A Survey of Short Text Clustering

JUNTONG NI, Shandong University, China

Analyzing short texts infers discriminative and coherent latent topics, which is a significant and essential problem because many real-world applications require semantic understanding of short messages. Due to the extremely low amount of word co-occurrence information available in short texts, conventional long text topic modeling techniques (such PLSA and LDA) based on word co-occurrences are unable to adequately address this issue. Since short texts tend to be sparse, short text topic modeling has already garnered a lot of interest from the machine learning research community in recent years. In this study, we review all available short text topic modeling strategies in-depth. With examples of representative techniques, we offer four kinds of methods based on similarity, topic model, deep learning, and generative adversarial networks.

CCS Concepts: • **General and reference** → **Surveys and overviews**.

Additional Key Words and Phrases: short text, clustering

ACM Reference Format:
JUNTONG NI. 2022. A Survey of Short Text Clustering. *ACM Trans. Graph.* 37, 4, Article 111 (August 2022), 7 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Short text have gotten to be an critical data source counting news features, status upgrades, web page scraps, tweets, question/answer sets, etc. Short text examination has been drawing in expanding consideration in later a long time due to the ubiquity of short text within the real-world [Lin et al. 2014] [Qiang et al. 2016] [Shi et al. 2018]. Compelling and productive models gather the idle subjects from short writings, which can offer assistance find the inactive semantic structures that happen in a collection of records. Short text theme modeling calculations are continuously connected into numerous assignments such as subject location [Wang et al. 2007], classification [Sriram et al. 2010], comment summarization [Ma et al. 2012], client intrigued profiling [Weng et al. 2010].

Conventional theme modeling calculations such as Probabilistic Latent Semantic Analysis (PLSA) [Hofmann 1999] and Latent Dirichlet Allocation (LDA) [Blei et al. 2003] are broadly embraced for finding inactive semantic structure from content corpus without requiring any earlier comments or labeling of the archives. In these calculations, each report may be seen as a blend of different themes and each point is characterized by a dissemination over all the words. Measurable procedures (e.g., Variational strategies and Gibbs inspecting) are at that point utilized to gather the inactive theme dispersion of each record and the word dispersion of each point utilizing higherorder word co-occurrence designs [Blei 2012]. These calculations and their variations have had a major affect on various connected areas in modeling content collections news articles, inquire about papers, and blogs [Hoffman et al. 2010] [Xie and Xing 2013] [Xie et al. 2015]. In any case, conventional point models involvement expansive execution corruption over short texts due to the need of word co-occurrence data in each short text [Lin et al. 2014], [Cheng et al. 2014]. Subsequently, short texts theme modeling has as of now pulled in much consideration from the machine learning investigate community in later a long time, which points at overcoming the issue of scantiness in short texts.

Prior works [Jin et al. 2011], [Phan et al. 2008] still utilized conventional theme models for short texts, but abused outside information or metadata to bring in extra valuable word co-occurrences over short texts, and so may boost the execution of theme models. For case, [Phan et al. 2008] to begin with learned latent topics from Wikipedia, and after that induced points from short texts. [Weng et al. 2010] and [Mehrotra et al. 2013] amassed tweets for pseudo-document utilizing hashtags and the same client individually. The issue lies in that assistant data or metadata isn't continuously accessible or fair as well expensive for sending. These ponders recommend that subject models particularly planned for common short texts are basic. This study will give a scientific categorization that captures the existing short texts clustering methods.

News conglomeration websites frequently depend on news headlines to cluster diverse source news approximately the same occasion. From these short texts, able to found these taking after characteristics. (1) Clearly, each short texts lacks enough word co-occurrence data. (2) Due to a couple of words in each text, most writings are likely produced by as it were one subject (e.g, text 1, text2, text 3). (3) Measurable data of words among writings cannot completely capture words that are semantically related but seldom co-occur. For illustration, President Trump of text 1 and White House of text 2 are profoundly semantically related, and AI is brief for Artifical Insights. (4) The single-topic presumption may be as well solid for a few short texts. For case, text 3 is likely related with a little number of points (e.g., one to three subjects). Considering these characteristics, existing short texts theme modeling calculations were proposed by attempting to unravel one or two of these characteristics. Here, we isolate the short texts theme modeling calculations fundamentally into the taking after five major categories, similarity based, topic model based, deep learning based, generative adversarial networks based, and stc-specific clustering methods.

