applied sciences

MDPI

Article

Mixup Based Cross-Consistency Training for Named Entity Recognition

Geonsik Youn, Bohan Yoon, Seungbin Ji, Dahee Ko and Jongtae Rhee *

Department of Industrial and Systems Engineering, Dongguk University, Seoul 04620, Korea
* Correspondence: jtrhee@dongguk.edu

Abstract: Named Entity Recognition (NER) is at the core of natural language understanding. The quality and amount of datasets determine the performance of deep-learning-based NER models. As datasets for NER require token-level or word-level labels to be assigned, annotating the datasets is expensive and time consuming. To alleviate efforts of manual anotation, many prior studies utilized weak supervision for NER tasks. However, using weak supervision directly would be an obstacle for training deep networks because the labels automatically annotated contain a a lot of noise. In this study, we propose a framework to better train the deep model for NER tasks using weakly labeled data. The proposed framework stems from the idea that mixup, which was recently considered as a data augmentation strategy, would be an obstacle to deep model training for NER tasks. Inspired by this idea, we used mixup as a perturbation function for consistency regularization, one of the semi-supervised learning strategies. To support our idea, we conducted several experiments for NER benchmarks. Experimental results proved that directly using mixup on NER tasks hinders deep model training while demonstrating that the proposed framework achieves improved performances compared to employing only a few human-annotated data.

Keywords: deep learning; named entity recognition; consistency regularization; semi-supervised learning; mixup

check for updates

Citation: Youn, G.; Yoon, B.; Ji, S.; Ko, D.; Rhee, J. Mixup Based Cross-Consistency Training for Named Entity Recognition. *Appl. Sci.* **2022**, *12*, 11084. https://doi.org/10.3390/app122111084

Academic Editor: Byung-Gyu Kim

Received: 18 October 2022
Accepted: 30 October 2022
Published: 1 November 2022

1. Introduction

This paper is an extension of work originally presented in [1]. Named Entity Recognition (NER) is the task of natural language processing that detects mentions of entities from text and classifies them into predefined entity types such as people, locations, and organizations. NER has a key role in natural language understanding, such as information extraction [2], question answering [3], translation systems [4], and automatic summarization systems [5,6]. In particular, the domain-specific NER is more important because it extracts expertise from domain-specific documents or sentences. For example, when dealing with a document from Information Technology companies, NER identifies where Information Technology terms appear and leads the development of task automation or interactive artificial intelligence, i.e., customer-automated response services (e.g., chatbots). Recently, various deep learning models have been applied to NER, achieving state-of-the-art performances, which was possible due to the large amount of datasets labeled strongly by humans [7–10]. However, as datasets for NER require token-level labels to be assigned, it is expensive and time consuming to generate them [11–13]. In particular, in terms of having to deal with specific domain texts, the datasets for specific domain NER are more expensive [12].

Several studies used the weak supervision for NER work to alleviate the trouble of manual annotations [11,13–15]. Weak supervision comprises assigning labels to unlabeled data via an automated process in some way and using them for model training. From the NER's point of view, researchers first collect raw corpus and knowledge base (e.g., entity dictionary) related to a specific domain. After that, if a word built according to the knowledge base is in the raw corpus' text, it is assigned as an entity mention (exact string matching). Finally, the NER model is trained with entity mention labels automatically assigned with a specific algorithm. However, due to the fact that the model is trained with automatically generated labels, weak supervision may lead deep networks in the wrong direction. We can guess that performance is improved by an increment in data for training. However, the quality of the data is insufficient, which rather hinders model training. The reason is derived from the characteristics of weakly labeled data. Even with an enormous knowledge base, the scope is bound to be limited and may not reflect newly added entities. In addition, some algorithms, such as exact string matching, automatically assign labels, and they would incorrectly assign some entity mentions as well as non-entity mentions [12]. Therefore, we should deal with weakly labeled data carefully.

This study can be categorized as a semi-supervised learning study in that the weakly labeling process uses unlabeled data. A representative strategy of semi-supervised learning is consistency regularization, which is based on the assumption that the model's prediction for the data point should be consistent when the data point is given a perturbation. In other words, the model should generate predictions in which one data point and the perturbed data point are the same [16]. We noted that the mixup, which has recently been proposed as a data augmentation strategy in computer vision, is an obstacle to training in NER tasks. Inspired by this, we propose to use mixup as a strategy of perturbation for consistency regularization.

In this study, we propose a framework for the appropriate use of weakly labeled data. In the first step, Stage 1, we generate weakly labeled data using a pre-built knowledge base. In Stage 2, we train the NER model with strongly labeled data. The trained model then performs pseudo-labeling on the weakly labeled data. Then, we use the pseudo-label and the weak label by mixing them up at following stages. In Stage 3, we use both strongly labeled data and weakly labeled data to train the model. In this framework, weakly labeled data is perturbed through mixup and used as a means of consistency regularization. We train the model by repeating Stage 3 until the model converges. We can summarize our contributions as follows:

- We propose the mixup, which was originally used as a data augmentation method, as perturbation to implement consistency regularization.
- We propose a framework to train the model for Named Entity Recognition tasks better using weakly labeled data. The proposed framework is based on semi-supervised learning combined with pseudo-labeling and consistency regularization by mixup.
- Our experimental results proved that if we directly utilize mixup on NER tasks, we yield worse performances with respect to deep NER models. Moreover, the results showed that the proposed framework improves performances compared to using only a few strongly labeled data.

The rest of this paper consists of the following. Section 2 introduces backgrounds of this study. Section 3 describes the proposed framework, which is an effective strategy to use weakly labeled data. In Section 4, we show the experimental results to compare the performance of the proposed framework, including experiments under low resource environments. Finally, Section 5 summarizes and concludes the study.

2. Preliminaries

In this section, we describe the tasks to be covered in this study: Named Entity Recognition. This section is followed Section 2.1, in which we introduce the techniques that we covered and prior studies. In Section 2.2, we introduce mixup, which is the technique mainly covered in this study. Lastly, in Section 2.3, we introduce semi-supervised learning, including how previous studies used semi-supervised learning on NER tasks.

2.1. Named Entity Recognition

Named Entity Recognition is a task that detects mentions of entities from texts and classifies them into predefined entity types. In other words, for the sentence or text

$\mathbf{x} = \{x_1, \ldots, x_i, \ldots, x_n\}$ constructed of n tokens, the NER task is the task that assign a sequence of label tokens $\mathbf{y} = \{y_1, \ldots, y_i, \ldots, y_n\}$. A label token $y_i \in \mathbf{y}$ is one of the entity tags according to input token x_i. The tags are `B-E`, `B-I`, and `O` tags, which mean beginning, inside, and outside of predefined entity `E`, respectively. Formally, the strongly labeled data, $D = \{(\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_N, \mathbf{y}_N)\}$, consist of N pairs. After the weakly labeling and pseudo-labeling phase, unlabeled texts $D' = \{\mathbf{x}'_1, \ldots, \mathbf{x}'_M\}$ would be weakly labeled data. Generally, M is greater than N $(M \gg N)$ because generating strongly labeled data for NER requires high costs, as mentioned in Section 1.

