Boundary Smoothing for Named Entity Recognition

Anonymous ACL submission

Abstract

Neural named entity recognition (NER) models may easily encounter the over-confidence issue, which degrades the performance and calibration. Inspired by label smoothing and driven by the ambiguity of boundary annotation in NER engineering, we propose *boundary smoothing* as a regularization technique for span-based neural NER models. It re-assigns entity probabilities from annotated spans to the surrounding ones. Built on a simple but strong baseline, our model achieves results better than or competitive with previous state-of-the-art systems on eight well-known NER benchmarks.[1] Further empirical analysis suggests that boundary smoothing effectively mitigates over-confidence, improves model calibration, and brings flatter neural minima and more smoothed loss landscapes.

1 Introduction

Named entity recognition (NER) is one of the fundamental natural language processing (NLP) tasks with extensive investigations. As a common setting, an entity is regarded as correctly recognized only if its type and two boundaries exactly match the ground truth.

The annotation of boundaries is more ambiguous, error-prone, and raises more inconsistencies than entity types. For example, the CoNLL 2003 task contains four entity types (i.e., person, location, organization, miscellaneous), which are easy to distinguish between. However, the boundaries of a entity mention could be ambiguous, because of the "boundary words" (e.g., articles or modifiers). Considerable efforts are required to specify the "gold standard practice" case by case. Table 1 presents some examples from CoNLL 2003 Annotation Guidelines.[2] In addition, some studies have also reported that incorrect boundary is a major source of entity recognition error (Wang et al., 2019; Eberts and Ulges, 2020).

Text	Boundary words
[The [White House]$_{\text{ORG}}$]$_{\text{ORG}}$	Article
[The [Godfather]$_{\text{PER}}$]$_{\text{PER}}$	Article
[[Clinton]$_{\text{PER}}$ government]$_{\text{ORG}}$	Modifier
[Mr. [Harry Schearer]$_{\text{PER}}$]$_{\text{PER}}$	Person title
[[John Doe]$_{\text{PER}}$, Jr.]$_{\text{PER}}$	Name appositive

Table 1: Examples of CoNLL 2003 Annotation Guidelines and potential alternatives. The gold annotations are marked in blue [*], whereas the alternative annotations are in red [*].

Recently, span-based models have gained much popularity in NER studies, and achieved state-of-the-art (SOTA) results (Eberts and Ulges, 2020; Yu et al., 2020; Li et al., 2021). This approach typically enumerates all candidate spans and classifies them into entity types (including a "non-entity" type); the annotated spans are scarce and assigned with full probability to be an entity, whereas all other spans are assigned with zero probability. This creates noticeable *sharpness* between the classification targets of adjacent spans, and may thus plague the trainability of neural networks. In addition, empirical evidence shows that these models easily encounter the *over-confidence* issue, i.e., the confidence of a predicted entity is much higher than its correctness probability. This is a manifestation of miscalibration (Guo et al., 2017).

Inspired by label smoothing (Szegedy et al., 2016; Müller et al., 2019), we propose boundary smoothing as a regularization technique for span-based neural NER models. By explicitly re-allocating entity probabilities from annotated spans to the surrounding ones, boundary smoothing can effectively mitigate over-confidence, and result in consistently better performance.

[1] Our code will be publicly released.

[2] https://www-nlpir.nist.gov/related_projects/muc/proceedings/ne_task.html.

Specifically, our baseline employs the contextualized embeddings from a pretrained Transformer of base size (768 hidden size, 12 layers), and the biaffine decoder proposed by Yu et al. (2020). With boundary smoothing, our model outperforms previous SOTA on four English NER datasets (CoNLL 2003, OntoNotes 5, ACE 2004 and ACE 2005) and two Chinese datasets (Weibo NER and Resume NER), and achieves competitive results on other two Chinese datasets (OntoNotes 4 and MSRA). Such extensive experiments support the effectiveness and robustness of our proposed technique.

In addition, we show that boundary smoothing can help the trained NER models to preserve calibration, such that the produced confidences can better represent the precision rate of a predicted entity. This corresponds to the effect of label smoothing on the image classification task (Müller et al., 2019). Further, visualization results qualitatively suggest that boundary smoothing can lead to flatter solutions and more smoothed loss landscapes, which are typically associated with better generalization and trainability (Hochreiter and Schmidhuber, 1997; Li et al., 2018).

2 Related Work

Named Entity Recognition The mainstream NER systems are designed to recognize flat entities and based on a sequence tagging framework. Collobert et al. (2011) introduced the linear-chain conditional random field (CRF) into neural network-based sequence tagging models, which can explicitly encode the transition likelihoods between adjacent tags. Many researchers followed this work, and employed LSTM as the encoder. In addition, character-level representations are typically used for English tasks (Huang et al., 2015; Lample et al., 2016; Ma and Hovy, 2016; Chiu and Nichols, 2016), whereas lexicon information is helpful for Chinese NER (Zhang and Yang, 2018; Ma et al., 2020; Li et al., 2020a).

Nested NER allows a token to belong to multiple entities, which conflicts with the plain sequence tagging framework. Ju et al. (2018) proposed to use stacked LSTM-CRFs to predict from inner to outer entities. Straková et al. (2019) concatenated the BILOU tags for each token inside the nested entities, which allows the LSTM-CRF to work as for flat entities. Li et al. (2020b) reformulated nested NER as a machine reading comprehension task. Shen et al. (2021) proposed to recognize nested entities by the two-stage object detection method widely used in computer vision.

Recent years, a body of literature emerged on span-based models, which were compatible with both flat and nested entities, and achieved SOTA performance (Eberts and Ulges, 2020; Yu et al., 2020; Li et al., 2021). These models typically enumerate all possible candidate text spans and then classify each span into entity types. In this work, the biaffine model (Yu et al., 2020) is chosen and re-implemented with slight modifications as our baseline, because of its high performance and compatibility with boundary smoothing.

