ONE-VS-ALL AUC MAXIMIZATION: AN EFFECTIVE SOLUTION TO THE LOW-RESOURCE NAMED ENTITY RECOGNITION PROBLEM

Anonymous authors
Paper under double-blind review

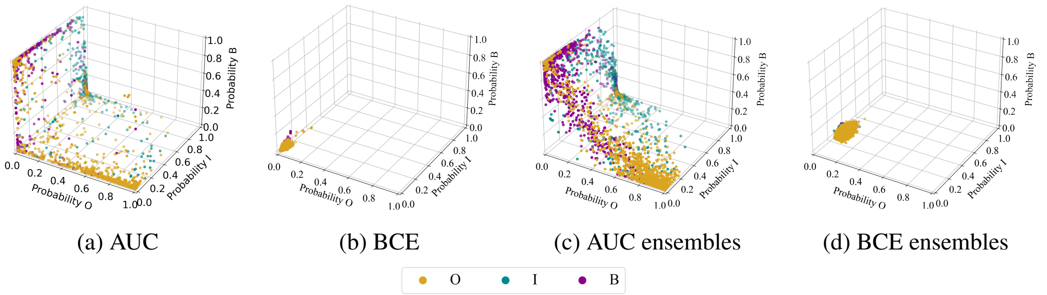
ABSTRACT

Named entity recognition (NER), a sequence labelling/token classification task, has been traditionally considered a multi-class classification problem, the learning objective of which is to either optimise the multi-class cross entropy loss (CE) or train a conditional random field (CRF). However, these standard learning objectives, though scalable to large NER datasets and often used in state-of-the-art work, largely ignore the problem of imbalanced label distributions that is inherent in all NER corpora. We show this leads to degraded performance in low-resource settings. While reformulating this standard multi-class labelling problem as a one-vs-all (OVA) learning problem, we propose to optimise the NER model with an AUC-based alternative loss function that is more capable of handling imbalanced datasets. As OVA often leads to a higher training time compared to the standard multi-class setting, we also develop two training strategies, one trains together the labels that share similar linguistic characteristics, and another employs a meta-learning approach to speed convergence. In order to motivate some of our experiments and better interpret the results, we also develop a Bayesian theory for what is the AUC function during learning. Experimental results under low-resource NER settings from benchmark corpora show that our methods can achieve consistently better performance compared with the learning objectives commonly used in NER. We also give evidence that our methods are robust and agnostic to the underlying NER embeddings, models, domains, and label distributions. The code to replicate this work will be released upon the publication of this paper.

1 INTRODUCTION

Named Entity Recognition (NER), a fundamental NLP task, aims to detect the semantic category of named entity (NE), *e.g.*, location, organization, or person. Being an important prerequisite for many language applications, NER is deeply integrated in several NLP tasks such as information extraction (Ritter et al., 2012), information retrieval (Banerjee et al., 2019), task oriented dialogues (Peng et al., 2021), and knowledge base construction (Etzioni et al., 2005). Recently, NER has gained significant performance improvements with the advances of state-of-the-art (SOTA) pre-trained language models (PLMs) (Devlin et al., 2019). Unfortunately, these PLMs rely on sizable training datasets to achieve high performance and the lack of such datasets in the specialized low resource domains (*e.g.*, biomedical domain) can often lead to sub-optimal performance (Yaseen & Langer, 2021).

As NER models rely heavily on human annotated data which can be expensive, time-consuming and often infeasible without domain expertise, existing machine learning approaches such as domain adaptation (Li et al., 2020), and data augmentation (Zhou et al., 2022) have been adapted to NER to alleviate this dependence on labeled data. Nonetheless, these approaches largely ignore the imbalanced label distribution that inherently exists in most NER corpora. Table 1 documents this imbalance issue where the majority of labels in the NER corpora is of non-entity type “O”, providing NER models with little learning signals. For specialized biomedical corpora such as NCBI (Doğan et al., 2014), and s800 (Pafilis et al., 2013), the corpus can be strongly imbalanced with more than 90% of its labels tagged as “O”. Although this problem can be mitigated given sizable training sets, most specialized domains lack such datasets (Giorgi & Bader, 2019; Yaseen & Langer, 2021).



(a) AUC (b) BCE (c) AUC ensembles (d) BCE ensembles

Figure 1: One-vs-All predicted probability on BIO-tag of each tokens for CoNLL 2003 test data. We visualise both the normal and deep ensemble performance of AUC and BCE loss/objective functions trained with 100 sentences. While the binary classifiers trained with BCE are not confident in classifying any tokens, as all the predictions are made to the lower-left corner. In contrast, the same binary classifiers can learn to classify the tokens appropriately if trained with the AUC loss function.

With inherent imbalanced label distributions for NER corpora, we argue that even though standard learning objective functions (*i.e.*, multi-class cross entropy loss or conditional random fields), given adequate annotated training data, are able to produce well-performing token classifiers or sequence labelers, their performance can substantially degrade in the low-resource settings. Consequently, we propose to directly address the imbalanced problem by training NER models with a surrogate loss that maximizes the area under ROC curve (AUC) score. AUC maximization has been shown to greatly improve the model prediction for imbalanced label distribution tasks (Gao et al., 2013; Ying et al., 2016; Yuan et al., 2021a; Huang et al., 2022). Unfortunately, most recent practical AUC surrogate loss/objective functions are solely designed to solve the binary classification task, so are not directly applicable to NER. We thus reformulate the standard sequence tagging problem as a one-vs-all (OVA) learning problem. Under the OVA setup, each unique label (*e.g.*, B-PER, I-ORG) will have its own binary classifier, which can be individually learned using an AUC objective function. We did not use the one-vs-other (OVO) reformulation, with well developed theory Yang et al. (2021), because of its computational complexity.

However, it is not unknown that OVA often suffers from the following two weaknesses: **(i)** OVA has higher training time compared to the traditional multiclass setup since each binary classifier should be independently trained, and **(ii)** OVA is not data efficient for learning from imbalanced data and often produces less accurate classifiers compared to the multiclass setup (Liu et al., 2017b), Figure 1 shows that the predictive confidence of OVA, learnt with the binary cross entropy loss (BCE), always concentrates on the lower probability region (*i.e.*, the lower left corner). We alleviate the first weakness with two new training strategies: one groups labels that share similar linguistic characteristics and trains their classifiers together; another adapts the idea of meta-learning (Finn et al., 2017) by selecting a random batch of binary classifiers for the model to learn. For the second weakness, we tune the binary classifier with the AUC surrogate loss function, which has shown its robustness/resilience to imbalanced data (Ying et al., 2016; Yuan et al., 2021a;b; Huang et al., 2022).

To demonstrate the effectiveness of our proposed method, we conduct extensive empirical studies on benchmark corpora that are from both generic domains, *e.g.*, CoNLL 2003 and OntoNotes5, and specialized domains, *e.g.*, NCBI and s800. Additionally, we implement several SOTA model architectures and embeddings to verify the agnosticity of our method. Our studies reveal that our OVA AUC NER setups, under the low resource settings, exhibit significant performance improvement over the standard multiclass objective functions by a large margin, regardless of the underlying NER embeddings, models, corpora, and label distributions. We also provide evidence to the theoretical proof to derive the ranking function that gives the optimal AUC score under Bayesian context.

We summarize the contributions of our work as follows:

- **Reformulation of NER as an OVA task**: We transform NER from a standard multi-class learning problem to an one-vs-all learning problem. Each unique label in the corpus will have its own binary classifier. This simple OVA reformulation makes the AUC maximization feasible for NER.
- **Effectiveness of AUC maximization for NER under OVA**: We show that learning the binary classifiers with AUC objective function can lead to well-tuned classifiers for the imbalanced label

Dataset	**# Sentences**	**# Tokens**	**# Labels**	**% label (B/I/O)**		
	Train/Dev/Test	Train/Dev/Test		Train	Dev	Test
OntoNotes5	20,000/3,000/3,000	364,344/54,372/55,754	37	6.2/4.8/89.0	6.2/4.8/89.0	6.1/4.9/89.0
CoNLL 2003	14,040/3,249/3,452	203,589/51,319/46,376	9	11.5/5.2/83.3	11.6/5.1/83.3	12.2/5.3/82.5
NCBI	5,424/923/940	135,597/23,969/24,481	3	3.8/4.5/91.7	3.3/4.5/92.2	3.9/4.5/91.6
s800	5,733/830/1,630	147,205/22,166/42,287	3	1.7/2.3/96.0	1.7/2.2/96.1	1.8/2.5/95.7

Table 1: Summary of dataset distribution. Please note that we use BIO here for ease of reference, detailed summarization for each corpus label distribution can be found in Table 4 in the Appendix.

distribution problem. This gives evidence that AUC objective function naturally works under the OVA setup, regardless of the inherent weakness of OVA to imbalanced label distribution problem.

- **Bayesian AUC Maximization**: We prove that ensembling provides a Bayes optimal AUC ranking function, and thus use it to push the performance of the binary classifiers, and consequently the OVA NER model classification performance.

2 Related Work

Acting as an integral part of NLP systems, NER can include semantic categorization of generic NEs (*e.g.*, person, organization, and location) and/or domain-specifc NEs (*e.g.*, virus, protein, and genome) (Li et al., 2020). Given a list of tokens $\mathbf{x} = \{x_1, \ldots, x_l\}$, NER models are expected to output a list of corresponding labels $\mathbf{y} = \{y_1, \ldots, y_l\}$. Due to the nature of sequence labelling, NER can be a challenging task for two reasons **(i)** most languages and domains lack sizable training datasets; and **(ii)** the NER corpora label distribution can be highly imbalanced (Lample et al., 2016).

The OVA approach is mostly used to expand binary models to multi-class classifications, such as logistic regression, support vector machines, etc (Galar et al., 2011; Liu et al., 2017a), with the OVO approach seeing little use due to its higher training time. The objective of OVA is to divide a K-class problem into K binary problems. For instance, K binary classifiers must be built, where K is the number of classes, and the i-th classifier is trained with positive data from class i and negative samples from the other $K - 1$ classes. When the classifier evaluates an unclassified sample, the highest confidence value of the sample is considered to have labelled corresponding to the specified class. In recent years, researchers have discovered how OVA methods can accomplish various tasks with deep neural networks. They found that OVA can improve the ability to identify more relevant hidden representations for unidentified instances than the popular Softmax function (Jang & Kim, 2020). It also improves calibration on image classification, outlier detection, and dataset shift tasks, reaching Softmax's predictive performance without increasing training or test time complexity (Padhy et al., 2020; Saito & Saenko, 2021; Lübbering et al., 2021). Although the algorithm is simple, it shows impressive results, demonstrating that its performance is usually at least as accurate as other multi-class algorithms when appropriately tuned (Rifkin & Klautau, 2004).

