Dice Loss for Data-imbalanced NLP Tasks

Xiaofei Sun[1*], **Xiaoya Li**[1*], **Yuxian Meng**[1],
Junjun Liang[1], **Fei Wu**[2] and **Jiwei Li**[1]
[1] ShannonAI
[2] Department of Computer Science and Technology, Zhejiang University
{xiaofei sun, xiaoya li, yuxian meng, jiwei li}@shannonai.com, wufei@zju.edu.cn

Abstract

Many NLP tasks such as tagging and machine reading comprehension are faced with the severe data imbalance issue: negative examples significantly outnumber positive examples, and the huge number of background examples (or easy-negative examples) overwhelms the training. The most commonly used cross entropy (CE) criteria is actually an accuracy-oriented objective, and thus creates a discrepancy between training and test: at training time, each training instance contributes equally to the objective function, while at test time F1 score concerns more about positive examples.

In this paper, we propose to use dice loss in replacement of the standard cross-entropy objective for data-imbalanced NLP tasks. Dice loss is based on the Sørensen–Dice coefficient (Sorensen, 1948) or Tversky index (Tversky, 1977), which attaches similar importance to false positives and false negatives, and is more immune to the data-imbalance issue. To further alleviate the dominating influence from easy-negative examples in training, we propose to associate training examples with dynamically adjusted weights to deemphasize easy-negative examples. Theoretical analysis shows that this strategy narrows down the gap between the F1 score in evaluation and the dice loss in training.

With the proposed training objective, we observe significant performance boost on a wide range of data imbalanced NLP tasks. Notably, we are able to achieve SOTA results on CTB5, CTB6 and UD1.4 for the part of speech tagging task; SOTA results on CoNLL03, OntoNotes5.0, MSRA and OntoNotes4.0 for the named entity recognition task; along with competitive results on the tasks of machine reading comprehension and paraphrase identification [1]

[1] Xiaoya and Xiaofei contribute equally to this work.

Task	# neg	# pos	ratio
CoNLL03 NER	170K	34K	4.98
OntoNotes5.0 NER	1.96M	239K	8.18
SQuAD 1.1 (Rajpurkar et al., 2016)	10.3M	175K	55.9
SQuAD 2.0 (Rajpurkar et al., 2018)	15.4M	188K	82.0
QUOREF (Dasigi et al., 2019)	6.52M	38.6K	169

Table 1: Number of positive and negative examples and their ratios for different data-imbalanced NLP tasks.

1 Introduction

Data imbalance is a common issue in a variety of NLP tasks such as tagging and machine reading comprehension. Table 1 gives concrete examples: for the Named Entity Recognition (NER) task (Sang and De Meulder, 2003; Nadeau and Sekine, 2007), most tokens are backgrounds with tagging class O. Specifically, the number of tokens tagging class O is 5 times as many as those with entity labels for the CoNLL03 dataset and 8 times for the OntoNotes5.0 dataset; Data-imbalanced issue is more severe for MRC tasks (Rajpurkar et al., 2016; Nguyen et al., 2016; Rajpurkar et al., 2018; Kočiskỳ et al., 2018; Dasigi et al., 2019) with the value of negative-positive ratio being 50-200.[2]

Data imbalance results in the following two issues: (1) **the training-test discrepancy**: Without balancing the labels, the learning process tends to converge to a point that strongly biases towards class with the majority label. This actually creates a discrepancy between training and test: at training time, each training instance contributes equally to the objective function while at test time, F1 score concerns more about positive examples; (2) **the overwhelming effect of easy-negative examples**. As pointed out by Meng et al. (2019), significantly large number of negative examples also

[2] This is because the task of MRC is formalized as predicting the *starting* and *ending* indexes conditioned on the query and the context, which means that given a chunk of text of an arbitrary length, only two tokens are positive (or of interest), with all the rest being background.

means that the number of easy-negative example is large. The huge number of easy examples tends to overwhelm the training, making the model not sufficiently learned to distinguish between positive examples and hard-negative examples. The cross-entropy objective (CE for short) or maximum likelihood (MLE) objective, which is widely adopted as the training objective for data-imbalanced NLP tasks (Lample et al., 2016; Wu et al., 2019; Devlin et al., 2018; Yu et al., 2018a; McCann et al., 2018; Ma and Hovy, 2016; Chen et al., 2017), handles neither of the issues.

To handle the first issue, we propose to replace CE or MLE with losses based on the Sørensen–Dice coefficient (Sorensen, 1948) or Tversky index (Tversky, 1977). The Sørensen–Dice coefficient, dice loss for short, is the harmonic mean of precision and recall. It attaches equal importance to false positives (FPs) and false negatives (FNs) and is thus more immune to data-imbalanced datasets. Tversky index extends dice loss by using a weight that trades precision and recall, which can be thought as the approximation of the F_β score, and thus comes with more flexibility. Therefore, We use dice loss or Tversky index to replace CE loss to address the first issue.

Only using dice loss or Tversky index is not enough since they are unable to address the dominating influence of easy-negative examples. This is intrinsically because dice loss is actually a hard version of the F1 score. Taking the binary classification task as an example, at test time, an example will be classified as negative as long as its probability is smaller than 0.5, but training will push the value to 0 as much as possible. This gap isn't a big issue for balanced datasets, but is extremely detrimental if a big proportion of training examples are easy-negative ones: easy-negative examples can easily dominate training since their probabilities can be pushed to 0 fairly easily. Meanwhile, the model can hardly distinguish between hard-negative examples and positive ones. Inspired by the idea of focal loss (Lin et al., 2017) in computer vision, we propose a dynamic weight adjusting strategy, which associates each training example with a weight in proportion to $(1-p)$, and this weight dynamically changes as training proceeds. This strategy helps to deemphasize confident examples during training as their p approaches the value of 1, makes the model attentive to hard-negative examples, and thus alleviates the dominating effect of easy-negative examples.