In addition, pretrained language models, also known as contextualized embeddings, were also widely introduced to NER models, and significantly boosted the model performance (Peters et al., 2018; Devlin et al., 2019). They are used in our baseline by default.

Label Smoothing Szegedy et al. (2016) proposed the label smoothing as a regularization technique to improve the accuracy of the Inception networks on ImageNet. By explicitly assigning a small probability to non-ground-truth labels, label smoothing can prevent the models from becoming too confident about the predictions, and thus improve generalization. It turned out to be a useful alternative to the standard cross entropy loss, and has been widely adopted to fight against the over-confidence (Zoph et al., 2018; Chorowski and Jaitly, 2017; Vaswani et al., 2017), improve the model calibration (Müller et al., 2019), and denoise incorrect labels (Lukasik et al., 2020).

Our proposed boundary smoothing applies the smoothing technique to entity boundaries, rather than labels. This is driven by the observation that entity boundaries are more ambiguous and inconsistent to annotate in NER engineering. To the best of our knowledge, this study is the first that focuses on the effect of smoothing regularization on NER models.

3 Methods

3.1 Biaffine Decoder

A neural network-based NER model typically encodes the input tokens to a sequence of representations $x = x_1, x_2, \dots, x_T$ of length T, and then decodes these representations to task outputs, i.e., a list of entities specified by types and boundaries.

We follow Yu et al. (2020) and use the biaffine decoder. Specifically, the representations x are separately affined by two feedforward networks, resulting in two representations $h^s \in \mathbb{R}^{T\times d}$ and $h^e \in \mathbb{R}^{T\times d}$, which correspond to the start and end positions of spans. For c entity types (a "non-entity" type included), given a span starting at the i-th token and ending at the j-th token, a scoring vector $r_{ij} \in \mathbb{R}^c$ can be computed as:

$$r_{ij} = (h_i^s)^{\mathrm{T}} U h_j^e + W(h_i^s \oplus h_j^e \oplus w_{j-i}) + b, \quad (1)$$

where $w_{j-i} \in \mathbb{R}^{d_w}$ is the $(j-i)$-th width embedding from a dedicated learnable matrix; $U \in \mathbb{R}^{d\times c\times d}$, $W \in \mathbb{R}^{c\times(2d+d_w)}$ and $b \in \mathbb{R}^c$ are learnable parameters. r_{ij} is then fed into a softmax layer:

$$\hat{y}_{ij} = \mathrm{softmax}(r_{ij}), \quad (2)$$

which yields the predicted probabilities over all entity types.

The ground truth $y_{ij} \in \mathbb{R}^c$ is an one-hot encoded vector, with value being 1 if the index corresponds with the annotated entity type, and 0 otherwise. Thus, the model can be optimized by the standard cross entropy loss for all candidate spans:

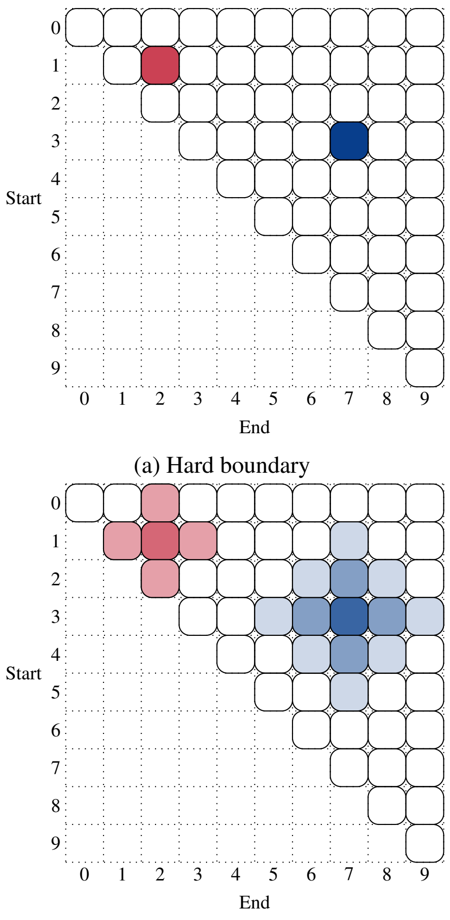
$$L_{\mathrm{CE}} = -\sum_{0\le i\le j<T} y_{ij}^{\mathrm{T}} \log(\hat{y}_{ij}). \quad (3)$$

In the inference time, the spans predicted to be "non-entity" are first discarded, and the remaining ones are ranked by their predictive confidences. Spans with lower confidences would also be discarded if they clash with the boundaries of spans with higher confidences. Refer to Yu et al. (2020) for more details.

3.2 Boundary Smoothing

Figure 1a visualizes the ground truth y_{ij} for an example sentence with two annotated entities. The valid candidate spans cover the upper triangular area of the matrix. In existing NER models, the annotated boundaries are considered to be absolutely reliable. Hence, each annotated span is assigned with the full probability to be an entity, whereas all unannotated spans are assigned with zero probability. We refer to this probability allocation as *hard boundary*, which is, however, probably not the best choice.

As aforementioned, the entity boundaries may be ambiguous and inconsistent, so the spans surrounding an annotated one deserve a small probability to be an entity. Figure 1b visualizes $\tilde{y}_{ij}$, the boundary smoothing version of y_{ij}. Specifically, given an annotated entity, a portion of probability ϵ is assigned to its surrounding spans, and the remaining probability $1-\epsilon$ is assigned to the originally annotated span. With smoothing size D, all the spans with Manhattan distance d $(d \le D)$ to the annotated entity equally share probability ϵ/D. After such entity probability re-allocation, any remaining probability of a span is assigned to be "non-entity". We refer to this as *smoothed boundary*.



