Adversarial Self-Supervised Data-Free Distillation for Text Classification

Xinyin Ma, Yongliang Shen, Gongfan Fang, Chen Chen, Chenghao Jia, Weiming Lu*
College of Computer Science, Zhejiang University
{maxinyin, syl, fgf, chen_double, chjia, luwm}@zju.edu.cn

Abstract

Large pre-trained transformer-based language models have achieved impressive results on a wide range of NLP tasks. In the past few years, Knowledge Distillation(KD) has become a popular paradigm to compress a computationally expensive model to a resource-efficient lightweight model. However, most KD algorithms, especially in NLP, rely on the accessibility of the original training dataset, which may be unavailable due to privacy issues. To tackle this problem, we propose a novel two-stage data-free distillation method, named ***Adversarial self-Supervised Data-Free Distillation*** (AS-DFD), which is designed for compressing large-scale transformer-based models (e.g., BERT). To avoid text generation in discrete space, we introduce a ***Plug & Play Embedding Guessing*** method to craft pseudo embeddings from the teacher's hidden knowledge. Meanwhile, with a ***self-supervised module*** to quantify the student's ability, we adapt the difficulty of pseudo embeddings in an adversarial training manner. To the best of our knowledge, our framework is the first data-free distillation framework designed for NLP tasks. We verify the effectiveness of our method on several text classification datasets.

1 Introduction

Recently, pre-trained language models (Devlin et al., 2018; Yang et al., 2019; Liu et al., 2019; Raffel et al., 2019) have achieved tremendous progress and reached the state-of-the-art performance in various downstream tasks such as text classification (Maas et al., 2011), language inference (Bowman et al., 2015) and question answering (Rajpurkar et al., 2016). These models become an indispensable part of current models for their transferability and generalizability.

However, such language models are huge in volume and demand highly in computational resources, making it impractical in deploying them on portable systems with limited resources (e.g., mobile phones, edge devices) without appropriate compression. Recent researches (McCarley, 2019; Gordon et al., 2020; Michel et al., 2019) focus on compressing the large-scale models to a shallow and resource-efficient network via weight pruning (Guo et al., 2019), knowledge distillation (Mukherjee and Awadallah, 2019), weight quantization (Zafrir et al., 2019) and parameter sharing (Lan et al., 2020). Among them, some methods (Sanh et al., 2019; Sun et al., 2019) draw on the idea of transfer learning, utilizing knowledge distillation (Hinton et al., 2015) to transfer latent representation information embedded in teachers to students. These knowledge distillation methods share some commonalities: they rely on the training data to achieve high accuracy. It will be intractable if we need to compress a model without publicly accessible data. Reasons for that include privacy protection, company assets, safety/security concerns and transmission. Representative samples include GPT2 (Radford et al., 2019), which has not released its training data with fears of abuse of language models. Google trains a neural machine translation system (Wu et al., 2016) using internal datasets owned and protected by the company. DeepFace (Taigman et al., 2014) is trained on user images under confidential policies for protecting users. Further, some datasets, like Common Crawl dataset used in GPT3 (Brown et al., 2020), contain nearly a trillion words and are difficult to transmit and store.

Conventional knowledge distillation methods are highly dependent on data. Some models or algorithms in Computer Vision like DAFL (Chen et al., 2019), ZSKD (Nayak et al., 2019) solve the data-free distillation by generating pseudo images or uti-

[*] Corresponding author

Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, pages 6182–6192,
November 16–20, 2020. ©2020 Association for Computational Linguistics

lizing metadata from teacher models. Exploratory researches (Micaelli and Storkey, 2019; Fang et al., 2019) also show that GANs can synthesize harder and more diversified images by exploiting disagreements between teachers and students. However, these models only make attempts in image tasks, designing for continuous and real-valued images. Applying these models to generate sentences is challenging due to the discrete representation of words (Huszár, 2015). Backpropagation on discrete words is not reasonable, and it seems unlikely to pass the gradient through the text to the generator. Apart from the discontinuity problem of text, some promotion strategies like layer-wise statistic matching in batch normalization (Yin et al., 2019) are not suitable for transformer-based models, which transposes batch normalization into layer normalization to fit with varied sentence length (Ba et al., 2016).

To address the above issues and distill without data, we propose a novel data-free distillation framework called "Adversarial self-Supervised Data-Free Distillation"(AS-DFD). We invert BERT to perform gradient updates on embeddings and consider parameters of the embedding layer as accessible knowledge for student models. Under constraints of constructing "BERT-like" vectors, pseudo embeddings extract underlying representations of each category. Besides, we employ a self-supervised module to quantify the student's ability and adversarially adjust the difficulty of pseudo samples, alleviating the insufficient supervisory problem controlled by the one-hot target. Our main contributions are summarized as follows:

- We introduce AS-DFD, a data-free distillation framework, to compress BERT. To the best of our knowledge, AS-DFD is the first model in NLP to distill knowledge without data.

- We propose a Plug & Play Embedding Guessing method and align the pseudo embeddings with the distribution of BERT's embedding. We also propose a novel adversarial self-supervised module to search for samples students perform poorly on, which also encourages diversity.

- We verify the effectiveness of AS-DFD on three popular text classification datasets with two different student architectures. Extensive experiments support the conjecture that synthetic embeddings are effective for data-free distillation.

