# A Self-Training Approach for Short Text Clustering

**Amir Hadifar** **Lucas Sterckx** **Thomas Demeester** **Chris Develder**
Ghent University – imec, IDLab
Department of Information Technology
firstname.lastname@ugent.be

## Abstract

Short text clustering is a challenging problem when adopting traditional bag-of-words or TF-IDF representations, since these lead to sparse vector representations for short texts. Low-dimensional 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 we 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. Experiments on three short text datasets empirically validate the effectiveness of our method.

## 1 Introduction

Text clustering groups semantically similar text without using supervision or manually assigned labels. Text clusters have proven to be beneficial in many applications including news recommendation (Wang et al., 2010), language modeling (Liu and Croft, 2004), query expansion (Amini and Usunier, 2007), visualization (Cadez et al., 2003), and corpus summarization (Schutze and Silverstein, 1997).

Due to the popularity of social media and online fora such as Twitter and Reddit, texts containing only few words have become prevalent on the web. Compared to clustering of long documents, Short Text Clustering (STC) introduces additional challenges. Traditionally, text is represented as a bag-of-words (BOW) or term-frequency inverse-document-frequency (TF-IDF) vectors, after which a clustering algorithm such as $k$-means is applied to partition the texts into homogeneous groups (Xu et al., 2017). Due to the short lengths of such texts, their vector representations tend to become very sparse. As a result, traditional measures for similarity, which rely on word overlap or distance between high-dimensional vectors, become ineffective (Xu et al., 2015).

Previous work on STC enriched short text representations by incorporating features from external resources. Hu et al. (2009) and Banerjee et al. (2007) extended short texts using articles from Wikipedia. In similar fashion, Hotho et al. (2003) and Wei et al. (2015) proposed different methods to enrich text representation using ontologies. More recently, low-dimensional representations have shown potential to counter the sparsity problem in STC. Combined with neural network architectures, embeddings of words (Mikolov et al., 2013; Pennington et al., 2014), sentences (Le and Mikolov, 2014; Kiros et al., 2015) and documents (Dai et al., 2015) were proven to be effective on a variety of tasks in machine learning for NLP.

*Deep clustering* methods first embed the high-dimensional data into a lower dimensional space, after which a clustering algorithm is applied. These methods either perform clustering after having trained the embedding transformation (Tian et al., 2014; De Boom et al., 2016), or jointly optimize both the embedding and clustering (Yang et al., 2016), and we situate our method in the former. Closely related to our work is the method of Deep Embedded Clustering (DEC) (Xie et al., 2016), which learns feature representations and cluster assignments using deep neural networks. DEC learns a mapping from the data space to a lower-dimensional feature space while iteratively optimizing a clustering objective. The self-taught convolutional neural network ($STC^2$) framework proposed by Xu et al. (2017) uses a dimensionality reduction technique to generate auxiliary targets for a neural network architecture. A convolutional neural network (CNN) learns feature rep-

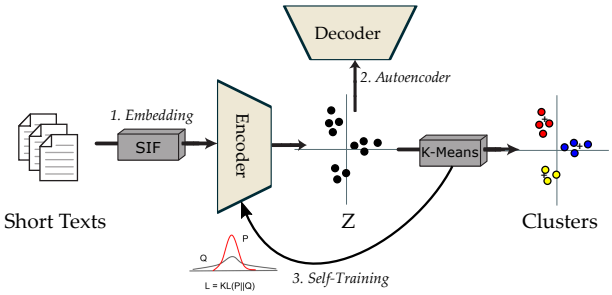


Figure 1: Short text clustering using SIF embedding, an autoencoder architecture and self-training.

resentations in order to reconstruct these auxiliary targets. Trained representations from the CNN are clustered using the $k$-means algorithm. Two recent surveys provide an overview of research on deep clustering methods (Aljalbout et al., 2018; Min et al., 2018).