(a) Hard boundary

(b) Smoothed boundary

Figure 1: An example of hard and smoothed boundaries. The example sentence has ten tokens and two entities of spans (1, 2) and (3, 7), colored in red and blue, respectively. The first subfigure presents the entity recognition targets of hard boundaries. The second subfigure presents the corresponding targets of smoothed boundaries, where the span (1, 2) is smoothed by a size of 1, and the span (3, 7) is smoothed by a size of 2.

Thus, the biaffine model can be optimized by the boundary-smoothing regularized cross entropy loss:

$$L_{\mathrm{BS}} = -\sum_{0\le i\le j<T} \tilde{y}_{ij}^{\mathrm{T}} \log(\hat{y}_{ij}). \quad (4)$$

Empirically, the positive samples (i.e., ground-truth entities) are sparsely distributed over the candidate spans. For example, the CoNLL 2003 dataset has about 35 thousand entities, which represent only 0.93% in the 3.78 million candidate spans. By explicitly assigning probability to surrounding spans, boundary smoothing prevents the model from concentrating all probability mass on the scarce positive samples. This intuitively helps alleviate over-confidence.

In addition, hard boundary presents noticeable sharpness between the classification targets of positive spans and surrounding ones, although they share similar contextualized representations. Smoothed boundary provides more continuous targets across spans, which are conceptually more compatible with the inductive bias of neural networks that prefers continuous solutions (Hornik et al., 1989).

4 Experiments

4.1 Experimental Settings

Datasets We use four English NER datasets: CoNLL 2003 (Tjong Kim Sang and Veenstra, 1999), OntoNotes 5[3], ACE 2004[4] and ACE 2005[5]; and four Chinese NER datasets: OntoNotes 4[6], MSRA (Levow, 2006), Weibo NER (Peng and Dredze, 2015) and Resume NER (Zhang and Yang, 2018). Among them, ACE 2004 and ACE 2005 are nested NER tasks, and the others are flat tasks.

Hyperparameters For English corpora, we use RoBERTa (Liu et al., 2019) followed by a BiLSTM layer to produce the contextualized representations. For Chinese, we choose the BERT pretrained with whole word masking (Cui et al., 2019). The BiLSTM has one layer and 200 hidden size with dropout rate of 0.5. The biaffine decoder follows Yu et al. (2020), with the affine layers of hidden size 150 and dropout rate 0.2. We additionally introduce a span width embedding of size 25. Note that the pretrained language models are all of the `base` size (768 hidden size, 12 layers), and the model is free of any additional auxiliary embeddings; this configuration is relatively simple, compared with those in related work.

The boundary smoothing parameter ϵ is selected in $\{0.1, 0.2, 0.3\}$; smoothing size D is selected in $\{1, 2\}$.

All the models are trained by the AdamW optimizer (Loshchilov and Hutter, 2018) with a gradient clipping at L2-norm of 5.0 (Pascanu et al., 2013). The models are trained for 50 epochs with batch size of 48. The learning rate is searched between 1e-3 and 3e-3 on the randomly initialized weights, and between 8e-6 and 3e-5 on the pretrained weights; a scheduler of linear warmup in the first 20% steps followed by linear decay is applied.

Evaluation A predicted entity is considered correct if its type and boundaries exactly match the ground truth. Hyperparameters are tuned according to the F_1 scores on the development set, and the evaluation metrics (precision, recall, F_1 score) are reported on the testing set.

4.2 Main Results

Table 2 presents the evaluation results on four English datasets, in which CoNLL 2003 and OntoNotes 5 are flat NER corpora, whereas ACE 2004 and ACE 2005 contains a high proportion of nested entities. Compared with previous SOTA systems, our simple baseline (RoBERTa-base + BiLSTM + Biaffine) achieves on-par or slightly inferior performance. Provided the strong baseline, our experiments show that boundary smoothing can effectively and consistently boost the F_1 score of entity recognition across different datasets. With the help of boundary smoothing, our model outperforms the best of the previous SOTA systems by a magnitude from 0.2 to 0.5 percentages.

Table 3 presents the results on four Chinese datasets, which are all flat NER corpora. Again, boundary smoothing consistently improves model performance against the baseline (BERT-base-wwm + BiLSTM + Biaffine) across all datasets. In addition, our model outperforms previous SOTA by 2.16 and 0.55 percentages on Weibo and Resume NER datasets, and achieves comparable F_1 scores on OntoNotes 4 and MSRA. Note that almost all previous systems solve these tasks within a sequence tagging framework; in contrast, this work is among the first to introduce a span-based approach to Chinese NER tasks and establish SOTA results.

[3] `https://catalog.ldc.upenn.edu/LDC2013T19`; Data splits follow Pradhan et al. (2013).

[4] `https://catalog.ldc.upenn.edu/LDC2005T09`; Data splits follow Lu and Roth (2015).

[5] `https://catalog.ldc.upenn.edu/LDC2006T06`; Data splits follow Lu and Roth (2015).

[6] `https://catalog.ldc.upenn.edu/LDC2011T03`; Data splits follow Che et al. (2013).