AUC (Area Under ROC Curve) has been traditionally treated as an important measuring criterion for model classification performance (Freund et al., 2003; Kotlowski et al., 2011; Zuva & Zuva, 2012). Due to its non-convex, and discontinuous nature, most works consider direct optimization of AUC score an NP-hard problem (Yuan et al., 2021a). Freund et al. (2003) tried to alleviate this computation difficulty through a pairwise surrogate loss, while Zhao et al. (2011) implemented a hinge loss. However, both of these surrogate losses lack scalability to large datasets and models. This led to the development of the least-square surrogate loss (Gao et al., 2013). Recent research on AUC maximization further optimize the least-square surrogate loss via deep margin surrogate loss (Yuan et al., 2021a) and compositional training (Yuan et al., 2021b). Overall, AUC maximization is documented to work well when there exists an imbalanced label distribution, or the AUC score is the default metric for evaluating and comparing different methods (Yuan et al., 2021a). To the best of our knowledge, this is the first work exploring AUC maximization in the context of NER tasks.

Theoretical results on AUC show consistency, training models under some univariate losses (not pairwise like AUC) is asymptotically equivalent to AUC training (Gao & Zhou, 2015), and the results have been extended to the OVO case Yang et al. (2021). Thus, special purpose AUC training should not be effective for larger data sets. Theory for Bayes optimal AUC scoring for finite data, however, has not been developed to the best of our knowledge.

3 AUC Maximization for NER

Given a set of training data $\mathcal{S} = \{(\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_n, \mathbf{y}_n)\}$, where $\mathbf{x}_i = x_i^1, \ldots, x_i^l$ represents the i-th training example (*i.e.*, a sentence of length l), and $\mathbf{y}_i \in \{\mathrm{B}, \mathrm{I}, \mathrm{O}\}^l$ denotes its corresponding sequence of labels, we would like to learn the objective mapping function $h : \mathcal{X} \rightarrow \mathcal{Y}$. This objective mapping function, traditionally learned with either CRFs or the CE loss, is parameterized with $\mathbf{w} \in \mathbb{R}^d$, *i.e.*, $h_{\mathbf{w}}(\mathbf{x}) = h(\mathbf{w}, \mathbf{x})$. Lastly, different corpora can have different label set, *e.g.*, $y_i \in \{\text{B-ORG}, \text{B-PER}, \text{B-MISC}, \text{B-LOC}, \text{I-ORG}, \text{I-PER}, \text{I-MISC}, \text{I-LOC}, \text{O}\}$ for CoNLL 2003 (Tjong Kim Sang & De Meulder, 2003); thus, $\mathbf{y}_i \in \{\mathrm{B}, \mathrm{I}, \mathrm{O}\}^l$ is mainly for ease of reference.

3.1 One-Vs-All AUC Maximization For Named Entity Recognition

To make direct AUC maximization applicable to NER, we first reformulate the standard NER multi-class setup as a one-vs-all learning problem. Consequently, each unique label is given its own binary classifier that can be learned using the AUC objective function. For instance, given the "O"-tag, the label of this tag is $\mathbf{y}_{\mathbf{O}_i} \in \{-1, 1\}^l$, while the parameter of this task is $\mathbf{w}_{\mathbf{O}} = \{\theta, \omega_{\mathbf{O}}\}$. θ denotes the shared embedding and pretrained language model parameters, while $\omega_{\mathbf{O}}$ denotes the parameters for the binary classifier for the "O"-tag. After the reformulation, we can maximize the AUC score of each binary classifier via the robust and practical deep AUC margin loss (DAM) (Yuan et al., 2021a).

$$\begin{aligned}
&\mathrm{AUC}_{\mathrm{M}}(\mathbf{w}_{\mathbf{O}}) \\
&= \mathbb{E}\left[\left(m_{\mathbf{O}} - h_{\mathbf{w}_{\mathbf{O}}}(x) + h_{\mathbf{w}_{\mathbf{O}}}(x')\right)^2 \mid y_{\mathbf{O}} = 1, y'_{\mathbf{O}} = -1\right] && \text{(1a)} \\
&= \min_{a_{\mathbf{O}}, b_{\mathbf{O}}} A_1(\mathbf{w}_{\mathbf{O}}) + A_2(\mathbf{w}_{\mathbf{O}}) + (m_{\mathbf{O}} - a_{\mathbf{O}} + b_{\mathbf{O}})^2 && \text{(1b)} \\
&= \min_{a_{\mathbf{O}}, b_{\mathbf{O}}} A_1(\mathbf{w}_{\mathbf{O}}) + A_2(\mathbf{w}_{\mathbf{O}}) + \max_{\alpha_{\mathbf{O}} \geq 0} \left\{2\alpha_{\mathbf{O}}(m_{\mathbf{O}} - a_{\mathbf{O}} + b_{\mathbf{O}}) - {\alpha_{\mathbf{O}}}^2\right\}, && \text{(1c)}
\end{aligned}$$

where $A_1(\mathbf{w}_{\mathbf{O}}) = \mathbb{E}[h^2_{\mathbf{w}_{\mathbf{O}}}(x) \mid y_{\mathbf{O}} = 1] - a^2_{\mathbf{O}}$, $A_2(\mathbf{w}_{\mathbf{O}}) = \mathbb{E}[h^2_{\mathbf{w}_{\mathbf{O}}}(x) \mid y_{\mathbf{O}} = -1] - b^2_{\mathbf{O}}$, and $m_{\mathbf{O}}$ is the margin that aims to push the expected prediction scores of negative and positive class far from each other (Yuan et al., 2021a). Examining equation 1, the minimization problem of $a_{\mathbf{O}}$ and $b_{\mathbf{O}}$ is achieved when $a_{\mathbf{O}} = a(\mathbf{w}_{\mathbf{O}}) = \mathbb{E}[h_{\mathbf{w}_{\mathbf{O}}}(x) \mid y_{\mathbf{O}} = 1]$, and $b_{\mathbf{O}} = b(\mathbf{w}_{\mathbf{O}}) = \mathbb{E}[h_{\mathbf{w}_{\mathbf{O}}}(x') \mid y_{\mathbf{O}} = -1]$ respectively (Ying et al., 2016; Yuan et al., 2021a). Thus, we expect that minimizing equation 1 with respect to $\mathbf{w}_{\mathbf{O}}$ can produce a well-tuned binary classifier for the imbalanced "O"-tag prediction. Given K unique labels, we can define K similar learning objective functions. Under OVA, these K objective functions can be independently minimized to produce the optimal binary classifier for each label.

At prediction time, we evaluate the individual classifiers by following the maximum confidence strategy (Galar et al., 2011) to generate the prediction tags expected by the corpus label set.

$$\hat{y_i} = \underset{1 \ldots K}{\arg\max} \left[h_{\mathbf{w}^*_{\mathbf{1}}}(x_i), \ldots, h_{\mathbf{w}^*_{\mathbf{K}}}(x_i)\right], \tag{2}$$

where $\mathbf{w}^*_{\mathbf{k}}$ represents the optimal parameters learnt by minimizing equation 1 for the k-label classifier and $h_{\mathbf{w}^*_{\mathbf{k}}}(x_i)$ represents the probability that the x_i token belongs to class k. We acknowledge that maximum confidence strategy, although producing the appropriate label predictions, ignores the inherent weakness of OVA, *i.e.*, OVA can lead to confusion areas where **(i)** two or more binary classifiers can be confident that the sample belong to their classes, or **(ii)** no classifiers are confident enough to claim the sample, especially under the imbalanced settings (Rifkin & Klautau, 2004; Liu et al., 2017b). However, since AUC maximization can lead to well-tuned binary classifiers under the imbalanced settings, we argue that this weakness should be naturally mitigated. Figure 1 shows that the NER model, trained with the AUC surrogate loss under the OVA setup, leads to smaller confusion areas (few samples in the lower left region) compared to the cross entropy loss, indicating that the binary classifiers, learnt with the AUC surrogate loss function, are more well tuned than those learnt with the cross entropy loss. More statistical results/discussions can be found in subsection 5.3.

3.2 Bayesian Optimal AUC Ranking Function

Consider a standard Bayesian learning context: assume the data distribution $\Pr(\mathbf{x})$ is known but we do not know the class distribution, though have a Bayesian formulation for it, for instance a posterior distribution $\Pr(\mathbf{w})$ for the weights of the class distribution, $\Pr(y \mid \mathbf{x}, \mathbf{w})$. A reasonable theoretical question to ask in this Bayesian context is what function $h(\mathbf{x})$ will result in the maximum posterior expected AUC given the common definition of AUC of

$$\mathrm{AUC} = \Pr\left(h(\mathbf{x}) \geq h\left(\mathbf{x}'\right) \mid y = 1, y' = -1\right) \tag{3}$$

The following theorem provides an answer.

Theorem 1. *In the Bayesian context above with a posterior distribution* $\Pr(\mathbf{w})$*, the posterior estimate of* AUC *can only be optimal when* $h(\mathbf{x_1}) = h(\mathbf{x_2})$ *implies* $\mathbb{E}_{\mathbf{w}}\left[\Pr\left(y = 1 \mid \mathbf{x_1}, \mathbf{w}\right)\right] = \mathbb{E}_{\mathbf{w}}\left[\Pr\left(y = 1 \mid \mathbf{x_2}, \mathbf{w}\right)\right]$.

This means that any monotonic function of $\mathbb{E}_{\mathbf{w}}\left[\Pr\left(y = 1 \mid \mathbf{x}, \mathbf{w}\right)\right]$ will work as the optimal $h(\mathbf{x})$. In other words, the standard posterior Bayesian classifier also achieves the optimal posterior estimate of AUC. In our case we use a common simple approximation, ensembling (Lakshminarayanan et al., 2017), to implement this.

To prove the theorem, we first present a simpler case, where we know the conditional distribution, $\Pr\left(y \mid \mathbf{x}\right)$, so there is no learning involved. This result aligns with prior theory of AUC consistency (Gao & Zhou, 2015). It also immediately follows that any strictly proper scoring rule Gneiting & Raftery (2007), including the majority of deep learning objectives as well as surrogate losses in AUC consistency theory Gao & Zhou (2015), can be used for asymptotically optimal AUC scoring. That is, AUC consistency theory becomes redundant once you know the Bayes optimal classifier yields optimal AUC.

Lemma 2. *Consider* AUC *defined in terms of a scoring function* $h(\mathbf{x})$*. What function* $h(\mathbf{x})$ *gives the largest* AUC*?* AUC *can only be optimal when for any* $\mathbf{x}_1$ *and* $\mathbf{x}_2$*,* $h(\mathbf{x}_1) = h(\mathbf{x}_2)$ *implies* $\Pr\left(y = 1 \mid \mathbf{x}_1\right) = \Pr\left(y = 1 \mid \mathbf{x}_2\right)$.