2 Related Work

2.1 Data-Driven Distillation for BERT

Knowledge Distillation (KD) compresses a large model (the teacher model) to a shallow model (the student model) by imitating the teacher's class distribution output (Hinton et al., 2015). Bert (Devlin et al., 2018) contains multiple layers of transformer blocks (Vaswani et al., 2017) which encodes contextual relationship between words. Recently, many works successfully compress BERT to a BERT-like model with knowledge distillation (Sanh et al., 2019) and achieve comparable performances on downstream-tasks. Patient-KD (Sun et al., 2019) bridges the student and teacher model between its intermediate outputs. TinyBERT (Jiao et al., 2019) captures both domain-general and domain-specific knowledge in a two-stage framework. Zhao et al. (2019) employs a dual-training mechanism and shared projection matrices to compress the model by more than 60x. BERT-of-Theseus (Xu et al., 2020) progressively module replacing and involves a replacement scheduler in the distillation process. Besides, some recent surveys focus on compress BERT to a CNN-based (Chia et al., 2019) or LSTM-based model to create a more lightweight model with additional training data (Tang et al., 2019a,b).

2.2 Data-Free Distillation Methods

Current methods for data-free knowledge distillation are applied in the field of computer vision. Lopes et al. (2017) leverages metadata of networks to reconstruct the original dataset. Chen et al. (2019) trains a generator to synthesize images that are compatible with the teacher. Nayak et al. (2019) models the output distribution space as a Dirichlet distribution and updated the random noisy images to compose a transfer set. Micaelli and Storkey (2019) and Fang et al. (2019) incorporate the idea of adversarial training into knowledge distillation, measuring the discrepancy between the student and teacher. Yin et al. (2019) introduces DeepInversion to synthesize class-conditional images. Due to the discrete nature of language, none of the above methods can be applied to natural language tasks. Melas-Kyriazi et al. (2020) proposes a generation-distillation framework in low-data settings, which employs a finetuned GPT2 as the generator and a CNN as the student model. Different from methods above, we investigate the problem of compressing BERT with no data.

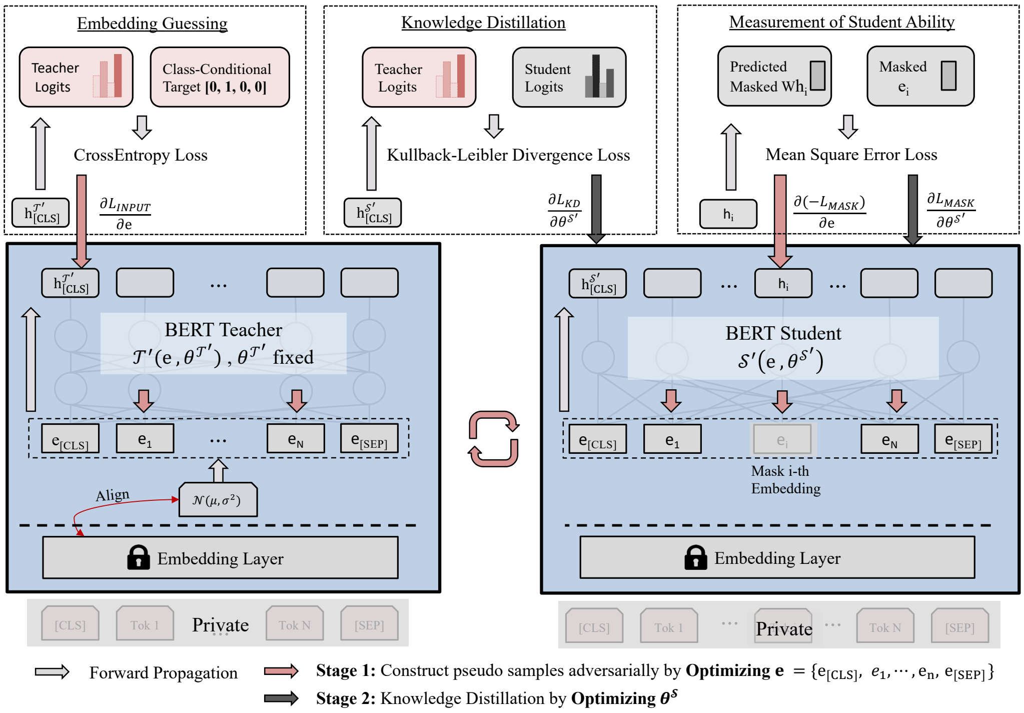


Figure 1: An overview of our two-stage Adversarial self-Supervised Data-Free Distillation framework. $\mathcal{T}'$ and $\mathcal{S}'$ contain transformer layers and classifier head. Firstly, when constructing synthetic samples, we iteratively guess and update the pseudo embeddings $\mathbf{e}$ under the feedback of the teacher's class-conditional supervision (top left) and the student's self-assessment (top right) in an adversarial training manner. Secondly, we use the generated sample $\mathbf{e}$ to distill knowledge (top middle). The parameters of embedding layer are fixed, and no inputs will go through the embedding layer when training.

3 Methods

In this section, we present our two-stage distillation framework named *Adversarial self-Supervised Data-Free Distillation* (AS-DFD). We craft well-trained embedding-level pseudo samples by controllable Plug & Play Embedding Guessing with alignment constraints (Section 3.1) and adversarially adapt synthetic embeddings under self-supervision of the student (Section 3.2). Using these pseudo samples, we transfer knowledge from the teacher to the student (Section 3.3). The workflow of AS-DFD is illustrated in Figure 1.

Problem Definition Knowledge Distillation is a compression technique to train a high-performance model with fewer parameters instructed by the teacher model (Hinton et al., 2015). Let $\mathcal{T}$ be a large transformer-based teacher model (12-layer BERT-base here) and $\mathcal{S}$ be a comparatively lightweight student model. For each sentence $\mathbf{x}$, the classification prediction can be formulated as:

$$\begin{aligned}\mathbf{e} &= \text{EmbeddingLayer}(\mathbf{x};\, \theta_{emb})\\ \mathbf{h} &= \text{TransformerLayers}(\mathbf{e};\, \theta_{layer})\\ \mathbf{y} &= \text{ClassifierLayer}(\mathbf{h}_{[\mathbf{CLS}]};\, \theta_{classifier})\end{aligned} \quad (1)$$

where θ_{emb}, θ_{layer}, $\theta_{classifier}$ represent parameters in the embedding layer, transformer layers and classification head respectively. $\mathbf{y}$ is the softmax probability output of $\mathbf{x}$ and $\mathbf{h}_{[\mathbf{CLS}]}$ denotes the hidden states in the last layer corresponding to the special token [CLS]. Parameters with superscript $\mathcal{T}$ belong to the teacher and $\mathcal{S}$ for the student.