CoNLL 2003			
Model	Prec.	Rec.	F1
Lample et al. (2016)	–	–	90.94
Chiu and Nichols (2016)†	91.39	91.85	91.62
Peters et al. (2018)	–	–	92.22
Akbik et al. (2018)†	–	–	93.07
Devlin et al. (2019)	–	–	92.8
Straková et al. (2019)†	–	–	93.38
Wang et al. (2019)†	–	–	93.43
Li et al. (2020b)	92.33	94.61	93.04
Yu et al. (2020)†	93.7	93.3	93.5
Baseline	92.93	94.03	93.48
Baseline + BS	93.61	93.68	**93.65**

OntoNotes 5			
Model	Prec.	Rec.	F1
Chiu and Nichols (2016)	86.04	86.53	86.28
Li et al. (2020b)	92.98	89.95	91.11
Yu et al. (2020)	91.1	91.5	91.3
Baseline	90.31	92.13	91.21
Baseline + BS	91.75	91.74	**91.74**

ACE 2004			
Model	Prec.	Rec.	F1
Katiyar and Cardie (2018)	73.6	71.8	72.7
Straková et al. (2019)†	–	–	84.40
Li et al. (2020b)	85.05	86.32	85.98
Yu et al. (2020)	87.3	86.0	86.7
Shen et al. (2021)	87.44	87.38	87.41
Baseline	86.67	88.42	87.54
Baseline + BS	88.43	87.53	**87.98**

ACE 2005			
Model	Prec.	Rec.	F1
Katiyar and Cardie (2018)	70.6	70.4	70.5
Straková et al. (2019)†	–	–	84.33
Li et al. (2020b)	87.16	86.59	86.88
Yu et al. (2020)	85.2	85.6	85.4
Shen et al. (2021)	86.09	87.27	86.67
Baseline	84.29	88.97	86.56
Baseline + BS	86.25	88.07	**87.15**

Table 2: Results of English named entity recognition. BS means boundary smoothing. † means that the model is trained with both the training and development splits.

OntoNotes 4			
Model	Prec.	Rec.	F1
Zhang and Yang (2018)	76.35	71.56	73.88
Ma et al. (2020)	83.41	82.21	82.81
Li et al. (2020a)	–	–	81.82
Li et al. (2020b)	82.98	81.25	82.11
Chen and Kong (2021)	79.25	80.66	79.95
Wu et al. (2021)	–	–	82.57
Baseline	82.79	81.27	82.03
Baseline + BS	81.65	84.03	**82.83**

MSRA			
Model	Prec.	Rec.	F1
Zhang and Yang (2018)	93.57	92.79	93.18
Ma et al. (2020)	95.75	95.10	95.42
Li et al. (2020a)	–	–	96.09
Li et al. (2020b)	96.18	95.12	95.75
Wu et al. (2021)	–	–	96.24
Baseline	95.82	95.78	95.80
Baseline + BS	96.37	96.15	**96.26**

Weibo NER			
Model	Prec.	Rec.	F1
Zhang and Yang (2018)	–	–	58.79
Ma et al. (2020)	–	–	70.50
Li et al. (2020a)	–	–	68.55
Chen and Kong (2021)	–	–	70.14
Wu et al. (2021)	–	–	70.43
Baseline	68.65	74.40	71.41
Baseline + BS	70.16	75.36	**72.66**

Resume NER			
Model	Prec.	Rec.	F1
Zhang and Yang (2018)	94.81	94.11	94.46
Ma et al. (2020)	96.08	96.13	96.11
Li et al. (2020a)	–	–	95.86
Wu et al. (2021)	–	–	95.98
Baseline	95.81	96.87	96.34
Baseline + BS	96.63	96.69	**96.66**

Table 3: Results of Chinese named entity recognition. BS means boundary smoothing.

In five out of the above eight datasets, integrating boundary smoothing significantly increases the precision rate with a slight drop in the recall, resulting in a better overall F_1 score. This is consistent with our expectation, because boundary smoothing discourages over-confidence when recognizing entities, which implicitly leads the model to establish a more critical threshold to admit entities.

Given the use of well pretrained language models, most of the performance gains are relatively marginal. However, boundary smoothing can work effectively and consistently for different languages and datasets. In addition, it is easy to implement and integrate into any span-based neural NER models, with almost no side effects.

4.3 Ablation Studies

We perform ablation studies on CoNLL 2003, ACE 2005 and Resume NER datasets (covering flat/nested and English/Chinese datasets), to evaluate the effects of boundary smoothing parameter ϵ and D, as well as other components of our NER system.

Boundary Smoothing Parameters We train the model with ϵ in $\{0.1, 0.2, 0.3\}$ and D in $\{1, 2\}$; the corresponding results are reported in Table 4. Most combinations of the two hyperparameters can achieve higher F_1 scores than the baseline, which

	CoNLL 2003	ACE 2005	Resume NER
Baseline	93.48	86.56	96.34
BS ($\epsilon = 0.1$, $D = 1$)	93.50	86.65	96.63
BS ($\epsilon = 0.2$, $D = 1$)	93.56	86.96	**96.66**
BS ($\epsilon = 0.3$, $D = 1$)	**93.65**	86.81	96.50
BS ($\epsilon = 0.1$, $D = 2$)	93.45	**87.15**	96.33
BS ($\epsilon = 0.2$, $D = 2$)	93.39	86.99	96.62
BS ($\epsilon = 0.3$, $D = 2$)	93.57	86.71	96.28
LS ($\alpha = 0.1$)	93.43	86.31	96.31
LS ($\alpha = 0.2$)	93.37	86.17	96.38
LS ($\alpha = 0.3$)	93.26	85.65	96.26

Table 4: Ablation studies of smoothing parameters. F_1 scores are reported. BS and LS mean boundary smoothing and label smoothing, respectively.

suggests the robustness of boundary smoothing. On the other hand, the best smoothing parameters are different across datasets. Hence, if the best performance is desired for a new NER task in practice, hyperparameter tuning would be necessary.

Label Smoothing We replace boundary smoothing with label smoothing in the span classifier. Label smoothing cannot improve, or may even impair the performance of the model, compared with the baseline (see Table 4). As aforementioned, we hypothesize that the semantic differences between the typical entity types are quite clear, so it is ineffective to smooth between them.

Pretrained Language Models We test if better pretrained language models can further improve the performance. For English datasets, we use RoBERTa of `large` size (1024 hidden size, 24 layers), and the F_1 scores increase by 0.12 and 0.87 percentages for CoNLL 2003 and ACE 2005, respectively. For Chinese, we use MacBERT (Cui et al., 2020) of `base` and `large` sizes, and both improve the F_1 score by 0.09 percentage on Resume NER (see Table 5).

