

Robust Representation Learning with Reliable Pseudo-labels Generation via Self-Adaptive Optimal Transport for Short Text Clustering

Xiaolin Zheng, Mengling Hu, Weiming Liu, Chaochao Chen*, and Xinting Liao

Zhejiang University, China

{xlzheng, humengling, 21831010, zjuccc, xintingliao}@zju.edu.cn

Abstract

Short text clustering is challenging since it takes imbalanced and noisy data as inputs. Existing approaches cannot solve this problem well, since (1) they are prone to obtain degenerate solutions especially on heavy imbalanced datasets, and (2) they are vulnerable to noises. To tackle the above issues, we propose a Robust Short Text Clustering (**RSTC**) model to improve robustness against imbalanced and noisy data. **RSTC** includes two modules, i.e., *pseudo-label generation module* and *robust representation learning module*. The former generates pseudo-labels to provide supervision for the later, which contributes to more robust representations and correctly separated clusters. To provide robustness against the imbalance in data, we propose self-adaptive optimal transport in the pseudo-label generation module. To improve robustness against the noise in data, we further introduce both class-wise and instance-wise contrastive learning in the robust representation learning module. Our empirical studies on eight short text clustering datasets demonstrate that **RSTC** significantly outperforms the state-of-the-art models. The code is available at: <https://github.com/hmllmh/RSTC>.

1 Introduction

Text clustering, one of the most fundamental tasks in text mining, aims to group text instances into clusters in an unsupervised manner. It has been proven to be beneficial in many applications, such as, recommendation system (Liu et al., 2021, 2022a,b), opinion mining (Stieglitz et al., 2018), stance detection (Li et al., 2022), etc. With the advent of digital era, more and more people enjoy sharing and discovering various of contents on the web, where short text is an import form of information carrier. Therefore, it is helpful to utilize short text clustering for mining valuable insights on the web.

However, short text clustering is not a trivial task. On the one hand, short text has many categories and the category distributions are diversifying, where the heavy imbalanced data is common. The heavy imbalanced data is prone to lead to degenerate solutions where the tail clusters (i.e., the clusters with a small proportion of instances) disappear. Specifically, the recent deep joint clustering methods for short text clustering, (Hadifar et al., 2019) and (Zhang et al., 2021), adopt the clustering objective proposed in (Xie et al., 2016), which may obtain a trivial solution where all the text instances fall into the same cluster (Yang et al., 2017; Ji et al., 2019). (Zhang et al., 2021) introduces instance-wise contrastive learning to train discriminative representations, which avoids the trivial solution to some extent. However, (Zhang et al., 2021) still tends to generate degenerate solutions, especially on the heavy imbalanced datasets.

On the other hand, short text is typically characterized by noises, which may lead to meaningless or vague representations and thus hurt clustering accuracy and stability. Existing short text clustering methods cope with the noise problem in three ways, i.e., (1) text preprocessing, (2) outliers post-processing, and (3) model robustness. Specifically, earlier methods (Xu et al., 2017; Hadifar et al., 2019) apply preprocessing procedures on the text (HaCohen-Kerner et al., 2020) for reducing the negative impact of noises. The recent method (Rakib et al., 2020) proposes to postprocess outliers by repeatedly reassigning outliers to clusters for enhancing the clustering performance. However, both preprocessing and postprocessing methods do not provide model robustness against the noise in data. The more recently short text clustering method SCCL (Zhang et al., 2021) proposes to utilize the instance-wise contrastive learning to support clustering, which is useful for dealing with the noises in the perspective of model robustness. However, the learned representations of SCCL lack discrim-

*Corresponding author.

inability due to the lack of supervision information, causing insufficiently robust representations.

In summary, there are two main challenges, i.e., **CH1**: How to provide model robustness to the imbalance in data, and avoid the clustering degeneracy? **CH2**: How to improve model robustness against the noise in data, and enhance the clustering performance?

To address the aforementioned issues, in this paper, we propose **RSTC**, an end-to-end model for short text clustering. In order to improve model robustness to the imbalance in data (solving **CH1**) and the noise in data (solving **CH2**), we utilize two modules in **RSTC**, i.e., *pseudo-label generation module* and *robust representation learning module*. The pseudo-label generation module generates pseudo-labels for the original texts. The robust representation learning module uses the generated pseudo-labels as supervision to facilitate intra-cluster compactness and inter-cluster separability, thus attaining more robust representations and more correctly separated clusters. The better cluster predictions in turn can be conducive to generate more reliable pseudo-labels. The iterative training process forms a virtuous circle, that is, the learned representations and cluster predictions will constantly boost each other, as more reliable pseudo-labels are discovered during iterations.

The key idea to solve **CH1** is to enforce a constraint on pseudo-labels, i.e., the distribution of the generated pseudo-labels should match the estimated class distribution. The estimated class distribution is dynamically updated and expected to get closer to the ground truth progressively. Meanwhile, the estimated class distribution are encouraged to be a uniform distribution for avoiding clustering degeneracy. We formalize the idea as a new paradigm of optimal transport (Peyré et al., 2019) and the optimization objective can be tractably solved by the Sinkhorn-Knopp (Cuturi, 2013) style algorithm, which needs only a few computational overheads. For addressing **CH2**, we further introduce *class-wise* contrastive learning and *instance-wise* contrastive learning in the robust representation learning module. The class-wise contrastive learning aims to use the pseudo-labels as supervision for achieving smaller intra-cluster distance and larger inter-cluster distance. While the instance-wise contrastive learning tends to disperse the representations of different instances apart for the separation of overlapped clusters. These two modules

cooperate with each other to provide better short text clustering performance.

We summarize our main contributions as follows: (1) We propose an end-to-end model, i.e., **RSTC**, for short text clustering, the key idea is to discover the pseudo-labels to provide supervision for robust representation learning, hence enhancing the clustering performance. (2) To our best knowledge, we are the first to propose self-adaptive optimal transport for discovering the pseudo-label information, which provides robustness against the imbalance in data. (3) We propose the combination of class-wise contrastive learning and instance-wise contrastive learning for robustness against the noise in data. (4) We conduct extensive experiments on eight short text clustering datasets and the results demonstrate the superiority of **RSTC**.

2 Related Work

2.1 Short Text Clustering

Short text clustering is not trivial due to imbalanced and noisy data. The existing short text clustering methods can be divided into three kinds: (1) traditional methods, (2) deep learning methods, and (3) deep joint clustering methods. The traditional methods (Scott and Matwin, 1998; Salton and McGill, 1983) often obtain very sparse representations that lack discriminations. The deep learning method (Xu et al., 2017) leverages pre-trained word embeddings (Mikolov et al., 2013) and deep neural network to enrich the representations. However, the learned representations may not appropriate for clustering. The deep joint clustering methods Hadifar et al. (2019); Zhang et al. (2021) integrate clustering with deep representation learning to learn the representations that are appropriate for clustering. Moreover, Zhang et al. (2021) utilizes the pre-trained SBERT (Reimers and Gurevych, 2019) and contrastive learning to learn discriminative representations, which is conducive to deal with the noises. However, the adopted clustering objectives are prone to obtain degenerate solutions (Yang et al., 2017; Ji et al., 2019), especially on heavy imbalance data.

