Improved Supporting Clustering with Cluster-level Contrastive Learning

JUNTONG NI, Shandong University, China

Analyzing short texts infers discriminative and coherent latent topics, which is a significant and essential problem because many real-world applications require semantic understanding of short messages. Due to the extremely low amount of word co-occurrence information available in short texts, conventional long text topic modeling techniques (such PLSA and LDA) based on word co-occurrences are unable to adequately address this issue. In this study, we propose an improved version of Supporting Clustering with Contastive Learning (SCCL) [Zhang et al. 2021b] named Improved Supporting Clustering with Contastive Learning (ISCCL). The key of the improvement is that we add cluster-level head to leverage cluster-level contrastive learning to promote better feature separation. We assess the performance of ISCCL on short text clustering benchmark dataset SearchSnippets and show that ISCCL significantly advances the state-of-art results with 0.7 improvement on Normalized Mutual Information and 0.4 improvement on Accuracy. We have released the source code to facilitate the reproduction [1].

CCS Concepts: • **Pattern Recognition → Clustering Algorithms**.

Additional Key Words and Phrases: Short text clustering, Contrastive Learning

ACM Reference Format:
JUNTONG NI. 2023. Improved Supporting Clustering with Cluster-level Contrastive Learning . *ACM Trans. Graph.* 37, 4, Article 111 (August 2023), 3 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Clustering, one of the most fundamental challenges in unsupervised learning, has been widely studied for decades. Long established clustering methods such as K-means [MacQueen 1967] and Gaussian Mixture Models [Celeux and Govaert 1995] rely on distance measured in the data space, which tends to be ineffective for high-dimensional data. On the other hand, deep neural networks are gaining momentum as an effective way to map data to a low dimensional and hopefully better separable representation space.

On the other hand, Instance-wise Contrastive Learning (Instance-CL) [Zhang et al. 2021b] has recently achieved remarkable success in self-supervised learning. Instance-CL usually optimizes on an auxiliary set obtained by data augmentation. As the name suggests, a contrastive loss is then adopted to pull together samples augmented from the same instance in the original dataset while pushing apart those from different ones. For clustering, Instance-CL better separates different clusters while tightening each cluster by explicitly bringing samples in that cluster together.

[1]https://github.com/LingFengGold/ISCCL

Author's address: JUNTONG NI, Shandong University, Binhai Street, Qingdao, China, juntongni02@gmail.com.

0730-0301/2023/8-ART111 $15.00
https://doi.org/XXXXXXX.XXXXXXX

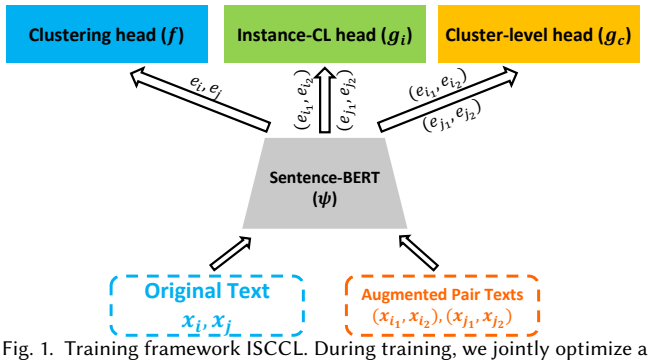

Fig. 1. Training framework ISCCL. During training, we jointly optimize a clustering loss over the original data instances, an instance-wise contrastive loss and an cluster-level contrastive loss over the associated augmented pairs.

However, we assume that this cluster-level contrastive learning is incredibly important in textual representation learning. To validate this hypothesis, in this work, we present an improvement based on SCCL [Zhang et al. 2021b], a Improved Supporting Clustering with Contrastive Learning (ISCCL) with Cluster-Level Contrastive Learning method that greatly advances the state-of-the-art text clustering models by simultaneously optimizing the clustering loss, instance-level and cluster-level contrastive loss. We demonstrate that our approach outperforms the state-of-the-art text clustering models on benchmark dataset SearchSnippets.

2 METHOD

We aim at developing a joint model that leverages the beneficial properties of Instance-CL and Cluster-level CL to improve unsupervised clustering. As illustrated in Figure 1, our model consists of three components. A neural network (*e.g.,* Sentence-BERT) $\psi(\cdot)$ first maps the input data to the representation space, which is then followed by three different heads $f(\cdot)$, $g_i(\cdot)$, and $g_c(\cdot)$ where the clustering loss, the Instance-CL and Cluster-level CL loss are applied, respectively.

Our data consists of both the original and the augmented data. Specifically, for a randomly sampled minibatch $\mathcal{B} = \{x_i\}_{i=1}^{M}$, we randomly generate a pair of augmentations for each data instance in $\mathcal{B}$, yielding an augmented batch $\mathcal{B}^a$ with size $2M$, denoted as $\mathcal{B}^a = \{\widetilde{x}_i\}_{i=1}^{M}$.