Note that boundary smoothing contributes to the F_1 scores by 0.17, 0.59 and 0.32 percentages on these three datasets, which are roughly comparable to the magnitudes by switching the pretrained language model from `base` size to `large` size.

BiLSTM Layer We remove the BiLSTM layer, directly feeding the output of pretrained language model into the biaffine decoder. Absence of the BiLSTM layer will result in drops of the F_1 scores by 0.35, 0.57 and 0.1 percentages on the three datasets (see Table 5).

	CoNLL 2003	ACE 2005	Resume NER
Baseline + BS	93.65	87.15	96.66
BS w/ RoBERTa-large	**93.77**	**88.02**	
BS w/ MacBERT-base			**96.75**
BS w/ MacBERT-large			**96.75**
BS w/o BiLSTM	93.30	86.58	96.56

Table 5: Ablation studies of model structure. F_1 scores are reported. BS and LS mean boundary smoothing and label smoothing, respectively.

5 Further In-Depth Analysis

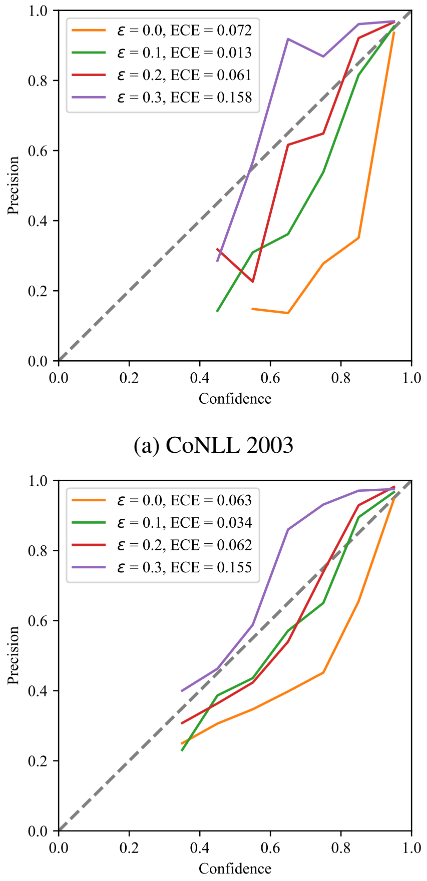
5.1 Over-Confidence and Entity Calibration

The model performance (evaluated by, e.g., accuracy or F_1 score) is certainly important. However, the *confidences* of model predictions are also of interest in many applications. For example, when it requires the predicted entities to be highly reliable (i.e., precision is of more priority than recall), we may filter out the entities with confidences lower than a specific threshold.

However, Guo et al. (2017) have indicated that modern neural networks are poorly calibrated, and typically over-confident with their predictions. By calibration, they mean the extent to which the prediction confidences produced by a model can represent the true correctness probability. We find neural NER models also easy to become miscalibrated and over-confident. We observe that, with the standard cross entropy loss, both the development loss and F_1 score increase in the later training stage, which goes against the common perception that the loss and F_1 score should change in the opposite directions. This phenomenon is similar to the disconnect between negative likelihood and accuracy in image classification described by Guo et al. (2017). We suppose that the model becomes over-confident with its predictions, including the incorrect ones, which contributes to the increase of loss (see Appendix A for more details).

To formally investigate the over-confidence issue, we plot the reliability diagrams and calculate expected calibration error (ECE). In brief, for an NER model, we group all the predicted entities by the associated confidences into ten bins, and then calculate the precision rate for each bin. If the model is well calibrated, the precision rate should be close to the confidence level for each bin (see Appendix B for more details).

Figure 2 compares the reliability diagrams and



(a) CoNLL 2003

(b) OntoNotes 5

Figure 2: Reliability diagram of recognized entities on CoNLL 2003 and OntoNotes 5. Results are computed on ten bins.

ECEs between models with different smoothness ϵ on CoNLL 2003 and OntoNotes 5. For the baseline model ($\epsilon = 0$), the precision rates are much lower than corresponding confidence levels, suggesting significant over-confidence. By introducing boundary smoothing and increasing the smoothness ϵ, the over-confidence is gradually mitigated, and shifted to under-confidence ($\epsilon = 0.3$). In general, the model presents best reliability diagrams when ϵ is 0.1 or 0.2. In addition, the ECEs of the baseline model are 0.072 and 0.063 on CoNLL 2003 and OntoNotes 5, respectively; with ϵ of 0.1, the ECEs are reduced to 0.013 and 0.034.

In conclusion, boundary smoothing can prevent the model from becoming over-confident with the predicted entities, and result in better calibration. In addition, as mentioned previously, spans with lower confidences are discarded if they clash with those of higher confidences when decoding. With the better calibration, the model can obtain a very marginal but consistent increase in the F_1 score.

