Multi-Grained Knowledge Distillation for Named Entity Recognition

Xuan Zhou [1], Xiao Zhang[1], Chenyang Tao[2], Junya Chen[2], Bing Xu[1], Wei Wang[1], and Jing Xiao[1]

[1]Ping An Technology (Shenzhen) Co., Ltd
[2]Duke University

{zhouxuan553,zhangxiao585,xubing962,wangwei192,xiaojing661}@pingan.com.cn
{chenyang.tao,junya.chen}@duke.edu

Abstract

Although pre-trained big models (*e.g.*, BERT, ERNIE, XLNet, GPT3 etc.) have delivered top performance in Seq2seq modeling, their deployments in real-world applications are often hindered by the excessive computations and memory demand involved. For many applications, including named entity recognition (NER), matching the state-of-the-art result under budget has attracted considerable attention. Drawing power from the recent advance in knowledge distillation (KD), this work presents a novel distillation scheme to efficiently transfer the knowledge learned from big models to their more affordable counterpart. Our solution highlights the construction of *surrogate labels* through the k-best Viterbi algorithm to distill knowledge from the teacher model. To maximally assimilate knowledge into the student model, we propose a multi-grained distillation scheme, which integrates cross entropy involved in conditional random field (CRF) and fuzzy learning. To validate the effectiveness of our proposal, we conducted a comprehensive evaluation on five NER benchmarks, reporting cross-the-board performance gains relative to competing prior-arts. We further discuss ablation results to dissect our gains.

1 Introduction

The task of named entity recognition (NER) aims to put named entity mentioned in a sentence into some pre-defined categories such as the person names, organizations, locations, etc. NER is a fundamental task in nature language processing (NLP), and often serves as an information extraction tool embedded in solutions for downstream tasks such as event recognition and aspect-level sentiment analysis. And therefore, the computational and memory efficiency of an NER model is often considered crucial in many empirical settings.

Given its practical significance, considerable research effort has been devoted to NER in recent years. One fruitful direction is to boost NER performance through the use of more sophisticated model architectures, such as Transformer and its variants (*e.g.*, BERT (Devlin et al., 2019), ERNIE (Sun et al., 2019c), XLNet (Yang et al., 2019), RoBERTa (Liu et al., 2019b), GPT3 (Brown et al., 2020), etc.). Many of these models are based on the attention mechanism, which allows the models to adaptively focus on different parts of the sentence based on its current understanding. This enables more accurate parsing of the context, which is critical for the NER task. While such advanced models have delivered substantial improvements, a major drawback is that they are typically computationally expensive and memory intensive, preventing their applications in many cost-sensitive settings.

As such, it is often desirable to reduce those big models into more affordable counterparts, preferably without any significant performance drop. One strategy is to compress or truncate the original model, examples in this category include parameter pruning (Srinivas and Babu, 2015; McCarley, 2019), low-rank approximation (Yu et al., 2017; Ma et al., 2019) and parameter quantization (Gong et al., 2014; Wu et al., 2016). The resulting model has a similar architecture to the original model, and consequently may suffer from similar limitations of the original model. Alternatively, knowledge distillation (KD) considers transferring knowledge into models with heterogeneous architectures (Hinton et al., 2015), which allows more flexibility in the control of resource usage. In KD, a new learner, often dubbed the student model, assimilates knowledge from a pre-trained model, commonly known as the teacher. For classification tasks, this is typically achieved via minimizing their discrepancy in the output, regardless of the internal model architectures. Teacher outputs, sometimes referred to as the soft-labels, typically encode more information than what a student might receive from the

Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 5704–5716
June 6–11, 2021.

raw training feature-label pairs, thus resulting the improved learning efficiency.

However, for sequence tasks such as NER, standard techniques from KD do not readily apply. While common KD seeks to minimize the KL divergence between the output label distributions, for NER, the number of label combinations grows exponentially wrt the sequence length. Extracting teacher knowledge as if each combination is a different label category would be largely inefficient, if possible at all. On the other hand, many SOTA NER models are built on conditional random fields (CRF) to incorporate the label dependencies, and it only offers an un-normalized likelihood that is not directly amenable to the computation of KL-divergence. Another challenge for training an NER model is the lack of labeled sequences for training, while the unlabeled sequences may be plentiful. The promise to leverage unlabeled data to improve NER accuracy is appealing.

To address the above challenges, in this paper, we present an efficient knowledge distillation scheme that trains a light model (*e.g.*, BiLSTM) which is able to retain the accuracy of its heavier counterparts, such as BERT, while significantly reduce cost. Our solution exploits the "soft surrogates", *i.e.*, the most probable label sequences under the teacher model, to inform the student learner. To efficiently identify the most likely label sequences and determine their relative likelihood, we explore the use of Viterbi algorithm to expedite computation. We also explore the use of unlabeled data to improve performance.

In summary, we highlight the following contributions in this study: (i) We present a novel multi-grained knowledge distillation strategy for sequence labeling via efficiently selecting k-best label sequence using Viterbi algorithm; (ii) We advocate the use of a complete cross entropy loss and fuzzy distillation loss to respectively account for probability mass of un-selected sequences and the uncertainties in teacher confidence; (iii) We present a comprehensive empirical analysis to dissect the gains from each individual components. We also show our model delivers substantial improvement relative to competing solutions on a wide-range of real-world benchmarks to demonstrate its utility.

2 Background

Our work is inspired by three lines of research: sequence-level knowledge distillation, k-best Viterbi algorithm and fuzzy conditional random field (CRF). In the following we review the technical backgrounds that are needed for the construction of our model.

Sequence-level Knowledge Distillation. Knowledge distillation (KD) is originally developed for classification tasks (Hinton et al., 2015; Tang et al., 2019). When dealing with sequence outputs (*e.g.*, machine translation, sequence labeling), where each unique combination of the output sequence is treated as different category, then the standard distillation objective is no longer appropriate, if feasible at all. This is because the number of unique combinations for a length L sequence with T possible tags scale at T^L. To combat such exponential scaling, Kim and Rush (2016) investigated using beam search for teacher output to select k-best candidates for KD in neural machine translation (NMT), and Mun'im et al. (2019) utilized an similar technique for KD in Large Vocabulary Continuous Speech Recognition (LVSCR) tasks.

k-best Viterbi Decoding. Viterbi decoding is a dynamic programming technique to find a sequence with the highest score in an exponential-growth domain, with only linear complexity (Viterbi, 1967). Generalization has been proposed to extend its original scope to find the top-K sequences that are most probable, see (Huang and Chiang, 2005; Nielsen, 2011) for details. A summary of the algorithm can be found in the supplementary material. In the context of KD in sequence tasks, a trained teacher model assigns varying probability to all sequence combinations. Our motivation is that the k-best Viterbi can be repurposed to pick out the K-most probable label sequences predicted by the teacher model, which plays an analogous role to the *soft-labels* (Figure 1a), without incurring unmanageable computational overhead.