Similar to Xie et al. (2016), we follow a multi-phase approach and train a neural network (which we will refer to as the encoder) to transform embeddings to a latent space before clustering. However, we apply two crucial modifications. As opposed to CNN-based encoders (Xu et al., 2017), we propose the use of Smooth Inverse Frequency (SIF) embeddings (Arora et al., 2017) in order to simplify and make clustering more efficient while maintaining performance.

During the second stage of clustering, we apply self-training using soft cluster assigments to fine-tune the encoder before applying a final clustering. We describe our methodology in more detail in Section 2. In Section 3, we evaluate our method using three short text datasets, measuring for clustering accuracy and normalized mutual information. Our model matches or produces better results compared to more sophisticated neural network architectures.

## 2 Methodology

Our model for short text clustering includes three steps: (1) Short texts are embedded using SIF embeddings (Section 2.1); (2) During a pre-training phase, a deep *autoencoder* is applied to encode and reconstruct the short text SIF embeddings (Section 2.2); (3) In a self-training phase, we use soft cluster assignments as an auxiliary target distribution, and jointly fine-tune the encoder weights and the clustering assignments (Section 2.3). The described setup is illustrated in Figure 1.

### 2.1 SIF Embedding

We apply a relatively simple and yet effective strategy for embedding short texts, called Smooth Inverse Frequency (SIF) embeddings. For SIF embedding, first, a weighted average of pre-trained word embeddings is computed. The contribution of each word is calculated as $\frac{a}{a+p(w)}$ with $a$ being a hyperparameter and $p(w)$ being the empirical word frequency in the text corpus. SIF embeddings are then produced by computing the first principal component of all the resulting vectors and removing it from the weighted embeddings.

### 2.2 Autoencoder

The parameters of the encoder network are initialized using a deep autoencoder architecture such as the one used by Hinton and Salakhutdinov (2006). The mean squared error is used to measure reconstruction loss after the encoded embeddings are decoded by the decoder subnetwork (see Fig. 1). This *non-clustering* loss is independent of the clustering algorithm and controls preservation of the original text representations. Yang et al. (2017) demonstrated that the absence of such a non-clustering loss can lead to worse representations, or trivial solutions where the clusters all collapse into a single representation.

### 2.3 Self-Training

After pre-training using the autoencoder architecture, we obtain an initial estimate of the non-linear mapping from the SIF embedding to a low-dimensional representation, on which a cluster algorithm is applied. Next, we improve clustering using a second *self-training* phase: we assign initial cluster centroids after which we alternate between two steps: (i) first, the probability of assigning a data point to each cluster is computed; (ii) second, an auxiliary probability distribution is calculated and used as target for the encoder network. Network weights and cluster centroids are updated iteratively until a stopping criterion is met.

For Step (i), we compute a *soft cluster assignment* for each data point. Maaten and Hinton (2008) propose the Student's $t$-distribution $Q$ with a single degree of freedom to measure the similarity between embedded points $z_i$ and centroids $\mu_j$:

$$q_{ij} = \frac{(1 + \|z_i - \mu_j\|^2)^{-1}}{\sum_{j'}(1 + \|z_i - \mu_{j'}\|^2)^{-1}}, \qquad (1)$$

in which $q_{ij}$ can be interpreted as the probability of assigning sample $i$ to cluster $j$. Then $q_{ij}$ can be used as a soft assignment of embeddings to centroids. The encoder is then fine-tuned to match this soft assignment $q_i$ to a target distribution $p_j$.

For Step (ii), as Xie et al. (2016), we use an auxiliary target distribution $P$ which has "stricter" probabilities compared to the similarity score $q_{ij}$, with the aim to improve cluster purity and put more emphasis on data points assigned with high confidence. This prevents large clusters from distorting the hidden feature space. The probabilities $p_{ij}$ in the proposed distribution $P$ are calculated as:

$$p_{ij} = \frac{q_{ij}^2 / \sum_{i'} q_{i'j}}{\sum_{j'} (q_{ij'}^2 / \sum_{i'} q_{i'j'})}, \quad (2)$$

in which the squared summation terms $q_{ij}^2$ are normalized by the soft cluster frequencies ($\sum_{i'} q_{i'j}$).