2.1 Instance-wise Contrastive Learning

For each minibatch $\mathcal{B}$, the Instance-CL loss is defined on the augmented pairs in $\mathcal{B}^a$. Let $i^1 \in \{1, \ldots, 2M\}$ denote the index of an arbitrary instance in augmented set $\mathcal{B}^a$, and let $i^2 \in \{1, \ldots, 2M\}$ be the index of the other instance in $\mathcal{B}^a$ augmented from the same instance in th e original set $\mathcal{B}$. We refer to $\widetilde{x}_{i^1}, \widetilde{x}_{i^2} \in \mathcal{B}^a$ as a *positive* pair, while treating the other $2M-2$ examples in $\mathcal{B}^a$ as *negative* instances regrading this positive pair. Let $\widetilde{z}_{i^1}$ and $\widetilde{z}_{i^2}$ be the corresponding outputs of the Instance-CL head g_i, *i.e.,* $\widetilde{z}_{i^j} = g_i(\psi(\widetilde{x}_j)), j = i^1, i^2$. g_i consist of two layer multilayer perceptron (MLP) and $\widetilde{z}_{i^j} \in \mathbb{R}^{M \times H}$,

where H denotes the hidden size. Then for $\widetilde{x}_{ij}$, we try to separate x_{i^2} apart from all negative instances in $\mathcal{B}^a$ by minimizing the following

$$\mathcal{L}_{i^1}^I = -\log \frac{\exp(\text{sim}(\widetilde{z}_{i^1}, \widetilde{z}_{i^2})/\tau)}{\sum_{j=1}^{2M} \mathbb{1}_{j \neq i^1} \cdot \exp(\text{sim}(\widetilde{z}_{i^1}, \widetilde{z}_j)/\tau)} \tag{1}$$

Here $\mathbb{1}_{j \neq i^1}$ is an indicator function and τ denotes the temperature parameter which we set as 0.5. Following [Chen et al. 2020], we choose $\text{sim}(\cdot)$ as the dot product between a pair of normalized outputs, *i.e.,* $\text{sim}(\widetilde{z}_i, \widetilde{z}_j) = \widetilde{z}_i^T \widetilde{z}_j / \|\widetilde{z}_i\|_2 \|\widetilde{z}_j\|_2$.

The Instance-CL loss is then averaged over all instances in $\mathcal{B}^a$,

$$\mathcal{L}_{Instance-CL} = \sum_{i=1}^{2M} \mathcal{L}_i^I / 2M \tag{2}$$

To apply the above constrastive loss in the text domain, we utilize contextual augmenter [Kobayashi 2018] to generate augmented pair texts.

2.2 Clustering

We simultaneously encode the semantic categorical structure into the representations via unsupervised clustering. Unlike Instance-CL, clustering focuses on the high-level semantic concepts and tries to bring together instances from the same semantic category together. Suppose our data consists of K semantic categories, and each category is characterized by its centroid in the representation space, denoted as $\mu_k, k \in \{1, \ldots, K\}$. Let $e_j = \psi(x_j)$ denote the representation of instance x_j in the original set $\mathcal{B}$. Following [Van der Maaten and Hinton 2008], we use the Student's t-distribution to compute the probability of assigning x_j to the k^th cluster,

$$q_{jk} = \frac{(1 + \|e_j - \mu_k\|_2^2/\alpha)^{-\frac{\alpha+1}{2}}}{\sum_{k'=1}^{K} (1 + \|e_j - \mu_{k'}\|_2^2/\alpha)^{-\frac{\alpha+1}{2}}} \tag{3}$$

Here α denotes the degree of freedom of the Student's t-distribution. Without explicit mention, we follow [Van der Maaten and Hinton 2008] by setting $\alpha = 1$ in this paper.

We use a linear layer, *i.e.,* the clustering head in Figure 1, to approximate the centroids of each cluster, and we iteratively refine it by leveraging an auxiliary distribution proposed by [Xie et al. 2016]. Specifically, let p_{jk} denote the auxiliary probability defined as

$$p_{jk} = \frac{q_{jk}^2 / f_k}{\sum_{k'} q_{jk}^2 / f_{k'}} \tag{4}$$

Here $f_k = \sum_{j=1}^{M} q_{jk}, k = 1, \ldots, K$ can be interpreted as the soft cluster frequencies approximated within a minibatch. This target distribution first sharpens the soft-assignment probability q_{jk} by raising it to the second power, and then normalizeds it by the associated cluster frequency. By doing so, we encourage learning from high confidence cluster assignments and simultaneously combating the bias caused by imbalanced clusters.