Conditional Random Field. CRF (Lafferty et al., 2001) is a classic and powerful energy-based model that is capable of capturing complex spatial or temporal dependency structures. In the context of NLP, it has been generally used as a refinement layer that accounts for correlations missed by the base NLP model, which typically brings in additional performance gains for various NLP tasks (Lafferty et al., 2001; Collobert et al., 2011a; Huang et al., 2015). Implementation-wise, CRF computes an energy given a candidate output $\mathbf{y}$ and a context $\mathbf{x}$ (*i.e.*, input sequence), followed by a *softmax* operator to obtain the conditional likelihood, *i.e.*,

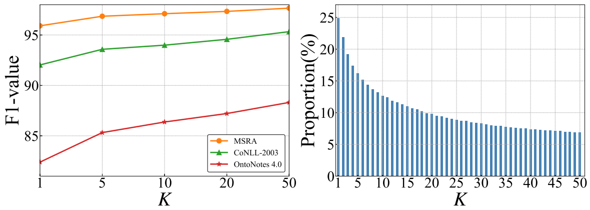

Figure 1: **Left.** The model prediction accuracy improves as one includes more candidate K. **Right.** The non-top-K probability remains non-neglectable even for moderately large K. Here, OntoNotes 4.0 is used for illustration, and Y-axis represents the proportion of samples, whose non-top-K cumulative probability is greater than 0.1.

$$\mathbb{P}(\mathbf{y}|\mathbf{x}) = \frac{e^{s(\mathbf{x},\mathbf{y})}}{\sum_{\tilde{\mathbf{y}} \in Y_{\text{all}}} e^{s(\mathbf{x},\tilde{\mathbf{y}})}}, \tag{1}$$

where Y_{all} is the set of all possible tag sequences, $s(\mathbf{x}, \mathbf{y})$ represents the *"compliance"* energy score between two sequences $\mathbf{x}$ and $\mathbf{y}$. In classical NLP models, $s(\mathbf{x}, \mathbf{y})$ is typically specified via handcrafted features and dependencies. In modern deep learning models, a typical decomposition of $s(\mathbf{x}, \mathbf{y})$ is given by two parts

$$s(\mathbf{x}, \mathbf{y}) = \sum_{j=1}^{L-1} \text{logit}_{j,\mathbf{y}_j}(\mathbf{x}) + A_{\mathbf{y}_j,\mathbf{y}_{j+1}}, \tag{2}$$

namely the *emission* and *transition* score. The transition matrix $A \in \mathbb{R}^{T \times T}$ characterizes the smoothness of the label sequence (probability of switching between consequent labels), and $\text{logit}_{j,k}(\mathbf{x})$ denotes the likelihood of seeing label k at position j as predicted by the base model. Note that we do not have to enumerate the exponentially many Y_{all} to compute the likelihood, which can be efficiently handled by the Viterbi algorithm introduced above in linear time (Collobert et al., 2011b).

Fuzzy CRF. It has been argued that properly adjusting the uncertainty of teacher prediction usually lends better knowledge transfer to the student, which can either be sharpening or relaxing the teacher predicted distributions. In standard KD, this is achieved by the incorporation of an annealing factor. In our setup, we consider relaxation via generalizing CRF to a candidate set of label sequences (Shang et al., 2018) rather than individual ones. Formally, we define the fuzzy loss as

$$\mathbb{P}(Y_{\text{candidate}}|\mathbf{x}) = \frac{\sum_{\mathbf{y}' \in Y_{\text{candidate}}} e^{s(\mathbf{x},\mathbf{y}')}}{\sum_{\tilde{\mathbf{y}} \in Y_{\text{all}}} e^{s(\mathbf{x},\tilde{\mathbf{y}})}}, \tag{3}$$

where $Y_{\text{candidate}}$ contains candidate label sequences. Here, we will use the k-best teacher predicted label sequences as the candidate set.

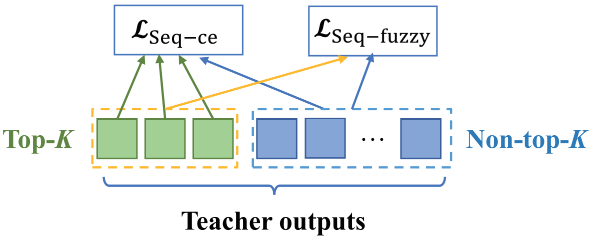
$\mathcal{L}_{\text{Seq-ce}}$
$\mathcal{L}_{\text{Seq-fuzzy}}$
Top-K
Non-top-K
···
Teacher outputs

Figure 2: k-best cross entropy loss uses predicted weights for each candidate label sequence, while k-best fuzzy loss group the candidates together and use their aggregated weight. The two losses represent fine-grained and coarse-grained learning from teacher respectively. Both schemes lump weights for the non-top-K labels.

3 Multi-Grained Distillation

In this section we detail the construction of our distillation scheme, with the overall architecture outlined in Figure 3.

3.1 Viterbi Distillation for Sequence Outputs

Now we are ready to present details of our main contribution, Viterbi distillation for sequence outputs. Our basic idea is to extract information from the teacher model via drawing a set of most probable sequences, together with the respective confidence to those sequences. Then these sequences are presented to the student model during its training, to pass on the knowledge from teacher through various loss functions.

More specifically, we apply the k-best Viterbi algorithm (see Algorithm 1 in the supplementary material) to get the pairs $\{(\mathbf{y}_1^{(i)}, p_1^{(i,\mathbf{t})}), \cdots, (\mathbf{y}_K^{(i)}, p_K^{(i,\mathbf{t})})\}$ for sample $\mathbf{x}^{(i)}$, where $\mathbf{y}_j^{(i)}$ is the j-th most-likely label sequence, and $p_j^{(i,\mathbf{t})}$ is the corresponding probability under the **teacher** model (indicated by the superscript $\mathbf{t}$). Similarly, in our subsequent discussions we denote $p_j^{(i,\mathbf{s})}$ as the probability produced by the **student** model on the label sequence $\mathbf{y}_j^{(i)}$.

3.1.1 Fine-grained k-best Cross Entropy Distillation

We proposed to approximate the complete cross entropy with the following k-best Vertibi approximation to avoid the exponential level of computational

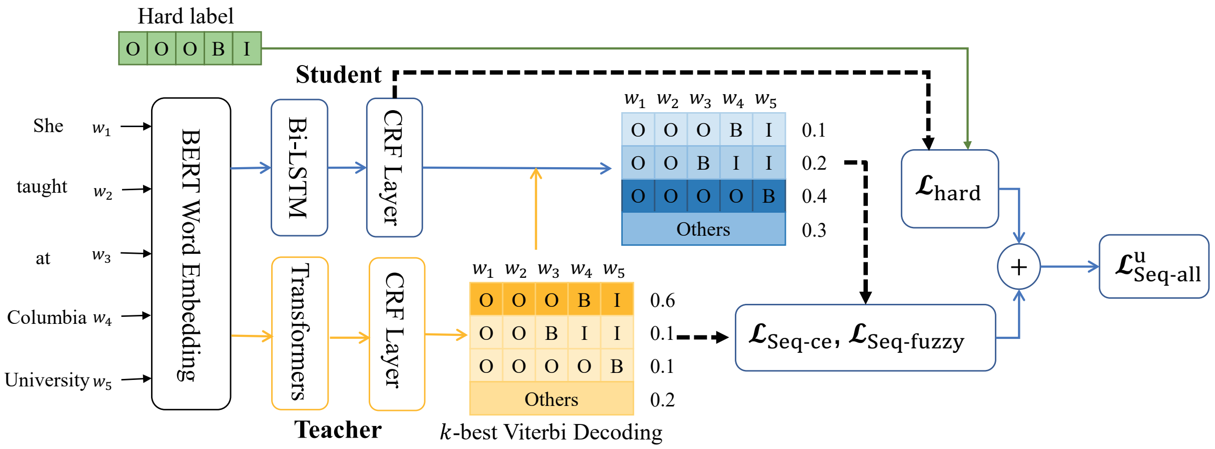

Figure 3: Model schematic for multi-grained knowledge distillation.