The KL-divergence between the two probability distributions $P$ and $Q$ is then used as training objective, i.e., the training loss $L$ is defined as:

$$L = \mathrm{KL}(P \| Q) = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}}. \quad (3)$$

The strategy outlined above can be seen as a form of self-supervision (Nigam and Ghani, 2000). Centroids of a standard clustering algorithm (e.g., $k$-means) are used to intialize the weights of the clustering layer, after which high confidence predictions are used to fine-tune the encoder and centroids. After convergence of this procedure, short texts are encoded and final cluster assignments are made using $k$-means.

## 3 Experimental Results

After describing the datasets (Section 3.1) and the experiment design (Section 3.2), we will present the results of these experiments (Section 3.3).

### 3.1 Data

We replicate the test setting used by Xu et al. (2017) and evaluate our model on three datasets for short text clustering: (1) **SearchSnippets:** a text collection comprising Web search snippets categorized in 8 different topics (Phan et al., 2008). (2) **Stackoverflow:** a collection of posts from question and answer site stackoverflow, published as part of a Kaggle challenge.[1] This subset contains question titles from 20 different categories selected by Xu et al. (2017). (3) **Biomedical,** a snapshot of one year of PubMed data distributed by BioASQ for evaluation of large-scale online biomedical semantic indexing.[2] Table 2 provides an overview of the main characteristics of the presented short text datasets.

### 3.2 Experimental Setup

We compare our method to baselines for STC including clustering of TF and TF-IDF representations, Skip-thought Vectors (Kiros et al., 2015) and the best reported STC$^2$ model by Xu et al. (2017). Following (Van Der Maaten, 2009; Xie et al., 2016), we set sizes of hidden layers to $d$:500:500:2000:20 for all datasets, where $d$ is the short text embedding dimension for all datasets. We used pre-trained word2vec embeddings[3] with fixed $\alpha = 0.1$ value for all corpora. We set the batch size to 64 and pre-trained the autoencoder for 15 epochs. We initialized stochastic gradient descent with a learning rate of 0.01 and momentum value of 0.9.

During experiments, the choice of initial centroids had considerable impact on clustering performance when applying the $k$-means algorithm. To reduce this influence of initialization, we restarted $k$-means 100 times with different initial centroids, as Huang et al. (2014); Xu et al. (2017), and selected the best centroids, which obtained the lowest sum of squared distances of samples to their closest cluster center. Similar to Xu et al. (2017), results are averaged over 5 trials and we also report the standard deviation on the scores.

### 3.3 Results and Discussion

We evaluate clustering performance based on the correspondence between clusters and partitions as per the ground truth class labels assigned to each of the short texts. We report two widely used performance metrics, the clustering accuracy (ACC) and the normalized mutual information (NMI) (Huang et al., 2014; Xu et al., 2017).

NMI measures the information shared between the predicted assignments $A$, and the ground truth

[1] https://www.kaggle.com/c/predict-closed-questions-on-stack-overflow/

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

[3] Available from https://github.com/jacoxu/STC2

| Method | SearchSnippets | | Stackoverflow | | Biomedical | |
|---|---|---|---|---|---|---|
| | ACC | NMI | ACC | NMI | ACC | NMI |
| TF | 24.7±2.22 | 9.0±2.30 | 13.5±2.18 | 7.8±2.56 | 15.2±1.78 | 9.4±2.04 |
| TF-IDF | 33.8±3.92 | 21.4±4.35 | 20.3±3.95 | 15.6±4.68 | 28.0±2.83 | 25.4±3.23 |
| Skip-Thought | 33.6±1.95 | 13.8±0.78 | 9.3±0.24 | 2.7±0.34 | 16.3±0.33 | 10.7±0.46 |
| SIF | 53.4±1.86 | 36.9±0.90 | 30.5±0.28 | 28.9±0.17 | 33.7±2.35 | 30.1±0.64 |
| STC$^2$ | 77.0±4.1 | **62.9±1.7** | 51.14±2.9 | 49.0±1.5 | 43.0±1.3 | 38.1±0.5 |
| **SIF + Aut., Self-Train.** | **77.1±1.1** | 56.7±1.0 | **59.8±1.9** | **54.8±1.0** | **54.8±2.3** | **47.1±0.8** |