We push the cluster assignment probability towards the target distribution by optimizing the KL divergence between them,

$$\mathcal{L}_j^C = \text{KL}[\text{p}_\text{j} || \text{q}_\text{j}] = \sum_{\text{k}=1}^{\text{K}} \text{p}_\text{jk} \log \frac{\text{p}_\text{jk}}{\text{q}_\text{jk}} \tag{5}$$

The clustering objective is then followed as

$$\mathcal{L}_{Cluster} = \sum_{j=1}^{M} \mathcal{L}_j^C / M \tag{6}$$

This clustering loss is first proposed in [Xie et al. 2016] and later adopted by [Hadifar et al. 2019] for short text clustering. However, they both require expensive layer-wise pretraining of the neural network, and update the target distribtuion (Eq (4)) through carefully chosen intervals that often vary across datasets. In contrast, we simplify the learning process to end-to-end training with the target distribution being updated per iteration.

2.3 Cluster-level Contrastive Learning

Similar to Instance-CL loss, for each minibatch $\mathcal{B}$, the Cluster-level CL loss is defined on the augmented pairs in $\mathcal{B}^a$. Let $\widetilde{z}_{i^1}$ and $\widetilde{z}_{i^2}$ be the corresponding outputs of the Cluster-level head g_c, *i.e.,* $\widetilde{z}_{ij} = g_c(\psi(\widetilde{x}_j)), j = i^1, i^2$. g_c consist of two layer multilayer perceptron (MLP) and $\widetilde{z}_{ij} \in \mathbb{R}^{C \times M}$, where C denotes the number of clusters. Then we calculate $\mathcal{L}_i^I$ by applying Eq (1).

The Cluster-level CL loss is then averaged over all instances in $\mathcal{B}^a$,

$$\mathcal{L}_{Cluster-level\ CL} = \sum_{k=1}^{2C} \mathcal{L}_k^{CCL} / 2C \tag{7}$$

To apply the above constrastive loss in the text domain, we utilize contextual augmenter [Kobayashi 2018] to generate augmented pair texts.

2.4 Overall objective

In summary, our overall objective is,

$$\begin{aligned} \mathcal{L} &= \mathcal{L}_{Cluster} + \eta \mathcal{L}_{Instance-CL} + \eta \mathcal{L}_{Cluster-level\ CL} \\ &= \sum_{j=1}^{M} \mathcal{L}_j^C / M + \eta \sum_{i=1}^{2M} \mathcal{L}_i^I / 2M + \eta \sum_{k=1}^{2C} \mathcal{L}_k^{CCL} / 2C \end{aligned} \tag{8}$$

$\mathcal{L}_j^C$, $\mathcal{L}_i^I$ and $\mathcal{L}_k^{CCL}$ are defined in Eq (5), Eq (2) and Eq (7), respectively. η balances between the contrastive loss and the clustering loss of ISCCL, which we set as 10 for simplicity. Also note that, the clustering loss is optimized over the original data only.

3 EXPERIMENTS

3.1 SearchSnippets Datasets

SearchSnippets is extracted from web search snippets, which contains 12,340 snippets associated with 8 groups [Phan et al. 2008].

3.2 Evaluate Metrics

To measure the performance of methods, prior studies utilize several metrics [Zhang et al. 2021a], the normalized mutual information (NMI) [Strehl and Ghosh 2002], and clustering accuracy (ACC) [Xie et al. 2016].

3.2.1 NMI. Normalized mutual information is defined as:

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

Dataset	SearchSnippets	
Metrics	NMI	ACC
Kmeans(TF)	9.0	24.7
Kmeans(TF-IDF)	21.4	33.8
Kmeans(Skip-Thought)	13.8	33.6
Kmeans(Sentence-BERT)	52.0	66.8
Kmeans(SIF)	36.9	53.4
DEC	64.9	76.9
STCC	62.9	77.0
Self-Train	56.7	77.1
SCCL	71.4	84.9
ISCCL(Ours)	72.1	85.3

Table 1. Overall results.

where T is the true labels sets, C is the predicted assignments, MI(T,C) is the mutual information between T and C, H is the entropy and $\sqrt{H(T)H(C)}$ is used for normalizing the mutual information to be $[0, 1]$.

3.2.2 ACC. Accuracy is defined as:

$$ACC = \frac{\sum_{i=1}^{n} \delta(y_i == map(c_i))}{n}$$

where n is the number of the text, $\delta(p == q)$ is the indicator function that equals one if $p = q$ and equlas zero otherwise, c_i is the clustering label of text x_i, y_i is the true group label of text x_i, $map(c_i)$ is a function that maps each cluster label c_i to the equivalent label from the text data by Hungarian algorithm [Kuhn 1955].