2.2. Mixup

Mixup is proposed for data augmentation in computer vision [17]. The main idea of mixup is to generate data points by a linear combination of two labeled data points (x, y) and (x', y'):

$$\begin{aligned} \tilde{x} &= \lambda x + (1-\lambda)x' \\ \tilde{y} &= \lambda y + (1-\lambda)y' \\ \lambda &\sim \text{Beta}(\alpha, \alpha) \\ \lambda &= \max(\lambda, 1-\lambda) \end{aligned} \tag{1}$$

where $\lambda \in [0, 1]$ follows a Beta distribution and controls the degree of mixing two data samples. Here, x is an image, and y is a one-hot label, which represents categorical variables as binary vectors. For example, as shown in Figure 1, mixing 0.7 of cat images and 0.3 of dog images generates a new data sample. Using the mixup technique on continuous data structures such as image pixel values works appropriately. However, using the mixup technique on discrete data structure such as words or tokens does not work appropriately.

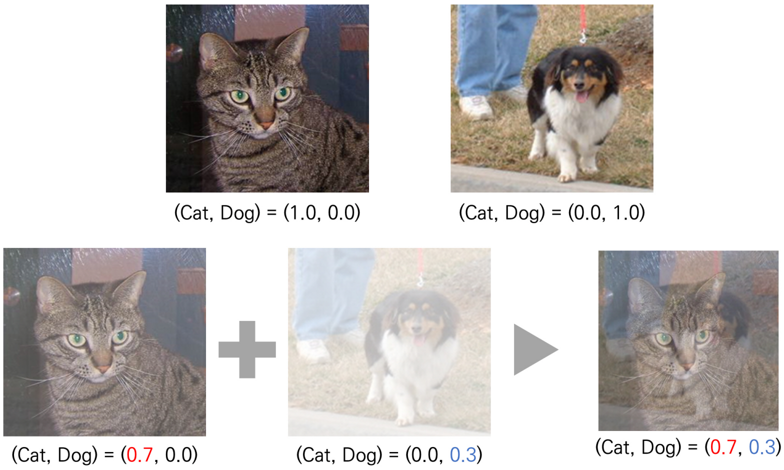

Figure 1. Mixup process.

TMix [18] is the mixup technique for natural language data, which mixes hidden representations of words or tokens. Specifically, TMix inputs two examples into multi-layer language models such as BERT [19] and then mixes the hidden representations of two inputs on the k-th layer of the language model consisting of L layers in total $(k \in [1, L])$. TMix shows that the mixup technique works well on text classification tasks. Unlike text classification tasks where one example has one label, NER tasks have multiple labels for one example. Furthermore, these labels are related to each other in a semantic context. NER examples generated by mixup contain too much noise and would lead model training in the wrong direction. In previous studies, researchers tried to make improvements to fit NER tasks, such as performing mixups between two similar sentences [20] or applying mixup to sentences where the entity's density is above a certain threshold [21]. Recently, RegMixup [22] utilized mixup as an additional regularizer on out-of-distribution detection tasks. Similarly to our point of view, the authors observed that mixup yields insufficient performances because mixup makes the task difficult to solve.

2.3. Semi-Supervised Learning

We use the semi-supervised learning strategy when we can use a few labeled data and a large amount of unlabeled data together.

2.3.1. Pseudo Labeling

Pseudo-labeling, also called self-training, is a strategy for assigning labels to unlabeled data as a prediction of the model trained with small strongly labeled data. In prior studies on computer vision, the authors [23] firstly trained teacher models with a few strongly labeled data. Then, the teacher model generated pseudo-labels that the student model would train with. Finally, both the teacher and the student model trained recursively with pseudo-data and labeled data to get better performance of the student model. For NER tasks, the researchers followed the strategy in which the teacher and student models recursively trained with pseudo-data and labeled data. In [24], the teacher named Judge model and the NER model trained by complementing each other in recursive manner. The authors in [25] focused on the resume text. They tried to find the education section in resumes. As the labeled data for detecting the education term was limited, they used the pseudo-labeling strategy by using the dictionary of the institution and degree names. In [12], the authors focused on product names in the query text of online shopping websites. They collected user behavior data on the website for weak annotations. They found that weak labels could contain noise, so they replaced the weak label to the model's prediction, which is trained with strongly labeled data. ROSE-NER [26] trained the ROSE-NER base with labeled data and then obtained pseudo-labels for the unlabeled dataset, which the ROSE-NER base predicted. Lastly, ROSE-NER is trained with a new dataset, a combination of pseudo-labeled data and labeled data.

2.3.2. Consistency Regularization

The main idea of consistency regularization is that the model's prediction for unlabeled data should be robust to any perturbations. In computer vision, the images consist of continuous pixel values, and images can easily create novel and realistic-looking augmented data. So, many studies could implement consistency regularization by applying various perturbations or augmentations to one example. CutMix [27] is a method for semantic segmentation. This method used the mixup technique as a perturbation method to force the consistencies between the model outputs and inputs of mixed examples. Cross-consistency training (CCT) [28] proposed applying perturbations to the hidden representations of examples and not the model's inputs. Moreover, several techniques for injecting noise such as dropout, adding noise tensors, or masking into features were used as perturbation techniques. Figure 2 shows that the variance in the decision boundary decreases as training considers the perturbations on the unlabeled examples [28]. The boundary (a) formed in a wide range may cause inaccurate predictions for some examples. On the other hand, boundary (b) separates the categories explicitly.

Unlike the continuous image pixel values, it is difficult to perturb discrete words or tokens. Thus, many researchers tried to find methods to provide perturbations, which are necessary for using consistency regularization on natural language data. The studies in [29] proposed filling input texts with blanks to provide perturbations. It is similar to our study and [28] in that the data are perturbed. However, it is different in that the perturbation point is the input level and not the hidden representation level.

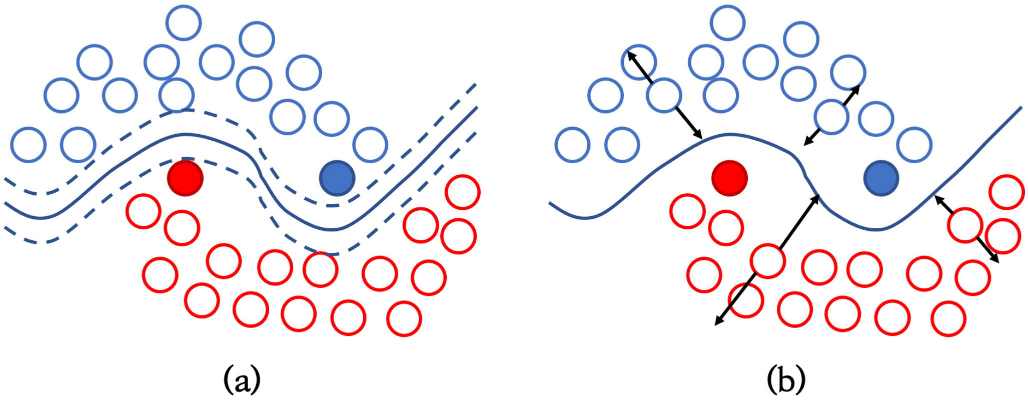

Figure 2. Illustration of the decision boundaries: (**a**) Decision boundary without the consideration of perturbation on examples; (**b**) decision boundary with consideration of perturbation on examples.

