THUNDER: Named Entity Recognition using a Teacher-student Model with Dual Classifiers for Strong and Weak Supervisions

Seongwoong Oh[a]**, Woohwan Jung**[b] **and Kyuseok Shim**[a;*]

[a]Department of Electrical and Computer Engineering, Seoul National University
[b]Department of Applied Artificial Intelligence, Hanyang University

Abstract. Strong and weak supervisions have complementary characteristics. However, utilizing both supervisions for named entity recognition (NER) has not been extensively studied. Moreover, the existing works address only incomplete annotations and neglects inaccurate annotations during NER model training.

To effectively utilize weak labels, we introduce an auxiliary classifier that learns from weak labels. Furthermore, we adopt the teacher-student framework to handle both incomplete and inaccurate weak labels. A teacher model is first trained using both strongly and weakly supervised data, and next generates pseudo labels to replace weak labels. Then, the student model is trained so that the main classifier learns from both strong labels and confident pseudo labels while the auxiliary classifier learns from less confident pseudo labels. We also incorporate data augmentation through ChatGPT to generate additional annotated sentences to improve model performance and generalization capabilities. The experimental results with different weak supervisions demonstrate that our proposed method surpasses existing techniques.

1 Introduction

Named entity recognition (NER) has been widely used in various natural language processing (NLP) applications such as relation extraction [14] and question answering [12]. NER locates the named entities mentioned in text and classifies them into predefined categories such as `person`, `location` and `organization`.

While human annotation (strong supervision) provides high-quality labels, it is expensive and time-consuming to obtain large-scale training data. On the other hand, to train deep learning models for NER, it is desirable to utilize large labeled data [30]. To overcome the lack of enough labeled data, one approach is to use weak supervision which automatically annotates unlabeled text by heuristic rules or machine learning models. Specifically, a popular weak supervision method is distant supervision [13] which matches the tokens in text with entities in knowledge bases and labels them with the matched entities' types. For instance, if 'New York' appears in the list of cities which can be collected from knowledge bases, 'New York' in a text can be annotated as `location`. On the other hand, ChatGPT [20], a large language model which has been trained on massive text data, can be used as weak supervision to find the entities in text and their types by using zero-shot NER [7, 21]. However, due to the limited knowledge and ambiguity of the entity types (e.g., mislabeling hyponyms to wrong types), weak supervisions inevitably introduce *incomplete* and *inaccurate* annotations. For instance, in Figure 1, a weak supervision annotates `non-entity` to the entity of *'Michelle Jones'* (i.e., *incomplete* annotation) and annotates the entity type of *'Amazon'* as `location` (i.e., *inaccurate* annotation) by misinterpreting it as *Amazon rainforest*.

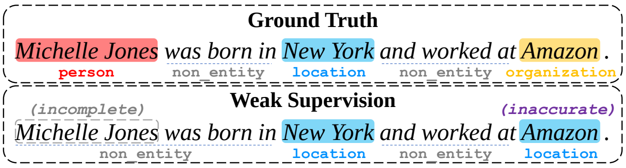


Figure 1: Noisy labels from weak supervision.

Table 1 shows the incomplete and inaccurate ratios of the weakly supervised NER datasets used for experiments in Section 5. The incomplete ratio is the proportion of the tokens whose weak labels are `non-entity` among the tokens whose ground truth labels are entity types. The inaccurate ratio is the proportion of the tokens whose ground truth labels are different from their weak labels among the tokens whose weak labels are entity types. Since the weak labels suffer from both *incompleteness* and *inaccurateness*, they may significantly degrade the generalization performance of NER models [27].

While strong and weak supervisions have complementary characteristics, most of the existing works focused on using either strong [4, 29] or weak supervision [13, 17]. Recently, the NEEDLE framework [8] is proposed to utilize both strong and weak supervisions. It corrects incomplete annotations of weak supervision using the initial NER model trained only with strongly supervised data. However, it does not correct any inaccurate annotation by weak supervision. Furthermore, since it uses a single classifier to train both strongly supervised data and weakly supervised data, inaccurate weak labels have an adverse effect of degrading the classifier accuracy.

To overcome such drawbacks, we propose the method THUN-

	CoNLL++		OntoNotes5.0		Wikigold	
	DS	ZS	DS	ZS	DS	ZS
Incomplete (%)	30.5	31.1	20.7	59.3	27.9	34.1
Inaccurate (%)	12.5	38.3	12.8	47.4	38.2	34.5

Table 1: The incomplete and inaccurate ratios of the distantly supervised data (DS) and zero-shot GPT labeled data (ZS).

[*] Corresponding Author. Email: kshim@snu.ac.kr.

DER which stands for "named entity recognition using a **T**eac**H**er-st**U**de**N**t model with **D**ual classifi**ER**s for strong and weak supervisions". A dual classifier consists of the main and auxiliary classifiers with a shared encoder. The purpose of using the auxiliary classifier is to make the main classifier less affected by noisy weak labels. To confirm the desirable capability of the auxiliary classifier, we present a theoretical and empirical analysis. On the other hand, a teacher-student model is introduced to correct incomplete and inaccurate weak labels.

A teacher model is first trained with both strong and weak labels. It next generates the pseudo labels of the weakly supervised data and replaces all the weak labels by the pseudo labels to correct incomplete and inaccurate labels. Then, a student model is trained using both strong labels and pseudo labels (instead of noisy initial weak labels). After splitting the pseudo labels into confident and unconfident labels by per-class confidence thresholds, we utilize the confident pseudo labels to train the main classifier and less confident pseudo labels to train the auxiliary classifiers. Finally, we use the student model to recognize named entities in text.

While the existing methods for zero-shot NER [7, 21] can be used to annotate unlabeled texts using ChatGPT, we propose to leverage the generative ability of ChatGPT to produce new texts with annotations for training. Specifically, we ask ChatGPT to generate additional weakly supervised data from strongly supervised data to expand the size and diversity of the training data. By conducting comprehensive experiments with the benchmark datasets as well as ChatGPT-generated datasets for NER, we show the superiority and effectiveness of our teacher-student model with dual classifiers compared to the existing methods. The results also confirm that the proposed data augmentation by ChatGPT improves the accuracy when the amount of training data is small.

2 Related Work

We categorize the existing work relevant to NER with strong and distant supervisions.

NER with weak supervision: There are several approaches that collect weak labels using heuristic labeling rules [6, 11, 13, 15, 24, 36] provided by domain experts or external information. A popular approach is distant supervision [13, 17, 34, 35] which labels the tokens in text with the entity types in knowledge bases by matching the tokens with entities in knowledge bases. To handle the noise of distant supervision for NER, partial annotation learning is used for training in [34] to ignore non-entity tokens and select only entity tokens that are likely to be correct based on reinforcement learning. On the other hand, recent works [13, 17, 35] adopt the teacher-student framework [33] which is originally proposed for semi-supervised learning [23]. It is adopted for NER with distant supervision to denoise the weak labels. The BOND method proposed in [13] is the first work that adapted the pre-trained BERT and the teacher-student framework for distantly supervised NER. Furthermore, the SCDL method presented in [35] utilizes two pairs of teacher-student models for self-collaborative denoising. In addition, the RoSTER method which utilizes a noise-robust loss function and the teacher-student framework with data augmentation is proposed in [17].

NER with strong and weak supervisions: The NEEDLE method in [8] investigates utilizing both strong and weak supervisions together for NER. It tries to correct incomplete annotations of weak supervision with an initial model trained only with strongly supervised data. It utilizes weakly supervised data without correcting any inaccurate annotation. On the other hand, our THUNDER method corrects both incomplete and inaccurate weak labels by employing a teacher-student framework with an additional auxiliary classifier to learn from weak labels.

Separating classifiers for different purposes: When there is domain shift to the test data, the method with two stages is proposed in [2] to improve the accuracy of a model. It first trains a naive model that predicts exclusively based on biases in the train data and next trains a robust model with the same train data as part of an ensemble with the naive one to focus on other patterns in the data that are not likely to be handled well by the naive one. However, it requires prior knowledge of bias to build the naive model. On the other hand, the dual supervision framework [9] utilizes two classifiers for relation extraction with strong and weak supervisions. However, none of them corrects weak labels to train the main classifier.

Generating labeled texts using ChatGPT: Some works [7, 21] utilize ChatGPT for zero-shot NER which can be used as a weak supervision. However, it only generates weak labels and does not generate new texts while our approach generates new texts with annotations. On the other hand, a data augmentation for text classification using ChatGPT is proposed in [3] which rephrases each sentence in the training data into multiple similar sentences. However, it is applicable to only text classification since it assumes that generated texts have the same label with the original text.

3 Preliminaries

We provide the problem statement of NER and describe how to train NER models.

3.1 Named Entity Recognition

NER identifies the named entity mentions in text and classifies them into predefined types [25] such as `person`, `location` and `organization`. To perform NER, we use the IO tagging scheme [17, 22] which assigns a tag `I-T` to all tokens appearing in each entity mention with type `T`, and assigns a tag `O` to all other tokens.

