CLUSTERBERT: MULTI-STAGE FINE-TUNING OF TRANSFORMERS FOR DEEP TEXT CLUSTERING

Anonymous authors
Paper under double-blind review

ABSTRACT

Transformer models have originally been designed for text generation, classification, and sequence labelling, and they have achieved new state-of-the-art results in those areas. Recent deep clustering methods learn cluster-friendly spaces for complex data and thereby outperform traditional clustering algorithms, especially on images and graphs. We propose ClusterBERT, an unsupervised algorithm that combines the strengths of both approaches. By tightly integrating transformer-based sentence representation learning with clustering, our method discovers a cluster-friendly representation of text data that retains useful semantic information. ClusterBERT is a multi-stage procedure that consists of domain adaptation, clustering, and hardening of the clusters. Starting from an initial representation obtained by transformer models, ClusterBERT learns a cluster-friendly space for text data by jointly optimizing the reconstruction loss and a clustering loss. Our experiments demonstrate that ClusterBERT outperforms state-of-the-art text clustering methods.

1 INTRODUCTION

Clustering is one of the key strategies to analyze, categorize and make sense of unstructured data in an unsupervised fashion. To achieve good clustering results, it is vital to provide the algorithms with a useful representation of the data. This is especially challenging for abstract data types like texts. Traditional approaches utilize text statistics to vectorize texts (e.g. Bag of Words (BoW) and weighting schemes such as term-frequency inverse-document-frequency (TF-IDF)). As these methods solely rely on word frequencies, they cannot capture structural compositionality or lexical variability and therefore cannot capture certain types of semantic similarity between instances. Neural network-based approaches (Mikolov et al., 2013; Le & Mikolov, 2014) use context, i.e., the surrounding words or documents, to learn a semantically dense representations, or embeddings. Transformer-based language models, e.g., BERT (Devlin et al., 2019), capture word interactions and vastly outperform other model types on many natural language processing tasks, such as natural language inferenceg (NLI), machine translation or question answering.

The breadth of tasks solvable by these large language models hints at the fact that a lot of useful information is stored within their representations. However, these models are not optimized to perform text clustering tasks. To improve upon this issue, a more cluster-friendly sentence representation is crucial.

To this end, we make use of the transformer fine-tuning paradigm and integrate it with a multi-stage deep clustering setting. We propose ClusterBERT, an unsupervised framework that aims to enforce a cluster separation in the embedded space into semantic categories while keeping the semantic information contained in the sentence embedding. Figure 1 depicts how our approach optimizes the sentence representation to achieve a good cluster separation. ClusterBERT extends a transformer-based encoder-decoder architecture with a deep clustering objective. First, in the *domain adaptation* stage, the base transformer model is pre-trained with an autoencoder (AE) objective function on the dataset to be clustered in order to adapt the model to the target domain. Second, in the *clustering* stage, we cluster the latent embedding using a traditional clustering method and combine the AE reconstruction loss with a classification loss.

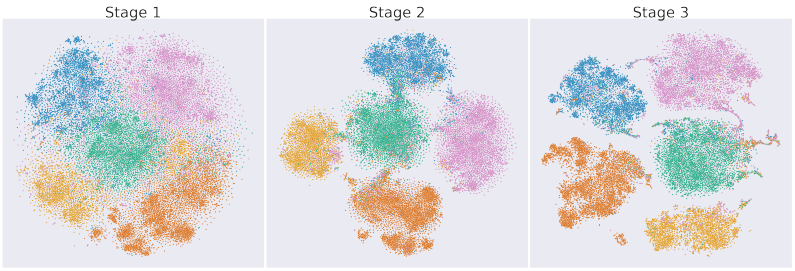

Figure 1: t-SNE embeddings of the sentence representations of the BBC-News dataset, colored by ground truth labels. Representations generated by ClusterBERT at the end of the domain adaptation stage (Stage 1), clustering (Stage 2) and hardening stage (Stage 3).

Last, in the *hardening* stage, we further improve the cluster purity by optimizing for an objective that sharpens the representations w.r.t. their cluster assignments. Throughout all stages, the same deep architecture is used. Only the training objective (the weighting of loss terms) is different in each step.

ClusterBERT outperforms traditional and state-of-the-art methods on clustering tasks. By additionally evaluating an unsupervised retrieval task, our ablations suggest a good trade-off between clustering efficacy and semantic information present in the learned representation. The contribution in this work can be summarized as follows:

1. We propose a multi-stage method that combines fine-tuning (an idea crucial for transformers in natural language processing (NLP)) with the reconstruction and clustering paradigm typical in deep clustering,
2. we evaluate several sentence embedding and clustering combinations in order to showcase how our algorithm produces improved cluster separation over several state-of-the-art methods,
3. we provide an analysis of model performance throughout the training stages, as well as ablations and qualitative analysis and
4. we make the code, data and resulting models publicly available. [1]

2 Related Work

The analysis of related work is separated into two parts. First, an overview of deep clustering methods is provided. Secondly, we give some background on text clustering and sentence embeddings. As the text clustering literature is heavily dependent on the progress in the text representation models, we will review the most important milestones in this section as well.

2.1 Deep Clustering

The success of deep neural networks has enabled the development of deep clustering. The general idea is to enforce an improved cluster separation on the latent layer to enhance the performance of clustering algorithms. To solve this objective, Xie et al. (2016) propose DEC, which imposes a student's t-distribution with k-means centroids upon the output layer of a neural network architecture.

Guo et al. (2017) extend this idea by using an AE, such that a reconstruction loss can be utilized to prevent degenerate solutions (IDEC). Adapting this framework, Dizaji et al. (2017) propose DEPICT, an algorithm that simultaneously denoises and clusters the input data. DEPICT minimizes the reconstruction between distorted and original samples along with a clustering loss. Jabi et al. (2019) derive a theoretical relationship between k-means clustering and the transformation performed by deep clustering using discriminative objectives. They show that predictions based on the softmax

[1] https://www.dropbox.com/sh/fxpjxzu18jindt2/AABCKSDashG0zXq6R08u7vA4a?dl=0

activation function are equivalent to assigning transformed data points to the closest centroid. Thus, forcing the linear layer with softmax activation to output a k-means-like label distribution improves the cluster separation in the hidden layer. Aside from deep clustering based on k-means centroids, VaDE (Jiang et al., 2017) combines a Gaussian mixture model prior with a variational autoencoder (VAE) (Kingma & Welling, 2014) to learn a deep generative clustering and DeepECT (Mautz et al., 2019) introduces a deep embedded cluster tree.

Most deep clustering algorithms, however, are tailored to the computer vision domain. There are a few approaches using sentence embeddings as input for deep clustering algorithms (Hadifar et al., 2019; Yin et al., 2021), i.e., an additional AE architecture is stacked on top of the encoder language model. For an overview of other deep clustering approaches see Aljalbout et al. (2018).