Proof of Lemma 2. To prove this define AUC as the limit of a sigmoid function, $\mathrm{AUC} = \lim_{a \to 0+} \mathrm{AUC}_a$, where AUC_a and its differential are given by

$$\mathrm{AUC}_a = \iint \frac{1}{1 + e^{(h(\mathbf{x_1}) - h(\mathbf{x_2}))/a}} dp(\mathbf{x_1} \mid y = 1) dp(\mathbf{x_2} \mid y = -1) \tag{4}$$

$$\delta\,\mathrm{AUC}_a = -\iint \frac{(\delta h(\mathbf{x_1}) - \delta h(\mathbf{x_2}))}{a} q_a(\mathbf{x_1}, \mathbf{x_2})(1 - q_a(\mathbf{x_1}, \mathbf{x_2})) dp(\mathbf{x_1} \mid y = 1) dp(\mathbf{x_2} \mid y = -1) \tag{5}$$

where $q_a(\mathbf{x_1}, \mathbf{x_2}) = \frac{1}{1 + e^{(h(\mathbf{x_1}) - h(\mathbf{x_2}))/a}}$. We can split up the terms in $\delta h(\mathbf{x_1})$ and $\delta h(\mathbf{x_2})$. Consider the second, exchange $\mathbf{x_1}$ and $\mathbf{x_2}$ and use the symmetry of $\mathbf{x_1}, \mathbf{x_2}$ in $q_a(\mathbf{x_1}, \mathbf{x_2})(1 - q_a(\mathbf{x_1}, \mathbf{x_2}))$,

$$\begin{aligned} & \iint \frac{1}{a} \delta h(\mathbf{x_2}) q_a(\mathbf{x_1}, \mathbf{x_2})(1 - q_a(\mathbf{x_1}, \mathbf{x_2}))\; dp(\mathbf{x_1} \mid y = 1) dp(\mathbf{x_2} \mid y = -1) \\ = \; & \iint \frac{1}{a} \delta h(\mathbf{x_1}) q_a(\mathbf{x_1}, \mathbf{x_2})(1 - q_a(\mathbf{x_1}, \mathbf{x_2}))\; dp(\mathbf{x_2} \mid y = 1) dp(\mathbf{x_1} \mid y = -1) \end{aligned} \tag{6}$$

Substituting back into (4), and replacing $p(\mathbf{x} \mid y = 1) = \frac{p(y=1|\mathbf{x})}{p(y=1)} p(\mathbf{x})$, yields $\delta\,\mathrm{AUC}_a$

$$\begin{aligned} = \; & \iint \frac{\delta h(\mathbf{x_1})}{a} \frac{q_a(\mathbf{x_1}, \mathbf{x_2})(1 - q_a(\mathbf{x_1}, \mathbf{x_2}))}{p(y = 1) p(y = -1)} \\ & (p(y = 1 \mid \mathbf{x_2}) p(y = -1 \mid \mathbf{x_1}) - p(y = 1 \mid \mathbf{x_1}) p(y = -1 \mid \mathbf{x_2}))\; dp(\mathbf{x_1}) dp(\mathbf{x_2}) \\ = \; & \iint \frac{\delta h(\mathbf{x_1})}{a} \frac{q_a(\mathbf{x_1}, \mathbf{x_2})(1 - q_a(\mathbf{x_1}, \mathbf{x_2}))}{p(y = 1) p(y = -1)} \left(p(y = 1 \mid \mathbf{x_2}) - p(y = 1 \mid \mathbf{x_1})\right) dp(\mathbf{x_1}) dp(\mathbf{x_2}) \end{aligned} \tag{7}$$

Consider the quantity of $\frac{1}{a} q_a(\mathbf{x_1}, \mathbf{x_2})(1 - q_a(\mathbf{x_1}, \mathbf{x_2})$. This approaches zero exponentially as $|h(\mathbf{x_1}) - h(\mathbf{x_2})| \gg a$ and is $O(1/a)$ when $|h(\mathbf{x_1}) - h(\mathbf{x_2})| \leq O(a)$. Thus the dominant contribution to the integral $\delta\,\mathrm{AUC}_a$ comes from the region where $h(\mathbf{x_1}) \approx h(\mathbf{x_2})$, but the integral itself will be $O(1/a)$. Now, if $p(y = 1 \mid \mathbf{x_1}) \neq p(y = 1 \mid \mathbf{x_2})$, then $p(y = 1 \mid \mathbf{x_2}) - p(y = 1 \mid \mathbf{x_1}) \neq 0$ and $\delta\,\mathrm{AUC}_a \to \frac{1}{a} C$ as $a \to 0$ for some non-zero constant C. Thus AUC can only have an optimum when $h(\mathbf{x_1}) = h(\mathbf{x_2})$ implies $\Pr\left(y = 1 \mid \mathbf{x_1}\right) = \Pr\left(y = 1 \mid \mathbf{x_2}\right)$. □

Proof of Theorem 1. Modify the previous proof. First add the parameters $\mathbf{w}$ to the right hand side of $\Pr\left(y = 1 \mid \mathbf{x}\right)$ and $\Pr\left(\mathbf{x} \mid y = 1\right)$, yielding $\Pr\left(y = 1 \mid \mathbf{x}, \mathbf{w}\right)$ and $\Pr\left(\mathbf{x} \mid y = 1, \mathbf{w}\right)$. Now rewrite equation 4. denoting this as $\mathrm{AUC}_{a,\mathbf{w}}$. The Bayesian posterior estimate of AUC becomes $\mathrm{AUC} = \lim_{a \to 0} \mathbb{E}_{\mathbf{w}}\left[\mathrm{AUC}_{a,\mathbf{w}}\right]$. The final expression above for AUC_a is linear in $p(y = 1 \mid \mathbf{x}, \mathbf{w})$, while $q_a(\mathbf{x_1}, \mathbf{x_2})$ and $p(\mathbf{x})$ do not contain $\mathbf{w}$. Thus, the expectation $\mathbb{E}_{\mathbf{w}}[\cdot]$ can be carried through and the same logic applies. □

4 EXPERIMENTAL SETTINGS

Domains and Corpora: We used corpora from both the general domain and the biomedical domain to benchmark the NER model performance. Table 1 and Table 4 summarize the label distribution statistics for these corpora. Both CoNLL 2003 (Tjong Kim Sang & De Meulder, 2003) and OntoNotes5 (Weischedel et al., 2014) are benchmark corpora from the general domains for SOTA NER works. Whereas NCBI (Doğan et al., 2014), and s800 (Pafilis et al., 2013) have been used in many SOTA works in biomedical named entity recognition (bioNER) (Xu et al., 2019; Lee et al., 2019). As these corpora are varying in terms of label distribution and linguistic characteristic, they serve to substantiate the usefulness of our method regardless of the underlying corpora or domains.

Model Architecture and Embedding: For embeddings and model architectures, we focused on the state-of-the-art NER and bioNER architectures and embeddings. CoNLL 2003 and OntoNotes5 are trained with "bert-base-cased" transformers (Devlin et al., 2019), while NCBI and s800 are trained with "biobert-base-cased-v1.1" (Lee et al., 2019) to avoid out-of-vocabulary (OOV) issues.

Low-Resource and Imbalanced Data Distribution Settings: We evaluated the performance of our proposed OVA AUC NER methods and selected baselines under the following experimental settings

- The size of training set $\mathcal{S}$: In order to simulate the low-resource scenarios, we used training set $\mathcal{S}$ with size $\in \{20, 50, 100, 200, 300, 400, 500\}$. We then trained our methods and all the baselines on 10 random training partitions of the same size and investigated their average F1-performance.
- We used an imbalance entity tag generator to sample $\mathcal{S}$ that contains $\{1, 2, 5, 10\}$ percentage of entity-type tokens, *i.e.*, tokens with labels that are not "O". By simulating for different data distributions, we can investigate the robustness of our methods under different distribution setups.

Baselines: For our baselines, the following traditional learning objective functions will be selected

- **CE**: The standard multiclass cross entropy loss, most commonly used in SOTA NER works, was used as one of the major baselines to verify and establish the significance/impact of our methods.
- **CRFs**: Representing our second major baseline, CRFs have been traditionally used in many NER works, such as those of (Lample et al., 2016; Xu et al., 2019). As CRFs produce a sequence labeller instead of a token classifier, we also used BiLSTM to push the performance of this baseline.
- **OVA-BCE**: This is our last baseline. This baseline is to indicate the ineffectiveness of binary cross entropy loss (BCE) as an objective function in OVA NER setups under the low-resource settings.

OVA AUC NER: As OVA has higher training time compared to multi-class CE since each binary classifier should be independently trained, we consider the following methods to alleviate this issue

- **OVA-AUC**: We group the binary classifiers for labels that share similar linguistic characteristics (*e.g.*, B-PER, B-ORG, B-MISC and B-LOC for CoNLL 2003) and train their classifiers together.
- **OVA-AUC-MAML**: We apply first-order meta-learning (MAML) (Finn et al., 2017; Nichol et al., 2018) and sample a random batch of m binary classifiers in each iteration for the model to learn.

Please note that our work compares between the baseline objective functions and our OVA AUC objective functions; thus, all objective functions share the same embeddings and language model, *e.g.*, Bert-CE, Bert-CRF, and Bert-BiLSTM-CRF v.s. Bert-OVA-AUC and Bert-OVA-AUC-MAML.

5 EXPERIMENTAL RESULTS & DISCUSSIONS

5.1 LOW-RESOURCE & ENSEMBLE STUDIES

Using the results from both Figure 2 and Table 2, we have the following observations:

- **CE vs. OVA-AUC**: Under extreme low-resource scenarios (*i.e.*, size $\{20, 50\}$), OVA-AUC outperforms CE by a significant margin, with the average F1-performance difference reaching 30%. When the training set size increases, OVA-AUC still exhibits substantial gains compared to CE at the 95% level of confidence the vast majority of the time. This indicates that OVA-AUC is a superior alternative to the standard multi-class CE objective function for low-resource NER scenarios.
- **CRF and BiLSTM-CRF vs. OVA-AUC**: It is apparent that OVA-AUC significantly outperforms CRF in all scenarios. As CRF is a sequence labeler, we additionally adopt BiLSTM to improve this baseline's performance. Nevertheless, based on the findings, we believe that OVA-AUC should remain a superior solution to both CRF and BiLSTM-CRF under the low-resource NER scenarios.

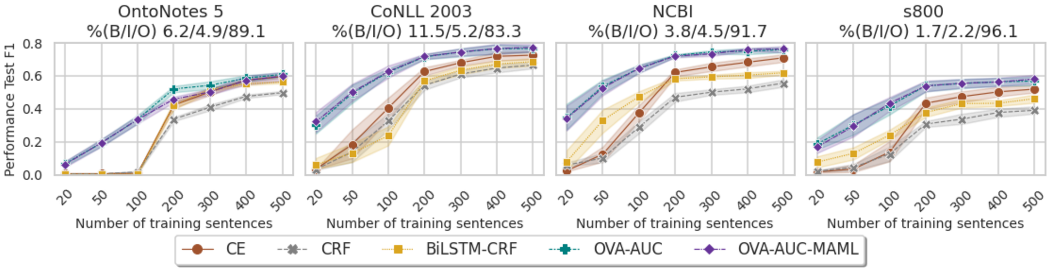


Figure 2: Learning curves of the average performance taken from 10 random training partitions of each training set $\mathcal{S}$ size for each loss/objective function. The error bands indicate the 95% confidence level of the scores. The title of the plot indicates the corpus, and the label distribution in BIO format. The embedding and language model used for CoNLL 2003 and OntoNotes5 is "bert-base-cased" (Devlin et al., 2019) while that of NCBI and s800 is "biobert-base-cased-v1.1" (Lee et al., 2019).