2 PROBLEM DEFINITION

Clustering is basically a technique that groups similar data points such that the points in the same group are more similar to each other than the points in the other groups. The group of similar data points is called a Cluster.

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

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
© 2022 Association for Computing Machinery.

0730-0301/2022/8-ART111 $15.00
https://doi.org/XXXXXXX.XXXXXXX

3 SIMILARITY-BASED CLUSTERING

3.1 Partitional algorithms

K-means [MacQueen 1967] clustering is one of the simplest and popular unsupervised machine learning algorithms. Typically, unsupervised algorithms make inferences from datasets using only input vectors without referring to known, or labelled, outcomes. The objective of K-means is simple: group similar data points together and discover underlying patterns. To achieve this objective, K-means looks for a fixed number (k) of clusters in a dataset. A cluster refers to a collection of data points aggregated together because of certain similarities. Define a target number k, which refers to the number of centroids you need in the dataset. A centroid is the imaginary or real location representing the center of the cluster. Every data point is allocated to each of the clusters through reducing the in-cluster sum of squares. In other words, the K-means algorithm identifies k number of centroids, and then allocates every data point to the nearest cluster, while keeping the centroids as small as possible. The 'means' in the K-means refers to averaging of the data; that is, finding the centroid.

To process the learning data, the K-means algorithm in data mining starts with a first group of randomly selected centroids, which are used as the beginning points for every cluster, and then performs iterative (repetitive) calculations to optimize the positions of the centroids. It halts creating and optimizing clusters when either:

- The centroids have stabilized — there is no change in their values because the clustering has been successful.
- The defined number of iterations has been achieved.

However, its performance is usually not as competitive as those of the other sophisticated clustering techniques because slight variations in the data could lead to high variance.

In conclusion, it is a classical and simple clustering approach, relying on hand-crafted features for text clustering. However, the number of clusters need to be specified in advance, and they are sensitive to the initialization.

3.2 Hierarchical algorithms

As we already have other clustering algorithms such as K-Means Clustering, then why we need hierarchical clustering? So, as we have seen in the K-means clustering that there are some challenges with this algorithm, which are a predetermined number of clusters, and it always tries to create the clusters of the same size. To solve these two challenges, we can opt for the hierarchical clustering algorithm because, in this algorithm, we don't need to have knowledge about the predefined number of clusters.

The agglomerative hierarchical clustering algorithm is a popular example of hierarchical algorithms. To group the datasets into clusters, it follows the bottom-up approach. It means, this algorithm considers each dataset as a single cluster at the beginning, and then start combining the closest pair of clusters together. It does this until all the clusters are merged into a single cluster that contains all the datasets.

This hierarchy of clusters is represented in the form of the dendrogram.

In conclusion, similar as K-means, Hierarchical Clustering (HieClu) is another simple baseline for clustering. The drawback of hierarchical algorithms is that they need to assume the true number of clusters or a similarity threshold, and they cannot scale well with large data sets.

3.3 Density-based algorithms

Density-Based Clustering refers to unsupervised learning methods that identify distinctive groups/clusters in the data, based on the idea that a cluster in data space is a contiguous region of high point density, separated from other such clusters by contiguous regions of low point density.

Ordering points to identify the clustering structure [Ankerst et al. 1999] (OPTICS) is an algorithm for finding density-based clusters in spatial data. It was presented by Mihael Ankerst, Markus M. Breunig, Hans-Peter Kriegel and Jörg Sander. Its basic idea is similar to DBSCAN [Ester et al. 1996], but it addresses one of DBSCAN's major weaknesses: the problem of detecting meaningful clusters in data of varying density. To do so, the points of the database are (linearly) ordered such that spatially closest points become neighbors in the ordering. Additionally, a special distance is stored for each point that represents the density that must be accepted for a cluster so that both points belong to the same cluster. This is represented as a dendrogram.

In conclusion, the advantage of density-based algorithms is that they do not need to specify the number of clusters in advance, and can detect the outliers of the dataset. However, they have limitations in handling high-dimensional data like text. Because the feature space of high-dimensional data is usually sparse, density-based algorithms have difficulty to distinguish high-density regions from low-density regions.

3.4 Clustering Short Texts using Wikipedia

Subscribers to the popular news or blog feeds (RSS/Atom) often face the problem of information overload as these feed sources usually deliver large number of items periodically. One solution to this problem could be clustering similar items in the feed reader to make the information more manageable for a user. Clustering items at the feed reader end is a challenging task as usually only a small part of the actual article is received through the feed. In this paper, we propose a method of improving the accuracy of clustering short texts by enriching their representation with additional features from Wikipedia [Banerjee et al. 2007].

