Improving Deep Embedded Clustering via Learning Cluster-level Representations

Qing Yin[21], Zhihua Wang[34], Yunya Song[5], Yida Xu[6], Shuai Niu[2], Liang Bai[7], Yike Guo[2] and Xian Yang[1*]

[1] Alliance Manchester Business School, The University of Manchester
[2] Department of Computer Science, Hong Kong Baptist University
[3] Shanghai Institute for Advanced Study of Zhejiang University
[4] Shanghai AI Laboratory
[5] Department of Journalism, Hong Kong Baptist University
[6] Department of Mathematics, Hong Kong Baptist University
[7] School of Computer and Information Technology, Shanxi University

Abstract

Driven by recent advances in neural networks, various Deep Embedding Clustering (DEC) based short text clustering models are being developed. In these works, latent representation learning and text clustering are performed simultaneously. Although these methods are becoming increasingly popular, they use pure cluster-oriented objectives, which can produce meaningless representations. To alleviate this problem, several improvements have been developed to introduce additional learning objectives in the clustering process, such as models based on contrastive learning. However, existing efforts rely heavily on learning meaningful representations at the instance level. They have limited focus on learning global representations, which are necessary to capture the overall data structure at the cluster level. In this paper, we propose a novel DEC model, which we named the deep embedded clustering model with cluster-level representation learning (DEC-CRL) to jointly learn cluster and instance level representations. Here, we extend the embedded topic modelling approach to introduce reconstruction constraints to help learn cluster-level representations. Experimental results on real-world short text datasets demonstrate that our model produces meaningful clusters.

1 Introduction

Short Text Clustering has gained increasing attention in many real-world applications, such as event discovery (Atefeh and Khreich, 2015), spam detection (Wu and Liu, 2018), and sentiment analysis (Paltoglou and Thelwall, 2012). Unlike long texts, which can be represented as, for instance, term frequency inverse-document-frequency (TF-IDF) vectors in the clustering task, short texts cannot be encoded in the same manner. This is because the vector representation of short texts is highly sparse, making it difficult to measure the similarity between two sets of short texts (Xu et al., 2017). With this observation, deep clustering methods are being developed to encode raw short texts into latent representational space using neural networks and to detect clusters based on the learned representations. Deep clustering methods generally fall into two categories: the two-stage methods and the deep embedded clustering (DEC) methods. The two-stage methods (Zakaria et al., 2012; Tian et al., 2014; Vincent et al., 2010) assign data samples to different clusters after latent representations are learned and fixed, while the DEC methods (Xie et al., 2016) simultaneously learn latent representations and discover clusters via end-to-end training. Different from the two-stage methods, the DEC methods explicitly define the cluster-oriented loss to jointly map raw data into latent representations and acquire cluster assignments.

As discussed in (Jiang et al., 2016; Xie et al., 2016; Aljalbout et al., 2018), significant improvement in clustering performance can be achieved by learning better representations of texts. However, it has been increasingly found that purely cluster-oriented loss driven methods tend to generate meaningless representations (Guo et al., 2017). The semantic meaning of raw data cannot be preserved in the latent space, which would, in turn, deteriorate the performance of clustering. To tackle this problem, sequence-to-sequence (seq2seq) based reconstruction models have been widely used to learn general representations from texts in an unsupervised manner. For example, (Kiros et al., 2015) generated the text representations by predicting the context sentences of a given sentence. The work in (Brahma, 2018) learned text representations by predicting

*This is the corresponding author.

Proceedings of the 29th International Conference on Computational Linguistics, pages 2226–2236
October 12–17, 2022.

multiple future sentences based on the seq2seq model. Recently, instance-wise contrastive learning has achieved remarkable success in representation learning by adopting the contrastive loss along with the cluster-oriented loss (Li et al., 2021; Tsai et al., 2020; Van Gansbeke et al., 2020; Zhang et al., 2021). However, the aforementioned methods focus on optimizing the representation at the instance level. For example, the contrastive learning-based methods heavily rely on the instance discrimination (Li et al., 2021) such that their learning objectives (i.e. instance-wise loss) do not perfectly align with the ultimate goal of clustering. In the clustering task, the final clustering performance heavily relies on learning representations which are capable of reflecting the overall semantic structure of data. Instance-level representation learning methods cannot guarantee that the structure of data can be easily obtained through clustering.

In this paper, we aim to develop a novel DEC method which learns cluster-level as well as instance-level representations to better capture the semantic data structure for clustering. In our approach, the cluster-level representations are defined as the representations of cluster centres. Different from the SCCL model (Zhang et al., 2021) which learns centre representations without imposing any direct constraints, we adopt the reconstruction constraints (Ma et al., 2019) to encode the whole set of raw texts into latent representations of cluster centres and then use these centre representations to reconstruct the text data. As reconstructing the whole dataset from a limited number of cluster-level representations is quite challenging, we designed a cluster-level representation learning (CRL) module to help the representations of cluster centres participate in the process of reconstructing input instances. More specifically, we extend the idea of embedded topic modelling (ETM) (Dieng et al., 2020) to reconstruct words from the representations of topics in latent space. The representations of cluster centres will be integrated into the topic representations, which will then be learned by optimizing both ETM guided reconstruction and clustering objectives.

Our proposed model, named as deep embedded clustering model with cluster-level representation learning (DECCRL), consists of three modules: an instance-level encoding module that maps the original data inputs into latent representations; a cluster selection module that generates cluster labels; and a CRL model that learns the cluster-level representations through reconstruction. Our main contributions are summarized as follows:

- We develop a novel deep embedded clustering method to learn cluster-level as well as instance-level representations to better capture the data structure.
- We extend the idea of embedded topic modelling to impose reconstruction constraints to the cluster-level representations.
- The experimental results show that our method achieves the best clustering performance compared with current state-of-the-art short text clustering methods.