For a sentence $\boldsymbol{X}$, denoted by $[x_1,\cdots,x_{|\boldsymbol{X}|}]$ where $|\boldsymbol{X}|$ is the number of tokens in $\boldsymbol{X}$, NER finds a label sequence $\boldsymbol{Y} = [\boldsymbol{y}_1,\cdots,\boldsymbol{y}_{|\boldsymbol{X}|}]$ of $\boldsymbol{X}$ such that $\boldsymbol{y}_i$ is the label of the i-th token x_i and is denoted by $[y_{i,1},\cdots,y_{i,|\boldsymbol{L}|}]$ where $y_{i,j}$ is the probability that x_i belongs to the j-th class and $\boldsymbol{L}$ is the class label set.

3.2 Training an NER Model

We denote the train data $\boldsymbol{D}$ by $\{(\boldsymbol{X}_m,\boldsymbol{Y}_m)\}_{m=1}^{|\boldsymbol{D}|}$ where $\boldsymbol{X}_m$ is the m-th sentence and $\boldsymbol{Y}_m$ is its label sequence. For a sentence $\boldsymbol{X}$ and its label sequence $\boldsymbol{Y}$, let $f_i(\boldsymbol{X};\theta)$ denote the vector of probabilities $f_{i,j}(\boldsymbol{X};\theta)$ predicted by the model that a token x_i belongs to the j-th class in C where θ is the parameter set of the model. Then, we define the training objective over training data $\boldsymbol{D}$ as

$$\mathcal{L}(\boldsymbol{D};\theta) = \sum_{(\boldsymbol{X},\boldsymbol{Y})\in\boldsymbol{D}} \sum_{i=1}^{|\boldsymbol{X}|} \ell(\boldsymbol{y}_i, f_i(\boldsymbol{X};\theta)) \tag{1}$$

where $\ell(\boldsymbol{y}_i, f_i(\boldsymbol{X};\theta))$ is the cross-entropy loss for a token x_i. Note that $\ell(\boldsymbol{y}_i, f_i(\boldsymbol{X};\theta)) = -\sum_{j=1}^{|\boldsymbol{L}|} y_{i,j} \log f_{i,j}(\boldsymbol{X};\theta)$

4 The Proposed THUNDER Method

We present the THUNDER (named entity recognition using a TeacHer-stUdeNt model with Dual classifiERs for strong and weak supervisions) method.

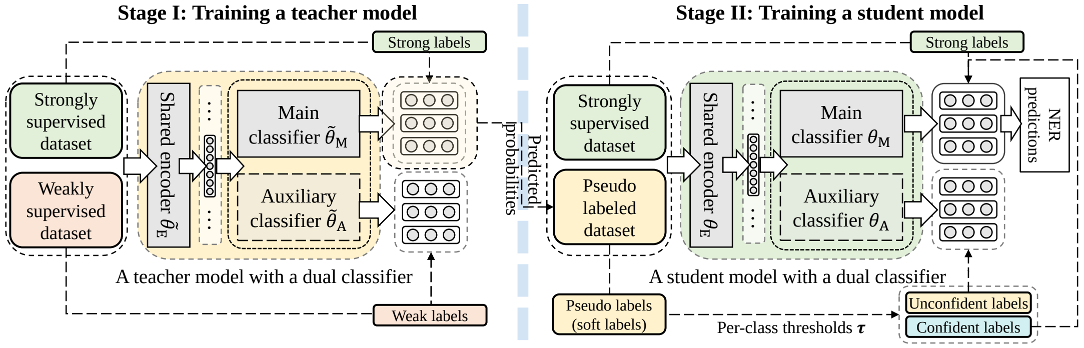

Stage I: Training a teacher model
Stage II: Training a student model
Strong labels
Strongly supervised dataset
Weakly supervised dataset
Shared encoder $\tilde{\theta}_{\mathrm{E}}$
Main classifier $\tilde{\theta}_{\mathrm{M}}$
Auxiliary classifier $\tilde{\theta}_{\mathrm{A}}$
A teacher model with a dual classifier
Weak labels
Predicted probabilities
Strong labels
Strongly supervised dataset
Pseudo labeled dataset
Shared encoder θ_{E}
Main classifier θ_{M}
Auxiliary classifier θ_{A}
NER predictions
A student model with a dual classifier
Pseudo labels (soft labels)
Per-class thresholds $\boldsymbol{\tau}$
Unconfident labels
Confident labels

Figure 2: An overview of the THUNDER method.

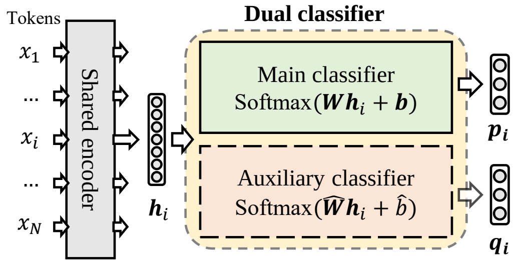


Figure 3: A dual classifier for a teacher/student model.

4.1 An Overview of the THUNDER method

Figure 2 provides an overview of the THUNDER method. Both teacher and student models used by the THUNDER method utilize dual classifiers each of which consists of a main classifier for strong labels and an auxiliary classifier for weak labels.

A teacher model is trained with strongly supervised data and weakly supervised data together. After training, to correct weak labels used for training, we generate pseudo labels for the weakly supervised data, which are the vectors consisting of the predicted probabilities of classes by the teacher model's main classifier. Then, the student model is trained so that the main classifier learns from both strong labels and confident pseudo labels while the auxiliary classifier learns from less confident pseudo labels. Finally, the main classifier of the student model is used to recognize named entities in text.

4.2 An NER Model with a Dual Classifier

As illustrated in Figure 3, a dual classifier consists of a *main classifier* and an *auxiliary classifier*, sharing the same encoder. The goal of using the auxiliary classifier is to make the main classifier to be less affected by noisy weak labels but to be more affected by accurate weak labels. Following the recent works [35, 17], we use the pretrained RoBERTa [16] as the encoder, too. The *shared encoder* takes as input a sequence of N tokens $\boldsymbol{X} = [x_1,\cdots,x_N]$ and outputs the token-wise embeddings $[\boldsymbol{h}_1,\cdots,\boldsymbol{h}_N]$ where $\boldsymbol{h}_i \in \mathbb{R}^d$ is an embedding vector of the i-th token x_i. Then, the main and auxiliary classifiers take as input the embedding $\boldsymbol{h}_i$ and predict labels with the softmax layers.

For a token x_i, the main classifier outputs the predicted probabilities of classes $\boldsymbol{p}_i = \mathrm{Softmax}(\boldsymbol{W}\boldsymbol{h}_i + \boldsymbol{b})$ where $\boldsymbol{W} \in \mathbb{R}^{|\boldsymbol{L}|\times d}$ and $\boldsymbol{b} \in \mathbb{R}^{|\boldsymbol{L}|}$ are the weight matrix and bias vector of the main classifier. Similarly, the auxiliary classifier outputs the predicted probabilities $\boldsymbol{q}_i = \mathrm{Softmax}(\hat{\boldsymbol{W}}\boldsymbol{h}_i + \hat{\boldsymbol{b}})$ where $\hat{\boldsymbol{W}} \in \mathbb{R}^{|\boldsymbol{L}|\times d}$ and $\hat{\boldsymbol{b}} \in \mathbb{R}^{|\boldsymbol{L}|}$ are the weight matrix and bias vector of the auxiliary classifier. This can also be interpreted as multi-task learning with an auxiliary task [5, 26] to predict weak labels.

4.3 Training the Teacher Model

To correct weak labels, we use a teacher model with a dual classifier. Since the main classifier should be trained with high-quality labels and the auxiliary classifier should be used for training low-quality labels, we utilize the main classifier for training with the strongly supervised data $\boldsymbol{D}_{\text{strong}}$ and the auxiliary classifiers for training weakly supervised data $\boldsymbol{D}_{\text{weak}}$. The training objective of a teacher model based on Equation (1) is

$$\mathcal{L}(\boldsymbol{D}_{\text{strong}};\tilde{\theta}_{\mathtt{E}}\cup\tilde{\theta}_{\mathtt{M}})+\mathcal{L}(\boldsymbol{D}_{\text{weak}};\tilde{\theta}_{\mathtt{E}}\cup\tilde{\theta}_{\mathtt{A}}) \qquad (2)$$

where $\tilde{\theta}_{\mathtt{E}}$ is the parameter set of the shared encoder, $\tilde{\theta}_{\mathtt{M}}$ is that of the main classifier and $\tilde{\theta}_{\mathtt{A}}$ is that of the auxiliary classifier. After training a teacher model, we produce the pseudo labeled data $\boldsymbol{D}_{\text{pseudo}} = \{(\boldsymbol{X},\boldsymbol{P})|(\boldsymbol{X},\boldsymbol{Y}) \in \boldsymbol{D}_{\text{weak}}\}$ such that $\boldsymbol{P} = [\boldsymbol{p}_1,\cdots,\boldsymbol{p}_{|\boldsymbol{X}|}]$ is the pseudo label sequence of $\boldsymbol{X}$ where $\boldsymbol{p}_i = [p_{i,1},\cdots,p_{i,|\boldsymbol{L}|}]$ is computed by $f_i(\boldsymbol{X};\tilde{\theta}_{\mathtt{E}}\cup\tilde{\theta}_{\mathtt{M}})$ defined in Section 3.2. Note that $p_{i,j}$ is used as the confidence score for the j-th class.