2.2 Text Clustering

Early approaches (Zhang et al., 2011; Zhao & Mao, 2017) used BoW and TF-IDF as representations of texts due to their good performance and efficiency and combine these with a clustering algorithm, such as k-means. However, as those methods solely utilize the frequency of words in a corpus, they are not capable of detecting certain types of semantic similarity between instances.

In the last decade, the NLP community switched their focus to dense neural network-based representations to find better semantic representations of texts. After Mikolov et al. (2013) introduced the Word2Vec algorithm, a plethora of neural word and document embedding models were proposed (Wang et al., 2020; Arora et al., 2017). Based on that, Zhang et al. (2021c) use an attention module to learn a relation between word embeddings and cluster representations. Various publications showcase how clustering algorithms combined with those models outperform prior statistical approaches (Xu et al., 2015; Hadifar et al., 2019).

Arguably, the biggest milestone in NLP over the last years is the introduction of transformer models (Vaswani et al., 2017), notably the encoder transformer architecture BERT (Devlin et al., 2019). Subakti et al. (2022) combine extracted BERT embeddings with various clustering algorithms, showcasing how these models outperform the previous state of the art over several datasets. Ait-Saada et al. (2021) propose a clustering ensemble approach which shows that it is beneficial to use all BERT-layers for clustering. Sentence-BERT (SBERT) (Reimers & Gurevych, 2019a) uses a siamese network architecture to fine-tune BERT with supervised datasets, e.g., the human-labelled NLI dataset, to learn sentence embeddings. These datasets explicitly reflect similarity and dissimilarity. Limited labelled datasets also motivate recent work (Gao et al., 2021; Yan et al., 2021) to build unsupervised learning frameworks. Wang et al. (2021) propose the transformer-based autoencoder (TSDAE) which is explained in detail in Section 3. Yin et al. (2021); Pugachev & Burtsev (2021) demonstrate that clustering algorithms based on transformer representations have the potential to outperform traditional text clustering methods.

Finally, there is work to use contrastive learning for the purpose of text clustering. Zhang et al. (2021a) propose SCCL to further improve upon those models by jointly optimizing a contrastive loss and a clustering loss. Similarly, VaSCL (Zhang et al., 2021b) also uses a constrastive loss but introduces a virtual augmentation method which constructs the top-K nearest neighbors of each training instance to generate data augmentations.

In contrast, we do not use contrastive learning but an AE to retain semantic information which is common and effective in image clustering (Guo et al., 2017; Dizaji et al., 2017). Furthermore, we implement a clustering loss that does not require cluster representatives, i.e. cluster-centers.

3 Methodology

The objective is to combine the clustering performance of deep clustering methods with the representational power of the transformer encoder. To this end, we implemented an encoder-decoder framework that simultaneously improves the semantic information in the sentence representations while enforcing a cluster-friendly embedding. A decoder transformer learns to reconstruct texts using the sentence embedding of the encoder. In order to optimize the sentence embedding for clustering, we employ a classification head on top of the embedding layer which is interpretable as a soft cluster prediction. We also refer to this layer as the clustering head.

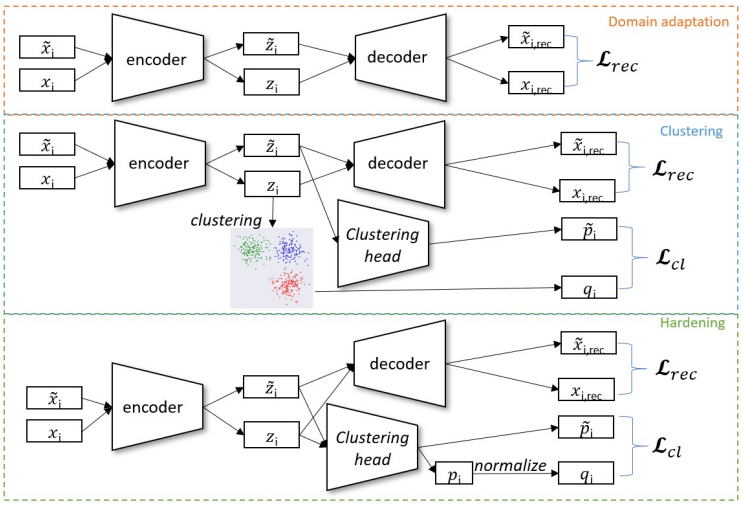

Figure 2: Training stages of ClusterBERT. The domain adaptation stage only optimizes reconstruction. The clustering stage optimizes the prediction with k-means labels as targets (jointly with the reconstruction loss). The hardening stage calibrates the confidence of cluster predictions.

3.1 Overview

The outlined idea is split into three dedicated stages which are shown in Figure 2. The corresponding pseudo-code is in Algorithm 1. First, a *domain adaptation* stage fine-tunes the used AE on the specific dataset. The second stage performs *clustering* on the sentence embeddings (with the added clustering head). The last stage, *hardening*, adapts the target of the clustering objective and aims to learn more robust representations with respect to cluster assignment.

Before diving into details, we introduce some notations. Whenever a sentence embedding is mentioned, it is obtained from the classifier (*CLS*) token embedding (Reimers & Gurevych, 2019a) of the transformer encoder. It contains information about the entire input sequence and is typically used for downstream tasks. Let $\mathbf{x}$ be an input sentence drawn from the text corpus $\mathbf{X}$. We want to optimize the sentence embedding $\mathbf{z} = \text{enc}(\mathbf{x}) \in \mathcal{R}^d$ such that (1) a decoder $\mathbf{x}_{rec} = \text{dec}(\mathbf{z})$ is able to reconstruct the entire sentence using solely the sentence embedding and (2) instances of the same cluster are separated from instances of different clusters. i.e., the inter-cluster distance is maximized while the intra-cluster distance is minimized. A linear layer with a softmax activation (clustering head/cl-head) produces predictions $\mathbf{P} = \text{cl-head}(\text{enc}(\mathbf{X})) \in [0,1]^{m \times k}$. The clustering head is optimized to learn a target cluster distribution denoted by $\mathbf{Q} \in [0,1]^{m \times k}$. Here m denotes the number of samples and k the number of clusters.

3.2 Domain Adaptation Stage

In the first stage, the model is trained on the dataset in order to prevent out-of-domain problems in the clustering phase. For this, we implemented the framework proposed by Wang et al. (2021), which is a novel autoencoder based on pre-trained transformer models to make the semantic information contained in the sentence embedding accessible.

