Pattern Recognition 130 (2022) 108787

Contents lists available at ScienceDirect

Pattern Recognition

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

Contrastive author-aware text clustering

Xudong Tang[a], Chao Dong[b], Wei Zhang[a,1,*]

[a] School of Computer Science and Technology, KLATASDS-MOE, East China Normal University, No. 3663, North Zhongshan Road, Shanghai, China
[b] School of Software Engineering, East China Normal University, No. 3663, North Zhongshan Road, Shanghai, China

ARTICLE INFO

Article history:
Received 10 January 2022
Revised 16 April 2022
Accepted 11 May 2022
Available online 12 May 2022

Keywords:
Text clustering
Contrastive learning
Representation learning

ABSTRACT

In the era of User Generated Content (UGC), authors (IDs) of texts widely exist and play a key role in determining the topic categories of texts. Existing text clustering efforts are mainly attributed to utilizing textual information, but the effect of authors on text clustering remains largely underexplored. To mitigate this issue, we propose a novel Contrastive Author-aware Text clustering approach, dubbed as CAT. CAT injects author information not only in characterizing texts through representations but also in pushing or pulling text representations of different authors through contrastive learning, which is rarely adopted by text clustering. Specifically, the developed contrastive learning method conducts both cluster-instance contrast by the text representation augmentation and instance-instance contrast by the multi-view representations. We perform comprehensive experiments on three public datasets, demonstrating that CAT largely outperforms strong competitive text clustering baselines and validating the effectiveness of the CAT's main components.

1. Introduction

Text clustering is an important branch of natural language processing. It is formulated as finding groups of similar texts for a given text collection and has some successful applications. For example, it can be used to group a large amount of daily news into different topics, which reduces the burden of news organization and summarization. Another common application is to find user interest patterns based on the clustering of the texts they are interested in. This is a crucial step towards intelligent information filtering and recommendation. As such, considerable efforts have been devoted to this task [1,2].

Some empirical clustering methods rely on predefined strategies to derive clusters, such as hierarchical clustering, density-based clustering, grid-based clustering, and partition-based clustering, to name a few. Since they lack principled objective functions to optimize clustering methods, the clustering results might be sub-optimal. Therefore, model-based clustering approaches [3–6] are devised for handling textual data in recent years. Usually, they contain two fundamental components: text representations and clustering objectives. Some studies aim at building effective text representation schemes and investigating unsupervised objective functions for clustering.

However, existing clustering methods largely overlook the author effect of texts in determining text clusters. As the intriguing part of the era of User Generated Content (UGC), authors (IDs) of texts are commonly known to other users. For example, in community question answering platforms (e.g., StackOverflow, Quora, and Yahoo! Answers), authors pose textual questions, each of which usually has a title and a body. Another example is news feeds (e.g., TouTiao and News Break), where users can release different types of news that also have the title and body parts. Actually, each author focuses on a limited number of topic labels and generates topic-matched texts (see Table 1 and Fig. 2). Therefore, considering the author effect in text clustering enables us to obtain additional clues to reduce the number of candidate topics. Consequently, a key research question arises: how to incorporate the author information into text clustering models?

To address this question, we propose a novel Contrastive Author-aware Text clustering approach, dubbed as CAT. The main idea of CAT is to inject author information not only in characterizing texts through representations but also in pushing or pulling text representations through contrastive learning (CL). This is inspired by the real examples shown in Fig. 1. Take the text "Text#2" as an example. The author, title, and body could be regarded as different views of the text. Its title is a question about cleaning a white cotton chair and its body is about explaining the situation of the white cotton chair. It is obvious that the title and the body

* Corresponding author.
E-mail address: zhangwei.thu2011@gmail.com (W. Zhang).

[1] This work was supported in part by National Natural Science Foundation of China under Grant (No. 62072182) and the Fundamental Research Funds for the Central Universities.

https://doi.org/10.1016/j.patcog.2022.108787
0031-3203/

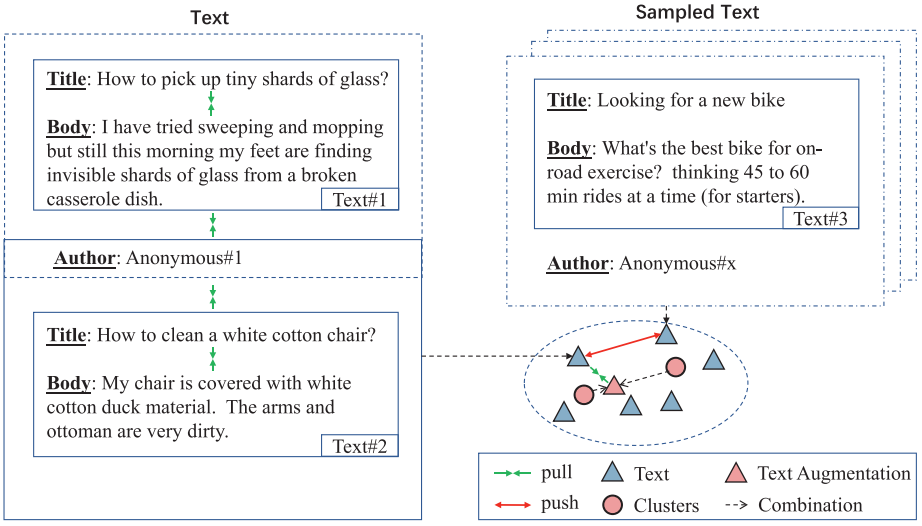

Fig. 1. Illustration of performing contrastive learning for text representations.

Table 1
Author topic coverage over the three datasets.

Dataset	SO	YA	TW
Topic Coverage	3.8%	4.7%	1.1%

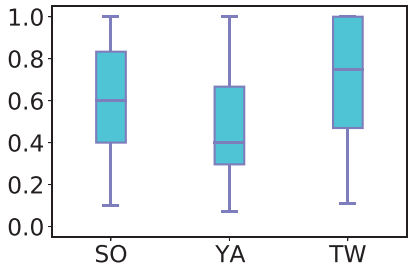

Fig. 2. Author topic concentration over the three datasets.

share a very similar semantic meaning. As such, the title and the body representations should be near to each other in the continuous space (**intuition 1**). Moreover, as analyzed in Section 3, an author in average has narrow topic coverage and high topic concentration. Thus an author could be characterized by a very few topics and is indicative of the topics of the author's texts. This motivates us to make the representations of the author and its texts (e.g., the author "Anonymous#1" and the text "Text#1" and the text "Text#2") to be near as well (**intuition 2**). Finally, for a given text representation, we can build a representation augmentation by referring to the cluster representations, as shown in the elliptic region of the figure. Intuitively, the original representation should be closer to the augmented representation than other text representations (**intuition 3**). Then the challenge is how to effectively encode these intuitions into the clustering model.

CAT addresses this by performing contrastive learning [7] with two novel contrast objectives, i.e., cluster-instance contrast and instance-instance contrast: (1) For cluster-instance contrast (**intuition 3**), CAT generates an augmentation of a given text representation by reconstructing its representation through similar cluster representations. Then cluster-instance contrast learning is performed on the given text representation, its cluster-based reconstructed representation (augmentation), and randomly sampled pseudo-negative text representations. (2) For instance-instance contrast, CAT devises author-text contrast learning (**intuition 2**) and title-body contrast learning (**intuition 1**) by regarding body, title, and author as different views of texts. On the one hand, author-text contrast learning is performed between the given text representation (only based on title and body) and the corresponding author representation. On the other hand, title-body contrast learning is for pushing a title representation and a body representation belonging to the same text, while pulling the representations from different texts. Thanks to the guidance of cluster-instance contrast and instance-instance contrast, CAT is trained in an end-to-end fashion. After CAT is well trained, the cluster assignment for each text is naturally determined by its similarities with the learned clusters.

