Reinforced Multi-Teacher Selection for Knowledge Distillation

Fei Yuan\textsuperscript{1}\textsuperscript{*}, Linjun Shou\textsuperscript{2}, Jian Pei\textsuperscript{3}, Wutao Lin\textsuperscript{2}, Ming Gong\textsuperscript{2}, Yan Fu\textsuperscript{1}, Daxin Jiang\textsuperscript{2,1}

\textsuperscript{1}University of Electronic Science and Technology of China
\textsuperscript{2}Microsoft STCA NLP Group
\textsuperscript{3}School of Computing Science, Simon Fraser University

Corresponding author.
Work is done during internship at Microsoft STCA NLP Group.

Abstract

In natural language processing (NLP) tasks, slow inference speed and huge footprints in GPU usage remain the bottleneck of applying pre-trained deep models in production. As a popular method for model compression, knowledge distillation transfers knowledge from one or multiple large (teacher) models to a small (student) model. When multiple teacher models are available in distillation, the state-of-the-art methods assign a fixed weight to a teacher model in the whole distillation. Furthermore, most of the existing methods allocate an equal weight to every teacher model. In this paper, we observe that, due to the complexity of training examples and the differences in student model capability, learning differentially from teacher models can lead to better performance of student models distilled. We systematically develop a reinforced method to dynamically assign weights to teacher models for different training instances and optimize the performance of student model. Our extensive experimental results on several NLP tasks clearly verify the feasibility and effectiveness of our approach.

Introduction

Deep pre-trained models, such as BERT (Devlin et al. 2018), XLNet (Yang et al. 2019), RoBERTa (Liu et al. 2019) and ALBERT (Lan et al. 2019), have proved effective on many NLP tasks by establishing record-breaking state-of-the-art results. However, due to huge amounts of model parameters, typically at the magnitude of hundreds of millions or even billions, the bottleneck for applying those pre-trained models in production is the slow inference speed and the huge footprints in using GPUs. To save the computation cost and speed up the inference process, knowledge distillation (KD) (Hinton, Vinyals, and Dean 2015) as an effective approach to compress large models into smaller ones stands out and becomes the de facto best choice among other alternatives, such as pruning (Han, Mao, and Dally 2015; Han et al. 2015) and quantization (Gong et al. 2014).

The knowledge distillation approach is based on a teacher-student learning paradigm, where a teacher model, which is often large, is taken and the output of the teacher model is integrated as a soft target in the loss function to train a student model, which is often small. The teacher-student learning paradigm demonstrates excellent performance on various NLP tasks (Kim and Rush 2016; Tang et al. 2019). Knowledge distillation methods start from a single teacher. Some recent methods employ multiple teachers and show great promises in further boosting student model performance effectively.

Most of the existing methods using multiple teachers simply assign an equal weight to all teacher models during the whole distillation process. The uniform distribution of weights among all teachers keeps the coordination, management and implementation of the multi-teacher framework simple. At the same time, the indifference to strengths and weaknesses of various teacher models leaves a huge untouched space for better knowledge distillation. To make a multi-teacher approach work well, teacher models have to be diverse. Exploiting the diverse strengths of various teacher models can bring in huge advantages.

\begin{table}
\centering
\begin{tabular}{llll}
\hline
Teacher & Student & MRPC & MNLI-mm \\
\hline
BERT-Base & N / A & 83.3 & 83.8 \\
RoBERTa-Base & N / A & 88.5 & 86.8 \\
BERT-Base & BERT\textsubscript{1} & 74.0 & 75.3 \\
RoBERTa-Base & BERT\textsubscript{1} & 73.0 & 71.9 \\
\hline
\end{tabular}
\caption{RoBERTa-Base performs better than BERT-Base. However, the student model distilled from BERT-Base, the weaker model, performs better than the same student model distilled from RoBERTa-Base, the stronger teacher model.}
\end{table}

Individual teacher models may perform differently on various instances. Different models may vary in hypothesis space, optimization strategy, parameter initialization and many other factors, which result in different performance among different cases. Ideally, we want to assign different weights to different teacher models for different training instances according to their performance in individual cases.