5.2 Loss Landscape Visualization

How does boundary smoothing improve the model performance? We originally conjectured that boundary smoothing can de-noise the inconsistently annotated entity boundaries (Lukasik et al., 2020), but failed to find enough evidence – the performance improvement did not significantly increase when we injected boundary noises into the training data.[7]

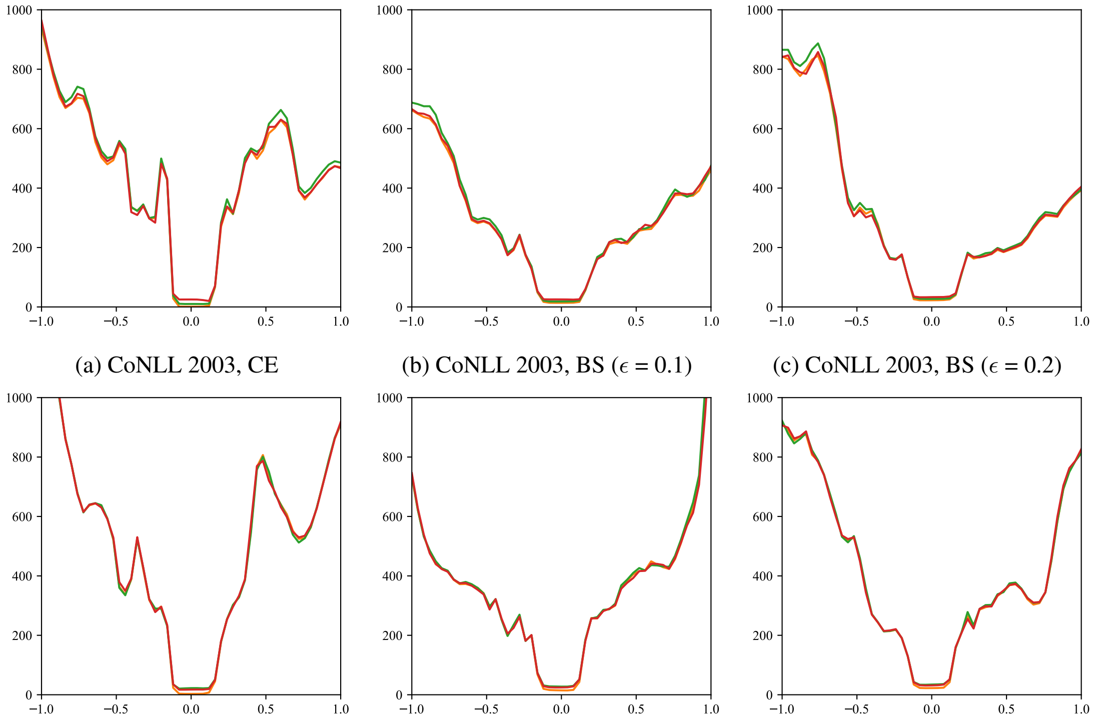
As aforementioned, positive samples are very sparse among the candidate spans. Without boundary smoothing, the annotated spans are regarded to be entities with full probability, whereas all other spans are assigned with zero probability. This creates noticeable *sharpness* between the targets of the annotated spans and surrounding ones, although their neural representations are similar. Boundary smoothing re-allocates the entity probabilities across contiguous spans, which mitigates the sharpness and results in more continuous targets. Conceptually, such targets are more compatible with the inductive bias of neural networks that prefers continuous solutions (Hornik et al., 1989).

Li et al. (2018) have shown that residual connections and well-tuned hyperparameters (e.g., learning rate, batch size) can produce flatter minima and less chaotic loss landscapes, which account for the better generalization and trainability. Their findings provide important insights into the geometric properties of non-convex neural loss functions.

Figure 3 visualizes the loss landscapes for models with different smoothness ϵ on CoNLL 2003 and OntoNotes 5, following Li et al. (2018). In short, for a trained model, a direction of the parameters is randomly sampled, normalized and fixed, and the loss landscape is computed by sampling over this direction (refer to Appendix C for more details).

The visualization results qualitatively show that, the solutions found by the standard cross entropy are relatively sharp, whereas boundary smoothing can help arrive at flatter minima. As many theoretical studies regard the flatness as a promising predictor for model generalization (Hochreiter and Schmidhuber, 1997; Jiang et al., 2019), this result may explain why boundary smoothing can improve the model performance. In addition, boundary smoothing is associated with more smoothed land-

[7] On the other hand, this cannot rule out the de-noising effect of boundary smoothing, because the synthesized boundary noises are differently distributed from the real noises.



(a) CoNLL 2003, CE

(b) CoNLL 2003, BS ($\epsilon = 0.1$)

(c) CoNLL 2003, BS ($\epsilon = 0.2$)

(d) OntoNotes 5, CE

(e) OntoNotes 5, BS ($\epsilon = 0.1$)

(f) OntoNotes 5, BS ($\epsilon = 0.2$)

Figure 3: Visualization of loss landscapes on CoNLL 2003 and OntoNotes 5. Training, development and testing losses are in orange, green and red, respectively. CE and BS mean cross entropy and boundary smoothing, respectively.

scapes – the surrounding local minima are small, shallow, and thus easy for the optimizer to escape. Intuitively, such geometric property suggests that the underlying loss functions are easier to train (Li et al., 2018).

We believe that the sharpness in the span-based NER targets is probably the reason for the sharp and chaotic loss landscape. Boundary smoothing can effectively mitigate the sharpness, and result in loss landscapes of better generalization and trainability.

6 Conclusion

In this study, we propose boundary smoothing as a regularization technique for span-based neural NER models. Boundary smoothing re-assigns entity probabilities from annotated spans to the surrounding ones. It can be easily integrated into any span-based neural NER systems, but consistently bring improved performance. Built on a simple but strong baseline (a `base`-sized pretrained language model followed by a BiLSTM layer, and the biaffine decoder), our model achieves SOTA results on eight well-known NER benchmarks, covering English and Chinese, flat and nested NER tasks.

In addition, experimental results show that boundary smoothing leads to less over-confidence, better model calibration, flatter neural minima and more smoothed loss landscapes. These properties plausibly explain the performance improvement. Our findings shed light on the effects of smoothing regularization technique in the NER task.

Acknowledgements

References

Alan Akbik, Duncan Blythe, and Roland Vollgraf. 2018. Contextual string embeddings for sequence labeling. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1638–1649, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Wanxiang Che, Mengqiu Wang, Christopher D. Manning, and Ting Liu. 2013. Named entity recognition with bilingual constraints. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 52–62, Atlanta, Georgia. Association for Computational Linguistics.