It is noteworthy that there are a few methods leveraging contrastive learning for image and graph clustering [8–13]. However, their data augmentation techniques, such as cropping and clipping for images and dropping for graphs, are not suitable for the text modality due to the discrete property of word tokens. To our knowledge, there is only one contrastive learning study for text clustering [14]. It realizes text augmentation through synonym replacement based on pre-trained language models. However, this replacement strategy might change the original semantic category of a text. For example, the word 'Flex' in the text "I need help styling FormItem components in Flex" is replaced by 'CSS' based on Roberta [15], and thus its cluster label is changed. By contrast, this work performs text augmentation on the continuous representation space.

To sum up, we make the following main contributions:

- We show empirical evidence that authors have narrow topic coverage and large topic concentration on their generated texts, which justifies the rationality of our research motivation on considering the author effect in text clustering.
- To learn separable and cluster-friendly text representations, we devise the novel contrastive learning based text clustering model, named CAT, which injects the author effect in characterizing text representations.
- Both cluster-instance contrast and instance-instance contrast are developed to encode the three aforementioned intuitions and achieve comprehensive contrast learning, combining the idea of data augmentation and multi-view representations.
- Experiments conducted on three public datasets demonstrate the superiority of CAT compared with the strong competitive text clustering methods, validating the benefits of considering

the role of authors. Upon publication, we will release the model source code to facilitate relevant studies.

2. Related work

2.1. Text clustering

Achieving good text clustering [1,2] has been a long-standing goal and observed much progress in the past decade. The key to the success of text clustering lies in two aspects: text representation and objective function.

For the former aspect, conventional approaches rely on manual-crafted textual features such as TF-IDF [16] to represent a text. The feature-based representations could be fed into clustering algorithms (e.g., K-means[17]) to derive cluster assignments. However, original textual feature vectors are commonly sparse due to the large size of a vocabulary. As such, low-dimensional representations techniques are adopted to alleviate this issue, e.g., Principal Component Analysis (PCA) [18] and probabilistic topic models [19,20]. Particularly, autoencoders [4,5,21] utilize the strong nonlinear capacity of neural networks. However, none of the existing studies have considered the effect of authors on text representations for text clustering, which motivates this work to delve into this direction.

For the latter aspect, in contrast to some empirical clustering methods such as density-based clustering (e.g., DBSCAN [22]) and hierarchical clustering, many modern clustering methods have exploited different types of objective functions: (1) K-means tries to minimize the sum of the square error w.r.t. texts and their belonging cluster centers. (2) Benefiting from probability theory, topic models [3] maximize the likelihood of generating texts. (3) The studies [4,23] generate pseudo labels for each text by the soft assignment strategy and apply KL divergence as used in supervised learning. (4) Autoencoders and their variants [5,6,20] build loss functions according to text reconstruction errors. Unlike prior studies, this work devises contrastive objectives to learn text representations for clustering.

2.2. Contrastive learning

Due to the impressive performance of contrastive learning (CL) in many tasks, it has become a major technical roadmap in self-supervised learning. The core idea of CL is to maximize the similarity of positive pairs while minimizing the similarity of negative pairs [24]. SimCLR [25] presents a simple but effective CL based pre-training fashion for image data. Combining the advantage of large-scale pre-trained language model and CL, SimCSE [26] further improves the ability of sentence embeddings.

However, only a very few studies have been found for applying CL to clustering tasks. The studies [8,9] are mainly proposed for image data. In their CL methods, images are augmented based on clipping, cropping, etc. Nevertheless, these techniques are not suitable for text data due to their discrete property. To our knowledge, the only CL study for text clustering is SCCL [14]. It utilizes a pre-trained language model to perform text data augmentation by synonym replacement. However, as discussed previously, this replacement strategy might change the original cluster of a text. By contrast, this paper performs text representation augmentation in continuous representation space. Moreover, the idea of contrastive learning from the multi-review perspective [27–31] is also leveraged, by regarding body, title, and author as different views of texts.

2.3. Author-aware text mining

Since considering author information when performing text clustering is the major focus of this paper, we describe the following studies attributed to author-aware text mining.

Author-Aware Text Clustering. There are only a few studies that investigate the role of authors in text clustering. ATM [32] is a pioneering study in this regard by associating topics of words with prior topic distributions of authors. PDA-LDA [33] is an extension of ATM that further incorporates the effect of writing objects (e.g., products) into text topic generation. However, these methods are limited by the bag-of-words assumption and cannot leverage the power of representation learning, especially for word embeddings. Moreover, PDA-LDA is tailored for user-item connected documents (e.g., Amazon reviews) and its model involves not only user embeddings but also item embeddings. As such, it does not satisfy our problem setting that only authors of text need to be known.

Text-Aware Author Profiling. Instead of performing text clustering, some studies employ texts to cluster their authors. Liang et al. [34,35] developed dynamic topic models and embeddings cluster users based on their historical tweets. Edouard et al. [36] proposed temporal language models to learn dynamic user representations. These studies have shown authors' texts contain their characteristics, which support this work.

Author-Aware Text Classification. In the literature of text classification, authors have been extensively applied to text modeling in a supervised scheme. For example, Tang et al. [37] introduced user and item representations to the method input. The following studies [38,39] heavily rely on attention mechanisms [40] to affect text representations. More recently, Zhang et al. [41] explored the advantage of pre-trained language models [42] over user-aware text classification. Nevertheless, we have not seen the efforts of author representation learning in unsupervised text clustering.

2.4. Author-Aware multimodal learning

In the literature of multimodal learning, especially the text-to-image and image-to-text learning, there are some studies that consider the author effect for solving the problems, including text generation[43–45], image generation [46], and text-image matching [47]. However, none of these studies address the clustering task. The intrinsic difference is that these tasks are supervised while clustering is an unsupervised task. Moreover, from the technical aspect, most of these studies enhance the image and text representations. In contrast, our study concentrates on developing the contrastive learning losses to enable the clustering model learning in an unsupervised way.

3. Preliminaries

This section firstly clarifies the problem setting of author-aware text clustering. Consequently, it shows the empirical evidence for the motivation that considers the author effect in the text clustering procedure.

Problem formulation. In the scenario of user generated texts, we have a text corpus $\mathcal{D} = \{D_1, \cdots, D_{|\mathcal{D}|}\}$. Given the text D, it is written by an author (e.g., $u \in \mathcal{U}$). $\mathcal{U} = \{u_1, \cdots, u_{|\mathcal{U}|}\}$ is an author set. Suppose that D consists of title T and body B. Title T is denoted as $T = \{w_1, \cdots, w_{|T|}\}$ and body B is denoted as $B = \{w_1, \cdots, w_{|B|}\}$. All the words occurring in the text corpus form a vocabulary $\mathcal{V}$. Then the problem of author-aware text clustering is to partition the texts of the corpus into K clusters, where K is the specified cluster number.