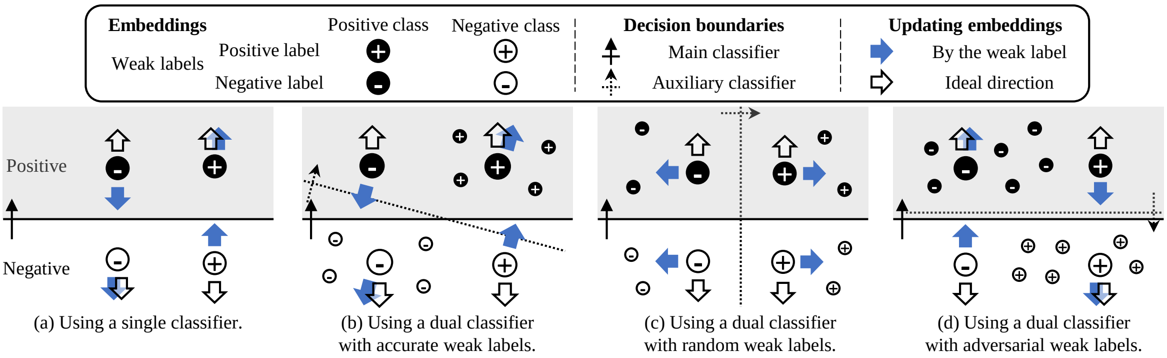
4.4 Training the Student Model

To obtain confident labels by using the probabilities of the pseudo labels to train the main classifier, we adopt the per-class confidence threshold [18]. The confidence threshold of the j-th class, denoted by τ_j, is the average confidence score of the j-th class for the tokens whose weak labels are the j-th class. In other words,

$$\tau_j = \frac{1}{N_j}\sum_{(\boldsymbol{X},\boldsymbol{Y})\in\boldsymbol{D}_{\text{weak}}}\sum_{i=1}^{|\boldsymbol{X}|} y_{i,j} f_{i,j}(\boldsymbol{X};\tilde{\theta}_{\mathtt{E}}\cup\tilde{\theta}_{\mathtt{M}}) \qquad (3)$$

where $y_{i,j}$ is the value of the j-th dimension in the weak label $\boldsymbol{y}_i = [y_{i,1},\cdots,y_{i,|\boldsymbol{L}|}]$ of the i-th token in $\boldsymbol{X}$ and N_j is the number of tokens whose weak labels are the j-th class.

For a sentence $\boldsymbol{X} = [x_1,\cdots,x_{|\boldsymbol{X}|}]$, let the pseudo label sequence be $\boldsymbol{P} = [\boldsymbol{p}_1,\cdots,\boldsymbol{p}_{|\boldsymbol{X}|}]$ where $\boldsymbol{p}_i = [p_{i,1},\cdots,p_{i,|\boldsymbol{L}|}]$ is the pseudo label of the i-th token x_i. For each token x_i, if $p_{i,j} \geq \tau_j$ where τ_j is defined in Equation (3), we want the token x_i to be predicted as the j-th class proportional to the value of $p_{i,j}$. Thus, from the pseudo label $\boldsymbol{p}_i$, we first compute the confident label $\boldsymbol{c}_i = [c_{i,1},\cdots,c_{i,|\boldsymbol{L}|}]$ and the unconfident label $\boldsymbol{u}_i = [u_{i,1},\cdots,u_{i,|\boldsymbol{L}|}]$ together such that $c_{i,j} = p_{i,j}$ and $u_{i,j} = 0$ if $p_{i,j} \geq \tau_j$, but otherwise $c_{i,j} = 0$ and $u_{i,j} = p_{i,j}$. We next generate the confident label sequence $\boldsymbol{C} = [\boldsymbol{c}_1,\cdots,\boldsymbol{c}_{|\boldsymbol{X}|}]$ and the unconfident label sequence $\boldsymbol{U} = [\boldsymbol{u}_1,\cdots,\boldsymbol{u}_{|\boldsymbol{X}|}]$ from $\boldsymbol{P}$.



(a) Using a single classifier.

(b) Using a dual classifier with accurate weak labels.

(c) Using a dual classifier with random weak labels.

(d) Using a dual classifier with adversarial weak labels.

Figure 4: Illustrations of updating the shared encoder's embeddings by weak labels with an auxiliary classifier.

Let us define $\boldsymbol{D}_{\text{conf}} = \{(\boldsymbol{X}, \boldsymbol{C}) | (\boldsymbol{X}, \boldsymbol{P}) \in \boldsymbol{D}_{\text{pseudo}}\}$ and $\boldsymbol{D}_{\text{unconf}} = \{(\boldsymbol{X}, \boldsymbol{U}) | (\boldsymbol{X}, \boldsymbol{P}) \in \boldsymbol{D}_{\text{pseudo}}\}$. We refer to $\boldsymbol{D}_{\text{conf}}$ as the confident labeled data and $\boldsymbol{D}_{\text{unconf}}$ as the unconfident labeled data.

Similar to the teacher model, the main classifier is for training high-quality labels and the auxiliary classifier is for training low-quality labels. Thus, we train the main classifier with both strongly supervised data $\boldsymbol{D}_{\text{strong}}$ and confident labeled data $\boldsymbol{D}_{\text{conf}}$ while training the auxiliary classifier with unconfident labeled data $\boldsymbol{D}_{\text{unconf}}$. The training objective of a student model is

$$\mathcal{L}(\boldsymbol{D}_{\text{strong}} \cup \boldsymbol{D}_{\text{conf}}; \theta_{\mathtt{E}} \cup \theta_{\mathtt{M}}) + \lambda_U \mathcal{L}(\boldsymbol{D}_{\text{unconf}}; \theta_{\mathtt{E}} \cup \theta_{\mathtt{A}}) \qquad (4)$$

where $\theta_{\mathtt{E}}$ is the parameter set of the shared encoder, $\theta_{\mathtt{M}}$ is that of the main classifier, $\theta_{\mathtt{A}}$ is that of the auxiliary classifier, and λ_U is a hyperparameter that represents the relative importance of $\mathcal{L}(\boldsymbol{D}_{\text{unconf}}; \theta_{\mathtt{E}} \cup \theta_{\mathtt{A}})$. Finally, the main classifier of the student model is used to recognize named entities from the test data.

4.5 Data Augmentation using ChatGPT for NER

Although the zero-shot NER [7, 21] using ChatGPT can generate weakly supervised data by labeling the unlabeled texts, it does not produce new texts with annotations. To fully leverage the powerful generative ability of ChatGPT and strongly supervised data, we propose a data augmentation technique with ChatGPT to create additional weakly supervised data by utilizing strongly supervised data to expand the size and diversity of the training data for better generalization capabilities of the model. The data augmentation process comprises the following steps:

1. **Generating annotated sentences:** For each sentence with at least one annotated entity from the strongly supervised data, we construct the prompt that is a string *"Generate an NER example by modifying the following:"* followed by the sentence in the next line where every entity is enclosed by <TYPE> and </TYPE> tags. We next invoke the GPT-3.5-turbo API with the prompt and the number of generated sentences for each prompt.
2. **Extracting the generated sentences:** We retrieve the generated sentences where annotated entities are enclosed by <TYPE> tags from the API responses. We next remove the sentences with mismatching tags introduced by ChatGPT.
3. **Consistency filtering:** Due to the generative and random nature of ChatGPT, it may generate an unwanted type. Thus, we retain only the sentences that do not have any entity type not appearing in the original sentences.

Due to the potential for inaccuracies in the labels of the generated sentences, we treat them as weakly supervised data.

4.6 Behaviors of Dual Classifiers

To demonstrate how the auxiliary classifier helps the main classifier, we provide intuitive illustrations of how the shared encoder embeddings change depending on the quality of the weak labels in Figure 4. We consider a two-dimensional embedding space of the shared encoder with binary classification for ease of understanding.

The *black circle* and *white circle* represent embeddings of weakly labeled examples whose true classes are positive and negative, respectively. In each circle, '+' denotes a positive label, while '-' denotes a negative label. The *solid line* represents the main classifier's decision boundary, and the *dotted line* indicates the auxiliary classifier's. Besides, the *blue arrows* show the update directions for embeddings based on weak labels, while the *white arrows* indicate the update directions based on true classes.