4 TOPIC MODEL-BASED CLUSTERING

4.1 Latent dirichlet allocation (LDA)

It is one of the most popular topic modeling methods [Blei et al. 2003]. Topic modeling provides methods for automatically organizing, understanding, searching, and summarizing large electronic archives. It can help with the following:

- discovering the hidden themes in the collection.
- classifying the documents into the discovered themes.
- using the classification to organize/summarize/search the documents.

For example, let's say a document belongs to the topics food, dogs and health. So if a user queries "dog food", they might find the above-mentioned document relevant because it covers those topics(among other topics). We are able to figure its relevance with respect to the query without even going through the entire document.

Each document is made up of various words, and each topic also has various words belonging to it. The aim of LDA is to find topics a document belongs to, based on the words in it.

LDA has some assumptions. First, Each document is just a collection of words or a "bag of words". Thus, the order of the words and the grammatical role of the words (subject, object, verbs, . . .) are not considered in the model. Second, Words like am/is/are/of/a/the/but/. . . don't carry any information about the "topics" and therefore can be eliminated from the documents as a preprocessing step. In fact, we can eliminate words that occur in at least %80 %90 of the documents, without losing any information. For example, if our corpus contains only medical documents, words like human, body, health, etc might be present in most of the documents and hence can be removed as they don't add any specific information which would make the document stand out. Third, we know beforehand how many topics we want. 'k' is pre-decided. Fourth, all topic assignments except for the current word in question are correct, and then updating the assignment of the current word using our model of how documents are generated

There are 2 parts in LDA. The words that belong to a document, that we already know. And the words that belong to a topic or the probability of words belonging into a topic, that we need to calculate. In the algorithm of LDA, we should first go through each document and randomly assign each word in the document to one of k topics (k is chosen beforehand). Then, for each document d, go through each word w and compute $p(topic \mid document)$ and $p(word \mid topic)$. Finally, update the probability for the word w belonging to topic t, as

$$p(word\ with\ topic) = p(topic \mid document) * p(word \mid topic)$$

In conclusion, LDA is a classical and standard generative statistical model which learns a topic (cluster) distribution for each document.

4.2 PLSA

Probabilistic Latent Semantic Analysis (PLSA) [Hofmann 2013] is a novel statistical technique for the analysis of two-mode and co-occurrence data, which has applications in information retrieval and filtering, natural language processing, machine learning from text, and in related areas. Compared to standard Latent Semantic Analysis which stems from linear algebra and performs a Singular Value Decomposition of co-occurrence tables, the proposed method is based on a mixture decomposition derived from a latent class model. This results. in a more principled approach which has a solid foundation in statistics. In order to avoid overfitting, they propose a widely applicable generalization of maximum likelihood model fitting by tempered EM.

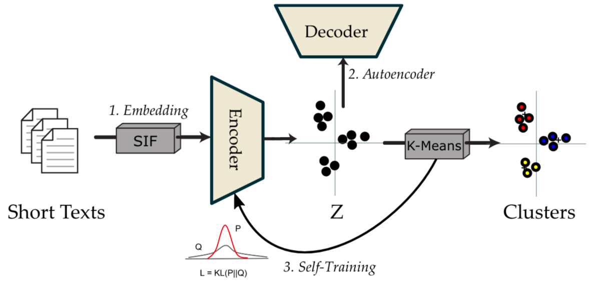

Fig. 1. Short text clustering using SIF embedding, an autoencoder architecture and self-training.

4.3 GSDMM

Short text clustering has become an increasingly important task with the popularity of social media like Twitter, Google+, and Facebook. It is a challenging problem due to its sparse, high-dimensional, and large-volume characteristics. In this paper, they proposed a collapsed Gibbs Sampling algorithm for the Dirichlet Multinomial Mixture model for short text clustering (abbr. to GSDMM) [Yin and Wang 2014]. They found that GSDMM can infer the number of clusters automatically with a good balance between the completeness and homogeneity of the clustering results, and is fast to converge. GSDMM can also cope with the sparse and high-dimensional problem of short texts, and can obtain the representative words of each cluster.

