CMU Advanced NLP Assignment 2: End-to-end NLP System Building

Yifan He
yifanhe@andrew.cmu.edu

Yi-Yu Zheng
yiyuz@andrew.cmu.edu

1 Related work

1.1 Conditional Random Field

Conditional Random Field (CRF) (Lafferty et al., 2001) is a statistical modeling method that is able to recognize pattern from data. In NLP, CRF is usually used as a refinement layer to model the correlations missed by the base model.

1.2 Knowledge Distillation

Despite pre-trained big models such as BERT have achieved top performance in various natural language processing tasks, they require expensive computations and memory. To address this issue, Knowledge Distillation (Zhou et al., 2021) is developed to transfer knowledge from large pre-trained (teacher) models to small student models by integrating cross entropy loss and distillation loss.

2 Dataset

To obtain the raw PDFs, we built a web scraper to find and download recent NLP paper PDFs from dblp. This paper web scraper supports the crawling of ACL Anthology, including ACL, EMNLP, COLING, NAACL, EACL, CoNLL, etc, as well as AAAI and IJCAI conference papers. First, it constructs the query for dblp based on the given search conditions, including keywords, year, and conference name, to perform a traversal search. Then it crawls the corresponding page and parses out the page URL of each paper, downloads the PDF, and saves it in the specified location locally. In this project, we performed queries based on [ACL, EMNLP, and AAAI], [2018, 2019, 2020, 2021], and [named entity recognition, automatic question and answer, text summary, text classification] and get raw PDFs.

The PdfReader Class from PyPDF2 package is used to read the downloaded PDFs and extract texts from each page. Then the English pipeline from spaCy package is used to firstly detect the sentence boundary and then tokenize the sentences. Sentences not longer than 10 words are omitted since they are more likely to be unrelated title words. The collected tokenized sentences are annotated manually using label-studio. 70 thousand unsupervised samples are obtained. We evenly selected about one thousand samples from each conference, which were then manually annotated with their entity labels in label-studio. We performed 8:2 data segmentation on the one thousand annotated data, the former was used as the training set and the latter was used as the test set. During model distillation, remaining unsupervised samples other than the labeled data were used for training.

3 Models

3.1 BERT-CRF

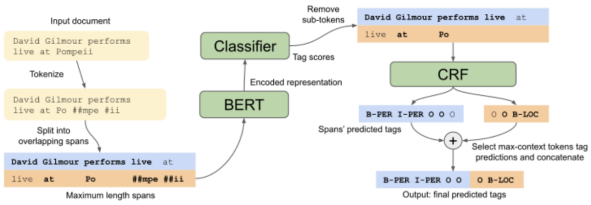

Figure 1: Model schematic for BERT-CRF.

The first method uses BERT-CRF (Souza et al., 2019) architecture that composed of a BERT model with a token-level classifier on top followed by a Linear-Chain CRF. For an input sequence of n tokens, BERT outputs an encoded token sequence with hidden dimension H. The classification model projects each token's encoded representation to the tag space, and its output scores are then fed to CRF layer, whose parameters are a matrix of tag transitions A. The matrix A is such that $A_{i,j}$ represents the score of transitioning from tag i to tag j, and also includes start and end of sequence as two additional states. For an input sequence $X = (x_1, ..., x_n)$ and a sequence of tag predic-

tions $y = (y_1, ..., y_n)$, $y_i \in [1, K]$, the score of the sequence is defined as:

$$s(\mathbf{X}, \mathbf{y}) = \sum_{i=0}^{n} A_{y_i, y_{i+1}} + \sum_{i=1}^{n} P_{i, y_i} \quad (1)$$

where y_0 and y_{n+1} are start and end tags. The model is trained to maximize the log-probability of the correct tag sequence:

$$\log(p(\mathbf{y} \mid \mathbf{X})) = s(\mathbf{X}, \mathbf{y}) - \log \left(\sum_{\tilde{\mathbf{y}} \in \mathbf{Y_X}} e^{s(\mathbf{X}, \tilde{\mathbf{y}})} \right) \quad (2)$$

where $\mathbf{Y_X}$ are all possible tag sequence. The summation is computed using dynamic programming. During evaluation, the most likely sequence is obtained by Viterbi decoding.