Combing both strategies, we observe significant performance boosts on a wide range of data imbalanced NLP tasks. Notably, we are able to achieve SOTA results on CTB5 (97.92, +1.86), CTB6 (96.57, +1.80) and UD1.4 (96.98, +2.19) for the POS task; SOTA results on CoNLL03 (93.33, +0.29), OntoNotes5.0 (92.07, +0.96)), MSRA 96.72(+0.97) and OntoNotes4.0 (84.47,+2.36) for the NER task; along with competitive results on the tasks of machine reading comprehension and paraphrase identification.

The rest of this paper is organized as follows: related work is presented in Section 2. We describe different training objectives in Section 3. Experimental results are presented in Section 4. We perform ablation studies in Section 5, followed by a brief conclusion in Section 6.

2 Related Work

2.1 Data Resample

The idea of weighting training examples has a long history. Importance sampling (Kahn and Marshall, 1953) assigns weights to different samples and changes the data distribution. Boosting algorithms such as AdaBoost (Kanduri et al., 2018) select harder examples to train subsequent classifiers. Similarly, hard example mining (Malisiewicz et al., 2011) downsamples the majority class and exploits the most difficult examples. Oversampling (Chen et al., 2010; Chawla et al., 2002) is used to balance the data distribution. Another line of data resampling is to dynamically control the weights of examples as training proceeds. For example, focal loss (Lin et al., 2017) used a soft weighting scheme that emphasizes harder examples during training. In self-paced learning (Kumar et al., 2010), example weights are obtained through optimizing the weighted training loss which encourages learning easier examples first. At each training step, self-paced learning algorithm optimizes model parameters and example weights jointly. Other works (Chang et al., 2017; Katharopoulos and Fleuret, 2018) adjusted the weights of different training examples based on training loss. Besides, recent work (Jiang et al., 2017; Fan et al., 2018) proposed to learn a separate network to predict sample weights.

2.2 Data Imbalance Issue in Object Detection

The background-object label imbalance issue is severe and thus well studied in the field of object detection (Li et al., 2015; Girshick, 2015; He et al., 2015; Girshick et al., 2013; Ren et al., 2015). The idea of hard negative mining (HNM) (Girshick et al., 2013) has gained much attention recently. Shrivastava et al. (2016) proposed the online hard example mining (OHEM) algorithm in an iterative manner that makes training progressively more difficult, and pushes the model to learn better. Liu et al. (2016) sorted all of the negative samples based on the confidence loss and picking the training examples with the negative-positive ratio at 3:1. Pang et al. (2019) proposed a novel method called IoU-balanced sampling and Chen et al. (2019) designed a ranking model to replace the conventional classification task with a average-precision loss to alleviate the class imbalance issue. The efforts made on object detection have greatly inspired us to solve the data imbalance issue in NLP.

3 Losses

3.1 Notation

For illustration purposes, we use the binary classification task to demonstrate how different losses work. The mechanism can be easily extended to multi-class classification.

Let $\{x_i\}$ denote a set of instances. Each x_i is associated with a golden label vector $y_i = [y_{i0}, y_{i1}]$, where $y_{i1} \in \{0, 1\}$ and $y_{i0} \in \{0, 1\}$ respectively denote the positive and negative classes, and thus y_i can be either $[0, 1]$ or $[0, 1]$. Let $p_i = [p_{i0}, p_{i1}]$ denote the probability vector, and p_{i1} and p_{i0} respectively denote the probability that a model assigns the positive and negative label to x_i.

3.2 Cross Entropy Loss

The vanilla cross entropy (CE) loss is given by:

$$\text{CE} = -\frac{1}{N}\sum_i \sum_{j\in\{0,1\}} y_{i,j} \log p_{i,j} \tag{1}$$

As can be seen from Eq.1, each x_i contributes equally to the final objective. Two strategies are normally used to address the the case where we wish that not all x_i are treated equal: associating different classes with different weighting factor α or resampling the datasets. For the former, Eq.1 is adjusted as follows:

$$\text{Weighted CE} = -\frac{1}{N}\sum_i \alpha_i \sum_{j\in\{0,1\}} y_{i,j} \log p_{i,j} \tag{2}$$

where $\alpha_i \in [0, 1]$ may be set by the inverse class frequency or treated as a hyperparameter to set by cross validation. In this work, we use $\lg(\frac{n-n_t}{n_t} + K)$ to calculate the coefficient α, where n_t is the number of samples with class t and n is the total number of samples in the training set. K is a hyperparameter to tune. The data resampling strategy constructs a new dataset by sampling training examples from the original dataset based on human-designed criteria, e.g., extract equal training samples from each class. Both strategies are equivalent to changing the data distribution and thus are of the same nature. Empirically, these two methods are not widely used due to the trickiness of selecting α especially for multi-class classification tasks and that inappropriate selection can easily bias towards rare classes (Valverde et al., 2017).