The main idea is to learn sentence embeddings using the objective of reconstructing distorted input into the original input. An input sample from the corpus $\mathbf{x} \in \mathbf{X}$ is transformed into a corresponding distorted sample $\tilde{\mathbf{x}}$ by adding noise, e.g., word deletion. In the following, we use tildes (e.g., $\tilde{\mathbf{x}}_i, \tilde{\mathbf{p}}_i$) whenever we refer to the models output produced with the perturbed input. During training, $\tilde{\mathbf{x}}$ is fed as input to the AE which generates the reconstruction $\tilde{\mathbf{x}}_{rec} = \text{dec}(\text{enc}(\tilde{\mathbf{x}}))$. The training objective minimizes the difference between probability distributions of $\mathbf{X}$ and $\tilde{\mathbf{X}}_{rec}$ using the standard cross entropy (CE) loss.

This can be summarized as the following objective function:

$$\mathcal{L}_{rec} = \frac{1}{m} \sum_{i=1}^{m} CE(\mathbf{x}_i, \tilde{\mathbf{x}}_{i,rec})) \tag{1}$$

where $CE(\mathbf{x}_i, \tilde{\mathbf{x}}_{i,rec}))$ is the (categorical) cross-entropy between the one-hot-encoded unperturbed sentence $\mathbf{x}_i$ and the probability distribution over reconstructions $\tilde{\mathbf{x}}_{i,rec}$. BERT-based models are used as the backbone for both encoder and decoder. In comparison to other transformer-based encoder-decoder models (e.g., Vaswani et al. (2017)), that take outputs of all word tokens from the encoder into account, we only use the *CLS* token embedding $\mathbf{z}$ of the encoder as the key and value for the attention mechanism in the decoder. Since all the information goes through $\mathbf{z}$, it is usable as the embedding for the input sentence.

3.3 Clustering Stage

The second stage introduces a cluster structure to the representations. First, a standard clustering algorithm, e.g., k-means, is applied to the latent space representations $\{\mathbf{z}_i\}_i$ to generate an initial target clustering $\mathbf{Q}$. We interpret these cluster predictions as pseudo-labels, that are now utilized to update our model. Next, the predictions p_{ij} are produced by the clustering head, which is a linear layer h followed by a softmax activation, i.e.,

$$p_{ij} = P(\mathbf{x}_i \in C_j | \mathbf{z}_i) = \frac{exp(h(\mathbf{z}_i)_j)}{\sum_k exp(h(\mathbf{z}_i)_k)}, \tag{2}$$

where C_j is the j-th cluster.

The clustering loss aims to minimize the cross entropy between the cluster predictions $\tilde{\mathbf{P}}$ and cluster targets $\mathbf{Q}$ by employing a standard clustering head:

$$\mathcal{L}_{cl} = CE\left(\tilde{\mathbf{P}}, \mathbf{Q}\right) = -\frac{1}{N} \sum_i \sum_j q_{ij} \log \tilde{p}_{ij} \tag{3}$$

The cluster prediction of the distorted input sample $\tilde{\mathbf{p}}_i$ are used in the clustering loss function. The perturbations can be viewed as adversarial noise that is inserted in order to improve the robustness of the learned predictions (see also Dizaji et al. (2017)). The cluster loss is interpolated with the reconstruction loss and both are jointly optimized (controlled by a hyperparameter $\lambda \in [0, 1]$):

$$\mathcal{L}_{full} = \lambda \cdot \mathcal{L}_{cl} + (1 - \lambda) \cdot \mathcal{L}_{rec}. \tag{4}$$

In summary, Equation 4 optimizes for a cluster-friendly representation (Equation 3) while simultaneously keeping a semantically meaningful sentence embedding space (Equation 1). Using the cluster predictions as labels for optimization runs the risk of a bad initial clustering to influence the results. To alleviate this, we run this stage multiple times, i.e., re-cluster to update $\mathbf{Q}$.

Algorithm 1 ClusterBERT

Require: BERT-based encoder and decoder, loss parameter λ
  # Domain Adaptation Stage
  for $i = 1, \ldots,$ num_epochs_1 **do**
    Train autoencoder by minimizing $\mathcal{L}_{\text{rec}}$ (Section 3.2)
  end for
  # Clustering Stage
  for $i = 1, \ldots,$ num_epochs_2 **do**
    Create cluster labeling $\mathbf{Q}$ using k-means
    Train $\mathcal{L}_{\text{full}}$ (Equation 4) using $\mathbf{Q}$ and λ
  end for
  # Hardening Stage
  for $i = 1, \ldots,$ num_epochs_3 **do**
    Update targets $\mathbf{Q}$ using Equation 5
    Train $\mathcal{L}_{\text{full}}$ (Equation 4) using $\mathbf{Q}$ and λ
  end for

Dataset	# of classes	# of instances	avg. length
StackOverflow	20	15.9k	50 CHAR
Biomedical	20	41.7k	88 CHAR
SearchSnippets	20	25.7k	87 CHAR
GoogleNews	152	32.5k	38 CHAR
BBC	5	22k	122 CHAR
AgNews	4	36k	194 CHAR

Table 1: Summary of data statistics.

3.4 HARDENING STAGE

The goal of the third and final stage of the training is to improve the cluster purity, i.e., to decrease the uncertainty of the cluster predictions. Thus, instead of re-clustering the hidden layer, the predictions of the clustering head are utilized to update the targets **Q**.

We compute the targets **Q** as follows (see also Xie et al. (2016))

$$q_{ij} = \frac{p_{ij}^2 / f_j}{\sum_k (p_{ik}^2 / f_k)}, \tag{5}$$

where $f_j = \sum_i p_{ij}$ denotes the cluster frequency of cluster C_j. This updating step aims to push the algorithm toward a balanced label distribution and to harden the predictions **P**, i.e., it ensures that the points are assigned with high confidence. In other words, this increased confidence enforces a stricter cluster separation in the embedding space. The predictions **P**, produced by the unperturbed sentences, are used in order to update the targets **Q** (Equation 5). Using these targets, the predictions $\tilde{\mathbf{P}}$ from the perturbed sentences are optimized as in the previous stage following Equation 3.

4 EXPERIMENTS

The experiments are separated into multiple parts. First, we have a general evaluation relating ClusterBERT to the baselines. Next, an in-depth cluster analysis is given with a special focus on the evolution over the stages. Then, ablation analyses are performed in order to understand the different elements of the model. To further demonstrate how our algorithm updates the sentence embeddings, a qualitative analysis looks at interesting samples and their embedding behavior from the initial to the final stage. This qualitative analysis can be found in the Appendix.

4.1 EXPERIMENTAL SETUP

First, we want to describe which datasets we evaluate our model and which baselines we choose for comparison. We use six commonly used text categorization datasets (used by, e.g., Shi & Wang (2021); Hadifar et al. (2019); Belford & Greene (2020). Namely, we evaluated the StackOverflow, Biomedical, SearchSnippets, AgNews, BBC, and GoogleNews datasets. For all datasets the ground truth labels are available. Table 1 depicts the data statistics.