In the remainder of this paper, we first summarize related works in Section 2. We formulate the problem and explain our DECCRL model in Section 3. The experiments are introduced in Sections 4 and 5 and the paper is concluded in Section 6.

2 Related Work

2.1 Deep Embedded Clustering

Deep clustering (Xie et al., 2016; Wu et al., 2019) applies deep neural networks to transform raw inputs into latent representations, based on which clustering is performed. Traditional approaches derive latent representations first, and then clusters are detected (Zakaria et al., 2012; Tian et al., 2014; Vincent et al., 2010). However, latent representations learned by these approaches are not cluster-oriented in that they are learned before hand. The deep embedded clustering (DEC) methods are then developed to simultaneously generate latent representations and cluster assignments through the end-to-end training (Dosovitskiy et al., 2015; Caron et al., 2018; Asano et al., 2019; Ghasedi et al., 2019; Yang et al., 2020).

However, methods that purely depend on the cluster-oriented loss cannot well preserve the local structure of raw data and are likely to generate corrupted latent space (Guo et al., 2017). To address the above-mentioned problem, researchers recently introduced extra reconstruction modules along with the clustering model. For example, to cluster images, Jiang et al. (2016); Madiraju (2018); Yang et al. (2019) adopted auto-encoders to learn latent representations and simultaneously perform clustering using the latent representations from auto-

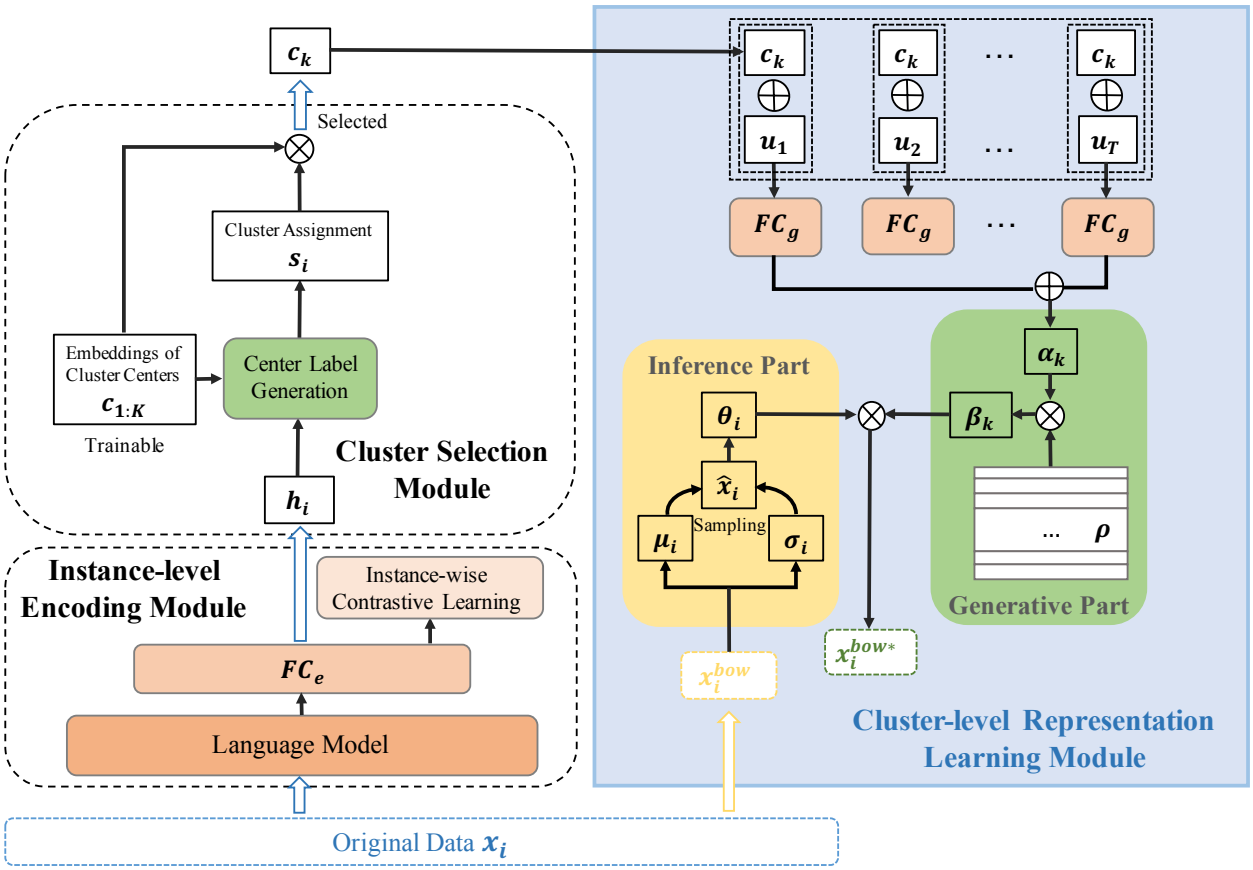


Figure 1: The overview of the proposed deep embedded clustering model with cluster-level representation learning (DECCRL). It contains three components: an instance-level encoding module, a cluster selection module and a cluster-level representation learning module based on the embedded topic modelling. Some key variables are described as follows: $\boldsymbol{c}_{1:K}$ contains embeddings of cluster centers; $\boldsymbol{\alpha}_k$ refers to embeddings of topics in the k-th cluster; $\boldsymbol{\theta}_i$ represents topic proportions of the i-th input text; $\boldsymbol{\rho}$ contains word embeddings of the vocabulary; $\boldsymbol{\beta}_k$ is the distribution over the vocabulary for topics within the k-th cluster.

encoders. To cluster time-series data, Ma et al. (2019) leveraged a seq2seq model to guide the generation of latent representations. To cluster text data, Zhao et al. (2021) utilized the idea of data reconstruction to reconstruct data in the latent space only other than the original space. In the same time, there are many seq2seq models have been used to assist text data clustering like (Kiros et al., 2015) generated the text representations by predicting the context sentences of a given sentence and the work in (Brahma, 2018) learned text representations by predicting multiple future sentences based on the seq2seq model.