3.3 Dice coefficient and Tversky index

Sørensen–Dice coefficient (Sorensen, 1948; Dice, 1945), dice coefficient (DSC) for short, is a F1-oriented statistic used to gauge the similarity of two sets. Given two sets A and B, the dice coefficient between them is given as follows:

$$\text{DSC}(A, B) = \frac{2|A \cap B|}{|A| + |B|} \tag{3}$$

In our case, A is the set that contains of all positive examples predicted by a specific model, and B is the set of all golden positive examples in the dataset. When applied to boolean data with the definition of true positive (TP), false positive (FP), and false negative (FN), it can be then written as follows:[3]

$$\text{DSC} = \frac{2\text{TP}}{2\text{TP} + \text{FN} + \text{FP}} = F1 \tag{4}$$

For an individual example x_i, its corresponding DSC loss is given as follows:

$$\text{DSC}(x_i) = \frac{p_{i1} \cdot y_{i1}}{p_{i1} + y_{i1}} \tag{5}$$

As can be seen, for a negative example with $y_{i1} = 0$, it does not contribute to the objective.

[3] $\frac{2\text{TP}}{2\text{TP}+\text{FN}+\text{FP}} = \frac{2\frac{\text{TP}}{\text{TP}+\text{FN}}\frac{\text{TP}}{\text{TP}+\text{FP}}}{\frac{\text{TP}}{\text{TP}+\text{FN}}+\frac{\text{TP}}{\text{TP}+\text{FP}}} = \frac{2\text{Pre}\times\text{Rec}}{\text{Pre}+\text{Rec}}$

Loss	Formula (one sample x_i)
CE	$-\sum_{j\in\{0,1\}} y_{i,j} \log p_{i,j}$
Weighted CE	$-\alpha_i \sum_{j\in\{0,1\}} y_{i,j} \log p_{i,j}$
FL	$-(1-p_i)^\gamma \log(p_i)$
DL	$1 - \frac{2p_{i1}y_{i1}+\gamma}{p_{i1}^2+y_{i1}^2+\gamma}$
TL	$1 - \frac{p_{i1}y_{i1}}{p_{i1}y_{i1}+\alpha\, p_{i1}y_{i0}+\beta\, p_{i0}y_{i1}}$
DSC	$1 - \frac{(1-p_{i1})p_{i1}\cdot y_{i1}}{(1-p_{i1})p_{i1}+y_{i1}}$

Table 2: Different losses and their formulas.

For smoothing purposes, it is common to add a γ factor to both the nominator and the denominator, making the form to be as follows:

$$\text{DSC}(x_i) = \frac{p_{i1}\cdot y_{i1} + \gamma}{p_{i1} + y_{i1} + \gamma} \tag{6}$$

As can be seen, negative examples, with y_{i1} being 0 and DSC being $\frac{\gamma}{p_{i1}+\gamma}$, also contribute to the training. Additionally, Milletari et al. (2016) proposed to change the denominator to the square form for faster convergence, which leads to the following dice loss (DL):

$$\text{DL} = \frac{1}{N}\sum_i \left[1 - \frac{2p_{i1}y_{i1}+\gamma}{p_{i1}^2+y_{i1}^2+\gamma}\right] \tag{7}$$

Another version of DL is to directly compute set-level dice coefficient instead of the sum of individual dice coefficient[4]. We choose the latter due to ease of optimization.

Tversky index (TI), which can be thought as the approximation of the F_β score, extends dice coefficient to a more general case. Given two sets A and B, tversky index is computed as follows:

$$\text{TI} = \frac{|A\cap B|}{|A\cap B| + \alpha|A\backslash B| + \beta|B\backslash A|} \tag{8}$$

Tversky index offers the flexibility in controlling the tradeoff between false-negatives and false-positives. It degenerates to DSC if $\alpha=\beta=0.5$. The Tversky loss (TL) for the training set $\{x_i, y_i\}$ is thus as follows:

$$\text{TL} = \frac{1}{N}\sum_i \left[1 - \frac{p_{i1}y_{i1}}{p_{i1}y_{i1} + \alpha\, p_{i1}y_{i0} + \beta\, p_{i0}y_{i1}}\right] \tag{9}$$

[4] $\text{DL} = 1 - \frac{2\sum_i p_{i1}y_{i1}+1}{\sum_i p_{i1}^2+\sum_i y_{i1}^2+1}$

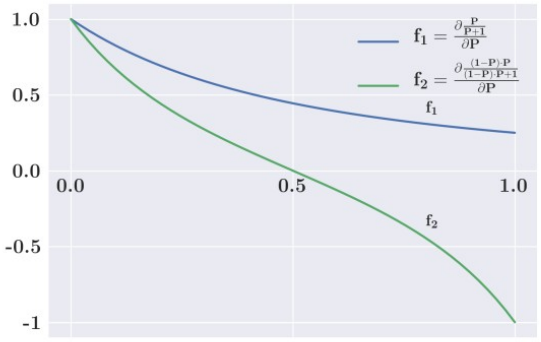

Figure 1: An illustration of the derivative of $f_1 = \frac{p}{1+p}$ and $f_2 = \frac{(1-p)p}{1+(1-p)p}$ with respect to p. As can be seen, the derivative for $\frac{(1-p)p}{1+(1-p)p}$ approaches 0 once p is larger than 0.5.

3.4 Self-adusting Dice Loss

Consider a simple case where the dataset consists of only one example x_i, which is classified as positive as long as p_{i1} is larger than 0.5. The computation of $F1$ score is actually as follows:

$$\text{F1}(x_i) = \frac{\mathbb{I}(p_{i1}>0.5)\cdot y_{i1}}{\mathbb{I}(p_{i1}>0.5) + y_{i1}} \tag{10}$$

Comparing Eq.5 with Eq.10, we can see that Eq.5 is actually a soft form of $F1$, using a continuous p rather than the binary $\mathbb{I}(p_{i1}>0.5)$. This gap isn't a big issue for balanced datasets, but is extremely detrimental if a big proportion of training examples are easy-negative ones: easy-negative examples can easily dominate training since their probabilities can be pushed to 0 fairly easily. Meanwhile, the model can hardly distinguish between hard-negative examples and positive ones, which has a huge negative effect on the final F1 performance.

