Short Text Clustering

倪浚桐

2023.1

Short Text Clustering

- A Survey of Short Text Clustering
- Improved Supporting Clustering with Cluster-level Contrastive Learning

Introduction

Task Definition:
Automatically group multiple unlabeled short texts into a number of clusters

Source:
Social media (Twitter, Instagram, and Sina Weibo)

Downstream application:
Event exploration(What is happening around the world? When and where?), trend detection, and online user clustering

Challenge:
Sparseness

Related work

Similarity-based Clustering

Partitional algorithms: K-means

Hierarchical algorithms: HieClu

Density-based algorithms: OPTICS

Problem:

Classical and simple clustering approaches, relying on **hand-crafted features** for text clustering. However, for Kmeans and HieClu, **the number of clusters** need to be **specified** in advance, and they are **sensitive** to the **initialization.** OPTICS have limitations in handling **high-dimensional** data like text.

Related work

Topic model-based Clustering

Latent Dirichlet Allocation (LDA)

Probabilistic Latent Semantic Analysis (PLSA)

Biterm Topic Model (BTM)

Gibbs Sampling algorithm for the Dirichlet Multinomial Mixture model (GSDMM)

Problem:

They are **Bayesian topic models**, which realize probabilistic text clustering by assuming that each document is associated with a distribution over topics, and each topic is a distribution over words. In this way, a topic is usually regarded as a cluster. But, the input representation of short text is commonly based on the **bag-of-words assumption** and **one-hot encoding**, which might be **sparse** and **lack the expressive ability.**

Related work

Deep learning based Clustering

Self-Taught Convolutional NN for short text Clustering (STCC)

Gaussian-LDA

Deep Embedded Clustering (DEC)

Self-Training for short text Clustering (Self-Train)

Variational Deep Embedding (VaDE)

Problem:

The optimization process is partitioned into **separate stages**. Thus it is incapable of guiding text representation learning by the clustering objective.

Related work

Generative adversarial networks for Clustering

Deep Adversarial gaussian mixture auto-encoder for Clustering (DAC)

Cluster Generative Adversarial Network (ClusterGAN)

Attentive Representation Learning (ARL)

Problem:

Most of the GAN-based methods are welcomed for clustering **images** due to their ability to handle **continuous values**, whereas it is nontrivial for applying GAN-based methods to clustering the **discrete textual data** due to the difficulty of optimization. A surrogate way is to represent text as fixed continuous vectors (e.g., TF-IDF or pre-trained word embeddings) for GAN-based models. However, it might inevitably limit the power of learning text representations.

Datasets

Datasets	$\lvert V \rvert$	Documents		Clusters	
		N^D	Len	N^C	L/S
TREC	-	4434	3.3	128	-
GooglenewsTS	20K	11109	28	152	143
GooglenewsS	18K	11109	22	152	143
GooglenewsT	8K	11109	26	152	143
Event	-	26619	8.78	69	-
StackOverflow	15K	20000	8	20	1
TweetSet	5K	2472	8	89	249
SearchSnippets	15K	12340	18	8	7
Biomedical	19K	20000	13	20	1
UCLNews	-	10,379	6.7	30	-
AgNews	21K	8000	23	4	1

Table 1. Dataset statistics. $\lvert V \rvert$: the vocabulary size; N^D: number of short text documents; Len: average number of words in each document; N^C: number of clusters; L/S: the ratio of the size of the largest cluster to that of the smallest cluster.

Evaluate Metrics

Normalized mutual information (NMI)

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

Ajusted rand index (ARI)

$$ARI = \frac{\sum_{ij}\binom{\bar{v}_{ij}}{2} - \left[\sum_i \binom{v_i}{2} \sum_j \binom{\widetilde{v}_j}{2}\right]/n}{\frac{1}{2}\left[\sum_i \binom{v_i}{2} + \sum_j \binom{\widetilde{v}_j}{2}\right] - \left[\sum_i \binom{v_i}{2} \sum_j \binom{\widetilde{v}_j}{2}\right]/n}$$