4.4 BTM

[Yan et al. 2013] Short texts are popular on today's Web, especially with the emergence of social media. Inferring topics from large scale short texts becomes a critical but challenging task for many content analysis tasks. Conventional topic models such as latent Dirichlet allocation (LDA) and probabilistic latent semantic analysis (PLSA) learn topics from document-level word cooccurrences by modeling each document as a mixture of topics, whose inference suffers from the sparsity of word co-occurrence patterns in short texts. In this paper, we propose a novel way for short text topic modeling, referred as biterm topic model (BTM). BTM learns topics by directly modeling the generation of word co-occurrence patterns (i.e., biterms) in the corpus, making the inference effective with the rich corpus-level information. To cope with large scale short text data, we further introduce two online algorithms for BTM for efficient topic learning. Experiments on real-word short text collections show that BTM can discover more prominent and coherent topics, and significantly outperform the state-of-the-art baselines. We also demonstrate the appealing performance of the two online BTM algorithms on both time efficiency and topic learning.

5 DEEP LEARNING BASED METHODS

5.1 STCC

The most relevant study is STCC [Xu et al. 2017] designed for short text clustering. The method proposes a flexible Self-Taught Convolutional neural network framework for Short Text Clustering (dubbed STC2), which can flexibly and successfully incorporate more useful semantic features and learn non-biased deep text representation in an unsupervised manner. In their framework, the original raw text features are firstly embedded into compact binary codes by using one existing unsupervised dimensionality reduction methods. Then, word embeddings are explored and fed into convolutional neural networks to learn deep feature representations, meanwhile the output units are used to fit the pre-trained binary codes in the training process. Finally, they get the optimal clusters by employing K-means to cluster the learned representations. However, it is not trained in an end-to-end fashion, leaving room for performance improvement.

In short, this model mainly consists of three separate steps. It first trains a convolutional neural network with the help of autoencoders. Afterwards, the well-trained model is employed to get text embeddings, which are fed into K-means for final clustering.

5.2 Gaussian-LDA

Continuous space word embeddings learned from large, unstructured corpora have been shown to be effective at capturing semantic regularities in language. In this paper they replace LDA's parameterization of "topics" as categorical distributions over opaque word types with multivariate Gaussian distributions on the embedding space. This encourages the model to group words that are a priori known to be semantically related into topics. To perform inference, they introduce a fast collapsed Gibbs sampling algorithm based on Cholesky decompositions of covariance matrices of the posterior predictive distributions. They further derive a scalable algorithm that draws samples from stale posterior predictive distributions and corrects them with a Metropolis–Hastings step.

5.3 DEC

Clustering is central to many data-driven application domains and has been studied extensively in terms of distance functions and grouping algorithms. In this paper, they propose Deep Embedded Clustering (DEC) [Xie et al. 2016], a method that simultaneously learns feature representations and cluster assignments using deep neural networks. DEC learns a mapping from the data space to a lower-dimensional feature space in which it iteratively optimizes a clustering objective. DEC utilizes the idea of self-training to measure the gap between the distribution of soft cluster assignments and its corresponding auxiliary distribution through Kullback-Leibler (KL) divergence. It is a deep embedded clustering model that leverages autoencoder, with TF-IDF features as input to map documents into low-dimensional embeddings. Then the mapping function and cluster representations are refined based on the idea of self-training.

5.4 STC

In Figure 1, STC [Hadifar et al. 2019] adopts self-training as well, meanwhile, Smoothed Inverse Frequency (SIF) is introduced to weight word embeddings in each document. STC adopts self-training inspired by DEC and uses Smoothed Inverse Frequency (SIF) to compute a weighted average of pre-trained word embeddings in a stage independent of optimizing its clustering model. Short text clustering is a challenging problem when adopting traditional bag-of-words or TFIDF representations, since these lead to sparse vector representations for short texts. Lowdimensional continuous representations or embeddings can counter that sparseness problem: their high representational power is exploited in deep clustering algorithms. While deep clustering has been studied extensively in computer vision, relatively little work has focused on NLP. The method the paper propose, learns discriminative features from both an autoencoder and a sentence embedding, then uses assignments from a clustering algorithm as supervision to update weights of the encoder network. Yet STC obtains text representations in a separate step and cannot optimize word embeddings along with learning text clustering.