Differentiating among teacher models is far from trivial. Surprisingly, a stronger teacher model may not necessarily lead to a better student model. As shown in Table 1 (Sun et al. (2019)), the RoBERTa-Base model performs better than...
the BERT-Base model on the MRPC and MNLI-mm tasks. However, the student model using three-layer transformer BERT distilled from the weaker teacher model performs better on the same tasks than the same student model distilled from the stronger teacher model. One possible reason is that the effectiveness of distillation may be bounded by the capability of the student model. A simple student model with fewer parameters may not be able to approximate a very complex teacher model, since the complex teacher model may capture finer-grained patterns in data and cause the student model to overfit in some parts of the data and under some other parts. To achieve good distillation, we have to choose teacher models matching capacities of student models.

Based on the above insights, in this paper, we systematically study how to coordinate teacher models and student models in knowledge distillation. Specifically, we investigate how to assign appropriate weights to different teacher models on various training samples. To the best of our knowledge, we are the first to treat teacher models differentially at instance level in knowledge distillation.

We formulate the teacher model selection problem under a reinforcement learning framework: the decision is made based on the characteristics of training examples and the outputs of teacher models, while the policy is learned towards maximizing the student performance as the return.

To verify the effectiveness of our approach in NLP, we conduct extensive experiments on several important tasks from the GLUE benchmark (Wang et al. 2019), including sentiment analysis, paraphrase similarity matching and natural language inference. Our experimental results clearly show that our reinforced multi-teacher selection strategy boosts the performance of student models substantially. Our method is not only principled but also practical in major NLP tasks.

Our contributions are twofold. First, this is the first systematic study developing sample instance-based weighting for multiple teacher models. This is a concrete and novel contribution to knowledge distillation and machine learning. Our idea is general and can be used in many applications of knowledge distillation. Second, we apply our novel idea of reinforced multi-teacher selection to a series of important NLP tasks. This is a novel contribution to the NLP domain.

Related Work

To achieve model compression, Hinton, Vinyals, and Dean (2015) propose a knowledge distillation (KD) approach based on a teacher-student framework, which substantially extends the method by Buciluă, Caruana, and Niculescu-Mizil (2006). The KD approach has been widely adopted in many applications. For example, Kim and Rush (2016) demonstrate that standard knowledge distillation is effective for neural machine translation. Recent studies (Tang et al. 2019; Sun et al. 2019; Jiao et al. 2019) distill knowledge from BERT (Devlin et al. 2018) into small student models and achieve competent results. All those methods distill knowledge from one teacher model.

To improve the performance of student models that employ deep neural networks, some recent studies leverage multiple teacher models in knowledge distillation. Chebotar and Waters (2016) simply leverage the weighted average of teacher models to distill student model, where weights are hyper-parameters and fixed during training. Fukuda et al. (2017) examine two strategies to leverage labels from multiple teacher models in training student models. The first strategy updates the parameters of the student model by combining the soft labels from the teacher models with fixed weights. The other strategy randomly selects one teacher model at the mini-batch level to provide soft target labels for training student models. Wu, Chiu, and Wu (2019) also assign a fixed weight to each teacher model and use the weighted average of the probability distributions from multiple teacher models to train a student model. Yang et al. (2020) propose a two-stage multi-teacher KD method, which first pre-trains a student model for the Q&A distillation task and then fine-tunes the pre-trained student model with multi-teacher distillation on downstream tasks. Teacher models are assigned with equal weights during distillation.

All previous knowledge distillation methods using multi-teacher models fix the same weight for a teacher model on all training examples. In this paper, we learn a policy to dynamically assign weights to teacher models based on individual examples.

Our Approach

In this section, we first recall the preliminaries of knowledge distillation and teacher models/student models. Then, we present our reinforced teacher selection method. Last, we introduce the model training algorithm.

Preliminaries

In general, the knowledge distillation approach (Buciluă, Caruana, and Niculescu-Mizil 2006; Hinton, Vinyals, and Dean 2015) uses the soft output (logits) of one or multiple large models as the knowledge and transfers the knowledge to a small student model. In this paper, we mainly target at NLP tasks and thus use the state-of-the-art NLP models as examples to illustrate our approach. Our approach in general can be applied to all knowledge distillation tasks using teacher/student models.

Given a NLP task, let $D = \{(x_i, y_i)\}_{i=1}^N$ be the training set with $N$ training examples, where $x_i$ is the $i$-th input, such as a single sentence or a pair of sentences, and $y_i$ is the corresponding ground truth label. Without loss of generality, we assume that a class label $c$ is an integer between 1 and $C$, where $C$ is the number of classes in the data set.

