Debiased and Denoised Entity Recognition from Distant Supervision (Appendix)

Anonymous Author(s)
Affiliation
Address
email

A Dataset-Task Taxonomy

A.1 Dataset Taxonomy

We provide Figure 5 to show the dataset taxonomy in the overall training workflow. We first split the initial dataset $\mathcal{D}$ into two sub-datasets $\mathcal{D}_b$ and $\mathcal{D}_e$ through decoupled learning to mitigate the distributional bias (first level split). Next, to process respective noise in $\mathcal{D}_b$ and $\mathcal{D}_e$, we conduct a selection and self-training framework. We divide the two parts into the clean set and noisy set by clean token selection (second level split), then perform a standard classification on each clean set. As for noisy token sets, we adopt debiased self-training to yield pseudo-labels for further training.

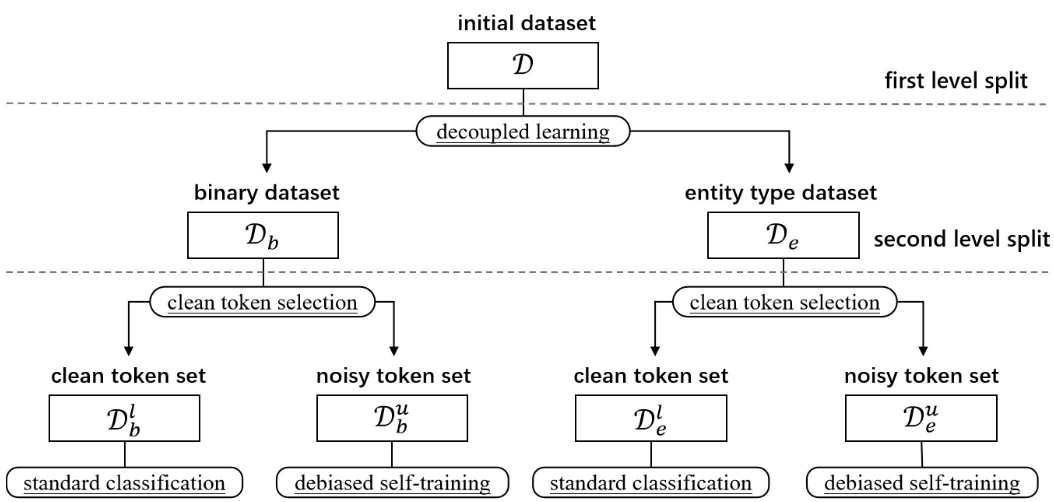

Figure 5: Illustration of dataset taxonomy.

A.2 Task Taxonomy and Model architecture

DesERT modifies the basic NER model architecture with the double-head pathway, yet reserves a shared pre-trained language model encoder such as RoBERTa-base denoted by ϕ. Given any sentence $\boldsymbol{x}$ with its binarilized labels $\boldsymbol{y}^b$ and entity type labels $\boldsymbol{y}^e$, $(\boldsymbol{x}, \boldsymbol{y}^b) \in \mathcal{D}^b$ and $(\boldsymbol{x}, \boldsymbol{y}^e) \in \mathcal{D}^e$. $\boldsymbol{x}$ is first fed to the PLM encoder ϕ and we take the last hidden layer output of ϕ as finally embeddings $\phi(\boldsymbol{x})$. Then the double-head pathway h_b and h_e take $\phi(\boldsymbol{x})$ as input to yield respective predictions. The binary pathway h_b generates the probablity of being entity tokens, $\boldsymbol{p}^b = \text{sigmoid}(h_b \circ \phi(\boldsymbol{x})) = [p_1^b, ..., p_n^b]$, then take $\boldsymbol{p}^b > 0.5$ as predicted binary labels $\hat{\boldsymbol{y}}^b$. While the entity pathway h_e offers fine-grained entity type probablities $\boldsymbol{p}^e = \text{softmax}(h_e \circ \phi(\boldsymbol{x})) = [\boldsymbol{p}^e_1, ..., \boldsymbol{p}^e_n]$, where each $\boldsymbol{p}^e_i$ has K entries. We take arg max$(\boldsymbol{p}^e)$ as predicted entity type labels $\hat{\boldsymbol{y}}^e$, and note that non-entity tokens are tagged with invalid labels. Finally, any standard classification loss can be calculated on the double-head pathway. We also refer the reader to Figure 1 for visualized illustration.