3.3 Baselines

- Representations Learning: Generic, low-dimensional representations such as TF, TF-IDF, Skip-Thought [Kiros et al. 2015], Sentence-BERT [Reimers and Gurevych 2019], and SIF [Arora et al. 2017] embeddings have demonstrated to be beneficial for NLP on many tasks. We utilize Kmeans on such low-dimensional representations.
- DEC [Xie et al. 2016] [2]: It is a deep embedded clustering model that leverages autoencoder, with TF-IDF features as input to map documents into low-dimensional embeddings. Then the mapping function and cluster representations are refined based on the idea of self-training. DEC has specified transforming and parameter settings for text data.
- STCC [Xu et al. 2017] [3]: This model mainly consists of three separate steps. It first trains a convolutional neural network with the help of autoencoders. Afterwards, the well-trained model is employed to get text embeddings, which are fed into K-means for final clustering.
- Self-Train [Hadifar et al. 2019] [4]: STC adopts self-training inspired by DEC and uses Smoothed Inverse Frequency (SIF) to compute a weighted average of pre-trained word embeddings in a stage independent of optimizing its clustering model.
- SCCL [Zhang et al. 2021b] [5]: They propose Supporting Clustering with Contrastive Learning (SCCL) by jointly optimizing a top-down clustering loss with a bottom-up instance-wise contrastive loss. Our method is a improved version of this baseline.

Performance results with average over 5 runs are from [Hadifar et al. 2019]. The results of Kmeans(Sentence-BERT) and DEC are from [Li and Wang 2022]. The results of SCCL are reproduced by official code.

4 CONCLUSION

As shown in Table 1, our method either substantilly outperforms or performs highly comparably to the state-of-art methods. We demonstrate that, by integrating the strengths of adding cluster-level contrastive learning, our model is capable of generating high-quality clusters with better intra-cluster and inter-clusters distances.

REFERENCES

Sanjeev Arora, Yingyu Liang, and Tengyu Ma. 2017. A simple but tough-to-beat baseline for sentence embeddings. In *International conference on learning representations*.

Gilles Celeux and Gérard Govaert. 1995. Gaussian parsimonious clustering models. *Pattern recognition* 28, 5 (1995), 781–793.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*. PMLR, 1597–1607.

Amir Hadifar, Lucas Sterckx, Thomas Demeester, and Chris Develder. 2019. A self-training approach for short text clustering. In *Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019)*. 194–199.

Ryan Kiros, Yukun Zhu, Russ R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Skip-thought vectors. *Advances in neural information processing systems* 28 (2015).

Sosuke Kobayashi. 2018. Contextual augmentation: Data augmentation by words with paradigmatic relations. *arXiv preprint arXiv:1805.06201* (2018).

Harold W Kuhn. 1955. The Hungarian method for the assignment problem. *Naval research logistics quarterly* 2, 1-2 (1955), 83–97.

Ruihui Li and Hongbin Wang. 2022. Clustering of Short Texts Based on Dynamic Adjustment for Contrastive Learning. *IEEE Access* 10 (2022), 76069–76078.

J MacQueen. 1967. Classification and analysis of multivariate observations. In *5th Berkeley Symp. Math. Statist. Probability*. 281–297.

Xuan-Hieu Phan, Le-Minh Nguyen, and Susumu Horiguchi. 2008. Learning to classify short and sparse text & web with hidden topics from large-scale data collections. In *Proceedings of the 17th international conference on World Wide Web*. 91–100.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084* (2019).

Alexander Strehl and Joydeep Ghosh. 2002. Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *Journal of machine learning research* 3, Dec (2002), 583–617.

Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research* 9, 11 (2008).

Junyuan Xie, Ross Girshick, and Ali Farhadi. 2016. Unsupervised deep embedding for clustering analysis. In *International conference on machine learning*. PMLR, 478–487.

Jiaming Xu, Bo Xu, Peng Wang, Suncong Zheng, Guanhua Tian, and Jun Zhao. 2017. Self-taught convolutional neural networks for short text clustering. *Neural Networks* 88 (2017), 22–31.

Dejiao Zhang, Feng Nan, Xiaokai Wei, Shangwen Li, Henghui Zhu, Kathleen McKeown, Ramesh Nallapati, Andrew Arnold, and Bing Xiang. 2021b. Supporting clustering with contrastive learning. *arXiv preprint arXiv:2103.12953* (2021).

Kai Zhang, Zheng Lian, Jiangmeng Li, Haichang Li, and Xiaohui Hu. 2021a. Short Text Clustering with a Deep Multi-embedded Self-supervised Model. In *International Conference on Artificial Neural Networks*. Springer, 150–161.

[2]https://github.com/piiswrong/dec
[3]https://github.com/jacoxu/STC2
[4]https://github.com/hadifar/stc_clustering
[5]https://github.com/amazon-science/sccl