We assume $K$ teacher models. For example, we can fine-tune the pre-trained models with 12-layer transformers, such as BERT (Devlin et al. 2018), XLNet (Yang et al. 2019), RoBERTa (Liu et al. 2019) or ALBERT (Lan et al. 2019), on the training data as possible teacher models. Denote by $M_k$ ($1 \leq k \leq K$) the $k$-th teacher model, and by $\Theta^k$ the set of model parameters of $M_k$. The output of teacher model $M_k$ for a given input $x_i$ is written as $\tilde{y}_{i,k} = (\tilde{y}_{i,k,1}, \ldots, \tilde{y}_{i,k,C})$, where $\tilde{y}_{i,k,c} = P^k(y_i=c|x_i; \Theta^k)$ is the probability of $x_i$ belonging to class $c$ ($1 \leq c \leq C$) computed by model $M_k$.

For student models, we consider transformer models with fewer layers, such as those with only 3 or 6 layers. To train a student model, the distillation loss is defined as the cross-entropy loss between the predictions of the teacher model
Knowledge Distillation

Evaluation Performance on Golden Labels

Student Model

Aggregated Soft Labels of Selected Teachers

Reward

Select or Not

Action

\( \pi_a(s, a) \)

Teacher Model 1

Soft Label

Select or Not

Teacher Model 2

Soft Label

... Teacher Model K

Soft Label

Select or Not

Select or Not

\[ P_1 \cdot P_2 \cdot P_k \]

Figure 1: Overview of RL-KD, Reinforced Multi-Teacher Selection for Knowledge Distillation.

and that of the student model, that is,

\[ \mathcal{L}_{DL} = -\frac{1}{K} \sum_{i, k, c} \left[ \tilde{y}_{i, k, c} \cdot \log P^s(y_i = c|x_i; \Theta^s) \right] \]  

(1)

where \( \Theta^s \) is the set of parameters for the student model and \( P^s(y_i = c|x_i; \Theta^s) \) is the probability of example \( x_i \) belonging to class \( c \) computed by the student model.

Besides the distillation loss, a ground-truth loss is also introduced in the training of student model.

\[ \mathcal{L}_{CE} = -\sum_{i, c} \mathbb{I}[y_i = c] \cdot \log P^s(y_i = c|x_i; \Theta^s) \]  

(2)

where \( \mathbb{I}[y_i = c] \) is the one-hot vector from the ground truth label. The final objective of KD is

\[ \mathcal{L}_{KD} = \alpha \mathcal{L}_{DL} + (1 - \alpha) \mathcal{L}_{CE} \]  

(3)

where \( \alpha \) is a hyper-parameter to balance between the soft teacher label \( \mathcal{L}_{DL} \) and the hard ground-truth label \( \mathcal{L}_{CE} \).

**RL based Teacher Selector (TS)**

The original KD methods (Buciluă, Caruana, and Niculescu-Mizil 2006; Hinton, Vinyals, and Dean 2015) assumes only one single teacher model. Although several recent studies employ multiple teacher models in distillation, as reviewed in Related Work, those methods assign a fixed weight to each teacher model. Inspired by the insights discussed in Introduction, we propose to dynamically assign weights to teacher models at instance level using a reinforcement approach (Sutton 1992). The weights are implemented as the sampling probabilities calculated by the reinforcement learning policy function, as to be explained in Equation 6.

Figure 1 shows the overview of our Reinforcement Learning based Knowledge Distillation approach (RL-KD for short). In each iteration, an agent interacts with the environment and receives the representation of a training instance.

The agent then applies the teacher selection policy and samples the teachers to participate in the knowledge distillation step. The outputs of the sampled teachers are integrated into a soft label to train the student model. After an episode of training examples, the performance of the trained student model is used as the reward to update the policy parameters. This process iterates on episodes until the performance of the student model converges.

In general, a reinforcement learning approach involves elements in the form of \( (state, action, reward) \). The elements in our method are as follows.

**State** Our reinforcement learning method maintains a series of environment states \( s_1, s_2, \ldots \) that summarize the characteristics of the input instances as well as the candidate teacher models, so that judicious decisions can be made accordingly. We design a state \( s_j \) as a vector of real-values \( F(s_j) \), which includes the concatenation of three features.