B Theoretical Insights of Debiased Semi-supervised Learning

Though there is no available theory on why the worst cross-entropy (WCE) [1] works, we would like to provide the following (relatively) theoretical insights that may help the readers to perceive our approach better.

Notably, the self-training bias is mainly caused by noisy tokens approaching the decision boundaries, whose pseudo-labels keep changing. To this end, we optimize WCE to learn compact token clusters for reducing wrong pseudo-label assignments. To see this, we start from the following simplified example to show that WCE does indeed concentrates the token representation.

Assumption. Consider a binary classification problem, as the simplest form of DSNER. Denote the input variable by X and the output variable by Y from binary labels $\{+1, -1\}$. The labeled data $\mathcal{D}^l$ are sampled from $X|Y = +1 \sim U(\mathcal{B}(u, r))$ and $X|Y = -1 \sim U(\mathcal{B}(v, r))$. Here $\mathcal{B}(u, r)$ denotes a spherical ball with center u and radius r. U denotes uniform distributions. The unlabeled sample $\mathcal{D}^u$ are all from $Y = +1$ but uniformly distributed inside the $\mathcal{B}((u + v)/2, r')$. Three balls have no intersection. Finally, we assume the maximum margin classifier is used, which is a hard proxy of the cross-entropy loss.

Derivation Sketch. At first glance, it is obvious that the optimal classifier on D^l is $f = (u + v)/2$ which misclassifies half of the examples in D^u. The worst classifier, however, amounts to be $f^w = (u + v)/2 - r'$ which perfectly classifies D^l but possesses the most side-way decision boundary. Next, we optimize the feature extractor ϕ to match the worst decision boundary by $\min_\phi L_U(y, f^w)$. This is to say, with ideally known labels, the unlabeled ball $\mathcal{B}((u + v)/2, r')$ converges to $\mathcal{B}(f = (u + v)/2 - r', r')$. With full samples $X|Y = +1$ getting closer, the classifier achieves better generalization with *compact clusters* and low-entropy decision boundaries.

In practice, since the true labels are unknown, we use pseudo-labels as a proxy since most unlabeled data are assigned true labels. So, the representation will be partially concentrated.

Empirical Covariance. We conduct experiments on CoNLL03 to show the covariance of data to their class centers and the quality of pseudo labels as follows. When DeSERT is run without WCE, the average covariance amongst classes is 0.0085. With WCE, the average covariance amongst classes becomes 0.0056. Thereafter, we can conclude that WCE indeed concentrates the tokens and mitigates the self-training bias.

C Additional Experimental Setups and Results

In what follows, we show more experimental details and results. In section C.1, we report more empirical results, including an interesting series of experiments where *additional distant supervision* comes from large language models like ChatGPT. In section C.2, we provide more details on our experiments and implementation.

C.1 Results with Additional Distant Supervision from ChatGPT

Recently, large language models (LLMs), including GPT-3 [2], ChatGPT, and GPT-4[1], have largely revolutionized the NLP landscape. Thanks to their emerging abilities like in-context learning (ICL) [3] and chain-of-thought [4], LLMs demonstrate remarkable zero-shot learning performance in a wide range of downstream NLP tasks. Despite the promise, some recent studies [5] have shown that LLMs are still legs behind the fine-tuned small language models in many NLP applications.

Motivated by this, we conduct experiments to show the zero-shot performance of ChatGPT on the NER problem. In Table 3, we observe that ChatGPT does indeed demonstrates inferior results even

[1]https://openai.com/blog/chatgpt

Prompted Input: You are required to produce 100 English sentences with annotations for the task of named entity recognition on the CONLL dataset. These samples will be used as prompt examples for the GPT model. There are four kinds of entity types:

'PER' for person, 'LOC' for location, 'ORG' for organization, and 'MISC' for other miscellaneous specific entities (e.g.,specific events, nationalities, products, or works of art..).