Consider the case where a single classifier (only a main classifier) is used for both strong and weak labels in Figure 4(a). The classifier predicts the positive class for data instances with their embeddings above the decision boundary, and the negative class otherwise. Ideally, the embeddings of the *positive class* (*black circles*) should be updated in the positive class direction, while those of the *negative class* (*white circles*) should be updated in the negative class direction. However, the embeddings with *positive weak labels*, denoted by '+', are updated to the positive class direction and those with *negative weak labels*, denoted by '-', are updated to the negative class direction. Note that both update directions are orthogonal to the main classifier's decision boundary. Since wrong weak labels may decrease the classifier accuracy, using a single classifier is effective only when weak labels are highly accurate.

Let us now examine the cases, where the dual classifiers are used, in Figures 4(b)-(d). Figure 4(b) shows the case where the weak and strong labels are *consistent* (i.e., the decision boundaries of both classifiers are *similar*). In this case, the update directions of the encoder's embeddings by using weak labels are similar to those in Figure 4(a) except that their directions are not orthogonal to the main classifier's decision boundary. Since the predicted probabilities of the main classifier change the most when the update directions of the embeddings are orthogonal to the decision boundary, the main classifier is less affected by weak labels.

Figure 4(c) shows the case where the weak and strong labels are *not correlated* at all (i.e., the decision boundaries of both classifiers are *orthogonal*). The update directions of the embeddings based on weak labels are now parallel to the main classifier's decision boundary, preventing the main classifier from being affected by weak labels. On the other hand, Figure 4(d) shows an extreme case where the weak labels are *adversarial* (i.e., the decision boundaries of both classifiers are *opposite* each other). The update directions of embeddings for weak labels are now opposite to those in Figure 4(a). Interestingly, the embeddings with wrong weak labels are updated towards the main classifier's correct predictions.

To confirm the desirable capability of the auxiliary classifier, we theoretically analyze its influence on the main classifier based on the degree of weak label accuracy. For the i-th token x_i, the main and auxiliary classifiers output $\boldsymbol{p}_i = \mathrm{Softmax}(\boldsymbol{W}\boldsymbol{h}_i + \boldsymbol{b})$ and $\boldsymbol{q}_i = \mathrm{Softmax}(\hat{\boldsymbol{W}}\boldsymbol{h}_i + \hat{\boldsymbol{b}})$, respectively, where $\boldsymbol{h}_i$ is its embedding vector output by the shared encoder. Let $\boldsymbol{w}_j$ and $\hat{\boldsymbol{w}}_j$ be the j-th rows of $\boldsymbol{W}$ and $\hat{\boldsymbol{W}}$, respectively. The next lemma shows that the cosine similarity between the loss gradients of the main and auxiliary classifiers with respect to the shared encoder is positively correlated to $cos(\boldsymbol{w}_j, \hat{\boldsymbol{w}}_j)$ for $1 \leq j \leq |\boldsymbol{L}|$. The implication of the lemma is that auxiliary classifiers have an adaptive capability to help the main classifiers.

Lemma 1. *For the weight matrix of the main classifier* $\boldsymbol{W}$ *and that of the auxiliary classifier* $\hat{\boldsymbol{W}}$, *assume* (1) $\boldsymbol{w}_j \cdot \boldsymbol{w}_k = \boldsymbol{w}_j \cdot \hat{\boldsymbol{w}}_k = \hat{\boldsymbol{w}}_j \cdot \hat{\boldsymbol{w}}_k = 0$ *with* $j \neq k$, (2) $\|\boldsymbol{w}_1\| = \|\boldsymbol{w}_2\| = \ldots = \|\boldsymbol{w}_{|\boldsymbol{L}|}\|$ *and* (3) $\|\hat{\boldsymbol{w}}_1\| = \|\hat{\boldsymbol{w}}_2\| = \ldots = \|\hat{\boldsymbol{w}}_{|\boldsymbol{L}|}\|$. *Then, for a token* x_i *with label* $\boldsymbol{y}_i$, $cos(\nabla_{\boldsymbol{h}_i}\ell(\boldsymbol{y}_i, \boldsymbol{p}_i), \nabla_{\boldsymbol{h}_i}\ell(\boldsymbol{y}_i, \boldsymbol{q}_i)) = \sum_{j=1}^{|\boldsymbol{L}|} \alpha_j cos(\boldsymbol{w}_j, \hat{\boldsymbol{w}}_j)$ *such that* $0 < \alpha_j < 1$ *for* $1 \leq j \leq |\boldsymbol{L}|$.

Proof. Given the label $\boldsymbol{y}_i$ of x_i, the gradient of the loss $\ell(\boldsymbol{y}_i, \boldsymbol{p}_i)$ with respect to $\boldsymbol{h}_i$ by the main classifier, denoted by $\nabla_{\boldsymbol{h}_i}\ell(\boldsymbol{y}_i, \boldsymbol{p}_i)$, is $\nabla_{\boldsymbol{z}}\ell(\boldsymbol{y}_i, \boldsymbol{p}_i)\nabla_{\boldsymbol{h}}\boldsymbol{z} = (\boldsymbol{p}_i - \boldsymbol{y}_i)^\top \boldsymbol{W} = \sum_j (p_{i,j} - y_{i,j})\boldsymbol{w}_j$ where $\boldsymbol{z} = \boldsymbol{W}\boldsymbol{h} + \boldsymbol{b}$ by the chain rule. Similarly, the gradient of the loss $\ell(\boldsymbol{y}_i, \boldsymbol{q}_i)$ with respect to $\boldsymbol{h}_i$ by the auxiliary classifier, denoted by $\nabla_{\boldsymbol{h}_i}\ell(\boldsymbol{y}_i, \boldsymbol{q}_i)$, becomes $\sum_j (q_{i,j} - y_{i,j})\hat{\boldsymbol{w}}_j$. By letting $\beta_j = p_{i,j} - y_{i,j}$, $\hat{\beta}_j = q_{i,j} - y_{i,j}$ and $\alpha_j = \frac{\beta_j \hat{\beta}_j}{\sqrt{\sum_j \beta_j^2}\sqrt{\sum_j \hat{\beta}_j^2}}$, we derive

$$cos(\nabla_{\boldsymbol{h}_i}\ell(\boldsymbol{y}_i, \boldsymbol{p}_i), \nabla_{\boldsymbol{h}_i}\ell(\boldsymbol{y}_i, \boldsymbol{q}_i)) = \frac{(\sum_j \beta_j \boldsymbol{w}_j) \cdot (\sum_j \hat{\beta}_j \hat{\boldsymbol{w}}_j)}{\|\sum_j \beta_j \boldsymbol{w}_j\| \|\sum_j \hat{\beta}_j \hat{\boldsymbol{w}}_j\|}$$

$$= \frac{\sum_j \beta_j \hat{\beta}_j \boldsymbol{w}_j \cdot \hat{\boldsymbol{w}}_j}{\sqrt{\sum_j \beta_j^2 \boldsymbol{w}_j \cdot \boldsymbol{w}_j}\sqrt{\sum_j \hat{\beta}_j^2 \hat{\boldsymbol{w}}_j \cdot \hat{\boldsymbol{w}}_j}} \text{ (by assumption (1))}$$

$$= \frac{\sum_j \beta_j \hat{\beta}_j \boldsymbol{w}_j \cdot \hat{\boldsymbol{w}}_j}{\sqrt{r^2 \sum_j \beta_j^2}\sqrt{\hat{r}^2 \sum_j \hat{\beta}_j^2}} \text{ (by assumptions (2) and (3))}$$

$$= \sum_{j=1}^{|\boldsymbol{L}|} \frac{\beta_j \hat{\beta}_j}{\sqrt{\sum_j \beta_j^2}\sqrt{\sum_j \hat{\beta}_j^2}} \frac{\boldsymbol{w}_j \cdot \hat{\boldsymbol{w}}_j}{r\hat{r}}$$

$$= \sum_{j=1}^{|\boldsymbol{L}|} \alpha_j \frac{\boldsymbol{w}_j \cdot \hat{\boldsymbol{w}}_j}{r\hat{r}} = \sum_{j=1}^{|\boldsymbol{L}|} \alpha_j cos(\boldsymbol{w}_j, \hat{\boldsymbol{w}}_j)$$

Since $\boldsymbol{p}_i$ and $\boldsymbol{q}_i$ are the outputs of the softmax function, we have $0 < p_{i,j} < 1$ and $0 < q_{i,j} < 1$. If $y_{i,j} = 1$, we have $\beta_j < 0$ and $\hat{\beta}_j < 0$. On the other hand, if $y_{i,j} = 0$, we obtain $\beta_j > 0$ and $\hat{\beta}_j > 0$. Thus, we get $\beta_j \hat{\beta}_j > 0$. Furthermore, since $\beta_j \neq 0$ and $\hat{\beta}_j \neq 0$, we have $\beta_j \hat{\beta}_j < \sqrt{\sum_j \beta_j^2}\sqrt{\sum_j \hat{\beta}_j^2}$. Thus, $0 < \alpha_j < 1$ holds for all j. □

We next illustrate the update directions of the shared encoder's embedding by the main and auxiliary classifiers based on Lemma 1.
When $cos(\nabla_{\boldsymbol{h}_i}\ell(\boldsymbol{y}_i, \boldsymbol{p}_i), \nabla_{\boldsymbol{h}_i}\ell(\boldsymbol{y}_i, \boldsymbol{q}_i)) \gg 0$**:** The main classifier is helped by weak labels since both directions are *similar* (Figure 4(b)).
When $cos(\nabla_{\boldsymbol{h}_i}\ell(\boldsymbol{y}_i, \boldsymbol{p}_i), \nabla_{\boldsymbol{h}_i}\ell(\boldsymbol{y}_i, \boldsymbol{q}_i)) \approx 0$**:** The main classifier is less affected by weak labels since both directions are *orthogonal* (Figure 4(c)).
When $cos(\nabla_{\boldsymbol{h}_i}\ell(\boldsymbol{y}_i, \boldsymbol{p}_i), \nabla_{\boldsymbol{h}_i}\ell(\boldsymbol{y}_i, \boldsymbol{q}_i)) \ll 0$**:** The main classifier is helped by the weak labels as shown in Figure 4(d).