Our goal of data-free knowledge distillation is to train the student parameters $\theta^{\mathcal{S}}$ with no data $\mathcal{X}$ available. In other words, we only have a teacher model $\mathcal{T}$ and we need to compress it.

3.1 Construct Pseudo Samples

Plug & Play Embedding Guessing In the data-free settings, we need to solve the dilemma of having no access to the original dataset. The major challenge is how to construct a set of highly reliable samples, from which the student can extract differential knowledge.

Our approach exploits representative knowledge hidden in the teacher's parameters in a Plug & Play manner (Nguyen et al., 2017; Dathathri et al., 2020). Given a sentence $\mathbf{x}$ and a label $\mathbf{y}$, the conditional probability can be written as $P(\mathbf{y}|\mathbf{x};\theta^{\mathcal{T}})$. When finetuning the teacher, we optimize parameters $\theta^{\mathcal{T}}$ towards higher probability. To capture impression of prior training data in the teacher's parameters, we invert the model and utilize the teacher's parameters to guide the generation of $\mathbf{x}$ by ascending $P(\mathbf{y}|\mathbf{x};\theta^{\mathcal{T}})$ with $\theta^{\mathcal{T}}$ fixed.

Due to the intractable discrete problem of text, gradients updated on $\mathbf{x}$ are pointless. Most language models transform discrete words into continuous embeddings. Inspired by this, we ignore the embedding layer and apply the updating on continuous representation space of embeddings. We name this generation process "Embedding Guessing". We randomly guess vectors $\mathbf{e} \in \mathbb{R}^{l \times d}$, feed them into the transformer blocks and get feedback from gradients to confirm or update our guess. l is the predefined length of sentence and d is the embedding dimensionality, which is 768 in BERT-base. Those target-aware embeddings can be obtained by minimizing the objective:

$$\mathcal{L}_{INPUT} = \sum_{\mathbf{e}\in\mathcal{E}} \mathrm{CE}\left(\mathcal{T}'(\mathbf{e};\theta^{\mathcal{T}'}), \hat{\mathbf{y}}\right) \qquad (2)$$

where $\mathcal{T}'$ takes pseudo embeddings $\mathbf{e}$ as input and contains $\mathrm{TransformerLayers}$ and $\mathrm{ClassifierLayer}$ in the teacher. $\theta^{\mathcal{T}'}$ includes $\theta^{\mathcal{T}}_{layer}$ and $\theta^{\mathcal{T}}_{classifier}$. $\hat{\mathbf{y}}$ is a random target class. CE refers to the cross-entropy loss. $\mathcal{E}$ is a batch of $\mathbf{e}$ initialized with Gaussian distribution. We update $\mathbf{e}$ for several iterations until convergence, representing that $\mathbf{e}$ is correct judged by the teacher. As for $\theta^{\mathcal{S}}_{emb}$, we share $\theta^{\mathcal{T}}_{emb}$ with $\theta^{\mathcal{S}}_{emb}$.

We argue that under the process of Embedding Guessing, pseudo embeddings $\mathbf{e}$ contain the target-specific information. Classification models need to find out differentiated characteristics which propitious to prediction over multiple categories. As the human learning process, examples given by teachers are encouraged to be representative and better reflecting the discrepancy among classes. Borrowed from this teaching strategy, we guess embeddings towards the direction of higher likelihood on target category and seek the local minimum regarding the target class, which reflects the characteristics of the target class within regions. In other words, these synthetic samples are more likely to comprise separation statistics between classes.

Making Pseudo-Embeddings More Realistic However, training on embeddings leads to a gap between the pseudo embeddings and the true underlying embeddings. Specifically, Embedding Guessing is independent of the parameter of the teacher's embedding and will shift the representational space. We add some additional constraints to ensure generated embeddings imitate the distribution of real data to a certain extent. Alignment strategies to restrain and reduce search space are listed as follows:

- Add $\mathbf{e}_{[\mathbf{CLS}]}$ and $\mathbf{e}_{[\mathbf{SEP}]}$ at both ends of the synthetic embeddings. $\mathbf{e}_{[\mathbf{CLS}]}$ and $\mathbf{e}_{[\mathbf{SEP}]}$ represent embeddings corresponding to [CLS] and [SEP].

- Continuously mask random length of embeddings from the tail of it. Lengths of sentences in batches are indeterminate and synthetic embeddings should cover this scenario.

- Adjust the Gaussian distribution to find the best initialization. Excessive initialization scope expands search space while small one converges to limited samples.