Entity Types Def.

The task is to classify the named entities in the given sentence as as one of the mentioned entity types. For example, 10 of the unlabeled samples in CONLL dataset are as follows:

[EU rejects German call to boycott British lamb .]...

Unsupervised Demo

Now you are required to mimic the style of these samples and produce 100 English labeled examples on CONLL dataset. Please ensure that only the mentioned entities are included. Do not generate samples that are too similar. It is important to maximize both the diversity of samples and the accuracy of labeling. The output format should be in this form, each sample is output in a single line:

[(example 1) | (entity content 1, entity type 1); (entity content 2, entity type 2)], (entity content 1, entity type 1); ...

Output Formatting

LLMs: [Jack decided to move to Paris for his new job] | (Jack, PER), (Paris, LOC)[The World Health Organization has declared a global pandemic.] | ...

Generated Samples

Figure 6: An example prompt for automatic demonstration data generation.

compared with distantly-supervised SLMs. Therefore, a question arises: *how can LLMs better support NER with minimal human annotation?* Among the numerous potential solutions, we propose a natural extension of the conventional distantly-supervised NER problem. This extension considers LLM's predictions on the training set as additional distant supervision. To achieve this, we present a new tagging scheme for NER and modified our algorithm to accommodate multiple sources of distant supervision. Subsequently, we provide a detailed elaboration on the aforementioned aspects.

C.1.1 Tagging by Auto-Demonstration

Instead of directly performing zero-shot testing, we introduce a new DSNER paradigm that generates distant labels by LLMs for *unsupervised training data* to improve the SLMs' performance. To generate distant labels, a straightforward solution is to send the raw training texts to the LLMs and ask them to output all the entities along with their types. However, we find such a naive strategy fails to achieve satisfactory NER performance. To mitigate this problem, we design a novel in-context learning algorithm that exploits self-generated text-tag pairs to guide the tagging process.

Automatic Text-Tag Pair Generation. Our ultimate goal is to perform few-shot in-context learning that better guides the LLMs to locate the entities and output their types. To achieve this, we may assume a set of demonstration text-tag pair samples are available. Besides, this set ought to be diverse and representative enough to guide in-context learning. However, the training samples are unsupervised and cannot be utilized directly. To address this problem, we propose to generate sentences by ChatGPT itself, while ensuring the diversity of the generated results to cover all entity types. Additionally, we randomly retrieve unlabeled samples from the training set and employ ChatGPT to automatically generate a comprehensive set of sentences, including their corresponding entity tags. An example prompt is shown in Figure 6.

Few-Shot In-Context Learning for NER Tagging. After that, we ask the ChatGPT to tag the whole training set by using its self-generated demonstrations. However, directly feeding all the generated samples for ICL may exceed contextual limits and incur high computational costs. To remedy this problem, before feeding the true query sentence, we retrieve the top-k demo samples by the cosine similarity. Empirically, we exploit the BERT embedding for similarity calculation, which performs generally better than ChatGPT embedding. Finally, we instruct ChatGPT to output the NER tags for the entire unsupervised training set. An example prompt is shown in Figure 7.

Prompted Input: You are a helpful assistant for the task of named entity recognition in a sentense. You reply with brief, to-the-point answers with no elaboration as truthfully as possible. There are four known entity types: 'PER' which refers to person, 'LOC' which refers to location, 'ORG' which refers to organization , 'MISC' which refers to other specific miscellaneous entities (e.g., specific events, nationalities, products, or works of art..). When the name of the country is encountered, it can be recognized as a LOC (location) entity.

User: Given a sentence: 'Marcos Ondruska (South Africa) vs. Felix Mantilla (Spain)'. Which entities may be included in the given sentence, please answer in a single line with the form (entity1 content, entity1 type);(entity2 content, entity2 type);(entity3 content, entity3 type);... . Please only output the token that you are very sure to be entity. If there is no entity in the given sentence, reply '(no entity)'.
Assistant: (Marcos Ondruska, PER);(South Africa, LOC);(Felix Mantilla, PER);(Spain, LOC)
...