Data analysis. To answer whether the author effect has some implications for text topics (clusters), we conduct data analysis on three real datasets, i.e., SO, YA, and TW, which will be introduced

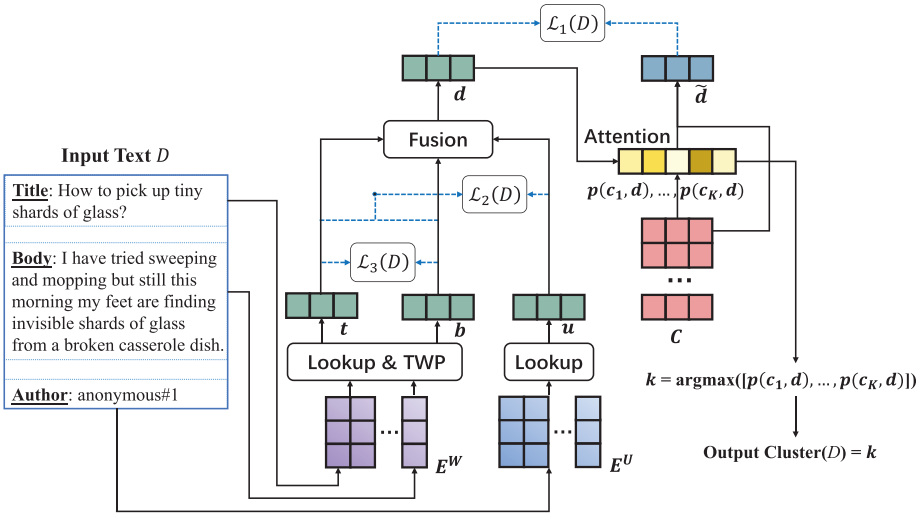

Fig. 3. Model architecture of CAT. **t**, **b**, **u**, and **d** denote the representations of title, body, author, and text, respectively. C means the embedding matrix of clusters. k is the output class label of the given document D. $\mathcal{L}_1$, $\mathcal{L}_2$, and $\mathcal{L}_3$ correspond to the losses of cluster-instance contrast, author-text contrast, and title-body contrast in Fig. 1, respectively. For simplicity, negative texts used in contrastive learning are omitted in the figure.

in Section 5.1.1. Although a few pioneering studies have considered the effect in text modeling for clustering, no data-driven evidence is provided for verifying the effect. To mitigate this gap, we introduce two novel author-centric concepts, i.e., author topic coverage and author topic concentration.

On the one hand, author topic coverage measures the ratio w.r.t. the number of topics that each author uses in its texts to the total number of topics. As Table 1 shows, the ratios of the three datasets are relatively low. This means that authors are usually interested in a small number of topics. On the other hand, author topic concentration measures the ratio w.r.t. the occurrence count of an author's most commonly used topic to the total occurrence count of all its topics. We use box plots to depict the ratio distributions of all the authors over the three datasets. As shown in Fig. 2, the average ratios are 0.606, 0.496, and 0.709 in SO, YA, and TW, respectively. Moreover, a majority of the ratios take relatively large values. These results reveal that among the few used topics, only a very few topics are usually used by the authors.

Based on the above observations, we can conclude that the authors have their own personal preferences for different topics. Thus considering the author effect can provide clues to correlate the texts of the same author with a few topics, which is promising to boost the text clustering performance. However, it is impractical to directly use authors' topics for clustering. This is because, in author-aware text clustering, we could not know the topics that authors have used, due to the available topics of the texts.

4. The computational approach

Approach overview: Fig. 3 presents the architecture of CAT. Basically speaking, the model takes a collection of author-aware text as input and is trained in an end-to-end fashion. After the model is well trained, the cluster label is naturally obtained for each text based on the largest cluster-level attention weight and no additional step is required. Within the model, it contains three computational procedures: (1) It takes the original representations of title, body, and author, and builds an integral representation for each text (see Section 4.1). (2) It performs text representation augmentation based on cluster-level attention over different cluster representations (see Section 4.2). (3) It builds a contrastive learning loss function, enabling to optimize the model parameters in an unsupervised manner (see Section 4.3). In what follows, the details of each procedure are elaborated.

Table 2
Basic statistics of the adopted datasets.

Data	SO	YA	TW
#Text	137,704	193,350	30,005
#Word	23,385	28,620	6,160
#Author	14,038	20,674	1,848
#Topic	111	158	280
Title Avg. Len.	5.4	4.7	-
Body Avg. Len.	69.7	18.4	8.8

4.1. Author-aware text representation formation

To illustrate the approach, we take text $D = \{u, T, B\}$ as an example, whose subscript is omitted for simplification. Nowadays the standardized operation for text modeling is to convert original one-hot encodings of words into low-dimensional embeddings, namely Lookup. Here we take this operation for both words and authors, which is defined as follows:

$$\mathbf{w} = \mathbf{E}^W \mathbf{x}_w, \qquad \mathbf{u} = \mathbf{E}^U \mathbf{x}_u, \tag{1}$$

where $\mathbf{x}_w \in \mathbb{Z}^{|\mathcal{V}|}$ and $\mathbf{x}_u \in \mathbb{Z}^{|\mathcal{U}|}$ are the one-hot encoding of word w and author u, respectively. $\mathbf{E}^W \in \mathbb{R}^{Z \times |\mathcal{V}|}$ denotes word embedding matrix and $\mathbf{E}^U \in \mathbb{R}^{Z \times |\mathcal{U}|}$ represents author embedding matrix. Z is the representation dimension.

After obtaining embedding $\mathbf{w}$ for each word in title T and body B, we propose the following Trainable Weighted Pooling (TWP) strategy to gain the title-level representation $\mathbf{t}$ and the body-level representation $\mathbf{b}$. Taking the title for clarification, the computational formula is given by:

$$\mathbf{t} = \sum_{w_i \in T} \omega_i \cdot \mathbf{w}_i \tag{2}$$

where ω_i is a trainable parameter corresponding to word w_i in vocabulary $\mathcal{V}$ shared by title and body. By inheriting the idea of Inverse Document Frequency (IDF) that a word occurring less in different texts is more distinguishable, we initialize ω by its corresponding IDF. The benefit is later verified in Table 7. Intuitively, the above equation makes texts with overlapped or similar discriminative words have similar representations. Analogously, the body-level representation $\mathbf{b}$ can be computed.

Based on the above representations, CAT fuses author representations $\mathbf{u}$, title representation $\mathbf{t}$, and body representation $\mathbf{b}$ to form

author-aware text representation $\mathbf{d}$. This is realized by the following weighted combination formula:

$$\mathbf{d} = \gamma_1 \cdot \mathbf{u} + \gamma_2 \cdot \mathbf{t} + \gamma_3 \cdot \mathbf{b}\,, \tag{3}$$

where $\gamma = [\gamma_1, \gamma_2, \gamma_3]$ are weight coefficients used to measure the relative influence of each representation part. To calculate the weight coefficients, similar to attention mechanism, we define the following computational formula to have non-negative coefficients:

$$\gamma = \cos\big(\tanh(\mathbf{Wq}), \tanh(\mathbf{Wk})\big) + 1 \tag{4}$$

where cos means the cosine function, $\mathbf{W}$ is a parameter matrix, $\mathbf{q} = \frac{1}{3}(\mathbf{u}+\mathbf{t}+\mathbf{b})$, and $\mathbf{k} = [\mathbf{u}, \mathbf{t}, \mathbf{b}]$. $\mathbf{q}$ could be regarded as the coarse-grained text representation, which is used to obtain the fine-grained text representation $\mathbf{d}$ through the above computation. We have also tried some different attention mechanisms (e.g, self-attention mechanism). However, no significant improvements are observed.

4.2. Cluster-based text representation augmentation

To perform cluster-instance contrastive learning, we build the text representation augmentation based on a cluster embedding matrix. Specifically, for the specified K clusters, we define a trainable cluster embedding matrix $\mathbf{C} \in \mathbb{R}^{Z\times K}$. CAT performs cluster-level attention to measure the correlation of each cluster with text representation $\mathbf{d}$, which is inspired by [6]. The probability that text D belongs to the k-th cluster is defined as follows:

$$p(c_k|\mathbf{d}) = \frac{\exp(\mathbf{c}_k^\top \mathbf{d})}{\sum_{k'=1}^{K} \exp(\mathbf{c}_{k'}^\top \mathbf{d})} \qquad k \in \{1, \cdots, K\}\,. \tag{5}$$

In light of the probabilities, the text representation augmentation can be obtained by the weighted combination of the cluster embeddings, given by:

$$\widetilde{\mathbf{d}} = \sum_{k=1}^{K} p(c_k|\mathbf{d}) \cdot \mathbf{c}_k\,. \tag{6}$$

The intuition of getting text representation augmentation through this way is that using the clusters similar to the text could represent the text to a certain extent.

