Unsupervised Fine-tuning for Text Clustering

Shaohan Huang, Furu Wei, Lei Cui, Xingxing Zhang, Ming Zhou
Microsoft Research, Beijing, China
{shaohanh, fuwei, lecu, xizhang, mingzhou}@microsoft.com

Abstract

Fine-tuning with pre-trained language models (e.g. BERT) has achieved great success in many language understanding tasks in supervised settings (e.g. text classification). However, relatively little work has been focused on applying pre-trained models in unsupervised settings, such as text clustering. In this paper, we propose a novel method to fine-tune pre-trained models unsupervisedly for text clustering, which simultaneously learns text representations and cluster assignments using a clustering oriented loss. Experiments on three text clustering datasets (namely TREC-6, Yelp, and DBpedia) show that our model outperforms the baseline methods and achieves state-of-the-art results.

1 Introduction

Pre-trained language models have shown remarkable progress in many natural language understanding tasks (Radford et al., 2018; Peters et al., 2018; Howard and Ruder, 2018). Especially, BERT (Devlin et al., 2018) applies the fine-tuning approach to achieve ground-breaking performance in a set of NLP tasks. BERT, a deep bidirectional transformer model (Vaswani et al., 2017), utilizes a huge unlabeled data to learn complex features and representations and then fine-tunes its pre-trained model on the downstream tasks with labeled data.

Although BERT has achieved great success in many natural language understanding tasks under supervised fine-tuning approaches, relatively little work has been focused on applying pre-trained models in unsupervised settings. In this paper, by a case study of text clustering, we investigate how to leverage the pre-trained BERT model and fine-tune it in unsupervised settings, such as text clustering.

Previous approaches have made some progress on text clustering using deep neural networks (Min et al., 2018; Aljalbout et al., 2018). Existing deep clustering approaches fall into two categories: *two-stage* and *jointly optimization*. Two-stage approach uses deep learning frameworks to learn the representation first and then run clustering algorithms (Chen, 2015; Yang et al., 2017). As the name implies, jointly optimization approaches learn the representations and clustering jointly (Xie et al., 2016; Guo et al., 2017). Inspired by those methods, we can fine-tune pre-trained models by learning text representations and cluster assignments simultaneously.

In this paper, we propose a novel method to fine-tune pre-trained models unsupervisedly for text clustering. Our model simultaneously learns text representations and cluster assignments by jointly optimizing both the masked language model loss and the clustering oriented loss. The masked language model loss can help learn domain-specific knowledge and guarantee that our model not be misguided such as all-zero vector (Yang et al., 2017). Clustering oriented loss is designed to make the latent representation space more separable. In our experiments, we evaluate our proposed method on three different types of text datasets (namely, TREC-6,Yelp, and DBpedia). Experimental results show that our model achieves the state-of-the-art performance on question, sentiment and topic text datasets.

2 Model

Consider a text dataset X with n samples where $\{x_i \in X\}_{i=1}^{n}$. The number of clusters K is known and lets $\{\mu_i\}_{i=1}^{k}$ denote cluster centers. We aim to learn a good encoder $f_\theta : x_i \rightarrow z_i$, which makes

Proceedings of the 28th International Conference on Computational Linguistics, pages 5530–5534
Barcelona, Spain (Online), December 8-13, 2020

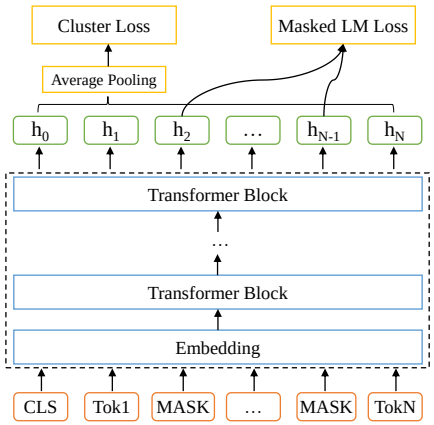


Figure 1: The overview architecture of unsupervised fine-tuning pre-trained model for text clustering.

the representation z_i of ith sample x_i more suitable for the clustering tasks. The θ and cluster centers μ_i are learnable parameters. As illustrated in Figure 1, we implement masked language model loss L_m and clustering loss L_c in our model. The masked language model loss helps to learn representations in a domain-specific dataset. The clustering loss is responsible for making representations more discriminative and separable. The loss function can be formulated as follows:

$$L = L_m + L_c \tag{1}$$

We first introduce the BERT model and masked language model loss L_m. We then describe the clustering loss L_c with KL-divergence. Finally, we present the parameter training details.

2.1 Pre-trained Model

BERT (Devlin et al., 2018), a pre-trained model, has achieved great success on many natural language processing tasks. The architecture of BERT model is a multi-layer bidirectional Transformer encoder, which takes words and their positions as input through embedding layer and transformer blocks and outputs the final hidden representations.