Among the above methods, only Zhang et al. (2021) provides model robustness to the noise in data. However, its robustness is still insufficient due to the lack of supervision information. Besides, Zhang et al. (2021) cannot deal with various imbalanced data due to the degeneracy problem. As a contrast, in this work, we adopt pseudo-label

technology to provide reliable supervision to learn robust representations for coping with imbalanced and noisy data.

2.2 Pseudo-labels for Unsupervised Learning

Pseudo-labels can be helpful to learn more discriminative representations in unsupervised learning (Hu et al., 2021). Caron et al. (2018) shows that k-means clustering can be utilized to generate pseudo-labels for learning visual representations. However, it does not have a unified, well-defined objective to optimize (i.e., there are two objectives: k-means loss minimization and cross-entropy loss minimization), which means that it is difficult to characterize its convergence properties. Asano et al. (2020) proposes SeLa to optimize the same objective (i.e., cross-entropy loss minimization) for both pseudo-label generation and representation learning, which can guarantee its convergence. Besides, SeLa transforms pseudo-label generation problem into an optimal transport problem. Caron et al. (2020) proposes SwAV which combines SeLa with contrastive learning to learn visual representations in an online fashion. However, both SeLa and SwAV add the constraint that the distribution of generated pseudo-labels should match the uniform distribution, to avoid clustering degeneracy. With the constraint, it is hard for them to cope with imbalanced data. As a contrast, in this work, we propose self-adaptive optimal transport to simultaneously estimate the real class distribution and generate pseudo-labels. Our method enforces the distribution of the generated pseudo-labels to match the estimated class distribution, and thus can avoid clustering degeneracy and adapt to various imbalanced data.

3 Methodology

3.1 An Overview of RSTC

The goal of **RSTC** is to discover and utilize the pseudo-labels to provide supervision for robust representation learning. **RSTC** consists of *pseudo-label generation module* and *robust representation learning module*, as illustrated in Fig.1. The pseudo-label generation module aims to generate reliable pseudo-labels for the robust representation learning module. To achieve this aim, we first obtain cluster predictions by the cluster assignment step, then we excavate pseudo-label information from the predictions by the self-adaptive optimal transport (SAOT) step. The robust representation

learning module aims to use the generated pseudo-labels as supervision to train robust representations. To achieve this goal, we introduce class-wise and instance-wise contrastive learning. In this way, **RSTC** can provide robustness to imbalanced and noisy data, thus enhancing the clustering performance.

3.2 Pseudo-label Generation Module

We first introduce the pseudo-label generation module. Although the deep joint clustering methods (Xie et al., 2016; Hadifar et al., 2019; Zhang et al., 2021) are popular these days, their clustering performance is limited due to the following reasons. Firstly, lacking supervision information prevents the deep joint clustering methods from learning more discriminative representations (Hu et al., 2021). Secondly, they are prone to obtain degenerate solutions (Yang et al., 2017; Ji et al., 2019), especially on heavy imbalanced datasets. Therefore, to provide reliable supervision information for various imbalanced data, we propose SAOT in the pseudo-label generation module to generate pseudo-labels for the robust representation learning module. The overview of pseudo-label generation module is shown in Fig.1(a), which mainly has two steps: **Step 1**: cluster assignment, and **Step 2**: SAOT.

Step 1: cluster assignment. Cluster assignment aims to obtain cluster predictions of the original texts. Specifically, we adopt SBERT (Reimers and Gurevych, 2019) as the encoding network Φ to encode the original text \mathbf{X} as $\Phi(\mathbf{X}) = \mathbf{E} \in \mathbb{R}^{N \times D_1}$ where N denotes batch size and D_1 is the dimension of the representations. We utilize the fully connected layers as the clustering network G_p to predict the cluster assignment probability (predictions), i.e., $G_p(\mathbf{E}) = \mathbf{P} \in \mathbb{R}^{N \times C}$, where C is the category number. The encoding network and the clustering network are fixed in this module.

Step 2: SAOT. SAOT aims to exploit the cluster predictions to discover reliable pseudo-label. Asano et al. (2020) extends standard cross-entropy minimization to an optimal transport (OT) problem to generate pseudo-labels for learning image representations. This OT problem can be regarded as seeking the solution of transporting the sample distribution to the class distribution. However, the class distribution is unknown. Although Asano et al. (2020) sets it to a uniform distribution to avoid degenerate solutions, the mismatched class

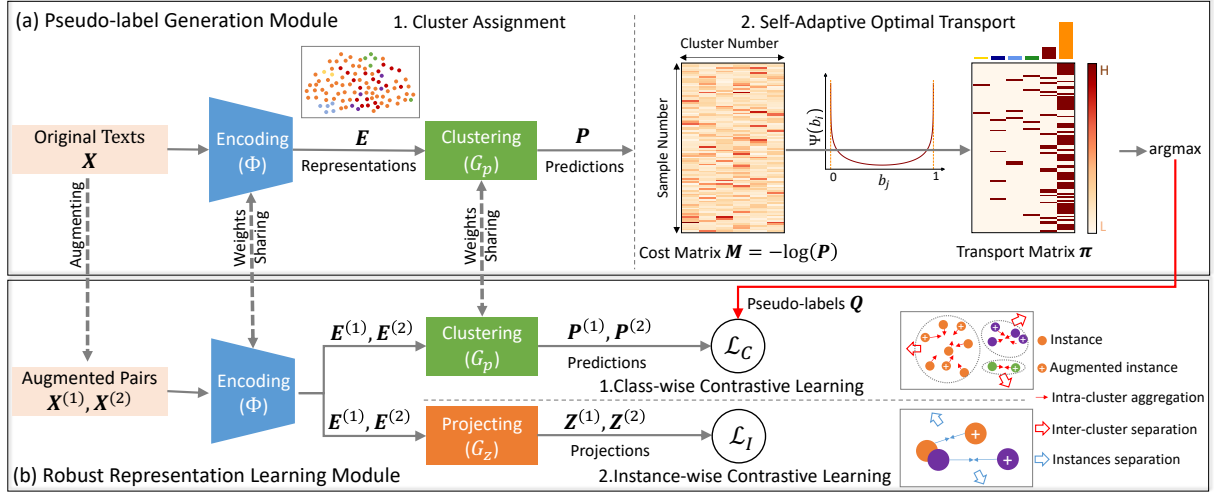


Figure 1: An overview of **RSTC**, which contains two modules: (a) Pseudo-label Generation Module, and (b) Robust Representation Learning Module.

distribution will lead to unreliable pseudo-labels. Therefore, it is essential to estimate real class distribution for addressing this issue. The recent research (Wang et al., 2022) studies the class distribution estimation, but it tends to cause clustering degeneracy on heavy imbalanced data, which we will further discuss in Appendix A. Hence, to discover reliable pseudo-labels on various imbalanced data, we propose SAOT.

