An Effective Incorporating Heterogeneous Knowledge Curriculum Learning for Sequence Labeling

Xuemei Tang[1], Jun Wang[2], Qi Su[3]*, Chu-Ren Huang[1], and Jinghang Gu[1]*
[1]The Hong Kong Polytechnic University, Hong Kong, China
[2]Department of Information Management, Peking University, Beijing, China
[3]School of Foreign Languages, Peking University, Beijing, China
xuemeitang00@gmail.com, {sukia,junwang}@pku.edu.cn
{churen.huang, jinghang.gu}@polyu.edu.hk

Abstract

Sequence labeling models often benefit from incorporating external knowledge. However, this practice introduces data heterogeneity and complicates the model with additional modules, leading to increased expenses for training a high-performing model. To address this challenge, we propose a dual-stage curriculum learning (DCL) framework specifically designed for sequence labeling tasks. The DCL framework enhances training by gradually introducing data instances from easy to hard. Additionally, we introduce a dynamic metric for evaluating the difficulty levels of sequence labeling tasks. Experiments on several sequence labeling datasets show that our model enhances performance and accelerates training, mitigating the slow training issue of complex models [1].

1 Introduction and Related Work

Sequence labeling is a core task in natural language processing (NLP) that involves assigning labels to individual elements in a sequence. Recent advancements in neural network methods have significantly improved performance in sequence labeling tasks (Zhang et al., 2014; Chen et al., 2017a; Zhang et al., 2018; Tian et al., 2020a; Nguyen et al., 2021; Hou et al., 2021; Liu et al., 2021). Some studies have explored integrating external knowledge, such as n-grams, lexicons, and syntax, to enhance these models. However, this integration adds heterogeneity and complexity to the input data. Additionally, incorporating such knowledge often necessitates extra encoding modules, like attention mechanisms (Liu et al., 2021; Tian et al., 2020b) or graph neural networks (GNN) (Chen et al., 2017b; Gui et al., 2019; Nie et al., 2022), which increase model parameters and make the system more computationally expensive to develop.

Curriculum Learning (CL) (Bengio et al., 2009) effectively addresses these challenges by simulating the human learning process, where training samples are introduced progressively from easy to hard. This approach facilitates efficient learning from heterogeneous data while enhancing both the speed and performance of the model (Bengio et al., 2009; Wang et al., 2021). CL has shown success in a variety of NLP tasks, including machine translation (Wan et al., 2020), dialogue generation (Zhu et al., 2021), and text classification (Zhang et al., 2022). Data-selection strategies are crucial in CL. However, these difficulty metrics primarily focus on the sentence level, such as Mohiuddin et al. (2022), Yuan et al. (2022) and Liu et al. (2024)'s works, and there is a lack of token-level and word-level metrics to measure the difficulty of sequence labeling tasks.

To address this gap, in this paper, we introduce a dual-stage curriculum learning (DCL) framework specifically designed for sequence labeling tasks. The first stage is data-level CL, where we train a basic teacher model on all available training data, aiming to alleviate the cold start problem of the student model. The second stage is model-level CL, where we start training the student model on a selected subset of the teacher model and gradually expand the training subset by considering the difficulty of the data and the state of the student model. Furthermore, we explore different difficulty metrics for sequence labeling tasks within the DCL framework. These metrics include a pre-defined metric, such as sentence length, and model-aware metrics, namely Top-N least confidence (TLC), Maximum normalized log-probability (MNLP), and Bayesian uncertainty (BU). Finally, we choose the classical sequence labeling tasks, Chinese word segmentation (CWS), part-of-speech (POS) tagging, and named entity recognition (NER), to validate our proposed approach.

*Corresponding authors.
[1]https://github.com/tangxuemei1995/DCL4SeqLabeling

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 495–503

2 Method

The framework proposed in this study consists of three main components: a teacher sequence labeling model, a student sequence labeling model, and a DCL training strategy. It is worth noting that the DCL is independent of the sequence labeling model.

Following previous works (Zhang et al., 2018; Gong et al., 2019; Fu et al., 2020), in sequence labeling tasks, we feed an input sentence $X = \{x_1, ...x_i..., x_M\}$ into the encoder, and the decoder then outputs a label sequence $Y^* = \{y_1^*, ...y_i^*...y_M^*\}$, where y_i^* represents a label from a pre-defined label set T, and M denotes the length of sentence.