4.3. Contrastive learning for model optimization

A crucial step towards end-to-end optimization of unsupervised clustering models is to devise effective objective functions. To achieve this, CAT develops contrastive learning losses based on cluster-instance contrast and instance-instance contrast. As shown in Fig. 3, the cluster-instance contrast corresponds to the cluster-instance contrastive loss $\mathcal{L}_1$, and the instance-instance contrast contains the author-text contrastive loss $\mathcal{L}_2$ and the title-body contrastive loss $\mathcal{L}_3$.

Based on the text representation $\mathbf{d}$ and its augmented representation $\widetilde{\mathbf{d}}$, CAT samples a text set $\mathcal{N}_D$ for text D and takes it as pseudo negative texts. Then the cluster-instance contrastive loss is devised to pull the representations of $\mathbf{d}$ and $\widetilde{\mathbf{d}}$ to be similar with each other while pushing the representations of negative texts away. Formally speaking, the cluster-instance contrastive loss is defined as follows:

$$\mathcal{L}_1(D) = \frac{1}{|\mathcal{N}_D|} \sum_{D_n \in \mathcal{N}_D} \Phi(\mathbf{d}, \widetilde{\mathbf{d}}, \mathbf{d}_n, \widetilde{\mathbf{d}_n})\,, \tag{7}$$

where $\mathbf{d}_n$ is the representation of the pseudo text D_n. We adopt the commonly used normalized temperature-scaled cross entropy loss [25] to implement Φ, which is defined as follows:

$$\Phi(\mathbf{d}, \widetilde{\mathbf{d}}, \mathbf{d}_n, \widetilde{\mathbf{d}_n}) = -\log \frac{\exp(\varphi(\mathbf{d}, \widetilde{\mathbf{d}})/\tau)}{\sum_{D_n \in \mathcal{N}_D} \exp(\varphi(\mathbf{d}, \mathbf{d}_n)/\tau) + \exp(\varphi(\mathbf{d}, \widetilde{\mathbf{d}_n})/\tau)}, \tag{8}$$

where φ is a similarity measure function that is implemented by a cosine similarity function. And τ is a temperature hyperparameter to be tuned.

To comprehensively consider the effect of the role of authors on text clustering, the author-text contrastive loss is developed. The basic intuition behind the loss is that the representations of a title and a body should be more similar to the corresponding author representation than the other author representations and text representations. Specifically, the loss is defined as follows:

$$\mathcal{L}_2(D) = \frac{1}{|\mathcal{N}_d|} \sum_{n \in \mathcal{N}_d} \Phi(\mathbf{t}+\mathbf{b}, \mathbf{u}, \mathbf{t}_n+\mathbf{b}_n, \mathbf{u}_n)\,. \tag{9}$$

Through the above manner, the text representations from the same author are pushed to be near to each other and thus might belong to a few clusters. This conforms to the conclusion in Section 3 that authors usually concentrate on a few topics. Besides, the topics of a title and a body in the same text are naturally consistent. They could be regarded as different views of a text. Therefore, we can also perform contrastive learning based on the title-body loss given by:

$$\mathcal{L}_3(D) = \frac{1}{|\mathcal{N}_d|} \sum_{n \in \mathcal{N}_d} \Phi(\mathbf{t}, \mathbf{b}, \mathbf{t}_n, \mathbf{b}_n)\,. \tag{10}$$

In light of the above three types of loss functions, we have the final loss function to optimize CAT in an end-to-end fashion:

$$\mathcal{L}(\mathbf{E}^W, \mathbf{E}^U, \mathbf{W}, \omega) = \sum_{D \in \mathcal{D}} \mathcal{L}_1(D) + \alpha\mathcal{L}_2(D) + \beta\mathcal{L}_3(D)\,, \tag{11}$$

where α and β are used to adjust the relative importance of $\mathcal{L}_2(D)$ and $\mathcal{L}_3(D)$, respectively. As we can see in the final loss function, the trainable parameters only include $\mathbf{E}^W$, $\mathbf{E}^U$, $\mathbf{W}$, and ω, which is simple, as compared to complicated neural networks for text modeling. When the well-trained CAT is used for identifying text clusters, it assigns a given text with the cluster label that has the largest probability in Eq. 5. We should emphasize that CAT could be easily generalized to clustering the texts that only have the body part. This is achieved by removing the title-body contrast loss in Eq. 11 and the title representation in Eq. 3.

5. Experiment

In this section, we first clarify the experimental setup and then discuss the results in detail.

5.1. Experimental setup

5.1.1. Dataset

In the experiments, we adopt three public datasets for comprehensive evaluations. Among them, the first two datasets, involving both title and body texts, are utilized as the main datasets. While the third dataset, containing no title texts, is used as the supplementary dataset for testing the clustering performance on the body-only texts. It is noteworthy that some other text datasets in previous studies are not used since the author IDs of these texts are unavailable.

Specifically, the first two public datasets are StackOverflow[1] (SO for short) and Yahoo! Answers[2] (YA for short). SO has questions

[1] https://archive.org/details/stackexchange

[2] https://webscope.sandbox.yahoo.com/catalog.php?datatype=l&did=11

covering a wide range of topics in computer programming while YA has a broader range of topics, including fashion, cooking, music, etc. The third dataset is Twitter [48] (TW for short), in which we use the tweet with a single hashtag for clustering. The ground-truth topics (cluster labels) are obtained based on the tags used in Stack-Overflow, the categories used in Yahoo! Answers, and the hashtags used in Twitter. Actually, these datasets might have some topics with overlapping semantics (e.g., "centos", "centos5", and "centos6" in StackOverflow, and "bestfeelingever" and "bestfeeling" in Twitter). To remedy this issue, we merge the overlapping topics into a coherent class for reliable experiments. Besides, meaningless hashtags such as "rt" are removed.

To preprocess the natural texts, we follow the typical preprocessing procedures, including converting words into lowercase, removing stop words and irregular words, and applying Porter stemming. To ensure statistical significance, we discard the words that occur less than 5 times and keep the authors with more than 4 texts. In summary, the brief statistics of the three datasets are shown in Table 2. *Title Avg. Len.* and *Body Avg. Len.* refers to the average number of words over the title and body of the texts, respectively.

5.1.2. Evaluation metric

We adopt the three commonly used clustering metrics, i.e., normalized mutual information (NMI), adjusted rand index (ARI), and clustering accuracy (ACC). The details about how to calculate these metrics can be referred to in the study [6].

5.1.3. Baseline

Considering the characteristics of author-aware text clustering, we choose general text clustering methods, as well as author-aware text clustering methods, as baselines for comparisons. The latest deep text clustering models, e.g., DKM [49], SCCL [14], and ARL [6], are also included.

General text clustering baselines:

- **Kmeans** adopts two types of textual features to build the input for this commonly-used clustering method. One is TF-IDF and the other is low-dimensional hidden vectors obtained by PCA.
- **LDA** [18] is a milestone topic model which behaves well on text mining and is followed by many variants.
- **GSDMM** [3] is a tailored topic model for user-generated short texts, with the assumption that all words in a text are generated by the same topic.
- **Doc2Vec** [50] expands Word2Vec [51] by incorporating text-level representations. The learned representations are then used for clustering algorithms (K-means used in this paper).
- **DEC** [4] performs text clustering by utilizing neural encoders and a soft-assignment based loss. It takes TF-IDF vectors as input features and firstly uses K-means clustering to initialize its cluster centers and then perform model optimization.
- **DKM** [49] builds upon a deep autoencoder network. The clustering is achieved by learning from a hybrid loss function, involving the reconstruction term and the Kmeans clustering term.
- **SCCL** [14] leverages the strengths of both bottom-up instance-wise contrastive learning and top-down clustering. We use the contextual augmenter [52] for text augmentation which consistently performs better as reported in the original paper.
- **ARL** [6] is a recently proposed representation learning approach that achieves the state-of-the-art performance on short text clustering.