5.5 VaDE

VaDE [Jiang et al. 2016] extends variational auto-encoder to integrate a Gaussian mixture model to generate latent vectors, instead of just using a Gaussian distribution. Clustering is among the most fundamental tasks in machine learning and artificial intelligence. In this paper, they propose Variational Deep Embedding (VaDE), a novel unsupervised generative clustering approach within the framework of Variational Auto-Encoder (VAE). Specifically, VaDE models the data generative procedure with a Gaussian Mixture Model (GMM) and a deep neural network (DNN): 1) the GMM picks a cluster; 2) from which a latent embedding is generated; 3) then the DNN decodes the latent embedding into an observable. Inference in VaDE is done in a variational way: a different DNN is used to encode observables to latent embeddings, so that the evidence lower bound (ELBO) can be optimized using the Stochastic Gradient Variational Bayes (SGVB) estimator and the reparameterization trick. Moreover, by VaDE's generative nature, they show its capability of generating highly realistic samples for any specified cluster, without using supervised information during training.

5.6 ARL

Short text clustering has far-reaching effects on semantic analysis, showing its importance for multiple applications such as corpus summarization and information retrieval. However, it inevitably encounters the severe sparsity of short text representations, making the previous clustering approaches still far from satisfactory. In this paper, they present a novel attentive representation learning model (ARL) [Zhang et al. 2021a] for shot text clustering, wherein cluster-level attention is proposed to capture the correlations between text representations and cluster representations. As shown in Fig. 3, the representation learning and clustering for short texts are seamlessly integrated into a unified model. To further ensure robust model training for short texts, they apply adversarial training to the unsupervised clustering setting, by injecting perturbations into the cluster representations. The model parameters and perturbations are optimized alternately through a minimax game.

6 GENERATIVE ADVERSARIAL NETWORKS FOR CLUSTERING METHODS

6.1 DAC

They develop a Deep Adversarial gaussian mixture auto-encoder for clustering (DAC) [Harchaoui et al. 2017]. Feature representation for clustering purposes consists in building an explicit or implicit mapping of the input space onto a feature space that is easier to cluster or classify. This paper relies upon an adversarial auto-encoder as a

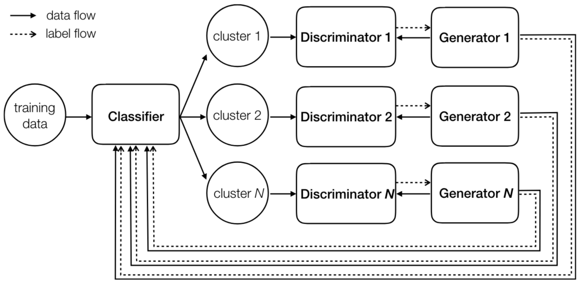

Fig. 2. The GANMM Architecture [Yu and Zhou 2018].

means of building a code space of low dimensionality suitable for clustering. They impose a tunable Gaussian mixture prior over that space allowing for a simultaneous optimization scheme.

6.2 GANMM

For data clustering, Gaussian mixture model (GMM) is a typical method that trains several Gaussian models to capture the data. Each Gaussian model then provides the distribution information of a cluster. For clustering of high dimensional and complex data, more flexible models rather than Gaussian models are desired. The generative adversarial networks (GANs) have shown effectiveness in capturing complex data distribution. Therefore, GAN mixture model (GANMM) [Yu and Zhou 2018] would be a promising alternative of GMM. The paper notices that the non-flexibility of the Gaussian model is essential in the expectation-maximization procedure for training GMM. GAN can have much higher flexibility, which disables the commonly employed expectation-maximization procedure, as that the maximization cannot change the result of the expectation. As shown in Fig. 2, they propose to use the $\epsilon - expectation - maximization$ procedure for training GANMM.

6.3 ClusterGAN

Generative Adversarial networks (GANs) have obtained remarkable success in many unsupervised learning tasks and unarguably, clustering is an important unsupervised learning problem. While one can potentially exploit the latentspace back-projection in GANs to cluster, they demonstrate that the cluster structure is not retained in the GAN latent space. In this paper, they propose ClusterGAN [Mukherjee et al. 2019] as a new mechanism for clustering using GANs. By sampling latent variables from a mixture of one-hot encoded variables and continuous latent variables, coupled with an inverse network (which projects the data to the latent space) trained jointly with a clustering specific loss, they are able to achieve clustering in the latent space.

To sum up, ClusterGAN is recently proposed to leverage GANs to encode continuous data to discrete cluster labels.

7 EXPERIMENT

7.1 Datasets

7.1.1 TREC. The dataset is from the Text REtrieval Conference on 2011-2015 tweet tracks [Yin and Wang 2014]. They are organized by their corresponding queries and evaluated into several relevance levels. Studies retain tweets labeled relevant or highly-relevant to their queries to ensure the quality of labels. You can get access to the dataset by https://github.com/jackyin12/MStream.