2.3.3. Holistic Methods

Holistic methods are strategies that combine mixup, pseudo-labeling, and consistency regularization. Holistic methods are similar to the proposed method in this study in that examples are mixed and used for training in a consistency regularization manner. The studies in [30] proposed a holistic method that constructed several steps, including data augmentation, label guessing, mixup, and joint training with consistency loss. In [31], the authors added two new techniques comprising distribution alignment and augmentation anchoring. The distribution alignment forces the distribution of predictions on unlabeled data and matches the predictions of the labeled data. With the augmentation anchoring technique, the predictions of weakly augmented data are assigned to the labels of augmented data. FixMatch [32] implemented the holistic method by subtracting the mixup technique and combining consistency regularization and pseudo-labeling. Instead, the authors assigned the label of a strongly augmented input as the model prediction on a weakly augmented input. Holistic methods have been limited to the studies in computer vision, where the mixup and data augmentation are freely adopted.

3. Method

In this section, we first demonstrate the background of this study. Then, from Sections 3.1–3.4, we describe the details of the components and framework with formulas. In Figure 3, some entity examples with blue-colored edges and marked with red color points are mislabeled as non-entities. The model trained on these mislabeled examples would fail to generate the decision boundary correctly. This case would occur frequently in NER tasks where mislabeled examples are easily found.

Meanwhile, cross-consistency training is a strategy that alleviates variances in the decision boundary by considering perturbations in examples. When we use cross-consistency training, we need to implement the perturbations of the examples. As shown in Figure 3, if we use direct supervision from the mixed examples, the model would create an extremely distorted decision boundary. Inspired by this point, we consider mixed examples as perturbated from weakly labeled examples. By cross-consistency training with mixed examples, we can alleviate variances in the decision boundary, such as the decision boundary in Figure 2b, and at the same time separate all examples despite the mislabeled examples.

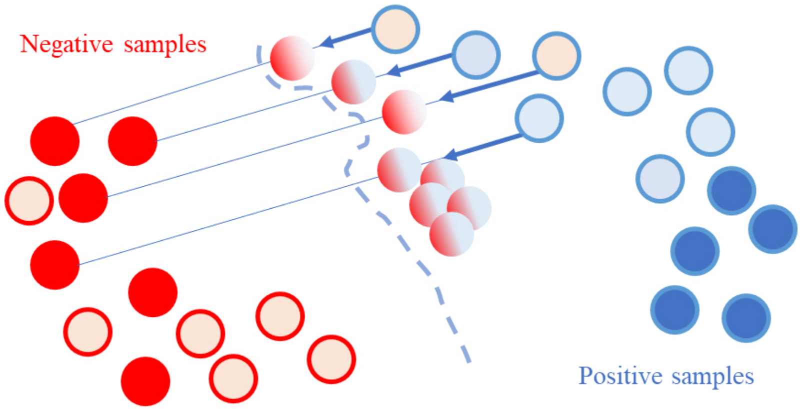

Figure 3. The data manifold under the environment to use weakly labeled data. The blue points are labeled as entities and the red points are labeled as non-entities. The light-colored points are weak entities or non-entities assigned by the domain-specific knowledge base.

3.1. Model Architecture

For the effective use of weakly labeled data, we propose a new framework that operates as follows, which is illustrated in Figure 4. (1) We assigned the weak labels of the unlabeled corpus using the previously collected knowledge base. (2) Then, we first trained the NER model with strongly labeled data. This NER model is based on BERT [19], a pretrained language model, and is structured with a fully connected layer at the end of BERT. This paper aims to alleviate human annotating efforts. With a full understanding of the purpose, we tried to utilize the generalized structure of the model and the loss function. (3) Afterwards, we assigned labels to the predictions of the pretrained model for the unlabeled corpus. That is, we assigned the weak label in Stage 1 and the pseudo-label in Stage 2 to the unlabeled corpus. Finally, strongly labeled data and weakly labeled data were used together for model training. We repeated the process of (3) until the model converges. Here, we used strongly labeled data for training with supervised learning as in previous studies [10,26], but weakly labeled data were used only for consistency regularization.

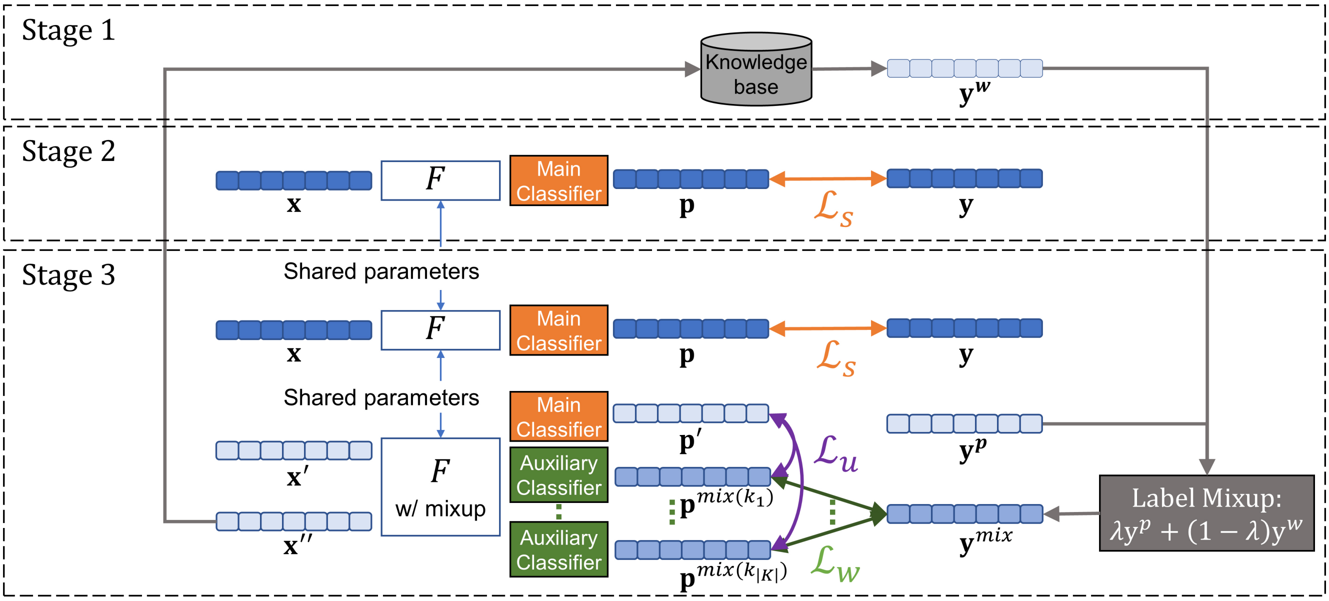

Figure 4. Framework definition.

The proposed model architecture, which is illustrated in Figure 4, is composed of three components. The first is F, a pretrained multi-layer language model. In this study, we use a variant of BERT that allows mixup to occur between examples in any intermediate k-th layer. The next one is a main classifier G, which performs the classification of the entity type of tokens using language representations of strongly labeled data. The last one is the auxiliary classifier, G_a, which can be more than one. The representations are mixed at layer k of F. Morevoer, let us consider K as a set of some k. The number of auxiliary classifiers would be $|K|$. In other words, the set of auxiliary classifiers is $G_a = \{G_a^{(k)} | k \in K\}$. The main and auxiliary classifiers can be various neural network layers. In this study, every classifier is stacked in the order of fully connected, tanh, dropout, and fully connected layers.