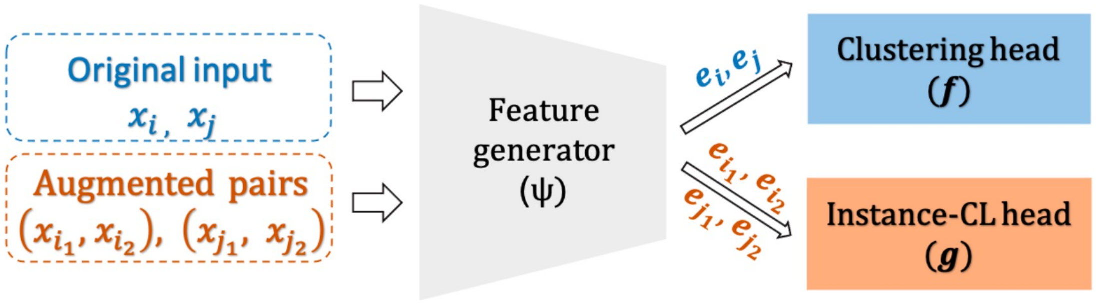
Accuracy (ACC)

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

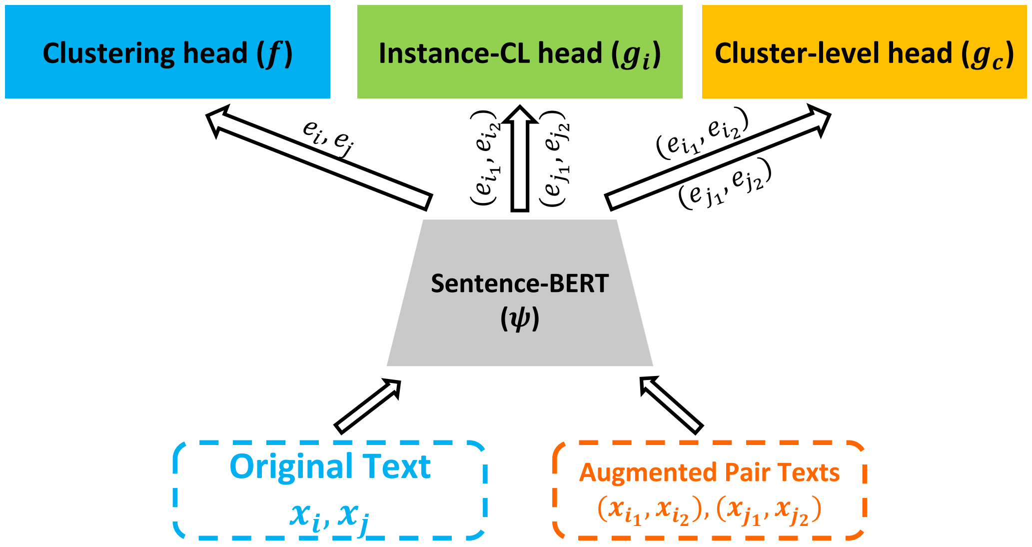
Short Text Clustering

- **A Survey of Short Text Clustering**
- **Improved Supporting Clustering with Cluster-level Contrastive Learning**

Improved Supporting Clustering with Cluster-level Contrastive Learning



Improve



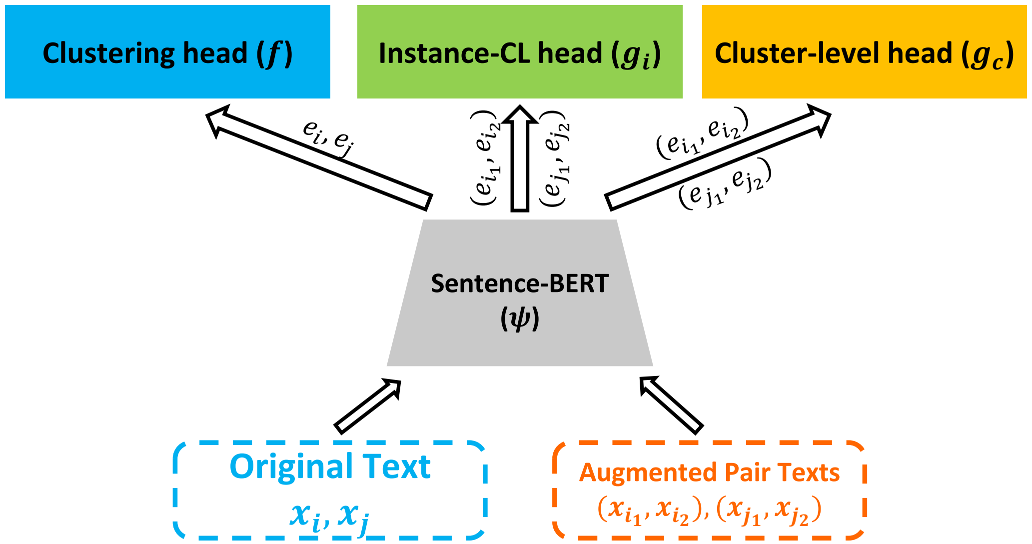
Supporting Clustering with Contrastive Learning (SCCl)