We think this approach would work based on the following considerations. First, since we need to manually label the dataset in a short amount of time, there is not a large training set available. In this case, we want to use a large-scale pre-trained model that has a large amount of text background. So BERT is a good choice, and BERT's attention mechanism can better capture semantics without being limited to sample size. Then, since we learned in the BERT layer what is the most likely entity label corresponding to each token in the sentence, this process takes into account the context information on both the left and right sides of each character, but the entity label corresponding to the maximum score of the output is still may be wrong. We can use CRF to learn transfer rules between adjacent entity labels. When predicting the output state of each position in the sequence, the output state of adjacent positions needs to be considered. At the same time, CRF does not have strict independence assumptions like HMM, so it can accommodate arbitrary context information. Since CRF calculates the conditional probability of the global optimal output node, it also overcomes the label bias problem of the maximum entropy Markov model.

We use bert-base-uncased (Devlin et al., 2018) as the pre-trained model and then fine-tuned on our dataset. This model was pre-trained on BookCorpus (Zhu et al., 2015), a dataset consisting of 11,038 unpublished books and English Wikipedia.

3.2 BiLSTM-CRF-Distill

The second method uses a knowledge distillation scheme that tries to efficiently transfer the knowledge learned from a big model to a light model

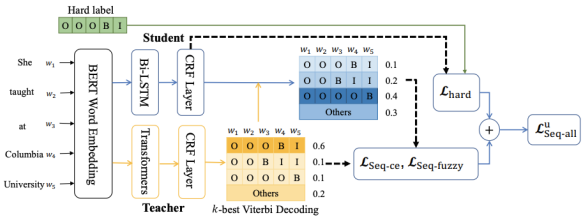

Figure 2: Model schematic for multi-grained knowledge distillation.

(Zhou et al., 2021). A major disadvantage of large models is that they are typically computationally expensive and memory intensive, while small models allow more flexibility in the control of resource usage. In knowledge distillation, a new learner is usually called a student model, and it learns from a pre-trained model, commonly called a teacher. Another reason we think of using this knowledge distillation scheme is that we have many unlabeled data, and using student model to learn from the output of teacher model as soft-labels could utilize those unsupervised samples.

We use the previously trained BERT-CRF model as our teacher model, and compact BiLSTM-CRF (Huang et al., 2015) architecture as our student. This model exploits a Bidirectional LSTM to map input sequence X into a sequence of feature vectors, which accounts for the context from both directions. And sentence level tag information also achieved via a CRF layer. We reuse the learned word embeddings and CRF layer from the teacher model and keep it frozen during training.

3.2.1 Objectives

The student model is allowed to learn from ground-truth labels directly and surrogate ones labeled by the teacher model. The loss related to the above labels is as follows,

$$L(x^{(i)}) := -\log p(\mathbf{y} \mid x^{(i)}) \quad (3)$$

The objective function is the weighted sum of two terms defined above,

$$L_{Seq-all} = \frac{1}{N} \sum_{i=1}^{N} \lambda_1 L(x^{(i)}) + \lambda_2 L_{Seq-ce}(x^{(i)}), \quad (4)$$

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

To balance multi-task objectives for both simplicity and robust performance, the loss weights in (4) is tuned automatically using uncertainty weighting strategy proposed in (cite) by adding uncertainty

Architecture	Prec.	Rec.	F1
BERT-CRF	97.04	89.45	93.09
DatasetName	97.83	73.77	84.11
HyperparameterName	95.40	95.40	95.40
HyperparameterValue	93.75	78.95	85.71
MethodName	97.40	87.72	92.31
MetricName	98.08	98.08	98.08
MetricValue	86.67	76.47	81.25
TaskName	98.54	94.41	96.43
BiLSTM-CRF-Distill	73.51	54.60	62.66
DatasetName	50.00	9.84	16.44
HyperparameterName	70.89	64.37	67.47
HyperparameterValue	92.31	63.16	75.00
MethodName	70.25	50.30	58.62
MetricName	70.59	46.15	55.81
MetricValue	81.25	76.47	78.79
TaskName	78.29	72.66	75.37