complexity:

$$\begin{aligned}
&\mathcal{L}_{\text{Seq-ce}}(\mathbf{x}^{(i)}) \\
&:= -\sum_{\tilde{\mathbf{y}}\in Y_{\text{all}}} \mathbb{P}^{\mathbf{t}}(\tilde{\mathbf{y}}|\mathbf{x}^{(i)}) \log\left(\mathbb{P}^{\mathbf{s}}(\tilde{\mathbf{y}}|\mathbf{x}^{(i)})\right) \\
&\approx -\sum_{j=1}^{K} p_j^{(i,\mathbf{t})} \log\left(p_j^{(i,\mathbf{s})}(\mathbf{y}_j^{(i)}|\mathbf{x}^{(i)})\right) \\
&\quad - \left(1-p^{\mathbf{t}}_{\text{sum}}\right)\log\left(1-p^{\mathbf{s}}_{\text{sum}}\right),
\end{aligned} \tag{4}$$

where $\mathbb{P}^{\mathbf{t}}(\cdot|\cdot)$ and $\mathbb{P}^{\mathbf{s}}(\cdot|\cdot)$ are the CRF likelihoods respectively for teacher and student, and we collect the probability mass of the top-K sequences into terms $p^{\mathbf{t}}_{\text{sum}}$ and $p^{\mathbf{s}}_{\text{sum}}$, *i.e.*,

$$p^{\mathbf{t}}_{\text{sum}} = \sum_{j=1}^{K} p_j^{(i,\mathbf{t})}, \quad p^{\mathbf{s}}_{\text{sum}} = \sum_{j=1}^{K} p_j^{(i,\mathbf{s})}. \tag{5}$$

Note that we have suppressed the dependency on $\mathbf{x}$ for notational clarity. We emphasize the residual probability mass $(1-p^{\mathbf{t}}_{\text{sum}}) \gg 0$ (see Figure 1b), and therefore should not be excluded from the loss as in Kim and Rush (2016); Wang et al. (2020).

3.1.2 Coarse-grained Fuzzy Distillation

To more appropriately account for the uncertainty in the teacher guidance, we advocate the use of a fuzzy objective that does not discriminate between the likelihood among the top-K picks. Concretely, this is given by the binary cross entropy in terms of $p^{\mathbf{t}}_{\text{sum}}$ and $p^{\mathbf{s}}_{\text{sum}}$,

$$\begin{aligned}
\mathcal{L}_{\text{Seq-fuzzy}}(\mathbf{x}^{(i)}) &:= -p^{\mathbf{t}}_{\text{sum}}\log(p^{\mathbf{s}}_{\text{sum}}) \\
&\quad - \left(1-p^{\mathbf{t}}_{\text{sum}}\right)\log\left(1-p^{\mathbf{s}}_{\text{sum}}\right).
\end{aligned} \tag{6}$$

Multi-grained Distillation. See Figure 2 for a graphical illustration of cross entropy and fuzzy objectives, which shows different granularity on learning from teacher.

3.1.3 Integrating Distillation Objectives

In addition to the "pure" distillation operations, we also allow direct learning from ground-truth labels and surrogate ones labeled by the teacher model (details found in Section 3.2). These two kinds of targets are named hard labels, and the related loss is as follows,

$$\mathcal{L}_{\text{hard}}(\mathbf{x}^{(i)}) := -\log \mathbb{P}(\mathbf{y}_{\text{hard}}|\mathbf{x}^{(i)}). \tag{7}$$

Thus, the final loss is the weighted sum of three terms defined above,

$$\begin{aligned}
\mathcal{L}_{\text{Seq-all}} = \tfrac{1}{N}\textstyle\sum_{i=1}^{N} \lambda_1 \mathcal{L}_{\text{hard}}(\mathbf{x}^{(i)}) \\
+\lambda_2 \mathcal{L}_{\text{Seq-fuzzy}}(\mathbf{x}^{(i)}) + \lambda_3 \mathcal{L}_{\text{Seq-ce}}(\mathbf{x}^{(i)}),
\end{aligned} \tag{8}$$

where $\lambda_i \geq 0, i = 1, 2, 3$.

Automated Tuning of Loss Weights. Tuning the loss weights $\{\lambda_i\}_{i=1}^{3}$ in (8) via standard techniques such as grid search can be laborious and costly. In this work, we leverage the *uncertainty weighting* strategy proposed in (Cipolla et al., 2018) to automate the weight selection procedure to balance multi-task objectives, both for its simplicity and robust performance. More specifically, we add uncertainty regularization terms for the weights, resulting

$$\mathcal{L}^{\text{u}}_{\text{Seq-all}} = \mathcal{L}_{\text{Seq-all}} - \frac{1}{2}\left(\log\lambda_1 + \log\lambda_2 + \log\lambda_3\right), \tag{9}$$

as our new learning objective.

3.2 Data Augmentation and Misc

Augmenting with Unlabeled Data. NER applications are often challenged with the lack of labeled instances for training. Motivated by the observations from prior studies that data augmentation using unlabeled data may improve distillation performance (Mukherjee and Awadallah, 2019; Tang et al., 2019), we also feed unlabeled data to the teacher model to construct additional surrogate label sequences using k-best Viterbi for distillation. We pool the surrogate sequences from both labeled and unlabeled data for cross entropy and fuzzy loss. Besides, we use Top-1 sequence as hard label to educate the student.

Token-level Distillation. To verify the necessity of using a sequence-level distillation, we additionally consider token-level distillation with standard KD techniques. Specifically, we consider the following loss

$$\mathcal{L}_{\text{Token}} := -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{T}\mathcal{L}_{i,j}, \tag{10}$$

where $\mathcal{L}_{i,j} = \sum_{k=1}^{L}\mathbb{P}^{\mathbf{t}}(\mathbf{y}_j = k|\mathbf{x}^{(i)})\log(\mathbb{P}^{\mathbf{s}}(\mathbf{y}_j = k|\mathbf{x}^{(i)}))$ is the token-level KL-divergence between the teacher's and student's predicted distributions for each token, rather than the token sequence distribution. This breaks the dependencies between tokens to avoid the combinatory complexity, thus enabling standard KD. The probability $\mathbb{P}(\mathbf{y}_j = k|\mathbf{x}^{(i)})$ can be simply calculated by *emission* score, in this case, the loss is denoted as $\mathcal{L}_{\text{Token-em}}$. To recover the spatial dependencies, we can add *transition* matrix to derive the probability $\mathbb{P}(\mathbf{y}_j = k|\mathbf{x}^{(i)})$ by calculating the marginal distribution of token from the probability distribution $\mathbb{P}(\mathbf{y}|\mathbf{x}^{(i)})$ given by CRF, similar to Wang et al. (2020). We denote this corresponding loss as $\mathcal{L}_{\text{Token-pos}}$.