Improved Supporting Clustering with Cluster-level Contrastive Learning (ISCCl)

Instance-wise Contrastive Learning

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

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



Improved Supporting Clustering with Cluster-level Contrastive Learning (ISCCl)

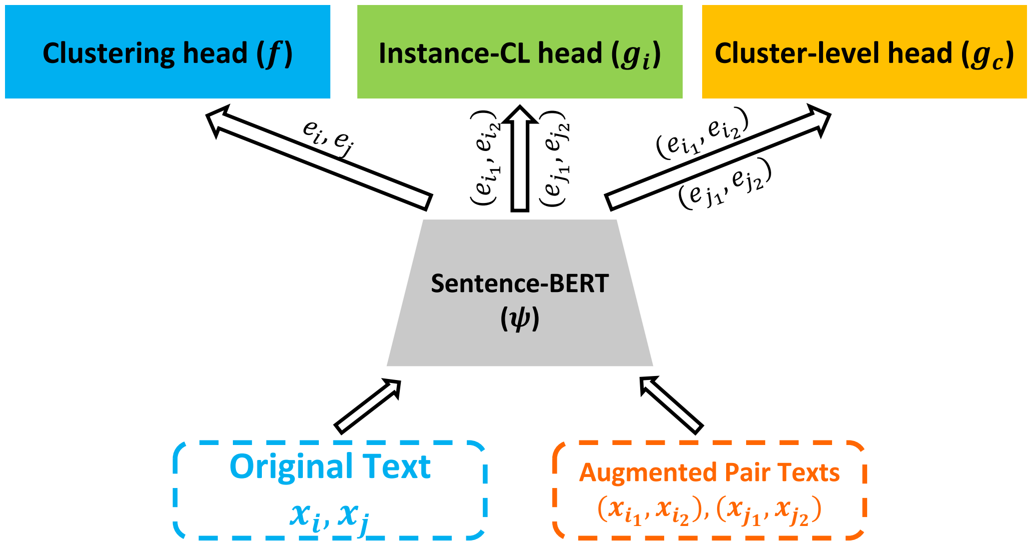
Clustering

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

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

$$\mathcal{L}_j^C = \mathrm{KL}[\mathrm{p_j}||\mathrm{q_j}] = \sum_{\mathrm{k}=1}^{\mathrm{K}} \mathrm{p_{jk}} \log \frac{\mathrm{p_{jk}}}{\mathrm{q_{jk}}}$$

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



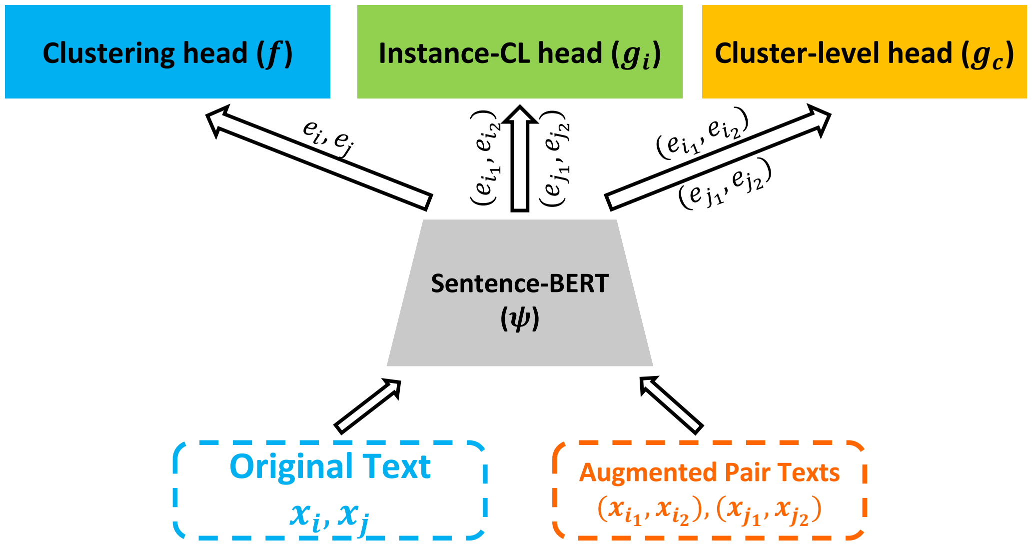
Improved Supporting Clustering with Cluster-level Contrastive Learning (ISCCl)