3.2 Adversarial self-Supervised Student

Modeling Learning Ability of the Student Effective teaching needs to grasp the student's current state of knowledge and dynamically adapt teaching strategies and contents. How to model the ability of the student without data? While processing natural language, the ability to analyze the context is an indicator of the student's capabilities and it can be quantified by a self-supervised module. Borrowing the idea of masking and predicting the entries randomly, we randomly mask one embedding in $\mathbf{e}$. Then, a new self-restricted objective is to predict the masked embedding with the following forums:

$$\begin{aligned} \mathbf{h} &= \mathcal{S}'\left(\mathbf{e}^{mask};\theta^{\mathcal{S}'}\right) \\ L_{MASK} &= \sum_{\mathbf{e}\in\mathcal{E}} \left\| \frac{\mathbf{e}_i}{\|\mathbf{e}_i\|_2} - \frac{W\mathbf{h}_i}{\|W\mathbf{h}_i\|_2} \right\|_2^2 \end{aligned} \qquad (3)$$

where $\mathbf{e}$ is randomly masked on position i and converted to $\mathbf{e}^{mask}$. $\mathbf{e}_i$ is the masked embedding and W is the parameters in the fully-connected supervised module for predicting masked embedding. $\mathcal{S}'$ acts the same way as $\mathcal{T}'$. Unlike the class-conditional guidance, the self-supervised module shifts the gradients with more concrete and diverse supervision from context.

Adversarial Training of the Student To enforce $\mathbf{e}$ with more valuable and diverse information, we encourage the student to adversarially search for samples that the student is not confident. Prior works (Micaelli and Storkey, 2019; Fang et al., 2019) maximize the discrepancy between the teacher and student to encourage difficulty in samples and avoid synthesizing redundant images. We design a self-assessed confrontational mechanism, which guides the pseudo embeddings towards greater difficulty by enlarging L_{MASK} in the constructing stage and enhances the student by decreasing L_{MASK} in the distillation stage. Here, L_{MASK} acts as the timely student's feedback to improve teaching.

3.3 Two-stage Training

Distillation Objective Students learn high-entropy knowledge from teachers by matching soft targets. Taking $\mathcal{E}$ as synthetic samples, we measure the distance between the teacher and student as:

$$L_{KL} = \sum_{\mathrm{e}\in\mathcal{E}} \mathrm{KL}\left(\mathcal{T}'(\mathrm{e};\theta^{\mathcal{T}'}), \mathcal{S}'(\mathrm{e};\theta^{\mathcal{S}'}), \tau\right) \quad (4)$$

where KL denotes the Kullback-Leibler divergence loss and τ is the distillation temperature.

We follow PKD (Sun et al., 2019) to learn more meticulous details for students. To capture rich features, we define the additional loss as:

$$L_{PT} = \sum_{\mathrm{e}\in\mathcal{E}} \left\| \frac{\mathbf{h}_{[\mathbf{CLS}]}{}^{\mathcal{T}}}{\left\|\mathbf{h}_{[\mathbf{CLS}]}{}^{\mathcal{T}}\right\|_2} - \frac{\mathbf{h}_{[\mathbf{CLS}]}{}^{\mathcal{S}}}{\left\|\mathbf{h}_{[\mathbf{CLS}]}{}^{\mathcal{S}}\right\|_2} \right\|_2^2 \quad (5)$$

The objective of distillation can be formulated as:

$$L_{KD} = L_{KL} + \alpha L_{PT} \quad (6)$$

where α balances these two losses.

Training Procedure We summarize the training procedure in algorithm 1. The multi-round training of AS-DFD splits into two steps: the construction stage and the distillation stage. In the construction stage, after randomly sampling vectors with alignment constraints, we repeat the adversarial training of pseudo embeddings for n_{iter} times. In each iteration, we guess embeddings under class-conditional supervision information for $n_{\mathcal{T}}$ steps, and the student is asked to predict and give negative feedback to guide pseudo-embeddings' generation for $n_{\mathcal{S}}$ steps. When distilling, we train $\theta^{\mathcal{S}'}$ as well as W with those pseudo samples.

4 Experiments

4.1 Datasets

We demonstrate the effectiveness of our methods on three widely-used text classification datasets: AG News, DBPedia, IMDb (Auer et al., 2007; Maas et al., 2011). The statistics of these datasets are shown in Table 1. For datasets without validation sets (DBPedia and IMDb), we randomly sample 10% of the train set as the validation set.

Algorithm 1: Two-stage Adversarial self-Supervised Data-Free Distillation

```
Input: Teacher model $\mathcal{T}$ with $\theta^{\mathcal{T}}$, $\mu$, $\sigma$
Output: Student model $\mathcal{S}$ with $\theta^{\mathcal{S}}$, $W$
Initial $\theta^{\mathcal{S}'}$ with $\theta^{\mathcal{T}'}$ and set $\theta^{\mathcal{S}}_{emb} \leftarrow \theta^{\mathcal{T}}_{emb}$
for $i \leftarrow 1$ to $N$ do
    // Stage 1: Construct Pseudo Samples
    Fix $\theta^{\mathcal{T}'}$, $\theta^{\mathcal{S}'}$ and $W$
    Sample $\mathcal{E} \sim \mathcal{N}(\mu, \sigma^2)$
    Add alignment constraints on $\mathcal{E}$
    for $iters \leftarrow 1$ to $n_{iter}$ do
        for $m \leftarrow 1$ to $n_{\mathcal{T}}$ do
            $\mathcal{E} \leftarrow \mathcal{E} - \eta \frac{\partial L_{INPUT}}{\partial \mathcal{E}}$
        end
        for $n \leftarrow 1$ to $n_{\mathcal{S}}$ do
            $\mathcal{E} \leftarrow \mathcal{E} - \eta \frac{\partial(-L_{MASK})}{\partial \mathcal{E}}$
        end
    end
    // Stage 2: Knowledge Distillation
    Fix $\theta^{\mathcal{T}'}$ and update $\theta^{\mathcal{S}'}$, $W$
    $\theta^{\mathcal{S}'} \leftarrow \theta^{\mathcal{S}'} - \xi \frac{\partial L_{KD}}{\partial \theta^{\mathcal{S}'}}$
    $W \leftarrow W - \xi \frac{\partial L_{MASK}}{\partial W}$
end
```

Dataset	Classes	Train	Valid	Test
AG News	4	114k	6k	7.6k
DBPedia	14	504k	56k	70k
IMDb	2	22.5k	2.5k	25k

Table 1: Statistics of AG News/DBPedia/IMDb. Training samples are only available when finetuning teacher models. AG News and DBPedia are topic classification datasets and IMDb is a dataset for binary sentiment classification.