We empirically verify the behavior of dual classifiers as well as the assumptions used in Lemma 1 later in Section 5.10.

Dataset	**CoNLL++**	**OntoNotes5.0**	**Wikigold**
# Entity types	4	18	4
# Train (strong)	3,250	8,528	280
# Train (weak)	14,041	59,924	1,142
# Augmented	11,501	22,436	1,334
# Test	3,453	8,262	274

Table 2: The statistics of the datasets.

Hyperparameter	**CoNLL++**	**OntoNotes5.0**	**Wikigold**
Batch size	32		
Learning rate	1e-5		
Dropout rate	0.1		
Warmup proportion	0.1		
λ_U	0.1		
# epochs (a teacher model)	5	5	8
# epochs (a student model)	7	5	16
Maximum sequence length	150	180	150

Table 3: Hyperparameter values.

5 Experiments

We empirically evaluate the THUNDER method and existing methods on three NER datasets. We use entity-level precision/recall/F1-score as the evaluation measures. We train the NER models with 3 random seeds and report the average results.

5.1 Datasets

We use three NER benchmark datasets: CoNLL++ [31], OntoNotes5.0 [32] and Wikigold [1]. The statistics of the datasets are summarized in Table 2. Note that we use the distantly supervised training data of the three datasets generated in [13]. We also generated weakly supervised data using ChatGPT by the zero-shot NER [7, 21] and the data augmentation in Section 4.5. Since we use 'gpt-3.5-turbo' model with a token cost of $0.002 per 1,000 tokens, the zero-shot NER costs $1.5, $8.0, and $0.2 for CoNLL++, OntoNotes5.0, and Wikigold, respectively. In addition, the data augmentation costs $2.4, $4.7, and $0.3 for the same datasets.

5.2 Compared Methods

We implement the following methods for named entity recognition.
NEEDLE: This is the state-of-the-art method for NER with both strong and weak supervisions in [8]. We use the original implementation in https://github.com/amzn/amazon-weak-ner-needle.
RoSTER: It is the state-of-the-art method for using only weak supervision in [17]. Although it takes weakly supervised data only, we also show its performance by taking the union of both weakly and strongly supervised datasets together as input when a strongly supervised dataset is also available. We use the original implementation in https://github.com/yumeng5/RoSTER.
THUNDER: It is our work in Section 4. The implementation is available at https://github.com/swoh91/THUNDER.

		CoNLL++			OntoNotes5.0			Wikigold		
		Prec	Rec	F1	Prec	Rec	F1	Prec	Rec	F1
SS	**RoSTER**	89.98	91.21	90.59	85.06	86.51	85.78	77.52	82.49	79.93
DS	**RoSTER**	88.09	85.19	86.62	75.72	79.00	77.33	60.50	76.13	67.42
SS + DS	**RoSTER**	90.04	90.99	90.51	78.88	81.38	80.11	69.11	81.46	74.75
	NEEDLE	89.93	90.55	90.24	86.38	87.95	87.16	78.75	82.22	80.45
	THUNDER	**92.87**	**92.93**	**92.90**	**87.55**	**88.87**	**88.20**	**82.16**	**86.41**	**84.23**
	(std.dev.)	(0.17)	(0.09)	(0.12)	(0.16)	(0.10)	(0.13)	(0.41)	(0.41)	(0.41)
SS + DS + GA	**RoSTER**	90.43	92.10	91.25	80.05	83.02	81.51	74.69	83.03	78.63
	NEEDLE	90.73	91.13	90.93	87.04	88.24	87.64	79.28	82.38	80.80
	THUNDER	**93.01**	**93.16**	**93.08**	**87.28**	**88.80**	**88.03**	**83.50**	**86.08**	**84.77**
	(std.dev.)	(0.09)	(0.11)	(0.09)	(0.03)	(0.09)	(0.04)	(1.21)	(1.06)	(1.09)

Table 4: Main results with strongly supervised data (SS), distantly supervised data (DS) and ChatGPT augmented data (GA).

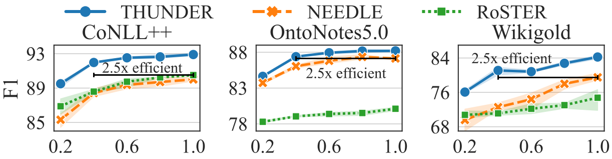


Figure 5: Varying the amount of strongly supervised data

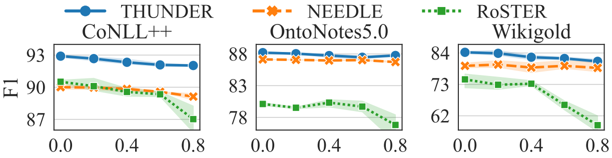


Figure 6: Varying the additional noise to distantly supervised data

5.3 Experimental Settings

We use the pre-trained RoBERTa-base [16] model as the encoder in every method for fair comparison. Note that the number of parameters in the encoder is 125M and that in the dual classifier is from 8K to 30K depending on the number of classes. We use Adam optimizer [10] and a linear learning rate schedule with its peak value 0.00001. The training batch size is set to 32 and the batches are alternately sampled from strongly supervised data and weakly supervised data. The parameter λ_U in Eq. (4) is set to 0.1. For fair comparison with RoSTER which uses an ensemble of teacher models, we make every method use an ensemble of three models to generate pseudo labels. All methods are implemented with PyTorch and trained on an NVIDIA RTX 3090 GPU. We report the hyperparameter values in Table 3.

5.4 Main Results

We present the precision, recall and F1-score of the compared methods trained on strongly supervised data (**SS**), distantly supervised data (**DS**), both strongly and distantly supervised data (**SS+DS**), and **SS+DS** with ChatGPT augmented data (**SS+DS+GA**) in Table 4. When trained on SS+DS, THUNDER consistently achieves the best performance among all compared methods. Since RoSTER is designed to be trained with weak supervision only, RoSTER treats SS+DS as weakly supervised data and large noisy labels degrade its performance. It indicates that THUNDER effectively takes advantage of both strongly and distantly supervised datasets by using the dual classifier. When GA is added to them, the performances of the compared methods are improved in most cases, especially when the amount of training data is small. Specifically, the F1-scores of all methods for Wikigold are increased 0.4~5.2% (2.1% on average).

	CoNLL	OntoNotes	Wikigold
THUNDER	**92.90**	**88.20**	**84.23**
w/o PCT (Per-class threshold)	92.71	87.96	83.30
w/o DC (Dual classifier)	89.46	81.95	82.61
w/o TS (Teacher-student) / PCT	91.83	87.25	80.41
w/o DC / PCT	88.00	80.23	80.57
w/o DC / TS / PCT	86.49	78.81	78.41

Table 5: Ablation study (F1-score).

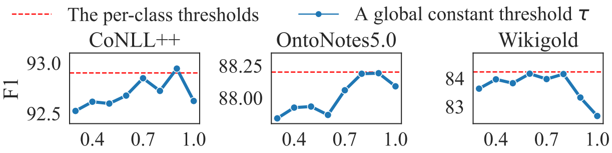


Figure 7: Comparison to global constant thresholds.

5.5 Data Efficiency and Noise Robustness

To demonstrate the effectiveness of THUNDER, we compare its F1-score with those of other methods by varying the strongly and distantly supervised data. We conduct the experiments with varying the amount of strongly supervised data from 20% to 100% and report the F1-scores in Figure 5. For every dataset, the performances of all methods are improved as the amount of strongly supervised data increases. The performance of THUNDER with only 40% of strong labels is comparable to that of NEEDLE with all the strong labels.

We also conduct the experiments with varying the ratio of additional noise from 0% to 80% by randomly flipping the correct labels of distantly supervised data using the ground truth labels of distantly supervised data and report the F1-scores in Figure 6. As expected, THUNDER consistently outperforms the other methods. The performances of both THUNDER and NEEDLE are relatively robust to the noise of distant supervision since both exploit strongly supervised data. However, the performance of RoSTER degrades significantly when we add a lot of additional noisy labels.