2.1 Dual-stage Curriculum Learning

Algorithm 1 Training Process with DCL

Input: Original corpus $\mathcal{D}$, difficulty metric $S(\cdot)$, teacher model epochs E_0, student model epochs E_s, scheduler λ, length function $|\cdot|$
Output: Trained student model θ
// Data-level Curriculum Learning
1: Train teacher model θ_0 on $\mathcal{D}$ for E_0 epochs
2: Compute $S(\theta_0)$ for each sample in $\mathcal{D}$
3: Sort $\mathcal{D}$ by $S(\theta_0)$ in ascending order to obtain ranked dataset $\mathcal{D}_r$
// Model-level Curriculum Learning
4: Initialize λ_0 (starting curriculum ratio)
5: $m \leftarrow \lambda_0 \cdot |\mathcal{D}|$
6: Student training set $\mathcal{D}_s \leftarrow \mathcal{D}_r[0:m]$
7: Remaining data $\mathcal{D}_o \leftarrow \mathcal{D}_r[m:]$
8: **for** *epoch* $= 1$ to E_s **do**
9: **if** $\lambda < 1$ **then**
10: *a)* Train student model on $\mathcal{D}_s$ to obtain current θ_*
11: *b)* Compute $S(\theta_*)$ for all samples in $\mathcal{D}_o$
12: *c)* Sort $\mathcal{D}_o$ by $S(\theta_*)$ in ascending order to get updated $\mathcal{D}_r$
13: *d)* Update λ using Eq. 6
14: *e)* Calculate new data size: $m \leftarrow \lambda \cdot |\mathcal{D}| - |\mathcal{D}_s|$
15: *f)* Expand $\mathcal{D}_s$ with new samples: $\mathcal{D}_s \mathrel{+}= \mathcal{D}_r[0:m]$
16: *g)* Update remaining data: $\mathcal{D}_o \leftarrow \mathcal{D}_r[m:]$
17: **else**
18: Train student model on $\mathcal{D}_s$
19: **end if**
20: **end for**

We propose a novel dual-stage curriculum learning approach: *data-level* CL and *model-level* CL, as detailed in Algorithm 1.

At the data level, we first train a basic teacher model on the entire dataset $\mathcal{D}$ for E_0 epochs, where E_0 is smaller than the total epochs needed for convergence (Line 1). The teacher model θ_0 is then used to calculate difficulty scores $S(\theta_0)$ for each sample (Line 2), and the samples are sorted by difficulty to form a ranked dataset $\mathcal{D}_r$ (Line 3).

At the model level, we address the cold-start issue by initializing the student model training set $\mathcal{D}_s$ with a subset of $\mathcal{D}_r$ (Lines 4-6). The proportion of samples, controlled by the parameter λ, governs the curriculum learning process. The remaining data, $\mathcal{D}_o$, is incorporated into $\mathcal{D}_s$ gradually as λ increases. The number of new samples to be added is denoted as m (Lines 5, 14).

The student model is trained on $\mathcal{D}_s$ to update the model parameters θ_* (Line 10). Then, θ_* is used for the difficulty calculation of the samples in $\mathcal{D}_o$ (Line 11). Next, $\mathcal{D}_o$ is ranked by new difficulty scores, forming a new ranked dataset $\mathcal{D}_r$ (Line 12). The threshold λ is updated (Line 13), and new samples are added to $\mathcal{D}_s$ based on λ (Lines 14-15). As λ approaches 1, all of $\mathcal{D}_o$ is added to $\mathcal{D}_s$. The complete dataset is then used to train the student model to convergence.

The key elements in Algorithm 1 are the difficulty metric $S(\cdot)$ and threshold λ, which control the difficulty ranking of samples and the progression of training, respectively. The design of these components will be discussed in the following sections.

2.2 Difficulty Metrics

We now provide a detailed formulation for calculating the difficulty $S(\cdot)$ in Algorithm 1. In sequence labeling tasks, sample difficulty is tied to individual tokens, but assessing token-level difficulty is challenging. We use uncertainty from active learning to measure the model's confidence in labeling training samples.

Bayesian Uncertainty (BU). Following Buntine and Weigend (1991), model uncertainty can be assessed using Bayesian Neural Networks. As noted by Wang et al. (2019), higher predicted probability variance indicates greater uncertainty, suggesting that the model is less confident about the sample. In this work, we employ the widely-used Monte Carlo dropout (Gal and Ghahramani, 2016) to approximate Bayesian inference.