To address this issue, we propose to multiply the soft probability p with a decaying factor $(1-p)$, changing Eq.10 to the following form:

$$\text{DSC}(x_i) = \frac{(1-p_{i1})p_{i1}\cdot y_{i1}}{(1-p_{i1})p_{i1} + y_{i1}} \tag{11}$$

One can think $(1-p_{i1})$ as a weight associated with each example, which changes as training proceeds. The intuition of changing p_{i1} to $(1-p_{i1})p_{i1}$ is to push down the weight of easy examples. For easy examples whose probability are approaching 0 or 1, $(1-p_{i1})p_{i1}$ makes the model attach significantly

less focus to them. Figure 1 gives gives an explanation from the perspective in derivative: the derivative of $\frac{(1-p)p}{1+(1-p)p}$ with respect to p approaches 0 immediately after p approaches 0, which means the model attends less to examples once they are correctly classified.

A close look at Eq.5 reveals that it actually mimics the idea of focal loss (FL for short) (Lin et al., 2017) for object detection in vision. Focal loss was proposed for one-stage object detector to handle foreground-background tradeoff encountered during training. It down-weights the loss assigned to well-classified examples by adding a $(1-p)^{\beta}$ factor, leading the final loss to be $(1-p)^{\beta}\log p$.

In Table 2, we show the losses used in our experiments, which is described in the next section.

4 Experiments

We evaluate the proposed method on four NLP tasks: part-of-speech tagging, named entity recognition, machine reading comprehension and paraphrase identification. Baselines in our experiments are optimized by using the standard cross-entropy training objective.

4.1 Part-of-Speech Tagging

Part-of-speech tagging (POS) is the task of assigning a label (e.g., noun, verb, adjective) to each word in a given text. In this paper, we choose BERT as the backbone and conduct experiments on three Chinese POS datasets. We report the span-level micro-averaged precision, recall and F1 for evaluation. Hyperparameters are tuned on the corresponding development set of each dataset.

Datasets We conduct experiments on the widely used Chinese Treebank 5.0[5], 6.0[6] as well as UD1.4[7].

- **CTB5** is a Chinese dataset for tagging and parsing, which contains 507,222 words, 824,983 characters and 18,782 sentences extracted from newswire sources.
- **CTB6** is an extension of CTB5, containing 781,351 words, 1,285,149 characters and 28,295 sentences.
- **UD** is the abbreviation of Universal Dependencies, which is a framework for consistent annotation of grammar (parts of speech, morphological features, and syntactic dependencies) across different human languages. In this work, we use UD1.4 for Chinese POS tagging.

Baselines We use the following baselines:

- **Joint-POS:** Shao et al. (2017) jointly learns Chinese word segmentation and POS.
- **Lattice-LSTM:** Zhang and Yang (2018) constructs a word-character lattice.
- **Bert-Tagger:** Devlin et al. (2018) treats part-of-speech as a tagging task.

Results Table 3 presents the experimental results on the POS task. As can be seen, the proposed DSC loss outperforms the best baseline results by a large margin, i.e., outperforming BERT-tagger by +1.86 in terms of F1 score on CTB5, +1.80 on CTB6 and +2.19 on UD1.4. As far as we are concerned, we are achieving SOTA performances on the three datasets. Weighted cross entropy and focal loss only gain a little performance improvement on CTB5 and CTB6, and the dice loss obtains huge gain on CTB5 but not on CTB6, which indicates the three losses are not consistently robust in resolving the data imbalance issue. The proposed DSC loss performs robustly on all the three datasets.

4.2 Named Entity Recognition

Named entity recognition (NER) refers to the task of detecting the span and semantic category of entities from a chunk of text. Our implementation uses the current state-of-the-art BERT-MRC model proposed by Li et al. (2019) as a backbone. For English datasets, we use $\text{BERT}_{\text{Large}}$ English checkpoints[8], while for Chinese we use the official Chinese checkpoints[9]. We report span-level micro-averaged pre-

[5] https://catalog.ldc.upenn.edu/LDC2005T01

[6] https://catalog.ldc.upenn.edu/LDC2007T36

[7] https://universaldependencies.org/#download

[8] https://storage.googleapis.com/bert_models/2019_05_30/wwm_uncased_L-24_H-1024_A-16.zip

[9] https://storage.googleapis.com/bert_models/2018_11_03/chinese_L-12_H-768_A-12.zip

	CTB5			CTB6			UD1.4		
Model	P	R	F	P	R	F	P	R	F
Joint-POS(Sig)(Shao et al., 2017)	93.68	94.47	94.07	-	-	90.81	89.28	89.54	89.41
Joint-POS(Ens)(Shao et al., 2017)	93.95	94.81	94.38	-	-	-	89.67	89.86	89.75
Lattice-LSTM(Zhang and Yang, 2018)	94.77	95.51	95.14	92.00	90.86	91.43	90.47	89.70	90.09
BERT-Tagger(Devlin et al., 2018)	95.86	96.26	**96.06**	94.91	94.63	**94.77**	95.42	94.17	**94.79**
BERT+WeightCE	96.45	96.41	96.43(+0.37)	95.34	96.22	95.78(+1.01)	96.09	97.08	96.58(+1.79)
BERT+FL	96.11	97.42	96.76(+0.70)	95.80	95.08	95.44(+0.67)	96.33	95.85	96.81(+2.02)
BERT+DL	96.77	98.87	97.81(+1.75)	94.08	96.12	95.09(+0.32)	96.10	97.79	96.94(+2.15)
BERT+DSC	97.10	98.75	**97.92(+1.86)**	96.29	96.85	**96.57(+1.80)**	96.24	97.73	**96.98(+2.19)**

Table 3: Experimental results for POS datasets. WeightCE denotes weighted cross-entropy, FL denotes focal loss, DL denotes dice loss and DSC denotes adjusted dice coefficient.