We use three types of baselines. Where-ever not stated explicitly, k-means is applied to the obtained embedding. First, the statistical word-count based method **TF-IDF**, term frequency-inverse document frequency, is used to transform the text into vectors. This traditional baseline is still competitive. Secondly, we compare to word embedding-based methods. To generate **Smooth Inverse Frequency (SIF) embeddings** (Arora et al., 2017) a weighted average of pre-trained word embeddings is computed. We include a model based on SIF embeddings with k-means and with IDEC (Hadifar et al., 2019). The final category are transformer-based baselines. Without fine-tuning, we use the classifier token embedding of **RoBERTa** (Liu et al., 2019) and **SentenceBERT** (SBERT) (Reimers & Gurevych, 2019b) which was trained on a natural language inference dataset. The next three baselines are fine-tuned on the specific dataset. The autoencoder **TSDAE** (Wang et al., 2021) is described in section 3.2. **SCCL** (Zhang et al., 2021a) combines contrastive learning with a clustering head. Lastly, **VaSCL** (Zhang et al., 2021b) combines contrastive learning with a virtual neighborhood augmentation.

	StackOverflow		Biomedical		SearchSnippets		BBC-News		GoogleNews		AGNews	
Model	NMI	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI	ACC
TF-IDF	67.8	65.4	30.0	27.2	44.9	51.8	4.2	27.2	45.3	60.5	2.0	29.5
SIF	22.0	24.4	20.0	24.0	38.0	37.7	7.8	32.8	66.1	39.8	19.7	45.6
SIF+idec	31.9	29.7	20.7	22.5	35.8	35.0	13.1	34.0	50.7	24.7	41.5	72.6
RoBerta	9.5	15.3	16.0	18.9	23.4	27.7	29.6	49.5	52.8	30.2	19.7	47.7
SBERT	12.8	19.7	5.0	34.7	16.8	22.2	5.0	34.7	37.7	23.1	8.7	41.3
TSDAE	49.2	56.3	27.9	29.4	39.9	42.7	44.8	66.7	64.4	43.8	62.9	86.2
SCCL	69.3	70.0	33.7	30.3	56.3	51.1	37.8	59.4	72.2	58.1	59.9	83.1
VaSCL	59.6	68.7	30.4	30.1	46.2	45.5	26.1	52.3	64.8	39.7	23.7	49.3
ClusterBERT	**75.6**	**79.9**	**43.4**	**38.2**	**65.5**	**57.6**	**67.9**	**87.8**	**76.9**	**63.2**	**67.2**	**87.1**

Table 2: Clustering NMI and ACC reported for six text datasets (using the k-means algorithm on the sentence embeddings). The best-achieved result is highlighted in bold and the second-best is underlined. Both Metrics multiplied by 100

Following previous work, we evaluate the clustering performance utilizing the two well-known metrics normalized mutual information (NMI) and cluster accuracy (ACC). More information about the implementation of the baselines, metrics, datasets and training details for our algorithm can be found in the Appendix.

4.2 Main Results

The outcome of the main experiments is summarized in Table 2. Here, we present the results achieved with the k-means algorithm on the generated embeddings (except for SIF+IDEC, which is the baseline proposed by Hadifar et al. (2019)), without any dimensionality reduction.

For an extended table (including various dimensionality reduction/clustering algorithm combinations) see Appendix. The results suggest that the clustering results based on the embeddings created using ClusterBERT outperform other baselines. The fact that the Roberta-, SBERT- and TSDAE-based clusterings are outperformed by most other baselines, even by the traditional TF-IDF method, confirms our initial intuition that transformer-based models need to be fine-tuned for the clustering task.

The main objective of our framework is to fine-tune a transformer model to improve the cluster separation in the embedding. This is the case for all the datasets we evaluated. We can see, that the base model (RoBerta) did not provide satisfying results. Comparing this model with the final model, one can observe that the NMI increases up to 65 points. Averaged over all datasets, our model improves the NMI by 39.2 points compared to the base model and by 16.3 points compared to the TSDAE model.
Looking at the extended results in Table 3, we observe that an additional dimensionality reduction, and/or IDEC instead of k-means, improves clustering results for most methods. However, the results of ClusterBERT are rather stable for all methods. We argue, that this stems from an improved cluster separation of the sentence embeddings.

We realized that the numbers reported in the text clustering literature often vary a lot. As our setup and our reproduced results also vary from the original publications, Table 4 provided in the Appendix contains reported numbers and a comparison to ClusterBERT's results. It is not guaranteed, that every publication utilizes the same datasets; therefore, different experimental results are to be expected.

4.3 Cluster Analysis

Furthermore, an analysis of the cluster separation throughout the training process is given. Therefore the two established cluster validation metrics Silhouette Index (Rousseeuw, 1987) S and Dunn Index (Dunn, 1973) D are computed. Intuitively, the Silhouette Index computes how similar an object is to its own cluster compared to other clusters. Therefore it uses the mean intra-cluster distance (distance between instances of the same cluster) and the mean nearest-cluster distance of each sample. A Silhouette Index close to 1 indicates that the resulting clusters are compact.

The Dunn index describes the minimum closest distance between any two clusters (inter-cluster distance) divided by the maximum distance between the two farthest points in the cluster. This index indicates the separation of the resulting clusters.

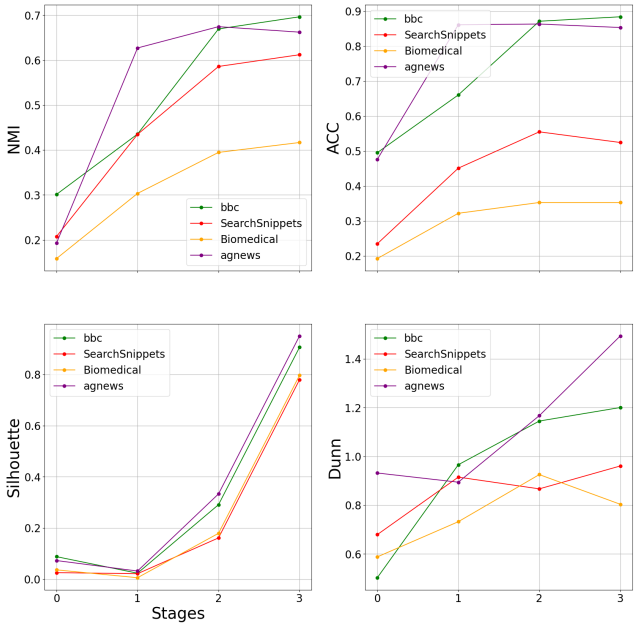

Figure 3: NMI and ACC, Silhouette Index, Dunn Index on four text datasets plotted against the Stages. Stage 0 uses embeddings by a pre-trained RoBERTa model, followed by the stages of ClusterBERT.

Again, higher values correspond to compact and well-separated clusters. A detailed explanation is presented in Bolshakova & Azuaje (2003).