3.2. Stage 1: Weakly Labeling

First, we obtained a knowledge base of the target domain by collecting raw texts of the target-specific domain and generating a weak label. We created an entity dictionary by a greedy-search-based maximum matching algorithm [33] as a method to assign a weak label. The weakly labeling process converts D' to $D' = \{(\mathbf{x}'_1, \mathbf{y}^w_1), (\mathbf{x}'_2, \mathbf{y}^w_2), \ldots, (\mathbf{x}'_M, \mathbf{y}^w_M)\}$.

3.3. Stage 2: Pretraining

In Stage 2, language model F and main classifier G were trained with strongly labeled data. In this case, we used Focal Loss [34] as the loss function for training, which was first proposed in object detection tasks in computer vision. The focal loss is defined as follows:

$$Focal(p_t) = -\alpha(1 - p_t)^\gamma \log(p_t) \tag{2}$$

where p_t is prediction probability, and $\gamma(\gamma \geq 0)$ is the focus parameter. The focus parameter adjusts how much the model focuses on difficult samples. $\alpha \in [0, 1]$ is the weighting factor, which acts the same way as focus parameters. In this study, we used $\gamma = 2, \alpha = 1$.

This is known to be effective when the number of samples per class or the distribution of samples is imbalanced in the classification tasks, and NER tasks follow similar tendency. This is because the number of non-entity mentions is generally greater than the number of entity mentions. In such cases, the use of the common cross-entropy loss function would overwhelm the effects for simple samples that are easy to classify.

3.4. Stage 3: Pseudo-Labeling and Joint Training

3.4.1. Pseudo labeling

The next stage assigns pseudo-label $\mathbf{y}^p$ for D' for the model prediction trained in Section 3.3 ($D' = \{(\mathbf{x}'_1, \mathbf{y}^w_1, \mathbf{y}^p_1), (\mathbf{x}'_2, \mathbf{y}^w_2, \mathbf{y}^p_2), \ldots, (\mathbf{x}'_M, \mathbf{y}^w_M, \mathbf{y}^p_M)\}$). The pseudo-label, which is generated to obtain the diversity of the target label that has undergone the label mix process, used the softmax probability of the model without an argmax operation. The distribution of the target label would be simplified if the pseudo-label would be a one-hot format. We believed that the model trained with the soft label would be more robust to noise than the model trained with a one-hot label as in a prior study [15].

3.4.2. Joint Training

In this process, strongly labeled data and weakly labeled data are used together for training. For the strongly labeled data, we calculate the loss function as shown in Equation (2). We calculated losses $\mathcal{L}_u$ and $\mathcal{L}_w$, which contribute to the training of language model F and auxiliary classifiers G_a, from the weakly labeled data.

Firstly, we sample the data from D', which is twice the batch size. Then, the labels of the samples were sampled by pseudo-labels for one half and weak labels for the other half, which are represented as $(\mathbf{x}', \mathbf{y}^p)$ and $(\mathbf{x}'', \mathbf{y}^w)$, respectively. Then, F receives sampled sentences and generates hidden representations. The hidden representations for sentence $\mathbf{x}$, $\mathbf{x}'$, and $\mathbf{x}''$ from the l-th layer of F, which has L layers, are defined as follows:

$$\begin{aligned} \mathbf{h}_l &= F_l(\mathbf{h}_{l-1}; \theta), \qquad l \in [1, k] \\ \mathbf{h}'_l &= F_l(\mathbf{h}'_{l-1}; \theta), \qquad l \in [1, k] \\ \mathbf{h}''_l &= F_l(\mathbf{h}''_{l-1}; \theta), \qquad l \in [1, k] \end{aligned} \tag{3}$$

where F_l refers to a l-th layer of F having θ as parameters. $\mathbf{h}_0$ is the word embedding for the tokens of sentence $\mathbf{x}$, and $\mathbf{h}_l = \{h_1, h_2, \ldots, h_n\}$ is the l-th hidden representation of the sentence $\mathbf{x}$. Then, the intermediate representations continue forward passing to the upper layers in three different ways.

The first is $\mathbf{h}_L$ to calculate the loss on strongly labeled data. $\mathbf{h}_L$ is the result of passing through all remaining F layers without mixing up even after the k-th layer. $\mathbf{h}_L$ is defined as follows.

$$\begin{aligned} \tilde{\mathbf{h}}_l &= F_l(\tilde{\mathbf{h}}_{l-1}; \theta), \qquad l \in [k+1, L] \\ \mathbf{h}_L &= F_L(\tilde{\mathbf{h}}_{L-1}; \theta) \end{aligned} \tag{4}$$

The second is $\mathbf{h}_L^{mix}$, which is linearly interpolated with $\mathbf{h}'_k$ and $\mathbf{h}''_k$ via the mixup. $\mathbf{h}'_k$ and $\mathbf{h}''_k$ refer to k-th representations of the language model from Equation (3). and then continues forward to the remaining layers of F:

$$\begin{aligned} \tilde{\mathbf{h}}_k^{mix} &= \lambda \mathbf{h}'_k + (1-\lambda)\mathbf{h}''_k \\ \tilde{\mathbf{h}}_l^{mix} &= F_l(\tilde{\mathbf{h}}_{l-1}^{mix}; \theta), \qquad l \in [k+1, L] \\ \mathbf{h}_L^{mix} &= F_L(\tilde{\mathbf{h}}_{L-1}^{mix}; \theta) \end{aligned} \tag{5}$$

where λ is the same as Equation (1). In this study, λ follows $\mathrm{Beta}(0.75, 0.75)$. The last one is $\mathbf{h}'_L$ from $\mathbf{h}'_l$, which is a result of passing through the remaining layers of F without mixup.

$$\begin{aligned} \tilde{\mathbf{h}}'_l &= F_l(\tilde{\mathbf{h}}'_{l-1}; \theta), \qquad l \in [k+1, L] \\ \mathbf{h}'_L &= F_L(\tilde{\mathbf{h}}'_{L-1}; \theta) \end{aligned} \tag{6}$$

In training, models F and G are supervised by strongly labeled data in Stage 2. The supervised loss for the strongly labeled data is defined as $\mathcal{L}_s$.

$$\begin{aligned} \mathbf{p} &= G(\mathbf{h}_L) = G \cdot F(\mathbf{x}) \\ \mathcal{L}_s &= Focal(\mathbf{y}, \mathbf{p}) \end{aligned} \tag{7}$$

For weakly labeled data, F and G_a are supervised. We use the same loss function as mentioned above. Note that we use mixed labels that are linearly interpolated with pseudo-labels and the weak label. The loss, $\mathcal{L}_w$, for the auxiliary classifier is defined as follows:

$$\begin{aligned} \mathbf{p}^{mix(k)} &= G_a^{(k)}(\mathbf{h}_L^{mix(k)}) \\ \mathcal{L}_w &= \sum_{k \in K} Focal(\tilde{\mathbf{y}}, \mathbf{p}^{mix(k)}) \end{aligned} \tag{8}$$

where upper subscript (k) denotes the language model that mixes up examples in the k-th layer. $\tilde{\mathbf{y}} = \lambda \mathbf{y}^p + (1-\lambda)\mathbf{y}^w$ refers to the mixed label. Note that $\mathbf{y}^p$ and $\mathbf{y}^w$ do not belong to the same sentence. We visualized the entire mixup process in Figure 5.