4 Related Work

NER without BERT. Extensive investigations in NER have been conducted without appealing to the BERT architecture, and instead using simpler architectures such as BiLSTM or convolutional neural network (CNN) (Huang et al., 2015; Strubell et al., 2017a). Special effort has been made to adapt the NER network architectures for better handling of lexicons. Liu et al. (2019a) leveraged hybrid semi-Markov CRFs to improve NER recognition with Gazetteers, where segments are used as the basic units instead of words. In Zhang and Yang (2018) a lattice-structured LSTM is proposed to encode a sequence of input characters as well as all potential words that match a lexicon. Gui et al. (2019) incorporates lexicons using a rethinking mechanism under a CNN setup, which renders faster inference compared with lattice-LSTM. Peng et al. (2019) explores weighted word embedding to match all probable words given the lexicons. In the method given by Ghaddar and Langlais (2018), a lexical representation is computed for each word with a 120-dimensional vector, where element encodes the similarity of the word with an entity type. Recently, Ding et al. (2019) leveraged the graph neural network (GNN) to exploit the additional rich information captured by the gazetteers, setting new SOTA performance. Our distillation work is orthogonal to these developments, and the techniques can be combined for further improvements.

Compressing BERT with Distillation. Various efforts have been made to reduce the size and cost of BERT or other big models while maximally maintaining their outstanding performance, and KD offers an appealing alternative to the direct amputation of the original models. Along this line, Tang et al. (2019) studied on compressing BERT to BiLSTM for classification, resulting a model with comparable performance to ELMo but 100× fewer parameters and 15× faster inference. Tsai et al. (2019) successfully applied KD for multilingual sequence labeling model, enabling SOTA results on an MiniBERT model afforded by a single CPU. Other developments include distillation on intermediate representations (Sun et al., 2019b) and student pre-training (Turc et al., 2019). The value of unlabeled data in KD has also been explored (Mukherjee and Awadallah, 2019). Our work complements these studies via presenting a framework of using top-K soft-surrogate labels for KD.

Distilling with top-K Picks. Parallel to our work, Tang et al. (2020) also considered a top-K scheme for KD. This work distincts from our proposal via assigning uniform weights to the un-selected labels sequences, a practice that can be largely inefficient. Another work close to our setup is Wang et al. (2020), where the authors tried to distill the structural knowledge from multiple monolingual teachers to a single student. In their loss, the probability mass of the non-top-K picks is discarded. In Figure 1 (right), we show that the probability for non-top-K picks is non-neglectable, which is properly accounted for in our proposed complete cross entropy loss.

5 Experiment

To validate the proposed solution and elaborate the gains, we benchmark it against state-of-the-art methods, through a wide range of experiments on real-world datasets. All experiments are implemented with Tensorflow and executed on a single NVIDIA P100 GPU. Details of the experimental setup are provided in supplementary material, due to space limits, and our code will be avialable at `https://github.com/11zhouxuan/multi_grained_kd_ner`.

5.1 Datasets

The following real-world datasets are considered in our study. Detailed summary statistics of the datasets can be found in supplementary material.

CoNLL-2003 NER (Sang and De Meulder, 2003) consists of newswire from Reuters RCV corpus. Unlabeled data from the Reuters RCV corpus is used for our data augmentation experiments.

OntoNotes (Consortium et al., 2011) is an annotated multilingual corpus consists of texts from a wide variety of sources, such as telephone conversation, broadcast and newswire. For our NER experiment, we consider

- **English NER** is derived from OntoNotes Release 5.0 and processed according to Pradhan et al. (2013).
- **Chinese NER** is derived from OntoNotes Release 4.0 and processed according to Che et al. (2013).

Text data from other OntoNotes tasks are used as unlabeled data.

MSRA (Levow, 2006) is a Chinese NER dataset with its corpus derived from news domain. We use the Chinese word segmentation dataset MSR (Levow, 2006) for unlabeled data.

Weibo (Peng and Dredze, 2015) is a Chinese NER dataset derived from social media contents. Only a small fraction of data is labeled in this dataset.

5.2 Model specification

We briefly describe the modeling and training specifications choices for the teacher and student models below.

Teacher Model. To build a strong learner, our teacher model is constructed by a BERT model followed by a CRF layer, and we denote it as BERT+CRF. Specifically, we use BERT$_{\text{BASE}}$[1] model as our feature encoder, which is known to perform strongly across a wide range of NLP tasks. A dropout layer is concatenated to the BERT, followed by a fully connected layer that computes the logit($\mathbf{x}$) for the labels at each location. We further apply an additional CRF-layer as defined in (2) to account for the temporal dependencies among the labels, similar to the work of Meng et al. (2019). The teacher model is trained using the standardized fine-tune paradigm (Devlin et al., 2019). Following Howard and Ruder (2018); Sun et al. (2019a), we set different learning rates for each layer. A larger rate is used for CRF, and for the BERT the learning rates decays by a factor of 0.9 as the layers approaches the input.

Student Model. For our student model, we want it to be light, fast yet still sufficiently expressive. To this end, we use the BiLSTM+CRF architecture proposed in Huang et al. (2015). This model exploits a Bidirectional LSTM to map input sequence $\mathbf{x}$ into a sequence of feature vectors, which accounts for the context from both directions. The rest of the construction follows what has been described for the teacher model, with the BERT part replaced by the BiLSTM. We reuse the learned word-embeddings from the teacher model and keep it frozen during training. Empirically, we find this strategy produces better results, possibly due to reduced effort transferring the knowledge of richer embedding representation compared to alternatives such as *word2vec*, and it also avoids over-fitting.

5.3 Baselines, Variations and Evaluation

In addition to the vanilla teacher and student models described above, we also considered the following strong established NER baselines in our experiments.

- **BERT teacher baselines (Chinese NER)** BERT+Glyce (Meng et al., 2019).
- **Non-BERT student baselines (English NER)** LSTM-CNNs (Chiu and Nichols, 2016), LEX (Ghaddar and Langlais, 2018), IDCNN (Strubell et al., 2017b), HSCRF (Liu et al., 2019a).
- **Non-BERT student baselines (Chinese NER)** SUL (Peng et al., 2019), Lattice-LSTM (Zhang and Yang, 2018), NMDM (Ding et al., 2019), ME-CNER (Xu et al., 2019).