5.6 Ablation Study

To evaluate the effectiveness of each component used by THUNDER, we conduct the ablation study by removing some of the following components: dual classifiers (**DC**), teacher-student models (**TS**) and per-class thresholds (**PCT**). We present the result in Table 5. If we disable **TS**, we train only the teacher model and we thus

		CoNLL++			OntoNotes5.0			Wikigold		
		Prec	Rec	F1	Prec	Rec	F1	Prec	Rec	F1
Distantly supervised data (DS)		82.38	62.33	70.97	81.96	72.44	76.91	47.90	47.63	47.76
SS+DS	**NEEDLE**	82.69	87.46	85.01	77.67	86.32	81.77	53.47	70.68	60.88
	THUNDER	**93.92**	**93.80**	**93.86**	**87.22**	**89.45**	**88.32**	**82.14**	**86.01**	**84.03**
Zero-shot GPT supervised data (ZS)		64.01	50.81	56.65	45.65	26.31	33.38	66.51	49.93	57.04
SS+ZS	**NEEDLE**	70.57	78.77	74.44	62.72	77.10	69.17	68.86	77.66	73.00
	THUNDER	**94.07**	**93.70**	**93.88**	**84.45**	**86.14**	**85.29**	**78.94**	**81.19**	**80.05**

Table 6: Comparison of the distant labels, the zero-shot GPT labels and the pseudo labels.

Ground truth: [...]$_{PER}$ will meet [...]$_{PER}$ in [***Gaza***]$_{LOC}$ on Thursday after [...]$_{MISC}$ said the right-wing [***Israeli***]$_{MISC}$ government had barred the [***Palestinian***]$_{MISC}$ leader from flying to the [***West Bank***]$_{LOC}$ for talks ...

Distant label: [...]$_{PER}$ will meet [...]$_{PER}$ in [***Gaza***]$_{PER}$ on Thursday after [...]$_{MISC}$ said the right-wing ***Israeli*** government had barred the [***Palestinian***]$_{MISC}$ leader from flying to the [***West Bank*** for]$_{ORG}$ talks ...

Pseudo label (NEEDLE): [...]$_{PER}$ will meet [...]$_{PER}$ in [***Gaza***]$_{PER}$ on Thursday after [...]$_{MISC}$ said the right-wing [***Israeli***]$_{MISC}$ government had barred the [***Palestinian***]$_{MISC}$ leader from flying to the [***West Bank*** for]$_{ORG}$ talks ...

Pseudo label (THUNDER): (identical to the ground truth)

Zero-shot GPT label: [...]$_{PER}$ will meet [...]$_{PER}$ in [***Gaza***]$_{LOC}$ on Thursday after [...]$_{MISC}$ said the right-wing [***Israeli*** government]$_{ORG}$ had barred the [***Palestinian*** leader]$_{PER}$ from flying to the [***West Bank***]$_{LOC}$ for talks ...

Pseudo label (NEEDLE): (identical to the zero-shot GPT label)

Pseudo label (THUNDER): (identical to the ground truth)

Table 7: Case study.

	CoNLL++	OntoNotes5.0	Wikigold
SS+DS+ZS	93.02	87.95	83.56
SS+DS+GA w/o CF	92.73	87.90	84.10
SS+DS+GA w/ CF	**93.08**	**88.03**	**84.77**

Table 8: F1-scores using zero-shot ChatGPT supervised data (ZS), ChatGPT augmented data (GA) and consistency filtering (CF).

cannot use **PCT** since it is used for training the student model. We obtain significant accuracy improvements (from 6% to 10%) in all datasets by using the dual classifiers. Furthermore, the student model consistently outperforms the teacher model since it is trained with corrected pseudo labels. It shows the effectiveness of our teacher-student model. Finally, utilizing confident pseudo labels filtered by the per-class threshold also improves the accuracies for all datasets.

5.7 *The Effect of Confident Thresholds*

To show the effectiveness of the per-class thresholds computed by Equation (3), we also test THUNDER with varying a single global pre-defined constant threshold as done in [17]. We report the F1-score of THUNDER in Figure 7. We find that THUNDER with the per-class thresholds are comparable to those with the best constant thresholds. Note that the best constant thresholds are different for the datasets and the per-class thresholds are automatically computed.

5.8 *Quality of Pseudo Labels*

To evaluate the quality of pseudo labels generated differently, we show the accuracy of pseudo labeling methods in Table 6. The pseudo labeling of THUNDER is more accurate than that of NEEDLE. Since NEEDLE does not correct inaccurate annotations of weakly supervised data, it suffers from low precision in all datasets. Meanwhile, since THUNDER corrects both incomplete and inaccurate annotations, THUNDER significantly outperforms NEEDLE in terms of precision, recall and F1-score. For example, with SS+DS, THUNDER achieved 40% higher F1-score than NEEDLE in Wikigold dataset that has the lowest accuracy for the distant labels.

Type of weak labels	CoNLL++		OntoNotes5.0		Wikigold	
	Single	**Dual**	**Single**	**Dual**	**Single**	**Dual**
Accurate	**93.68**	93.54	89.57	**89.82**	86.20	**86.24**
Similar	89.46	**92.90**	81.95	**88.20**	82.61	**84.23**
Random	54.48	**92.16**	20.91	**86.81**	45.48	**82.19**
Adverse	30.02	**93.67**	9.49	**90.01**	52.42	**85.70**

Table 9: Noise-robustness of dual classifiers (F1-score).

	CoNLL++	OntoNotes5.0	Wikigold
$cos(\boldsymbol{w}_i, \boldsymbol{w}_j)$	-0.04 ± 0.03	-0.02 ± 0.04	-0.01 ± 0.03
$cos(\boldsymbol{w}_i, \hat{\boldsymbol{w}}_j)$	-0.02 ± 0.04	-0.01 ± 0.04	0.00 ± 0.04
$cos(\hat{\boldsymbol{w}}_i, \hat{\boldsymbol{w}}_j)$	-0.02 ± 0.04	-0.01 ± 0.05	-0.00 ± 0.04
$\|\boldsymbol{w}_i\|$	0.59 ± 0.01	0.67 ± 0.04	0.56 ± 0.01
$\|\hat{\boldsymbol{w}}_i\|$	0.57 ± 0.02	0.63 ± 0.03	0.56 ± 0.02

Table 10: Validation of the assumptions (1)-(3) in Lemma 1. The weight vectors $\boldsymbol{w}_i$ and $\boldsymbol{w}_j$ are the i-th and j-th rows of the weight matrix $\boldsymbol{W}$ in the main classifier. Similarly, $\hat{\boldsymbol{w}}_i$ and $\hat{\boldsymbol{w}}_j$ are those of $\hat{\boldsymbol{W}}$ in the auxiliary classifier. Besides, $\|\boldsymbol{w}_i\|$ is the norm of $\boldsymbol{w}_i$.

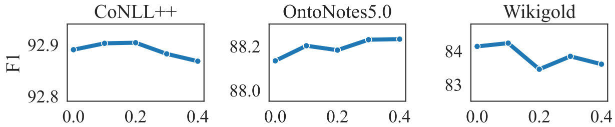


Figure 8: The effect of the hyperparameter λ_U.

Table 7 shows the pseudo labels generated by different methods by using a single sentence from CoNLL++ dataset. Note that weakly supervised CoNLL++ datasets contain non-negligible inaccurate and incomplete annotations that are colored red. It has an incomplete annotation *'Israeli'*. In addition, it generates inaccurate annotations of `person` for *'Gaza'* and `organization` for *'West Bank for'*. On the other hand, zero-shot ChatGPT generates inaccurate annotations of `organization` for *'Israeli government'* and `person` for *'Palestinian leader'*. Since NEEDLE corrects only incomplete annotations, the inaccurate annotations remain unchanged. In contrast, the pseudo labels produced by THUNDER trained on SS+DS and SS+ZS are identical to the ground truth.

5.9 *Comparison of Weak Supervisions using ChatGPT*

Table 8 presents empirical results that compare the efficacy of two data generation techniques using ChatGPT. Our proposed data augmentation approach with ChatGPT consistently surpasses the existing weak labeling method based on zero-shot NER with ChatGPT across all datasets. As illustrated in Table 6, the weak labels generated by the zero-shot NER method are inaccurate, which consequently hinders improvement in overall accuracy. Moreover, the fact that DS and ZS share identical input texts leads to a reduction in the diversity of the data. In contrast, our method employs ChatGPT's ability to paraphrase input data while leveraging strong labels, enabling us to obtain comparatively accurate data. In addition, *CF* (consistency filtering) is also effective to improve the performance. Consequently, our ChatGPT-based data augmentation method outperforms the zero-shot NER labeling approach using ChatGPT.

5.10 Analysis of Dual Classifiers

Noise-robustness of dual classifiers: To empirically verify the analysis in Section 4.6, we present the F1-score of our method using a *Single* classifier or a *Dual* classifier by varying the quality of weak labels in Table 9 where *Accurate* is the ground truth labels, *Similar* is the distant labels, *Random* is uniform random labels and *Adverse* is synthetic labels that have completely wrong associations of the classes [28]. While *Single* performs poorly with the *Random* and *Adverse* weak labels, *Dual* is robust to noisy labels. Even more, *Dual* effectively utilizes *Adverse* weak labels.