Author-aware text clustering baselines:

- **ATM** [32] is the first topic model that considers the role of authors in text generation.
- **MvSCN** [53] is a deep version of multi-view spectral clustering which incorporates the local invariance within every single view and the consistency across different views into a novel objective function. We regard author as a special view to make **MvSCN** fits our setting.
- **DEC-Author** is an extension of DEC by firstly concatenating (better than adding in our experiments) text representations introduced in DEC with their corresponding author representations like CAT to form integrated text representations, and then training DEC as usual.
- **Kmeans-AuthorID** is an extension of the Kmeans method by concatenating the one-hot encoding of author IDs with the TF-IDF features of the target texts.
- **Kmeans-AuthorW** is also an extension of the Kmeans method. But in contrast to Kmeans-AuthorID, Kmeans-AuthorW first fuses all the texts of the same author to form a long text for each author. Then the author is represented by the TF-IDF word feature of the corresponding long text. Finally, the new TF-IDF features are combined with the original TF-IDF features of the target texts.
- **ARL-Author** is similar to the spirit of DEC-Author by adding (better than concatenating in our experiments) the author representations to the text representations of ARL as well.

5.1.4. Implementation

The proposed approach and all baselines set the cluster number as the ground-truth topic number. We also explored how different cluster numbers affect the performance change of our model and the best two baselines in Fig. 8. For the proposed approach, we empirically set dimension $Z = 64$, hyperparameter $\alpha = 1$, $\beta = 1$, and $\tau = 0.8$. Adam[54] is used as the optimizer, with its learning rate equal to 0.003 and batch size equal to 64. For ease of implementation, the size of the negative sets in Eq. 11 is uniformity set to 63 in a batch. Before training CAT, we utilize Word2Vec to initialize its word embeddings and perform K-means on text representations (i.e., IDF-based combination of word embeddings) to initialize its cluster representations, the same as [4,6]. Author embeddings are initialized by averaging their word embeddings. Since text clustering is an unsupervised task, we follow [4] by only performing simple hyperparameter tuning and avoiding dataset-specific tuning.

For the baselines, we follow the study [3,32] to set the hyperparameters of LDA, ATM, and GSDMM for achieving their good performance. The representation dimension and hyperparameters of Doc2Vec and other deep text clustering methods, including DEC, MvSCN, DKM, SCCL, and ARL are tuned. Besides, the dimension for PCA is set to 300. Most of the baseline models adopted in this paper have open implementations (except DEC-Author, Kmeans-AuthorID, Kmeans-AuthorW, and ARL-Author) to ensure a fair comparison. To reduce the impact of noise, all results in our experiments are averaged over 5 runs.

5.2. Experimental result

5.2.1. Performance comparison

Table 3 presents the performance comparison over all the text clustering methods on the two main datasets that contain both title and body texts. In addition, we also conduct the test on the supplementary dataset TW that does not have title part. This supplementary experiment could be leveraged to reveal the generalization of our proposed approach and we only present the performance of the well-performed methods (based on the results in Table 3) in Table 4. Based on the results, we have the following key observations:

◇ For the general text clustering approaches, Kmeans(PCA) does not achieve improvements over Kmeans(TF-IDF) on all the three datasets, indicating that the obtained low-dimensional representations independent of text clustering might not be beneficial for this task. Compared with the standard topic model LDA, GSDMM

Table 3
Performance comparison on the two main datasets.

Method	SO			YA		
	NMI	ARI	ACC	NMI	ARI	ACC
Kmeans(TF-IDF)	0.5110	0.2193	0.3764	0.3612	0.1088	0.2318
Kmenas(PCA)	0.4599	0.2000	0.3338	0.3411	0.1029	0.2127
LDA	0.3291	0.2143	0.3122	0.3990	0.1706	0.3680
Doc2Vec	0.5056	0.2392	0.3308	0.3466	0.0921	0.1371
DEC	0.5568	0.3124	0.4222	0.3595	0.1098	0.2135
SCCL	0.3225	0.3153	0.2893	0.4525	0.4452	0.2613
GSDMM	0.4679	0.1694	0.2983	0.5625	0.4096	0.4782
DKM	0.5267	0.2637	0.4384	0.3158	0.0994	0.2025
ARL	0.6297	0.4761	0.5022	0.5568	0.4188	0.4755
ATM	0.4306	0.2017	0.3501	0.3800	0.2291	0.3832
MvSCN	0.4721	0.2596	0.3648	0.3681	0.0942	0.1811
DEC-Author	0.5729	0.3353	0.4434	0.3924	0.1242	0.2499
Kmeans-AuthorID	0.5150	0.2115	0.3759	0.3701	0.0914	0.2379
Kmeans-AuthorW	0.6296	0.3787	0.4942	0.4313	0.1424	0.2427
ARL-Author	0.6215	0.4758	0.5046	0.5670	0.4315	0.4791
CAT	**0.8214**	**0.7297**	**0.7501**	**0.6365**	**0.4889**	**0.5225**

Table 4
Performance comparison on the supplementary dataset.

Method	TW		
	NMI	ARI	ACC
Kmeans(TF-IDF)	0.5895	0.1826	0.4149
Kmenas(PCA)	0.4736	0.1305	0.2659
DKM	0.5107	0.1508	0.3132
ARL	0.6559	0.6419	0.4924
Kmeans-AuthorID	0.5973	0.1869	0.4194
Kmeans-AuthorW	0.7775	0.2547	0.5889
ARL-Author	0.7597	0.6178	0.5741
CAT	**0.8454**	**0.7761**	**0.6815**

achieves much better performance on the two main datasets. This meets the expectation because GSDMM is an elaborate topic model for short text clustering. ARL, as the state-of-the-art deep text clustering model, consistently obtains the best performance among the general text clustering baselines on the three datasets, verifying the promising potential of tailoring deep representation learning for discrete word tokens. Besides, although SCCL utilizes contrastive learning for text clustering, its performance is not satisfactory on the two main datasets.

⋄ For the author-aware text clustering approaches, ATM, DEC-Author, Kmeans-AuthorID, Kmeans-AuthorW and ARL-Author gain better improvements over their corresponding foundation models, i.e., LDA, DEC, Kmeans(TF-IDF), and ARL in the first part of all the approaches. Considering their main difference is the introduction of authors, we could see the benefits of modeling the author effect on text clustering. One interesting observation is that Kmeans-AuthorW performs much better than Kmeans(TF-IDF), while Kmeans-AuthorID does not exhibit significantly better performance. The reason might be that: (1) The one-hot encodings of user IDs are too sparse and directly feeding them to Kmeans is not profitable for text distance computation. (2) The texts from the same author might share similar topics with the target text and thus they could bring more useful semantic information to complement the target text. Moreover, ARL-Author is only slightly better than ARL in some cases. This indicates that, for the better-performed deep text clustering model ARL, it is not trivial to have an effective way for incorporating the author effect. In addition, although MvSCN adopts title, body, and user as three views to perform clustering, it does not perform very well. This might be because each author is associated with multiple texts and directly using it for clustering is insufficient [55].

⋄ On the whole, CAT is superior among all the text clustering approaches and largely outperforms, showing the success of exploiting cluster-instance and instance-instance contrast learning for author-aware text clustering.

5.2.2. Model analysis

We dive into CAT for a deeper understanding by conducting ablation study and investigation of alternatives on CAT.