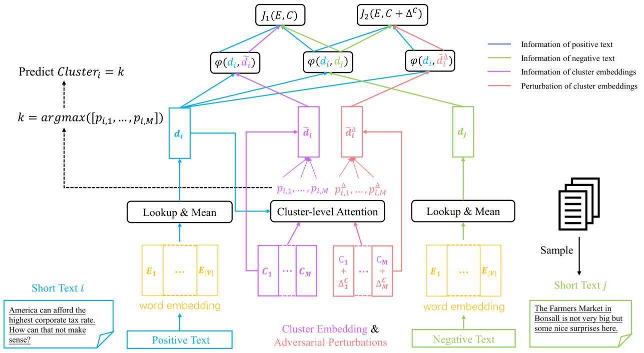
$J_1(E,C)$
$J_2(E,C+\Delta^C)$
$\varphi(d_i,\tilde{d}_i)$
$\varphi(d_i,d_j)$
$\varphi(d_i,\tilde{d}_i^\Delta)$
Information of positive text
Information of negative text
Information of cluster embeddings
Perturbation of cluster embeddings
Predict $Cluster_i = k$
$k = argmax([p_{i,1},\dots,p_{i,M}])$
d_i
$\tilde{d}_i$
$\tilde{d}_i^\Delta$
d_j
$p_{i,1},\dots,p_{i,M}$
$p_{i,1}^\Delta,\dots,p_{i,M}^\Delta$
Lookup & Mean
Cluster-level Attention
Lookup & Mean
Sample
E_1
...
$E_{|V|}$
C_1
...
C_M
$C_1 + \Delta_1^C$
...
$C_M + \Delta_M^C$
E_1
...
$E_{|V|}$
Short Text i
Short Text j
America can afford the highest corporate tax rate. How can that not make sense?
word embedding
Positive Text
Cluster Embedding &
Adversarial Perturbations
word embedding
Negative Text
The Farmers Market in Bonsall is not very big but some nice surprises here.

Fig. 3. Architecture of the proposed ARL-Adv [Zhang et al. 2021a]. Four types of marks with different colors are used to label different information flow. The dotted lines in the middle left of the figure denote how the cluster label of short text i is derived after the model is well trained.

7.1.2 GoogleNews. Similar to [Banerjee et al. 2007], Google News is one of the labeled datasets to evaluate the clustering performance [Yin and Wang 2014]. In the Google News, the news articles are grouped into clusters (stories) automatically. We took a snapshot of the Google News on November 27, 2013, and crawled the titles and snippets of 11,109 news articles belonging to 152 clusters. We manually examined this dataset, and found that it is with really good quality (Almost all articles in the same cluster are about the same event, and articles in different clusters are about different events). You can get access to the dataset by http://news.google.com. In a prior work [Yin and Wang 2014], they further divided the dataset into three datasets: TitleSet (GooglenewsT), SnippetSet (GooglenewsS), and TitleSnippetSet (GooglenewsTS). The GooglenewsT and GooglenewsS only contain the titles and snippets, respectively, while the GooglenewsTS contains both the titles and snippets. They use these three datasets to test the performance of different clustering methods on short texts with different length.

7.1.3 Event. Studies extract event-related tweets from an off-the-shelf tweet dataset crawled in 20163 [Yin and Wang 2014]. Prior knowledge about the events, including the time window, relevant entities, and keywords, is fetched from Wikipedia. You can get access to the dataset by https://archive.org/.

7.1.4 StackOverflow. It is created based on the questions posted in Stack Overflow [Phan et al. 2008]. ARL [Zhang et al. 2021a] require each of the selected questions to be associated with only one tag. And the tags are regarded as the ground-truth cluster labels for the questions. The created dataset is substantially larger than the above three datasets, hoping to provide a more convincing empirical study. For simplicity, we sometimes use the name SO to denote this dataset. You can get access to the dataset by https://archive.org/download/stackexchange.

7.1.5 TweetSet. In the 2011 and 2012 microblog tracks at Text REtrieval Conference (TREC), totally 109 queries were used [Yin and Wang 2014]. Using a standard polling strategy, the NIST assessors evaluated the tweets submitted for each query by the participants into: spam, not relevant, relevant, and highly-relevant. We regard

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.

the queries as clusters and the highly-relevant tweets of each query as documents in each cluster. After removing the queries with none highly-relevant tweets, we constructed a dataset with 89 clusters and totally 2,472 tweets. You can get access to the dataset by http://trec.nist.gov/data/microblog.html.