Validation of Lemma 1: Table 10 shows that the similarities between the weight vectors of the different classes are close to 0 indicating that the assumption (1) in Lemma 1 is reasonable. In addition, to validate the assumptions (2) and (3) of Lemma 1, Table 10 also shows the means and standard deviations of the norms of the weight vectors. Since the standard deviations of the norms are much smaller than their means, the norms of the weight vectors are very similar although their norms are not exactly the same. Thus, we can expect that the cosine similarity between the gradients of the losses obtained by training the main and auxiliary classifiers with respect to the shared encoder is positively correlated to $cos(\boldsymbol{w}_j, \hat{\boldsymbol{w}}_j)$ for all j by Lemma 1.

5.11 The Effect of the Hyperparameter λ_U

We study the effect of the hyperparameter λ_U used in Equation (4). Figure 8 shows the changes in the performance of THUNDER measured by F1. We vary the values of λ_U in the range of $[0, 0.1, 0.2, 0.3, 0.4]$. When λ_U increases, the relative importance of the unconfident pseudo labels increases. In general, the performance is insensitive to the hyperparameter in the range. We manually tune λ_U in the range and select $\lambda_U = 0.1$ based on the average F1-score.

5.12 Comparison of Weak Supervisions

We report the accuracies of distant supervision and zero-shot NER using ChatGPT. Table 11, 12 and 13 show the accuracies on CoNLL++, OntoNotes5.0 and Wikigold datasets, respectively. For CoNLL++ and OntoNotes5.0, the accuracy of the zero-shot NER is worse than distant supervision in every type. For Wikigold, zero-shot NER performs better than distant supervision in most types except `miscellaneous` type.

5.13 Examples of Data Augmentation using ChatGPT

Table 14 shows the consistent and inconsistent data generated and classified by our method for a sentence in CoNLL++ dataset. The input text is annotated by enclosing the entities *'Hong Kong'* and *'New York'* by <LOC> tags. Similarly, the entities in the generated texts such as *'San Francisco'*, *'London'* and *'Tokyo'* are also enclosed by <LOC> tags. On the other hand, ChatGPT also generates texts with the type tags inconsistent to the input text. These examples tend to include more inaccurate annotations (e.g., '*<PER> Apple </PER>*') that are colored red by human evaluation. The results demonstrate that consistency filtering effectively eliminates erroneous annotations produced by ChatGPT.

	Distant supervision			Zero-shot NER		
Type	Prec	Rec	F1	Prec	Rec	F1
ALL	82.38	62.33	70.97	64.01	50.81	56.65
LOC	99.98	74.59	85.44	64.65	64.65	64.65
PER	63.46	72.45	67.66	65.64	61.23	63.36
ORG	90.83	59.39	71.82	69.57	48.28	57.00
MISC	100.00	22.86	37.22	23.74	6.69	10.44

Table 11: Comparison of weak supervisions on CoNLL++ dataset.

	Distant supervision			Zero-shot NER		
Type	Prec	Rec	F1	Prec	Rec	F1
ALL	81.96	72.44	76.91	45.65	26.31	33.38
PERSON	94.08	65.25	77.05	61.56	27.92	38.42
GPE	81.77	90.45	85.89	71.53	44.44	54.82
ORG	85.71	62.44	72.25	44.80	39.83	42.17
DATE	81.05	84.96	82.96	57.73	9.09	15.71
CARDINAL	60.75	68.44	64.36	43.80	7.38	12.64
NORP	88.73	74.75	81.14	29.64	16.89	21.51
MONEY	81.10	61.01	69.64	40.09	24.45	30.37
PERCENT	76.73	79.30	77.99	53.56	52.07	52.80
ORDINAL	82.56	34.63	48.80	20.47	2.13	3.87
LOC	88.74	51.52	65.19	16.72	15.85	16.28
TIME	60.49	42.09	49.64	15.81	4.22	6.66
WORK OF ART	89.37	89.73	89.55	9.30	22.18	13.11
FAC	89.80	61.40	72.93	13.10	14.07	13.57
EVENT	84.55	76.07	80.08	9.39	9.89	9.64
QUANTITY	71.18	56.01	62.69	20.20	15.22	17.36
PRODUCT	86.65	76.07	81.02	4.55	16.34	7.12
LANGUAGE	87.12	46.71	60.81	46.38	31.58	37.57
LAW	69.59	42.20	52.54	8.86	8.51	8.68

Table 12: Comparison of weak supervisions on OntoNotes5.0 dataset.

	Distant supervision			Zero-shot NER		
Type	Prec	Rec	F1	Prec	Rec	F1
ALL	47.90	47.63	47.76	66.51	49.93	57.04
LOC	79.57	49.78	61.24	72.95	62.11	67.09
PER	50.85	58.50	54.41	74.54	59.80	66.36
ORG	33.61	43.50	37.92	64.59	52.35	57.83
MISC	36.14	34.87	35.49	34.29	15.79	21.62

Table 13: Comparison of weak supervisions on Wikigold dataset.

We request 10 responses for each sentence by setting the number of generation n to 10 in the API function 'openai.ChatCompletion.create' [1].

6 Conclusion

We study the named entity recognition with strong and weak supervisions. We first propose the THUNDER method by employing a teacher-student model with dual classifiers. We next analyze the effect of the dual classifiers to the shared encoder. In addition, we introduce a data augmentation using ChatGPT for NER. The experimental results confirm that the proposed method effectively utilizes both strong and weak labels for NER and achieves the best performance.

[1] https://platform.openai.com/docs/api-reference/chat/create

Acknowledgements

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No. 2020-0-00857, Development of cloud robot intelligence augmentation, sharing and framework technology to integrate and enhance the intelligence of multiple robots). Furthermore, it was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No. 2022-0-00516, Derivation of a Differential Privacy Concept Applicable to National Statistics Data While Guaranteeing the Utility of Statistical Analysis). This work was also supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (No. NRF-2022R1G1A1013549).

References

[1] Dominic Balasuriya, Nicky Ringland, Joel Nothman, Tara Murphy, and James R Curran, 'Named entity recognition in wikipedia', in *Proceedings of the 2009 Workshop on People's Web*, pp. 10–18, (2009).

[2] Christopher Clark, Mark Yatskar, and Luke Zettlemoyer, 'Don't take the easy way out: Ensemble based methods for avoiding known dataset biases', in *Proceedings of EMNLP-IJCNLP 2019*, pp. 4069–4082, Hong Kong, China, (November 2019). ACL.

[3] Haixing Dai, Zhengliang Liu, Wenxiong Liao, Xiaoke Huang, Yihan Cao, Zihao Wu, Lin Zhao, Shaochen Xu, Wei Liu, Ninghao Liu, Sheng Li, Dajiang Zhu, Hongmin Cai, Lichao Sun, Quanzheng Li, Dinggang Shen, Tianming Liu, and Xiang Li. Auggpt: Leveraging chatgpt for text data augmentation, 2023.

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, 'BERT: Pre-training of deep bidirectional transformers for language understanding', in *Proceedings of NAACL 2019*, pp. 4171–4186, Minneapolis, Minnesota, (June 2019). ACL.

[5] Yunshu Du, Wojciech M. Czarnecki, Siddhant M. Jayakumar, Mehrdad Farajtabar, Razvan Pascanu, and Balaji Lakshminarayanan. Adapting auxiliary losses using gradient similarity, 2020.

[6] Jason A. Fries, Sen Wu, Alexander Ratner, and Christopher Ré, 'Swellshark: A generative model for biomedical named entity recognition without labeled data', *CoRR*, **abs/1704.06360**, (2017).

[7] Yan Hu, Iqra Ameer, Xu Zuo, Xueqing Peng, Yujia Zhou, Zehan Li, Yiming Li, Jianfu Li, Xiaoqian Jiang, and Hua Xu. Zero-shot clinical entity recognition using chatgpt, 2023.

[8] Haoming Jiang, Danqing Zhang, Tianyu Cao, Bing Yin, and Tuo Zhao, 'Named entity recognition with small strongly labeled and large weakly labeled data', in *Proceedings of ACL/IJCNLP 2021*, pp. 1775–1789. ACL, (2021).

[9] Woohwan Jung and Kyuseok Shim, 'Dual supervision framework for relation extraction with distant supervision and human annotation', in *Proceedings of COLING 2020*, pp. 6411–6423, (2020).

[10] Diederik P. Kingma and Jimmy Ba, 'Adam: A method for stochastic optimization', in *ICLR 2015*, (2015).

[11] Jiacheng Li, Haibo Ding, Jingbo Shang, Julian McAuley, and Zhe Feng, 'Weakly supervised named entity tagging with learnable logical rules', in *Proceedings of ACL/IJCNLP 2021*, pp. 4568–4581, Online, (August 2021). Association for Computational Linguistics.

[12] Xiaoya Li, Fan Yin, Zijun Sun, Xiayu Li, Arianna Yuan, Duo Chai, Mingxin Zhou, and Jiwei Li, 'Entity-relation extraction as multi-turn question answering', in *Proceedings of ACL 2019*, pp. 1340–1350, Florence, Italy, (July 2019). ACL.