We will provide the details of SAOT below. We expect to minimize the cross entropy loss to generate the pseudo-labels by solving a discrete OT problem. Specifically, we denote the pseudo-labels as $Q \in \mathbb{R}^{N \times C}$. Let $\pi = \frac{1}{N}Q$ be the transport matrix between samples and classes, $M = -\log P$ be the cost matrix to move probability mass from samples to classes. The reason that we use $\frac{1}{N}$ between π and Q is the transport matrix should be a joint probability (Cuturi, 2013), i.e., the sum of all values in the π should be 1, while the sum of each row in Q is 1. We have, $Q^* = \underset{Q}{\operatorname{argmin}} \langle Q, -\log P \rangle = N \underset{\pi}{\operatorname{argmin}} \langle \pi, M \rangle$. Thus, the OT problem is as follows:

$$\begin{aligned} \min_{\pi} \langle \pi, M \rangle + \epsilon H(\pi) \\ \text{s.t. } \pi \mathbf{1} = \mathbf{a}, \pi^T \mathbf{1} = \mathbf{b}, \pi \geq 0, \end{aligned} \quad (1)$$

where ϵ is a balance hyper parameter, $H(\pi) = \langle \pi, \log \pi - 1 \rangle$ is the entropy regularization (Cuturi, 2013), $\mathbf{a} = \frac{1}{N} \mathbf{1}$ is the sample distribution, and \mathbf{b} is an unknown class distribution. To avoid clustering degeneracy and obtain reliable transport matrix with randomly initialized \mathbf{b} , we introduce a penalty function about \mathbf{b} to the OT objective and update \mathbf{b}

during the process of solving the transport matrix. We formulate the SAOT optimization problem as:

$$\begin{aligned} \min_{\pi, \mathbf{b}} \langle \pi, M \rangle + \epsilon_1 H(\pi) + \epsilon_2 (\Psi(\mathbf{b}))^T \mathbf{1} \\ \text{s.t. } \pi \mathbf{1} = \mathbf{a}, \pi^T \mathbf{1} = \mathbf{b}, \pi \geq 0, \mathbf{b}^T \mathbf{1} = 1, \end{aligned} \quad (2)$$

where ϵ_1 and ϵ_2 are balance hyper-parameters, $\Psi(\mathbf{b}) = -\log \mathbf{b} - \log(1 - \mathbf{b})$ is the penalty function about \mathbf{b} . The penalty function not only limits b_j (a value of \mathbf{b} ranges from 0 to 1, but also avoids clustering degeneracy by encouraging \mathbf{b} to be a uniform distribution. The encouragement is achieved by increasing the punishment for b_j that is close to 0 or 1. Besides, the level of the encouragement can be adjusted by ϵ_2 . Specifically, there are two critical terms in Equation (2) for exploring \mathbf{b} , i.e., (1) the cost matrix M and (2) the penalty function $\Psi(\mathbf{b})$, and we use ϵ_2 to balance these two terms. For balanced data, both M and $\Psi(\mathbf{b})$ encourage \mathbf{b} to be a uniform distribution. For imbalanced data, M encourages the head clusters (i.e., the clusters with a large proportion of instances) to have larger b_j and the tail clusters (i.e., the clusters with a small proportion of instances) to have smaller b_j . When b_j of a tail cluster approaches 0, this tail cluster tends to disappear (clustering degeneracy). Whereas $\Psi(\mathbf{b})$ still encourages \mathbf{b} to be a uniform distribution for avoiding the degeneracy. With a decent trade-off parameter ϵ_2 , SAOT can explore appropriate \mathbf{b} and obtain reliable π for various imbalanced data. We provide the optimization details in Appendix B. After obtaining π , we can get pseudo-labels by

argmax operation, i.e.,

$$\mathbf{Q}_{ij} = \begin{cases} 1, & \text{if } j = \operatorname{argmax}_{j'} \pi_{ij'} \\ 0, & \text{otherwise.} \end{cases} \quad (3)$$

It should be noted that, for convenience, we let $\pi = \frac{1}{N}\mathbf{Q}$ before. However, π is essentially a joint probability matrix and π_{ij} can be decimals, while each row of \mathbf{Q} is a one-hot vector.

Through the steps of cluster assignment and self-adaptive optimal transport, we can generate reliable pseudo-labels on various imbalanced data for the robust representation learning module.

3.3 Robust Representation Learning module

We then introduce the robust representation learning module. To begin with, motivated by (Wenzel et al., 2022), we propose to adopt instance augmentations to improve the model robustness against various noises. Furthermore, inspired by (Chen et al., 2020), (Zhang et al., 2021) and (Dong et al., 2022), we adopt both class-wise and instance-wise contrastive learning to utilize the pseudo-labels and the augmented instance pairs for robust representation learning, as shown in Fig.1(b). The class-wise contrastive learning uses pseudo-labels as the supervision to pull the representations from the same cluster together and push away different clusters. While the instance-wise contrastive learning disperses different instances apart, which is supposed to separate the overlapped clusters.

Next, we provide the details of the robust representation learning module. We utilize contextual augmenter (Kobayashi, 2018; Ma, 2019) to generate augmented pairs of the original texts as $\mathbf{X}^{(1)}$ and $\mathbf{X}^{(2)}$. Like the cluster assignment step in the pseudo-labels generation module, we can obtain the representations of augmented pairs $\mathbf{X}^{(1)}$ and $\mathbf{X}^{(2)}$ as $\mathbf{E}^{(1)} \in \mathbb{R}^{N \times D_1}$ and $\mathbf{E}^{(2)} \in \mathbb{R}^{N \times D_1}$, respectively. We can obtain the predictions of them as $\mathbf{P}^{(1)} \in \mathbb{R}^{N \times C}$ and $\mathbf{P}^{(2)} \in \mathbb{R}^{N \times C}$, respectively. We use the fully connected layers as the projecting network G_z to map the representations to the space where instance-wise contrastive loss is applied, i.e., $G_z(\mathbf{E}^{(1)}) = \mathbf{Z}^{(1)} \in \mathbb{R}^{N \times D_2}$ and $G_z(\mathbf{E}^{(2)}) = \mathbf{Z}^{(2)} \in \mathbb{R}^{N \times D_2}$, where D_2 is the dimension of the projected representations. The encoding network and the clustering network share weights with the pseudo-label generation module.

The class-wise contrastive learning enforces consistency between cluster predictions of positive

pairs. Specifically, the two augmentations from the same original text are regarded as a positive pair and the contrastive task is defined on pairs of augmented texts. Moreover, the pseudo-label of an original text is considered as the target of corresponding two augmented texts. We use the augmented texts with the targets as supervised data for cross-entropy minimization to achieve the consistency. The class-wise contrastive loss is defined as below:

$$\mathcal{L}_C = \frac{1}{N} \langle \mathbf{Q}, -\log \mathbf{P}^{(1)} \rangle + \frac{1}{N} \langle \mathbf{Q}, -\log \mathbf{P}^{(2)} \rangle. \quad (4)$$

The instance-wise contrastive learning enforces consistency between projected representations of positive pairs while maximizing the distance between negative pairs. Specifically, for a batch, there are $2N$ augmented texts, their projected representations are $\mathbf{Z} = [\mathbf{Z}^{(1)}, \mathbf{Z}^{(2)}]^T$, given a positive pair with two texts which are augmented from the same original text, the other $2(N-1)$ augmented texts are treated as negative samples. The loss for a positive pair (i, j) is defined as:

$$\ell(i, j) = -\log \frac{\exp(\operatorname{sim}(\mathbf{Z}_i, \mathbf{Z}_j)/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{k \neq i} \exp(\operatorname{sim}(\mathbf{Z}_i, \mathbf{Z}_k)/\tau)}, \quad (5)$$

where $\operatorname{sim}(\mathbf{u}, \mathbf{v})$ denotes cosine similarity between \mathbf{u} and \mathbf{v} , τ denotes the temperature parameter, and $\mathbb{1}$ is an indicator. The instance-wise contrastive loss is computed across all positive pairs in a batch, including both (i, j) and (j, i) . That is,

$$\mathcal{L}_I = \frac{1}{2N} \sum_{i=1}^N (\ell(i, 2i) + \ell(2i, i)). \quad (6)$$

By combining the pseudo-supervised class-wise contrastive learning and the instance-wise contrastive learning, we can obtain robust representations and correctly separated clusters.