The first feature is a vector representation \( R(x_i) \in \mathbb{R}^d \) of an input instance \( x_i \). In general, any sentence-encoding network can be applied here to embed a sentence in a NLP task into a vector. In this paper, we use the embedding for the [CLS] token in the last layer of the pre-trained BERT-Base model (Devlin et al. 2018).

The representation model \( R \) here is different from a BERT-Base teacher model \( M_k \). \( R \) is a pre-trained BERT-Based model aiming to effectively represent the content of input instances. A teacher model \( M_k \) is fine-tuned for a specific task, such as predicting the class label distribution given an input instance.

The second feature is the probability vector \( \{P^i_k(y_i = c|x_i; \Theta^i_k)\}_{c=1}^C \) on all classes \( 1, \ldots, C \) predicted by the teacher model \( M_k \) with parameters \( \Theta^i_k \) on the input instance \( x_i \). In practice, the probability is often derived from a softmax function on the hidden representation of the input, that is,

\[ P^i_k(y_i = c|x_i; \Theta^i_k) = \text{softmax}(W_k \cdot M_k(x_i)) \]  

(4)

where \( [] \) refers to the \( c \)-th element in a vector, \( W_k \in \mathbb{R}^{C \times d} \) is a trainable weight matrix, and \( M_k(x_i) \) is the embedding of the [CLS] token in the last layer of \( M_k \).

The third feature is the cross entropy loss \( L_k(x_i) \) on the input instance \( x_i \) of the teacher model \( M_k \) with parameters \( \Theta^i_k \), which is the ground-truth loss of teacher model for \( x_i \).

\[ L_k(x_i) = -\sum_c \mathbb{I}[y_i = c] \cdot \log P^i_k(y_i = c|x_i; \Theta^i_k) \]

(5)

For the second and third features, we also concatenate the predictions of other teacher models as state features for the policy agent to better select teachers.

**Action** Each teacher model \( M_k \) is associated with one agent. An agent chooses between two possible actions, selecting the teacher model or not for the current instance. A policy function \( \pi_\theta(s_j, a_j) \) determines the distribution over the states, from which the action value of \( a_j \in \{0, 1\} \) is sampled. Denote by \( \theta \) the trainable parameters in the policy function. In this paper, we adopt a simple logistic function as
Algorithm 1 shows the overall RL-KD approach. At the beginning, we initialize the student model $\Theta_0$ by employing all teacher models in the knowledge distillation, that is, $a_{jk}$ is set to 1 for all $\mathcal{M}_k$ and for all instances $x_i$. We then initialize parameter $\theta$ for the policy function using the same setting under $\Theta_0$. After initialization, we iteratively conduct knowledge distillation and teacher selection in turn.

As described in Algorithm 2, in the KD process, we fix the teacher selection policy $\theta$ and learn the student model $\Theta^\circ$. In the TS process, we fix $\Theta^\circ$ to calculate the return and optimize the teacher selection policy $\theta$. The iteration continues for $L$ epochs. For more implementation details, please refer to Section “Implementation Details”.

### Experiments

In this section, we first describe our experimental setup, and then introduce the baseline methods and implementation details of our method. Last, we report the experimental results.

#### Data Sets and Evaluation Metric

Following Patient KD (Sun et al. 2019), we evaluate our proposed approach on three different NLP tasks from the GLUE benchmark (Wang et al. 2019), namely Sentiment Classification (SC), Paraphrase Similarity Matching (PSM) and Natural Language Inference (NLI). The statistics of the data sets are shown in Table 2. We use prediction accuracy as the metric in evaluation.

For SC, we experiment on Stanford Sentiment Treebank (SST-2) (Socher et al. 2013). The target is to predict the sentiment of a given sentence.
We fine-tune the pre-trained models BERT, ALBERT, and XLNet on task specific training data to get four candidate teacher models, where the subscript 12 means 12 layers of transformers. Similar to Patient KD (Sun et al. 2019), we consider the BERT models with 3 layers and 6 layers of transformers as two student models, denoted by BERT3 and BERT6, respectively.

To validate the effectiveness of our proposed method, we compare with various baselines reviewed in Related work.

**Single Teacher KD** (Hinton, Vinyals, and Dean 2015) is applied to train student models from BERT12, RoBERTa12, ALBERT12 and XLNet12 on task specific training data to get four candidate teacher models, where the subscript 12 means 12 layers of transformers. Similar to Patient KD (Sun et al. 2019), we consider the BERT models with 3 layers and 6 layers of transformers as two student models, denoted by BERT3 and BERT6, respectively.