[13] Chen Liang, Yue Yu, Haoming Jiang, Siawpeng Er, Ruijia Wang, Tuo Zhao, and Chao Zhang, 'BOND: bert-assisted open-domain named entity recognition with distant supervision', in *KDD 2020*, pp. 1054–1064. ACM, (2020).

[14] Yankai Lin, Zhiyuan Liu, and Maosong Sun, 'Neural relation extraction with multi-lingual attention', in *Proceedings of ACL 2017*, pp. 34–43, Vancouver, Canada, (July 2017). ACL.

[15] Pierre Lison, Jeremy Barnes, Aliaksandr Hubin, and Samia Touileb, 'Named entity recognition without labelled data: A weak supervision approach', in *Proceedings of ACL 2020*, pp. 1518–1533, Online, (July 2020). Association for Computational Linguistics.

[16] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov, 'Roberta: A robustly optimized bert pretraining approach', *arXiv preprint arXiv:1907.11692*, (2019).

[17] Yu Meng, Yunyi Zhang, Jiaxin Huang, Xuan Wang, Yu Zhang, Heng Ji, and Jiawei Han, 'Distantly-supervised named entity recognition with noise-robust learning and language model augmented self-training', in *Proceedings of EMNLP 2021*, pp. 10367–10378. ACL, (2021).

[18] Curtis G. Northcutt, Lu Jiang, and Isaac L. Chuang, 'Confident learning: Estimating uncertainty in dataset labels', *J. Artif. Intell. Res.*, **70**, 1373–1411, (2021).

[19] Seongwoong Oh, Woohwan Jung, and Kyuseok Shim. Thunder: Named entity recognition using a teacher-student model with dual classifiers for strong and weak supervisions. techinical report. https://github.com/swoh91/ecai23/raw/main/extended.pdf.

[20] OpenAI. ChatGPT. https://openai.com/blog/chatgpt.

[21] Chengwei Qin, Aston Zhang, Zhuosheng Zhang, Jiaao Chen, Michihiro Yasunaga, and Diyi Yang. Is chatgpt a general-purpose natural language processing task solver?, 2023.

[22] Lance Ramshaw and Mitch Marcus, 'Text chunking using transformation-based learning', in *Third Workshop on Very Large Corpora*, (1995).

[23] Chuck Rosenberg, Martial Hebert, and Henry Schneiderman, 'Semi-supervised self-training of object detection models', in *(WACV/MOTION 2005*, pp. 29–36. IEEE Computer Society, (2005).

[24] Esteban Safranchik, Shiying Luo, and Stephen H. Bach, 'Weakly supervised sequence tagging from noisy rules', in *AAAI2020*, pp. 5570–5578. AAAI Press, (2020).

[25] Erik F. Tjong Kim Sang and Fien De Meulder, 'Introduction to the conll-2003 shared task: Language-independent named entity recognition', in *Proceedings of HLT-NAACL 2003*, pp. 142–147. ACL, (2003).

[26] Baifeng Shi, Judy Hoffman, Kate Saenko, Trevor Darrell, and Huijuan Xu, 'Auxiliary task reweighting for minimum-data learning', in *NeurIPS 2020*, (2020).

[27] Hwanjun Song, Minseok Kim, Dongmin Park, Yooju Shin, and Jae-Gil Lee. Learning from noisy labels with deep neural networks: A survey, 2020.

[28] Ryutaro Tanno, Ardavan Saeedi, Swami Sankaranarayanan, Daniel C. Alexander, and Nathan Silberman, 'Learning from noisy labels by regularized estimation of annotator confusion', in *CVPR 2019*, pp. 11244–11253. Computer Vision Foundation / IEEE, (2019).

[29] Xinyu Wang, Yong Jiang, Nguyen Bach, Tao Wang, Zhongqiang Huang, Fei Huang, and Kewei Tu, 'Automated concatenation of embeddings for structured prediction', in *Proceedings of ACL/IJCNLP 2021*, pp. 2643–2660, Online, (August 2021). ACL.

[30] Yaqing Wang, Subhabrata Mukherjee, Haoda Chu, Yuancheng Tu, Ming Wu, Jing Gao, and Ahmed Hassan Awadallah, 'Meta self-training for few-shot neural sequence labeling', in *KDD 2021*, pp. 1737–1747, (2021).

[31] Zihan Wang, Jingbo Shang, Liyuan Liu, Lihao Lu, Jiacheng Liu, and Jiawei Han, 'Crossweigh: Training named entity tagger from imperfect annotations', in *Proceedings of EMNLP-IJCNLP 2019*, pp. 5153–5162. ACL, (2019).

[32] Ralph Weischedel, Martha Palmer, Mitchell Marcus, Eduard Hovy, Sameer Pradhan, Lance Ramshaw, Nianwen Xue, Ann Taylor, Jeff Kaufman, Michelle Franchini, et al., 'Ontonotes release 5.0 ldc2013t19', *Linguistic Data Consortium, Philadelphia, PA*, **23**, (2013).

[33] Qizhe Xie, Minh-Thang Luong, Eduard H. Hovy, and Quoc V. Le, 'Self-training with noisy student improves imagenet classification', in *CVPR 2020*, pp. 10684–10695. Computer Vision Foundation / IEEE, (2020).

[34] Yaosheng Yang, Wenliang Chen, Zhenghua Li, Zhengqiu He, and Min Zhang, 'Distantly supervised NER with partial annotation learning and reinforcement learning', in *Proceedings of COLING 2018*, pp. 2159–2169, Santa Fe, New Mexico, USA, (August 2018). ACL.

[35] Xinghua Zhang, Bowen Yu, Tingwen Liu, Zhenyu Zhang, Jiawei Sheng, Mengge Xue, and Hongbo Xu, 'Improving distantly-supervised named entity recognition with self-collaborative denoising learning', in *Proceedings of EMNLP 2021*, pp. 10746–10757. ACL, (2021).

[36] Xinyan Zhao, Haibo Ding, and Zhe Feng, 'GLaRA: Graph-based labeling rule augmentation for weakly supervised named entity recognition', in *Proceedings of EACL 2021*, pp. 3636–3649, Online, (April 2021). Association for Computational Linguistics.

Input text:

Generate an NER example by modifying the following:
Gold closed down at $ 387.00 / $ 387.50 an ounce in <LOC> Hong Kong </LOC> on Friday , versus <LOC> New York </LOC> 's $ 387.60 / $ 388 finish on Thursday .

Generated texts (consistent):

Apple's stocks finished trading at $ 154.94 in <LOC> San Francisco </LOC> on Monday, versus <LOC> London </LOC>'s $ 153.80 closing on Friday.

Apple Inc.'s stock price soared to $150.00 / $151.00 per share in <LOC> New York </LOC> on Wednesday, breaking the previous record set in <LOC> San Francisco </LOC> last year.

Apple's stock rose to $135.54 in <LOC> Tokyo </LOC> on Monday, compared to <LOC> New York </LOC>'s $134.87 close on Friday.

Apple closed down at $118.03 / $118.69 a share in <LOC> Tokyo </LOC> on Monday, versus <LOC> NASDAQ </LOC>'s $119.05 / $119.79 finish on Friday.
Entities:
1. Stock- "Apple"
2. Price- "$118.03 / $118.69" and "$119.05 / $119.79"
3. Location- "Tokyo" and "NASDAQ"
4. Day- "Monday" and "Friday".

Apple's stock price jumped to $156.23 in <LOC> Tokyo </LOC> on Monday , compared to Friday's closing of $155.54 in <LOC> San Francisco </LOC> .

Apple's stock price rose by $5.00 in <LOC> San Francisco </LOC> on Friday, compared to <LOC> London </LOC>'s closing price of $120.50 on Thursday.

Generated texts (inconsistent):

The CEO of <PER> Apple </PER> announced at the <LOC> Worldwide Developers Conference </LOC> in <LOC> San Francisco </LOC> on Monday that the company's latest <MISC> iPhone </MISC> model will hit stores next week, starting at a price of $699.

In <LOC>Washington D.C.</LOC>, <PER>President Biden</PER> announced that <ORG>Amazon</ORG> has agreed to purchase <MONEY>$2 billion</MONEY> worth of <ORG>green energy</ORG> from <ORG>renewable energy farms</ORG>, as a part of his initiative to reduce carbon emissions.

<TIME> On Monday </TIME>, Apple's stock <ACTION> dropped </ACTION> to $110.13 per share <MONEY> in the <LOC> NASDAQ </LOC> stock exchange </MONEY> , <ORG> following </ORG> disappointing <MISC> quarterly earnings </MISC>.

<BODY> Apple Inc . </BODY> stock reached a new all-time high of $ 145.09 in <LOC> San Francisco </LOC> on Wednesday , topping its previous record of $ 144.12 set just last month in <LOC> New York </LOC> . </BODY>

Table 14: Some examples of data augmentation using ChatGPT for CoNLL++ dataset.