Demonstration

User: Given a sentence:"[Query Sentence]".Which entities may be included in the given sentence, please answer in a single line with the form (entity1 content, entity1 type);(entity2 content, entity2 type);(entity3 content, entity3 type);... .Please only output the token that you are very sure to be entity. If there is no entity in the given sentence, reply '(no entity)'.

Query

LLMs: (entity1 content, entity1 type);(entity2 content, entity2 type);(entity3 content, entity3 type);...

Answer to Query

Figure 7: An example prompt for few-shot in-context tagging for a *query sentence*. The demonstrations are automatically generated and selected.

C.1.2 Modification of DesERT for Multi-Source Distant Labels

One may directly use the ChatGPT labels to train the DSNER models. However, the original knowledge base-driven distant supervision is also free lunch for DSNER. and can be further incorporated. Notably, such a hybrid distant label from multiple sources problem has never been touched in the NER community. Fortunately, our DesERT algorithm can well address such hybrid labels with only a few modifications. Assume we are given knowledge base-driven labels y_{kb} (KB Labels) and ChatGPT Labels y_{cg} for a token x, where we slightly abuse the subscript to distinguish these two labels. To warm up the model, we calculate a mean soft label by,

$$\hat{y}^{mean} = (\mathrm{OneHot}(y^{kb}) + \mathrm{OneHot}(y^{cg}))/2$$

Therefore, the model fits equal confidence on these two labels when a token receives two disagreed labels. After that, we develop a modified selection protocol. Take the binary head as an example, we receive a set of transformed distant labels $\tilde{Y}^b = \{\tilde{y}^b_{kb}, \tilde{y}^b_{cg}\}$ for each token and then perform the token selection by,

$$\mathcal{D}^l_b = \{(x, \tilde{Y}^b) | \mathbb{I}(\hat{y}^b \in \tilde{Y}^b) \wedge (\max(p^b, 1 - p^b) > \tau)\}$$

In other words, we regard either distant label as a candidate of the ground truth. Once there is one label in this label set that exhibits high confidence, we regard it as a clean token. Finally, we run the DesERT algorithm without any further modification. Notably, these can be easily extended to more sources of distant supervision, e.g., when there is more than one LLM available.

C.1.3 Experimental Results with ChatGPT Supervision

Our experiments are conducted on the CoNLL03 dataset. In specific, we generate a total of 100 automatically generated samples with the help of 10 demonstration training samples. When tagging training data, we select the 10 most similar generated samples for ICL. In Figure 8, we plot the confusion matrix of the ChatGPT labels. It can be observed that ChatGPT supervision demonstrates similar trends to KB labels and is still biased. Nevertheless, ChatGPT labels have three main characteristics: (1) it classifies far more non-entity tokens as an entity; (2) it produces more balanced clean tokens; (3) the confusion patterns on fine-grained entity types are different than KB labels.

In Table 3, we report the results of DesERT when it faces different sources of distant supervision signals. In particular, we compare three types of baselines: (1) **ChatGPT**: we employ the ChatGPT model to produce zero-shot tagging on testing data; (2) **ChatGPT-A**: we employ ChatGPT to generate

Table 3: Performance of DesERT with different sources of distant labels.

Supervision	**Unsupervised**		**ChatGPT Labels**		**KB Labels**		**Hybrid Labels**	
Model	ChatGPT	ChatGPT-A	SCDL	DesERT	SCDL	DesERT	SCDL*	DesERT*
Precision	68.95	79.11	68.39	81.91	**87.96**	86.23	83.87	87.24
Recall	64.16	63.13	72.74	77.38	79.82	87.28	85.50	**88.93**
F1	66.47	70.22	70.50	79.58	83.69	86.75	84.67	**88.08**

a set of text-tag pairs and use it for few-shot ICL on testing data; (3) **SCDL**: the most competitive baseline in our main Table. From Table 3, we have the following observation:

- ChatGPT-A is much better than vanilla ChatGPT, verifying the superiority of our automatic demonstration process. But, ChatGPT and ChatGPT-A underperform other DSNER algorithms on the testing set. Though ChatGPT is a wonderful general-purpose LLM, we may draw the same conclusion as [5] that fine-tuned SLMs still play an important role in NLP.
- Given ChatGPT labels, both SCDL and DesERT underperform their counterparts when supervised by KB labels. We postulate that current DSNER algorithms are particularly designed for KB-based supervision and thus can not fully handle such new sources of labels.
- Our proposed DesERT algorithm consistently outperforms baselines on different supervised signals. In particular, when trained with hybrid labels, the modified DesERT (DesERT*) improves the KB label-trained counterpart by $+\mathbf{1.32}$ F1 score. It suggests that distant supervision from LLMs does indeed brings helpful information for the DSNER task.

In summary, our work makes the first attempt to employ LLMs to generate distant supervision. Moreover, our DesERT algorithm can be easily extended to learn from multi-source distant labels and demonstrates improved performance.

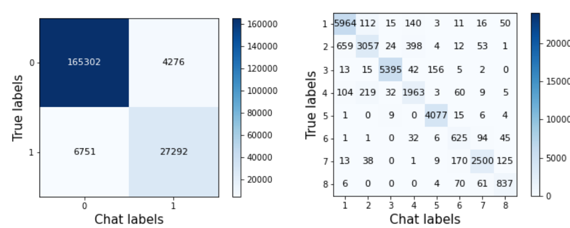

Figure 8: **Left**: Confusion matrix of true labels and ChatGPT labels on CoNLL03. **Right**: The confusion matrix displays noise among true entity-type labels in ChatGPT labels.

C.2 More Experimental Details

In this section, we provide more experimental details for a better understanding of our training process and also to ensure thee reproducibility.

C.2.1 Computation Resources

All experiments are conducted on a workstation with 8 NVIDIA RTX A6000 GPUs. It takes about $\{8, 72, 0.5, 1, 2\}$ hours for training on five benchmarking datasets (ordered as in Table 1) with one single GPU. We adopt the Huggingface Transformer library for the RoBERTa-base (125M parameters) and DistilRoBERTa-base (66M parameters) models: https://huggingface.co/transformers/. We run all the experiments three times and report the mean results.

C.2.2 Datasets Details

The data statistics of five benchmarking NER datasets are shown in Table 4.

C.3 More Implementation Details

Tagging scheme for NER As for tagging scheme, we follow the classic BIO format. To be specific, the first token of an entity mentioned with type `X` is tagged as `B-X` while the remaining tokens inside that entity are labeled as `I-X`, and the non-entity tokens are annotated as `O`. Such a scheme is more difficult than a simple IO format, especially for distant supervision.

Table 4: The statistics of five datasets, shows the number of entity types and the number of sentences in the Train/Dev/Test set.

Dataset	Types	Train	Dev	Test
CoNLL03	4	14,041	3,250	3,453
OntoNotes5.0	18	115,812	15,680	12,217
Webpage	4	385	99	135
Wikigold	4	1,142	280	274
Twitter	10	2,393	999	3,844

Clean token selection for the binary pathway In general, the precision of entity labels is relatively high, e.g., about $97.96\%(23,649/24,141)$ in the CoNLL03 dataset. That indicates us most distantly-labeled entity tokens are real entities if omit fine-grained entity types. Therefore, when performing clean token selection on the binary pathway, it only selects non-entity tokens by the matched and high-confidence strategy while including all tokens labeled as entities.

Soft label for the entity pathway When training on the selected clean token set, the binary pathway regards distant labels as true labels. However, we discard the distant hard labels but adopt the teacher model's output logits to derive soft labels [6] for the entity pathway, given by:

$$\hat{y}^s_{i,j} = \frac{\boldsymbol{p}^2_{i,j} / \sum_i \boldsymbol{p}_{i,j}}{\sum_{j'} (\boldsymbol{p}^2_{i,j} / \sum_i \boldsymbol{p}_{i,j'})}$$

where $\boldsymbol{p}_{i,j} = \text{softmax}(f_{i,j}(\boldsymbol{x};\theta_t))$, is the probability of i-th token belonging to class j in sentence $\boldsymbol{x}$, then calculate a Kullback-Leibler divergence loss. Because soft labels usually preserve sufficient information and encourage a more balanced assignment of target labels.