7.1.6 SearchSnippets. SearchSnippets is a text collection composed of Google search snippets on 8 different topics [Rakib et al. 2020]. The texts in the SearchSnippets dataset represent sets of keywords, rather than being coherent texts.

7.1.7 Biomedical. The Biomedical corpus is a subset of one of the BioAsQ(http://bioasq.org) challenge datasets. The texts in this dataset are paper titles with many special terms from biology and medicine. The Stack Overflow is a subset of the challenge on Kaggle and contains texts with question titles. (Self-taught convolutional neural networks for short text clustering)

7.1.8 UCLnews. The UCInews dataset we used is a subset of the UCI News Aggregator dataset [Fu et al. 2022]. The "STORY" identifier is treated as a label, and we use only the 30 stories with the most documents.

7.1.9 AgNews. AG News is a subset of the dataset that was used in [Zhang and LeCun 2015], where 2000 samples from each of the four categories were taken randomly.

7.2 Evaluate Metrics

To measure the performance of methods, prior studies utilize several metrics [Zhang et al. 2021b], the normalized mutual information (NMI) [Strehl and Ghosh 2002], ajusted rand index (ARI) [Hubert and Arabie 1985], and clustering accuracy (ACC) [Xie et al. 2016].

7.2.1 NMI. Normalized mutual information is defined as:

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

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

7.2.2 ARI. Ajusted rand index is defined as:

$$ARI = \frac{\sum_{ij}\left(\begin{array}{c}\bar{v}_{ij}\\2\end{array}\right) - \left[\sum_i\left(\begin{array}{c}v_i\\2\end{array}\right)\sum_j\left(\begin{array}{c}\widetilde{v}_j\\2\end{array}\right)\right]/n}{\frac{1}{2}\left[\sum_i\left(\begin{array}{c}v_i\\2\end{array}\right)+\sum_j\left(\begin{array}{c}\widetilde{v}_j\\2\end{array}\right)\right] - \left[\sum_i\left(\begin{array}{c}v_i\\2\end{array}\right)\sum_j\left(\begin{array}{c}\widetilde{v}_j\\2\end{array}\right)\right]/n}$$

Suppose v_i denotes the number of short texts in the i-th true topic, and $\widetilde{v}_j$ represents the number of short texts in the j-th inferred cluster. We use $\bar{v}_{i,j}$ to denote the number of short texts simultaneously appearing in two clusters.

7.2.3 ACC. Accuracy is defined as:

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

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

8 CONCLUSION

Analyzing short texts infers discriminative and coherent latent topics that is a critical and fundamental task since many real-world applications require semantic understanding of short texts. Traditional long text topic modeling algorithms (e.g., PLSA and LDA) based on word co-occurrences cannot solve this problem very well since only very limited word co-occurrence information is available in short texts. Therefore, short text topic modeling has already attracted much attention from the machine learning research community in recent years, which aims at overcoming the problem of sparseness in short texts. In this survey, we conduct a comprehensive review of various short text topic modeling techniques. We present four categories of methods based on similarity, topic model, deep learning, and generative adversarial networks, with example of representative approaches in each category.

REFERENCES

Mihael Ankerst, Markus M Breunig, Hans-Peter Kriegel, and Jörg Sander. 1999. OPTICS: Ordering points to identify the clustering structure. *ACM Sigmod record* 28, 2 (1999), 49–60.

Somnath Banerjee, Krishnan Ramanathan, and Ajay Gupta. 2007. Clustering short texts using wikipedia. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval.* 787–788.

David M Blei. 2012. Probabilistic topic models. *Commun. ACM* 55, 4 (2012), 77–84.

David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research* 3, Jan (2003), 993–1022.

Xueqi Cheng, Xiaohui Yan, Yanyan Lan, and Jiafeng Guo. 2014. Btm: Topic modeling over short texts. *IEEE Transactions on Knowledge and Data Engineering* 26, 12 (2014), 2928–2941.

Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise.. In *kdd*, Vol. 96. 226–231.

Bo Fu, Ximing Li, Yuanyuan Guan, and Zhongxuan Luo. 2022. Anomaly Aware Symmetric Non-negative Matrix Factorization for Short Text Clustering. (2022).

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

Warith Harchaoui, Pierre-Alexandre Mattei, and Charles Bouveyron. 2017. Deep adversarial Gaussian mixture auto-encoder for clustering. (2017).