2.2 Neural Topic Modelling in Various NLP tasks

With the development of neural networks, there has been a surge of methods that seek to combine deep neural networks with probabilistic topic models (Srivastava and Sutton, 2017; Cong et al., 2017; Zhang et al., 2018). Most of these methods used amortized inference and variational auto-encoder to reduce the dimension of the input data (Rezende et al., 2014; Dieng et al., 2020). For example, ETM (Dieng et al., 2020) is a neural topic model that uses word embeddings from Word2Vec (Mikolov et al., 2013). Neural topic modelling is not only used to learn hidden topics from a collection of texts, but has also been increasingly used to assist other NLP tasks. For example, See et al. (2017); Ailem et al. (2019); Wang et al. (2020) combined seq2seq models with topic models in the abstract generation task. (Dieng et al., 2016) incorporated topic modelling with the recurrent neural network to capture the long-range dependencies. Zeng et al. (2018); Wang and Yang (2020) integrated NTM with a memory network for short text classification. Tang et al. (2019); Wang et al. (2019) used neural topic modelling to assist text generation. Our work aims to improve clustering by incorporating ETM.

3 Method

The overview of the proposed deep embedded clustering model with cluster-level representation learning (DECCRL) is shown in Figure 1. It contains three components: an instance-level encoding module, a cluster selection module and a cluster-level representation learning module (CRL) based on the embedded topic modelling. The instance-level encoding module generates the latent representations of texts; the cluster selection module takes the outputs of the encoding module as the input to generate the cluster assignments; CRL attempts to optimise the overall structure of data by connecting the latent representations of cluster centres with the embedded topic model. More detailed descriptions of our method will be provided in the following.

3.1 The Instance-level Encoding Module

In the instance-level encoding module, we aim to generate optimised instance-level representation by contrastive learning. Suppose the inputs of the encoding module include: the original texts $\mathcal{B} = \{\boldsymbol{x}_i\}_{i=1}^{M}$ and its corresponding augmentation set $\mathcal{B}^a = \{(\tilde{\boldsymbol{x}}_{i_1}, \tilde{\boldsymbol{x}}_{i_2})\}_{i=1}^{M}$, where $\boldsymbol{x}_i$ is a sample of input texts, M is the number of samples, and $(\tilde{\boldsymbol{x}}_{i_1}, \tilde{\boldsymbol{x}}_{i_2})$ contains augmented versions of $\boldsymbol{x}_i$ to enable contrastive learning. We will apply the *Contextual Augmenter* (Kobayashi, 2018), which utilizes the pre-trained transformer-based models to find suitable words for synonym substitution (Kobayashi, 2018): Bertbase (Devlin et al., 2018) and Roberta (Liu et al., 2019) are used for generating $\tilde{\boldsymbol{x}}_{i_1}$ and $\tilde{\boldsymbol{x}}_{i_2}$, respectively.

As shown in the lower left part of Figure 1, a language model followed by a fully connected neural network FC_e are used to map the data from the original space $\mathcal{X}$ to latent space $\mathcal{H}$. Here, Sentence-BERT (Reimers and Gurevych, 2019) is chosen as the language model since it has fine-tuned BERT (Devlin et al., 2018) for better measuring sentence similarities which would suit short text clustering. The outputs of the encoding module can be represented as:

$$\begin{aligned} &\boldsymbol{h}_i = FC_e(SentenceBERT(\boldsymbol{x}_i)), \\ &\tilde{\boldsymbol{h}}_{i_j} = FC_e(SentenceBERT(\tilde{\boldsymbol{x}}_{i_j})), \\ &i \in \{1, ..., M\}, j \in \{1, 2\}. \end{aligned} \tag{1}$$

In order to optimise instance-level representations in latent space, we follow the work in (Zhang et al., 2021) and introduce the same contrasting module to DECCRL for leveraging the power of contrastive learning. For any $\boldsymbol{x}_i$ in $\mathcal{B}$, we refer to its augmented versions, $\tilde{\boldsymbol{x}}_{i_1}$ and $\tilde{\boldsymbol{x}}_{i_2}$ in $\mathcal{B}^a$, as the positive pair, while treating the other elements in sample pairs of $\mathcal{B}^a$ as negative instances. The contrasting module adopts fully connected neural networks to transform latent representations $\tilde{\boldsymbol{h}}_{i_j}$ into $\tilde{\boldsymbol{v}}_{i_j}$.

The contrastive loss is defined to make the positive samples closer and negative samples further apart from each other as follows:

$$l_{i_1}^{CL} = -log\frac{exp(sim(\tilde{\boldsymbol{v}}_{i_1}, \tilde{\boldsymbol{v}}_{i_2})/\tau)}{\sum_{m=1}^{M} \mathbb{1}_{m \neq i} \cdot exp(sim(\tilde{\boldsymbol{v}}_{i_1}, \tilde{\boldsymbol{v}}_{m_2})/\tau)}, \tag{2}$$

$$l_{i_2}^{CL} = -log\frac{exp(sim(\tilde{\boldsymbol{v}}_{i_2}, \tilde{\boldsymbol{v}}_{i_1})/\tau)}{\sum_{m=1}^{M} \mathbb{1}_{m \neq i} \cdot exp(sim(\tilde{\boldsymbol{v}}_{i_2}, \tilde{\boldsymbol{v}}_{m_1})/\tau)}, \tag{3}$$

where $\mathbb{1}_{m \neq i}$ is an indicator function, τ denotes the temperature parameter, and $sim(.)$ measures the cosine similarity between two vectors (Chen et al., 2020). The contrastive loss averaged across samples is:

$$\mathcal{L}_{CL} = \sum_{i=1}^{M} (l_{i_1}^{CL} + l_{i_2}^{CL})/M. \tag{4}$$

3.2 The Cluster Selection Module

The *cluster selection module* is designed to assign each data sample to a certain cluster as shown in the upper left of Figure 1. Assume there are K clusters, where each cluster can be characterized by its centroid $\boldsymbol{c}_k$ for $k \in \{1, ..., K\}$ in the latent space $\mathcal{H}$. Following the approach developed in Van der Maaten and Hinton (2008); Zhang et al. (2021), we calculate the probability of assigning the i-th input text $\boldsymbol{x}_i$ to the k-th cluster based on the Student's t-distribution as follows:

$$o_{ik} = \frac{(1 + \|\boldsymbol{h}_i - \boldsymbol{c}_k\|_2^2/\alpha)^{-\frac{\alpha+1}{2}}}{\sum_{k'=1}^{K}(1 + \|\boldsymbol{h}_i - \boldsymbol{c}_{k'}\|_2^2/\alpha)^{-\frac{\alpha+1}{2}}}, \tag{5}$$

where α is the degree of freedom of the Student's t-distribution, and $\boldsymbol{h}_i$ is the latent representation of $\boldsymbol{x}_i$ generated by the *instance-level encoding module* using Eq (1). A $softmax$ layer is then used to normalize o_{ik} as:

$$\boldsymbol{s}_i = softmax([o_{i,1}; ...; o_{i,K}]), \tag{6}$$

from which the cluster assignment of the i-th text sample can be sampled.

To optimize the estimation of the centroid $\boldsymbol{c}_k$ for each cluster, we utilize an auxiliary probability t_{ik} as discussed in Xie et al. (2016):

$$t_{ik} = \frac{o_{ik}^2/f_k}{\sum_{k'} o_{ik}^2/f_{k'}}, \tag{7}$$

where $f_k = \sum_{i=1}^{M} o_{ik}$ is the soft cluster frequency. To make the cluster assignment probability close to the auxiliary probability, we will minimize the KL divergence between them, which is defined as:

$$l_i^C = \sum_{k=1}^{K} t_{ik} log \frac{t_{ik}}{o_{ik}}, \tag{8}$$

The cluster-oriented loss averaged across M samples is:

$$\mathcal{L}_{Cluster} = \sum_{i=1}^{M} l_i^C / M. \tag{9}$$

3.3 The Cluster-level Representation Learning Module

In order to optimise the cluster-level representations by reconstructing, the cluster-level representation learning module (CRL) extend the idea of the embedded topic modelling (ETM) to reconstruct words from cluster center representations. The representations of cluster centroids $\boldsymbol{c}_k$ for $k \in \{1, ..., K\}$ obatined from the *cluster selection module* are used to generate latent representations of topics, from which the text data can be reconstructed. As shown in the right part of Figure 1, the CRL has two main parts: the generative part and the inference part, which will be explained in detail in the following paragraphs.

3.3.1 The Generative Part

Suppose L is the embedding length of vectors in the latent space $\mathcal{H}$. Let us denote the embeddings of words obtained from Word2Vec (Mikolov et al., 2013) as $\boldsymbol{\rho} = [\boldsymbol{\rho}_1, ..., \boldsymbol{\rho}_V]$, where $\boldsymbol{\rho}_v \in \mathbb{R}^{2L}$ is for the v-th word and V is the vocabulary size. The embedding of each topic t from the cluster k is represented as $\boldsymbol{\alpha}_k^t \in \mathbb{R}^{2L}$. In our model, the embeddings of topics from each cluster k are related to the cluster centroid $\boldsymbol{c}_k$ as follows:

$$\boldsymbol{\alpha}_k^t = FC_g(\boldsymbol{c}_k \oplus \boldsymbol{u}_t) \; for \; t \in \{1, .., T\}, \tag{10}$$

where T is the total number of topics, FC_g is a fully connected neural network, $\boldsymbol{u}_t \in \mathbb{R}^L$ is a trainable vector, and $\oplus$ is the concatenation operator. Using $\boldsymbol{\alpha}_k = [\boldsymbol{\alpha}_k^1, ..., \boldsymbol{\alpha}_k^T]$ and $\boldsymbol{\rho}$, the distribution over the vocabulary for topics within the k-th cluster can be obtained from:

$$\boldsymbol{\beta}_k = softmax((\boldsymbol{\rho})' \boldsymbol{\alpha}_k), \tag{11}$$

where $(.)'$ is the matrix transpose operator, and $\boldsymbol{\beta}_k \in \mathbb{R}^{V \times T}$ is a collection of simplexes achieved by computing the semantic similarity between topics and words.

For the i-th text, its topic proportions $\boldsymbol{\theta}_i$, indicating the prevalence of different topics in the text. Let $w_{i,n}$ denote the n-th word in the i-th text, whose topic assignment $z_{i,n}$ is assumed to be drawn from $z_{i,n} \sim Cat(\boldsymbol{\theta_i})$, where $Cat(.)$ denotes the categorical distribution. With $z_{i,n}$ and $\boldsymbol{\beta}_k$, the probability of observing $w_{i,n}$ is then:

$$p(w_{i,n}|\boldsymbol{\beta}_k, z_{i,n}) = Multi(\boldsymbol{\beta}_k^{z_{i,n}}), \tag{12}$$

where $Multi(.)$ is the Multinomial distribution and $\boldsymbol{\beta}_k^{z_{i,n}}$ is the $z_{i,n}$-th column of $\boldsymbol{\beta}_k$. Then, the log marginal likelihood of observing the i-th text can be represented as:

$$\begin{aligned} \log p(w_{i,n}|\hat{\boldsymbol{x}}_i) &= \sum_{z_{i,n}} \log p(w_{i,n}|\boldsymbol{\beta}_k^{z_{i,n}}) p(z_{i,n}|\hat{\boldsymbol{x}}_i) \\ &= \log \boldsymbol{\beta}_k \boldsymbol{\theta}_i, \end{aligned} \tag{13}$$

where $\boldsymbol{\theta}_i = softmax(\hat{\boldsymbol{x}}_i)$, and $\hat{\boldsymbol{x}}_i$ will be approximated from the BoW form of original text $\boldsymbol{x}_i$ as to be explained in the following paragraphs.