F and G_a also use supervision from consistency regularization. With various mixup settings, we can obtain $|K|$ perturbed versions of $\mathbf{h}_L^{mix(k)}$. Moreover, auxiliary classifiers output predictions from various hidden representations. Consistency loss $\mathcal{L}_u$ calculates the consistencies between prediction from the main classifier and the auxiliary classifier. By doing so, we can make F robust to the perturbations on the same example:

$$\begin{aligned} \mathbf{p}' &= G(\mathbf{h}'_L) = G \cdot F(\mathbf{x}') \\ \mathcal{L}_u &= \sum_{k \in K} \mathrm{MSE}(\mathbf{p}', \mathbf{p}^{mix(k)}) \end{aligned} \tag{9}$$

where MSE denotes the mean squared error. We do not backpropagate $\mathcal{L}_u$ to the main classifier so that the main classifier leads the auxiliary classifier. Lastly, the final loss is a sum of the losses mentioned above.

$$\mathcal{L} = \mathcal{L}_s + \mathcal{L}_w + \mathcal{L}_u \tag{10}$$

Since the first pseudo-labeling only reflects the supervision for strongly labeled data, we assumed that the pseudo-label generated by only strong supervision would provide the biased pseudo-label for the small strongly labeled data. Therefore, during training, we repeated the pseudo-labeling process periodically. That is, we performed pseudo-labeling once, and we performed joint training ϕ times until the model reached convergence. In testing, the proposed model outputs the prediction of sample $\mathbf{x}^*$ from the main classifier, i.e., $G \cdot F(\mathbf{x}^*)$.

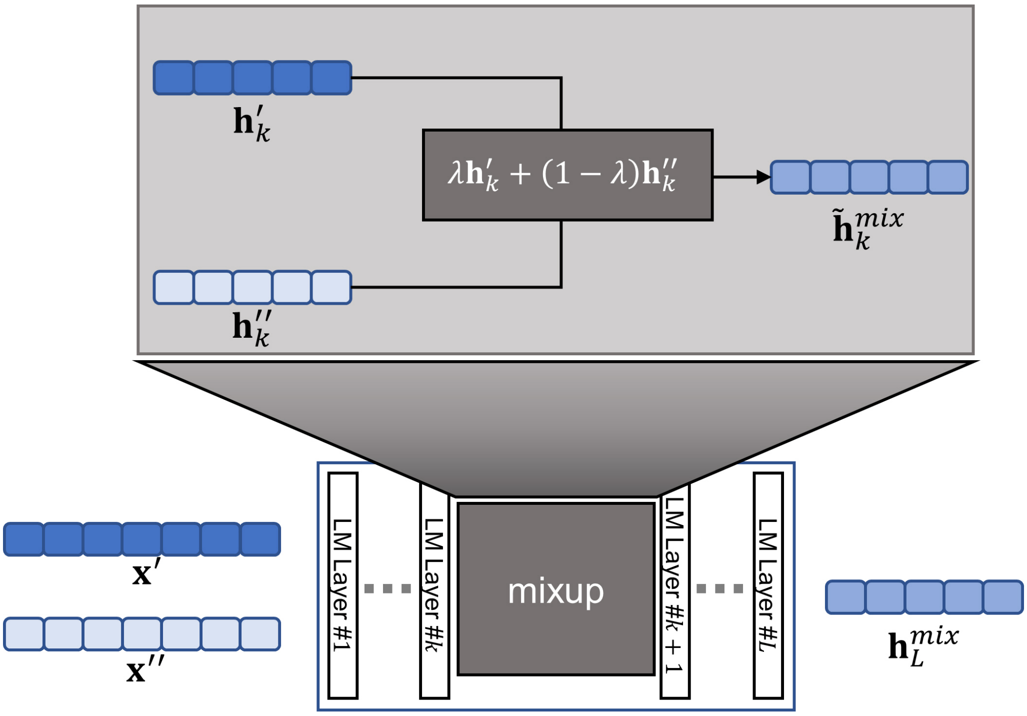

Figure 5. Illustration of the mixup process. We use the TMix [18] strategy to mix text examples.

4. Experiments

We show the effectiveness of the proposed method by using the results of experiments on the biomedical domain NER and tech domain NER.

4.1. Datasets

For biomedical NERs, we used BC5CDR-chem, BC5CDR-disease, and NCBI-disease, which are broadly used as benchmark datasets in the biomedical NER task. These datasets consist of PubMed articles and have single entity-type mentions: disease or chemical. The NCBI-disease [35] is a collection of PubMed articles with human-annotated mentions. We split BC5CDR [36], which is proposed for disease mention recognition and chemically induced disease relation-extraction tasks into chemical and disease entity recognition datasets. We collected unlabeled raw texts from the PubMed 2021 baseline and used a combination of the MeSH database and CTD chemical and disease vocabulary as a knowledge base to generate weak labels. For the tech NER, we used LaptopReview [37]. LaptopReview consists of English product reviews that are annotated on laptop domains. It has a single entity type, AspectTerm, at the sentence level. The raw text for the tech term NER is the laptop subset of Amazon Reviews [38]. It also contains review texts about laptop products obtained from Amazon websites. The statistics of the datasets after weakly labeling process are shown in Table 1.

Table 1. The statistical results on experimental datasets.

Dataset	The Number of Examples			
	Train	Dev	Test	Weak
BC5CDR-chem	4560	4581	4797	107,827 [1]
BC5CDR-disease	4560	4581	4797	67,182 [1]
NCBI-disease	5424	923	940	67,182 [1]
LaptopReview	2436	609	800	28,063 [2]

[1] PubMed corpus + the combination of MeSH and CTD dictionary. [2] Subset laptop of Amazon Reviews + Tech terms crawled on public website.

4.2. Experimental Setup

In the experiments conducted in this study, the weight of BERT varies depending on the dataset. Biomedical NER used the weights of BioBERT [39] pretrained on biomedical corpus, while tech term NER used "bert-base-cased" weights. When using only strongly labeled data, we trained the models for one epoch with a batch size of 128, while we used a batch size of 32 for one epoch for the models using both strongly labeled data and weakly labeled data (16 for strong and 16 for weak). For a fair comparison, we evaluate Dev and Test set using batch size 16 for all experiments. We used a maximum sentence length of 256 tokens in all experiments. The learning rate is fixed on 5×10^{-5} in the pretraining stage. In contrast, for iterative Stage 3, we conducted pseudo-labeling after performing joint training 5 times ($\phi = 5$). And during Stage 3, we used learning rate schedule with the warm-up in 10% of steps, peaked at 5×10^{-5}, and cool-down in 90% of steps. The Adam optimizer was used for optimization. Following the studies in [18,40], we mixed hidden representations at the 7-th, 9-th, and 12-th layers, which have the most representation power of the language model, i.e., $K = \{7, 9, 12\}$.

We compared the proposed framework with the following options to evaluate the proposed framework:

- BERT baseline: A supervised learning baseline. We construct the representative baseline model for NER. The BERT baseline is structured with the multi-layer pretrained model BERT and some fully connected layers. That is, we stacked the main classifier on top of the language model.
- Mixup baseline: A baseline to show the effect of mixup on the NER task. The mixup baseline has the same structure as the BERT baseline. However, it is trained with examples generated by mixup as well as strongly labeled data.
- Mixup-CCT: The method proposed in [1]. CCT means cross-consistency training. The architecture is the same as the one proposed in this paper. There is no pseudo-labeling; instead, it gradually increases the affection of the loss $\mathcal{L}_w$ and $\mathcal{L}_u$.
- Proposed: It performs consistency regularization from mixed examples generated from the pseudo-label and weak label, while mixup-CCT is mixed by strongly labeled and weakly labeled data.