The analysis is shown in Figure 3. Interestingly, we observe that the domain adaptation stage generates slightly worse clusters with respect to the Silhouette Index, which we think is due to the fact that no explicit clustering optimization is performed.

As expected, Stage 1 and 2 improve the NMI and ACC by a wide margin. In the hardening stage, those two metrics are not improved anymore, however, the silhouette index sharply increases. This indicates, that the cluster separation further improves in this last stage. Similarly, the hardening stage appears to increase the Dunn index (however not for the Biomedical dataset). Overall, the cluster validity indices are aligned with the visualization shown in Figure 1.

4.4 Reconstruction Ablation

The model is based on a trade-off between clustering and reconstruction. Thus, an ablation is provided which leaves out the reconstruction term. The goal is to understand 1) how does reconstruction effect clustering performance and 2) how much semantic information is kept.

The results are shown in Figure 4.4. All lines represent the average across all six datasets with respect to the corresponding metrics. The left side analysis the cluster performance 1) using NMI and ACC. Wang et al. (2021) argue that standard sentence similarity tasks are not enough as real-world scenarios such as web search usually only aim to retrieve a few related items. To answer 2) we also use the information retrieval task askUbuntu (Lei et al., 2016). For a given input, a post, the models are supposed to rank candidate questions according to their similarity.

This task is evaluated using the information retrieval metrics mean reciprocal rank (MRR) and mean average precision (MAP). Intuitively, they indicate how well the highest ranked item matches and how well all correct items are ranked, respectively. On the clustering performance 1), full ClusterBERT outperforms the ablated model which only uses a clustering loss by about 2 points. To answer 2) we observe that the MAP and MRR values decrease up to 5 points more than the full model. This leads us to the conclusion that the reconstruction is needed to uphold the performance of ClusterBERT,

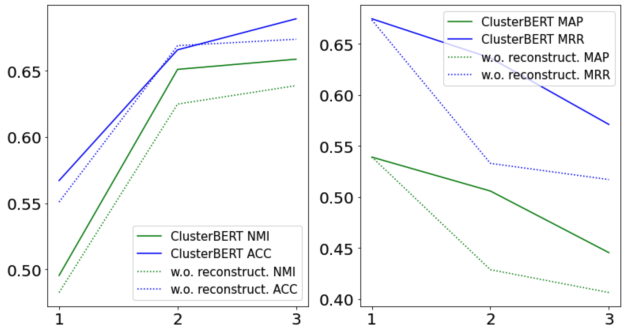

Figure 4: Ablation comparing ClusterBERT with (solid lines) and without (dotted lines) reconstruction loss. The left side shows clustering metrics, and the right side information retrieval metrics throughout the stages. All lines represent the metrics averaged over all datasets.

5 Conclusions

Fine-tuning is crucial for transformer-based language models to achieve top performance (as they have on many tasks). In clustering, there is no obvious target to fine-tune to, and transformers previously had difficulty performing on par with traditional, often much simpler methods.

In this work, we present ClusterBERT, a method for fine-tuning transformer-based language models specifically for clustering in three stages: Domain adaptation, clustering with cluster prediction, and hardening. ClusterBERT fine-tunes the transformer towards two objectives, reconstruction from perturbed inputs, and prediction of auxiliary cluster labels, that are jointly optimized, and slightly differently instantiated in each stage.

ClusterBERT outperforms previously proposed traditional and recent methods and establishes a new state-of-the-art for six standard text clustering datasets. We showcase, that, without our fine-tuning, transformer-based language models are not able to produce good clustering results. Our analysis shows the importance of each stage with respect to clustering performance and cluster separation. An ablation analysis on an additional information retrieval task shows that the reconstruction loss helps to retain semantic information in the fine-tuned representations.

REFERENCES

Mira Ait-Saada, François Role, and Mohamed Nadif. How to leverage a multi-layered transformer language model for text clustering: An ensemble approach. CIKM '21. Association for Computing Machinery, 2021.

Elie Aljalbout, Vladimir Golkov, Yawar Siddiqui, and Daniel Cremers. Clustering with deep learning: Taxonomy and new methods. *CoRR*, abs/1801.07648, 2018.

Sanjeev Arora, Yingyu Liang, and Tengyu Ma. A Simple but Tough-to-Beat Baseline for Sentence Embeddings. 2017.

Mark Belford and Derek Greene. Ensemble topic modeling using weighted term co-associations. *Expert Systems with Applications*, 161:113709, 2020.

Nadia Bolshakova and F. Azuaje. Cluster validation techniques for genome expression data. *Signal Processing*, 83, 2003.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics, 2019.

Kamran Dizaji, Amirhossein Herandi, Cheng Deng, Weidong Cai, and Heng Huang. Deep clustering via joint convolutional autoencoder embedding and relative entropy minimization. 2017.

J. C. Dunn. A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters. *Journal of Cybernetics*, 3(3):32–57, 1973.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. SimCSE: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2021.

Derek Greene and Pádraig Cunningham. Practical solutions to the problem of diagonal dominance in kernel document clustering. In *Proc. 23rd International Conference on Machine learning (ICML'06)*, pp. 377–384. ACM Press, 2006.

Xifeng Guo, Long Gao, Xinwang Liu, and Jianping Yin. Improved Deep Embedded Clustering with Local Structure Preservation. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*. International Joint Conferences on Artificial Intelligence Organization, 2017.

Amir Hadifar, Lucas Sterckx, Thomas Demeester, and Chris Develder. A Self-Training Approach for Short Text Clustering. Association for Computational Linguistics, 2019.

Mohammed Jabi, Marco Pedersoli, Amar Mitiche, and Ismail Ben Ayed. Deep clustering: On the link between discriminative models and k-means. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

Zhuxi Jiang, Yin Zheng, Huachun Tan, Bangsheng Tang, and Hanning Zhou. Variational deep embedding: An unsupervised and generative approach to clustering. In *IJCAI*, 2017.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR (Poster)*, 2015. URL `http://arxiv.org/abs/1412.6980`.

Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *ICLR*, 2014.

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

Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. *31st International Conference on Machine Learning, ICML 2014*, 4, 2014.

Tao Lei, Hrishikesh Joshi, Regina Barzilay, Tommi Jaakkola, Kateryna Tymoshenko, Alessandro Moschitti, and Lluís Màrquez. Semi-supervised question retrieval with gated convolutions. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 1279–1289, San Diego, California, June 2016. Association for Computational Linguistics. doi: 10.18653/v1/N16-1153. URL https://aclanthology.org/N16-1153.

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

S. Lloyd. Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 1982.

Dominik Mautz, Claudia Plant, and Christian Böhm. Deep embedded cluster tree. In *ICDM*, pp. 1258–1263. IEEE, 2019.

Leland McInnes, John Healy, and James Melville. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. Technical report, 2020.