3.3.1 Putting Together

The total loss of **RSTC** could be obtained by combining the pseudo-supervised class-wise contrastive loss and the instance-wise contrastive loss. That is, the loss of **RSTC** is given as:

$$\mathcal{L} = \mathcal{L}_C + \lambda_I \mathcal{L}_I, \quad (7)$$

where λ_I is a hyper-parameter to balance the two losses. By doing this, **RSTC** not only provides robustness to the imbalance in data, but also improve robustness against the noise in data.

The whole model with two modules forms a closed loop and self evolution, which indicates that the learned representations (more robust) and cluster predictions (more accurate) elevate each other progressively, as more reliable pseudo-labels are discovered during the iterations. Specifically, we firstly initialize the pseudo-labels Q by performing k-means on text representations. Next, we train the robust representation learning module by batch with the supervision of pseudo-labels. Meanwhile, we update Q throughout the whole training process in a logarithmic distribution, following (Asano et al., 2020). Finally, we can obtain the cluster assignments by the column index of the largest entry in each row of P . The training stops if the change of cluster assignments between two consecutive updates for P is less than a threshold δ or the maximum number of iterations is reached.

4 Experiment

In this section, we conduct experiments on several real-world datasets to answer the following questions: (1) **RQ1**: How does our approach perform compared with the state-of-the-art short text clustering methods? (2) **RQ2**: How do the SAOT, and the two contrastive losses contribute to the performance improvement? (3) **RQ3**: How does the performance of **RSTC** vary with different values of the hyper-parameters?

4.1 Datasets

We conduct extensive experiments on eight popularly used real-world datasets, i.e., **AgNews**, **StackOverflow**, **Biomedical**, **SearchSnippets**, **GoogleNews-TS**, **GoogleNews-T**, **GoogleNews-S** and **Tweet**. Among them, **AgNews**, **StackOverflow** and **Biomedical** are balanced datasets, **SearchSnippets** is a light imbalanced dataset, **GoogleNews**, **GoogleNews-T**, **GoogleNews-S** and **Tweet** are heavy imbalanced datasets. Following (Zhang et al., 2021), we take unprocessed data as input to demonstrate that our model is robust to noise, for a fair comparison. More details about the datasets are shown in Appendix C.1.

4.2 Experiment Settings

We build our model with PyTorch (Paszke et al., 2019) and train it using the Adam optimizer (Kingma and Ba, 2015). We study the effect of hyper-parameters ϵ_1 and ϵ_2 on SAOT by varying them in $\{0.05, 0.1, 0.2, 0.5\}$ and

$\{0, 0.001, 0.01, 0.1, 1\}$, respectively. Besides, we study the effect of the hyper-parameter λ_I by varying it in $\{0, 1, 5, 10, 20, 50, 100\}$. The more details are provided in Appendix C.2. Following previous work (Xu et al., 2017; Hadifar et al., 2019; Rakib et al., 2020; Zhang et al., 2021), we set the cluster numbers to the ground-truth category numbers, and we adopt Accuracy (ACC) and Normalized Mutual Information (NMI) to evaluate different approaches. The specific definitions of the evaluation methods are shown in Appendix C.3. For all the experiments, we repeat five times and report the average results.

4.3 Baselines

We compare our proposed approach with the following short text clustering methods. **BOW** (Scott and Matwin, 1998) & **TF-IDF** (Salton and McGill, 1983) applies k-means on the TF-IDF representations and BOW representations respectively. **STC²-LPI** (Xu et al., 2017) first uses word2vec to train word embeddings on the in-domain corpus, and then uses a convolutional neural network to obtain the text representations where k-means is applied for clustering. **Self-Train** (Hadifar et al., 2019) follows (Xie et al., 2016) uses an autoencoder to get the representations, and finetunes the encoding network with the same clustering objective. The difference are that it uses the word embeddings provided by (Xu et al., 2017) with SIF (Arora et al., 2017) to enhance the pretrained word embeddings, and obtains the final cluster assignments via k-means. **K-means_IC** (Rakib et al., 2020) first applies k-means on the TF-IDF representations and then enhances clustering by the iterative classification algorithm. **SCCL** (Zhang et al., 2021) is the more recent short text clustering model which utilizes SBERT (Reimers and Gurevych, 2019) as the backbone and introduces instance-wise contrastive learning to support clustering. Besides, **SCCL** uses the clustering objective proposed in (Xie et al., 2016) for deep joint clustering and obtains the final cluster assignments by k-means.

4.4 Clustering Performance (RQ1)

Results and discussion The comparison results on eight datasets are shown in Table 1. **SBERT(k-means)** denotes the pre-trained SBERT model with k-means clustering, which is the initial state of our **RSTC**.

From the results, we can find that: (1) Only adopting traditional text representations (**BOW** and

	AgNews		SearchSnippets		Stackoverflow		Biomedical	
	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI
BOW	28.71	4.07	23.67	9.00	17.92	13.21	14.18	8.51
TF-IDF	34.39	12.19	30.85	18.67	58.52	59.02	29.13	25.12
STC ² -LPI	-	-	76.98	62.56	51.14	49.10	43.37	38.02
Self-Train	-	-	72.69	56.74	59.38	52.81	40.06	34.46
K-means_IC	66.30	42.03	63.84	42.77	<u>74.96</u>	<u>70.27</u>	40.44	32.16
SCCL	<u>83.10</u>	<u>61.96</u>	<u>79.90</u>	<u>63.78</u>	70.83	69.21	42.49	<u>39.16</u>
SBERT(k-means)	65.95	31.55	55.83	32.07	60.55	51.79	39.50	32.63
RSTC-OT	65.94	41.86	70.79	59.30	56.77	60.17	38.14	34.89
RSTC-C	78.08	49.39	62.59	44.02	78.33	70.28	46.74	38.69
RSTC-I	85.39	62.79	79.26	68.03	31.31	28.66	34.39	31.20
RSTC	84.24	62.45	80.10	69.74	83.30	74.11	48.40	40.12
Improvement(↑)	1.14	0.49	0.20	5.96	8.34	3.84	5.03	0.96
	GoogleNews-TS		GoogleNews-T		GoogleNews-S		Tweet	
	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI
BOW	58.79	82.59	48.05	72.38	52.68	76.11	50.25	72.00
TF-IDF	69.00	87.78	58.36	79.14	62.30	83.00	54.34	78.47
K-means_IC	79.81	92.91	68.88	83.55	<u>74.48</u>	<u>88.53</u>	66.54	84.84
SCCL	<u>82.51</u>	<u>93.01</u>	<u>69.01</u>	<u>85.10</u>	73.44	87.98	<u>73.10</u>	<u>86.66</u>
SBERT(k-means)	65.71	86.60	55.53	78.38	56.62	80.50	53.44	78.99
RSTC-OT	63.97	85.79	56.45	79.49	59.48	81.21	56.84	79.16
RSTC-C	78.48	90.59	63.08	82.16	65.05	83.88	76.62	85.61
RSTC-I	75.44	92.06	64.84	85.06	66.22	86.93	61.12	84.53
RSTC	83.27	93.15	72.27	87.39	79.32	89.40	75.20	87.35
Improvement(↑)	0.76	0.14	3.26	2.29	4.84	0.87	2.1	0.69

Table 1: Experimental results on eight short text datasets. We bold the **best result**, underline the runner-up.