Chun Chen and Fang Kong. 2021. Enhancing entity boundary detection for better Chinese named entity recognition. In *Proceedings of the 59th Annual*

Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers), pages 20–25, Online. Association for Computational Linguistics.

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.

Jan Chorowski and Navdeep Jaitly. 2017. Towards better decoding and language model integration in sequence to sequence models. In *INTERSPEECH 2017*, pages 523–527.

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

Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, Shijin Wang, and Guoping Hu. 2020. Revisiting pre-trained models for Chinese natural language processing. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 657–668, Online. Association for Computational Linguistics.

Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, Ziqing Yang, Shijin Wang, and Guoping Hu. 2019. Pre-training with whole word masking for Chinese BERT. *arXiv preprint arXiv:1906.08101*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Markus Eberts and Adrian Ulges. 2020. Span-based joint entity and relation extraction with Transformer pre-training. In *Proceedings of the 24th European Conference on Artificial Intelligence*, Santiago de Compostela, Spain.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. 2017. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1321–1330. PMLR.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Flat minima. *Neural Computation*, 9(1):1–42.

Kurt Hornik, Maxwell Stinchcombe, and Halbert White. 1989. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366.

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

Yiding Jiang, Behnam Neyshabur, Hossein Mobahi, Dilip Krishnan, and Samy Bengio. 2019. Fantastic generalization measures and where to find them. *arXiv preprint arXiv:1912.02178*.

Meizhi Ju, Makoto Miwa, and Sophia Ananiadou. 2018. A neural layered model for nested named entity recognition. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1446–1459, New Orleans, Louisiana. Association for Computational Linguistics.

Arzoo Katiyar and Claire Cardie. 2018. Nested named entity recognition revisited. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 861–871, New Orleans, Louisiana. Association for Computational Linguistics.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 260–270, San Diego, California. Association for Computational Linguistics.

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*, pages 108–117, Sydney, Australia. Association for Computational Linguistics.

Fei Li, ZhiChao Lin, Meishan Zhang, and Donghong Ji. 2021. A span-based model for joint overlapped and discontinuous named entity recognition. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4814–4828, Online. Association for Computational Linguistics.

Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein. 2018. Visualizing the loss landscape of neural nets. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 6391–6401.

Xiaonan Li, Hang Yan, Xipeng Qiu, and Xuanjing Huang. 2020a. FLAT: Chinese NER using flat-lattice transformer. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6836–6842, Online. Association for Computational Linguistics.

Xiaoya Li, Jingrong Feng, Yuxian Meng, Qinghong Han, Fei Wu, and Jiwei Li. 2020b. A unified MRC framework for named entity recognition. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5849–5859, Online. Association for Computational Linguistics.

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

Ilya Loshchilov and Frank Hutter. 2018. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Wei Lu and Dan Roth. 2015. Joint mention extraction and classification with mention hypergraphs. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 857–867, Lisbon, Portugal. Association for Computational Linguistics.

Michal Lukasik, Srinadh Bhojanapalli, Aditya Menon, and Sanjiv Kumar. 2020. Does label smoothing mitigate label noise? In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 6448–6458. PMLR.

Ruotian Ma, Minlong Peng, Qi Zhang, Zhongyu Wei, and Xuanjing Huang. 2020. Simplify the usage of lexicon in Chinese NER. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5951–5960, Online. Association for Computational Linguistics.

Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1064–1074, Berlin, Germany. Association for Computational Linguistics.

Rafael Müller, Simon Kornblith, and Geoffrey E Hinton. 2019. When does label smoothing help? In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. 2013. On the difficulty of training recurrent neural networks. In *International Conference on Machine Learning*, pages 1310–1318. PMLR.

Nanyun Peng and Mark Dredze. 2015. Named entity recognition for Chinese social media with jointly trained embeddings. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 548–554, Lisbon, Portugal. Association for Computational Linguistics.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

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 *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 143–152, Sofia, Bulgaria. Association for Computational Linguistics.

Yongliang Shen, Xinyin Ma, Zeqi Tan, Shuai Zhang, Wen Wang, and Weiming Lu. 2021. Locate and label: A two-stage identifier for nested named entity recognition. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2782–2794, Online. Association for Computational Linguistics.

Jana Straková, Milan Straka, and Jan Hajic. 2019. Neural architectures for nested NER through linearization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5326–5331, Florence, Italy. Association for Computational Linguistics.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2818–2826.

Erik F. Tjong Kim Sang and Jorn Veenstra. 1999. Representing text chunks. In *Ninth Conference of the European Chapter of the Association for Computational Linguistics*, pages 173–179, Bergen, Norway. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

Zihan Wang, Jingbo Shang, Liyuan Liu, Lihao Lu, Jiacheng Liu, and Jiawei Han. 2019. CrossWeigh: Training named entity tagger from imperfect annotations. 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)*, pages 5154–5163, Hong Kong, China. Association for Computational Linguistics.

Shuang Wu, Xiaoning Song, and Zhenhua Feng. 2021. MECT: Multi-metadata embedding based cross-transformer for Chinese named entity recognition.

In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1529–1539, Online. Association for Computational Linguistics.

Juntao Yu, Bernd Bohnet, and Massimo Poesio. 2020. Named entity recognition as dependency parsing. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6470–6476, Online. Association for Computational Linguistics.

Yue Zhang and Jie Yang. 2018. Chinese NER using lattice LSTM. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1554–1564, Melbourne, Australia. Association for Computational Linguistics.

Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V Le. 2018. Learning transferable architectures for scalable image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8697–8710.