4.2 Teacher/Student Models

We experiment with official uncased BERT-base (Devlin et al., 2018) as the teacher model (BERT$_{12}$) for its widespread use in downstream tasks. BERT-base has 12 layers of Transformer (Vaswani et al.,

	AG News	DBPedia	IMDb
Distill on Original Dataset			
Teacher - BERT_{12}	94.2	99.4	88.5
Student - BERT_6	94.1	99.3	87.0
Student - BERT_4	93.8	99.3	85.9
Train on Part of the Dataset			
fastText (Chia et al., 2019)	75.2	91.0	/
8-layer BlendCNN(Chia et al., 2019)	87.6	94.6	/
Data-Free Distillation - $BERT_6$ as student			
Random Text	85.4	93.9	77.1
Modified-ZSKT	88.4	-	78.1
Modified-ZSKD	88.6	97.1	78.2
AS-DFD (Ours)	**90.4**	**98.2**	**79.8**
Data-Free Distillation - $BERT_4$ as student			
Random Text	78.5	77.3	67.6
Modified-ZSKT	81.1	-	70.4
Modified-ZSKD	83.8	83.0	70.7
AS-DFD (Ours)	**88.2**	**94.1**	**77.2**

Table 2: Distillation accuracy on three datasets: AG news, DBPedia and IMDb. FastText and 8-layer BlendCNN are trained on 100 sentences per class. For fair comparision, Modified-ZSKT and Modified-ZSKD synthetic embeddings rather than images compared with its original algorithm. '-' means that accuracy cannot exceed the result of Random Text and '/' means the results are not reported in the paper. Results show that AS-DFD outperforms other baselines in data-free distillation.

2017) with 12 attention heads in each layer. We conduct experiments on student models with different transformer layers: 4-layer BERT (BERT_4) or 6-layer BERT (BERT_6). Statistics of parameters and inference time are listed in Table 3.

Layers	Params	Inference Time(s)
12	109M (1×)	26.9s (1×)
6	67M (1.63×)	14.1s (1.91×)
4	52M (2.10×)	9.5s (2.84×)

Table 3: Number of parameters and inference time for BERT_{12}, BERT_6 and BERT_4. Inference speed is tested on 7.6K samples from AG News.

4.3 Baselines

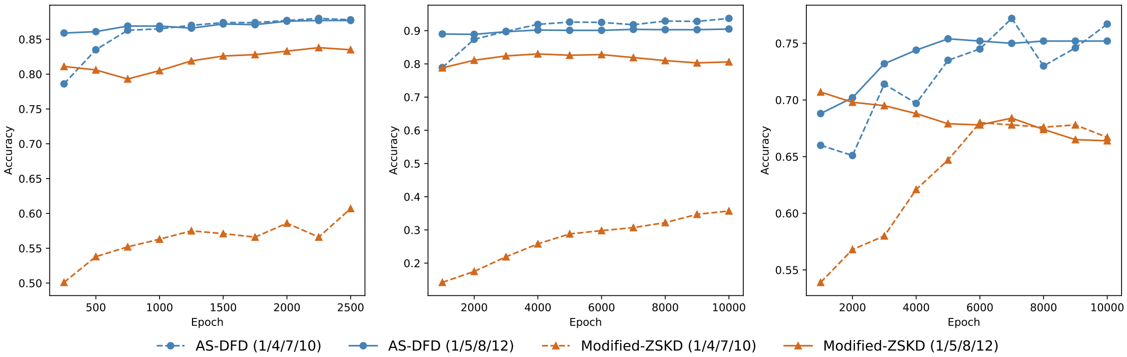
To the best of our knowledge, there is no data-free distillation method for language tasks. However, when slightly modifying the data-free distillation models that are effective in Computer Vision, these models can also work on language tasks. Imitating Plug & Play Embedding Guessing method, we plug those image generators/generation methods above the embedding layer to synthesize continuous embeddings (instead of images).

Except for a baseline of random selection of words, we choose two models that represent the mainstream approaches in data-free distillation of image classification. Baselines are described as follows:

Random Text We randomly select words from vocabulary and construct literally-uninterpretable sentences.

Modified-ZSKT Modified-ZSKT is extended from ZSKT (Micaelli and Storkey, 2019). ZSKT trains an adversarial generator to search for images in which the student's prediction poorly matches that of the teacher's and reaches state-of-the-art performance.

Modified-ZSKD Modified-ZSKD is derived from ZSKD (Nayak et al., 2019). ZSKD performs Dirichlet sampling on class probability and craft Data Impression. DeepInversion (Yin et al., 2019) extends ZSKD with feature distribution regularization in batch normalization and outperforms ZSKD. However, BERT is not suitable for this performance-enhancing approach (BERT has no BN or structure like BN to store statistics of training data) and DeepInversion cannot be the baseline of our method.



(a) AG News

(b) DBPedia

(c) IMDb

Figure 2: Comparison of AS-DFD versus Modified-ZSKD using different initializations. Experiments are conducted on the 4-layer BERT student. Dash lines show the result initialized with BERT's {1, 4, 7, 10} layers and solid lines with {1, 5, 8, 12} layers.