TF-IDF) cannot obtain satisfying results due to the data sparsity problem. (2) Deep learning methods (**STC²-LPI** and **Self-Train**) outperform traditional ones, indicating that the application of pre-trained word embeddings and deep neural network alleviates the sparsity problem. (3) **SCCL** obtains better results by introducing instance-wise contrastive learning to cope with the noises problem. However, the clustering objective of **SCCL** is prone to obtain degenerate solutions (Yang et al., 2017; Ji et al., 2019). Besides, it is suboptimal for extra application of k-means (Van Gansbeke et al., 2020). The degeneracy gives the representation learning wrong guidance, which degrades the final k-means clustering performance. (4) **RSTC** outperforms all baselines, which proves that the robust representation learning with pseudo-supervised class-wise contrastive learning and instance-wise contrastive learning can significantly improve the clustering performance. To better show the clustering degeneracy problem, we visualize how the number of predicted clusters are changing over iterations on **SCCL** and **RSTC** in Appendix C.4. From the results, we verify that **SCCL** has relatively serious clustering degeneracy problem while **RSTC** solves it. The visualization results illustrate the validity of our model.

4.5 In-depth Analysis (RQ2 and RQ3)

Ablation (RQ2) To study how each component of **RSTC** contribute to the final performance, we compare **RSTC** with its several variants, including **RSTC-OT**, **RSTC-C** and **RSTC-I**. **RSTC-OT** adopts both the pseudo-supervised class-wise contrastive learning and instance-wise contrastive learning, while the pseudo-labels are generated by traditional OT with fixed random class distribution. **RSTC-C** only adopts the pseudo-supervised class-wise contrastive learning, the pseudo-labels are generated by SAOT. **RSTC-I** only adopts the instance-wise contrastive learning while the clustering results are obtained by k-means.

The comparison results are shown in Table 1. From it, we can observe that they all cannot achieve satisfactory results due to their limitations. Specifically, (1) **RSTC-OT** will be guided by the mismatched distribution constraint to generate unreliable pseudo-labels. (2) **RSTC-C** is good at aggregating instances, but it has difficulties to address the situation when different categories are overlapped with each other in the representation space at the beginning of the learning progress, which may lead to a false division. (3) **RSTC-I** is good at dispersing different instances, but it has limited ability to aggregate instances which

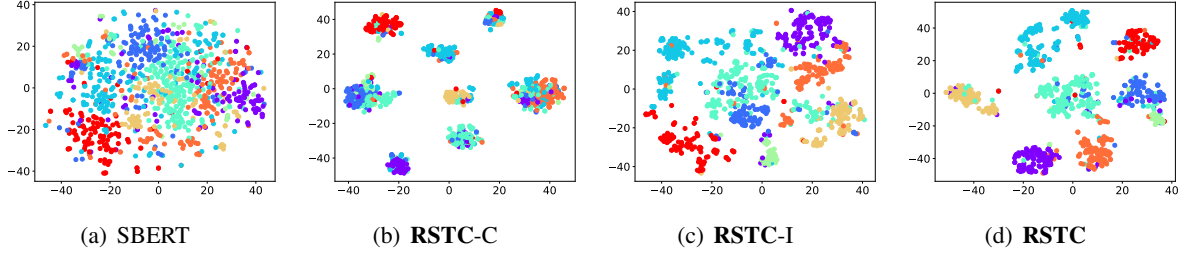


Figure 2: TSNE visualization of the representations on SearchSnippets, each color indicates a ground truth category.

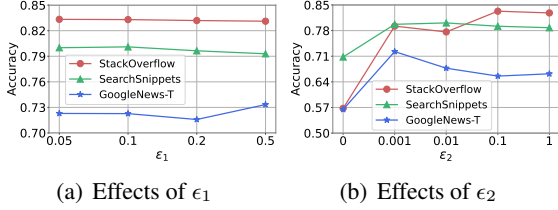


Figure 3: The effect of ϵ_1 and ϵ_2 on model accuracy.

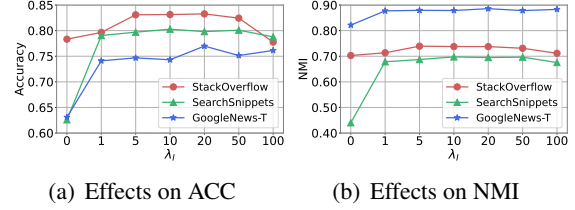


Figure 4: The effect of λ_I on model performance.

may lead to the unclear boundaries between clusters. (4) **RSTC** achieves the best performance with the combination of pseudo-supervised class-wise contrastive learning and instance-wise contrastive learning while the pseudo-labels are generated by SAOT. Overall, the above ablation study demonstrates that our proposed SAOT and robust representation learning are effective in solving the short text clustering problem.

Visualization To further show the functions and the limitations of the pseudo-supervised class-wise contrastive learning and the instance-wise contrastive learning, we visualize the representations using t-SNE (Van der Maaten and Hinton, 2008) for **SBERT** (initial state), **RSTC-C**, **RSTC-I** and **RSTC**. The results on **SearchSnippets** are shown in Fig.2(a)-(d). From it, we can see that: (1) **SBERT** (initial state) has no boundaries between classes, and the points from different clusters have significant overlap. (2) Although **RSTC-C** groups the representations to exact eight clusters, a large proportion of points are clustered by mistake. (3) **RSTC-I** disperses the overlapped categories to some extent, but the points are not clustered. (4) With the combination of **RSTC-C** and **RSTC-I**, **RSTC** obtains best text representations with small intra-cluster distance, large inter-cluster distance while only a small amount of points are clustered wrongly. The reasons for these phenomenons are the same as previous results analyzed in **Ablation**.

Effect of hyper-parameter (RQ3) We now study the effects of hyper-parameters on model

performance, including ϵ_1 , ϵ_2 and λ_I . We first study the effects of ϵ_1 and ϵ_2 by varying them in $\{0.05, 0.1, 0.2, 0.5\}$ and $\{0, 0.001, 0.01, 0.1, 1\}$, respectively. The results are reported in Fig. 3(a) and Fig. 3(b). Fig. 3(a) shows that the accuracy are not sensitive to ϵ_1 . Fig. 3(b) shows that choosing the proper hyper-parameters for different imbalance levels of datasets is important, especially on the heavy imbalanced dataset **GoogleNews-T**. Empirically, we choose $\epsilon_1 = 0.1$ on all datasets, $\epsilon_2 = 0.1$ on the balanced datasets, $\epsilon_2 = 0.01$ on the light imbalanced datasets, and $\epsilon_2 = 0.001$ on the heavy imbalanced datasets. Then we perform experiments by varying λ_I in $\{0, 1, 5, 10, 20, 50, 100\}$. The results on three datasets are shown in Fig. 4. From them, we can see that the performance improves when λ_I increases, then keeps a relatively stable level after λ_I reaches 1 and finally decreases when λ_I becomes too large. We can conclude that when λ_I is too small, the ability of instance-wise contrastive learning cannot be fully exploited. When λ_I is too large, the ability of class-wise contrastive learning will be suppressed, which also reduces the clustering performance. Empirically, we choose $\lambda_I = 10$ for all datasets.