Table 1: Results of two models on test set

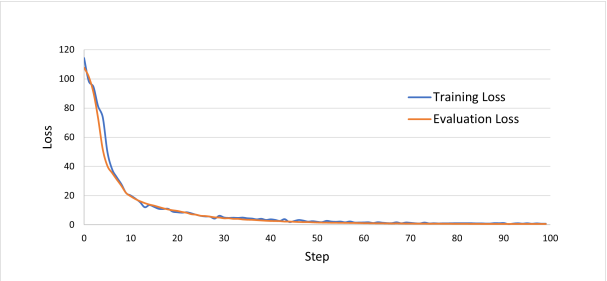

Figure 3: The training loss and evaluation loss of BERT (Teacher) model at each step.

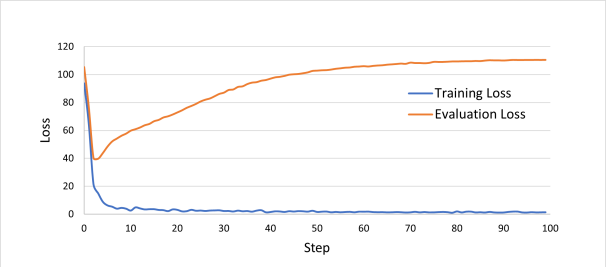

Figure 4: The training loss and evaluation loss of Bi-LSTM (Student) model at each step.

regularization terms for the weights:

$$L^{u}_{Seq-all} = L_{Seq-all} - \frac{1}{2}(\log \lambda_1 + \log \lambda_2) \quad (5)$$

3.2.2 Data Augmentation

Due to the label shortage in our training data, the unlabeled data is also fed into the teacher model (BERT-CRF) to construct additional surrogate label sequences by selecting the best Viterbi for distillation.

4 Experiments

We trained both models for 100 epochs. Their precision, recall, and F1 along with their fine-grained level entities are in Table 1. Their loss along training steps are in Figure 3 and Figure 4. We performed pairwise significant test of the two models and the results are in Table 2. We can say that BERT-CRF performed significantly better than BiLSTM-CRF-Distill since the p-value is below 0.05. The win ratio of BERT-CRF is 1.0 which means it beats BiLSTM-CRF-Distill in every bootstrapping, which fit our expectations since it is the teacher model.

5 Comparative Analysis

5.1 Quantitative Analysis

From Table 1, we can find that BERT-CRF outperforms the BiLSTM-CRF-Distill overall. For BERT-CRF, after we calculated the precision, recall, and F1 of various entities on the test set, we found that the precisions are relatively high while the recalls are generally low, especially in DatasetName, HyperparameterValue, and MetricValue. The reason here could be these labels are relatively few in the training set, and their data distribution is not enough to fully describe their entities. So although precision can be well learned, the model performance would be poor for unseen distributions. The F1 value has a better reflection on the overall performance of the model. The performance of BiLSTM-CRF-Distill in precision is also higher than that of recall. The main reason here is that the student still learns the distribution of the teacher, so this phenomenon is in line with our expectations. In addition, for the entities DatasetName that BERT-CRF performed the worst recall at, BiLSTM-CRF-Distill also achieved the lowest recall. This shows that the teacher has some deviation in the

	BERT-CRF	**BiLSTM-CRF**
win ratio	1.000	0.000
mean	0.996	0.954
median	0.996	0.954
95% confidence interval	[0.994, 0.997]	[0.950, 0.959]

Table 2: tie=0.000. BERT-CRF is superior with p value=0.000

distribution of the DatasetName entity, and the student even amplifies this deviation. However, other entities do not fully align with the trends in the teacher model. Strangely we found that BERT-CRF performed better in alphabetic entities (DatasetName, HyperparameterName, MethodName, MetricName, TaskName) than numerical entities (HyperparameterValue, MetricValue), while BiLSTM-CRF-Distill performed almost exactly the opposite. We don't know the exact reason here. The reason we suspect is that the teacher model itself has overfitting, and what the student learns is an over-fitted data. However, the proportion of number entities in the dataset is larger and easier to learn, so the degree of student fitting to this type of data is deeper. Overall we need to measure the fit of the model to the data in multiple dimensions when evaluating the model performance.