Variations. To further understand the contributions from different components of our proposal, we run different variations of the model to dissect the gains. We use BiLSTM+CRF as our

[1] `https://github.com/google-research/bert`

Table 1: Comparison of F1 scores. ↑ denotes the gain relative to Vanilla BiLSTM+CRF baseline. Best results in each category are shown in bold. Results for the competing baselines are collected from the original papers.

Models		Datasets				
		English Datasets		Chinese Datasets		
English	Chinese	CoNLL-2003	OntoNotes 5.0	MSRA	Weibo	OntoNotes 4.0
BERT+CRF (teacher)	Vanilla BERT+CRF	NA	NA	94.80	67.33	79.16
	Glyce+BERT	NA	NA	95.54	67.60	80.62
	BERT+CRF (teacher)	**92.03**	**89.92**	**95.91**	**71.22**	**82.39**
LSTM-CNNs	Lattice-LSTM	90.91	86.17	93.18	58.79	73.88
IDCNN	NMDM	90.54	86.84	**94.4**	59.5	**76.0**
LEX	SUL	90.52	87.95	93.44	61.24	75.54
HSCRF	ME-CNER	**91.10**	**89.94**	91.45	**68.93**	NA
Vanilla BiLSTM+CRF		80.08	82.58	90.70	51.85	64.40
+ TE (no distillation)		88.35 ↑ 8.27	87.35 ↑ 4.77	91.69 ↑ 0.99	54.87 ↑ 3.02	69.78 ↑ 5.38
+ TE + TOKEN-EM		88.46 ↑ 8.38	87.39 ↑ 4.81	91.51 ↑ 0.81	53.55 ↑ 1.70	70.46 ↑ 6.06
+ TE + TOKEN-POS		88.66 ↑ 8.58	88.17 ↑ 5.59	91.77 ↑ 1.07	54.34 ↑ 2.49	71.19 ↑ 6.79
+ TE + SEQ		**89.54** ↑ 9.46	**88.34** ↑ 5.76	**91.98** ↑ 1.98	**57.14** ↑ 5.29	**72.33** ↑ 7.93
+ TE + DA (no distillation)		90.69 ↑ 10.61	87.52 ↑ 4.94	92.68 ↑ 1.96	69.74 ↑ 17.89	74.53 ↑ 10.13
+ TE + DA + TOKEN-EM		90.49 ↑ 10.41	88.05 ↑ 5.47	92.66 ↑ 1.96	69.83 ↑ 17.98	74.52 ↑ 10.12
+ TE + DA + TOKEN-POS		90.98 ↑ 10.90	88.29 ↑ 5.71	92.59 ↑ 1.89	69.90 ↑ 18.05	74.88 ↑ 10.48
+ TE + DA + SEQ (student)		**91.17** ↑ 11.09	**88.91** ↑ 6.32	**92.99** ↑ 2.29	**71.62** ↑ 19.77	**76.05** ↑ 11.65

Table 2: Performance with different loss combinations.

Loss Combinations	Datasets					
	OntoNotes 4.0			CoNLL-2003		
	P	R	F1	P	R	F1
$\mathcal{L}^{\mathrm{u}}_{\text{Seq-all}}$	77.96	74.23	**76.05**	**90.85**	91.50	**91.17**
+ $\mathcal{L}_{\text{Token-pos}}$	76.63	74.47	75.53	90.63	91.51	91.07
- $\mathcal{L}_{\text{Seq-fuzzy}}$	**78.11**	73.43	75.70	90.43	91.11	90.77
- $\mathcal{L}_{\text{Seq-ce}}$	75.71	**74.87**	75.29	90.35	**91.60**	90.97

Table 3: Comparison of automated uncertainty weighting (auto) defined in (9) versus equal weighting (equal).

Datasets	Without DA		With DA	
	auto	equal	auto	equal
MSRA	91.98	91.77	92.99	92.83
Weibo	57.14	55.59	71.62	70.09
OntoNotes 4.0	72.32	71.32	76.05	75.83
CoNLL-2003	89.54	89.26	91.17	91.09
OntoNote 5.0	88.34	88.27	88.91	88.42

base model, and consider variants with combinations of the following components in our experiments: (i) TE: use fixed pre-trained teacher embedding; (ii) TOKEN-EM: token-level distillation with loss $\mathcal{L}_{\text{Token-em}}$; ($iii$) TOKEN-POS: token-level distillation with with loss $\mathcal{L}_{\text{Token-pos}}$; ($iv$) SEQ: multi-grained sequence-level distillation with loss $\mathcal{L}^{\mathrm{u}}_{\text{Seq-all}}$; ($v$) DA: augmented with unlabeled data. We also vary the size of best candidate set from $K = 1$ to $K = 15$.

Evaluation. We report the F1 score following Sang and De Meulder (2003), and relegate other quantitative metrics such as *Precision* and *Recall* to the supplementary material. We apply early stopping with max patience set to 5 based on the performance of the development set.

5.4 Analysis of Results

From Table 1 we first find out that, in all three Chinese datasets, our teacher model outperforms two baselines (Meng et al., 2019). We owe this to the layer-wise learning rate and discriminative fine-tuning strategies (Howard and Ruder, 2018; Sun et al., 2019a). Another analysis in terms of teacher is that directly copying the teacher embedding to the student model can be most helpful, for both English and Chinese datasets.

Regarding distillation, it achieved cross-the-board performance gains relative to the no-distillation TE baseline. Our final proposal, namely TE+DA+SEQ, performs better or similarly to almost all of its non-BERT baselines. Note that these competing baselines have leveraged additional domain knowledge, such as lexicon information, to boost NER performance (see Related Work). Such practice not only adds specialized modeling effort and complexity, but also make the resulting architecture less generalizable to other tasks. Our results show that we can match the performance via reaping the knowledge from more sophisticated models using general purpose sequence-level distillation, rather than appealing to dedicated modeling effort.

We also observed that inducing data augmentation consistently improves student learning. And notably, in all cases, the sequence-level distillation outperforms token-level distillation, especially in the absence of data augmentation.

Table 4: Comparison of efficiency.

Models	No. of parameters	Inference time (s)
Student	2.6M	1.5×10^{-3}
Teacher	110.9M	7.0×10^{-2}

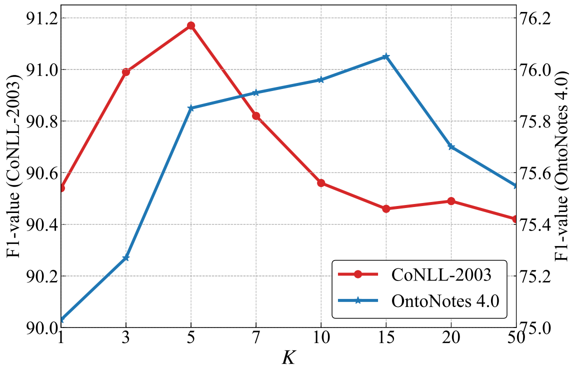

Figure 4: Performance with different candidate size K.

5.5 Ablation Study

Size of Candidate Set. In Figure 4, we compare how varying the size of candidate set affects performance. The performance peaks at a moderate K, after that the gain tapers off. This is because when using an excessive K, the distillation process starts to include more inaccurate label-sequences, compromising the learning efficiency (as top-K label-sequences are treated equally in the fuzzy objective).