Table 1: Clustering results (accuracy ACC and normalized mutal information NMI) for three short text collections using various representations and self-training methods. STC$^2$ and **our method** involve additional fine-tuning of encoders, others apply $k$-means directly on short text representations. Performance results are average and standard deviations over 5 runs.

| Dataset | $C$ | $N$ | $T$ | $\|V\|$ |
|---|---|---|---|---|
| SearchSnippets | 8 | 12.3k | 17.9 | 31k |
| StackOverflow | 20 | 20k | 8.3 | 23k |
| Biomedical | 20 | 20k | 12.9 | 19k |

Table 2: Statistics for the short text clustering datasets as used by Xu et al. (2017): number of classes ($C$), number of short texts ($N$), average number of tokens per text ($T$) and vocabulary size ($|V|$).

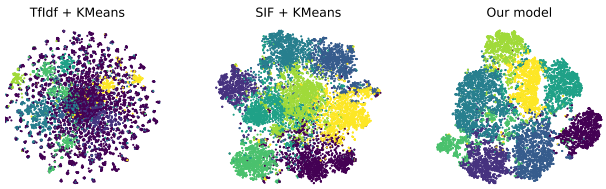


Figure 2: Two dimensional representations of *SearchSnippets* short texts before application of $k$-means. Colors indicate the $C = 8$ different ground truth labels.

assignments $B$, and is defined as

$$NMI(A, B) = \frac{I(A, B)}{\sqrt{H(A)H(B)}}, \quad (4)$$

where $I$ is the mutual information and $H$ is the entropy. When data is partitioned perfectly, the NMI score is 1, and when A and B are independent, it becomes 0.

The clustering accuracy is defined as

$$ACC = \frac{\sum_{i=1}^{N} \delta(y_i = map(c_i))}{N}, \quad (5)$$

where $\delta()$ is an indicator function, $c_i$ is the clustering label for $x_i$, $map()$ transforms the clustering label $c_i$ to its group label by the Hungarian algorithm (Papadimitriou and Steiglitz, 1982), and $y_i$ is the true group label of $x_i$. Results for NMI and accuracy of existing work and the presented model are shown in Table 1.

While generic, low-dimensional representations such as Skip-Thought or SIF embeddings have demonstrated to be beneficial for NLP on many tasks, for STC, additional fine-tuning and self-training leads to improved cluster quality. The evaluation results show the superiority of our approach, compared to the STC$^2$ model, on all but one of the metrics.

Qualitatively, the improved cluster quality is also visually apparent in Figure 2, which shows a two-dimensional t-SNE (Maaten and Hinton, 2008) representation of the *SearchSnippets* short texts before clustering.

The source code of our model, implemented using Tensorflow, is publicly available to encourage further research on STC.[4]

## 4 Conclusion

We proposed a method for clustering of short texts using sentence embeddings and a multi-phase approach, starting from unsupervised SIF embeddings for the short texts. Our STC model then adopts an autoencoder architecture which is fine-tuned for clustering using self-training. Our empirical evaluation on three short text clustering datasets demonstrates resulting accuracies ranging from at least as good up to 12 percentage points, compared to the state-of-the-art STC$^2$ method.

[4] https://github.com/hadifar/stc_clustering

## Acknowledgments

The authors would like to thank the anonymous reviewers for their constructive feedback.

## 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*.

Massih R Amini and Nicolas Usunier. 2007. A contextual query expansion approach by term clustering for robust text summarization. In *Proceedings of DUC*, pages 48–55.

Sanjeev Arora, Yingyu Liang, and Tengyu Ma. 2017. A simple but tough-to-beat baseline for sentence embeddings. In *Proceedings of ICLR*.

Somnath Banerjee, Krishnan Ramanathan, and Ajay Gupta. 2007. Clustering short texts using Wikipedia. In *Proceedings of SIGIR*, pages 787–788. ACM.