4.4 Experimental Results

We first show the performance of data-driven knowledge distillation. Then we show the effectiveness of AS-DFD methods. As shown in Table 2, AS-DFD with BERT_4 and BERT_6 performs the best on three datasets. For 6-layer BERT, our algorithm improves 1.8%, 1.1% and 1.6% compared to Modified-ZSKD, closing the distance between the teacher and student. Furthermore, when coaching the 4-layer student, our methods gain 4.4%, 11.1% and 6.5% increases, which significantly improves the distillation accuracy. It seems that AS-DFD performs better with higher compression rates compared with other data-free methods. However, there is still a large gap between the performance of data-drive distillation and data-free distillation.

As for other baselines, Random Text can be regarded as a special case of unlabeled text where models can extract information to infer on, especially on text classification tasks. We use it as a criterion to judge whether a model works. Modified-ZSKT performs worse than Random Text on DBPedia. The reason lies in the structure of the generator, which is designed for image generation and is not suitable for language generation. The strength of CNN-based generators lies in its ability to capture local and hierarchical features. However, it is difficult for CNN to capture global and sequential structures, which is essential for languages.

Implementation Details We train the AS-DFD with $n_{\mathcal{T}} = 5, n_{\mathcal{S}} = 1$ and $n_{iter} = 5$. Maximum sequence lengths for three datasets are set to 128. Ideally, the more samples generated, the higher the accuracy. We impose restrictions on the number of generated samples for each dataset. Training epochs are 2.5k(AG News), 10k(DBPedia), 10k(IMDb) with 48 samples per batch for all methods except ZSKT, which needs to train its generator from scratch (25k epochs in Modified-ZSKT). In our experiments, these samples are enough for models to reach a stable status. More implementation details about finetuning teachers and distilling students are listed in Appendix A.1.

Initialization We observe that students' performance is highly sensitive to initializations (especially the Random Text baseline). Fan et al. (2019) argues that different layers play different roles in BERT. We report results using different initialization schemes and show the stability of AS-DFD. Considering that the embedding layer is separated from transformer blocks when training, we strongly recommend sharing the first layer's parameters of the teacher with the student, which is also suggested in Xu et al. (2020). Specifically, we choose two sets of layer weights. One is {1, 4, 7, 10}, which is common in data-driven distillation, and the other is {1, 5, 8, 12}, which intentionally put the last layer's parameters in. We evaluate these initialization schemes on AS-DFD and Modified-ZSKD. To eliminate the effects of distillation, we ensure that hyperparameters in the distillation step are consistent in two models, which intuitively shows the disparity in samples' quality. We do not include Modified-ZSKT because samples of Modified-ZSKT vastly outnumber the other two approaches.

Experimental results are shown in Figure 2. Modified-ZSKD highly dependent on initialization, especially on AG news and DBPedia with 23.1% and 47.1% performance drop relatively. On the

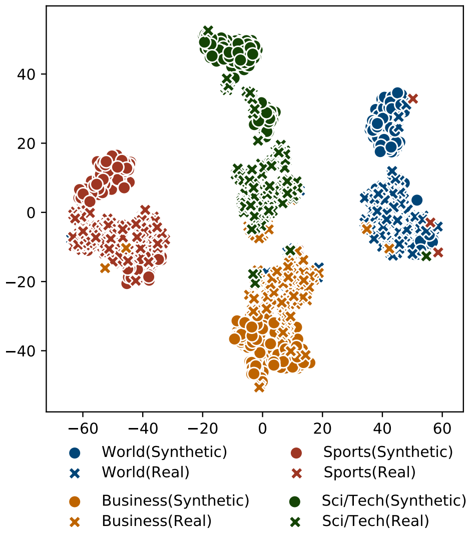


Figure 3: t-SNE dimensionality reduction results between real and synthetic samples on output of last layer.

contrary, initialization has limited impacts on AS-DFD. If pseudo-embeddings are initialized with worse parameters, our method still achieves better accuracy than other baselines (87.7% on AG News, 90.5% on DBPedia and 75.4% on IMDb). It shows that our method synthesizes higher-quality samples compared with Modified-ZSKD. Additionally, AS-DFD maintains an upward trend when the size of synthetic samples grows, suggesting that synthetic samples are useful for knowledge transfer.

Validity of Synthetic Embeddings Embeddings we generated are incomprehensible. We use t-SNE (Maaten and Hinton, 2008) to visualize the synthetic embeddings in comparison with the original dataset. As shown in Figure 3, samples generated by Embedding Guessing are close to the real samples and overlap with them to a certain extent.

4.5 Module Analysis

To verify the contribution of each module, we perform an ablation study and summarize it in Table 4.

Embedding Guessing is the foundation of the entire model. After drawing into the idea of Plug & Play Embedding Guessing, distillation performance is improved with stability, demonstrating that knowledge extracted from the teacher makes the synthetic samples reasonable. The embedding layer of the student model is completely separated in the generation-distillation process. Imitating BERT's input precisely narrows this gap, leading to a large improvement in accuracy. Additionally, choosing an appropriate normal distribution can effectively reduce search space and avoid generating completely irrelevant samples. We conduct experiments on different normal distributions in Appendix A.2.

Method	Accuracy
Random Noise	25.1
+ Embedding Guessing	44.2
+ Alignment Constraints	
+ Add [CLS] and [SEP]	80.3
+ Variable Length	82.2
+ Appropriate Gaussian Distribution	87.4
+ Adversarial self-Supervised Module	88.2

Table 4: Ablation study on AG News dataset. The Student model BERT$_4$ is initialized with BERT's 1st, 4th, 7th and 10th layers.

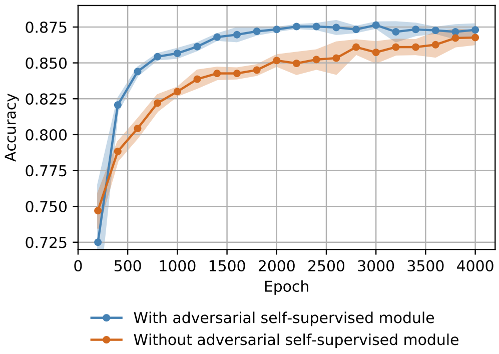


Figure 4: Accuracy curve with / without adversarial self-supervised module. The shaded area around the curve is the standard deviation over three seeds.