Effect of Augmenting with Unlabeled Data. In Figure 5, we report the F1 scores with different number of unlabeled instances to augment the distillation phase for on Weibo dateset, which has very few training instances but an enormous unlabeled set. The use of data augmentation drastically boosts student's performance (F1: $57.14 \rightarrow 71.62$). The distillation performance monotonically increases as more unlabeled instances are used, and the performance closes, even beats the teacher model in the large sample limit.

Loss Combinations. We further explore additional combinations of losses to sharpen our understanding, with main results summarized in Table 2. Combinations of sequence-level and token-level loss (*i.e.*, $\mathcal{L}^{\mathrm{u}}_{\text{Seq-all}} + \mathcal{L}_{\text{Token-pos}}$) reveal a slight drop in performance. We also subtract each individual $\mathcal{L}_{\text{Seq-fuzzy}}$ and $\mathcal{L}_{\text{Seq-ce}}$ from $\mathcal{L}^{\mathrm{u}}_{\text{Seq-all}}$. It appears that the coarse-grained loss $\mathcal{L}_{\text{Seq-fuzzy}}$ is propitious to increase the recall value, while using the fine-grained loss $\mathcal{L}_{\text{Seq-ce}}$ will get better precision value.

Effect of Uncertainty Weighting. In this paper, we balance our loss functions with different scales by injecting uncertainty (Cipolla et al., 2018). This method yields a regularization term of weight, preventing the weight tends to zero. Table 3 shows that, regularized weighting outperforms equal weighting in all cases.

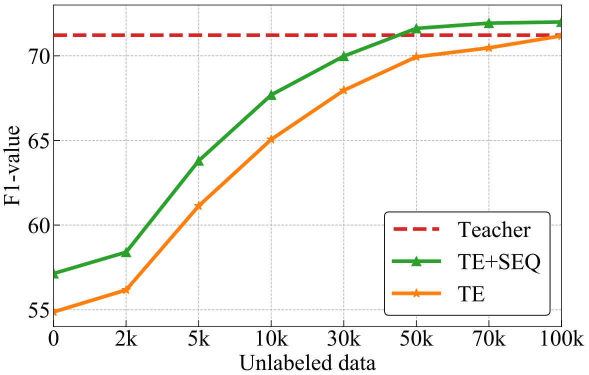

Figure 5: Performance gains using more unlabeled data on the Weibo dataset. Dashed line denotes the teacher performance baseline.

Computation Efficiency. To examine the efficiency gains, we compare inference time difference between the teacher and student model, as reported in Table 4. Our student model achieves over 40× reduction and speedup, while achieving comparable performance compared with the teacher model.

Alternative Choice of Student Model. We also exam the performance of multi-grained distillation method on another student model, that is four transformer layers, which has a similarity structure as BERT. The results are showed in supplementary material (Table 9).

6 Conclusions

In this work, we develop novel multi-grained knowledge distillation techniques to train a light NER model with comparable performance with their more sophisticated counterparts. In particular, we show that Viterbi algorithm can be exploited to impart the knowledge of k-best predictions from the teacher model to the student. We further advocated the use of CRF adjustments, fuzzy objective and data augmentation to improve performance. Our empirical experiments carefully analyze the gains from our proposal on a wide range of NER benchmarks, via efficiently transferring knowledge from a powerful BERT model to a much more compact BiLSTM student. In future work, we seek to extend the proposed framework to more general distillation applications where CRF is used, such as speech recognition, and distill with more general representation transfer schemes (Chen et al., 2020).

Acknowledgements

The authors would like to thank the anonymous reviewers for their insightful comments. The research conducted at Duke was supported in part by the DOE, NSF and ONR.

References

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*.

Wanxiang Che, Mengqiu Wang, Christopher D Manning, and Ting Liu. 2013. Named entity recognition with bilingual constraints. In *NAACL*.

Junya Chen, Zidi Xiu, Benjamin Goldstein, Ricardo Henao, Lawrence Carin, and Chenyang Tao. 2020. Supercharging imbalanced data learning with causal representation transfer. *arXiv preprint arXiv:2011.12454*.

Jason P.C. Chiu and Eric Nichols. 2016. Named entity recognition with bidirectional LSTM-CNNs. *Transactions of the Association for Computational Linguistics*, 4:357–370.

R. Cipolla, Y. Gal, and A. Kendall. 2018. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *CVPR*.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011a. Natural language processing (almost) from scratch. *Journal of machine learning research*, 12(ARTICLE):2493–2537.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011b. Natural language processing (almost) from scratch. *Journal of machine learning research*, 12(Aug):2493–2537.

Linguistic Data Consortium et al. 2011. Ontonotes release 4.0 [online] available from: https://catalog. ldc. upenn. edu. *LDC2011T03*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*.

Ruixue Ding, Pengjun Xie, Xiaoyan Zhang, Wei Lu, Linlin Li, and Luo Si. 2019. A neural multi-digraph model for chinese ner with gazetteers. In *ACL*.

Abbas Ghaddar and Phillippe Langlais. 2018. Robust lexical features for improved neural network named-entity recognition. In *COLING*.

Yunchao Gong, Liu Liu, Ming Yang, and Lubomir Bourdev. 2014. Compressing deep convolutional networks using vector quantization. *arXiv preprint arXiv:1412.6115*.

Tao Gui, Ruotian Ma, Qi Zhang, Lujun Zhao, Yu-Gang Jiang, and Xuanjing Huang. 2019. Cnn-based chinese ner with lexicon rethinking. In *IJCAI*.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *Computer Science*, 14(7):38–39.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *ACL*.

Liang Huang and David Chiang. 2005. Better k-best parsing. In *Proceedings of the Ninth International Workshop on Parsing Technology*.

Zhiheng Huang, Yi Chang, Bo Long, Jean-Francois Crespo, Anlei Dong, Sathiya Keerthi, and Su-Lin Wu. 2012. Iterative Viterbi A* algorithm for k-best sequential decoding. In *ACL*.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional LSTM-CRF models for sequence tagging. *CoRR*, abs/1508.01991.

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*.

Yoon Kim and Alexander M. Rush. 2016. Sequence-level knowledge distillation. In *EMNLP*.

John Lafferty, Andrew Mccallum, and Fernando Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML*.

Gina-Anne Levow. 2006. The third international chinese language processing bakeoff: Word segmentation and named entity recognition. In *Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing*.

Tianyu Liu, Jin-Ge Yao, and Chin-Yew Lin. 2019a. Towards improving neural named entity recognition with gazetteers. In *ACL*.

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

Xindian Ma, Peng Zhang, Shuai Zhang, Nan Duan, Yuexian Hou, Dawei Song, and Ming Zhou. 2019. A tensorized transformer for language modeling. *arXiv preprint arXiv:1906.09777*.

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

Yuxian Meng, Wei Wu, Fei Wang, Xiaoya Li, Ping Nie, Fan Yin, Muyu Li, Qinghong Han, Xiaofei Sun, and Jiwei Li. 2019. Glyce: Glyph-vectors for chinese character representations. In *NIPS*.