For every method except mixup-CCT, we used the focal loss for supervision on strongly labeled data. The results of mixup-CCT were derived from [1].

We used span-level Precision, Recall, and F1-score as evaluation metrics, which are the most significant metrics for classification tasks. First of all, we obtained results with respect to True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) for each span in sentences. Then, using their notions, we calculated Precision, Recall, and F1-score as follows.

$$\begin{aligned} \text{Precision} &= \frac{\text{TP}}{\text{TP} + \text{FP}} \\ \text{Recall} &= \frac{\text{TP}}{\text{TP} + \text{FN}} \\ \text{F1-score} &= 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \end{aligned} \tag{11}$$

4.3. Results

Table 2 presents the experimental results. Weakly labeled data, which have undergone automatic assignment of labels, contain noise that hinders training [12]. Therefore, rather than directly using weakly labeled data for training, it seemed to be a convincing alternative to create and use new data by mixing weakly and strongly labeled data. Interestingly, despite using more data than the BERT baseline, the mixup baseline had the lowest performance in F1-scores on all benchmark datasets. The results support our assumption that the mixup technique for data augmentation in NER tasks worsens the model's performance if it is used inappropriately. The proposed framework performed the overall best F1-score on BC5CDR-chem, NCBI-disease, and LaptopReview. For BC5CDR-disease, mixup-CCT performed better with F1-score while the proposed performed better with Recall. On the other hand, for BC5CDR-chem and LaptopReview, we observed that the proposed method achieved the best score for all evaluation metrics. Moreover, the proposed showed the smallest difference between Precision and Recall except for BC5CDR-disease, which suggests that training has been conducted stably without being biased on one metric.

Table 2. Results of the test set's span-level Precision/Recall/F1-score when the F1-score for the Dev set was the highest during training. The best performance is bolded. Pre, rec, and F1 mean a Precision, Recall, and F1-score, respectively.

Method	**BC5CDR-Chem**			**BC5CDR-Disease**			**NCBI-Disease**			**LaptopReview**		
	Pre	**Rec**	**F1**	**Pre**	**Rec**	**F1**	**Pre**	**Rec**	**F1**	**Pre**	**Rec**	**F1**
BERT baseline	90.88	92.08	91.38	83.44	85.40	84.32	83.69	**90.91**	86.86	79.76	79.74	79.72
mixup baseline	90.95	92.49	91.21	83.65	85.53	84.11	85.94	87.76	86.60	80.56	77.49	78.54
mixup-CCT	91.98	92.78	92.19	**86.69**	85.46	**85.86**	**88.64**	89.23	88.82	83.40	78.09	80.46
proposed	**92.12**	**93.14**	**92.23**	84.19	**86.81**	85.03	87.98	90.25	**88.92**	**84.85**	**80.90**	**82.24**

4.4. Low Resource Environments

To demonstrate that the proposed framework is robust to the size of strongly labeled data, we test the performance with the sub-split of 20%, 40%, 60%, and 80% of strongly labeled data. For a fair comparison between experiments, we used the same sample indices when splitting datasets.

As shown in Figure 6, the proposed framework demonstrated better performance than the BERT baseline on all experiments. Particularly for NCBI-disease and LaptopReview, while the performance of BERT baseline dramatically decreased on 20% of the sub-sampled strongly labeled dataset, the proposed framework relatively maintained stable performances. Furthermore, we also observed that as the strongly labeled data size decreases, the performance decline decreases. Based on experimental results under low-resource environments, the proposed framework showed similar performances to environments where sufficient datasets are available, even with limited datasets. This demonstrates that the proposed framework would save the cost of assigning datasets for domain-specific NER tasks.

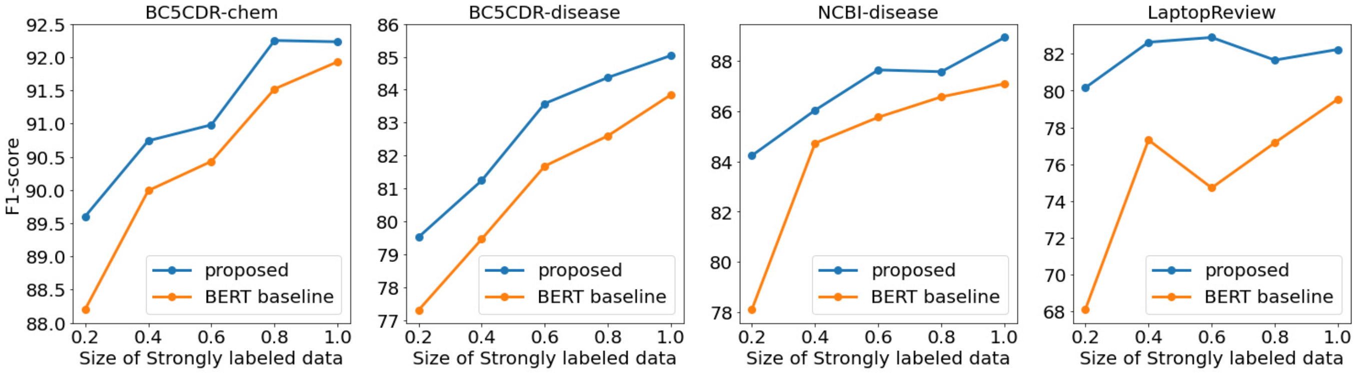

Figure 6. The trends of the test set's span-level F1-score according to the size of the strongly labeled dataset. We described detailed numbers in Appendix A.

4.5. Comparison of Embedding Vectors

Because we exploited shallow classifiers, such as two fully connected layers, as the main classifier and auxiliary classifiers, the embedding extracted from the language model is important for classifying entity labels. When the language model learns the entity embedding space better, well-projected embeddings make classifiers assign entity mentions more [41]. To demonstrate that the language model in the proposed framework finds the appropriate embedding for classifiers to find entity mentions, we projected span-level representations obtained from the language model. Figure 7 presents the visualization results onto a two-dimensional space using t-SNE [42] on the BC5CDR-chem test sets. As shown in Figure 7, some B and I embeddings were projected in the O cluster. We even found the embedding of the entity mention located in the middle of the O cluster. It can be assumed that some incorrectly located embeddings make it difficult for the classifier to assign appropriate labels to them. On the other hand, the proposed framework projected each entity and non-entity mention in the embedding space quite appropriately. Along with the experimental results shown in Table 2, this supports our argument that the proposed framework better conducts NER tasks than the BERT baseline.