Ablation study. We reveal the performance contributions of the following four main components involved in CAT: the author representations (A-Rep), the author-text contrastive loss (AT-CL), the title-body contrastive loss (TB-CL), and the Trainable Weighted Pooling (TWP) strategy. The ablation study is realized by removing one or some of these components. And the detailed results are shown in Table 5 and 6, where ✓ denotes using the corresponding component. In the two tables, we do not consider removing the cluster-instance contrastive loss. This is because the clustering labels obtained by the cluster-level attention must be learned by the contrastive loss. Besides, we do not include TB-CL in Table 6 because the texts in TW does not contain titles and thus title-body contrast learning could not be performed.

As can be seen in the two tables, all the four components contribute positively to the final clustering performance. In particular, the component of the contrastive loss AT-CL makes the best contribution. It indicates that controlling textual similarities from different authors is beneficial. This is because any two authors probably have different topics due to their smaller topic coverage and larger topic concentration (See Table 1 and Fig. 2). Moreover, A-Rep and AT-CL bring larger improvements on SO and TW than that on YA. This observation meets expectations since SO and TW has smaller author topic coverage and larger author topic concentration. To sum up, these results support our viewpoint of considering the author effect in author-aware text clustering.

Investigation of alternatives. We further analyze CAT by developing some variants that modify one of its computational units each time. Specifically, "Word Att." and "Word Mean" denote that instead of proposing a trainable weighted pooling strategy to constitute title and body representations, CAT conducts an attention computation and a mean-pooling operation over words, respectively. "Fixed ω" means ω is not optimized during the model training process and "RandInit ω" represents randomly initializing ω rather than employing IDF. "BERT" replaces Word2Vec in CAT and is fine-tuned along with the training process. "UBT Mean" sets $\gamma_1 = \gamma_2 = \gamma_3 = \frac{1}{3}$ in Eq. 3.

Table 7 presents the corresponding results on the main two datasets. Firstly, by comparing CAT with "w/ Word Att." and "w/

Table 5
Ablation study of CAT on the two main datasets.

Component				SO			YA		
A-Rep	AT-CL	TB-CL	TWP	NMI	ARI	ACC	NMI	ARI	ACC
✓	✓	✓	✓	**0.8214**	**0.7297**	**0.7501**	**0.6365**	**0.4889**	**0.5225**
	✓	✓	✓	0.7950	0.6924	0.7162	0.6214	0.4756	0.5161
✓		✓	✓	0.7296	0.6161	0.6242	0.6116	0.4505	0.4946
✓	✓		✓	0.8015	0.7039	0.7278	0.6076	0.4427	0.4810
		✓	✓	0.7256	0.6158	0.6300	0.5984	0.4454	0.4915
			✓	0.6545	0.5363	0.5641	0.5612	0.3520	0.4332
				0.6465	0.4840	0.5142	0.5569	0.3344	0.4118

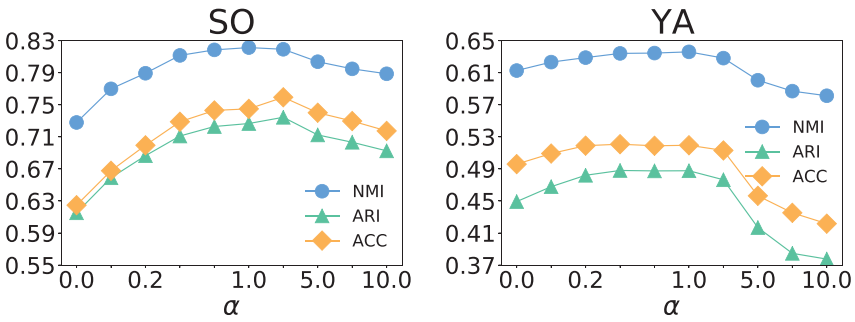

Fig. 4. Performance change w.r.t. α (β=1, τ=0.8).

Table 6
Ablation study of CAT on the supplementary dataset.

Component			TW		
A-Rep	AT-CL	TWP	NMI	ARI	ACC
✓	✓	✓	**0.8454**	**0.7761**	**0.6815**
	✓	✓	0.8201	0.6925	0.6393
✓		✓	0.7638	0.5791	0.5583
		✓	0.6904	0.6673	0.5318

Table 7
Performance comparison with alternatives.

Method	SO			YA		
	NMI	ARI	ACC	NMI	ARI	ACC
CAT	**0.8214**	**0.7297**	**0.7501**	**0.6365**	**0.4889**	**0.5225**
w/ Word Att.	0.7322	0.6028	0.6382	0.5852	0.4554	0.4828
w/ Word Mean	0.7769	0.6799	0.7094	0.6115	0.4681	0.4859
w/ Fixed ω	0.7951	0.7005	0.7275	0.6217	0.4549	0.4869
w/ RandInit ω	0.7700	0.6881	0.7170	0.5696	0.4202	0.4693
w/ BERT	0.6665	0.3370	0.3593	0.5989	0.4179	0.4523
w/ UBT Mean	0.8194	0.7254	0.7411	0.6344	0.4865	0.5185

Word Mean", it is obvious that trainable weighted pooling is a better choice under the framework. Moreover, we can see fixing ω or randomly initializing ω incurs a performance drop, as compared to using IDF for initialization. We have also considered standard BERT introduced by [42] by directly using pre-trained BERT or firstly pre-training BERT on our datasets and then using it for clustering. Unexpectedly, grafting BERT to our model degrades the performance to some extent. The reason might be that the syntactic and structural information of texts learned by BERT is not very crucial for text clustering tasks. Finally, although CAT marginally outperforms "w/ UBT Mean", its consistent improvements verify the gain of the adopted attention formula.

5.2.3. Hyperparameter analysis

This section analyzes how some key hyperparameters affect the clustering performance on the main two datasets.

Effect of α. Fig. 4 shows how the performance curves fluctuate w.r.t. the variation of α when setting β to 1 and τ to 0.8. As can be seen, the results are significantly improved when increasing α starting from 0. And relatively good results are achieved when α takes the value 1. This phenomenon reveals the positive contribution of the author-text contrast loss. When further increasing α, the results become notably worse.

Effect of β. Fig. 5 shows the performance curves when we change β after setting α=1 and τ to 0.8. They exhibit similar trends as the above curves of α. In particular, better performance is achieved when β takes the values in the intermediate range. As such, the title-body contrastive loss is demonstrated to present a stable contribution to the clustering performance.

Effect of τ. Following previous study [25], we also adjust the temperature τ in Eq. 8 to find how it affects the performance. Based on the curves in Fig. 6, we can see when τ lies in the range of $[0.6, 0.8]$, the results are satisfied on both datasets. As such, the hyperparameter is not sensitive in practice.

Effect of embedding dimension. Fig. 7 presents how the embedding dimension affects the performance. When increasing the dimension from 16 to 64, the improvements are obvious. After the dimension surpasses 64, the results remain relatively stable.

Effect of cluster number. Fig. 8 compares CAT with ARL and ARL-Author over different cluster numbers, wherein Ratio indicates the ratio of the specified cluster number to the original topic number. NMI is shown as a representative. As can be seen, CAT consistently improves ARL and ARL-Author by large margins.

5.2.4. Clustering result analysis

This section analyzes the clustering results by studying error cases and investigating the different number of text clusters for different authors.

Error case study. To give some intuitive understanding of clustering results, we summarize two main error types.