English CoNLL 2003			
Model	P	R	F
ELMo(Peters et al., 2018)	-	-	92.22
CVT(Clark et al., 2018)	-	-	92.6
BERT-Tagger(Devlin et al., 2018)	-	-	92.8
BERT-MRC(Li et al., 2019)	92.33	94.61	**93.04**
BERT-MRC+WeightCE	93.32	92.78	93.05(+0.01)
BERT-MRC+FL	93.13	93.09	93.11(+0.06)
BERT-MRC+DL	93.22	93.12	93.17(+0.12)
BERT-MRC+DSC	**93.41**	**93.25**	**93.33(+0.29)**
English OntoNotes 5.0			
Model	P	R	F
CVT (Clark et al., 2018)	-	-	88.8
BERT-Tagger (Devlin et al., 2018)	90.01	88.35	89.16
BERT-MRC(Li et al., 2019)	92.98	89.95	**91.11**
BERT-MRC+WeightCE	89.99	92.92	91.43(+0.32)
BERT-MRC+FL	90.13	92.34	91.22(+0.11)
BERT-MRC+DL	91.70	92.06	91.88(+0.77)
BERT-MRC+DSC	**91.59**	**92.56**	**92.07(+0.96)**
Chinese MSRA			
Model	P	R	F
Lattice-LSTM (Zhang and Yang, 2018)	93.57	92.79	93.18
BERT-Tagger (Devlin et al., 2018)	94.97	94.62	94.80
Glyce-BERT (Wu et al., 2019)	95.57	95.51	95.54
BERT-MRC(Li et al., 2019)	96.18	95.12	**95.75**
BERT-MRC+WeightCE	96.08	94.79	95.43(-0.32)
BERT-MRC+FL	95.45	95.89	95.67(-0.08)
BERT-MRC+DL	96.20	96.68	96.44(+0.69)
BERT-MRC+DSC	**96.67**	**96.77**	**96.72(+0.97)**
Chinese OntoNotes 4.0			
Model	P	R	F
Lattice-LSTM (Zhang and Yang, 2018)	76.35	71.56	73.88
BERT-Tagger (Devlin et al., 2018)	78.01	80.35	79.16
Glyce-BERT (Wu et al., 2019)	81.87	81.40	80.62
BERT-MRC(Li et al., 2019)	[illegible]	[illegible]	**82.11**
BERT-MRC+WeightCE	83.45	83.87	83.66(+1.55)
BERT-MRC+FL	83.63	82.97	83.30(+1.19)
BERT-MRC+DL	83.97	84.05	84.01(+1.90)
BERT-MRC+DSC	**84.22**	**84.72**	**84.47(+2.36)**

Table 4: Experimental results for NER task. WeightCE denotes weighted cross-entropy, FL denotes focal loss, DL denotes dice loss and DSC denotes adjusted dice coefficient.

cision, recall and F1-score. Hyperparameters are tuned on the development set of each dataset.

Datasets For the NER task, we consider both Chinese datasets, i.e., OntoNotes4.0 (Pradhan et al., 2011) and MSRA (Levow, 2006), and English datasets, i.e., CoNLL2003 (Sang and Meulder, 2003) and OntoNotes5.0 (Pradhan et al., 2013).

- **CoNLL2003** is an English dataset with 4 entity types: Location, Organization, Person and Miscellaneous. We followed data processing protocols in (Ma and Hovy, 2016).
- **English OntoNotes5.0** consists of texts from a wide variety of sources and contains 18 entity types. We use the standard train/dev/test split of CoNLL2012 shared task.
- **Chinese MSRA** performs as a Chinese benchmark dataset containing 3 entity types. Data in MSRA is collected from news domain. Since the development set is not provided in the original MSRA dataset, we randomly split the training set into training and development splits by 9:1. We use the official test set for evaluation.
- **Chinese OntoNotes4.0** is a Chinese dataset and consists of texts from news domain, which has 18 entity types. In this paper, we take the same data split as Wu et al. (2019) did.

Baselines We use the following baselines:

- **ELMo:** a tagging model from Peters et al. (2018).
- **Lattice-LSTM:** Zhang and Yang (2018) constructs a word-character lattice, only used in Chinese datasets.
- **CVT:** from Clark et al. (2018), which uses Cross-View Training(CVT) to improve the representations of a Bi-LSTM encoder.
- **Bert-Tagger:** Devlin et al. (2018) treats NER as a tagging task.
- **Glyce-BERT:** Wu et al. (2019) combines glyph information with BERT pretraining.