3.3.2 The Inference Part

Let us define the approximated distribution of $\hat{\boldsymbol{x}}_i$ as $q(\hat{\boldsymbol{x}}_i|\boldsymbol{x}_i^{bow*})$, where $\boldsymbol{x}_i^{bow*}$ is the normalized representation of BoWs data $\boldsymbol{x}_i^{bow}$. In the inference part, $\boldsymbol{x}_i^{bow*}$ is first passed through a fully connected neural network to get its latent representation, which is then fed into two parallel fully connected neural networks to get two vectors: $\boldsymbol{\mu}_i$ and $\boldsymbol{\sigma}_i$. By treating $\boldsymbol{\mu}_i$ and $\boldsymbol{\sigma}_i$ as the mean and standard deviation, $\hat{\boldsymbol{x}}_i$ can be sampled from:

$$\hat{\boldsymbol{x}}_i = \boldsymbol{\mu}_i + \boldsymbol{\epsilon} \cdot \boldsymbol{\sigma}_i, \tag{14}$$

where $\boldsymbol{\epsilon} \in \mathcal{N}(0, \mathbf{I})$. In our CRL, we choose the negative evidence lower bound as the cluster-level

representation loss:

$$\mathcal{L}_{RC} = -\mathbb{E}_{q(\hat{\boldsymbol{x}}_i|\boldsymbol{x}_i^{bow*})}[\sum_{n=1}^{N_i} \log p(w_{i,n}|\hat{\boldsymbol{x}}_i)] + D_{KL}[q(\hat{\boldsymbol{x}}_i|\boldsymbol{x}_i^{bow*})||p(\hat{\boldsymbol{x}}_i)], \quad (15)$$

where N_i is the number of words in the ith text, and $p(\hat{\boldsymbol{x}}_i)$ is the prior distribution of $\hat{\boldsymbol{x}}_i$, assumed to be normally distributed.

To train our model, the overall optimization objective is defined as:

$$\mathcal{L} = \mathcal{L}_{CL} + \lambda_c * \mathcal{L}_{Cluster} + \lambda_r * \mathcal{L}_{RC}, \quad (16)$$

where λ_c and λ_r are the weights of $\mathcal{L}_{Cluster}$ and $\mathcal{L}_{RC}$, respectively.

4 Experimental Setup

4.1 Dataset

We evaluate our model using three benchmark datasets. Table 1 briefly summarizes them with some details elaborated as follows.

- **AgNews** is a collection of news titles (Zhang and LeCun, 2015). In our experiment, we use a subset version from (Rakib et al., 2020), which contains 8,000 documents from four different categories. For performance evaluation, 6,400 documents are used for training while 1,600 documents are used for testing.
- **StackOverflow** is a subset of the challenge data released by Kaggle[1]. This dataset contains 20,000 documents, which can fall into 20 different categories (Xu et al., 2017). For model training and testing, 15,084 and 4,916 documents are used respectively.
- **Biomedical** is the challenge data published in BioASQ[2]. The version provided by Xu et al. (2017) is used in our experiment, which contains 20,000 paper titles from 20 categories.

4.2 Baseline

- **BoW & TF-IDF** (Zhang et al., 2021) together with the K-means clustering is used as a baseline method, where the length of BoW or TF-IDF vectors is set to 1,500.
- **STCC** (Xu et al., 2017) is a typical two-step deep clustering method. It first used Word2Vec (Mikolov et al., 2013) to embed words in the original text. The resulting word embeddings are fed into convolutional neural networks to get latent representations. Then, K-means is used to detect clusters using representations obtained from the previous step.
- **HAC-SD** (Rakib et al., 2020) introduces iterative classification to boost the performance of clustering. It considers outlier removal to generate outlier-free clusters for short texts. The outlier removed data is used to train a classification algorithm based on the cluster assignments.
- **SCCL** (Zhang et al., 2021) is a state-of-the-art deep embedded clustering method for short texts, which leverages the power of contrastive learning to improve clustering.

4.3 Settings

In our approach, we use the Adam optimizer (Kingma and Ba, 2014) with the batch size of 200. We choose *distilbert-base-nli-stsb-mean-tokens* for SentenceBERT and set the maximum input length to 32. Same as Zhang et al. (2021), we set α = 10.0 for the Biomedical dataset and α = 1.0 for the other datasets. The temperature parameter used in the contrasting module is 0.5. As used in the recent works for short text clustering (Xu et al., 2017; Rakib et al., 2020; Zhang et al., 2021), we adopt the clustering accuracy (ACC) (Xie et al., 2016) and the normalized mutual information (NMI) (Strehl and Ghosh, 2002) to show the performance of clustering models. For fair comparison, supervised pre-trained models are not applied. Since most existing works have pre-defined cluster numbers and reported results, we adopt this practice and follow their training/test protocols stated in their paper.

5 Experimental Result

5.1 Clustering performance compared with baselines

Table 2 shows the results of baseline methods along with our model on 3 benchmark datasets. The observations can be summarized as follows. First, the deep clustering methods, including STCC, SCCL and DECCRL, outperform conventional clustering methods which are based on BoW or TF-IDF for feature extraction and k-means for cluster detec-

[1]https://www.kaggle.com/c/predict-closed-questions-on-stackoverflow/download/train.zip

[2]http://participants-area.bioasq.org/

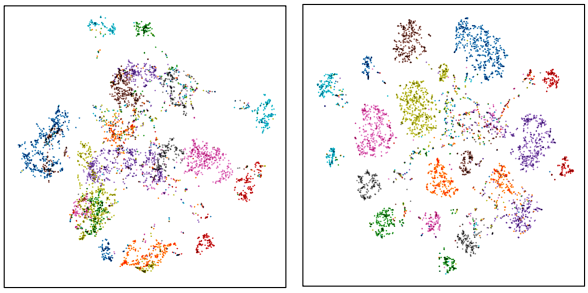
Table 1: Summary statistics of three benchmark datasets.