	Training Size	20	50	100	200	300	400	500
OntoNotes5	**CE**	0.0060$_{0.0001}$	0.0061$_{0.0001}$	0.0089$_{0.0019}$	0.4211$_{0.0073}$	0.5021$_{0.0102}$	0.5727$_{0.0048}$	0.5944$_{0.0056}$
	OVA-AUC	0.0665$_{0.0074}$	0.1926$_{0.0120}$	0.3357$_{0.0133}$	$\underline{0.5198}_{0.0099}$	0.5418$_{0.0119}$	0.5857$_{0.0092}$	0.6078$_{0.0068}$
	OVA-AUC-MAML	0.0622$_{0.0100}$	0.1934$_{0.0123}$	0.3340$_{0.0123}$	0.4573$_{0.0086}$	0.5022$_{0.0096}$	0.5722$_{0.014}$	0.5973$_{0.0134}$
	CE-ENS	0.0070$_{0.0003}$	0.0068$_{0.0002}$	0.0142$_{0.0027}$	0.4592$_{0.0081}$	0.5354$_{0.0097}$	0.5971$_{0.0052}$	0.6179$_{0.0051}$
	OVA-AUC-ENS	$\underline{0.0961}_{0.0126}$	$\underline{0.2341}_{0.0125}$	$\underline{0.3689}_{0.0140}$	**0.5237**$_{0.0100}$	**0.5765**$_{0.0084}$	**0.6067**$_{0.0070}$	**0.6266**$_{0.0051}$
	OVA-AUC-MAML-ENS	**0.0975**$_{0.0142}$	**0.2457**$_{0.0166}$	**0.3736**$_{0.0135}$	0.4941$_{0.0081}$	$\underline{0.5499}_{0.0108}$	$\underline{0.6032}_{0.0079}$	$\underline{0.6239}_{0.0053}$
CoNLL 2003	**CE**	0.0338$_{0.0003}$	0.1830$_{0.0439}$	0.4043$_{0.0339}$	0.6248$_{0.0131}$	0.6793$_{0.0094}$	0.7178$_{0.0125}$	0.7248$_{0.0127}$
	OVA-AUC	0.3017$_{0.0266}$	0.4928$_{0.0280}$	0.6218$_{0.0106}$	0.7145$_{0.0078}$	0.7436$_{0.0121}$	0.7636$_{0.0147}$	0.7650$_{0.0103}$
	OVA-AUC-MAML	0.3213$_{0.0329}$	0.4997$_{0.0268}$	0.6259$_{0.0197}$	0.7167$_{0.0083}$	0.7437$_{0.0112}$	0.7651$_{0.0148}$	0.7722$_{0.0111}$
	CE-ENS	0.0608$_{0.0050}$	0.2248$_{0.0431}$	0.4444$_{0.0292}$	0.6521$_{0.0117}$	0.7104$_{0.0130}$	0.7443$_{0.0143}$	0.7515$_{0.0139}$
	OVA-AUC-ENS	$\underline{0.3227}_{0.0957}$	**0.5308**$_{0.0213}$	**0.6466**$_{0.0101}$	$\underline{0.7176}_{0.0079}$	$\underline{0.7515}_{0.0118}$	$\underline{0.7693}_{0.0145}$	$\underline{0.7739}_{0.0117}$
	OVA-AUC-MAML-ENS	**0.3453**$_{0.0354}$	$\underline{0.5259}_{0.0250}$	$\underline{0.6381}_{0.0155}$	**0.7225**$_{0.0088}$	**0.7619**$_{0.0109}$	**0.7703**$_{0.0137}$	**0.7843**$_{0.0122}$
NCBI	**CE**	0.0233$_{0.0018}$	0.1241$_{0.0227}$	0.3746$_{0.0252}$	0.6198$_{0.0110}$	0.6536$_{0.0125}$	0.6819$_{0.0133}$	0.7089$_{0.0120}$
	OVA-AUC	0.3451$_{0.0399}$	0.5358$_{0.0218}$	0.6447$_{0.0129}$	0.7227$_{0.0042}$	0.7438$_{0.0056}$	0.7491$_{0.0089}$	0.7580$_{0.0076}$
	OVA-AUC-MAML	0.3413$_{0.0420}$	0.5267$_{0.0231}$	0.6438$_{0.0123}$	0.7188$_{0.0038}$	0.7327$_{0.0065}$	$\underline{0.7582}_{0.0046}$	0.7623$_{0.0090}$
	CE-ENS	0.0646$_{0.0020}$	0.1231$_{0.0204}$	0.4143$_{0.0245}$	0.6551$_{0.0095}$	0.6833$_{0.0070}$	0.7087$_{0.0064}$	0.7300$_{0.0076}$
	OVA-AUC-ENS	**0.3746**$_{0.0382}$	$\underline{0.5605}_{0.0221}$	**0.6661** 0.0118	**0.7322**$_{0.0046}$	$\underline{0.7501}_{0.0051}$	0.7580$_{0.0084}$	**0.7692**$_{0.0074}$
	OVA-AUC-MAML-ENS	$\underline{0.3716}_{0.0416}$	**0.5691**$_{0.0224}$	$\underline{0.6646}_{0.0083}$	$\underline{0.7320}_{0.0055}$	**0.7511**$_{0.0052}$	**0.7594**$_{0.0060}$	$\underline{0.7668}_{0.0072}$
s800	**CE**	0.0160$_{0.0032}$	0.0032$_{0.0116}$	0.1368$_{0.0302}$	0.4330$_{0.018}$	0.4734$_{0.0177}$	0.5020$_{0.0147}$	0.5189$_{0.0120}$
	OVA-AUC	0.1869$_{0.0183}$	0.3009$_{0.0331}$	0.4169$_{0.0250}$	0.5387$_{0.0152}$	0.5532$_{0.0147}$	0.5634$_{0.0123}$	0.5677$_{0.0137}$
	OVA-AUC-MAML	0.1679$_{0.0204}$	0.2941$_{0.0358}$	0.4331$_{0.0213}$	0.5384$_{0.0171}$	0.5543$_{0.0139}$	0.5619$_{0.0132}$	$\underline{0.5802}_{0.0133}$
	CE-ENS	0.0224$_{0.0012}$	0.0347$_{0.0107}$	0.1423$_{0.0337}$	0.4504$_{0.0153}$	0.4841$_{0.0188}$	0.5085$_{0.0143}$	0.5256$_{0.0128}$
	OVA-AUC-ENS	**0.1872**$_{0.0179}$	**0.3238**$_{0.0304}$	$\underline{0.4447}_{0.0228}$	**0.5420**$_{0.0152}$	$\underline{0.5595}_{0.0151}$	$\underline{0.5706}_{0.0119}$	0.5801$_{0.0130}$
	OVA-AUC-MAML-ENS	$\underline{0.1863}_{0.0218}$	$\underline{0.3342}_{0.0304}$	**0.4558**$_{0.0252}$	$\underline{0.5407}_{0.0147}$	**0.5628**$_{0.0152}$	**0.5741**$_{0.0130}$	**0.5882**$_{0.0134}$

Table 2: Average test F1-performance taken from 10 random partitions of different training set sizes for different corpora. The non-parametric bootstrapped standard errors from these experiments are under-scripted. The best performance for each setting is bold while the second best is underlined.

- **OVA-AUC vs. OVA-AUC-MAML**: Although the F1-score of OVA-AUC can be higher on average than that of OVA-AUC-MAML under some settings, the difference between the two approaches is not significant, except for OntoNotes5 with 200 training sentence size. As OVA-AUC groups the labels that share similar linguistic characteristics (*e.g.*, B-PER, B-ORG) and trains their classifiers together, the difference can be attributed to longer training time as shown in Table 6 in the Appendix. Since OVA-AUC-MAML performs on par with OVA-AUC, it consequently outperforms all the baselines in most low-resource NER settings across all our benchmark corpora.

From the literature, AUC can be further enhanced via compositional training by alternating between the standard multi-class cross entropy loss function and the AUC loss function during training as proven by Yuan et al. (2021b). Consequently, we also apply compositional training in our OVA NER methods (COMAUC) and provide the results for these experiments in the Appendix, Figure 5.

Motivated by Theorem 1, we generate the posterior distribution via deep ensemble to derive the optimal ranking function $h(\mathbf{x})$ for the binary classifiers. We show the empirical results in Table 2, from which we can observe that knowing the posterior distribution indeed improves the performance of the individual classifiers, raising the NER model prediction performance. As the number of ensembles used is 5, more performance gain can be obtained by increasing the size of ensembles.

In our results, most of the largest gains from our OVA AUC NER methods compared to the baselines are observed when the training size is of {20,50,100} and as the size increases, the performance difference begins to shrink. As mentioned in subsection 3.2, an asymptotically consistent estimator can generate an approximately optimal ranking function and no AUC optimization is required. Further-

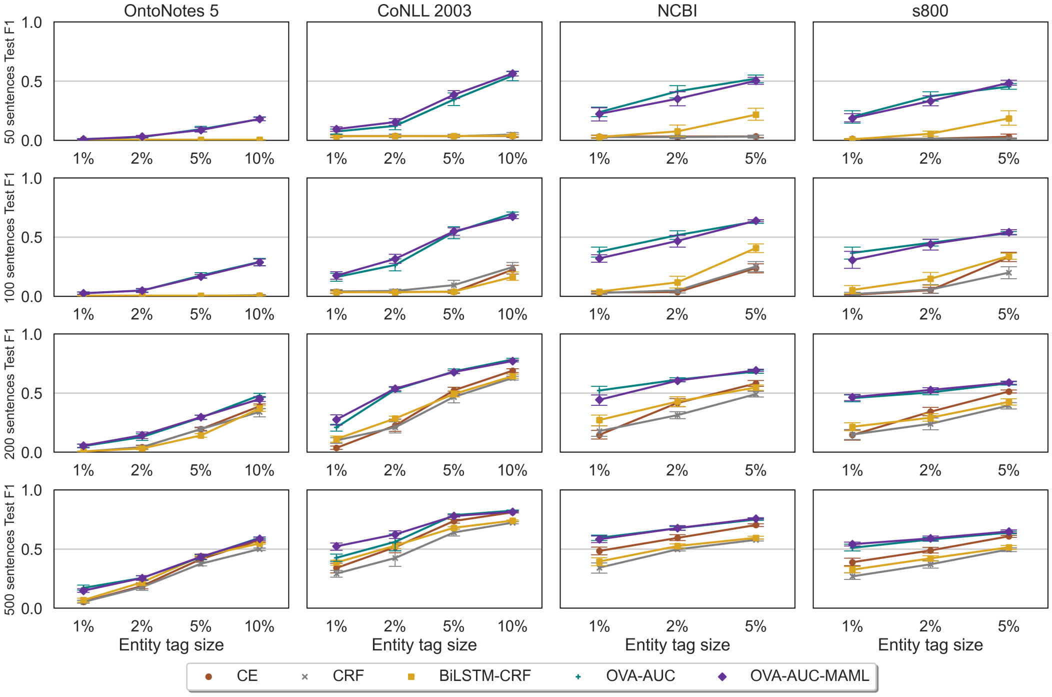


Figure 3: Average performance taken from 10 random training partitions of each training set $\mathcal{S}$ size for each loss/objective function. The entity tag size represents the percentage of entity-tokens to the total number of tokens in the training set $\mathcal{S}$. The error bars indicate the 95% confidence level of the scores. The embedding and language model for CoNLL 2003 and OntoNotes5 is "bert-base-cased" (Devlin et al., 2019) while that for NCBI and s800 is "biobert-base-cased-v1.1" (Lee et al., 2019).

more, on the basis of the F1-performance on CoNLL 2003, similar observations can be made for ensemble training where no noticeable differences can be observed for our OVA AUC NER methods at size of 500 sentences. As OntoNotes5 is a harder corpus to learn (37 classes for OntoNotes5 v.s. 9 classes for CoNLL 2003, see Table 4), we surmise that it would take a bigger training data for us to see no noticeable differences between ensemble and non-ensemble training. Overall, we believe the results give evidence that both AUC and ensemble training are important to derive the optimal classifiers under the low-resource scenarios.