First, we apply Monte Carlo dropout (Gal and Ghahramani, 2016) to obtain each sample of token-level tagging probabilities. Specifically, for each token x_i, we perform K stochastic forward passes through the model, resulting in K predicted distributions $P(y_i \mid x_i)_1, \ldots, P(y_i \mid x_i)_K$. This provides K predictions with associated probabilities for each token. Then the expectation of token-level

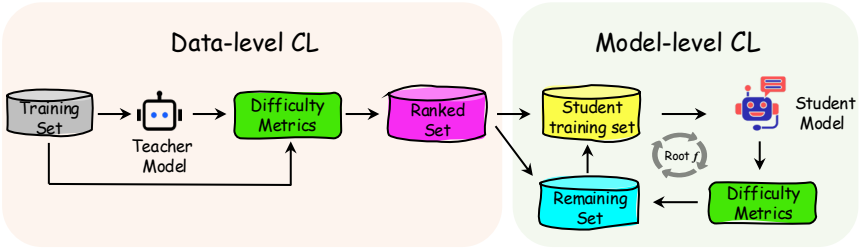

Figure 1: The framework of the proposed model consists of a teacher model, a student model, and a DCL strategy. Here, "Root f" represents *Root function*.

tagging probability can be approximated by

$$\mathbb{E}[P(y_i|x_i)] \approx \frac{1}{K}\sum_{k=1}^{K} P(y_i|x_i)_k \quad (1)$$

The variance of token-level tagging probability on the label set can be approximated by

$$var(x_i,\theta) \approx \sum_{y_i \in T}(\frac{1}{K}\sum_{k=1}^{K} P(y_i|x_i)_k^2 - \mathbb{E}[P(y_i|x_i)]^2) \quad (2)$$

Now, we obtain the variance of each token $var(x_i,\theta)$, then we use the average variance score of all tokens in the sequence as the sentence-level variance as follows.

$$var(\theta)_{aver.} = \frac{1}{M}\sum_{i=1}^{M} var(x_i,\theta) \quad (3)$$

The maximum variance score $var(\theta)_{max}$ also is valuable, which reflects the highest uncertainty in the sequence.

$$var(\theta)_{max} = max_{i\in[1,M]} var(x_i,\theta) \quad (4)$$

The final uncertainty score or difficulty score of each sequence is calculated as follows.

$$S(\theta)^{BU} = var(\theta)_{max} + var(\theta)_{aver.} \quad (5)$$

Both at the data level and model level, the difficulty of training samples is measured by the above various $S(\theta)$.

2.3 Training Scheduler

The training scheduler regulates the pace of CL. In our approach, we employ the *Root function* as the control mechanism. This function ensures that the model receives sufficient time to learn newly introduced examples while gradually decreasing the number of newly added examples throughout the training process.

$$\lambda = \min\left(1, \sqrt{\frac{1-\lambda_0^2}{E_{grow}} \cdot t + \lambda_0^2}\right) \quad (6)$$

where E_{grow} denotes the number of epochs required for λ to reach 1, while $\lambda_0 > 0$ represents the initial proportion of the easiest training samples. t indicates the t_{th} training epochs. When λ reaches 1, the model has access to the entire training dataset.

3 Experiments

3.1 Dataset and Experimental Configurations

Dataset. Chinese word segmentation (CWS) and part-of-speech (POS) tagging are representative sequence labeling tasks. So we evaluate our approach using three CWS and POS tagging datasets, including Chinese Penn Treebank version 5.0 [2], 6.0 [3], and PKU. More dataset details can be found in Appendix A.

Teacher and student models. In this study, the basic transfer teacher framework is RoBERTa + Softmax. For the student model, we select two representative complex models introduced by Tian et al. (2020b) and Tang et al. (2024). In their work, Tian et al. (2020b) employed an attention mechanism framework, McASP, to integrate lexicons and n-grams for the joint CWS and POS tagging task, using BERT as the encoder. Meanwhile, Tang et al. (2024) incorporated syntax and semantic knowledge into sequence labeling tasks through a GCN framework called SynSemGCN, with RoBERTa as the sequence encoder.

Curriculum learning baselines. We compare our difficulty metric with four baseline difficulty metrics for CL: ***a. Random***: Samples are assigned in random order; ***b. Sentence Length (Length)***: Samples are ranked from shortest to longest, based on the intuition that longer sequences are more challenging to encode; (Random and Length metics represent simple CL, namely without the teacher model). ***c. Top-N Least Confidence (TLC)***: The difficulty of a sequence is determined by using the

[2] https://catalog.ldc.upenn.edu/LDC2005T01
[3] https://catalog.ldc.upenn.edu/LDC2007T36

Model	CL Setting	CTB5		CTB6		PKU	
		CWS	POS	CWS	POS	CWS	POS
Tian et al. (2020a)	-	98.73	96.60	97.30	94.74	-	-
Tian et al. (2020b) (McASP)	-	98.77	96.77	97.43	94.82	-	-
Liu et al. (2021)	-	-	97.14	-	-	-	-
Tang et al. (2024) (SynSemGCN)	-	98.83	96.77	97.86	94.98	98.05	95.50
McASP	Rand.	98.81	96.84	97.37	94.90	98.38	96.27
	Length	98.83	96.85	97.35	94.82	98.40	96.25
	TLC	98.83	96.89	97.37	94.83	98.41	96.30
	MNLP	98.85	96.81	97.41	94.92	98.41	96.30
	BU	**98.91**	96.87	97.42	94.90	98.43	96.32
SynSemGCN	Rand.	98.84	97.86	97.99	95.05	98.48	96.40
	Length	98.80	96.84	97.40	94.94	98.53	96.48
	TLC	98.83	97.81	97.98	95.02	**98.61**	**96.55**
	MNLP	98.78	97.72	98.04	95.13	98.56	96.48
	BU	98.90	**97.95**	**98.05**	**95.14**	98.59	96.54