Igor Cadez, David Heckerman, Christopher Meek, Padhraic Smyth, and Steven White. 2003. Model-based clustering and visualization of navigation patterns on a web site. *Data Mining and Knowledge Discovery*, 7:399–424.

Andrew M. Dai, Christopher Olah, and Quoc V. Le. 2015. Document embedding with paragraph vectors. *CoRR*, abs/1507.07998.

Cedric De Boom, Steven Van Canneyt, Thomas Demeester, and Bart Dhoedt. 2016. Representation learning for very short texts using weighted word embedding aggregation. *Pattern Recogn. Lett.*, pages 150–156.

Geoffrey E. Hinton and Ruslan R. Salakhutdinov. 2006. Reducing the dimensionality of data with neural networks. *Science*, pages 504–507.

Andreas Hotho, Steffen Staab, and Gerd Stumme. 2003. Ontologies improve text document clustering. In *Proceedings of ICDM*, pages 541–544. IEEE.

Xiaohua Hu, Xiaodan Zhang, Caimei Lu, Eun K. Park, and Xiaohua Zhou. 2009. Exploiting Wikipedia as external knowledge for document clustering. In *Proceedings of SIGKDD*, pages 389–396. ACM.

Peihao Huang, Yan Huang, Wei Wang, and Liang Wang. 2014. Deep embedding network for clustering. In *Proceedings of ICPR*, pages 1532–1537. IEEE.

Ryan Kiros, Yukun Zhu, Ruslan R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Skip-thought vectors. In *Proceedings of the NIPS*, pages 3294–3302.

Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *Proceedings of the ICML*, pages 1188–1196.

Xiaoyong Liu and W. Bruce Croft. 2004. Cluster-based retrieval using language models. In *Proceedings of SIGIR*, pages 186–193. ACM.

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

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Proceedings of NIPS*, pages 3111–3119.

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 *Proceedings of CIKM*, pages 86–93.

Christos H. Papadimitriou and Kenneth Steiglitz. 1982. *Combinatorial Optimization: Algorithms and Complexity*. Prentice-Hall.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of EMNLP*, pages 1532–1543.

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 WWW*, pages 91–100. ACM.

Hinrich Schutze and Craig Silverstein. 1997. Projections for efficient document clustering. In *Proceedings of SIGIR*, pages 74–81.

Fei Tian, Bin Gao, Qing Cui, Enhong Chen, and Tie-Yan Liu. 2014. Learning deep representations for graph clustering. In *Proceedings of AAAI*, page 1293–1299.

Laurens Van Der Maaten. 2009. Learning a parametric embedding by preserving local structure. In *Proceedings of AIStats*, pages 384–391.

Jia Wang, Qing Li, Yuanzhu Peter Chen, and Zhangxi Lin. 2010. Recommendation in internet forums and blogs. In *Proceedings of ACL*, pages 257–265. ACL.

Tingting Wei, Yonghe Lu, Huiyou Chang, Qiang Zhou, and Xianyu Bao. 2015. A semantic approach for text clustering using WordNet and lexical chains. *Expert Systems with Applications*, 42:2264–2275.

Junyuan Xie, Ross Girshick, and Ali Farhadi. 2016. Unsupervised deep embedding for clustering analysis. In *Proceedings of ICML*, pages 478–487.

Jiaming Xu, Peng Wang, Guanhua Tian, Bo Xu, Jun Zhao, Fangyuan Wang, and Hongwei Hao. 2015. Short text clustering via convolutional neural networks. In *Workshops at the ACL Conference*, pages 62–69. ACL.

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.

Bo Yang, Xiao Fu, Nicholas D Sidiropoulos, and Mingyi Hong. 2017. Towards k-means-friendly spaces: Simultaneous deep learning and clustering. In *Proceedings of ICML*, pages 3861–3870.

Jianwei Yang, Devi Parikh, and Dhruv Batra. 2016. Joint unsupervised learning of deep representations and image clusters. In *Proceedings of CVPR*, pages 5147–5156.