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

Raden Mu'az Mun'im, Nakamasa Inoue, and Koichi Shinoda. 2019. Sequence-level knowledge distillation for model compression of attention-based sequence-to-sequence speech recognition. In *ICASSP*.

Jesper Nielsen. 2011. A coarse-to-fine approach to computing the k-best viterbi paths. In *CPM*.

Minlong Peng, Ruotian Ma, Qi Zhang, and Xuanjing Huang. 2019. Simplify the usage of lexicon in chinese ner. *arXiv preprint arXiv:1908.05969*.

Nanyun Peng and Mark Dredze. 2015. Named entity recognition for chinese social media with jointly trained embeddings. In *EMNLP*.

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Hwee Tou Ng, Anders Björkelund, Olga Uryupina, Yuchen Zhang, and Zhi Zhong. 2013. Towards robust linguistic analysis using ontonotes. In *CoNLL*.

Erik F Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. *arXiv preprint cs/0306050*.

Jingbo Shang, Liyuan Liu, Xiaotao Gu, Xiang Ren, Teng Ren, and Jiawei Han. 2018. Learning named entity tagger using domain-specific dictionary. In *EMNLP*.

Suraj Srinivas and R. Venkatesh Babu. 2015. Data-free parameter pruning for deep neural networks. *Computer Science*, pages 2830–2838.

Emma Strubell, Patrick Verga, David Belanger, and Andrew McCallum. 2017a. Fast and accurate entity recognition with iterated dilated convolutions. In *EMNLP*.

Emma Strubell, Patrick Verga, David Belanger, and Andrew McCallum. 2017b. Fast and accurate entity recognition with iterated dilated convolutions. In *EMNLP*.

Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. 2019a. How to fine-tune bert for text classification? *arXiv preprint arXiv:1905.05583*.

Siqi Sun, Yu Cheng, Zhe Gan, and Jingjing Liu. 2019b. Patient knowledge distillation for bert model compression. *arXiv preprint arXiv:1908.09355*.

Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Xuyi Chen, Han Zhang, Xin Tian, Danxiang Zhu, Hao Tian, and Hua Wu. 2019c. Ernie: Enhanced representation through knowledge integration. *arXiv preprint arXiv:1904.09223*.

Jiaxi Tang, Rakesh Shivanna, Zhe Zhao, Dong Lin, Anima Singh, Ed H. Chi, and Sagar Jain. 2020. Understanding and Improving Knowledge Distillation. *arXiv:2002.03532 [cs, stat]*. ArXiv: 2002.03532.

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

Henry Tsai, Jason Riesa, Melvin Johnson, Naveen Arivazhagan, Xin Li, and Amelia Archer. 2019. Small and Practical BERT Models for Sequence Labeling. In *EMNLP-IJCNLP*.

Iulia Turc, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Well-read students learn better: On the importance of pre-training compact models. *arXiv preprint arXiv:1908.08962*.

Andrew Viterbi. 1967. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE transactions on Information Theory*, 13(2):260–269.

Xinyu Wang, Yong Jiang, Nguyen Bach, Tao Wang, Fei Huang, and Kewei Tu. 2020. Structure-level knowledge distillation for multilingual sequence labeling. In *ACL*.

Jiaxiang Wu, Cong Leng, Yuhang Wang, Qinghao Hu, and Jian Cheng. 2016. Quantized convolutional neural networks for mobile devices. In *CVPR*.

Canwen Xu, Feiyang Wang, Jialong Han, and Chenliang Li. 2019. Exploiting multiple embeddings for chinese named entity recognition. In *CIKM*.

Jie Yang and Yue Zhang. 2018. NCRF++: An open-source neural sequence labeling toolkit. In *ACL*.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *arXiv preprint arXiv:1906.08237*.

Xiyu Yu, Tongliang Liu, Xinchao Wang, and Dacheng Tao. 2017. On compressing deep models by low rank and sparse decomposition. In *CVPR*.

Yue Zhang and Jie Yang. 2018. Chinese NER using lattice LSTM. In *ACL*.

Appendix

A Datasets statistic

Detailed summary statistics of the datasets.

Table 5: Summary statistics for the datasets. #Ent: number of entities; S: sentence size; T: token size.

Language	Dataset	#Ent	Type	Train	Unlabeled	Dev	Test
English	OntoNotes 5.0	18	S	59.9k	36.0k	8.5k	8.3k
			T	1088.5k	676.6k	147.7k	152.7k
	CoNLL-2003	4	S	15.0k	50.0k	3.3k	3.5k
			T	204.6k	1485.4k	51.4k	46.4k
	SemEval-2016 Task5	1	S	1.8k	3.9k	0.2k	0.7k
			T	27.6k	78.8k	3.7k	12.6k
Chinese	MSRA	3	S	46.4k	25.5k	-	4.4k
			T	979.2k	1190.2k	-	172.6k
	Weibo	4	S	1.4k	50.0k	0.3k	0.3k
			T	73.5k	2118.9k	14.4k	14.8k
	OntoNotes 4.0	4	S	15.7k	21.3k	4.3k	4.3k
			T	491.9k	659.0k	200.5k	208.1k

B Hyperparameters

In Table 6 we list the search grids for our hyperparameter tuning. For each dataset we evaluate all combinations and report the test set results for best-performing model on the validation set.

Table 6: Hyperparameters for student and teacher models.

Hyperparameters	Teacher	Student
Batch size	32	{16,32,64,128}
Learning rate	Logit and BERT initial : le-5	{1e-3,5e-4}
	CRF: le-3	
Dropout rate	Embedding: 0.5	Embedding: {0.5,0.6}
	BERT output: 0.5	LSTM output: 0.5
LSTM hidden size	-	300

C k-best Viterbi Algorithm

We implemented the k-best Viterbi algorithm generalized from the classic Viterbi algorithm, that is to storage the Top-K rather than maximum scores at each time step and tag type. Though exists more efficient k-best Viterbi implementation (Huang et al., 2012) for large label scenarios, it only offers marginal efficiency gains as in our tasks. Unlike existing work (Yang and Zhang, 2018), we remove short examples where K is larger than the possible number of label sequences. In our application, we also output the probability rather than the path score, and this needs another dynamic programming to calculate the denominator in (1). The transition matrix is rule-constrained such that invalid transitions (*e.g.*, B-ORG→ I-PER) are prohibited.

D Results on student with transformer structure

Though our main result is reported as the BiLSTM student, the multi-grained distillation method proposed in this article is not restricted to this student model. We show herein the experiment results (Table 9) on the four transformer layers as student model.