5.2 Imbalanced Data Distribution Studies

We deployed an imbalance entity tag generator (subsection A.5) to demonstrate the robustness of our methods on the diverse training sets for the NER task. The generator simulates scenarios in which the training set $\mathcal{S}$ data distribution changes from that of $\mathcal{S}^{\text{test}}$ in order to test the resilience of the baselines and our methods. Figure 3 illustrates the performance differences for those methods according to the size of the entity tag. From these results, we provide the following observations:

- **CE vs. OVA-AUC**: Across all imbalanced setting of entity tags, we observed that OVA-AUC outperforms CE. On the basis of the F1-performance on CoNLL, NCBI, and s800, OVA-AUC is significantly superior to CE in the most extreme imbalanced scenarios (*i.e.*, entity label size of 1 and 2%). As the amount of entity tags rises in OntoNotes5, OVA-AUC performance improves greatly. On the other side, CE performs inadequately when the entity tag size is extremely small.
- **CRF and BiLSTM-CRF vs. OVA-AUC**: OVA-AUC outperforms CRF and BiLSTM-CRF substantially in all scenarios. Similar to CE, neither CRF nor BiLSTM performs effectively when the entity tag size is extremely small as the sequence labelers are not fed with enough learning signals.
- **OVA-AUC vs. OVA-AUC-MAML**: We observe no significant difference between OVA-AUC and OVA-AUC-MAML based on the F1-performance in most settings across all datasets. Although the F1-performance of OVA-AUC appears to be higher than that of OVA-AUC-MAML on CoNLL for the 50 and 100 sentences, OVA-AUC-MAML performs better on NCBI for the same scenario.

	OVA-AUC						**OVA-BCE**					
	Confusion area			**Non-confusion area**			**Confusion area**			**Non-confusion area**		
	B	**I**	**O**	**B**	**I**	**O**	**B**	**I**	**O**	**B**	**I**	**O**
# incorrect predictions	69	53	86	288	148	461	222	2,431	35,081	0	0	0
# correct predictions	70	43	167	5,213	2,214	37,564	5,418	27	3,197	0	0	0
# total predictions	**139**	**96**	**253**	**5,501**	**2,362**	**38,025**	**5,640**	**2,458**	**38,278**	**0**	**0**	**0**

	OVA-AUC ensembles						**OVA-BCE ensembles**					
	Confusion area			**Non-confusion area**			**Confusion area**			**Non-confusion area**		
	B	**I**	**O**	**B**	**I**	**O**	**B**	**I**	**O**	**B**	**I**	**O**
# incorrect predictions	56	75	32	329	222	306	1,572	2,264	33,514	0	0	0
# correct predictions	51	57	75	5,204	2,104	37,865	4,068	194	4,764	0	0	0
# total predictions	**107**	**132**	**107**	**5,533**	**2,326**	**38,171**	**5,640**	**2,458**	**38,278**	**0**	**0**	**0**

Table 3: Confusion matrix on test data for CoNLL 2003, both our OVA-AUC and OVA-BCE are trained with 100 random sentences. We also include the ensemble performance for both approaches.

5.3 OVA-AUC as an Effective Solution to NER

To properly assess the effectiveness of our OVA-AUC compared to the traditional OVA-BCE, we transcribe Figure 1 into Table 3 and analyze the statistical results. We define the confusion areas under the OVA framework as when **(i)** two or more binary classifiers are confident ($p > 0.5$) that the sample belong to their classes, or **(ii)** no classifiers are confident enough to claim the sample ($p < 0.5$). Both OVA-BCE and OVA-BCE with ensemble show weak performance as the classifiers are always confused when making decision. Even though $\Pr\left(y \equiv \hat{y} \mid \text{Confusion}\right) = \frac{\text{\# Correct Predictions}}{\text{\# Total Confusion}}$ increase from 0.186 for non-ensemble training to 0.195 for ensemble training, this performance is still far below what should be, training with the standard multi-class cross entropy learning objective.

On the other hands, not only does OVA-AUC have a better average test F1-performance compared to that of the standard CE (see Figure 2 and Table 2), it also eliminates most of the weaknesses that is natural to OVA. The probability that any predictions in the confusion area is only 0.0105 for normal training and 0.0075 for ensemble training, meaning that the binary classifiers are less likely to be confused. Furthermore, $\Pr\left(y \equiv \hat{y} \mid \text{Confusion}\right)$ also increases to 0.5738 and 0.5289 for normal and ensemble training respectively. Although the ensemble OVA-AUC $\Pr\left(y \equiv \hat{y} \mid \text{Confusion}\right)$ is down from 0.5738, we surmise this happens as the number of confusion points is reduced, leaving only the hard-to-classify tokens in the confusion area. Additionally, we observe for the non confusion area that $\Pr\left(y \equiv \hat{y} \mid \text{Non-Confusion}\right) = \frac{\text{\# Correct Predictions}}{\text{\# Total Non-Confusion}}$ is 0.9805 and 0.9814 for the normal and ensemble training respectively, suggesting that the binary classifiers, trained with the AUC surrogate loss, are well-tuned. Lastly, we believe that future improvements to OVA AUC NER methods can be made by focusing on the hardest label set in the corpus, those that start with "I", as it has the highest confusion and error rate out of all label sets for both our OVA-AUC and OVA-BCE implementations.

Overall, we observe that the ensemble performance for both OVA-AUC and OVA-BCE are better compared to their non-ensemble counterparts. This supports Theorem 1 in that optimal ranking occurs with ensembles, regardless of the estimator.

6 Conclusion

In this paper, we provide an effective solution to the low-resource and imbalanced data difficulties that afflict many NER/BioNER tasks. To address these two problems, we first reformulated the traditional NER multi-class learning problem as a one-vs-all learning problem and then used an AUC surrogate loss to train the binary classifiers. Extensive experiments on multiple datasets in different scenarios, reflecting the low-resource and the data imbalance challenges, demonstrated that our OVA AUC NER approaches perform significantly better than the generally used CE and CRF, independent of the underlying NER models, embeddings, or domains being used. Moreover, our Bayesian theory of optimal AUC mutually reinforces the result that ensembling improves AUC, and the benefit of any special purpose AUC training should only be substantial in the low data setting.

Of our approaches, OVA AUC NER with meta-learning gives significantly better results with comparable training time to existing approaches. There are still some limitations, among which are the confusion regions resulting from OVA training. We consider the one-vs-one (OVO) setting might serve as an alternative, which is subject to further development due to OVO high time complexity.

7 Reproducibility Statement

For our reproducibility statement, we use the "The Machine Learning Reproducibility Checklist".

1. **For all models and algorithms presented, check if you include:**
   (a) A clear description of the mathematical setting, algorithm, and/or model. [Yes] These are listed in the Table 1, in subsection A.3 and the theorem in section 3.
   (b) An analysis of the complexity (time, space, sample size) of any algorithm. [Yes] These are listed in the Table 1 and in Table 4 in Appendix.
2. **For any theoretical claim, check if you include:**
   (a) A clear statement of the claim. [Yes] These are in section 3.
   (b) A complete proof of the claim. [Yes] These are in subsection 3.2.
3. **For all datasets used, check if you include:**
   (a) The relevant statistics, such as number of examples. [Yes] Table 1 and Table 4 in Appendix.
   (b) The details of train / validation / test splits. [Yes] Table 1.
   (c) An explanation of any data that were excluded, and all pre-processing step. [Yes] There are listed in README.MD of the zip file.
   (d) A link to a downloadable version of the dataset or simulation environment. [Partial] The link will be on Github once the paper is published. CoNLL, NCBI and s800 are uploaded into our Github link. We can not upload the OntoNotes 5 into our Github's repository since it has the copyright owned by Linguistic Data Consortium. However, we provide the script to generate the experimental dataset of OntoNotes 5 for reproducibility purposes.
   (e) For new data collected, a complete description of the data collection process, such as instructions to annotators and methods for quality control. [No] We did not collect the new data, only used the existing partially benchmark datasets.
4. **For all shared code related to this work, check if you include:**
   (a) Specification of dependencies. [Yes] There are listed in the requirements.txt of the zip file.
   (b) Training code. [Yes] There are listed in losses.py, optimizers.py, prepro.py and train.py of the zip file.
   (c) Evaluation code. [Yes] There are listed in the train.py of the zip file.
   (d) (Pre-)trained model(s). [Yes] There are listed in the losses.py, model.py and train.py of the zip file.
   (e) README file includes table of results accompanied by precise command to run to produce those results. [Partial] It is listed in README.MD of the zip file.
5. **For all reported experimental results, check if you include:**
   (a) The range of hyper-parameters considered, method to select the best hyper-parameter configuration, and specification of all hyper-parameters used to generate results. [Yes] There are listed in section 4, subsection A.5 and subsection A.2.
   (b) The exact number of training and evaluation runs. [Yes] There are listed in subsection A.2.
   (c) A clear definition of the specific measure or statistics used to report results. [Yes] There are listed in section 5 and subsection A.6.
   (d) A description of results with central tendency (e.g. mean) & variation (e.g. error bars). [Yes] We use the average results from certain runs of the experiments. All results are plotted with error bars/ error bands.
   (e) The average runtime for each result, or estimated energy cost. [Partial] There are listed in subsection A.4.
   (f) A description of the computing infrastructure used. [Yes] There are listed in section 4.