Tomas Mikolov, Kai Chen, G.s Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *Proceedings of Workshop at ICLR*, 2013.

Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar, 2014. Association for Computational Linguistics.

Xuan Hieu Phan, Minh Le Nguyen, and Susumu Horiguchi. Learning to classify short and sparse text web with hidden topics from large-scale data collections. In *WWW*, pp. 91–100, 2008. URL https://doi.org/10.1145/1367497.1367510.

Leonid Pugachev and Mikhail Burtsev. Short Text Clustering with Transformers, 2021. arXiv:2102.00541 [cs].

Jipeng Qiang, Qian Zhenyu, Yun Li, Yunhao Yuan, and Xindong Wu. Short text topic modeling techniques, applications, and performance: A survey. *arXiv preprint arXiv:1904.07695*, 2019.

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

Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. 2019b. doi: 10.18653/v1/D19-1410.

Peter Rousseeuw. Rousseeuw, p.j.: Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. comput. appl. math. 20, 53-65. *Journal of Computational and Applied Mathematics*, 20:53–65, 11 1987.

Haoxiang Shi and Cen Wang. Self-supervised Document Clustering Based on BERT with Data Augment. *arXiv:2011.08523 [cs]*, 2021.

Alvin Subakti, Hendri Murfi, and Nora Hariadi. The performance of BERT as data representation of text clustering. *Journal of Big Data*, 2022.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is All you Need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

Kexin Wang, Nils Reimers, and Iryna Gurevych. Tsdae: Using transformer-based sequential denoising auto-encoder for unsupervised sentence embedding learning. In *EMNLP*, 2021.

Shirui Wang, Wenan Zhou, and Chao Jiang. A survey of word embeddings based on deep learning. *Computing*, 2020.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 38–45, Online, October 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-demos.6. URL https://aclanthology.org/2020.emnlp-demos.6.

Junyuan Xie, Ross Girshick, and Ali Farhadi. Unsupervised deep embedding for clustering analysis. In Maria Florina Balcan and Kilian Q. Weinberger (eds.), *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pp. 478–487, New York, New York, USA, 2016. PMLR.

Jiaming Xu, Peng Wang, Guanhua Tian, Bo Xu, Jun Zhao, Fangyuan Wang, and Hongwei Hao. Short text clustering via convolutional neural networks. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, pp. 62–69, Denver, Colorado, June 2015. Association for Computational Linguistics. doi: 10.3115/v1/W15-1509. URL https://aclanthology.org/W15-1509.

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

Yuanmeng Yan, Rumei Li, Sirui Wang, Fuzheng Zhang, Wei Wu, and Weiran Xu. Consert: A contrastive framework for self-supervised sentence representation transfer. *arXiv preprint arXiv:2105.11741*, 2021.

Hui Yin, Xiangyu Song, Shuiqiao Yang, Guangyan Huang, and Jianxin Li. Representation learning for short text clustering. pp. 321–335. Springer-Verlag, 2021. ISBN 978-3-030-91559-9.

Dejiao Zhang, Feng Nan, Xiaokai Wei, Shang-Wen Li, Henghui Zhu, Kathleen McKeown, Ramesh Nallapati, Andrew O. Arnold, and Bing Xiang. Supporting clustering with contrastive learning. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, 2021a.

Dejiao Zhang, Wei Xiao, Henghui Zhu, Xiaofei Ma, and Andrew O Arnold. Virtual augmentation supported contrastive learning of sentence representations. *arXiv preprint arXiv:2110.08552*, 2021b.

Wei Zhang, Chao Dong, Jianhua Yin, and Jianyong Wang. Attentive Representation Learning with Adversarial Training for Short Text Clustering. *IEEE Transactions on Knowledge and Data Engineering*, 2021c. Conference Name: IEEE Transactions on Knowledge and Data Engineering.

Wen Zhang, Taketoshi Yoshida, and Xijin Tang. A comparative study of tf* idf, lsi and multi-words for text classification. *Expert Systems with Applications*, 38(3):2758–2765, 2011.

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

Rui Zhao and Kezhi Mao. Fuzzy bag-of-words model for document representation. *IEEE transactions on fuzzy systems*, 26(2):794–804, 2017.

A APPENDIX

1.1 EXPERIMENTAL SETUP

1.2 BASELINES

TF-IDF The text samples are lower-cased, a selection of english stopwords is removed and subsequently transformed into vectors by using Tf-Idf with a maximum amount of 2000 features.
SIF (Arora et al., 2017) embeddings are weighted averages or pre-trained word vectors. Hadifar et al. (2019) proposed a model that combines SIF and IDEC (Guo et al., 2017). As the authors did not specify which word-embedding model they used, we used Glove Common Crawl (840B) (Pennington et al., 2014)[2] word vectors as described by Arora et al. (2017).

RoBERTa/ Sentence-BERT(Liu et al., 2019; Reimers & Gurevych, 2019b) In order to directly evaluate clustering on embeddings of BERT-based models, we use the classifier token embedding of a pretrained RoBERTa, namely *"roberta-base"* and the sentence transformer model *"sentence-transformers/nli-roberta-base"*. Both are hosted on the huggingface (Wolf et al., 2020) page.
TSDAE (Wang et al., 2021). Details about TSDAE are given in Section 3.2. As it is used as base model, we give more information in Training Details.
SCCL (Zhang et al., 2021a). As suggested by the authors, we use *distilbert-base-nli-stsb-mean-token* as a backbone, the learning rate is set to $1e-5$ and learning scale to 100 and 1000 iterations. Dropout values for the data augmentation are taken from the official repository [3].
VaSCL (Zhang et al., 2021b) Pre-trained models are available on huggingface, for our experiments *"vascl-roberta-large"* is used to generate the embeddings.

1.2.1 DATASETS

The first three datasets, where the preprocessed version by Xu et al. (2017)[4] is taken, exhibit rather similar characteristics. Additionally, we evaluate three news datasets exhibiting more diverse sentence lengths and number of classes.

StackOverflow is created as a subset of a Kaggle challenge, having $20k$ question titles associated with 20 categories from the StackOverflow website.

Biomedical is a dataset created from PubMed by randomly selecting $40k$ titles belonging to 20 different groups.

SearchSnippets was created from web search snippets (Phan et al., 2008) and categorized into 8 classes.

AgNews is a subset of AG's large corpus of news articles. In Zhang & LeCun (2015) the authors extracted the 4 largest classes to derive a classification dataset. It has been used for clustering before. Unfortunately, these versions include preprocessing steps such as stop word removal which are not ideal for BERT-based models. Thus, we create a new dataset and randomly sampled 8000 samples for each of the 4 classes.

BBC was created in Greene & Cunningham (2006) for the purpose of news topic classification. We split the articles into sentences and assigned the label of its corresponding document to each sentence.