1. Sample level error. We select one cluster from SO as an example to show the case of the wrong classification made by CAT and similar phenomena could be observed for YA. The cluster consists of *algorithm* (72.1%), *MATLAB* (5.3%), and *R* (4.2%). While the cluster is mainly about algorithm design, some texts refer to implementing algorithms in specific environments. For in-

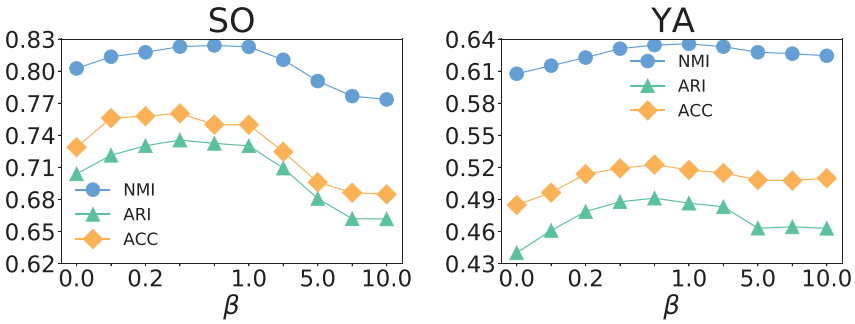

Fig. 5. Performance change w.r.t. β (α=1, τ=0.8).

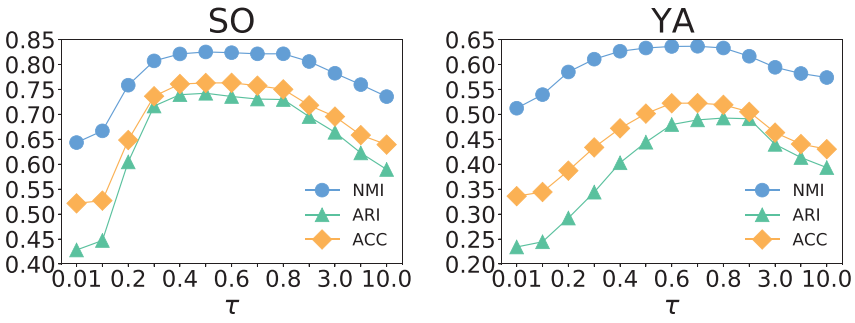

Fig. 6. Performance change w.r.t. τ (α=1, β=1).

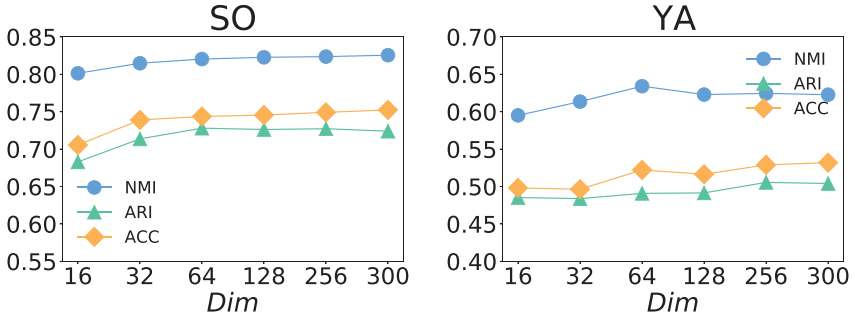

Fig. 7. Performance change w.r.t. dimension.

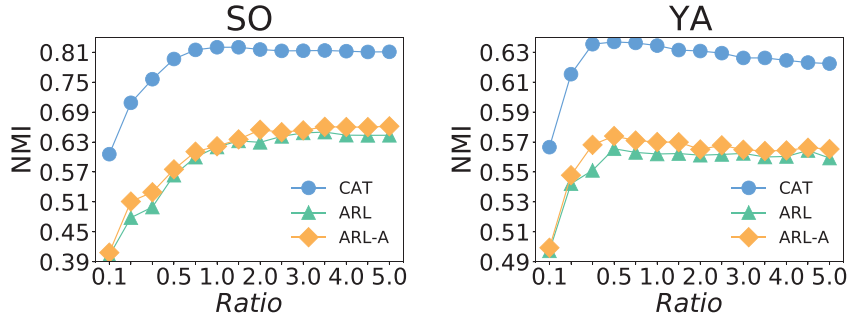

Fig. 8. Performance comparison w.r.t. different cluster numbers. Note that ARL-A is short for ARL-Author.

stance, "*Estimating the number of arithmetic operations in an algorithm/code/program.*" talks about a statistical operation in MATLAB according to its ground-truth topic. However, manual validation into its title and body has found no signal of MATLAB. Such absence of keywords is common, which impedes the overall performance.

2. Cluster level error. One cluster contains similar topics. Example in SO: A cluster consist of *iOS* (34.7%), *iPhone* (34.4%), *Swift* (14.3%). Since iPhone is equipped with the iOS system and the iOS system could be developed with Swift, these topics are related to some extent. Similar phenomenon in YA: (Newborn & Baby, 70.2%), (Pregnancy, 20.4%).

Text clusters of authors. After clustering, we calculate the average cluster count that each author is associated with for CAT and ARL. On SO, it is 4.04 for CAT and 5.18 for ARL, while on YA, it is 5.12 for CAT and 5.83 for ARL. The smaller counts of CAT reveal that considering the role of authors narrows down author topic coverage in results.

6. Conclusion

This paper studies text clustering by considering how the author's information of texts affects the clustering performance. A novel contrastive learning based representation learning model, i.e., CAT is devised. It develops both cluster-instance contrast and instance-instance contrast losses to enable the end-to-end unsupervised learning of the model. The author's information is not only used to enhance the text representations but also taken as a different view of texts for building contrastive learning loss. Experimental results demonstrate that our model performs much better than the competitive baselines and injecting the author effect into text clustering is profitable. In future work, we will consider the author effect in data clustering of other modalities, such as images and even multi-modal data.