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).

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

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

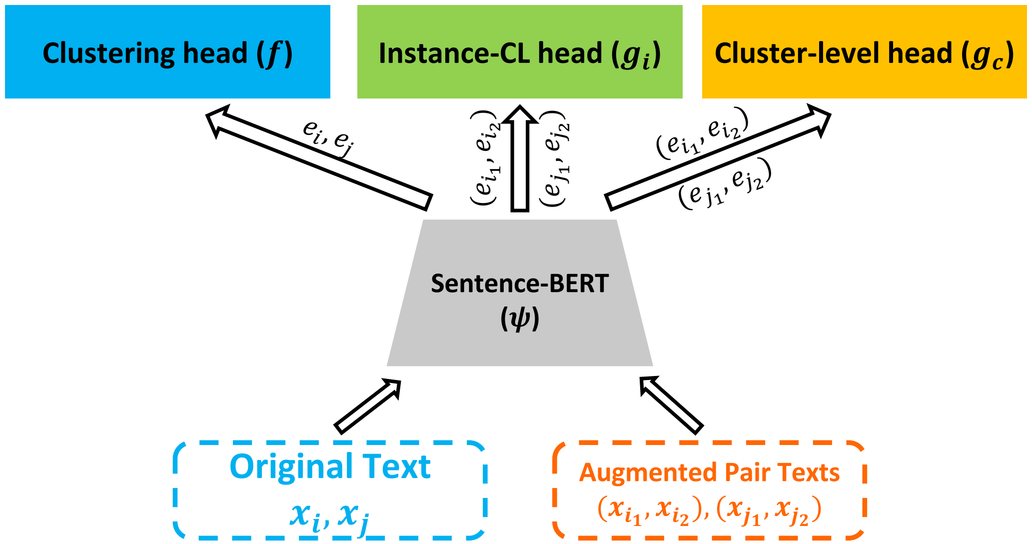


Improved Supporting Clustering with Cluster-level Contrastive Learning (ISCCl)

Overall objective

$$\mathcal{L} = \mathcal{L}_{Cluster} + \eta\mathcal{L}_{Instance-CL} + \eta\mathcal{L}_{Cluster-level\ CL}$$

$$= \sum_{j=1}^{M} \mathcal{L}_j^C / M + \eta \sum_{i=1}^{2M} \mathcal{L}_i^I / 2M + \eta \sum_{k=1}^{2C} \mathcal{L}_k^{CCL} / 2C$$



Improved Supporting Clustering with Cluster-level Contrastive Learning (ISCCl)

Data Preprocess

使用nlpaug库对original text进行了数据增强，分别通过BERT和RoBERT生成两段augmented text，采用替换策略，替换比例为20%

SearchSnippets

Original Text: turtlesoft goldenseal software reference banktran bank transactions goldenseal accounting software business

Augmented Text1: turtlesoft development software reference banktran bank transactions information accounting software store

Augmented Text2: turtlesoft goldenseal software reference manufacturing bank hardware industry accounting software business

Experiments

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.

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

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

SearchSnippets: 12,340 snippets, 8 groups

Evaluate Metrics: Normalized mutual information (NMI), Accuracy (ACC)

Source code: https://github.com/LingFengGold/ISCCL

Thank You!