**U-Ensemble Teacher** is our implementation of the equal weight method by Yang et al. (2020). Every teacher model is assigned an equal weight in KD, and the student model learns from an aggregated distribution by averaging the outputs of all teacher models.

**Rand-Single-Ensemble Teacher** uses the strategy of Fukuda et al. (2017) to randomly select one teacher from the teacher model candidates at mini-batch level to provide soft-targets for training student model.

**W-Ensemble Teacher** is a weighted assemble method following Chebotar and Waters (2016); Fukuda et al. (2017); Wu, Chiu, and Wu (2019). We assign a different weight to each teacher model. The weights are fixed during the whole distillation.

Besides these existing baseline methods, we further propose two strong baselines for comparison. **LR-Ensemble Teacher** uses Logistic Regression (LR) model to model the best weights for each teacher candidate, instead of setting weights using heuristics like U-Ensemble/W-Ensemble. Specifically, let the aggregated distribution $P(y_i = c|x_i) = \sum_k w_k P_k(y_i = c|x_i; \Theta^k)$, where $w_k$ by maximizing the performance of the weighted ensemble teacher model. Since the training of the best weights can be conducted on either the training set or the development set, we denote by LR-Train-Ensemble and LR-Dev-Ensemble the corresponding LR models, respectively.

All the above baseline methods either use a single teacher model or assign fixed weights to ensemble multiple teacher models. As the last baseline, **Best-Single-Ensemble Teacher** considers an instance-level teacher model selection method, where for each instance in the training set (where ground-truth is available), the best performing single teacher model, that is, the one achieving the lowest cross entropy loss, is selected to train student model. We can treat this model as the upper bound of selecting best teacher model. Please note that this model cannot be evaluated on any test set since we are not supposed to use the ground-truth labels there for tuning models.

### Implementation Details

The training code is built on top of the code repository of Patient KD (Sun et al. 2019). All tasks in our experiments can be treated as classification problems where the input is either a single sentence or a pair sentences. For the tasks whose inputs are individual sentences, the model input has the form of [CLS] sentence1 [SEP]. For tasks whose input is sentence pairs, the input form is [CLS] sentence1 [SEP] sentence2 [SEP].

To fine-tune the teacher models, we adopt the open-sourced pre-trained weights for BERT12, RoBERTa12, ALBERT12 and XLNet12 as initialization.

The learning rate is set to $\{1e-5, 2e-5, 5e-5\}$. The batch size is set to 32. The maximum sequence length is set to 128. The number of epochs is set to 4. The best model is selected according to the accuracies on the development set.

The student models, BERT3 and BERT6, are initialized by the bottom 3 and 6 layers of BERT-Base, respectively. Meanwhile, we set the batch size to 32, the number of epochs to 4, the maximum length of sequence to 128, the learning rate to $\{1e-5, 2e-5, 5e-5\}$, the distillation temperature $T$ to $\{5, 10, 20\}$, and the loss equilibrium coefficient $\gamma$ to $\{0.2, 0.5, 0.7\}$. We choose the best model based on the performance on the development set. The $\gamma$ in the experiments ranges from $\{0.3, 0.5, 0.7, 0.9\}$, which is selected based on development set performance.

For the teacher model selector, our policy function is a simple logistic regression model. After feeding the input sequence encoded by the original pre-trained BERT-Base model into the teacher model selector, we adopt a standard

<table>
<thead>
<tr>
<th>Dataset</th>
<th>#Train</th>
<th>#Dev</th>
<th>#Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>RTE</td>
<td>2,490</td>
<td>277</td>
<td>3,000</td>
</tr>
<tr>
<td>MRPC</td>
<td>3,668</td>
<td>408</td>
<td>1,725</td>
</tr>
<tr>
<td>SST-2</td>
<td>67,349</td>
<td>872</td>
<td>1,821</td>
</tr>
<tr>
<td>QNLI</td>
<td>104,743</td>
<td>5,463</td>
<td>5,463</td>
</tr>
<tr>
<td>MNLI-mm</td>
<td>392,702</td>
<td>9,832</td>
<td>9,847</td>
</tr>
<tr>
<td>MNLI-m</td>
<td>392,702</td>
<td>9,815</td>
<td>9,796</td>
</tr>
<tr>
<td>QQP</td>
<td>363,849</td>
<td>40,430</td>
<td>390,965</td>
</tr>
</tbody>
</table>

Table 2: Statistics of the datasets for experiment.
Monte-Carlo based policy gradient method (Williams 1992) to optimize the parameters of the policy model.

