOverflow: We use the challenge data published in Kaggle.com3. This dataset contains 3,370,528 documents through July 31st, 2012 to August 14, 2012. In our experiments, we randomly select 20, 000 question titles from 20 different tags.

Fisher English corpus: This corpus consists of 11,699 recorded telephone conversations with corresponding text transcriptions. This corpus contains 40 topics in total (such as Movies, Health and Fitness, and Airport Security), and each conversation is assigned to a specific topic. The transcriptions are employed for text clustering. In the experiments, we randomly chose 9,918 documents, resulting in 94∼448 documents for each topic.

3.2. Evaluation Metrics

The clustering performance is evaluated by comparing the clustering labels of the documents with the true labels. Two metrics, the accuracy (ACC) and the normalized mutual information (NMI), are used to measure the performance [18].

3.3. Experimental Settings

In this section, we describe the system configurations, namely, the configurations of the unsupervised model and the proposed ADSL approach.

For unsupervised model, we mainly adopt AE in this paper. In AE,we adopt the same architecture as [17] and set network dimensions to d-500-2000-200-2000-500-d. The input of AE is TF-IDF features which are computed on the d most frequently occurring word stems, and d is set to 3000 for three datasets. The Adam optimizer is used to update parameters, and mini-batch training size is set to 64. Moreover, the learning rate is set to 5e-4 for Fisher and Reuters10k, and 5e-5 for StackOverflow.

For DCCNN, we adopted 3 convolutional blocks for the three datasets. We chose a window size w=3 and a feature dimension k=128. The Adam optimizer is used to update parameters, and the learning rate and mini-batch training size are set to 0.001 and 64, respectively. We initialized the word vectors with 300-dimensional, pretrained, word2vec vectors [20]. During the DCCNN training, all of the pretrained word embeddings are updated along with other model parameters.

3.4. Comparison

The proposed method is compared with the following state-of-the-art models:

DEC: A method that simultaneously learns feature representations and cluster assignments using deep neural networks [17].

STCC[2]: A flexible self-taught convolutional neural network framework for short text clustering, which can flexibly and successfully incorporate more useful semantic features and learn non-biased deep text representation in an unsupervised manner [18].

Consensus: A consensus framework that employs a consensus analysis to select high quality training samples to train RCNN for semantic feature extraction [19].

3.5. Experiments Results

We first conducted the hierarchical clustering algorithm on the document representations inferred by TF-IDF, LDA, KATE and AE. The numbers of clusters are set to 4 for Reuters10k, 20 for StackOverflow and 40 for Fisher(same as the actual number

Table 1: *Comparison of the different unsupervised models.*

Model	Reuters10k	StackOverflow	Fisher
	ACC/NMI(%)	ACC/NMI(%)	ACC/NMI(%)
TF-IDF	53.63/37.23	29.15/22.79	73.05/74.34
LDA	64.02/41.64	55.73/46.58	81.42/81.97
KATE	67.82/**51.56**	32.40/28.77	80.46/81.56
AE	**71.80**/41.73	**64.74/65.06**	**85.23/84.36**

Table 2: *Comparison of the different neural network models.*

Model	Reuters10k	StackOverflow	Fisher
	ACC/NMI(%)	ACC/NMI(%)	ACC/NMI(%)
NN+one	72.32/42.15	66.75/66.53	87.17/86.61
NN+SL	74.38/56.34	74.90/74.16	90.80/90.94
NN+DSL	76.45/57.13	75.30/74.32	91.08/91.34
NN+ADSL	**76.72/57.41**	**75.54/74.60**	**91.31/91.58**
DEC	72.17/-	-/-	-/-
STCC[2]	-/-	51.14/49.08	-/-
Consensus	-/-	-/-	90.75/91.17

of classes) to obtain the best performance. The experimental results of these baseline models are listed in Table 1.

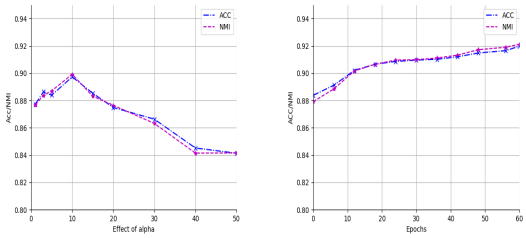
From Table 1, it can be observed that all of these different baseline systems have achieved results comparable with other researchers [17, 18, 19]. We also found that AE performs better than other methods in most cases, so the following experiments are based on AE.

In Table 2, we report the clustering results based on various neural networks. The first 4 rows adopt similar procedure, which use a well-trained DCCNN as feature extractor for the following AHC clustering. NN+one stands for the DCCNN trained with one-hot pseudo-labels. NN+SL stands for the DCCNN trained with the fixed soft labels (use $\boldsymbol{y}^{ori}$ as training target and do not update them in the training procedure). NN+DSL stands for the DCCNN trained with the dynamic soft labels (use $\boldsymbol{y}^n$ in equation (5) as training target). NN+ADSL stands for the DCCNN with document-level attention and trained with the dynamic soft labels (use $\boldsymbol{y}^n$ in equation (5) as training target and apply document-level attention to DCCNN loss function).

The NN+one system can only achieve marginal improvements over the baseline systems in Table 2. The reason may be wrong labels of the training samples and these noisy data will have very negative impact in model training. In contrast, the NN+SL can obviously improve the performance. Because soft labels can reduce the impact of wrong labelling samples during DCCNN training. We can also find that NN+DSL generally performs better than NN+SL. It indicates that dynamic mechanism can correct some wrong labelling samples during training. The best results can be obtained by the proposed NN+ADSL system which integrates dynamic soft label and document-level attention. Compare to NN+DSL, NN+ADSL generally performs better. Since we regard each document unequally through document-level attention in ADSL system, the wrong labelling samples will have less influence on DCCNN training.