Matthew Hoffman, Francis Bach, and David Blei. 2010. Online learning for latent dirichlet allocation. *advances in neural information processing systems* 23 (2010).

Thomas Hofmann. 1999. Probabilistic latent semantic indexing. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval.* 50–57.

Thomas Hofmann. 2013. Probabilistic latent semantic analysis. *arXiv preprint arXiv:1301.6705* (2013).

Lawrence Hubert and Phipps Arabie. 1985. Comparing partitions. *Journal of classification* 2, 1 (1985), 193–218.

Zhuxi Jiang, Yin Zheng, Huachun Tan, Bangsheng Tang, and Hanning Zhou. 2016. Variational deep embedding: An unsupervised and generative approach to clustering. *arXiv preprint arXiv:1611.05148* (2016).

Ou Jin, Nathan N Liu, Kai Zhao, Yong Yu, and Qiang Yang. 2011. Transferring topical knowledge from auxiliary long texts for short text clustering. In *Proceedings of the 20th ACM international conference on Information and knowledge management.* 775–784.

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

Tianyi Lin, Wentao Tian, Qiaozhu Mei, and Hong Cheng. 2014. The dual-sparse topic model: mining focused topics and focused terms in short text. In *Proceedings of the 23rd international conference on World wide web.* 539–550.

Zongyang Ma, Aixin Sun, Quan Yuan, and Gao Cong. 2012. Topic-driven reader comments summarization. In *Proceedings of the 21st ACM international conference on Information and knowledge management.* 265–274.

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

Rishabh Mehrotra, Scott Sanner, Wray Buntine, and Lexing Xie. 2013. Improving lda topic models for microblogs via tweet pooling and automatic labeling. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval.* 889–892.

Sudipto Mukherjee, Himanshu Asnani, Eugene Lin, and Sreeram Kannan. 2019. Clustergan: Latent space clustering in generative adversarial networks. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 33. 4610–4617.

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

Jipeng Qiang, Ping Chen, Wei Ding, Tong Wang, Fei Xie, and Xindong Wu. 2016. Topic discovery from heterogeneous texts. In *2016 IEEE 28th International Conference on Tools with Artificial Intelligence (ICTAI).* IEEE, 196–203.

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*. Springer, 105–117.

Tian Shi, Kyeongpil Kang, Jaegul Choo, and Chandan K Reddy. 2018. Short-text topic modeling via non-negative matrix factorization enriched with local word-context correlations. In *Proceedings of the 2018 World Wide Web Conference.* 1105–1114.

Bharath Sriram, Dave Fuhry, Engin Demir, Hakan Ferhatosmanoglu, and Murat Demirbas. 2010. Short text classification in twitter to improve information filtering. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval.* 841–842.

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

Xuanhui Wang, ChengXiang Zhai, Xiao Hu, and Richard Sproat. 2007. Mining correlated bursty topic patterns from coordinated text streams. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining.* 784–793.

Jianshu Weng, Ee-Peng Lim, Jing Jiang, and Qi He. 2010. Twitterrank: finding topic-sensitive influential twitterers. In *Proceedings of the third ACM international conference on Web search and data mining.* 261–270.

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

Pengtao Xie and Eric P Xing. 2013. Integrating document clustering and topic modeling. *arXiv preprint arXiv:1309.6874* (2013).

Pengtao Xie, Diyi Yang, and Eric Xing. 2015. Incorporating word correlation knowledge into topic modeling. In *Proceedings of the 2015 conference of the north American chapter of the association for computational linguistics: human language technologies.* 725–734.

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

Xiaohui Yan, Jiafeng Guo, Yanyan Lan, and Xueqi Cheng. 2013. A biterm topic model for short texts. In *Proceedings of the 22nd international conference on World Wide Web.* 1445–1456.

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.* 233–242.

Yang Yu and Wen-Ji Zhou. 2018. Mixture of GANs for Clustering.. In *IJCAI.* 3047–3053.

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

Wei Zhang, Chao Dong, Jianhua Yin, and Jianyong Wang. 2021a. Attentive representation learning with adversarial training for short text clustering. *IEEE Transactions on Knowledge and Data Engineering* (2021).

Xiang Zhang and Yann LeCun. 2015. Text understanding from scratch. *arXiv preprint arXiv:1502.01710* (2015).

Received 18 December 2022; revised 18 December 2022; accepted 18 December 2022