Table 1: Experimental results of different models using different CL settings on test sets of three datasets. Here, "CWS" represents the F1 value of CWS, and "POS" means the F1 value of the joint CWS and POS tagging. "-" means without the CL training strategy, and "TLC", "MNLP", and "BU" means using the DCL setting with different difficulty metrics. The maximum F1 scores for each dataset are highlighted.

N tokens with the lowest confidence; ***d. Maximum Normalized Log-Probability (MNLP)***: The difficulty is assessed by calculating the product of the label probabilities for all tokens in the sequence. The detailed computation processes for TLC and MNLP are provided in Appendix B.

For further details on the important hyper-parameters of the model, please refer to Appendix C. We discuss the selection process of these parameter values in detail in the Appendix D.

3.2 Overall Experimental Results

Table 1 presents the experimental results of baselines and two models with different CL settings. The experimental results reveal several noteworthy conclusions.

Firstly, the DCL methodology introduced in this paper is flexible and can be integrated with various complex models. As shown in Table 1, the difficulty metrics proposed here outperform the Random and Length metrics across most datasets. Specifically, the BU metric consistently delivers the best performance on the majority of datasets when applied to the SynSemGCN model, surpassing the TLC and MNLP metrics.

Additionally, we compare our approach with previous methods that incorporate external knowledge or resources into the encoder. The results reveal that models using CL exhibit significant performance improvements, surpassing the performance of earlier methods.

Model	CTB5		Time
	CWS	POS	
Ours	**98.90**	**97.95**	287m
w/o data CL(BU)	**98.90**	97.88	-
w/o model CL(BU)	98.85	97.51	-
w/o DCL	98.75	96.73	393m

Table 2: Ablation experimental results of DCL. The baseline model "w/o DCL" denotes the model SynSemGCN; "w/o model CL" means the student model always uses the initial data order sorted by the transfer teacher model; "w/o data CL" indicates the initial training samples for the student model is drawn randomly from the training set; "Ours" indicates "SynSemGCN+DCL(BU)". Both teacher and student models with DCL in this table use BU as the difficulty metric. "Time" means the training time (in minutes).

Model	CTB5	
	CWS	POS
McASP with BU	**98.91**	**96.87**
w/o $var(\theta)_{max}$	98.78	96.78
w/o $var(\theta)_{aver.}$	98.86	96.74
McASP	98.73	96.60

Table 3: Ablation experimental results of two parts in BU metrics (Eq. 5).

Models	Weibo (Chinese)	Note4 (Chinese)	CoNLL-2003 (English)
BERT	66.22	79.15	90.94
BERT + CL (Length)	66.81	79.63	90.79
BERT + DCL (TLC)	**67.52**	79.53	91.30
BERT + DCL (MNLP)	65.73	79.95	91.15
BERT + DCL (BU)	66.74	**80.02**	**91.77**

Table 4: Performance comparison of different difficulty metrics on three NER datasets.

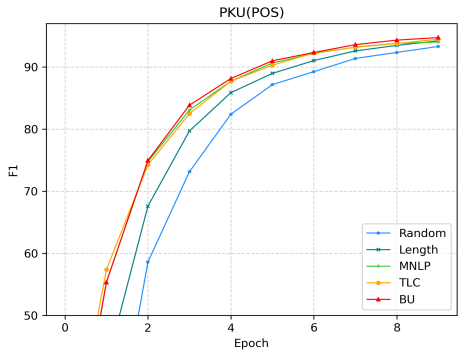

Figure 2: The F1 scores on the dev set of PKU with different difficulty metrics in the model-level CL training process.

3.3 Effect of Dual-stage Curriculum Learning

In this section, we discuss the impact of DCL. We perform ablation studies by removing either the data-level CL or the model-level CL. The results are summarized in Table 2. Model-level CL has a more significant impact than data-level CL. This is intuitive, as model-level CL influences the entire training process, while data-level CL primarily affects the early stages of student model training.

We also compare the training time of models with and without DCL. The experimental results in Table 2 show that all models were trained for 50 epochs. The training time for models using DCL includes the time spent on training the teacher model and calculating the difficulty values for the student model. The results indicate that DCL improves model performance and reduces training time by over 25%.