Effect of Adversarial self-Supervised Module To investigate whether the adversarial self-supervised module help data-free distillation, we conduct experiments on AG News to demonstrate the advantage of it in Figure 4.

We repeat each experiment 3 times and plot mean and standard deviation to reduce the contingency of experiments. With the adversarial self-supervised module, distillation converges faster and achieves higher accuracy. The number of epochs can be reduced to 2500, saving half of the time. As shown in the curve, AS-DFD does not perform well in the early stage since the self-supervised module is underfitting. After training for a while, the self-supervised module can grasp the student's ability and provide corrective feedback to synthesize more challenging samples.

5 Conclusion

In this paper, we propose AS-DFD, a novel data-free distillation method applied in text classification tasks. We use Plug & Play Embedding Guessing with alignment constraints to solve the problem that gradients cannot update on the discrete text. To dynamically adjust synthetic samples according to students' situations, we involve an adversarial self-supervised module to quantify students' abilities. Experimental results on three text datasets demonstrate the effectiveness of AS-DFD.

However, it's still challenging to ensure the diversity of generated embeddings under the weak supervision signal and we argue that the gap between synthetic and real sentences still exists. In the future, we would like to explore data-free distillation on more complex tasks.

Acknowledgments

This work is supported by the National Key Research and Development Project of China (No. 2018AAA0101900), the Fundamental Research Funds for the Central Universities (No. 2019FZA5013), the Zhejiang Provincial Natural Science Foundation of China (No. LY17F020015), the Chinese Knowledge Center of Engineering Science and Technology (CKCEST) and MOE Engineering Research Center of Digital Library.

References

Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. 2007. Dbpedia: A nucleus for a web of open data. In *The semantic web*, pages 722–735. Springer.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450*.

Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015. A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326*.

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

Hanting Chen, Yunhe Wang, Chang Xu, Zhaohui Yang, Chuanjian Liu, Boxin Shi, Chunjing Xu, Chao Xu, and Qi Tian. 2019. Data-free learning of student networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3514–3522.

Yew Ken Chia, Sam Witteveen, and Martin Andrews. 2019. Transformer to cnn: Label-scarce distillation for efficient text classification. *arXiv preprint arXiv:1909.03508*.

Sumanth Dathathri, Andrea Madotto, Janice Lan, Jane Hung, Eric Frank, Piero Molino, Jason Yosinski, and Rosanne Liu. 2020. Plug and play language models: A simple approach to controlled text generation. In *International Conference on Learning Representations*.

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*.

Angela Fan, Edouard Grave, and Armand Joulin. 2019. Reducing transformer depth on demand with structured dropout. *arXiv preprint arXiv:1909.11556*.

Gongfan Fang, Jie Song, Chengchao Shen, Xinchao Wang, Da Chen, and Mingli Song. 2019. Data-free adversarial distillation. *arXiv preprint arXiv:1912.11006*.

Mitchell A Gordon, Kevin Duh, and Nicholas Andrews. 2020. Compressing bert: Studying the effects of weight pruning on transfer learning. *arXiv preprint arXiv:2002.08307*.

Fu-Ming Guo, Sijia Liu, Finlay S Mungall, Xue Lin, and Yanzhi Wang. 2019. Reweighted proximal pruning for large-scale language representation. *arXiv preprint arXiv:1909.12486*.

Geoffrey Hinton, Oriol Vinyals, and Jeffrey Dean. 2015. Distilling the knowledge in a neural network. In *NIPS Deep Learning and Representation Learning Workshop*.

Ferenc Huszár. 2015. How (not) to train your generative model: Scheduled sampling, likelihood, adversary? *arXiv preprint arXiv:1511.05101*.

Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. 2019. Tinybert: Distilling bert for natural language understanding. *arXiv preprint arXiv:1909.10351*.

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

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. Albert: A lite bert for self-supervised learning of language representations. In *International Conference on Learning Representations*.

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

Raphael Gontijo Lopes, Stefano Fenu, and Thad Starner. 2017. Data-free knowledge distillation for deep neural networks. *arXiv preprint arXiv:1710.07535*.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA. Association for Computational Linguistics.

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

J Scott McCarley. 2019. Pruning a bert-based question answering model. *arXiv preprint arXiv:1910.06360*.

Luke Melas-Kyriazi, George Han, and Celine Liang. 2020. Generation-distillation for efficient natural language understanding in low-data settings. *arXiv preprint arXiv:2002.00733*.

Paul Micaelli and Amos J Storkey. 2019. Zero-shot knowledge transfer via adversarial belief matching. In *Advances in Neural Information Processing Systems*, pages 9547–9557.

Paul Michel, Omer Levy, and Graham Neubig. 2019. Are sixteen heads really better than one? In *Advances in Neural Information Processing Systems*, pages 14014–14024.

Subhabrata Mukherjee and Ahmed Hassan Awadallah. 2019. Distilling transformers into simple neural networks with unlabeled transfer data. *arXiv preprint arXiv:1910.01769*.

Gaurav Kumar Nayak, Konda Reddy Mopuri, Vaisakh Shaj, R Venkatesh Babu, and Anirban Chakraborty. 2019. Zero-shot knowledge distillation in deep networks. *arXiv preprint arXiv:1905.08114*.