GoogleNews was compiled by Qiang et al. (2019) and is publicly available[5]. The authors downloaded titles and snippets belonging to 152 clusters from the Google News site in 2013.

[2] publicly available at `https://nlp.stanford.edu/projects/glove/`
[3] `https://github.com/amazon-research/sccl`
[4] `https://github.com/jacoxu/STC2`
[5] `https://github.com/qiang2100/STTM`

1.2.2 METRICS AND EVALUATION

As stated in Section 4.1 we utilized the metrics NMI and ACC to evaluate our experiments. For ground-truth cluster assignments C and cluster predictions P, the NMI is defined by

$$\text{NMI}(C, P) = \frac{I(C, P)}{\sqrt{H(C)H(P)}} \tag{6}$$

where I denotes the mutual information and H the entropy. Additionally, we use the cluster accuracy, which is defined by

$$\text{ACC} = \frac{\sum_{i=1}^{N} \delta\left(c_i = \text{map}(p_i)\right)}{N} \tag{7}$$

where c_i is the ground truth label, p_i the cluster prediction of sample i, δ the Kronecker-delta function and map a function that uses the Hungarian algorithm (Kuhn, 1955) to find the best assignment between clusters and classes.

To evaluate the clustering capabilities of text representations, standard clustering algorithms are applied to the generated embeddings. The k-means algorithm (Lloyd, 1982) is used with k set to the number of ground truth labels (see Table 1). For IDEC (Guo et al., 2017) we use 150 epochs, the Adam optimizer (Kingma & Ba, 2015) with learning rate $1e-4$ and a hidden dimension of ground truth clusters " $+\,1$". When the UMAP (McInnes et al., 2020) dimension reduction technique is applied, the dimension is again set to the number of ground truth clusters " $+\,1$" and the number of neighbors to 15.

1.2.3 TRAINING DETAILS

For training ClusterBERT we used *"roberta-base"* as a backbone, fine-tuned with the Transformers and Sequential Denoising Autoencoder (TSDAE) objective on 10^6 randomly sampled sentences from English Wikipedia which were collected in Gao et al. (2021). The models are trained using grid search on the following hyperparameters. A batch size of 8 and the Adam optimizer (Kingma & Ba, 2015) with learning rate $2e-5$ is used. Stage 1 is trained for 2 epochs. In the clustering stage, the number of re-clusterings is in $\{1, 2\}$ with the number of epochs in $\{2, 4\}$ and $\lambda = 0.8$. In the hardening stage we run 2 epochs with a value of $\lambda = 0.75$. The best result on the train set is reported.

1.3 SUPPLEMENTARY RESULTS

1.3.1 FULL EVALUATION

In Table 3 we show all models with multiple dimension reduction and clustering variants.

1.3.2 COMPARISON TO REPORTED RESULTS

As we realized that the reproduced results of our experiments often vary compared to the original publications, they are summarized in Table 4. For example, the original SIF+IDEC (Hadifar et al., 2019) and our results vary by a large margin. There are many potential reasons for this gap. One might be that the authors utilized different word embeddings. Given that embeddings were often not available, we decided to use the same backbone as in the SIF paper (Arora et al., 2017). Apart from that, various text preprocessing steps or different subsets of the data might influence the results. A good indicator is the results using the simple model TF-IDF combined with k-means clustering, where our results (Table 2) are significantly higher (up to 52 higher NMI score) compared to other studies. However, our model is still able to be on par with the state-of-the-art. For the StackOverflow dataset we achieve the highest results for both metrics compared to the baselines and with the NMI score for Biomedical we rank second highest.

1.3.3 QUALITATIVE ANALYSIS

The goal of this analysis is to investigate qualitatively how the representation space changes over the training. Therefore, we look at sentence pairs from the BBC-News dataset and inspect changes

	StackOverflow		Biomedical		SearchSnippets		BBC-News		GoogleNews		AGNews	
Model	NMI	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI	ACC
TF-IDF+kmeans	67.8	65.4	30.0	27.2	44.9	51.8	4.2	27.2	45.3	60.5	2.0	29.5
SIF+kmeans	22.0	24.4	20.0	24.0	38.0	37.7	7.8	32.8	66.1	39.8	19.7	45.6
SIF+idec	31.9	29.7	20.7	22.5	35.8	35.0	13.1	34.0	50.7	24.7	41.5	72.6
SIF+umap+kmeans	49.2	52.0	31.0	30.9	47.9	44.6	28.6	54.5	73.6	53.8	55.6	81.0
SIF+umap+idec	44.7	43.1	22.7	22.3	41.1	37.6	26.9	53.5	68.0	43.5	54.0	81.2
RoBerta+kmeans	9.5	15.3	16.0	18.9	23.4	27.7	29.6	49.5	52.8	30.2	19.7	47.7
RoBerta+idec	2.1	8.1	8.6	11.9	14.8	15.4	52.7	76.9	17.6	4.5	54.3	80.7
RoBerta+umap+kmeans	22.7	26.6	24.1	25.2	30.2	35.6	33.9	48.4	62.1	40.2	47.0	64.5
RoBerta+umap+idec	21.6	27.6	17.6	18.5	25.3	23.0	33.6	49.7	44.0	19.5	47.7	64.6
SBERT+kmeans	12.8	19.7	5.0	34.7	16.8	22.2	5.0	34.7	37.7	23.1	8.7	41.3
SBERT+idec	17.8	22.9	2.6	31.1	25.1	31.4	2.6	31.1	25.4	12.3	16.2	48.5
SBERT+umap+kmeans	19.1	25.2	5.7	32.2	29.7	35.5	5.7	32.2	44.4	30.4	29.7	50.2
SBERT+umap+idec	19.5	25.2	6.0	31.7	19.9	23.5	6.0	31.7	37.3	20.5	30.4	62.2
TSDAE+kmeans	49.2	56.3	27.9	29.4	39.9	42.7	44.8	66.7	64.4	43.8	62.9	86.2
TSDAE+idec	32.4	31.9	18.3	17.8	40.6	36.7	44.1	71.3	29.2	8.4	57.5	80.4
TSDAE+umap+kmeans	52.4	60.3	37.6	36.3	50.4	51.3	50.6	60.2	67.9	52.8	66.2	87.0
TSDAE+umap+idec	45.2	47.9	29.2	26.4	33.4	27.8	55.4	80.0	56.4	30.4	65.7	**87.3**
SCCL+kmeans	69.3	70.0	33.7	30.3	56.3	51.1	37.8	59.4	72.2	58.1	59.9	83.1
SCCL+idec	66.0	67.1	33.3	33.5	56.0	47.1	39.3	67.1	75.6	62.0	53.9	80.2
SCCL+umap+kmeans	66.5	71.5	35.5	31.8	56.3	51.7	66.5	71.5	72.7	57.3	44.8	63.0
SCCL+umap+idec	63.3	69.0	35.0	31.7	57.3	53.7	43.1	62.3	67.1	46.5	39.1	54.3
VaSCL+kmeans	59.6	68.7	30.4	30.1	46.2	45.5	26.1	52.3	64.8	39.7	23.7	49.3
VASCL+idec	71.8	79.5	25.1	24.5	47.4	44.4	28.0	56.4	34.3	15.0	33.4	66.6
VASCL+umap+kmeans	72.2	**82.3**	38.9	34.2	58.6	55.2	**72.2**	82.3	76.0	60.0	55.8	79.3
VASCL+umap+idec	70.9	78.2	32.6	27.9	56.2	47.8	49.1	76.2	69.1	43.7	53.3	79.5
CBERTdec+kmeans	75.6	79.9	**43.4**	**38.2**	65.5	**57.6**	67.9	**87.8**	76.9	63.2	67.2	87.1
CBERTdec+idec	75.1	76.6	**43.4**	**38.2**	64.8	55.7	67.9	**87.8**	**78.9**	**68.9**	**67.3**	**87.3**
CBERTdec+umap+kmeans	**75.7**	80.0	**43.4**	**38.2**	**65.6**	**57.6**	68.0	**87.8**	75.4	48.5	67.2	87.1
CBERTdec+umap+idec	74.9	78.6	**43.4**	**38.2**	**65.6**	**57.6**	66.5	87.2	74.3	48.7	62.4	85.3