3.4 Ablation Study on BU Difficulty Components

We adopt McASP (Tian et al., 2020b) as the backbone model, incorporating DCL as the training strategy and BU as the difficulty metric. To examine the contribution of each component in BU, we conduct ablation experiments on its two parts: $var(\theta)_{max}$ and $var(\theta)_{aver.}$, as shown in Table 3. Removing either component results in performance degradation, indicating that both components are crucial. Moreover, the comparable drop in performance suggests that $var(\theta)_{max}$ and $var(\theta)_{aver.}$ contribute similarly to the effectiveness of DCL.

3.5 Comparison of Difficulty Metrics

In this section, we examine the impact of different difficulty metrics during the model-level CL training process for the SynSemGCN model. Figure 2 shows the F1 score change on the PKU dataset development set over the first 10 epochs of model-level CL training. After 10 epochs, all training data are used, so the initial 10 epochs highlight the effect of different metrics. From the figure, we observe that BU, in particular, achieves the best performance, indicating that uncertainty-based metrics can select samples that better align with the model's learning trajectory, leading to faster learning.

3.6 Generalization Capability

We conduct additional experiments to demonstrate the applicability of our method to the NER task. We select two Chinese and one English NER datasets: Weibo [4], OntoNotes4 [5] and CoNLL-2003 (Tjong Kim Sang and De Meulder, 2003). The statistics of three datasets are shown in Table 6. We compare the performance of models using DCL and CL (Length) with a model without CL on these datasets. As shown in Table 4, the results of the DCL method outperform those of BERT+CL (Length) and BERT (no CL), indicating the effectiveness of our method. This also suggests that our method can be applied to sequence labeling tasks beyond CWS and POS tagging.

4 Conclusion

This paper introduces a novel dual-stage curriculum learning framework aimed at enhancing performance and accelerating the training process for sequence labeling tasks. Focusing on the sequence labeling task of CWS, POS tagging, and NER, this framework demonstrates its effectiveness.

[4] https://catalog.ldc.upenn.edu/LDC2013T19/
[5] https://github.com/cchen-nlp/weiboNER

Limitations

There are several limitations to our study. First, the design of our difficulty metrics involves the tuning of multiple hyperparameters, which may complicate optimization. Second, we did not explore a curriculum learning process that progresses from hard to easy examples. Third, we focused on a single variation of the λ parameter to control CL and did not investigate alternative methods for adding training data.

Acknowledgments

This research is supported by the NSFC project "The Construction of the Knowledge Graph for the History of Chinese Confucianism" (Grant No. 72010107003) and The Hong Kong Polytechnic University Project "Evaluating the Syntax-Semantics Knowledge in Large Language Models" (Grant No. P0055270).

References

Ankit Agrawal, Sarsij Tripathi, and Manu Vardhan. 2021. Active learning approach using a modified least confidence sampling strategy for named entity recognition. *Progress in Artificial Intelligence*, 10(2):113–128.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, page 41–48, New York, NY, USA. Association for Computing Machinery.

Wray L. Buntine and Andreas S. Weigend. 1991. Bayesian back-propagation. *Complex Systems*.

Xinchi Chen, Xipeng Qiu, and Xuanjing Huang. 2017a. A feature-enriched neural model for joint chinese word segmentation and part-of-speech tagging. In *IJCAI*.

Xinchi Chen, Zhan Shi, Xipeng Qiu, and Xuanjing Huang. 2017b. Dag-based long short-term memory for neural word segmentation. *Preprint*, arXiv:1707.00248.

Aron Culotta and Andrew McCallum. 2005. *Reducing Labeling Effort for Structured Prediction Tasks:*. Fort Belvoir, VA.

Jinlan Fu, Pengfei Liu, Qi Zhang, and Xuanjing Huang. 2020. Rethink cws: Is chinese word segmentation a solved task? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5676–5686, Online. Association for Computational Linguistics.

Yarin Gal and Zoubin Ghahramani. 2016. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. (arXiv:1506.02142). ArXiv:1506.02142 [cs, stat].

Jingjing Gong, Xinchi Chen, Tao Gui, and Xipeng Qiu. 2019. Switch-lstms for multi-criteria chinese word segmentation. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 6457–6464. AAAI Press.

Tao Gui, Yicheng Zou, Qi Zhang, Minlong Peng, Jinlan Fu, Zhongyu Wei, and Xuanjing Huang. 2019. A lexicon-based graph neural network for chinese ner. page 11.

Yang Hou, Houquan Zhou, Zhenghua Li, Yu Zhang, Min Zhang, Zhefeng Wang, Baoxing Huai, and Nicholas Jing Yuan. 2021. A coarse-to-fine labeling framework for joint word segmentation, pos tagging, and constituent parsing. In *Proceedings of the 25th Conference on Computational Natural Language Learning*, page 290–299, Online. Association for Computational Linguistics.