**KD Pretraining** Our KD model is initialized using the pre-trained BERT model weights. Then, we use the distillation task in question to pre-train the KD model to learn from the average ensemble teacher model (i.e., average of all teacher model predictions). We set the batch size to 32 and max epochs to 4. The other hyper-parameters are kept the same as the normal KD training mentioned above.

**TS Pretraining** Pretraining is recommended by many reinforcement learning methods (Bahdanau et al. 2016; Feng et al. 2018). For TS pretraining, to avoid the instability of initial student model and speed up the training process, we leverage the performance of the average ensemble of the selected teacher models as the reward to train the TS Model.

**Iterative Training** After the pretraining stage, the KD and TS models are jointly trained alternatively in a batch-wise way. At batch #1, KD is trained while keeping TS fixed. At batch #2, TS is trained while keeping KD fixed, so on and so forth. If TS selects no teacher, this instance will not receive any reward from KD.

### Experimental Results

#### Results on Teacher Models

We start with the performance of individual teacher models and ensemble teacher models shown in Table 3. *Individual teacher models perform differently in different tasks.* No one single model wins on all tasks. RoBERTa achieves better results than the other models in three tasks (MNLI-mm, MNLI-m, QNLI), ALBERT shows higher accuracies on tasks MRPC and RTE, while XLNet scores the highest on SST-2. This suggests that different models may learn different local optimum and bear different biases.

Moreover, *ensemble of teacher models can lead to better performance.* All ensemble teacher models achieve better results than a single teacher model. This verifies that model ensemble is effective to mitigate the biases in various teacher models. By comparing different ensemble strategies, the weighted ensemble methods including W-Ensemble and LR-Ensembles outperform U-Ensemble. It indicates that smartly assigning weights to different models can lead to better results. Among all the ensemble models, LR-Dev-Ensemble achieves the best performance. Intuitively, training the weights based on the development set may help this model gain better generalization capability. Therefore, in the next experiments where we compare student models, we pick this best teacher model as the strongest baseline to represent ensemble methods.

#### Results on Student Models

The performance of student model is shown in Table 4. *A stronger teacher model may not necessarily lead to a better student.* Table 3 shows that ensemble teacher models are better than single teacher models. However, the corresponding student models distilled from those stronger teacher models are not always stronger. For example, the BERT3 student model distilled from ALBERT12 is better than that from LR-Dev-Ensemble on MNLI-mm, MNLI-m and SST-2. The BERT6 student model distilled from XLNET12 is also better than or on par with that from LR-Dev-Ensemble on MRPC and SST-2.

Even in the extreme case where Best-Single-Ensemble is used as the teacher model and we always choose the best performing teacher whose output is the closest to the ground-truth label, the corresponding student model does not always perform the best. Take BERT6 as example. The student model from the stronger teacher, Best-Single-Ensemble, is inferior to that from LR-Dev-Ensemble on almost all data sets. This observation indeed motivates our RL-KD method, that is, the best teacher model may not necessarily performs the best in distillation. Otherwise, we just use the ground truth label as reward.

### Table 3: Performance of various teacher models on test sets, including both individual models and ensemble models by various ensemble strategies.

