Distilling Task-Specific Knowledge from BERT via Adversarial Belief Matching

Huanru Henry Mao  
CSE, UC San Diego  
hhmao@eng.ucsd.edu

Bodhisattwa Prasad Majumder  
CSE, UC San Diego  
bmajumde@eng.ucsd.edu

Julian McAuley  
CSE, UC San Diego  
jmcauley@eng.ucsd.edu

Garrison W. Cottrell  
CSE, UC San Diego  
gary@eng.ucsd.edu

1 Motivation

Large pre-trained language models such as BERT [1] have achieved strong results when fine-tuned on a variety of natural language tasks but are cumbersome to deploy. Applying knowledge distillation (KD) [2] to compress these pre-trained models for a specific downstream task is challenging due to the small amount of task-specific labeled data, resulting in poor performance by the compressed model. Considerable efforts have been spent to improve the distillation process for BERT, involving techniques such as leveraging intermediate hints [3], student pre-training [4] and data augmentation [5].

The success of these methods lead us to hypothesize that the core issue of distilling fine-tuned models is due to the small quantity of data used during the distillation phase, which inhibits seeing sufficient variety of the teacher’s output distribution. Although broadly applicable to KD [6], this problem is exacerbated in the fine-tuned setting because much of the teacher’s knowledge is learned in the pre-training phase and may not be accessible by the student through task-specific data. Using a small number of samples lead to the student to only see a small portion of the teacher’s knowledge, making the knowledge transfer process incomplete. TinyBERT [4] mitigated this by applying a general pre-training stage on the student using BERT’s masked language objective. However, this pretraining step is computationally expensive, requires obtaining a large amount of pre-training data, and is not optimally tailored to the downstream task. Furthermore, in some scenarios the pre-training data may be inaccessible or the pre-training objective may be unknown.

In this paper, we focus on tackling task-specific distillation in the low data setting. We aim to distill a BERT model that is fine-tuned on some downstream task (referred to as teacher) into a smaller student model. Instead of pre-training the student using a general pre-training method, we propose to tailor our pre-training to the task by training the student on adversarially generated data. To compliment learning, we also incorporate intermediate hints [6] and Sobolev distillation [7] to our learning objective, which enables us to extract more information from the teacher per example.

2 Method

Our experiments aim to perform task-specific distillation of a 12 layer BERT into a 6 layer student Transformer. We first pre-train BERT on a downstream GLUE task to obtain a task-specific BERT model and refer to this model as the teacher. We then initialize our 6 layer student model by cloning the teacher and deleting every other layer. Similar to [8], we find that having teacher initialization is important for successful KD. This yields us a 6 layer Transformer model. Our KD method consists of two stages as described below.

Preprint. Under review.
Table 1: Validation accuracy for distilled models on a subset of GLUE tasks.

<table>
<thead>
<tr>
<th>Model</th>
<th>MRPC</th>
<th>SST</th>
<th>SST-10K</th>
</tr>
</thead>
<tbody>
<tr>
<td>Teacher</td>
<td>84.56%</td>
<td>90.83%</td>
<td>90.83%</td>
</tr>
<tr>
<td>Knowledge Distillation</td>
<td>80.15%</td>
<td>88.30%</td>
<td>86.90%</td>
</tr>
<tr>
<td>Our Method</td>
<td>85.54%</td>
<td>89.33%</td>
<td>88.75%</td>
</tr>
<tr>
<td>w/o Sobolev Distillation</td>
<td>85.05%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>w/o Intermediate Hints</td>
<td>84.31%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Replace ABM with Random Pre-training</td>
<td>84.07%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>w/o ABM or Random Pre-training</td>
<td>82.11%</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

2.1 Pre-training Stage

The aim of the pre-training stage is to distill a broad set of knowledge from the teacher by asking the student model to explore the teacher’s input space. A naive approach to exploring the teacher’s input space is by randomly sampling a string of garbage inputs that do not make sense to humans. This method of extracting a teacher’s output has been shown to be surprisingly useful for black box model extraction [9]. We propose a more nuanced approach based on Adversarial Belief Matching (ABM) [10] which crafts a targeted input that maximizes the KL divergence between the teacher’s output distribution and the student’s output distribution. We adapt the original ABM method for discrete sequences of text by applying first permuting the input sequence within the continuous embedding space, then quantizing the input such that it becomes a valid discrete sequence. Intuitively, this adversarial input explores the input spaces that the student fails to approximate the teacher well, which leads to more efficient training.

2.2 Fine-tuning Stage

The fine-tuning stage involves training the student via KD on the target downstream dataset. We minimize the cross entropy loss between the student and teacher’s output distributions. We can express the training objective of the student as

\[
\text{CE}(\sigma(z), \sigma(\tilde{z})) + \frac{1}{2} \|u_h(x; W_{\text{teacher}}) - v_g(x; W_{\text{student}})\|^2 + \mathbb{E}_v \left[ \|\nabla_z \langle z, v \rangle - \nabla_{\tilde{z}} \langle \tilde{z}, v \rangle \|^2 \right]
\]

where \(z\) and \(\tilde{z}\) are the logits of the student and teacher model respectively, CE denotes the cross entropy function. For intermediate hints, \(u_h\) and \(v_g\) are the deep functions for the teacher and the student respectively and we calculate the MSE between the outputs of hidden layers from teacher and student. For Sobolev, we compute the squared error between the output jacobian estimates of teacher and student, with an expectation taken with respect to \(v\) coming from a uniform distribution over a unit sphere.

3 Results

We choose two downstream GLUE tasks (MRPC, SST) for our experiments. MRPC has 3.7K and SST has 67K training samples. We trained a baseline “Knowledge Distillation” setup with teacher initialization for comparison, which is similar to DistillBERT [8]. Our results (Table 1) indicate that our two-stage distillation method significantly improves upon the baseline for MRPC (and even exceeding the teacher), while marginally improving upon the result for SST. We suspect this is because SST is a 18x larger dataset than MRPC, and thus the benefit of exploring the teacher’s input space diminishes as more samples cover the teacher’s input distribution. We confirm this by rerunning the same setup with SST-10K, which is a smaller version of SST that contains only 10K random samples. We find that our method improves performance more when there are fewer samples.

In addition, we performed ablation experiments on MRPC to see the effects of each technique to the overall performance. Our results show that pre-training is the most important contribution and capable of lifting performance by 3% from final performance. Performing KD pre-training using random garbage inputs yields a strong baseline when compared to ABM pre-training. Sobolev distillation and intermediate hints provide complementary and minor benefits.
References