5 Conclusion

In this paper, we propose a robust short text clustering (**RSTC**) model, which includes *pseudo-label generation module* and *robust representation learning module*. The former generates pseudo-labels as the supervision for the latter. We innovatively propose SAOT in the pseudo-label generation mod-

ule to provide robustness against the imbalance in data. We further propose to combine class-wise contrastive learning with instance-wise contrastive learning in the robust representation learning module to provide robustness against the noise in data. Extensive experiments conducted on eight real-world datasets demonstrate the superior performance of our proposed **RSTC**.

6 Limitations

Like existing short text clustering methods, we assume the real cluster number is known. In the future, we would like to explore a short text clustering method with an unknown number of clusters. Moreover, the time complexity of self-adaptive optimal transport is $O(n^2)$, we are going to seek a new computation to reduce the complexity.

Acknowledgements

This work was supported in part by the Leading Expert of “Ten Thousands Talent Program” of Zhejiang Province (No.2021R52001) and National Natural Science Foundation of China (No.72192823).

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*.
- Yuki M. Asano, Christian Rupprecht, and Andrea Vedaldi. 2020. Self-labelling via simultaneous clustering and representation learning. In *International Conference on Learning Representations (ICLR)*.
- Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. 2018. Deep clustering for unsupervised learning of visual features. In *Proceedings of the European conference on computer vision (ECCV)*, pages 132–149.
- Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. 2020. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in Neural Information Processing Systems*, 33:9912–9924.
- 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*, pages 1597–1607. PMLR.
- Marco Cuturi. 2013. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems*, 26.
- Bo Dong, Yiyi Wang, Hanbo Sun, Yunji Wang, Alireza Hashemi, and Zheng Du. 2022. Cml: A contrastive meta learning method to estimate human label confidence scores and reduce data collection cost. In *Proceedings of The Fifth Workshop on e-Commerce and NLP (ECNLP 5)*, pages 35–43.
- Yaakov HaCohen-Kerner, Daniel Miller, and Yair Yigal. 2020. The influence of preprocessing on text classification using a bag-of-words representation. *PLoS one*, 15(5):e0232525.
- 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)*, pages 194–199.
- Weibo Hu, Chuan Chen, Fanghua Ye, Zibin Zheng, and Yunfei Du. 2021. Learning deep discriminative representations with pseudo supervision for image clustering. *Information Sciences*, 568:199–215.
- Xu Ji, Joao F Henriques, and Andrea Vedaldi. 2019. Invariant information clustering for unsupervised image classification and segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9865–9874.
- Diederik P. Kingma and Jimmy Ba. 2015. [Adam: A method for stochastic optimization](#). In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.
- Sosuke Kobayashi. 2018. [Contextual augmentation: Data augmentation by words with paradigmatic relations](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 2 (Short Papers)*, pages 452–457. Association for Computational Linguistics.
- Jinning Li, Huajie Shao, Dachun Sun, Ruijie Wang, Yuchen Yan, Jinyang Li, Shengzhong Liu, Hanghang Tong, and Tarek Abdelzaher. 2022. Unsupervised belief representation learning with information-theoretic variational graph auto-encoders. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1728–1738.
- Weiming Liu, Jiajie Su, Chaochao Chen, and Xiaolin Zheng. 2021. Leveraging distribution alignment via stein path for cross-domain cold-start recommendation. *Advances in Neural Information Processing Systems*, 34:19223–19234.
- Weiming Liu, Xiaolin Zheng, Mengling Hu, and Chaochao Chen. 2022a. Collaborative filtering with attribution alignment for review-based non-overlapped cross domain recommendation. In *Proceedings of the ACM Web Conference 2022*, pages 1181–1190.

- Weiming Liu, Xiaolin Zheng, Jiajie Su, Mengling Hu, Yanchao Tan, and Chaochao Chen. 2022b. Exploiting variational domain-invariant user embedding for partially overlapped cross domain recommendation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 312–321.
- Edward Ma. 2019. Nlp augmentation. <https://github.com/makcedward/nlpaug>.
- Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26.
- Christos H Papadimitriou and Kenneth Steiglitz. 1998. *Combinatorial optimization: algorithms and complexity*. Courier Corporation.
- Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. *Pytorch: An imperative style, high-performance deep learning library*. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.
- F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.
- Gabriel Peyré, Marco Cuturi, et al. 2019. Computational optimal transport: With applications to data science. *Foundations and Trends® in Machine Learning*, 11(5-6):355–607.
- 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*, pages 91–100.
- Md Rashadul Hasan Rakib, Norbert Zeh, Magdalena Jankowska, and Evangelos Milios. 2020. Enhancement of short text clustering by iterative classification. In *International Conference on Applications of Natural Language to Information Systems*, pages 105–117. Springer.
- Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992.
- Gerard Salton and Michael J McGill. 1983. *Introduction to modern information retrieval*. mcgraw-hill.
- Sam Scott and Stan Matwin. 1998. Text classification using wordnet hypernyms. In *Usage of WordNet in natural language processing systems*.
- Stefan Stieglitz, Milad Mirbabaie, Björn Ross, and Christoph Neuberger. 2018. Social media analytics—challenges in topic discovery, data collection, and data preparation. *International journal of information management*, 39:156–168.
- Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of machine learning research*, 9(11).
- Wouter Van Gansbeke, Simon Vandenhende, Stamatios Georgoulis, Marc Proesmans, and Luc Van Gool. 2020. Scan: Learning to classify images without labels. In *European conference on computer vision*, pages 268–285. Springer.
- Haobo Wang, Mingxuan Xia, Yixuan Li, Yuren Mao, Lei Feng, Gang Chen, and Junbo Zhao. 2022. Solar: Sinkhorn label refinery for imbalanced partial-label learning. *NeurIPS*.
- Florian Wenzel, Andrea Dittadi, Peter Vincent Gehler, Carl-Johann Simon-Gabriel, Max Horn, Dominik Zietlow, David Kernert, Chris Russell, Thomas Brox, Bernt Schiele, et al. 2022. Assaying out-of-distribution generalization in transfer learning. *arXiv preprint arXiv:2207.09239*.
- Junyuan Xie, Ross Girshick, and Ali Farhadi. 2016. Unsupervised deep embedding for clustering analysis. In *International conference on machine learning*, pages 478–487. PMLR.
- Jiaming Xu, Bo Xu, Peng Wang, Suncong Zheng, Guanhua Tian, Jun Zhao, and Bo Xu. 2017. Self-taught convolutional neural networks for short text clustering. *Neural Networks*, 88:22–31.
- Bo Yang, Xiao Fu, Nicholas D Sidiropoulos, and Mingyi Hong. 2017. Towards k-means-friendly spaces: Simultaneous deep learning and clustering. In *international conference on machine learning*, pages 3861–3870. PMLR.
- Jianhua Yin and Jianyong Wang. 2014. A dirichlet multinomial mixture model-based approach for short text clustering. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 233–242.
- Jianhua Yin and Jianyong Wang. 2016. A model-based approach for text clustering with outlier detection. In *2016 IEEE 32nd International Conference on Data Engineering (ICDE)*, pages 625–636. IEEE.
- Dejiao Zhang, Feng Nan, Xiaokai Wei, Shang-Wen Li, Henghui Zhu, Kathleen R. McKeown, Ramesh Nallapati, Andrew O. Arnold, and Bing Xiang. 2021. *Supporting clustering with contrastive learning*. In

Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021, pages 5419–5430. Association for Computational Linguistics.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28.