Algorithm 1: Pseudocode of k-best Viterbi Algorithm

```
Input: The length of input sequence $\mathbf{x}^{(i)}$ as $L$, the
       number of tags $T$, transition matrix $A[T \times T]$,
       the $logits[L \times T]$ of $\mathbf{x}^{(i)}$, $K$
Output: $k$-best paths and corresponding probabilities
Initialize:CurrentPath = $1 : T$;
Initialize:CurrentScore = $logits[0, :]$;
for $m = 0$ to $L - 1$ do
    $w$ =CurrentPath.shape[1];
    Update CurrentPath by adding $T$ possible tags
      on the end of every path;
    Update CurrentScore  of each path through (2);
    if CurrentPath.shape[0] $< K \times T$ then
        Continue
    end
    Group sort $k$-best on CurrentScore;
    Update  CurrentPath using the indexes of $k$-best
      scores, resulting $[w + 1, K \times T]$;
    Update  CurrentScore using the indexes of
      $k$-best scores, resulting $[K, 1]$;
end
Calculate the denominator in (1) using another
dynamic programming, for details see (Collobert
et al., 2011b);
return  $k$-best paths and $k$-best probabilities or
  $\{(\mathbf{y}_1^{(i)}, p_1^{(i,\mathbf{t})}), \cdots, (\mathbf{y}_K^{(i)}, p_K^{(i,\mathbf{t})})\}$,
```

E Results on Sentiment Analysis Task

Table 7 shows the experimental results on dataset SemEval-2016 Task 5 of aspect based sentiment analysis. In details, we choose subtask 1: Aspect term extraction of the restaurants domain, and split 10% of the training data as the development set. Texts from datasets SemEval-2014 and SemEval-2015 are used as unlabeled data.

Table 7: Results on SemEval-2016 Task 5.

Model	P	R	F1
BERT+CRF (teacher)	80.42	79.90	80.16
Vanilla BiLSTM+CRF	76.70	65.11	70.43
+ TE	74.86	66.08	71.20
+ TE + TOKEN-EM	73.69	68.49	71.12
+ TE + TOKEN-POS	76.33	64.79	70.09
+ TE + SEQ	77.02	67.36	71.87
+ TE + DA	77.08	67.04	71.71
+ TE + DA + TOKEN-EM	68.04	74.60	71.17
+ TE + DA + TOKEN-POS	69.59	72.83	71.17
+ TE + DA + SEQ (student)	75.26	70.90	73.01

F Experiment Results Details

This section contains detail experimental results for precision, recall and F1-value, see Table 8, 10,11, 12,13.

Table 8: Results on MSRA.

Model	P	R	F1
BERT+CRF (teacher)	95.96	95.86	95.91
Vanilla BiLSTM+CRF	91.57	89.84	90.70
+ TE	93.06	90.88	91.96
+ TE + TOKEN-EM	92.37	90.67	91.51
+ TE + TOKEN-POS	93.23	90.36	91.77
+ TE + SEQ	93.26	90.73	91.98
+ TE + DA	93.77	91.61	92.68
+ TE + DA + TOKEN-EM	93.77	91.57	92.66
+ TE + DA + TOKEN-POS	93.61	91.59	92.59
+ TE + DA + SEQ (student)	93.90	92.11	92.99

Table 10: Results on Weibo.

Model	P	R	F1
BERT+CRF (teacher)	74.30	68.38	71.22
Vanilla BiLSTM+CRF	64.08	43.54	51.85
+ TE	65.67	47.13	54.87
+ TE + TOKEN-EM	60.79	47.85	53.55
+ TE + TOKEN-POS	67.02	45.69	54.34
+ TE + SEQ	66.90	49.87	57.14
+ TE + DA	72.73	66.99	69.74
+ TE + DA + TOKEN-EM	71.04	68.66	69.83
+ TE + DA + TOKEN-POS	72.80	67.22	69.90
+ TE + DA + SEQ (student)	74.29	69.14	71.62

Table 11: Results on OntoNotes 4.0.

Model	P	R	F1
BERT+CRF (teacher)	81.87	82.91	82.39
Vanilla BiLSTM+CRF	66.97	62.02	64.4
+ TE	72.62	67.15	69.78
+ TE + TOKEN-EM	72.13	68.85	70.46
+ TE + TOKEN-POS	72.18	70.22	71.19
+ TE + SEQ	75.34	69.55	72.33
+ TE + DA	76.06	74.80	75.43
+ TE + DA + TOKEN-EM	75.21	72.84	74.52
+ TE + DA + TOKEN-POS	75.99	73.80	74.88
+ TE + DA + SEQ (student)	77.94	74.23	76.05

Table 12: Results on CoNLL-2003.

Model	P	R	F1
BERT+CRF (teacher)	91.46	92.61	92.03
Vanilla BiLSTM+CRF	80.67	79.5	80.08
+ TE	88.43	88.27	88.35
+ TE + TOKEN-EM	87.70	89.23	88.46
+ TE + TOKEN-POS	88.38	88.95	88.66
+ TE + SEQ	89.29	89.79	89.69
+ TE + DA	90.20	91.18	90.69
+ TE + DA + TOKEN-EM	90.18	90.80	90.49
+ TE + DA + TOKEN-POS	90.70	91.27	90.98
+ TE + DA + SEQ (student)	90.85	91.50	91.17

Table 13: Results on OntoNotes 5.0.

Model	P	R	F1
BERT+CRF (teacher)	89.51	88.35	89.92
Vanilla BiLSTM+CRF	82.88	82.29	82.58
+ TE	87.48	87.22	87.35
+ TE + TOKEN-EM	87.41	87.37	87.39
+ TE + TOKEN-POS	88.51	87.34	88.17
+ TE + SEQ	88.72	87.96	88.34
+ TE + DA	88.10	86.85	87.52
+ TE + DA + TOKEN-EM	88.31	87.80	88.05
+ TE + DA + TOKEN-POS	89.02	87.57	88.29
+ TE + DA + SEQ (student)	89.51	88.31	88.91

Table 9: Comparison of F1 scores on four transformer layers as student model. The abbreviations herein is as same as Table 1, except BP means BERT pretrained parameters are used. Noting that, we simply choose the first four transformer layers from BERT model to initialize the student model, however, other reasonable strategies as in (Sun et al., 2019b; Jiao et al., 2019) will be studied in our future work.

Models		**Datasets**				
		English Datasets		**Chinese Datasets**		
English	**Chinese**	**CoNLL-2003**	**OntoNotes 5.0**	**MSRA**	**Weibo**	**OntoNotes 4.0**
BERT+CRF (teacher)	Vanilla BERT+CRF	NA	NA	94.80	67.33	79.16
	Glyce+BERT	NA	NA	95.54	67.60	80.62
	BERT+CRF (teacher)	**92.03**	**89.92**	**95.91**	**71.22**	**82.39**
+ BP (no distillation)		89.01	88.16	93.68	64.09	75.07
+ BP + TOKEN-EM		88.53	88.49	94.47	65.99	75.59
+ BP + TOKEN-POS		89.21	88.46	94.55	65.10	75.35
+ BP + SEQ		**89.81**	**88.60**	**94.62**	**66.42**	**77.53**
+ BP + DA (no distillation)		90.86	88.59	94.83	70.11	79.17
+ BP + DA + TOKEN-EM		90.83	88.67	94.69	70.09	79.35
+ BP + DA + TOKEN-POS		91.04	88.78	94.84	69.73	79.31
+ BP + DA + SEQ (student)		**91.21**	**89.06**	**95.07**	**70.65**	**79.91**