	SQuAD v1.1		SQuAD v2.0		QuoRef	
	EM	F1	EM	F1	EM	F1
Model						
QANet (Yu et al., 2018b)	73.6	82.7	-	-	34.41	38.26
BERT (Devlin et al., 2018)	84.1	90.9	78.7	81.9	58.44	64.95
BERT+FL	84.67(+0.57)	91.25(+0.35)	78.92(+0.22)	82.20(+0.30)	60.78(+2.34)	66.19(+1.24)
BERT+DL	84.83(+0.73)	91.86(+0.96)	78.99(+0.29)	82.88(+0.98)	62.03(+3.59)	66.88(+1.93)
BERT+DSC	**85.34(+1.24)**	**91.97(+1.07)**	**79.02(+0.32)**	**82.95(+1.05)**	**62.44(+4.00)**	**67.52(+2.57)**
XLNet (Yang et al., 2019)	88.95	94.52	86.12	88.79	64.52	71.49
XLNet+FL	88.90(-0.05)	94.55(+0.03)	87.04(+0.92)	89.32(+0.53)	65.19(+0.67)	72.34(+0.85)
XLNet+DL	89.13(+0.18)	95.36(+0.84)	87.22(+1.10)	89.44(+0.65)	65.77(+1.25)	72.85(+1.36)
XLNet+DSC	**89.79(+0.84)**	**95.77(+1.25)**	**87.65(+1.53)**	**89.51(+0.72)**	**65.98(+1.46)**	**72.90(+1.41)**

Table 5: Experimental results for MRC task. FL denotes focal loss, DL denotes dice loss and DSC denotes adjusted dice coefficient.

Model	MRPC F1	QQP F1
BERT (Devlin et al., 2018)	88.0	91.3
BERT+FL	88.43(+0.43)	91.86(+0.56)
BERT+DL	88.71(+0.71)	91.92(+0.62)
BERT+DSC	**88.92(+0.92)**	**91.57(+0.27)**
XLNet (Yang et al., 2019)	89.2	91.8
XLNet+FL	89.25(+0.05)	91.19(-0.61)
XLNet+DL	89.33(+0.13)	91.37(-0.43)
XLNet+DSC	**89.78(+0.58)**	**92.53(+0.73)**

Table 6: Experimental results for PI task. FL denotes focal loss, DL denotes dice loss and DSC denotes adjusted dice coefficient.

- **BERT-MRC:** The current SOTA model for both Chinese and English NER datasets proposed by Li et al. (2019), which formulate NER as machine reading comprehension task.

Results Table 4 shows experimental results on NER datasets. For English datasets including CoNLL2003 and OntoNotes5.0, our proposed method outperforms BERT-MRC(Li et al., 2019) by +0.29 and +0.96 respectively. We observe huge performance boosts on Chinese datasets, achieving F1 improvements by +0.97 and +2.36 on MSRA and OntoNotes4.0, respectively. As far as we are concerned, we are setting new SOTA performances on all of the four NER datasets.

4.3 Machine Reading Comprehension

Machine reading comprehension (MRC) (Seo et al., 2016; Wang et al., 2016; Wang and Jiang, 2016; Wang et al., 2016; Shen et al., 2017; Chen et al., 2017) has become a central task in natural language understanding. MRC in the SQuAD-style is to predict the answer span in the passage given a question and the passage. In this paper, we choose the SQuAD-style MRC task and report Extract Match (EM) in addition to F1 score on validation set. All hyperparameters are tuned on the development set of each dataset.

Datasets The following five datasets are used for MRC task: SQuAD v1.1, SQuAD v2.0 (Rajpurkar et al., 2016, 2018) and Quoref (Dasigi et al., 2019).

- **SQuAD v1.1 and SQuAD v2.0** are the most widely used QA benchmarks. SQuAD1.1 is a collection of 100K crowdsourced question-answer pairs, and SQuAD2.0 extends SQuAD1.1 allowing no short answer exists in the provided passage.
- **Quoref** is a QA dataset which tests the coreferential reasoning capability of reading comprehension systems, containing 24K questions over 4.7K paragraphs from Wikipedia.

Baselines We use the following baselines:

- **QANet:** Yu et al. (2018b) builds a model based on convolutions and self-attention. Convolution to model local interactions and self-attention to model global interactions.
- **BERT:** Devlin et al. (2018) treats NER as a tagging task.
- **XLNet:** Yang et al. (2019) proposes a generalized autoregressive pretraining method that enables learning bidirectional contexts.

Results Table 5 shows the experimental results for MRC tasks. With either BERT or XLNet, our proposed DSC loss obtains significant performance boost on both EM and F1. For SQuADv1.1, our proposed method outperforms XLNet by +1.25 in

terms of F1 score and +0.84 in terms of EM and achieves 87.65 on EM and 89.51 on F1 for SQuAD v2.0. Moreover, on QuoRef, the proposed method surpasses XLNet results by +1.46 on EM and +1.41 on F1. Another observation is that, XLNet outperforms BERT by a huge margin, and the proposed DSC loss can obtain further performance improvement by an average score above 1.0 in terms of both EM and F1, which indicates the DSC loss is complementary to the model structures.

4.4 Paraphrase Identification

Paraphrases are textual expressions that have the same semantic meaning using different surface words. Paraphrase identification (PI) is the task of identifying whether two sentences have the same meaning or not. We use BERT (Devlin et al., 2018) and XLNet (Yang et al., 2019) as backbones and report F1 score for comparison. Hyperparameters are tuned on the development set of each dataset.

Datasets We conduct experiments on two widely used datasets for PI task: MRPC (Dolan and Brockett, 2005) and QQP[10].

- **MRPC** is a corpus of sentence pairs automatically extracted from online news sources, with human annotations of whether the sentence pairs are semantically equivalent. The MRPC dataset has imbalanced classes (68% positive, 32% for negative).
- **QQP** is a collection of question pairs from the community question-answering website Quora. The class distribution in QQP is also unbalanced (37% positive, 63% negative).

Results Table 6 shows the results for PI task. We find that replacing the training objective with DSC introduces performance boost for both BERT and XLNet. Using DSC loss improves the F1 score by +0.58 for MRPC and +0.73 for QQP.