A Different Class Distribution Estimation Methods

We have tried three class distribution estimation methods, including: (1) **M1(ours)**: The method proposed in our paper with the penalty function $\Psi(\mathbf{b}) = -\log(\mathbf{b}) - \log(1 - \mathbf{b})$, and \mathbf{b} can be updated during the process of solving the OT problem. (2) **M2**: The method proposed in (Wang et al., 2022) holds no assumption on \mathbf{b} , and \mathbf{b} can be updated by the model predicted results with moving-average mechanism, that is, $\mathbf{b} = \mu\hat{\mathbf{b}} + (1 - \mu)\gamma$, where μ is the moving-average parameter, $\hat{\mathbf{b}}$ is the last updated \mathbf{b} and $\gamma_j = \frac{1}{N} \sum_{i=1}^N \mathbb{1}(j = \arg \max \mathbf{P}_i)$. (3) **M3**: This method replaces the penalty function in our method with the common entropy regularization $\Psi(\mathbf{b}) = KL(\mathbf{b} \parallel \hat{\mathbf{b}})$, where $\hat{\mathbf{b}}$ is the last updated \mathbf{b} , and the current \mathbf{b} can be updated the same way our method does. Note that, the parameters of **M2** are following (Wang et al., 2022), the parameters of **M3** are the same as **M1(ours)**.

For comprehensive comparison, we conduct the experiments on one imbalanced dataset **GoogleNews-T** and one balanced dataset **StackOverflow** with randomly initialized \mathbf{b} for visualizing how the accuracy and the number of predicted clusters are changing over iterations. Moreover, except the update of \mathbf{b} , everything else about the experiments is the same for three methods. The results are shown in Fig.5(a)-(d). From them, we can find that: (1) For the imbalanced dataset, **M1(ours)** achieves the best accuracy and converges to the real category number, while other methods have clustering degeneracy problem. (2) For the balanced dataset, **M2** achieves best accuracy more quickly while **M1(ours)** catches up in the end, and all methods obtain real category number. Although **M3** can obtain good accuracy on the imbalanced dataset, it has the worst accuracy on the balanced dataset. In addition, although **M2** achieves good accuracy on the balanced dataset, it has the worst accuracy on the imbalanced dataset. Only **M1(ours)** achieves

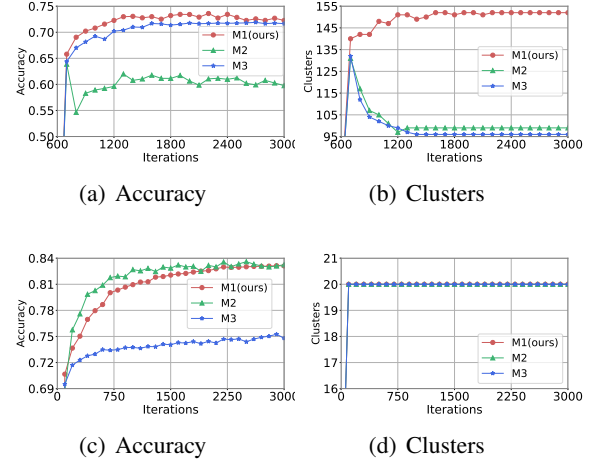


Figure 5: The accuracy and the number of predicted clusters at different iterations on GoogleNews-T (first row) and StackOverflow (second row). Note that because the samples in GoogleNews-T are too short, which makes it difficult to generate relatively reliable pseudo-labels, we pre-train the representations with \mathcal{L}_I for three methods in the first 600 steps. Due to the same pre-training process, we omit the curves in the first 600 steps on GoogleNews-T.

fairly good performance on both datasets, which indicates that our method are robust to various imbalance levels of datasets. The experiments prove the effectiveness of our class distribution estimation method.

B SAOT

As mentioned in Section 3.2, the SAOT problem is formulated as:

$$\begin{aligned} \min_{\pi, \mathbf{b}} \quad & \langle \pi, \mathbf{M} \rangle + \epsilon_1 H(\pi) + \epsilon_2 (\Psi(\mathbf{b}))^T \mathbf{1}, \\ \text{s.t.} \quad & \pi \mathbf{1} = \mathbf{a}, \pi^T \mathbf{1} = \mathbf{b}, \pi \geq 0, \mathbf{b}^T \mathbf{1} = 1. \end{aligned} \quad (8)$$

where ϵ_1 and ϵ_2 are balance hyper-parameters, $\Psi(\mathbf{b}) = -\log \mathbf{b} - \log(1 - \mathbf{b})$ is the penalty function about \mathbf{b} . We adopt the Lagrangian multiplier algorithm to optimize the problem:

$$\begin{aligned} \min_{\pi, \mathbf{b}} \quad & \langle \pi, \mathbf{M} \rangle + \epsilon_1 H(\pi) + \epsilon_2 (\Psi(\mathbf{b}))^T \mathbf{1} \\ & - \mathbf{f}^T (\pi \mathbf{1} - \mathbf{a}) - \mathbf{g}^T (\pi^T \mathbf{1} - \mathbf{b}) - h(\mathbf{b}^T \mathbf{1} - 1), \end{aligned} \quad (9)$$

where \mathbf{f} , \mathbf{g} , and h are all Lagrangian multipliers. Taking the differentiation of Equation (9) on the variable π , we can obtain:

$$\pi_{ij} = \exp\left(\frac{f_i + g_j - M_{ij}}{\epsilon_1}\right) > 0. \quad (10)$$

We first fix \mathbf{b} , due to the fact that $\pi \mathbf{1} = \mathbf{a}$ and $\pi^T \mathbf{1} = \mathbf{b}$, we can get:

$$\exp\left(\frac{f_i}{\epsilon_1}\right) = \frac{a_i}{\sum_j^C \exp\left(\frac{g_j - M_{ij}}{\epsilon_1}\right)}, \quad (11)$$

$$\exp\left(\frac{g_j}{\epsilon_1}\right) = \frac{b_j}{\sum_i^N \exp\left(\frac{f_i - M_{ij}}{\epsilon_1}\right)}. \quad (12)$$

Then we fix \mathbf{f} and \mathbf{g} , and update \mathbf{b} by:

$$\min_{\mathbf{b}} [\epsilon_2 (\Psi(\mathbf{b}))^T \mathbf{1} + \mathbf{g}^T \mathbf{b} - h(\mathbf{b}^T \mathbf{1} - 1)]. \quad (13)$$

Taking the differentiation of Equation (13) on the variable \mathbf{b} , we can obtain:

$$(g_j - h)b_j^2 - ((g_j - h) + 2\epsilon_2)b_j + \epsilon_2 = 0. \quad (14)$$

It is easy to get the discriminant of Equation (14)

$$\Delta_j = (g_j - h)^2 + 4\epsilon_2^2 > 0,$$

$$b_j(h) = \frac{(g_j - h + 2\epsilon_2) \pm \sqrt{\Delta_j}}{2(g_j - h)}. \quad (15)$$

Note that,

$$b_j(h) = \frac{((g_j - h) + 2\epsilon_2) + \sqrt{\Delta_j}}{2(g_j - h)} \geq 1. \quad (16)$$

Thus, we choose the following $b_j(h)$:

$$b_j(h) = \frac{((g_j - h) + 2\epsilon_2) - \sqrt{\Delta_j}}{2(g_j - h)}. \quad (17)$$

Taking Equation (17) back to the original constraint $\mathbf{b}^T \mathbf{1} = 1$, the formula is defined as below:

$$(\mathbf{b}(h))^T \mathbf{1} - 1 = 0, \quad (18)$$

where h is the root of Equation (18), and we can use Newton's method to work out it. Specifically, we first define that $f(h) = (\mathbf{b}(h))^T \mathbf{1} - 1$, then h can be updated by:

$$h \leftarrow h - \frac{f(h)}{f'(h)}, \quad (19)$$

where the iteration number is set to 10. Then we can obtain \mathbf{b} by Equation (17). In short, through iteratively updating Equation (11), (12), (19), and (17), we can obtain the transport matrix π on Equation (10). We show the iteration optimization scheme of SAOT in Algorithm 1.