Table 3: Results Text Clustering, both Metrics multiplied by 100

from the initial to the final model with respect to their cosine similarity. More specifically, the base TSDAE model (see Section 4.1) and the trained ClusterBERT model are compared. We formalize five types of pairs based on their properties of ground-truth labels and similarity changes:

- $=$ pairs that have the same ground-truth label and are close in the beginning and in the end of the training.
- $\rightrightarrows\!\!\leftleftarrows$ pairs which have the same ground-truth label and the largest increase in similarity, i.e., pairs that should be moved together and are actually moved closer.
- $\leftrightarrows\!\!\rightleftarrows$ pairs which have different ground-truth labels and the largest increase in similarity, i.e., pairs that should be moved apart and are actually moved closer.
- $\leftleftarrows\!\!\rightrightarrows$ pairs which have different ground-truth labels and the largest decrease in similarity, i.e., pairs that should be moved apart and are actually moved further.
- $\rightleftarrows\!\!\leftrightarrows$ pairs which have the same ground-truth label and the largest decrease in similarity, i.e., pairs that should be moved together and are actually moved further.

For each type, a selected subsample of the top 10 largest changes in similarity of each category is chosen and shown in Table 5. Note that the models are called *init* and *final*.

	StackOverflow		Biomedical		SearchSnippets	
Model	NMI	ACC	NMI	ACC	NMI	ACC
TF-IDF[1]	15.6	20.3	25.4	27.5	21.4	33.8
SIF[1]	28.9	30.5	30.1	33.7	36.9	53.4
SIF+IDEC[1]	54.8	59.8	**47.1**	**54.8**	56.7	77.1
SCCL[2]	74.5	75.5	41.5	46.2	**71.1**	**85.2**
VaSCL [3]	-	76.2	-	42.6	-	50.1
ClusterBERT	**75.6**	**79.9**	43.4	38.2	65.5	57.6

reported in [1] Hadifar et al. (2019) [2] Zhang et al. (2021a) [3] Zhang et al. (2021b)

Table 4: Reported Results STC, both Metrics multiplied by 100

type	Label	Sentence pair	*init*	*final*
=	business	The UK economy could suffer a backlash from the slowdown in the housing market, triggering a fall in consumer spending and a rise in unemployment.	0.999	0.997
	business	The fall reflects weak exports and a slowdown in consumer spending, and follows similar falls in GDP in the two previous quarters.		
=	business	Another investor in Deutsche Boerse has supported the view that a payout to shareholders would be preferable to Deutsche Boerse overpaying for the LSE, Reuters news agency reported.	0.999	0.991
	business	The Deutsche Boerse was prepared to pay for the LSE "exceeds the potential benefits of this acquisition", said TCI.		
$\rightrightarrows\leftleftarrows$	business	It had been one of the quickest to deal with difficulties faced by the aviation industry after the 9/11 attacks in 2001.	0.333	0.998
	business	Continuing demand for Inbev's products in the South American markets where its Brazilian arm is most popular means it expects to keep boosting its turnover.		
$\leftrightarrows\rightleftarrows$	politics	But she said they were treated more sceptically than non-Roma passengers by immigration officers "acting on racial grounds".	0.351	0.997
	entertainment	The Big Issue magazine, which supports homelessness charities, prints the last known picture of Edwards in a fresh plea for information.		
$\leftleftarrows\rightrightarrows$	business	The parent or guardian will be responsible for the Bonds and will receive notification of the purchase.	0.499	0.392
	entertainment	They teamed up again for a concert to mark their induction into the UK Music Hall of Fame, and were joined by Taylor.		
$\leftleftarrows\rightrightarrows$	business	Mr Ebbers has pleaded not guilty to charges of fraud and conspiracy.	0.522	0.387
	entertainment	At the time of her death she was working on a film about the last two men pulled from the rubble of the Twin Towers following the 11 September terror attacks in 2001.		
$\rightleftarrows\leftrightarrows$	politics	Tony Blair is pressing the US to cut greenhouse gases despite its unwillingness to sign the Kyoto Protocol, Downing Street has indicated.	0.776	0.374
	politics	The prime minister is said to believe the United States' refusal to sign the Kyoto Protocol on emissions is undermining other countries' resolve to cut carbon dioxide production.		

Table 5: Cosine similarity between sentence pairs with the same and with different ground truth labels, for the initial TSDAE embeddings and the final ClusterBERT embeddings. Samples are taken from the BBC-News dataset

It is clearly observable from the results of the $\rightrightarrows\leftleftarrows$ pairs that the algorithm has a strong capability to move sentences toward each other. For example, the third $\rightrightarrows\leftleftarrows$ pair in Table 5 can not be easily found by human judgment.

Furthermore, the algorithm correctly pushes the last $\leftleftarrows\rightrightarrows$ pair in the Table apart, even though, when solely analyzing the words, one might assume that a sentence containing *guilty* and *conspiracy* might be similar to a sentence with *11 September* and *attack*.

The $\leftrightarrows\rightleftarrows$ pairs and $\rightleftarrows\leftrightarrows$ pairs are examples that our algorithm falsely pushed together and falsely pushed apart respectively. We observe how sentences from these types are sometimes ambiguous. The reason is that one sentence is often not enough for reliable categorization if it is taken from a newspaper article. Additionally, some of the categories appearing in the BBC dataset, e.g., *politics* and *business*, might have a large overlap of articles.