As illustrated in Figure 1, we fine-tune our model as a masked-language model as in (Devlin et al., 2018), which masks some of the input tokens randomly, and then predicts those masked tokens. The final hidden representations corresponding to the mask tokens are fed into a softmax layer over the vocabulary. The masked language model loss L_m is optimized by minimizing the negative log-likelihood.

In a vanilla BERT model, the hidden representations of [CLS] token is used as a symbol to represent one sentence or a pair of sentences. In the unsupervised setting, we do not have labeled data to fine-tune our model and the hidden vector of [CLS] token may not capture all information without fine-tuning. In order to obtain better representation, we implement an average pooling to compute the text representation as $z_i = \sum_j^N h_{i,j}/N$, where $h_{i,j}$ is the hidden vector of jth token in sample x_i. In the next section, we introduce how to compute the clustering loss with the representation z_i.

2.2 Clustering Loss

The clustering loss with the representation z_i is designed to learn representation distribution with the help of an auxiliary target distribution (Xie et al., 2016). The clustering loss is defined as Kullback-Leibler (KL) divergence between distribution *P* and *Q*, where *Q* is the distribution of soft assignment by Student's *t*-distribution (Maaten and Hinton, 2008) and *P* is the target distribution derived from *Q*. The clustering loss is defined as:

$$L_c = KL(P||Q) = \sum_i \sum_j p_{ij} log \frac{p_{ij}}{q_{ij}} \tag{2}$$

where q_{ij} is the similarity between text representation z_i and clustering centroid μ_j:

$$q_{ij} = \frac{(1 + ||z_i - \mu_j||^2/\alpha)^{-\frac{\alpha+1}{2}}}{\sum_{j'}(1 + ||z_i - \mu_{j'}||^2/\alpha)^{-\frac{\alpha+1}{2}}} \tag{3}$$

Since we cannot cross-validate α on a validation set in the unsupervised setting, we let $\alpha = 1$ for all experiments as in (Xie et al., 2016; Guo et al., 2017). We use the distribution of soft assignment q_{ij} to assign the label l_i to x_i as follows:

$$l_i = arg \max_j q_{ij} \tag{4}$$

The target distribution p_{ij} puts more emphasis on data points assigned with high confidence and normalizes loss contribution (Xie et al., 2016). It is computed as follows:

$$p_{ij} = \frac{q_{ij}^2 / f_j}{\sum_{j'} q_{ij'}^2 / f_{j'}} \tag{5}$$

where $f_j = \sum_i q_{ij}$ are soft cluster frequencies. For clustering part, the target distribution *P* is derived from *Q*, so minimizing clustering loss is a self-training process (Nigam and Ghani, 2000).

To initialize the cluster centroids, we first extract representations z_i through the original pre-trained model. Then employ standard k-means clustering in the representations space $\{z_i\}_{i=1}^n$ to obtain k initial centroids $\{\mu_j\}_{j=1}^k$.

To avoid instability, we update the target distribution *P*, which depends on the predicted soft labels, per epoch rather than per batch. To make the target distribution *P* towards "groundtruth" distribution, we update *P* without masking any token.

3 Experiments

3.1 Dataset and Evaluation Metrics

We conducted experiments on three types of text datasets. **TREC** dataset (Voorhees and Tice, 1999) is an open-domain, fact-based questions dataset, which contains six categories and 5,452 examples. **DBpedia** ontology datasets are constructed by picking 14 non-overlapping classes from DBpedia 2014 (Zhang et al., 2015). **Yelp** reviews dataset is constructed to predict number of stars the user has given, which has 5 classes (Zhang et al., 2015). Since some algorithms do not scale to the full DBpedia and Yelp datasets (Xie et al., 2016; Guo et al., 2017), we randomly choose 10,000 samples for clustering.

To evaluate whether the clustering results, we measure the clustering purity, which is a well-known metric for evaluating clustering (Manning et al., 2008). To compute purity, each cluster is assigned to the class which is most frequent in the cluster, and then the accuracy of this assignment is measured by counting the number of correctly assigned instances and dividing by the number of instances.

3.2 Implementation

We implement our model based on the bert-base-uncased version of BERT. We set the learning rate as $3e^{-5}$. During training, we replace 10% of tokens with mask token at random. We set the max sequence length is 128, the batch size is 16, and maximum epoch as 10.

3.3 Baseline Methods

We compare our method against traditional clustering algorithms k-means and three deep clustering algorithms. **DEC** represents Deep Embedded Clustering (Xie et al., 2016) that pre-trains autoencoder to learn feature representation and uses cluster assignment hardening loss as a regularization. **IDEC** is an improved Deep Embedded Clustering (Guo et al., 2017) method adding reconstruction term to preserve local structure. **DCN** represents Deep Clustering Network model, which is a "two-stage" model proposed by Yang et al. (2017).