From Figure 3 and Figure 4, according to the decreasing trend of the loss both on the training set and the test set, despite the limited dataset size, the BERT-CRF model can still learn the distribution of entities. so both losses continue to decrease as the epoch increase. In the loss figure of BiLSTM-CRF-Distill, the loss of the training set decreases significantly faster than that of the test set, which indicates that the student fits the teacher well. This also shows that the data distribution learned by the teacher is relatively simple, so the learning of the student is fast. But the loss of the test set quickly starts to rebound, and the overfitting is serious. This shows that the difference between the actual data distribution and the teacher's label still exists, and the difference is relatively large.

5.2 Qualitative Analysis

We found that BERT-CRF can fit on our small dataset well, but there is obvious over-fitting. This is due to the difference between the distribution of the dataset and the distribution of actual entities due to the size of the training set. But this is also expected, and we believe that with the increase of entity labels in the dataset, the recall rate of the model would be higher and then achieve a better F1.

Our BiLSTM-CRF-Distill can fit the logistic distribution of the deep teacher model on a lighter and faster architecture, which is very helpful for practical applications. Although our distillation loss is relatively large in the current experiments, the loss probably would be lower in real data distribution. In fact, we believe that BiLSTM-CRF-Distill has learned some of the semantic representation of BERT. However, due to the scale of our dataset, the logistic distribution of the teacher is relatively naive. In the Figure 5, the BERT-CRF model predicts the sentence perfectly. BiLSTM-CRF-Distill predicts incorrectly on alphabetic entities MetricName, MethodName, and HyperparameterName, while numerical entities are all correct. This is consistent with the trend we see in Table 1. The numerical entities that are correctly predicted in this part have a larger contribution to the F1 of the BiLSTM-CRF-Distill compared to that of BERT-CRF.

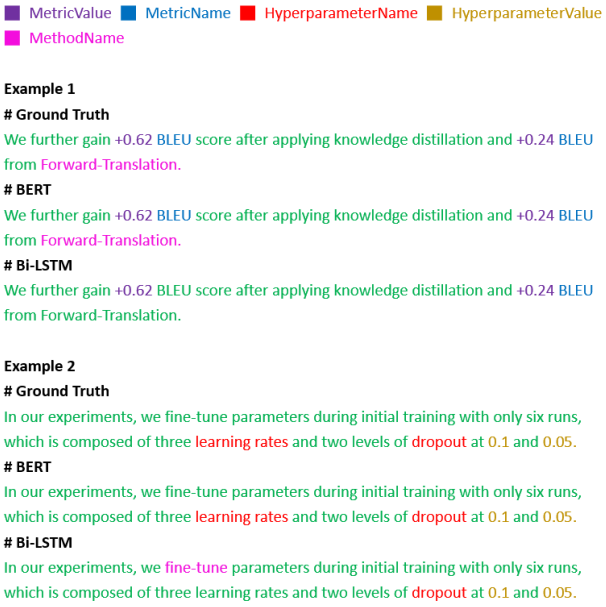

Figure 5: Examples of outputs from BERT-CRF and BiLSTM-CRF-Distill.

References

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional lstm-crf models for sequence tagging. *arXiv preprint arXiv:1508.01991*.

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

F´abio Souza, Rodrigo Nogueira, and Roberto Lotufo. 2019. Portuguese named entity recognition using bert-crf. arXiv:1909.10649.

Xuan Zhou, Xiao Zhang, Chenyang Tao, Junya Chen, Bing Xu, Wei Wang, and Jing Xiao. 2021. Multi-grained knowledge distillation for named entity recognition. In *Proceedings of the NAACL*, pages 5704–5716.

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE international conference on computer vision*, pages 19–27.