References

Partha Sarathy Banerjee, Baisakhi Chakraborty, Deepak Tripathi, Hardik Gupta, and Sourabh S Kumar. A information retrieval based on question and answering and NER for unstructured information without using SQL. *Wireless Personal Communications*, 108(3):1909–1931, 2019.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 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)*, pp. 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL `https://aclanthology.org/N19-1423`.

Rezarta Islamaj Doğan, Robert Leaman, and Zhiyong Lu. Special report: NCBI disease corpus: A resource for disease name recognition and concept normalization. *J. of Biomedical Informatics*, 47:1–10, February 2014. ISSN 1532-0464.

Oren Etzioni, Michael Cafarella, Doug Downey, Ana-Maria Popescu, Tal Shaked, Stephen Soderland, Daniel S Weld, and Alexander Yates. Unsupervised named-entity extraction from the web: An experimental study. *Artificial intelligence*, 165(1):91–134, 2005.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, pp. 1126–1135. PMLR, 2017.

Yoav Freund, Raj Iyer, Robert E Schapire, and Yoram Singer. An efficient boosting algorithm for combining preferences. *Journal of machine learning research*, 4(Nov):933–969, 2003.

Mikel Galar, Alberto Fernández, Edurne Barrenechea, Humberto Bustince, and Francisco Herrera. An overview of ensemble methods for binary classifiers in multi-class problems: Experimental study on one-vs-one and one-vs-all schemes. *Pattern Recognition*, 44(8):1761–1776, 2011. ISSN 0031-3203. doi: https://doi.org/10.1016/j.patcog.2011.01.017. URL `https://www.sciencedirect.com/science/article/pii/S0031320311000458`.

Wei Gao and Zhi-Hua Zhou. On the consistency of AUC pairwise optimization. In *Proceedings of the 24th International Conference on Artificial Intelligence*, pp. 939–945. AAAI Press, 2015.

Wei Gao, Rong Jin, Shenghuo Zhu, and Zhi-Hua Zhou. One-pass AUC optimization. In *International conference on machine learning*, pp. 906–914. PMLR, 2013.

John M Giorgi and Gary D Bader. Towards reliable named entity recognition in the biomedical domain. *Bioinformatics*, 36(1):280–286, 06 2019. ISSN 1367-4803. doi: 10.1093/bioinformatics/btz504. URL `https://doi.org/10.1093/bioinformatics/btz504`.

Tilmann Gneiting and Adrian E Raftery. Strictly proper scoring rules, prediction, and estimation. *J. Amer. Statistical Assoc.*, 102(477):359–378, 2007.

Mengda Huang, Yang Liu, Xiang Ao, Kuan Li, Jianfeng Chi, Jinghua Feng, Hao Yang, and Qing He. AUC-oriented graph neural network for fraud detection. In *Proceedings of the ACM Web Conference 2022*, pp. 1311–1321, 2022.

Jaeyeon Jang and Chang Ouk Kim. One-vs-rest network-based deep probability model for open set recognition. *CoRR*, abs/2004.08067, 2020. URL `https://arxiv.org/abs/2004.08067`.

Wojciech Kotlowski, Krzysztof Dembczynski, and Eyke Huellermeier. Bipartite ranking through minimization of univariate loss. In *ICML*, 2011.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, pp. 6405–6416, 2017.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 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*, pp. 260–270, San Diego, California, June 2016. Association for Computational Linguistics. doi: 10.18653/v1/N16-1030. URL `https://www.aclweb.org/anthology/N16-1030`.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. BioBERT: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240, 09 2019. ISSN 1367-4803. doi: 10.1093/bioinformatics/btz682. URL `https://doi.org/10.1093/bioinformatics/btz682`.

J. Li, A. Sun, J. Han, and C. Li. A survey on deep learning for named entity recognition. *IEEE Transactions on Knowledge and Data Engineering*, pp. 1–1, 2020.

Jing Li, Shuo Shang, and Ling Shao. Metaner: Named entity recognition with meta-learning. In *Proceedings of The Web Conference 2020*, WWW '20, pp. 429–440, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450370233. doi: 10.1145/3366423.3380127. URL `https://doi.org/10.1145/3366423.3380127`.

Hui Liu, Wengming Zheng, Gaopeng Sun, Yanhua Shi, Yue Leng, Pan Lin, Ruimin Wang, Yuankui Yang, Jun-feng Gao, Haixian Wang, Keiji Iramina, and Sheng Ge. Action understanding based on a combination of one-versus-rest and one-versus-one multi-classification methods. In *2017 10th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)*, pp. 1–5, 2017a. doi: 10.1109/CISP-BMEI.2017.8302159.

Hui Liu, Wengming Zheng, Gaopeng Sun, Yanhua Shi, Yue Leng, Pan Lin, Ruimin Wang, Yuankui Yang, Jun-feng Gao, Haixian Wang, et al. Action understanding based on a combination of one-versus-rest and one-versus-one multi-classification methods. In *2017 10th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)*, pp. 1–5. IEEE, 2017b.

Max Lübbering, Michael Gebauer, Rajkumar Ramamurthy, Christian Bauckhage, and Rafet Sifa. Decoupling autoencoders for robust one-vs-rest classification. In *2021 IEEE 8th International Conference on Data Science and Advanced Analytics (DSAA)*, pp. 1–10, 2021. doi: 10.1109/DSAA53316.2021.9564136.

Alex Nichol, Joshua Achiam, and John Schulman. On first-order meta-learning algorithms. *CoRR*, abs/1803.02999, 2018. URL `http://arxiv.org/abs/1803.02999`.

Shreyas Padhy, Zachary Nado, Jie Ren, Jeremiah Z. Liu, Jasper Snoek, and Balaji Lakshminarayanan. Revisiting one-vs-all classifiers for predictive uncertainty and out-of-distribution detection in neural networks. *CoRR*, abs/2007.05134, 2020. URL `https://arxiv.org/abs/2007.05134`.

Evangelos Pafilis, Sune P. Frankild, Lucia Fanini, Sarah Faulwetter, Christina Pavloudi, Aikaterini Vasileiadou, Christos Arvanitidis, and Lars Juhl Jensen. The species and organisms resources for fast and accurate identification of taxonomic names in text. *PLOS ONE*, 8(6):1–6, 06 2013. doi: 10.1371/journal.pone.0065390. URL `https://doi.org/10.1371/journal.pone.0065390`.

Baolin Peng, Chunyuan Li, Jinchao Li, Shahin Shayandeh, Lars Liden, and Jianfeng Gao. Soloist: Building task bots at scale with transfer learning and machine teaching. *Transactions of the Association for Computational Linguistics*, 9:807–824, 2021. doi: 10.1162/tacl_a_00399. URL `https://aclanthology.org/2021.tacl-1.49`.

Ryan Rifkin and Aldebaro Klautau. In defense of one-vs-all classification. *The Journal of Machine Learning Research*, 5:101–141, 2004.

Alan Ritter, Oren Etzioni, and Sam Clark. Open domain event extraction from Twitter. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 1104–1112, 2012.

Kuniaki Saito and Kate Saenko. Ovanet: One-vs-all network for universal domain adaptation. *CoRR*, abs/2104.03344, 2021. URL `https://arxiv.org/abs/2104.03344`.

Erik F. Tjong Kim Sang and Fien De Meulder. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pp. 142–147, 2003. URL `https://www.aclweb.org/anthology/W03-0419`.

Ralph Weischedel, Sameer Pradhan, Lance Ramshaw, Martha Palmer, Nianwen Xue, Mitchell Marcus, Ann Taylor, Craig Greenberg, Eduard Hovy, Robert Belvin, et al. Ontonotes release 5.0. *LDC2011T03, Philadelphia, Penn.: Linguistic Data Consortium*, 2014.

Kai Xu, Zhenguo Yang, Peipei Kang, Qi Wang, and Wenyin Liu. Document-level attention-based BiLSTM-CRF incorporating disease dictionary for disease named entity recognition. *Computers in Biology and Medicine*, 108:122 – 132, 2019. ISSN 0010-4825. doi: https://doi.org/10.1016/j.compbiomed.2019.04.002. URL `http://www.sciencedirect.com/science/article/pii/S0010482519301106`.

Zhiyong Yang, Qianqian Xu, Shilong Bao, Xiaochun Cao, and Qingming Huang. Learning with multiclass AUC: Theory and algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

Usama Yaseen and Stefan Langer. Data augmentation for low-resource named entity recognition using backtranslation. *ArXiv*, abs/2108.11703, 2021.

Yiming Ying, Longyin Wen, and Siwei Lyu. Stochastic online AUC maximization. *Advances in neural information processing systems*, 29, 2016.

Yuan et al. LibAUC: A deep learning library for x-risk optimization, 2022.

Z. Yuan, Y. Yan, M. Sonka, and T. Yang. Large-scale robust deep AUC maximization: A new surrogate loss and empirical studies on medical image classification. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 3020–3029, Los Alamitos, CA, USA, oct 2021a. IEEE Computer Society. doi: 10.1109/ICCV48922.2021.00303. URL `https://doi.ieeecomputersociety.org/10.1109/ICCV48922.2021.00303`.

Zhuoning Yuan, Zhishuai Guo, Nitesh Chawla, and Tianbao Yang. Compositional training for end-to-end deep AUC maximization. In *International Conference on Learning Representations*, 2021b.

Peilin Zhao, Steven C. H. Hoi, Rong Jin, and Tianbao Yang. Online AUC maximization. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, ICML'11, pp. 233–240, Madison, WI, USA, 2011. Omnipress. ISBN 9781450306195.

Ran Zhou, Xin Li, Ruidan He, Lidong Bing, Erik Cambria, Luo Si, and Chunyan Miao. MELM: Data augmentation with masked entity language modeling for low-resource NER. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2251–2262, 2022.

Keneilwe Zuva and Tranos Zuva. Evaluation of information retrieval systems. *AIRCC's International Journal of Computer Science and Information Technology*, 4(3):35–43, 2012.