Honglin Liu, Peng Hu, Changqing Zhang, Yunfan Li, and Xi Peng. 2024. Interactive deep clustering via value mining. In *Advances in Neural Information Processing Systems*, volume 37, pages 42369–42387. Curran Associates, Inc.

Wei Liu, Xiyan Fu, Yue Zhang, and Wenming Xiao. 2021. Lexicon enhanced chinese sequence labeling using bert adapter. *arXiv:2105.07148 [cs]*. ArXiv: 2105.07148.

Tasnim Mohiuddin, Philipp Koehn, Vishrav Chaudhary, James Cross, Shruti Bhosale, and Shafiq Joty. 2022. Data selection curriculum for neural machine translation. (arXiv:2203.13867). ArXiv:2203.13867 [cs].

Duc-Vu Nguyen, Linh-Bao Vo, Ngoc-Linh Tran, Kiet Van Nguyen, and Ngan Luu-Thuy Nguyen. 2021. Joint chinese word segmentation and part-of-speech tagging via two-stage span labeling. *arXiv:2112.09488 [cs]*. ArXiv: 2112.09488.

Yu Nie, Yilai Zhang, Yongkang Peng, and Lisha Yang. 2022. Borrowing wisdom from world: modeling rich external knowledge for chinese named entity recognition. *Neural Computing and Applications*, 34(6):4905–4922.

Yan Shao, Christian Hardmeier, Jörg Tiedemann, and Joakim Nivre. 2017. Character-based joint segmentation and pos tagging for chinese using bidirectional rnn-crf. *Preprint*, arXiv:1704.01314.

Yanyao Shen, Hyokun Yun, Zachary C. Lipton, Yakov Kronrod, and Animashree Anandkumar. 2018. Deep active learning for named entity recognition. (arXiv:1707.05928). ArXiv:1707.05928 [cs].

Xuemei Tang, Jun Wang, and Qi Su. 2024. Incorporating knowledge for joint chinese word segmentation and part-of-speech tagging with synsemgcn. *Aslib Journal of Information Management*, ahead-of-print(ahead-of-print).

Yuanhe Tian, Yan Song, Xiang Ao, Fei Xia, Xiaojun Quan, Tong Zhang, and Yonggang Wang. 2020a. Joint Chinese word segmentation and part-of-speech tagging via two-way attentions of auto-analyzed knowledge. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8286–8296, Online. Association for Computational Linguistics.

Yuanhe Tian, Yan Song, and Fei Xia. 2020b. Joint Chinese word segmentation and part-of-speech tagging via multi-channel attention of character n-grams. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2073–2084, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.

Yu Wan, Baosong Yang, Derek F. Wong, Yikai Zhou, Lidia S. Chao, Haibo Zhang, and Boxing Chen. 2020. Self-paced learning for neural machine translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, page 1074–1080, Online. Association for Computational Linguistics.

Kun Wang, Chengqing Zong, and Keh-Yih Su. 2010. A character-based joint model for chinese word segmentation. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, page 1173–1181, Beijing, China. Coling 2010 Organizing Committee.

Shuo Wang, Yang Liu, Chao Wang, Huanbo Luan, and Maosong Sun. 2019. Improving back-translation with uncertainty-based confidence estimation. 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)*, page 791–802, Hong Kong, China. Association for Computational Linguistics.

Xin Wang, Yudong Chen, and Wenwu Zhu. 2021. A survey on curriculum learning. (arXiv:2010.13166). ArXiv:2010.13166 [cs].

Siyu Yuan, Deqing Yang, Jiaqing Liang, Zhixu Li, Jinxi Liu, Jingyue Huang, and Yanghua Xiao. 2022. Generative entity typing with curriculum learning. (arXiv:2210.02914). ArXiv:2210.02914 [cs].

Meishan Zhang, Nan Yu, and Guohong Fu. 2018. A simple and effective neural model for joint word segmentation and pos tagging. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 26(9):1528–1538.

Meishan Zhang, Yue Zhang, Wanxiang Che, and Ting Liu. 2014. Type-supervised domain adaptation for joint segmentation and POS-tagging. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 588–597, Gothenburg, Sweden. Association for Computational Linguistics.

Xulong Zhang, Jianzong Wang, Ning Cheng, and Jing Xiao. 2022. Improving imbalanced text classification with dynamic curriculum learning. (arXiv:2210.14724). ArXiv:2210.14724 [cs].

Ying Zhang, Stephan Vogel, and Alex Waibel. 2004. Interpreting bleu/nist scores: How much improvement do we need to have a better system? In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*, Lisbon, Portugal. European Language Resources Association (ELRA).