Dataset	# Docs	# Training	# Test	# Words	# Classes	# Average Length
AgNews	8,000	6,400	1,600	21,063	4	23
StackOverflow	20,000	15,084	4,916	10,941	20	8
Biomedical	20,000	15,583	4,417	18,244	20	13

Table 2: The short text clustering results for three benchmark text datasets. Our result are averaged over five random runs. The ACC and NMI values for baseline methods are directly obtained from (Zhang et al., 2021).

Models	AgNews Dataset		StackOverflow Dataset		Biomedical Dataset	
	ACC	NMI	ACC	NMI	ACC	NMI
BoW	27.6	2.6	18.5	14.0	14.3	9.2
TF-IDF	34.5	11.9	58.4	58.7	28.3	23.2
STCC	-	-	51.1	49.0	43.6	38.1
HAC-SD	81.8	54.6	64.8	59.5	40.1	33.5
SCCL(Zhang et al., 2021)	88.2	68.2	75.5	74.5	46.2	**41.5**
DECCRL	**88.9**	**69.2**	**82.3**	**76.7**	**47.0**	**41.5**

tion. HAC-SD, considering outlier removal, shows better performance than conventional k-means clustering approaches. Secondly, SCCL has shown better performance than the other baseline models, reflecting the need of introducing contrastive learning into the clustering models. More importantly, our model outperforms all other baseline models for all datasets, especially for the StackOverflow Dataset. Compared with SCCL, our model has achieved higher ACC and NMI values.



(a) (b)

Figure 2: The TSNE visualization of the latent representations for the StackOverflow dataset, where (a) is from-SCCL and (b) is from DECCRL. Each color indicates a ground-truth cluster category.

To further demonstrate the importance of introducing the CRL into the clustering model, we visualize the distribution of samples from a random subset of StackOverflow (n=4,916) using the t-SNE (Van der Maaten and Hinton, 2008) visualization algorithm. In Figure 2, samples are assigned to different colours based on their ground-truth categories, where the total number of categories is 20. By comparing the results from SCCL and DECCRL under the same settings as shown in Figure 2(a) and Figure 2(b) respectively, we can find that our model were able to learn representations that are more separable in the latent space. With this observation, it is more confident to predicate that the representations generated from our model would lead a better clustering results.

Table 3: The results of different representation guidance strategies. **AGN**, **SO** and **BIO** refer to the AgNews, StackOverflow and Biomedical datasets respectively.

Metric	Model	AGN	SO	BIO
NMI	Ours w/o CRL	59.2	74.0	28.0
	Ours w/o CRL w LSR	62.3	74.5	31.2
	Ours	**69.1**	**76.7**	**41.5**
ACC	Ours w/o CRL	81.5	78.8	29.4
	Ours w/o CRL w LSR	83.9	81.2	31.3
	Ours	**88.9**	**82.3**	**47.0**

5.2 The influence of cluster-level representation learning module

In this section, we investigate the performance of different cluster-level representation learning strategies, which are designed as follows:

- **Ours w/o CRL** – DECCRL without CRL. The model learns cluster-level representation without imposing any direct contraints.

- **Ours w/o CRL w LSR** – Replacing the CRL of DECCRL with a latent space reconstruction module (LSR). The L-dimensional latent representations are fed into a deep neural network, where an encoder generates $L/2$-dimensional vectors and a decoder returns L-dimensional vectors. The difference between the inputs and outputs of this network is considered as an extra loss for model training.

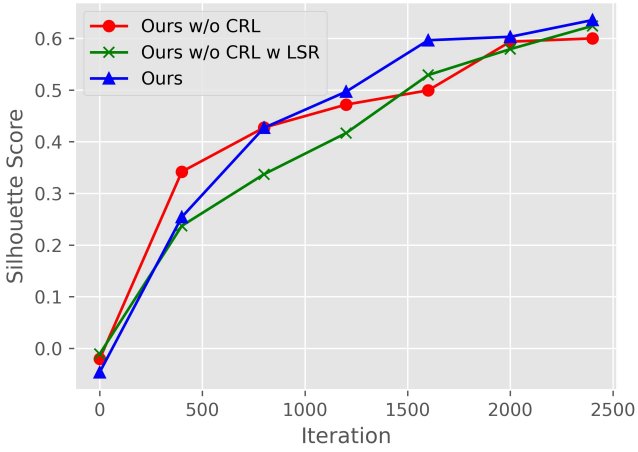


Figure 3: Silhouette scores from different cluster-level representation learning strategies during training process.

Table 3 shows the ACC and NMI values from modes with different cluster-level representation learning strategies. By comparing the performance of DECCRL with DECCRL w/o CRL, we find that the CRL has greatly improved for all three datasets. DECCRL w/o CRL w lSR, which constrains features at the latent level, has shown better performance than DECCRL w/o CRL but no better than DECCRL.

Without referring to the ground-truth labels of clusters, we use a pure clustering metric, Silhouette score (Rousseeuw, 1987), to investigate the performance of different cluster-level representation learning strategies during the training process as shown in Figure 3. The results from Figure 3 and Table 3 show that representations from all models seem to return clusters of similar characteristics (e.g., compactness and separation indicated by the Silhouette score). However, without adopting the proposed CRL, the learned representations cannot well preserve semantic information contained in the original text data such that the ACC and NMI scores generated using other methods are not as high as ours. Given the above observations, we find that using embedded topic modelling to guide the latent representation is a promising strategy.

Table 4: Selected clusters and their corresponding representative hidden topics.