A APPENDIX

A.1 DETAILED LABEL DISTRIBUTION

Dataset	# Label	and size(%)		
		Train	Dev	Test
OntoNotes5	B-CARDINAL/ B-PERSON/ I-PERSON/ B-GPE/ O	0.54/ 0.95/ 0.50/ 1.05/ 88.98	0.51/ 1.0/ 0.51/ 1.08/ 88.97	0.53/ 0.98/ 0.47/ 1.04/ 89.04
	I-GPE/ B-NORP/ I-DATE/ B-DATE/ I-PRODUCT	0.37/ 0.47/ 0.97/ 0.95/ 0.03	0.36/ 0.44/ 0.92/ 0.95/ 0.03	0.37/ 0.44/ 1.03/ 0.95/ 0.02
	B-EVENT/ I-EVENT/ B-ORG/ I-ORG/ B-PRODUCT	0.05/ 0.09/ 1.18/ 1.47/ 0.05	0.06/ 0.11/ 1.17/ 1.48/ 0.04	0.04/ 0.08/ 1.14/ 1.39/ 0.05
	B-FAC/ B-ORDINAL/ B-TIME/ I-TIME/ I-NORP	0.05/ 0.11/ 0.09/ 0.08/ 0.04	0.05/ 0.11/ 0.09/ 0.10/ 0.01	0.05/ 0.12/ 0.09/ 0.10/ 0.03
	B-LOC/ I-FAC/ I-CARDINAL/ B-MONEY/ I-MONEY	0.10/ 0.07/ 0.15/ 0.24/ 0.48	0.08/ 0.10/ 0.14/ 0.24/ 0.39	0.11/ 0.08/ 0.17/ 0.23/ 0.45
	I-LOC/ B-PERCENT/ I-PERCENT/ B-WORK OF ART	0.09/ 0.18/ 0.24/ 0.07	0.06/ 0.18/ 0.25/ 0.07	0.11/ 0.22/ 0.32/ 0.05
	B-QUANTITY/ I-QUANTITY/ B-LAW/ B-LANGUAGE	0.06/ 0.08/ 0.02/ 0.02	0.06/ 0.08/ 0.03/ 0.01	0.04/ 0.06/ 0.02/ 0.02
	I-ORDINAL/ I-LANGUAGE/ I-WORK OF ART/ I-LAW	0.00/ 0.00/ 0.15/ 0.05	0.00/ 0.00/ 0.22/ 0.06	0.00/ 0.00/ 0.10/ 0.04
CoNLL 2003	B-MISC/ I-MISC/ B-PER/ I-PER	1.7/ 0.6/ 3.2/ 2.2	1.8/ 0.7/ 3.6/ 2.5	1.5/ 0.5/ 3.5/ 2.5
	B-ORG/ I-ORG/ B-LOC/ I-LOC/ O	3.1/ 1.8/ 3.5/ 0.6/ 83.3	2.6/ 1.5/ 3.6/ 0.5/ 83.3	3.6/ 1.8/ 3.6/ 0.6/ 82.5

Table 4: The detail of label distribution for benchmark datasets

Table 4 gives detailed label distribution for OntoNotes5 (Weischedel et al., 2014) and CoNLL 2003 (Tjong Kim Sang & De Meulder, 2003). As OntoNotes5 has 37 unique labels, the label distribution can be quite sparse, presenting challenges to the traditional multi-class NER objective functions.

A.2 HYPERPARAMETER SETTINGS

Hyperparameter	Settings	Hyperparameter	Settings
Maximum Sequence Length	128	Drop-out Probability	1e-1
Number of Epochs	100	Weight decay	1e-4
Batch Size	64	Epsilon	1e-6
Learning Rate (LR) (CE, CRF)	1e-5	LR (BiLSTM-CRF)	3e-4
LR (AUC/COMAUC)	1e-1	Margin (AUC/COMAUC)	1

Table 5: Hyperparameter settings for low-resource and imbalanced data distribution experiments.

In this sub-section, we provide the hyperparameter settings to reproduce our works. These settings, listed in Table 5, are obtained from a grid search to find the optimal values. The optimizer for our baselines is AdamW, while that of our OVA AUC NER methods is PESG Yuan et al. (2022) All experiments use PyTorch [1] and run on Intel Core i9 Processors CPU and Nvidia RTX 3090 GPUs.

A.3 OVA AUC NER TRAINING ALGORITHMS

Algorithm 1 OVA-AUC

Input: Training set $\mathcal{S}$, $\theta, \{\omega_1, \ldots, \omega_K\}$, $\{a, b, \alpha\}^K$
Output: $\theta^*, \{\omega_1^*, \ldots, \omega_K^*\}$

```
for epoch in range(num epochs) do
    for prefix in {B,I, O} do
        $\mathcal{L} := 0$
        for $i$ in range(K) do
            if i.startswith(prefix) then
                $\mathcal{L} += $ equation 1
            end if
        end for
        $\mathcal{L}$.optimize()
    end for
end for
```

Algorithm 2 OVA-AUC-MAML First Order Approximation

Input: Training set $\mathcal{S}$, $\theta, \{\omega_1, \ldots, \omega_K\}$, $\{a, b, \alpha\}^K$
Output: $\theta^*, \{\omega_1^*, \ldots, \omega_K^*\}$

```
for epoch in range(num epochs) do
    $\mathcal{L} := 0$
    sample m classes
    for $i$ in range(K) do
        if $i$ in m then
            $\mathcal{L} += $ equation 1
        end if
    end for
    $\mathcal{L}$.optimize()
end for
```

[1]
```
pip3 install torch==1.9.1+cu111 torchvision==0.10.1+cu111
torchaudio==0.9.1 -f https://download.pytorch.org/whl/torch_stable.html
```

A.4 Report on Training times

	CoNLL 2003					OntoNotes5				
Number of Training Sentences	**50**	**100**	**200**	**300**	**400**	**50**	**100**	**200**	**300**	**400**
CE	849	852	888	907	926	784	792	830	844	876
CRF	897	917	961	974	1016	1,425	1,497	1,539	1,587	1,674
OVA-AUC	1,758	1,770	1,807	1,845	1,906	2,374	2,390	2,530	2,559	2,694
OVA-AUC-MAML	1,692	1,714	1,780	1,830	1,902	1,500	1,532	1,608	1,646	1,744

Table 6: Highest training time recorded for each method measured in second(s) across different sentence levels on CoNLL 2003 and OntoNotes5

A.5 Imbalance Entity Tag Generator

Algorithm 3 Imbalance Entity Tag Generator

Input: $\mathcal{S}_{\text{ne}}$ is a subset of sentence sets including only non-entity tags $\{\text{O}\}$. $\mathcal{S}_{\text{e}}$ is a subset of sentence sets including any of entity tags from $\{\text{B, I}\}$. N_{tot} is total number of sentences for output set $\mathcal{S}_{\text{out}}$. P_{pref} is a pre-defined entity label size. M_{iter} is the maximum number of iterations for the greedy search. N_{sd} is the input seed number.
Output: $\mathcal{S}_{\text{out}}$

```
random.seed($\text{N}_{\text{sd}}$)
for $\text{i} = 1 \ldots \text{M}_{\text{iter}}$ do
    for $\text{k} = 1 \ldots \text{N}_{\text{tot}}$ do
        sample $k$ sentences from $\mathcal{S}_{\text{e}}$ into $\mathcal{S}_{\text{en}}$
        sample $\text{N}_{\text{tot}} - \text{k}$ sentences from $\mathcal{S}_{\text{ne}}$ into $\mathcal{S}_{\text{nen}}$
        $\mathcal{S}_{\text{out}} \equiv \{\mathcal{S}_{\text{ne}}, \mathcal{S}_{\text{nen}}\}$
        if $\text{P}_{\text{pref}} \approx \frac{\text{\# of B, and I tokens in } \mathcal{S}_{\text{out}}}{\text{\# tokens in } \mathcal{S}_{\text{out}}}$ then
            return $\mathcal{S}_{\text{out}}$
        end if
    end for
end for
```

We use an imbalanced entity tag generator and generate the imbalanced training set $\mathcal{S}$ with specific percentage of the entity labels to evaluate the resilience of both the baselines and our methods. For each corpus from Table 1 and Table 4, we choose 4 specific training set sizes $\{50, 100, 200, 500\}$, and for each training set size, we generate the training set $\mathcal{S}$ with certain percentages of entity label, *e.g.*, 1% entity label size. Algorithm 3 presents the greedy procedure to generate the training set $\mathcal{S}$.

A.6 Miscellaneous Results

Since F1-score is not the only metrics to measure NER model performance, we also included the precision and recall performance for both our OVA AUC NER methods and the baselines, this is shown in Figure 4. It is not surprising to see that our methods outperform all the baselines on average for both metrics as they are already substantially better than the baselines on the basis of F1-performance. Moreover, it is important to note that our methods always outperform the baselines on the basis of the recall performance, meaning that the classifier, trained with the AUC surrogate loss, are better at picking out the entity-type tokens, signifying the importance of AUC loss function dealing with the imbalanced distribution difficulties.

Other results include:

- Low-resource studies from compositional training perspective, illustrated by Figure 5. As compositional training benefits from both AUC and CE, it is unsurprisingly better than all the baselines.
- Precision and Recall score for imbalanced distribution studies, shown in Figure 6 and Figure 7

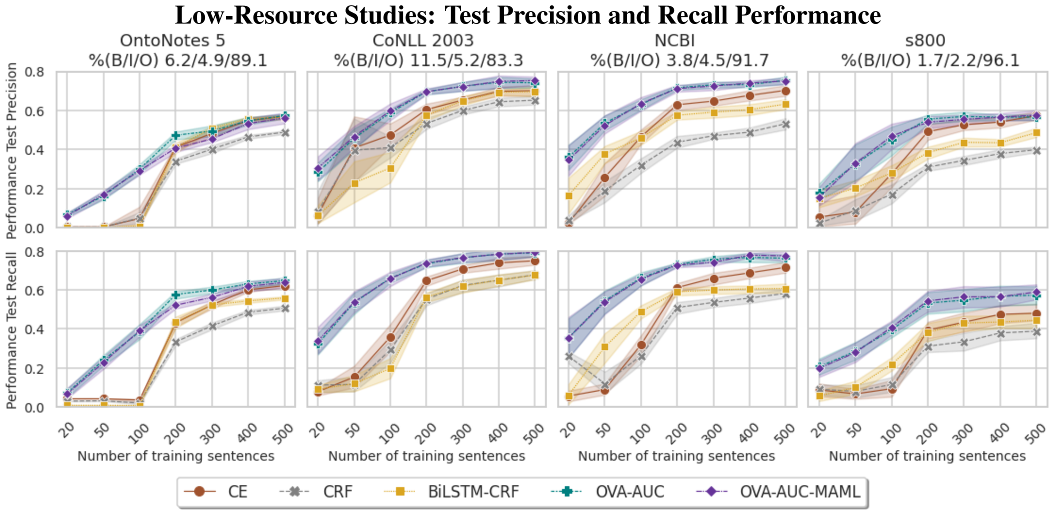


Figure 4: Learning curves of the average performance taken from 10 random training partitions of each training set $\mathcal{S}$ size for each loss/objective function. The error bands indicate the 95% confidence level of the scores. The title of the plot indicates the corpus, and the label distribution in BIO format. The embedding and language model used for CoNLL 2003 and OntoNotes5 is “bert-base-cased” (Devlin et al., 2019) while that of NCBI and s800 is “biobert-base-cased-v1.1” (Lee et al., 2019).