Qingqing Zhu, Xiuying Chen, Pengfei Wu, JunFei Liu, and Dongyan Zhao. 2021. Combining curriculum learning and knowledge distillation for dialogue generation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, page 1284–1295, Punta Cana, Dominican Republic. Association for Computational Linguistics.

A Dataset

The details of the three datasets are given in Table 5. Regarding the CTB datasets, we follow the same approach as previous works (Shao et al., 2017; Tian et al., 2020a) by splitting the data into train/dev/test sets. In the case of PKU, We randomly select 10% of the training data to create the development set.

Datasets		CTB5	CTB6	PKU
Train	Sent.	18k	23K	17k
	Word	494k	99k	482k
Dev	Sent.	350	2K	1.9k
	Word	7k	60K	53k
Test	Sent.	348	3K	3.6k
	Word	8k	12k	97k

Table 5: Detail of the three datasets.

Datasets	Type	Train	Dev	Test
Weibo	Sentences	1.35K	0.27K	0.27K
	Entities	1.89K	0.39K	0.42K
OntoNotes	Sentences	15.7K	4.3K	4.3K
	Entities	13.4K	6.95K	7.7K
CoNLL2003	Sentences	15.0K	3.5K	3.7K
	Entities	23.5K	5.9K	5.7K

Table 6: Detail of the two NER datasets.

B Difficulty Metric Baselines

Top-N least confidence (TLC). Culotta and McCallum (2005) proposed a confidence-based strat-

egy for sequence models called least confidence (LC). This approach sorts the samples in ascending order based on the probability of the most possible label predicted by the model.

The least confidence of each token is calculated as follows.

$$\phi^{LC}_{(x_i,\theta)} = 1 - max_{y_i \in T} P(y_i|x_i) \quad (7)$$

where x_i is the i_{th} token in the input sentence, θ denotes model parameters, y_i is a pre-defined label, T represents the pre-defined label set. $max_{y_i \in T} P(y_i|x_i)$ aims to find the probability of the most possible label predicted by the model. The smaller $\phi^{LC}_{(x_i,\theta)}$ reflects the more confident the model is in predicting the label of x_i.

According to Agrawal et al. (2021), the confidence level of a sentence in a sequence labeling task is typically determined based on a set of representative tokens. Therefore, we select the top N tokens with the highest least confidence in the sentence and then use their average value as the difficulty score of the sentence. Finally, the TLC difficulty metric is formulated as follows.

$$S(\theta)^{TLC} = \frac{1}{N} \sum_{n=1}^{N} \phi^{LC}_{(x_n,\theta)} \quad (8)$$

Maximum normalized log-probability (MNLP). Shen et al. (2018) used MNLP as a confidence strategy to find the product of the maximum probabilities of each token, which is equivalent to taking the logarithm of each probability and summing them. Finally, it is normalized to obtain the confidence score of the sentence as follows.

$$\begin{aligned} &\prod_{i=1}^{M} max_{y_i \in T} P(y_i|x_i) \iff \\ &\sum_{i=1}^{M} log\{max_{y_i \in T} P(y_i|x_i)\} \end{aligned} \quad (9)$$

where M is the length of the sentence. The difficulty of a sentence decreases as the confidence level increases. To account for this relationship, we introduce a negative sign. Additionally, in order to reduce the impact of sentence length, we apply a normalization operation. Finally, MNLP is formulated as follows.

$$S(\theta)^{MNLP} = -\frac{1}{M} \sum_{i=1}^{M} log\{max_{y_i \in T} P(y_i|x_i)\} \quad (10)$$

C Parameters Setting

The key experimental parameter settings are shown in Table 7.

Hyper-parameters	Value
E_0	5
E_s	50
λ_0	0.3
E_{grow}	10
K	3
N	5

Table 7: Experiment hyper-parameters setting.

E_{grow}	CTB5		PKU	
	CWS	POS	CWS	POS
5	98.88	97.89	98.65	**97.01**
10	**99.06**	**98.96**	**98.77**	96.97
15	98.84	97.69	98.70	96.90

Table 8: The effect of E_{grow} in Eq. 6.

D Effect of Hyper-parameters

In this section, we explore the impact of the hyperparameters on the performance of DCL. The adjustment of the parameters is based on the SynSemGCN+DCL(BU) model.

First, we investigate the impact of the hyperparameter λ_0 on DCL performance. We conduct the experiments on the CTB5 dataset, tuning the value of λ_0 in the model-level pacing function Eq. 6, and the experimental results are represented by a line graph as shown in Figure 3. As observed, the model achieves optimal performance when $\lambda_0 = 0.3$. However, when the value exceeds 0.4, the model's performance gradually deteriorates.

Additionally, we examine the impact of E_{grow} in Eq. 6, which controls the number of epochs for λ to reach 1. As shown in Table 8, when E_{grow} is set to 10, the model exhibits superior performance on both the CTB5 and PKU datasets. Therefore, we adopt E_{grow} as 10 epochs in our experiments.