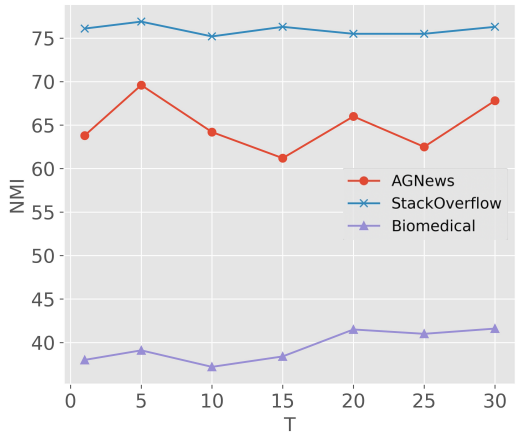
Cluster Label	Representative Topics
osx	Topic1: ['terminal', 'mac', 'command', 'stdin'] Topic2: ['max', 'os', '**osx**', 'console'] Topic3: ['file', 'application', 'set', 'create']
excel	Topic1: ['data', 'xml', 'cell', 'table'] Topic2: ['**excel**', 'list', 'files', 'worksheet'] Topic3: ['file', 'create', 'application', 'xml']
oracle	Topic1: ['**oracle**', 'db', 'view', 'connection'] Topic2: ['sql', 'table', 'data', 'database'] Topic3: ['file', 'application', 'data', 'multiple']

5.3 Understanding clusters

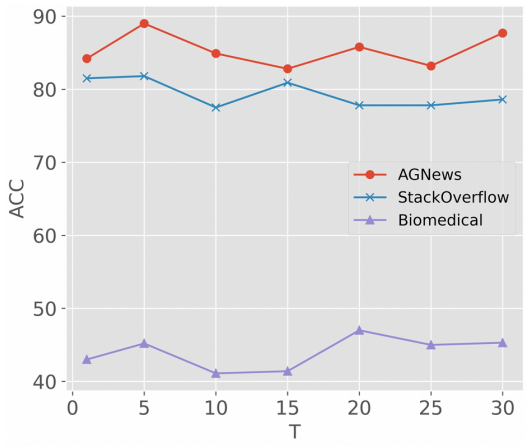
This subsection shows that our embedded topic modelling based CRL does not only improve the clustering performance but can also be used to characterize each cluster using learned topics. Table 4 shows topics learned from three representative clusters from the StackOverflow Dataset whose cluster labels are: 'osx', 'excel' and 'oracle'. These labels are the ground-truth labels provided by the dataset, and indicate the meaning of each cluster at a coarse-grained level. To check whether the learned topics generated from our CRL are consistent with these cluster labels, Table 4 shows some selected topics (characterized by the top four key words) from each cluster. We can find that the class labels can be found as key words of topics. The learned topics can provide more detailed understanding of clusters. For example, for cluster 'oracle', its first topic indicates that 'oracle' might be a 'db' (database) having operations like 'view', 'connection' and 'access'. Apart from these observations, different clusters are also found to have topics with similar meaning. For example, the third topics from three clusters are all about 'file'.

5.4 Clustering performance with different topic numbers

We investigate the impacts of T (i.e. the total number of topics) on the clustering performance. Figure 4 (a) and (b) show the values of NMI and ACC for the three benchmark datasets with the topic number T chosen from $\{1, 5, 10, 15, 20, 25, 30\}$. For the BioMedical Dataset, we can see that the NMI and ACC values become stable when T exceeds 20. Therefore, in the experiment for this dataset, we set the topic number $T = 20$. For the AgNews



(a)



(b)

Figure 4: The clustering performance in terms of (a) NMI and (b) ACC with different topic number T.

and StackOverflow datasets, the highest NMI and ACC values are obtained at $T = 5$. Thus, we set the number of topics to 5 for these two datasets.

6 Conclusion

This paper proposes a deep embedded clustering method for short text clustering by developing a cluster-level representation learning module (CRL) to capture the overall structure of data and hence improve the clustering performance. Our model comprises three main parts: the instance-level encoding, the cluster selection, and CRL. To show the performance of our model, we utilize three benchmark datasets. The clustering performance has not only been quantitatively evaluated by ACC and NMI values but are also qualitatively assessed by case studies and visualization. The comparison of different cluster-level representation strategies shows the effectiveness of our CRL. The proposed model is expected to be generalizable to meet various text clustering challenges, not only limited to short texts. In the future, we will extend our model to capture dynamics changes of cluster centers that might evolve over time, where dynamic ETM learning smooth trajectories of topic embeddings can be considered. Another future research direction is to adopt non-parametric Bayesian approaches (e.g., Dirichlet process mixture model) to improve our clustering model so that the exact number of clusters does not need to be predefined.

7 Acknowledge

This work was supported by the National Key Research and Development Program of China (No. 2021ZD0113303).

References

Melissa Ailem, Bowen Zhang, and Fei Sha. 2019. Topic augmented generator for abstractive summarization. *arXiv preprint arXiv:1908.07026*.

Elie Aljalbout, Vladimir Golkov, Yawar Siddiqui, Maximilian Strobel, and Daniel Cremers. 2018. Clustering with deep learning: Taxonomy and new methods. *arXiv preprint arXiv:1801.07648*.

Yuki Markus Asano, Christian Rupprecht, and Andrea Vedaldi. 2019. Self-labelling via simultaneous clustering and representation learning. *arXiv preprint arXiv:1911.05371*.

Farzindar Atefeh and Wael Khreich. 2015. A survey of techniques for event detection in twitter. *Computational Intelligence*, 31(1):132–164.

Siddhartha Brahma. 2018. Unsupervised learning of sentence representations using sequence consistency. *arXiv preprint arXiv:1808.04217*.

Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. 2018. Deep clustering for unsupervised learning of visual features. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 132–149.

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

Yulai Cong, Bo Chen, Hongwei Liu, and Mingyuan Zhou. 2017. Deep latent dirichlet allocation with topic-layer-adaptive stochastic gradient riemannian mcmc. In *International Conference on Machine Learning*, pages 864–873. PMLR.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep

bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Adji B Dieng, Francisco JR Ruiz, and David M Blei. 2020. Topic modeling in embedding spaces. *Transactions of the Association for Computational Linguistics*, 8:439–453.