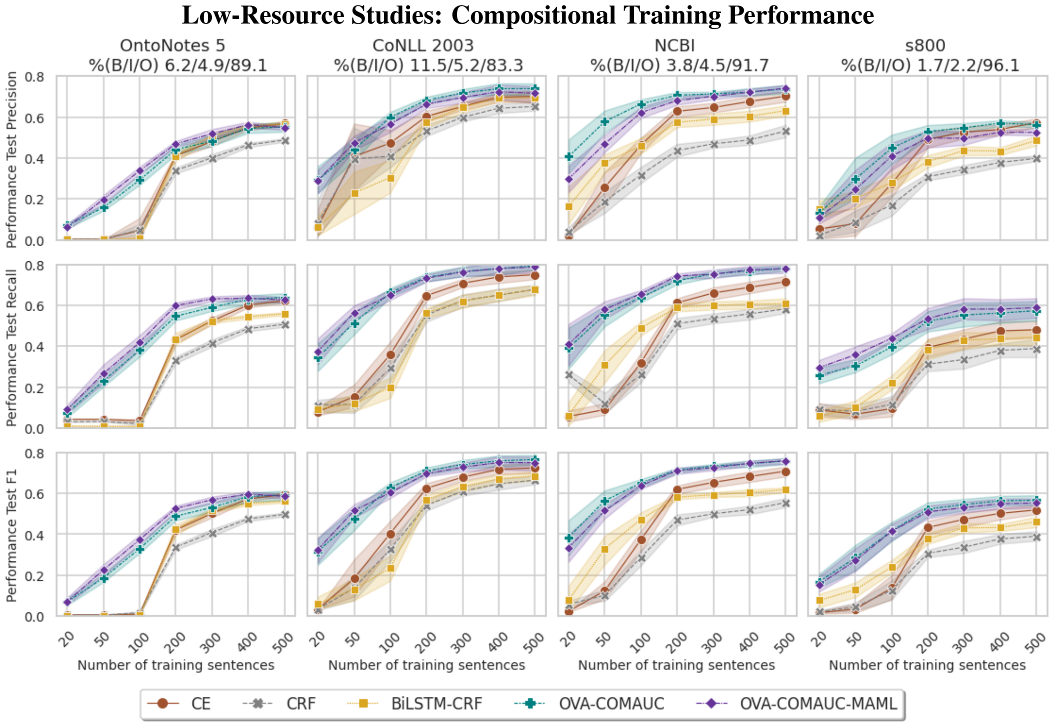


Figure 5: Learning curves of the average performance taken from 10 random training partitions of each training set $\mathcal{S}$ size for each loss/objective function. The error bands indicate the 95% confidence level of the scores. The title of the plot indicates the corpus, and the label distribution in BIO format. The embedding and language model used for CoNLL 2003 and OntoNotes5 is “bert-base-cased” (Devlin et al., 2019) while that of NCBI and s800 is “biobert-base-cased-v1.1” (Lee et al., 2019).

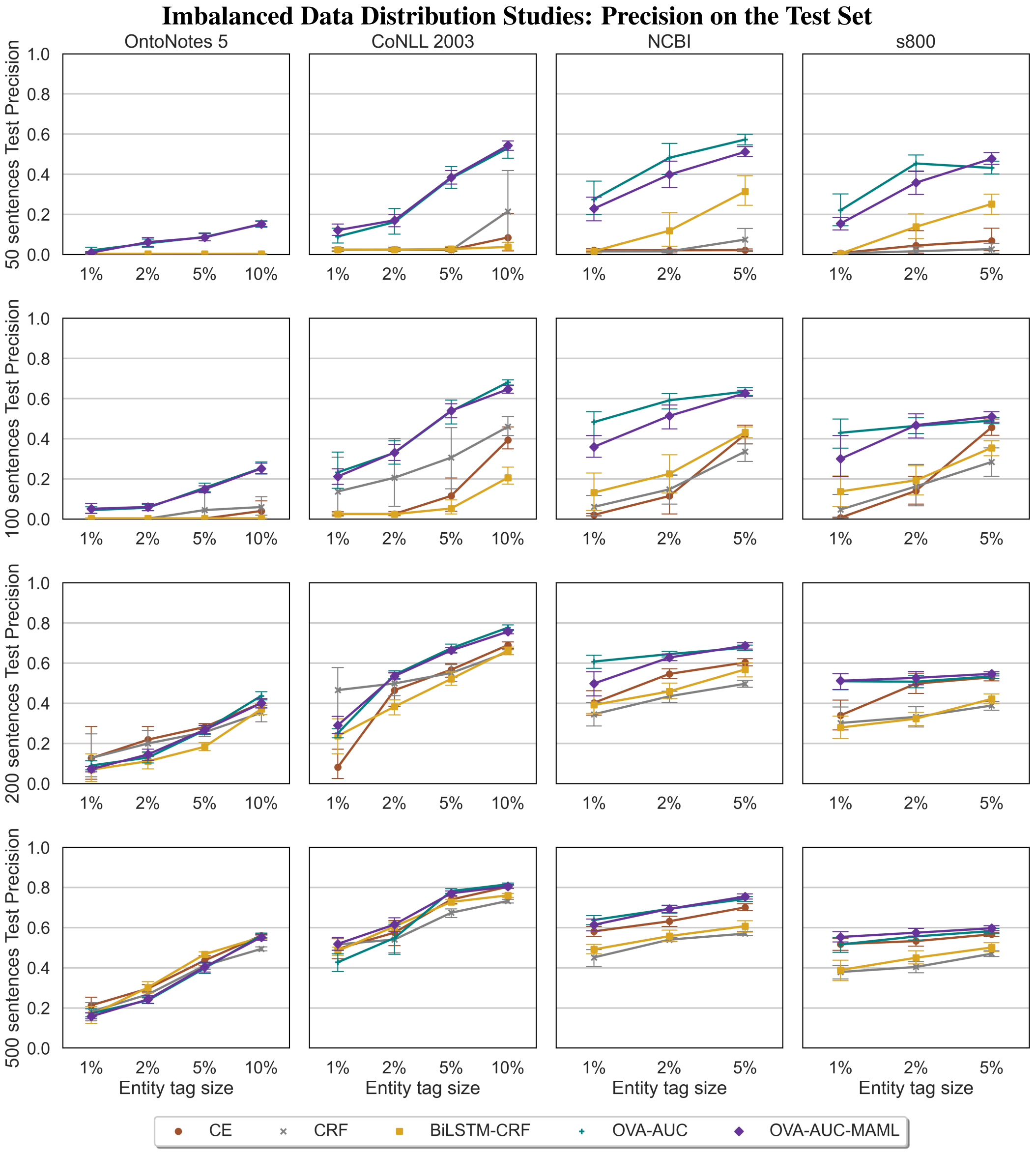


Figure 6: Average performance taken from 10 random training partitions of each training set $\mathcal{S}$ size for each loss/objective function. The entity tag size represents the percentage of entity-tokens to the total number of tokens in the training set $\mathcal{S}$. The error bars indicate the 95% confidence level of the scores. The embedding and language model for CoNLL 2003 and OntoNotes5 is "bert-base-cased" (Devlin et al., 2019) while that for NCBI and s800 is "biobert-base-cased-v1.1" (Lee et al., 2019).

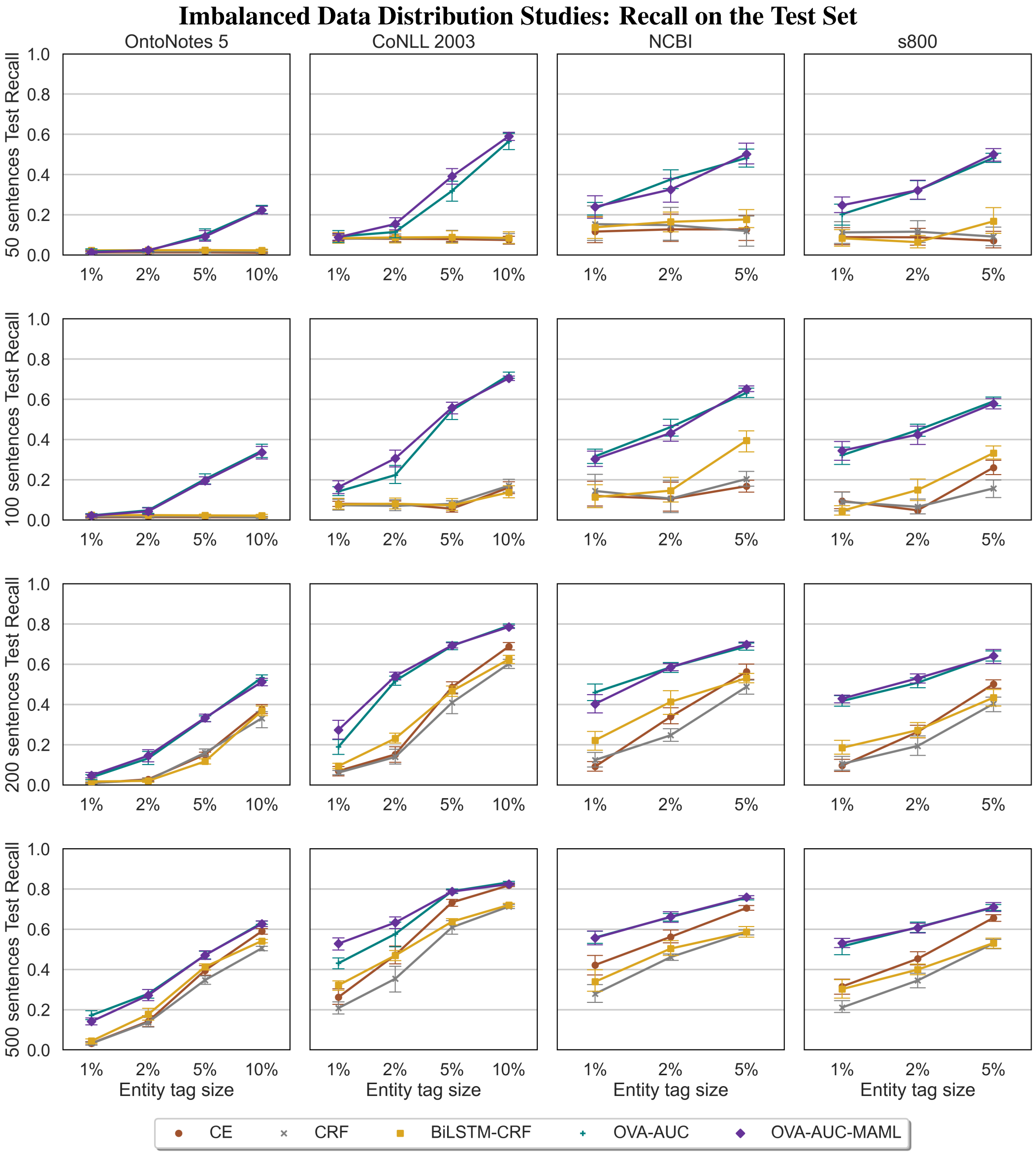


Figure 7: Average performance taken from 10 random training partitions of each training set $\mathcal{S}$ size for each loss/objective function. The entity tag size represents the percentage of entity-tokens to the total number of tokens in the training set $\mathcal{S}$. The error bars indicate the 95% confidence level of the scores. The embedding and language model for CoNLL 2003 and OntoNotes5 is “bert-base-cased” (Devlin et al., 2019) while that for NCBI and s800 is “biobert-base-cased-v1.1” (Lee et al., 2019).