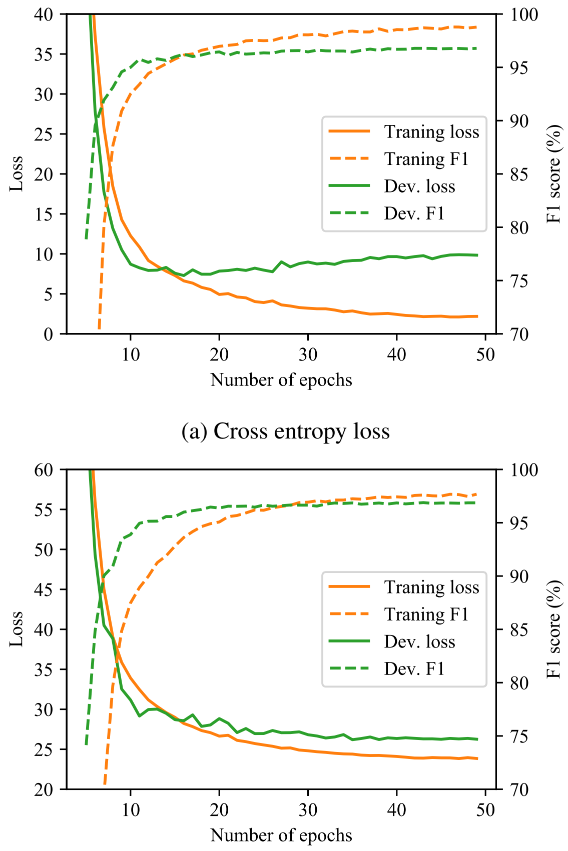
A Disconnect between Development Loss and F_1 Score

For most machine learning tasks, the desired metric (e.g., accuracy or F_1 score) is non-differentiable and thus cannot be optimized via back-propagation. The loss, on the other hand, is a designed differentiable proxy such that minimizing it can increase the original metric.

However, as illustrated in Figure 4a, when training an NER model by the standard cross entropy loss, although the development F_1 score keeps increasing throughout, the development loss also increases in the later stage (e.g., after ten epochs) of the training process. Guo et al. (2017) describe this phenomenon as a disconnect – the neural network overfits to the loss without overfitting to the metric. They regard this as indirect evidence for miscalibration.

One plausible explanation is that in the later training stage, the model becomes too confident with its predicted outcomes, including both the correct and incorrect ones. Therefore, although slightly more spans are correctly classified on the development set (as the F_1 score increases), a small portion of incorrectly classified spans is assigned with much more confidence and contributes to the increase of loss.

Figure 4b presents the curves for boundary smoothing loss. The development loss decreases throughout the training process, opposite to the increasing F_1 score. This result suggests that boundary smoothing can help mitigate over-confidence.



(a) Cross entropy loss

(b) Boundary smoothing loss (ϵ=0.2, D=1)

Figure 4: Training/development losses and F_1 scores of models with cross entropy loss and boundary smoothing loss on CoNLL 2003. Both the cross entropy loss and corresponding F_1 score on the development set experience an ascending trend after about ten epochs, suggesting the existence of over-confidence. However, the boundary smoothing loss on the development set keeps decreasing through the whole training process.

B Reliability Diagrams and Expected Calibration Error

We generally follow Guo et al. (2017)'s approach to plot reliability diagrams and calculate expected calibration error (ECE).

Given an NER dataset and a model trained on it, denote the gold and predicted entity sets as $\mathcal{E}$ and $\hat{\mathcal{E}}$, respectively; the model produces a confidence $\hat{p}_e$ for each entity $e \in \hat{\mathcal{E}}$. With K confidence interval bins, the predicted entities are grouped such that those with confidences falling into the k-th bin constitute a subset:

$$\hat{\mathcal{E}}_k = \left\{ e \mid e \in \hat{\mathcal{E}}, \hat{p}_e \in \left(\frac{k-1}{K}, \frac{k}{K} \right] \right\}.$$

The precision rate (equivalent to the accuracy with regard to a predicted set) of k-th group $\hat{\mathcal{E}}_k$ is:

$$\mathrm{Prec}_k = \frac{|\hat{\mathcal{E}}_k \cap \mathcal{E}|}{|\hat{\mathcal{E}}_k|},$$

and the corresponding average confidence is:

$$\mathrm{Conf}_k = \frac{\sum_{e \in \hat{\mathcal{E}}_k} \hat{p}_e}{|\hat{\mathcal{E}}_k|}.$$

The reliability diagrams plot Prec_k against Conf_k for $k = 1, 2, \ldots, K$. ECE is estimated by the weighted average of absolute difference between Prec_k and Conf_k:

$$\mathrm{ECE} = \sum_{k=1}^{K} \frac{|\hat{\mathcal{E}}_k|}{|\hat{\mathcal{E}}|} \cdot \left| \mathrm{Prec}_k - \mathrm{Conf}_k \right|$$

By definition, a perfectly calibrated model will have $\mathrm{Prec}_k = \mathrm{Conf}_k$ for $k = 1, 2, \ldots, K$. In this case, the reliability diagrams should lie along the identity line, and ECE equals to 0.

C Loss Landscape Visualization

We generally follow Li et al. (2018)'s approach to visualize the loss landscape.

Given a trained model of parameters $\theta^\star$, we sample a random direction δ from a normal distribution, and rescale it by:

$$\delta_i \leftarrow \frac{||\theta_i^\star||}{||\delta_i||} \delta_i,$$

where δ_i is the i-th weight of δ.[8] On a data set/split $\mathcal{D}$, the loss landscape plots the function:

$$f(\alpha) = L(\mathcal{D}; \theta^\star + \alpha\delta),$$

where $L(\mathcal{D}; \theta)$ is the average loss value (in the evaluation mode) on $\mathcal{D}$ if the model takes parameters of θ. In practice, we evenly sample 51 points in the interval $[-1, 1]$ for α, and plot the loss values against α.

[8] Li et al. (2018) use filter-wise normalization for convolutional networks, whereas our models have no convolutional layers, so we simplify it as weight-wise normalization.