5 Ablation Studies

5.1 The Effect of Dice Loss on Accuracy-oriented Tasks

We argue that the most commonly used cross-entropy objective is actually accuracy-oriented, whereas the proposed dice loss (DL) performs as a hard version of F1-score. To explore the effect of the dice loss on accuracy-oriented tasks such as text classification, we conduct experiments on the Stanford Sentiment Treebank sentiment classification datasets including SST-2 and SST-5. We fine-tune $\text{BERT}_{\text{Large}}$ with different training objectives. Experiment results for SST are shown in 7. For SST-5, BERT with CE achieves 55.57 in terms of accuracy, with DL and DSC losses slightly degrade the accuracy performance and achieve 54.63 and 55.19, respectively. For SST-2, BERT with CE achieves 94.9 in terms of accuracy. The same as SST-5, we observe a slight performance drop with DL and DSC, which means that the dice loss actually works well for F1 but not for accuracy.

	SST-2	SST-5
Model	Acc	Acc
BERT+CE	94.9	55.57
BERT+DL	94.37	54.63
BERT+DSC	94.84	55.19

Table 7: The effect of dice loss on sentiment classification. BERT+CE refers to fine-tune BERT model and set cross-entropy as the training objective.

5.2 The Effect of Hyperparameters in Tversky index

As mentioned in Section 3.3, Tversky index (TI) offers the flexibility in controlling the tradeoff between false-negatives and false-positives. In this subsection, we explore the effect of hyperparameters (i.e., α and β) in TI to test how they manipulate the tradeoff. We conduct experiments on the Chinese OntoNotes4.0 NER dataset and English QuoRef MRC dataset to examine the influence of tradeoff between precision and recall. Experiment results are shown in Table 8. The highest F1 for Chinese OntoNotes4.0 is 84.67 when α is set to 0.6 while for QuoRef, the highest F1 is 68.44 when α is set to 0.4. In addition, we can observe that the performance varies a lot as α changes in distinct datasets, which shows that the hyperparameters α, β play an important role in the proposed method.

6 Conclusion

In this paper, we alleviate the severe data imbalance issue in NLP tasks. We propose to use dice

[10] https://www.quora.com/q/quoradata/First-Quora-Dataset-Release-Question-Pairs

α	Chinese Onto4.0	English QuoRef
$\alpha = 0.1$	80.13	63.23
$\alpha = 0.2$	81.17	63.45
$\alpha = 0.3$	84.22	65.88
$\alpha = 0.4$	84.52	**68.44**
$\alpha = 0.5$	84.47	67.52
$\alpha = 0.6$	**84.67**	66.35
$\alpha = 0.7$	81.81	65.09
$\alpha = 0.8$	80.97	64.13
$\alpha = 0.9$	80.21	64.84

Table 8: The effect of α in Tversky Index.

loss in replacement of the standard cross-entropy loss, which performs as a soft version of F1 score. Using dice loss can help narrow the gap between training objectives and evaluation metrics. Empirically, we show that the proposed training objective leads to significant performance boost for part-of-speech, named entity recognition, machine reading comprehension and paraphrase identification tasks.

References

Haw-Shiuan Chang, Erik G. Learned-Miller, and Andrew McCallum. 2017. Active bias: Training more accurate neural networks by emphasizing high variance samples. In *NIPS*.

N. V. Chawla, K. W. Bowyer, Lawrence O. Hall, and W. P. Kegelmeyer. 2002. Smote: Synthetic minority over-sampling technique. *J. Artif. Intell. Res.*, 16:321–357.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading wikipedia to answer open-domain questions. *arXiv preprint arXiv:1704.00051*.

Kean Chen, Jianguo Li, Weiyao Lin, John See, Ji Wang, Lingyu Duan, Zhibo Chen, Changwei He, and Junni Zou. 2019. Towards accurate one-stage object detection with ap-loss. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 5119–5127.

Shijuan Chen, Haibo He, and Edwardo A. Garcia. 2010. Ramoboost: Ranked minority oversampling in boosting. *IEEE Transactions on Neural Networks*, 21:1624–1642.

Kevin Clark, Minh-Thang Luong, Christopher D. Manning, and Quoc V. Le. 2018. Semi-supervised sequence modeling with cross-view training. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Procfessing, Brussels, Belgium, October 31 - November 4, 2018*, pages 1914–1925.

Pradeep Dasigi, Nelson F Liu, Ana Marasovic, Noah A Smith, and Matt Gardner. 2019. Quoref: A reading comprehension dataset with questions requiring coreferential reasoning. *arXiv preprint arXiv:1908.05803*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Lee R Dice. 1945. Measures of the amount of ecologic association between species. *Ecology*, 26(3):297–302.

William B. Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*.

Yang Fan, Fei Tian, Tao Qin, Xiuping Li, and Tie-Yan Liu. 2018. Learning to teach. *ArXiv*, abs/1805.03643.

Ross B. Girshick. 2015. Fast r-cnn. *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1440–1448.

Ross B. Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. 2013. Rich feature hierarchies for accurate object detection and semantic segmentation. *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 580–587.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778.

Lu Jiang, Zhengyuan Zhou, Thomas Leung, Li-Jia Li, and Li Fei-Fei. 2017. Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In *ICML*.

H. Kahn and A. W. Marshall. 1953. Methods of reducing sample size in monte carlo computations. *Operations Research*, 1(5):263–278.

Anil Kanduri, Mohammad Hashem Haghbayan, Amir M. Rahmani, Muhammad Shafique, Axel Jantsch, and Pasi Liljeberg. 2018. adboost: Thermal aware performance boosting through dark silicon patterning. *IEEE Trans. Computers*, 67(8):1062–1077.

Angelos Katharopoulos and François Fleuret. 2018. Not all samples are created equal: Deep learning with importance sampling. In *ICML*.