Furthermore, the proposed method can achieve much better performance than DEC, STCC2 and Consensus.

The α in equation (3) is a hyperparameter, and its selection may affect the final performance. To test the influence of α, we conducted experiments on the Fisher corpus using different values. Figure 4a depicts the ACC and the NMI curves. The best



(a) (b)

Figure 4: *(a) Effect of alpha with different values (b) ACC(%) and NMI(%) of $\boldsymbol{y}^n$ with varying training epochs.*

Table 3: *Cluster performance with different numbers*

Model	30		50	
	ACC(%)	NMI(%)	ACC(%)	NMI(%)
TF-IDF	64.52	71.62	69.53	74.23
LDA	72.34	78.95	74.85	80.13
KATE	71.70	77.98	73.03	76.95
AE	74.16	82.05	79.34	83.25
Consensus	77.50	86.06	**84.84**	89.09
NN+one	75.40	84.04	80.36	84.16
NN+ADSL	**77.71**	**86.87**	82.72	**89.25**

performance can be achieved at α=10, and it deteriorates with large α. One plausible reason is that soft labels start to evolve into one-hot labels with large α. In the above and subsequent experiments, the parameter α is set to 10 for all three datasets.

Figure 4b shows the ACC and NMI of soft pseudo-labels (we use $argmax(\boldsymbol{y}^n)$ as cluster label) in different iterations on fisher datasets using the proposed method. It can be found that the ACC and NMI of $\boldsymbol{y}^n$ rise steadily as the iteration number increases, which demonstrates that the ADSL method can correct some incorrect soft pseudo-labels during training.

So far, we have set the final cluster number to the actual cluster number to simplify the experiments. However, this number is often unknown in practice. To demonstrate the generalizability of our approach, experiments are further conducted with cluster numbers of 30 and 50 for the Fisher corpus. Table 3 lists the clustering results with different cluster numbers. It can be seen that the proposed approach can also yield a performance improvement compared to a baseline of different cluster numbers.

4. Conclusions

In this paper, we carried out text clustering experiments based on the semantic features learned by the DCCNN. The proposed dynamic soft label targets can alleviate the wrong labelling problem for pseudo-supervised DCCNN training. The experimental results show that the proposed method can significantly and consistently outperform state-of-the-art methods.

5. Acknowledgment

This work was partially funded by the National Key Research and Development Program of China (Grant No. 2016YFB1001303) and the National Natural Science Foundation of China (Grant No. U1836219).

6. References

[1] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," *Information processing & management*, vol. 24, no. 5, pp. 513–523, 1988.

[2] T. Hofmann, "Probabilistic latent semantic indexing," in *ACM SIGIR Forum*, vol. 51, no. 2. ACM, 2017, pp. 211–218.

[3] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of machine Learning research*, vol. 3, no. Jan, pp. 993–1022, 2003.

[4] H. Larochelle and S. Lauly, "A neural autoregressive topic model," in *Advances in Neural Information Processing Systems*, 2012, pp. 2708–2716.

[5] G. E. Hinton and R. R. Salakhutdinov, "Replicated softmax: an undirected topic model," in *Advances in neural information processing systems*, 2009, pp. 1607–1614.

[6] A. Makhzani and B. Frey, "K-sparse autoencoders," *arXiv preprint arXiv:1312.5663*, 2013.

[7] Y. Chen and M. J. Zaki, "Kate: K-competitive autoencoder for text," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2017, pp. 85–94.

[8] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Ng, and C. Potts, "Recursive deep models for semantic compositionality over a sentiment treebank," in *Proceedings of the 2013 conference on empirical methods in natural language processing*, 2013, pp. 1631–1642.

[9] D. Tang, B. Qin, and T. Liu, "Document modeling with gated recurrent neural network for sentiment classification," in *Proceedings of the 2015 conference on empirical methods in natural language processing*, 2015, pp. 1422–1432.

[10] J. Y. Lee and F. Dernoncourt, "Sequential short-text classification with recurrent and convolutional neural networks," *arXiv preprint arXiv:1603.03827*, 2016.

[11] B. Wang, "Disconnected recurrent neural networks for text categorization," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, vol. 1, 2018, pp. 2311–2320.

[12] Y. Kim, "Convolutional neural networks for sentence classification," *arXiv preprint arXiv:1408.5882*, 2014.

[13] A. Conneau, H. Schwenk, L. Barrault, and Y. Lecun, "Very deep convolutional networks for natural language processing," *arXiv preprint*, 2016.

[14] S. Wang, M. Huang, and Z. Deng, "Densely connected cnn with multi-scale feature attention for text classification." in *IJCAI*, 2018, pp. 4468–4474.

[15] L. Rocach and O. Maimon, "Clustering methods data mining and knowledge discovery handbook," *Springer US*, p. 321, 2005.

[16] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks." in *CVPR*, vol. 1, no. 2, 2017, p. 3.

[17] J. Xie, R. Girshick, and A. Farhadi, "Unsupervised deep embedding for clustering analysis," in *International conference on machine learning*, 2016, pp. 478–487.

[18] J. Xu, B. Xu, P. Wang, S. Zheng, G. Tian, and J. Zhao, "Self-taught convolutional neural networks for short text clustering," *Neural Networks*, vol. 88, pp. 22–31, 2017.

[19] P. Chen, W. Guo, L. Dai, and Z. Ling, "Pseudo-supervised approach for text clustering based on consensus analysis," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 6184–6188.

[20] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.