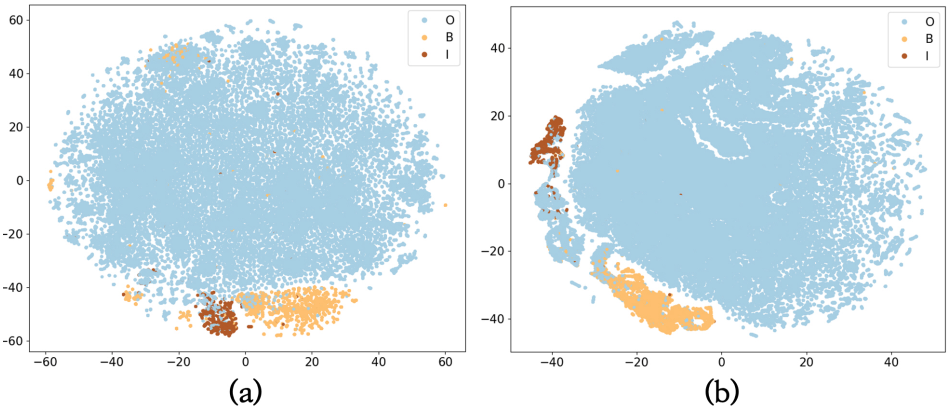

Figure 7. t-SNE visualization of BC5CDR-chem. The representations are extracted from language model F. (**a**) shows the result of the BERT baseline, and (**b**) shows the result of the proposed framework.

5. Conclusions

Mixup, which generates a new data point from mixing two different data points, was proposed as a data augmentation technique in computer vision. Although several prior studies showed the effectiveness of mixup, it would lead to the NER model training in an undesirable direction if we use it directly on NER examples. Based on the hinderence this mixup results in when applied on NER examples, we thought mixup could be a perturbation on NER examples and used it as a perturbation function for consistency regularization. In this study, we proposed a novel framework that uses the mixup based the consistency regularization technique for Named Entity Recognition tasks. To validate our proposed framework, we conducted experiments on domain-specific NER benchmark datasets. The experimental results demonstrated that the mixup for weakly labeled data would act as a perturbation function for NER examples, especially on low-resource environments. We expect that the proposed framework can mitigate efforts for building datasets for domain-specific NER tasks.

Author Contributions: Conceptualization, G.Y. and J.R.; methodology G.Y.; software, G.Y.; validation, B.Y.; investigation, S.J. and D.K.; data curation, G.Y.; writing—original draft preparation, G.Y.; writing—review and editing, G.Y., B.Y., S.J. and D.K.; visualization, G.Y. and D.K.; supervision, J.R.; project administration, J.R.; funding acquisition, J.R. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Technology development Program (S2957729) funded by the Ministry of SMEs and Startups (MSS, Sejong, Korea).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Publicly available datasets were analyzed in this study. For biomedical domains, the dataset can be found here: https://github.com/cambridgeltl/MTL-Bioinformatics-2016, accessed on 29 October 2022. For weakly labeled data, we used PubMed 2021 baseline (https://ftp.ncbi.nlm.nih.gov/pubmed/, downloaded on 11 February 2022 and accessed on 29 October 2022) as the biomedical raw corpus. we used a combination of the MeSH database (https://www.nlm.nih.gov/databases/download/mesh.html, downloaded on 11 February 2022 and accessed on 29 October 2022) and CTD Chemical and Disease vocabulary (http://ctdbase.org/downloads/, downloaded on 11 February 2022 and accessed on 29 October 2022) to build a entity dictionary. For tech term domains, the dataset can be found here: https://alt.qcri.org/semeval2014/task4/index.php?id=data-and-tools, accessed on 29 October 2022. For the tech term dictionary, we searched public websites (https://www.computerhope.com/jargon.htm, accessed on 29 October 2022). The laptop subset of Amazon Reviews is downloaded on 10 February 2022 from the following: http://jmcauley.ucsd.edu/data/amazon/index_2014.html, accessed on 29 October 2022.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

Here, we demonstrate detailed numbers in Figure 6.

Table A1. Table of performance according to the size of strongly labeled data. Pre, rec, and f1 denote Precision, Recall, and F1-score, respectively.

Dataset	Method	20%			40%			60%			80%		
		Pre	Rec	F1	Pre	Rec	F1	Pre	Rec	F1	Pre	Rec	F1
BC5CDR-chem	BERT baseline	87.86	89.31	88.20	89.48	90.66	89.99	89.31	91.72	90.43	91.15	92.05	91.52
	proposed	88.91	91.31	89.60	90.77	91.55	90.74	90.65	92.36	90.98	92.27	93.08	92.25
BC5CDR-disease	BERT baseline	75.86	78.99	77.30	77.88	81.27	79.46	80.05	83.55	81.68	81.43	83.96	82.60
	proposed	79.65	80.59	79.52	81.09	82.59	81.24	83.18	84.99	83.58	83.87	85.76	84.37
NCBI-disease	BERT baseline	73.10	83.97	78.12	80.85	89.05	84.72	84.36	87.28	85.75	84.26	89.06	86.56
	proposed	82.80	86.20	84.23	85.66	86.85	86.03	86.03	89.62	87.63	86.22	89.35	87.56
LaptopReview	BERT baseline	68.46	67.84	68.12	75.07	79.79	77.32	74.81	74.63	74.72	76.04	78.45	77.17
	proposed	80.73	80.24	80.16	83.95	82.13	82.63	82.96	83.56	82.89	81.77	82.46	81.66