Anh Nguyen, Jeff Clune, Yoshua Bengio, Alexey Dosovitskiy, and Jason Yosinski. 2017. Plug & play generative networks: Conditional iterative generation of images in latent space. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4467–4477.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100, 000+ questions for machine comprehension of text. In *EMNLP*.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Siqi Sun, Yu Cheng, Zhe Gan, and Jingjing Liu. 2019. Patient knowledge distillation for BERT model compression. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4323–4332, Hong Kong, China. Association for Computational Linguistics.

Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, and Lior Wolf. 2014. Deepface: Closing the gap to human-level performance in face verification. *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1701–1708.

Raphael Tang, Yao Lu, and Jimmy Lin. 2019a. Natural language generation for effective knowledge distillation. In *Proceedings of the 2nd Workshop on Deep Learning Approaches for Low-Resource NLP (DeepLo 2019)*, pages 202–208, Hong Kong, China. Association for Computational Linguistics.

Raphael Tang, Yao Lu, Linqing Liu, Lili Mou, Olga Vechtomova, and Jimmy Lin. 2019b. Distilling task-specific knowledge from bert into simple neural networks. *arXiv preprint arXiv:1903.12136*.

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.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Gregory S. Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *ArXiv*, abs/1609.08144.

Canwen Xu, Wangchunshu Zhou, Tao Ge, Furu Wei, and Ming Zhou. 2020. Bert-of-theseus: Compressing bert by progressive module replacing. *arXiv preprint arXiv:2002.02925*.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*, pages 5754–5764.

Hongxu Yin, Pavlo Molchanov, Zhizhong Li, Jose M Alvarez, Arun Mallya, Derek Hoiem, Niraj K Jha, and Jan Kautz. 2019. Dreaming to distill: Data-free knowledge transfer via deepinversion. *arXiv preprint arXiv:1912.08795*.

Ofir Zafrir, Guy Boudoukh, Peter Izsak, and Moshe Wasserblat. 2019. Q8bert: Quantized 8bit bert. *arXiv preprint arXiv:1910.06188*.

Sanqiang Zhao, Raghav Gupta, Yang Song, and Denny Zhou. 2019. Extreme language model compression with optimal subwords and shared projections. *arXiv preprint arXiv:1909.11687*.

A Appendices

A.1 Implementation Details

Hyperparameters in Finetuning Teachers We finetune BERT-base on three datasets mentioned above. We train our teacher models with Adam (Kingma and Ba, 2014) in 4 epochs. Learning rate is set to 2e-5 with a scheduler that linearly decreases it after 10% warmup steps. We set the maximum sequence length to 128 and batch size to 32 for all datasets.

Hyperparameters in Data-Free Distillation AS-DFD is trained on 1 TITAN Xp GPU. We set batch size to 48 with the student's learning rate ξ from $\{5 \times 10^{-5}, 2 \times 10^{-5}, 1 \times 10^{-5}\}$ and embedding learning rate η from $\{1 \times 10^{-2}, 5 \times 10^{-3}, 1 \times 10^{-3}\}$. We conduct an additional search over α from $\{100, 200, 250, 350, 500\}$ and select the hyperparameters with the highest accuracy. In our experiment, η equals to 1×10^{-2} and ξ equals to 1×10^{-5}. α is set to 250. Temperature $\tau = 1$ works well in our model. In the distillation step, we use Adam with a warmup proportion of 0.1 and we linearly decay the learning rate. In the construction step, the learning rate is fixed with Adam optimizer. There may be no validation set under data-free settings, which makes tuning parameters impossible. We experiment with the hyperparameters performed best on AG News and find that this set of parameters also performs well on the other two datasets.

A.2 Adjust Gaussian Distributions

The other two parameters are the mean and standard deviation for Gaussian sampling. We found in our experiments that standard deviation has a great influence on the student's performance. If vectors are initialized with small standard deviation(e.g. std=0.05, see Figure 5.b), generated samples in each category gather together, meaning that they aggregate to limited regions and leading to insufficient diversity of pseudo samples. Real data samples show no aggregation under t-SNE(see Figure 5.a). A higher standard deviation(e.g. std=1) indicates that samples are spread out from the mean, which will increase the search space and far from the embedding's distribution of BERT. It is also reflected in our testing accuracy with 83.2, 85.3, 88.2, 83.2 corresponding to $\mathcal{N}(0, 0.05^2)$, $\mathcal{N}(0, 0.2^2)$, $\mathcal{N}(0, 0.35^2)$, $\mathcal{N}(0, 1^2)$. We search standard deviations over {0.05, 0.1, 0.2, 0.25, 0.3, 0.35, 0.4, 0.5, 1} and choose 0.35 to be the best standard deviation, which works well on all three datasets.

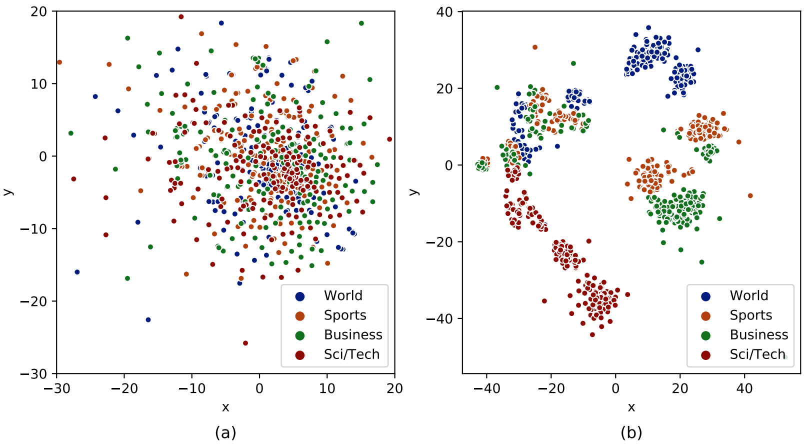


Figure 5: t-SNE results on real samples(a) or synthetic samples(b)