Algorithm 1 The optimization scheme of SAOT

Input: The cost distance matrix: \mathbf{M} .

Output: The transport matrix: π .

Procedure:

- 1: Initialize \mathbf{f} and \mathbf{g} randomly;
 - 2: Initialize \mathbf{b} randomly and perform normalization so that $\mathbf{b}^T \mathbf{1} = 1$;
 - 3: Initialize $h = 1$.
 - 4: **for** $i = 1$ to T **do**
 - 5: Update \mathbf{f} in Equation (11);
 - 6: Update \mathbf{g} in Equation (12);
 - 7: Update \mathbf{b} in Equation (17) with the constraint $\mathbf{b}^T \mathbf{1} = 1$.
 - 8: **end for**
 - 9: Calculate π in Equation (10).
-

C Experiment

C.1 Datasets

We conduct extensive experiments on eight popularly used real-world datasets. The details of each dataset are as follows.

AgNews (Rakib et al., 2020) is a subset of AG's news corpus collected by (Zhang et al., 2015) which consists of 8,000 news titles in 4 topic categories. **StackOverflow** (Xu et al., 2017) consists of 20,000 question titles associated with 20 different tags, which is randomly selected from the challenge data published in Kaggle.com¹. **Biomedical** (Xu et al., 2017) is composed of 20,000 paper titles from 20 different topics and it is selected from the challenge data published in BioASQ's official website². **SearchSnippets** (Phan et al., 2008) contains 12,340 snippets from 8 different classes, which is selected from the results of web search transaction. **GoogleNews** (Yin and Wang, 2016) consists of the titles and snippets of 11,109 news articles about 152 events (Yin and Wang, 2014) which is divided into three datasets: the full dataset is **GoogleNews-TS**, while **GoogleNews-T** only contains titles and **GoogleNews-S** only has snippets. **Tweet** (Yin and Wang, 2016) consists of 2,472 tweets related to 89 queries and the original data is from the 2011 and 2012 microblog track at the Text REtrieval Conference³. The detailed statistics of these datasets are shown in Table 2.

¹<https://www.kaggle.com/c/predict-closed-questions-on-stack-overflow/>

²<http://participants-area.bioasq.org/>

³<https://trec.nist.gov/data/microblog.html>

Dataset	C	N	A	R
AgNews	4	8,000	23	1
StackOverflow	20	20,000	8	1
Biomedical	20	20,000	13	1
SearchSnippets	8	12,340	18	7
GoogleNews-TS	152	11,109	28	143
GoogleNews-T	152	11,108	6	143
GoogleNews-S	152	11,108	22	143
Tweet	89	2,472	9	249

Table 2: The statistics of the datasets. C means the number of classes, N means the dataset size, A is the average number of words per instance and R is the ratio of the largest cluster size to the smallest one.

C.2 Experiment Settings

We choose distilbert-base-nli-stsb-mean-tokens in Sentence Transformer library (Reimers and Gurevych, 2019) to encode the text, and the maximum input length is set to 32. The learning rate is set to 5×10^{-6} for optimizing the encoding network, and 5×10^{-4} for optimizing both the projecting network and clustering network. The dimensions of the text representations and the projected representations are set to $D_1 = 768$ and $D_2 = 128$, respectively. The batch size is set to $N = 200$. The temperature parameter is set to $\tau = 1$. The threshold δ is set to 0.01. The datasets specific tuning is avoided as much as possible. For **BOW** and **TF-IDF**, we achieved the code with scikit-learn (Pedregosa et al., 2011). For all the other baselines, i.e., **STC²-LPI⁴**, **Self-Train⁵**, **K-means_IC⁶**, and **SCCL⁷** (MIT-0 license), we used their released code. Besides, we substitute the accuracy evaluation code of **K-means_IC** with the evaluation method described in our paper.

In addition, as **STC²-LPI** and **Self-Train** use the word embeddings pre-trained with in-domain corpus, and there are only three datasets’ pre-trained word embeddings provided, therefore we do not report the results of other five datasets for them.

C.3 Evaluation Metrics

We report two widely used evaluation metrics of text clustering, i.e., accuracy (ACC) and normalized mutual information (NMI), following (Xu et al., 2017; Hadifar et al., 2019; Zhang et al., 2021).

⁴<https://github.com/jacoxu/STC2>

⁵https://github.com/hadifar/stc_clustering

⁶<https://github.com/rashadulrakib/short-text-clustering-enhancement>

⁷<https://github.com/amazon-science/sccl>

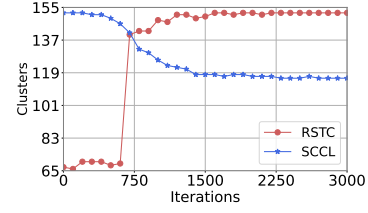


Figure 6: The number of predicted clusters at different iterations on GoogleNews-T. The steep rise of the *clusters* for **RSTC** is because we first adopt 600 steps of \mathcal{L}_I training and then adopt the training of both \mathcal{L}_C and \mathcal{L}_I , for a pre-training representations purpose.

Accuracy is defined as:

$$ACC = \frac{\sum_{i=1}^N \mathbb{1}_{y_i = \text{map}(\hat{y}_i)}}{N}, \quad (20)$$

where y_i and \hat{y}_i are the ground truth label and the predicted label for a given text x_i respectively, $\text{map}()$ maps each predicted label to the corresponding target label by Hungarian algorithm (Papadimitriou and Steiglitz, 1998). Normalized mutual information is defined as:

$$NMI(Y, \hat{Y}) = \frac{I(Y, \hat{Y})}{\sqrt{H(Y)H(\hat{Y})}}, \quad (21)$$

where Y and \hat{Y} are the ground truth labels and the predicted labels respectively, $I()$ is the mutual information, and $H()$ is the entropy.

C.4 Visualization

To better show the clustering degeneracy problem, we visualize how the number of predicted clusters (we call it *clusters* later) are changing over iterations on **SCCL** and **RSTC**. The results are shown in Fig.6. From it, we verify that **SCCL** has relatively serious clustering degeneracy problem while **RSTC** solves it to some extent. Specifically, the *clusters* of **SCCL** is much less than the real category number. Moreover, the degeneracy has a negative effect on the final k-means clustering performance because it makes the representations getting worse. Whereas the *clusters* of **RSTC** almost convergent to real category number, which assures the high accuracy of **RSTC**. The visualization results illustrate the validity of our model.

C.5 Computational Budget

The number of parameters in our model is 68M. Our training for each dataset takes about 10-30 minutes, using a GeForce RTX 3090 GPU.