Adji B Dieng, Chong Wang, Jianfeng Gao, and John Paisley. 2016. Topicrnn: A recurrent neural network with long-range semantic dependency. *arXiv preprint arXiv:1611.01702*.

Alexey Dosovitskiy, Philipp Fischer, Jost Tobias Springenberg, Martin Riedmiller, and Thomas Brox. 2015. Discriminative unsupervised feature learning with exemplar convolutional neural networks. *IEEE transactions on pattern analysis and machine intelligence*, 38(9):1734–1747.

Kamran Ghasedi, Xiaoqian Wang, Cheng Deng, and Heng Huang. 2019. Balanced self-paced learning for generative adversarial clustering network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4391–4400.

Xifeng Guo, Long Gao, Xinwang Liu, and Jianping Yin. 2017. Improved deep embedded clustering with local structure preservation. In *Ijcai*, pages 1753–1759.

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*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

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

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

Yunfan Li, Peng Hu, Zitao Liu, Dezhong Peng, Joey Tianyi Zhou, and Xi Peng. 2021. Contrastive clustering. In *2021 AAAI Conference on Artificial Intelligence (AAAI)*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Qianli Ma, Jiawei Zheng, Sen Li, and Gary W Cottrell. 2019. Learning representations for time series clustering. *Advances in neural information processing systems*, 32:3781–3791.

Naveen Sai Madiraju. 2018. *Deep temporal clustering: Fully unsupervised learning of time-domain features*. Ph.D. thesis, Arizona State University.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Georgios Paltoglou and Mike Thelwall. 2012. Twitter, myspace, digg: Unsupervised sentiment analysis in social media. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 3(4):1–19.

Md Rashadul Hasan Rakib, Norbert Zeh, Magdalena Jankowska, and Evangelos Milios. 2020. Enhancement of short text clustering by iterative classification. In *International Conference on Applications of Natural Language to Information Systems*, pages 105–117. Springer.

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

Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. 2014. Stochastic backpropagation and approximate inference in deep generative models. In *International conference on machine learning*, pages 1278–1286. PMLR.

Peter J Rousseeuw. 1987. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65.

Abigail See, Peter J Liu, and Christopher D Manning. 2017. Get to the point: Summarization with pointer-generator networks. *arXiv preprint arXiv:1704.04368*.

Akash Srivastava and Charles Sutton. 2017. Autoencoding variational inference for topic models. *arXiv preprint arXiv:1703.01488*.

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

Hongyin Tang, Miao Li, and Beihong Jin. 2019. A topic augmented text generation model: Joint learning of semantics and structural features. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5090–5099.

Fei Tian, Bin Gao, Qing Cui, Enhong Chen, and Tie-Yan Liu. 2014. Learning deep representations for graph clustering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 28.

Tsung Wei Tsai, Chongxuan Li, and Jun Zhu. 2020. Mice: Mixture of contrastive experts for unsupervised image clustering. In *International Conference on Learning Representations*.

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

Wouter Van Gansbeke, Simon Vandenhende, Stamatios Georgoulis, Marc Proesmans, and Luc Van Gool. 2020. Scan: Learning to classify images without labels. In

European Conference on Computer Vision, pages 268–285. Springer.

Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, Pierre-Antoine Manzagol, and Léon Bottou. 2010. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of machine learning research*, 11(12).

Wenlin Wang, Zhe Gan, Hongteng Xu, Ruiyi Zhang, Guoyin Wang, Dinghan Shen, Changyou Chen, and Lawrence Carin. 2019. Topic-guided variational autoencoders for text generation. *arXiv preprint arXiv:1903.07137*.

Xinyi Wang and Yi Yang. 2020. Neural topic model with attention for supervised learning. In *International Conference on Artificial Intelligence and Statistics*, pages 1147–1156. PMLR.

Zhengjue Wang, Zhibin Duan, Hao Zhang, Chaojie Wang, Long Tian, Bo Chen, and Mingyuan Zhou. 2020. Friendly topic assistant for transformer based abstractive summarization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 485–497.

Jianlong Wu, Keyu Long, Fei Wang, Chen Qian, Cheng Li, Zhouchen Lin, and Hongbin Zha. 2019. Deep comprehensive correlation mining for image clustering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8150–8159.

Liang Wu and Huan Liu. 2018. Tracing fake-news footprints: Characterizing social media messages by how they propagate. In *Proceedings of the eleventh ACM international conference on Web Search and Data Mining*, pages 637–645.

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

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

Linxiao Yang, Ngai-Man Cheung, Jiaying Li, and Jun Fang. 2019. Deep clustering by gaussian mixture variational autoencoders with graph embedding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6440–6449.

Xu Yang, Cheng Deng, Kun Wei, Junchi Yan, and Wei Liu. 2020. Adversarial learning for robust deep clustering. *Advances in Neural Information Processing Systems*, 33.

Jesin Zakaria, Abdullah Mueen, and Eamonn Keogh. 2012. Clustering time series using unsupervised-shapelets. In *2012 IEEE 12th International Conference on Data Mining*, pages 785–794. IEEE.

Jichuan Zeng, Jing Li, Yan Song, Cuiyun Gao, Michael R Lyu, and Irwin King. 2018. Topic memory networks for short text classification. *arXiv preprint arXiv:1809.03664*.

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

Hao Zhang, Bo Chen, Dandan Guo, and Mingyuan Zhou. 2018. Whai: Weibull hybrid autoencoding inference for deep topic modeling. *arXiv preprint arXiv:1803.01328*.

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

Jun Zhao, Tao Gui, Qi Zhang, and Yaqian Zhou. 2021. A relation-oriented clustering method for open relation extraction. *arXiv preprint arXiv:2109.07205*.