Declaration of Competing Interest

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

References

[1] C.C. Aggarwal, C. Zhai, A Survey of Text Clustering Algorithms, in: Mining text data, Springer, 2012, pp. 77–128.
[2] M. Filippone, F. Camastra, F. Masulli, S. Rovetta, A survey of kernel and spectral methods for clustering, Pattern Recognit 41 (1) (2008) 176–190.
[3] J. Yin, J. Wang, A dirichlet multinomial mixture model-based approach for short text clustering, in: SIGKDD, 2014, pp. 233–242.
[4] J. Xie, R. Girshick, A. Farhadi, Unsupervised deep embedding for clustering analysis, in: ICML, 2016, pp. 478–487.
[5] Z. Jiang, Y. Zheng, H. Tan, B. Tang, H. Zhou, Variational deep embedding: An unsupervised and generative approach to clustering, in: IJCAI, 2017, pp. 1965–1972.
[6] W. Zhang, C. Dong, J. Yin, J. Wang, Attentive representation learning with adversarial training for short text clustering, IEEE TKDE online (online) (2021) online.
[7] A. van den Oord, Y. Li, O. Vinyals, Representation learning with contrastive predictive coding, arXiv abs/1807.03748 (2018).
[8] Y. Li, P. Hu, J.Z. Liu, D. Peng, J.T. Zhou, X. Peng, Contrastive clustering, in: AAAI, 2021, pp. 8547–8555.
[9] H. Zhong, J. Wu, C. Chen, J. Huang, M. Deng, L. Nie, Z. Lin, X.-S. Hua, Graph contrastive clustering, in: ICCV, 2021, pp. 9204–9213.
[10] H. Zhao, X. Yang, Z. Wang, E. Yang, C. Deng, Graph debiased contrastive learning with joint representation clustering, in: IJCAI, 2021, pp. 3434–3440.
[11] E. Pan, Z. Kang, Multi-view contrastive graph clustering, NeurIPS, 2021.
[12] L. Liu, Z. Kang, L. Tian, W. Xu, X. He, Multilayer graph contrastive clustering network, arXiv preprint arXiv:2112.14021 (2021).
[13] W. Xia, Q. Gao, M. Yang, X. Gao, Self-supervised contrastive attributed graph clustering, arXiv preprint arXiv:2110.08264 (2021).
[14] D. Zhang, F. Nan, X. Wei, S. Li, H. Zhu, K.R. McKeown, R. Nallapati, A.O. Arnold, B. Xiang, Supporting clustering with contrastive learning, in: NAACL, 2021, pp. 5419–5430.
[15] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, Roberta: a robustly optimized bert pretraining approach, arXiv preprint arXiv:1907.11692 (2019).
[16] P. Bafna, D. Pramod, A. Vaidya, Document clustering: Tf-idf approach, in: ICEEOT, 2016, pp. 61–66.
[17] J. MacQueen, et al., Some methods for classification and analysis of multivariate observations, in: In Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, 1967, pp. 281–297.
[18] D.M. Blei, A.Y. Ng, M.I. Jordan, Latent dirichlet allocation, JMLR 3 (Jan) (2003) 993–1022.
[19] S. Deerwester, S.T. Dumais, G.W. Furnas, T.K. Landauer, R. Harshman, Indexing by latent semantic analysis, JASIS 41 (6) (1990) 391–407.
[20] F. Nan, R. Ding, R. Nallapati, B. Xiang, Topic modeling with wasserstein autoencoders, in: ACL, 2019, pp. 6345–6381.
[21] G.E. Hinton, R.R. Salakhutdinov, Reducing the dimensionality of data with neural networks, Science 313 (5786) (2006) 504–507.
[22] M. Ester, H. Kriegel, J. Sander, X. Xu, A density-based algorithm for discovering clusters in large spatial databases with noise, in: KDD, 1996, pp. 226–231.
[23] A. Hadifar, L. Sterckx, T. Demeester, C. Develder, A self-training approach for short text clustering, in: RepL4NLP, 2019, pp. 194–199.
[24] R. Hadsell, S. Chopra, Y. LeCun, Dimensionality reduction by learning an invariant mapping, in: CVPR, 2006, pp. 1735–1742.
[25] T. Chen, S. Kornblith, M. Norouzi, G.E. Hinton, A simple framework for contrastive learning of visual representations, in: ICML, 2020, pp. 1597–1607.
[26] T. Gao, X. Yao, D. Chen, Simcse: Simple contrastive learning of sentence embeddings, in: M. Moens, X. Huang, L. Specia, S.W. Yih (Eds.), EMNLP, 2021, pp. 6894–6910.
[27] M. Schreyer, T. Sattarov, D. Borth, Multi-view contrastive self-supervised learning of accounting data representations for downstream audit tasks, arXiv preprint arXiv:2109.11201 (2021).
[28] C. Yang, Z. An, Y. Xu, Multi-view contrastive learning for online knowledge distillation, in: ICASSP, 2021, pp. 3750–3754.
[29] J. Ma, Y. Zhang, L. Zhang, Discriminative subspace matrix factorization for multiview data clustering, Pattern Recognit 111 (2021) 107676.
[30] Y. Chen, X. Xiao, Y. Zhou, Multi-view subspace clustering via simultaneously learning the representation tensor and affinity matrix, Pattern Recognit 106 (2020) 107441.
[31] S. Huang, Z. Kang, Z. Xu, Auto-weighted multi-view clustering via deep matrix decomposition, Pattern Recognit 97 (2020) 107015.
[32] M. Rosen-Zvi, T.L. Griffiths, M. Steyvers, P. Smyth, The author-topic model for authors and documents, in: UAI, 2004, pp. 487–494.
[33] W. Zhang, J. Wang, Prior-based dual additive latent dirichlet allocation for user-item connected documents, in: IJCAI, 2015, pp. 1405–1411.
[34] S. Liang, Z. Ren, Y. Zhao, J. Ma, E. Yilmaz, M. de Rijke, Inferring dynamic user interests in streams of short texts for user clustering, ACM TOIS 36 (1) (2017) 10:1–10:37.
[35] S. Liang, X. Zhang, Z. Ren, E. Kanoulas, Dynamic embeddings for user profiling in twitter, in: SIGKDD, 2018, pp. 1764–1773.
[36] E. Delasalles, S. Lamprier, L. Denoyer, Learning dynamic author representations with temporal language models, in: J. Wang, K. Shim, X. Wu (Eds.), ICDM, 2019, pp. 120–129.
[37] D. Tang, B. Qin, T. Liu, Learning semantic representations of users and products for document level sentiment classification, in: ACL, 2015, pp. 1014–1023.
[38] H. Chen, M. Sun, C. Tu, Y. Lin, Z. Liu, Neural sentiment classification with user and product attention, in: EMNLP, 2016, pp. 1650–1659.
[39] Z. Wu, X. Dai, C. Yin, S. Huang, J. Chen, Improving review representations with user attention and product attention for sentiment classification, in: AAAI, 2018, pp. 5989–5996.
[40] D. Bahdanau, K. Cho, Y. Bengio, Neural machine translation by jointly learning to align and translate, ICLR, 2015.
[41] Y. Zhang, W. Zhang, Improved representations for personalized document-level sentiment classification, in: DASFAA, 2020, pp. 769–785.
[42] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: pre-training of deep bidirectional transformers for language understanding, in: NAACL, 2019, pp. 4171–4186.
[43] C.C. Park, B. Kim, G. Kim, Attend to you: Personalized image captioning with context sequence memory networks, in: CVPR, 2017, pp. 6432–6440.
[44] S. Li, T. Xiao, H. Li, W. Yang, X. Wang, Identity-aware textual-visual matching with latent co-attention, in: ICCV, 2017, pp. 1908–1917.
[45] W. Zhang, Y. Ying, P. Lu, H. Zha, Learning long- and short-term user literal-preference with multimodal hierarchical transformer network for personalized image caption, in: AAAI, 2020, pp. 9571–9578.
[46] Y. Lin, P. Ren, Z. Chen, Z. Ren, J. Ma, M. de Rijke, Improving outfit recommendation with co-supervision of fashion generation, in: WWW, 2019, pp. 1095–1105.
[47] K. Shuster, S. Humeau, H. Hu, A. Bordes, J. Weston, Engaging image captioning via personality, in: CVPR, 2019, pp. 12516–12526.
[48] Z. Cheng, J. Caverlee, K. Lee, You are where you tweet: a content-based approach to geo-locating twitter users, in: J. Huang, N. Koudas, G.J.F. Jones, X. Wu, K. Collins-Thompson, A. An (Eds.), CIKM, 2010, pp. 759–768.
[49] M.M. Fard, T. Thonet, É. Gaussier, Deep k-means: jointly clustering with k-means and learning representations, Pattern Recognit. Lett. 138 (2020) 185–192.
[50] Q. Le, T. Mikolov, Distributed representations of sentences and documents, in: ICML, 2014, pp. 1188–1196.
[51] T. Mikolov, I. Sutskever, K. Chen, G.S. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, in: NeurIPS, 2013, pp. 3111–3119.
[52] S. Kobayashi, Contextual augmentation: Data augmentation by words with paradigmatic relations, in: NAACL, 2018, pp. 452–457.
[53] Z. Huang, J.T. Zhou, X. Peng, C. Zhang, H. Zhu, J. Lv, Multi-view spectral clustering network, in: IJCAI, 2019, pp. 2563–2569.
[54] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, ICLR, 2015.
[55] Y. Yang, H. Wang, Multi-view clustering: a survey, Big Data Mining and Analytics 1 (2) (2018) 83–107.

Xudong Tang is currently working toward an M.S. degree in Computer Science and Technology with the School of Computer Science and Technology, East China Normal University, Shanghai, China. His research interests include unsupervised representation learning and clustering.

Chao Dong received his M.S. degree in the School of Software Engineering from East China Normal University, Shanghai, China in 2020. His research interests include natural language processing and text clustering.

Wei Zhang received his Ph.D. degree in computer science and technology from Tsinghua University, Beijing, China, in 2016. He is currently a professor in the School of Computer Science and Technology, East China Normal University, Shanghai, China. His research interests mainly include data mining and machine learning applications.