Next, we assess the impact of the training epochs E_0 of the teacher model, which initializes the difficulty ranking of the training data for the student model. We aim to investigate whether a more mature teacher model contributes to improved performance. For this purpose, we conduct experiments on both the CTB5 and PKU datasets, utilizing teacher models trained for 5, 10, and 15 epochs

E_0	CTB5		PKU	
	CWS	POS	CWS	POS
5	**99.06**	**98.96**	**98.77**	**96.97**
10	98.98	97.90	98.54	96.69
15	98.73	96.87	98.53	96.54

Table 9: The impact of the number of epochs of teacher model, E_0.

Para.	CTB6		PKU	
	CWS	POS	CWS	POS
K=2	**98.17**	95.43	98.73	96.58
K=3	98.10	**95.59**	**98.77**	**96.97**
K=4	98.09	95.56	98.69	96.38

Table 10: The effect of of K times dropout in BU difficulty metric.

to rank the initial training data for the student models.

The experimental results, as shown in Table 9, reveal that a more mature teacher model does not necessarily lead to better performance. Instead, the student model achieves optimal results when the teacher model is trained for 5 epochs. One possible explanation for this finding is that a teacher model with fewer training epochs aligns better with the initial state of the student model, allowing for a more suitable estimation of sample difficulty.

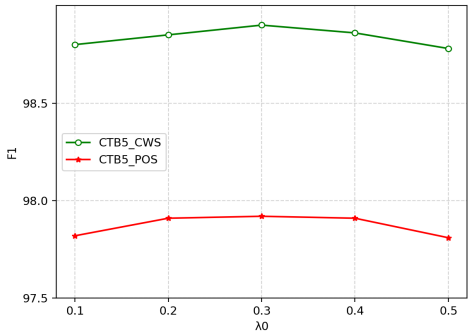

Figure 3: The impact of model-level curriculum learning hyper-parameters λ_0.

Then, we explore the impact of different K values on the BU difficulty metric, which determines the number of dropout times. The experiments are conducted on the CTB6 dataset, and the results are summarized in Table 10. Notably, the model achieves optimal performance when $K = 3$. Therefore, we select $K = 3$ for all the above experiments.

Finally, we evaluate the effect of varying N in the TLC metric. As shown in Table 11, the best performance is achieved when $N = 5$.

E Statistical Significance Test

In this section, we conduct significance testing experiments. Following Wang et al. (2010), we use the bootstrapping method proposed by Zhang et al. (2004), which is operated as follows. In this process, starting with a test set T_0 comprising N test examples, we repeatedly sample N samples from T_0 to form T_1 and then repeat the process for M times to form the test set collection, $\{T_1, T_2, ..., T_M\}$, where M is set to 1000 in our testing procedure. Two systems denoted as A and B, are assessed on the initial test set T_0, resulting in scores a_0 and b_0, respectively. The disparity between the two systems, labeled as δ_0, is calculated as $\delta_0 = a_0 - b_0$. Repeating this process for each test set produces a set of M discrepancy scores, denoted as $\{\delta_0, \delta_1, ..., \delta_M\}$.

Para.	PKU	
	CWS	POS
N=1	98.55	96.49
N=2	98.53	96.49
N=3	98.56	96.52
N=4	98.55	96.51
N=5	**98.71**	**96.64**
N=6	98.52	96.52
N=7	98.56	96.51
N=8	98.55	96.48
N=9	98.53	96.49
N=10	98.55	96.51

Table 11: The impact of N in TLC difficulty metric.

Models		CTB5	
A	B	CWS	POS
BERT+DCL(BU)	BERT	>	>
BERT+DCL(MNLP)	BERT	>	>
BERT+DCL(TLC)	BERT	>	>
BERT+DCL(BU)	BERT+CL(Length)	>	>
BERT+DCL(MNLP)	BERT+CL(Length)	>	>
BERT+DCL(TLC)	BERT+CL(Length)	>	>

Table 12: Statistical significance test of F-score for our method and baselines on the CTB5 dataset.

Following the methodology proposed by Zhang et al. (2004), we compute the 95% confidence interval for the discrepancies (i.e., the 2.5th percentile and the 97.5th percentile) between the two models. If the confidence interval does not overlap with zero, it is affirmed that the differences between systems A and B are statistically significant (Zhang et al., 2004).

Table 12 lists the significant differences between our system and the baseline system, where ">" indicates that the average value of δ exceeds zero, meaning that System A is better than System B; "<" indicates that the average value of δ does not exceed zero, meaning that System A is worse than System B; "~" indicates that there is no significant difference between the two systems. Finally, the comparison also indicates that our models are superior to the baseline.