Tomáš Kočiskỳ, Jonathan Schwarz, Phil Blunsom, Chris Dyer, Karl Moritz Hermann, Gábor Melis, and Edward Grefenstette. 2018. The narrativeqa reading comprehension challenge. *Transactions of the Association for Computational Linguistics*, 6:317–328.

M. Pawan Kumar, Benjamin Packer, and Daphne Koller. 2010. Self-paced learning for latent variable models. In *Advances in Neural Information Processing Systems 23: 24th Annual Conference on Neural Information Processing Systems 2010. Proceedings of a meeting held 6-9 December 2010, Vancouver, British Columbia, Canada.*, pages 1189–1197.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. *arXiv preprint arXiv:1603.01360*.

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.

H. Li, Z. Lin, X. Shen, J. Brandt, and G. Hua. 2015. A convolutional neural network cascade for face detection. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5325–5334.

Xiaoya Li, Jingrong Feng, Yuxian Meng, Qinghong Han, Fei Wu, and Jiwei Li. 2019. A unified MRC framework for named entity recognition. *CoRR*, abs/1910.11476.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2017. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988.

Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott E. Reed, Cheng-Yang Fu, and Alexander C. Berg. 2016. SSD: single shot multibox detector. In *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part I*, pages 21–37.

Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via bi-directional lstm-cnns-crf. *arXiv preprint arXiv:1603.01354*.

Tomasz Malisiewicz, Abhinav Gupta, and Alexei A. Efros. 2011. Ensemble of exemplar-svms for object detection and beyond. In *IEEE International Conference on Computer Vision, ICCV 2011, Barcelona, Spain, November 6-13, 2011*, pages 89–96.

Bryan McCann, Nitish Shirish Keskar, Caiming Xiong, and Richard Socher. 2018. The natural language decathlon: Multitask learning as question answering. *arXiv preprint arXiv:1806.08730*.

Yuxian Meng, Muyu Li, Wei Wu, and Jiwei Li. 2019. Dsreg: Using distant supervision as a regularizer. *arXiv preprint arXiv:1905.11658*.

Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. 2016. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *2016 Fourth International Conference on 3D Vision (3DV)*, pages 565–571. IEEE.

David Nadeau and Satoshi Sekine. 2007. A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30(1):3–26.

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. Ms marco: A human generated machine reading comprehension dataset. *arXiv preprint arXiv:1611.09268*.

Jiangmiao Pang, Kai Chen, Jianping Shi, Huajun Feng, Wanli Ouyang, and Dahua Lin. 2019. Libra R-CNN: towards balanced learning for object detection. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 821–830.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*.

Sameer Pradhan, Mitchell P. Marcus, Martha Palmer, Lance A. Ramshaw, Ralph M. Weischedel, and Nianwen Xue, editors. 2011. *Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task, CoNLL 2011, Portland, Oregon, USA, June 23-24, 2011*. ACL.

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.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for squad. *arXiv preprint arXiv:1806.03822*.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*.

Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39:1137–1149.

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

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, CoNLL 2003, Held in cooperation with HLT-NAACL 2003, Edmonton, Canada, May 31 - June 1, 2003*, pages 142–147.

Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2016. Bidirectional attention flow for machine comprehension. *arXiv preprint arXiv:1611.01603*.

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

Yelong Shen, Po-Sen Huang, Jianfeng Gao, and Weizhu Chen. 2017. Reasonet: Learning to stop reading in machine comprehension. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1047–1055. ACM.

Abhinav Shrivastava, Abhinav Gupta, and Ross Girshick. 2016. Training region-based object detectors with online hard example mining. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.

Th A Sorensen. 1948. A method of establishing groups of equal amplitude in plant sociology based on similarity of species content and its application to analyses of the vegetation on danish commons. *Biol. Skar.*, 5:1–34.

Amos Tversky. 1977. Features of similarity. *Psychological review*, 84(4):327.

Sergi Valverde, Mariano Cabezas, Eloy Roura, Sandra González-Villà, Deborah Pareto, Joan C Vilanova, Lluís Ramió-Torrentà, Àlex Rovira, Arnau Oliver, and Xavier Lladó. 2017. Improving automated multiple sclerosis lesion segmentation with a cascaded 3d convolutional neural network approach. *NeuroImage*, 155:159–168.

Shuohang Wang and Jing Jiang. 2016. Machine comprehension using match-lstm and answer pointer. *arXiv preprint arXiv:1608.07905*.

Zhiguo Wang, Haitao Mi, Wael Hamza, and Radu Florian. 2016. Multi-perspective context matching for machine comprehension. *arXiv preprint arXiv:1612.04211*.

Wei Wu, Yuxian Meng, Qinghong Han, Muyu Li, Xiaoya Li, Jie Mei, Ping Nie, Xiaofei Sun, and Jiwei Li. 2019. Glyce: Glyph-vectors for chinese character representations. *arXiv preprint arXiv:1901.10125*.

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

Adams Wei Yu, David Dohan, Minh-Thang Luong, Rui Zhao, Kai Chen, Mohammad Norouzi, and Quoc V Le. 2018a. Qanet: Combining local convolution with global self-attention for reading comprehension. *arXiv preprint arXiv:1804.09541*.

Adams Wei Yu, David Dohan, Minh-Thang Luong, Rui Zhao, Kai Chen, Mohammad Norouzi, and Quoc V. Le. 2018b. Qanet: Combining local convolution with global self-attention for reading comprehension. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*.

Yue Zhang and Jie Yang. 2018. Chinese ner using lattice lstm. *arXiv preprint arXiv:1805.02023*.