<table>
<thead>
<tr>
<th>Teacher Model</th>
<th>QQP</th>
<th>MRPC</th>
<th>MNLI-mm</th>
<th>RTE</th>
<th>MNLI-m</th>
<th>QNLI</th>
<th>SST-2</th>
<th>AVG.</th>
</tr>
</thead>
<tbody>
<tr>
<td>BERT12</td>
<td>90.9</td>
<td>83.3</td>
<td>83.8</td>
<td>63.9</td>
<td>83.6</td>
<td>91.3</td>
<td>92.1</td>
<td>84.1</td>
</tr>
<tr>
<td>RoBERTa12</td>
<td>91.3</td>
<td>88.5</td>
<td>86.8</td>
<td>70.8</td>
<td>86.5</td>
<td>91.9</td>
<td>93.7</td>
<td>87.1</td>
</tr>
<tr>
<td>XLNet12</td>
<td>91.0</td>
<td>87.5</td>
<td>85.8</td>
<td>71.8</td>
<td>85.3</td>
<td>91.5</td>
<td>93.8</td>
<td>86.7</td>
</tr>
<tr>
<td>ALBERT12</td>
<td>90.6</td>
<td>88.7</td>
<td>84.3</td>
<td>74.0</td>
<td>83.9</td>
<td>91.6</td>
<td>91.2</td>
<td>86.3</td>
</tr>
<tr>
<td>U-Ensemble</td>
<td>92.3</td>
<td>89.2</td>
<td>87.5</td>
<td>72.6</td>
<td>87.3</td>
<td>93.0</td>
<td>93.7</td>
<td>87.9</td>
</tr>
<tr>
<td>W-Ensemble</td>
<td>92.3</td>
<td>89.5</td>
<td>87.5</td>
<td>72.9</td>
<td>87.4</td>
<td>93.0</td>
<td>93.6</td>
<td>88.0</td>
</tr>
<tr>
<td>Rand-Single-Ensemble</td>
<td>90.7</td>
<td>88.5</td>
<td>85.4</td>
<td>72.6</td>
<td>84.4</td>
<td>91.3</td>
<td>92.7</td>
<td>86.5</td>
</tr>
<tr>
<td>LR-Train-Ensemble</td>
<td>92.1</td>
<td>89.5</td>
<td>86.7</td>
<td>71.8</td>
<td>86.5</td>
<td>92.5</td>
<td>92.9</td>
<td>87.4</td>
</tr>
<tr>
<td>LR-Dev-Ensemble</td>
<td>92.3</td>
<td>90.0</td>
<td>87.3</td>
<td>74.4</td>
<td>87.1</td>
<td>93.0</td>
<td>94.0</td>
<td>88.3</td>
</tr>
</tbody>
</table>

**Table 3**
Table 4: Student model performance on test set. We compare three training strategies for student models: (1) FT: The student model is directly fine-tuned on task label data without any soft target labels from teacher models. (2) V-KD: vanilla KD from single teacher, teacher ensemble, or the best single teacher for each instance. (3) RL-KD: our RL method to select from multiple teacher models at instance level. For RL-KD, we consider all single model teachers (BERT12, Robert12, XLNet12, ALBERT12). For our method, we compare three different reward functions: (1) $reward_1$: we use the minus of ground-truth loss (CE) of student model as the reward function for teacher model selection. (2) $reward_2$: Besides the minus of ground-truth loss (CE), we also introduce the minus of knowledge distillation loss (DL) into the reward function. (3) $reward_3$: Besides $reward_2$, we also take the Accuracy metric of student model on development set into account for better generalization.

Table 5: Mean and Standard deviation values and statistically significance T-test (p-Value) across five different runs on RTE and MRPC. 1 and 2 denote statistically significant improvements over W-Ensemble and LR-Ensemble.

Table of contents:

- As a comparison, for student model BERT
- Table 2. For a weak student model such as BERT
- the policy for teacher model selection may be compromised. As a comparison, for student model BERT i, the model generalization ability is stronger. Therefore, even a small size of data can already yield a reasonable model, which in turn provides reliable reward as the feedback for policy training. That explains why student model BERTi achieves better performance than the other KD methods even on relatively small data sets.

Variance Analysis

We conduct 5 runs of models training and calculate the mean and standard deviation (Stdev) values. Besides, we also conduct a two-sided statistically significant t-test (p-Value) with threshold 0.05 comparing baseline methods with our RL-KD method. The experimental results are listed in Table 5. Results shows that (1) the variance of our approach is similar to baseline KD. (2) our method outperforms baselines with statistical significance.

Conclusions

In this paper, we tackle the problem of teacher model selection in knowledge distillation when multiple teacher models are available. We propose a novel RL-based approach, which dynamically assigns weights to teacher models at instance level to better adapt to the strengths of teacher models as well as the capability of student models. The extensive experiments on several NLP tasks verify the effectiveness of our proposed approach.
Acknowledgments
Jian Pei’s research is supported in part by the NSERC Discovery Grant program. All opinions, findings, conclusions and recommendations in this paper are those of the authors and do not necessarily reflect the views of the funding agencies.

References