We also evaluate two two-stage clustering algorithms: **AE+k-means** represents performing k-means algorithm on features of the pre-trained autoencoder and **BERT+k-means** is applying k-means algorithm on average hidden vectors of original BERT.

Dataset	k-means	DEC	IDEC	DCN	AE+k-means	BERT+k-means	Our method
TREC-6	20.87%	23.52%	26.33%	24.63%	21.96%	38.51%	**39.91%**
DBpedia	15.01%	21.77%	20.43%	18.51%	17.15%	63.25%	**66.99%**
Yelp	20.64%	23.12%	23.51%	22.34%	21.53%	30.02%	**33.40%**

Table 1: Clustering purity on three datasets.

Dataset	Full	w/o masked LM loss	w/o average pooling
TREC-6	39.91%	39.10%	39.79%
DBpedia	66.99%	64.25%	65.18%
Yelp	33.40%	32.08%	33.04%

Table 2: Model ablation tests.

3.4 Results

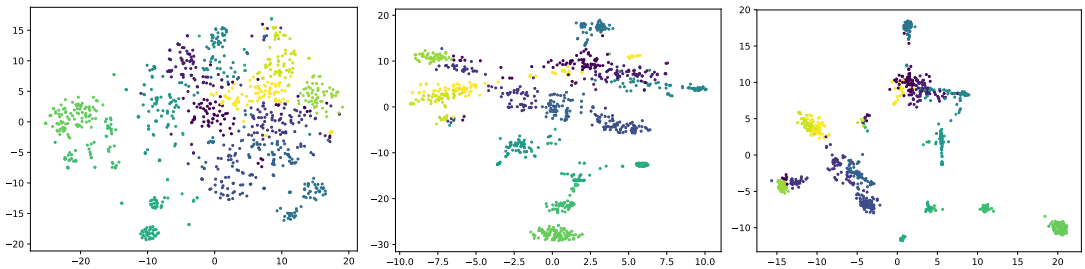
We report the clustering purity results on three text datasets in Table 1. As it shows, our model outperforms the baseline methods and achieves state-of-the-art results. Comparing BERT+k-means method with AE+k-means method, there is a large gap between them, which indicates that we can learn a better text representation from pre-trained model than autoencoder framework for text clustering tasks. The improvement of our method over BERT+k-means method reflects that our unsupervised fine-tuning method can help improve clustering performance.

We train variants of our model by removing the masked LM loss and using the hidden representations of [CLS] to represent sentences. The results are shown in Table 2. We can find that there was a marked decrease when the masked LM loss is removed, which indicates that the masked LM loss is crucial for text clustering. The results also shows that there has only been a marginal improvement using average pooling instead of the hidden representations of [CLS].

Figure 2 describes the visualization of the representations during training on DBpedia dataset. We use t-SNE (Maaten and Hinton, 2008) to visualize the latent representation on a random subset (1000 examples) of DBpedia dataset, where different colors stand for different clusters. It's the visualization result of BERT+k-means method when the epoch number equals 1. As shown in Figure 2, it is clear that the clusters are becoming increasingly better separated from epoch 1 to epoch 5. Meanwhile, the results demonstrate that our model doesn't require a large number of epochs to converge.

4 Conclusion

We proposed a method to fine-tune the pre-trained language models in unsupervised settings for text clustering. It provides a method to leverage pre-trained model for text clustering. Experimental results show that our model achieves the state-of-the-art performance on TREC-6, Yelp, and DBpedia datasets.



(a) Epoch 1 (b) Epoch 3 (c) Epoch 5

Figure 2: Visualization of representations on DBpedia dataset during training. Different colors mark different clusters.

References

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*.

Gang Chen. 2015. Deep learning with nonparametric clustering. *arXiv preprint arXiv:1501.03084*.

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*.

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

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. *arXiv preprint arXiv:1801.06146*.

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605.

Christopher D Manning, Hinrich Schütze, and Prabhakar Raghavan. 2008. Introduction to information retrieval. *Cambridge, UP*.

Erxue Min, Xifeng Guo, Qiang Liu, Gen Zhang, Jianjing Cui, and Jun Long. 2018. A survey of clustering with deep learning: From the perspective of network architecture. *IEEE Access*, 6:39501–39514.

Kamal Nigam and Rayid Ghani. 2000. Analyzing the effectiveness and applicability of co-training. In *Cikm*, volume 5, page 3.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*.

Alec Radford, Karthik Narasimhan, Time Salimans, and Ilya Sutskever. 2018. Improving language understanding with unsupervised learning. Technical report, Technical report, OpenAI.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Ellen M Voorhees and Dawn M Tice. 1999. The trec-8 question answering track evaluation. In *TREC*, volume 1999, page 82. Citeseer.

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

Bo Yang, Xiao Fu, Nicholas D Sidiropoulos, and Mingyi Hong. 2017. Towards k-means-friendly spaces: Simultaneous deep learning and clustering. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 3861–3870. JMLR. org.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in neural information processing systems*, pages 649–657.