References

1. Youn, G.; Yoon, B.; Ji, S.; Ko, D.; Rhee, J. MixUp based Cross-Consistency Training for Named Entity Recognition. In Proceedings of the 6th International Conference on Advances in Artificial Intelligence, Birmingham, UK, 21–23 October 2022.
2. Danger, R.; Pla, F.; Molina, A.; Rosso, P. Towards a Protein–Protein Interaction information extraction system: Recognizing named entities. *Knowl.-Based Syst.* **2014**, *57*, 104–118. [CrossRef]
3. Mollá, D.; Van Zaanen, M.; Smith, D. Named entity recognition for question answering. In Proceedings of the Australasian Language Technology Workshop 2006, Sydney, Australia, 11 November 2006; pp. 51–58.
4. Chen, Y.; Zong, C.; Su, K.Y. A joint model to identify and align bilingual named entities. *Comput. Linguist.* **2013**, *39*, 229–266. [CrossRef]
5. Baralis, E.; Cagliero, L.; Jabeen, S.; Fiori, A.; Shah, S. Multi-document summarization based on the Yago ontology. *Expert Syst. Appl.* **2013**, *40*, 6976–6984. [CrossRef]
6. Nobata, C.; Sekine, S.; Isahara, H.; Grishman, R. Summarization System Integrated with Named Entity Tagging and IE pattern Discovery. In Proceedings of the Third International Conference on Language Resources and Evaluation (LREC'02), Las Palmas, Spain, 28 May–3 June 2002; European Language Resources Association (ELRA): Las Palmas, Spain, 2002.
7. Chiu, J.P.; Nichols, E. Named entity recognition with bidirectional LSTM-CNNs. *Trans. Assoc. Comput. Linguist.* **2016**, *4*, 357–370. [CrossRef]
8. Collobert, R.; Weston, J.; Bottou, L.; Karlen, M.; Kavukcuoglu, K.; Kuksa, P. Natural Language Processing (Almost) from Scratch. *J. Mach. Learn. Res.* **2011**, *12*, 2493–2537.
9. Huang, Z.; Xu, W.; Yu, K. Bidirectional LSTM-CRF models for sequence tagging. *arXiv* **2015**, arXiv:1508.01991.
10. Li, J.; Sun, A.; Han, J.; Li, C. A survey on deep learning for named entity recognition. *IEEE Trans. Knowl. Data Eng.* **2020**, *34*, 50–70. [CrossRef]
11. Fang, Z.; Cao, Y.; Li, T.; Jia, R.; Fang, F.; Shang, Y.; Lu, Y. TEBNER: Domain Specific Named Entity Recognition with Type Expanded Boundary-aware Network. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, Punta Cana, Dominican Republic, 7–11 November 2021; pp. 198–207. [CrossRef]
12. Jiang, H.; Zhang, D.; Cao, T.; Yin, B.; Zhao, T. Named Entity Recognition with Small Strongly Labeled and Large Weakly Labeled Data. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), Virtual Event, 1–6 August 2021; pp. 1775–1789. [CrossRef]
13. Liu, S.; Sun, Y.; Li, B.; Wang, W.; Zhao, X. HAMNER: Headword amplified multi-span distantly supervised method for domain specific named entity recognition. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 8401–8408.
14. Shang, J.; Liu, L.; Ren, X.; Gu, X.; Ren, T.; Han, J. Learning named entity tagger using domain-specific dictionary. *arXiv* **2018**, arXiv:1809.03599.
15. Liang, C.; Yu, Y.; Jiang, H.; Er, S.; Wang, R.; Zhao, T.; Zhang, C. Bond: Bert-assisted open-domain named entity recognition with distant supervision. In Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Virtual Event, 6–10 July 2020; pp. 1054–1064.
16. Ouali, Y.; Hudelot, C.; Tami, M. An overview of deep semi-supervised learning. *arXiv* **2020**, arXiv:2006.05278.
17. Zhang, H.; Cisse, M.; Dauphin, Y.N.; Lopez-Paz, D. mixup: Beyond empirical risk minimization. *arXiv* **2017**, arXiv:1710.09412.
18. Chen, J.; Yang, Z.; Yang, D. MixText: Linguistically-Informed Interpolation of Hidden Space for Semi-Supervised Text Classification. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 5–10 July 2020; pp. 2147–2157. [CrossRef]
19. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805.
20. Chen, J.; Wang, Z.; Tian, R.; Yang, Z.; Yang, D. Local Additivity Based Data Augmentation for Semi-supervised NER. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Online, 16–20 November 2020; pp. 1241–1251. [CrossRef]
21. Zhang, R.; Yu, Y.; Zhang, C. SeqMix: Augmenting Active Sequence Labeling via Sequence Mixup. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Online, 16–20 November 2020; pp. 8566–8579.
22. Pinto, F.; Yang, H.; Lim, S.N.; Torr, P.H.; Dokania, P.K. RegMixup: Mixup as a Regularizer Can Surprisingly Improve Accuracy and Out Distribution Robustness. *arXiv* **2022**, arXiv:2206.14502.
23. Xie, Q.; Luong, M.T.; Hovy, E.; Le, Q.V. Self-training with noisy student improves imagenet classification. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 10687–10698.
24. Li, Z.z.; Feng, D.W.; Li, D.S.; Lu, X.C. Learning to select pseudo labels: A semi-supervised method for named entity recognition. *Front. Inf. Technol. Electron. Eng.* **2020**, *21*, 903–916. [CrossRef]
25. Gaur, B.; Saluja, G.S.; Sivakumar, H.B.; Singh, S. Semi-supervised deep learning based named entity recognition model to parse education section of resumes. *Neural Comput. Appl.* **2021**, *33*, 5705–5718. [CrossRef]
26. Chen, H.; Yuan, S.; Zhang, X. ROSE-NER: Robust Semi-supervised Named Entity Recognition on Insufficient Labeled Data. In Proceedings of the The 10th International Joint Conference on Knowledge Graphs, Virtual Event, 6–8 December 2021; pp. 38–44.

27. French, G.; Laine, S.; Aila, T.; Mackiewicz, M.; Finlayson, G. Semi-supervised semantic segmentation needs strong, varied perturbations. *arXiv* **2019**, arXiv:1906.01916.
28. Ouali, Y.; Hudelot, C.; Tami, M. Semi-supervised semantic segmentation with cross-consistency training. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 12674–12684.
29. Clark, K.; Luong, M.T.; Manning, C.D.; Le, Q. Semi-Supervised Sequence Modeling with Cross-View Training. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, 31 October–4 November 2018; Association for Computational Linguistics: Brussels, Belgium, 2018; pp. 1914–1925. [CrossRef]
30. Berthelot, D.; Carlini, N.; Goodfellow, I.; Papernot, N.; Oliver, A.; Raffel, C.A. Mixmatch: A holistic approach to semi-supervised learning. *Adv. Neural Inf. Process. Syst.* **2019**, *32* , 5050–5060.
31. Berthelot, D.; Carlini, N.; Cubuk, E.D.; Kurakin, A.; Sohn, K.; Zhang, H.; Raffel, C. Remixmatch: Semi-supervised learning with distribution alignment and augmentation anchoring. *arXiv* **2019**, arXiv:1911.09785.
32. Sohn, K.; Berthelot, D.; Carlini, N.; Zhang, Z.; Zhang, H.; Raffel, C.A.; Cubuk, E.D.; Kurakin, A.; Li, C.L. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 596–608.
33. Peng, M.; Xing, X.; Zhang, Q.; Fu, J.; Huang, X. Distantly Supervised Named Entity Recognition using Positive-Unlabeled Learning. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 28 July–2 August 2019; Association for Computational Linguistics: Florence, Italy, 2019; pp. 2409–2419. [CrossRef]
34. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal loss for dense object detection. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2980–2988.
35. Doğan, R.I.; Leaman, R.; Lu, Z. NCBI disease corpus: A resource for disease name recognition and concept normalization. *J. Biomed. Inform.* **2014**, *47*, 1–10. [CrossRef]
36. Li, J.; Sun, Y.; Johnson, R.J.; Sciaky, D.; Wei, C.H.; Leaman, R.; Davis, A.P.; Mattingly, C.J.; Wiegers, T.C.; Lu, Z. BioCreative V CDR task corpus: A resource for chemical disease relation extraction. *Database* **2016**, *2016* , baw068. [CrossRef]
37. Pontiki, M.; Galanis, D.; Papageorgiou, H.; Androutsopoulos, I.; Manandhar, S.; Al-Smadi, M.; Al-Ayyoub, M.; Zhao, Y.; Qin, B.; De Clercq, O.; et al. Semeval-2016 task 5: Aspect based sentiment analysis. In Proceedings of the International Workshop on Semantic Evaluation, San Diego, CA, USA, 16–17 June 2016; pp. 19–30.
38. Wang, H.; Lu, Y.; Zhai, C. Latent aspect rating analysis without aspect keyword supervision. In Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Diego, CA, USA, 21–24 August 2011; pp. 618–626.
39. Lee, J.; Yoon, W.; Kim, S.; Kim, D.; Kim, S.; So, C.H.; Kang, J. BioBERT: A pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics* **2020**, *36*, 1234–1240. [CrossRef]
40. Jawahar, G.; Sagot, B.; Seddah, D. What does BERT learn about the structure of language? In Proceedings of the ACL 2019—57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 28 July–2 August 2019.
41. Lee, H.G.; Park, G.; Kim, H. Effective integration of morphological analysis and named entity recognition based on a recurrent neural network. *Pattern Recognit. Lett.* **2018**, *112*, 361–365. [CrossRef]
42. Van der Maaten, L.; Hinton, G. Visualizing data using t-SNE. *J. Mach. Learn. Res.* **2008**, *9*, 2579–2605.