The implementation of teacher-student newtork When splitting the clean token set and the noisy token set, we let the teacher model select respective clean tokens to train the student model. Specifically, the teacher model's double-head pathway filters reliable clean tokens independently following the previous criterion. Then the double-head of the student model is trained with selected clean tokens and corresponding labels. The teacher model parameters are periodically updated by the student model with EMA, given by:

$$\theta_t = \alpha\theta_t + (1-\alpha)\theta_s$$

where α is a positive constant and is empirically fixed as 0.99 for Webpage/Wikigold and 0.995 for the remaining datasets. Finally, to train the entity pathway, we adopt a KL divergence loss on the student model's output logits and corresponding soft labels from the teacher model's prediction. The formulation is:

$$\mathcal{L}_{e\_cls}(\hat{\boldsymbol{y}}^s, f(\boldsymbol{x};\theta_s)) = \sum_i \sum_j -\hat{y}^s_{ij} \log f_{ij}(\boldsymbol{x};\theta_s) + f_{ij}(\boldsymbol{x};\theta_s)\log(f_{ij}(\boldsymbol{x};\theta_s))$$

Then, the student model's entity pathway is trained to approximate the soft labels. While for the binary pathway, we calculate a standard binary cross-entropy loss, which is given by:

$$\mathcal{L}_{b\_cls}(\tilde{y}^b, \hat{y}^b) = -\tilde{y}^b \log \hat{y}^b - (1-\tilde{y}^b)\log(1-\hat{y}^b)$$

where $\tilde{y}^b$ is the given distant label and $\hat{y}^b$ is generated by student model's binary pathway.

D Pseudo-Code of DesERT

We describe the overall training pipeline of DesERT in Algorithm 1.

E Limitations

While DesERT has been proven to be effective for distant supervision, it is still subject to certain limitations. First, in our debiased self-training procedure, our WCE loss is estimated from pseudo-labels instead of the real ones. While we empirically find our WCE loss works well, its performance

Algorithm 1 Training workflow of DesERT

Input: Training data $\mathcal{D} = \{(\boldsymbol{x}_i, \tilde{\boldsymbol{y}}_i)\}_{i=1}^{M}$ with distant labels; two sets of teacher-student networks, θ_{t1}, θ_{s1} and θ_{t2}, θ_{s2};

$t \leftarrow 0$

/ Selection and self-training */*

while $t < T_1$ **do**

  Get a batch $\mathcal{B} = \{(\boldsymbol{x}_i, \tilde{\boldsymbol{y}}_i)\}_{i=1}^{B} \subset \mathcal{D}$;

  if $t < k$ **then**

    $\mathcal{B}_b \cup \mathcal{B}_e \leftarrow \mathcal{B}$; //decoupled datasets

    $\overline{\mathcal{B}}_b^1 = \{\mathcal{B}_b^{l,1}, \mathcal{B}_b^{u,1}\} \leftarrow \text{Sel}(f(\theta_{t1}), \mathcal{B}_b)$;

    $\overline{\mathcal{B}}_e^1 = \{\mathcal{B}_e^{l,1}, \mathcal{B}_e^{u,1}\} \leftarrow \text{Sel}(f(\theta_{t1}), \mathcal{B}_b)$;

    $\overline{\mathcal{B}}_b^2 = \{\mathcal{B}_b^{l,2}, \mathcal{B}_b^{u,2}\} \leftarrow \text{Sel}(f(\theta_{t2}), \mathcal{B}_b)$;

    $\overline{\mathcal{B}}_e^2 = \{\mathcal{B}_e^{l,2}, \mathcal{B}_e^{u,2}\} \leftarrow \text{Sel}(f(\theta_{t2}), \mathcal{B}_b)$;

    Update θ_{s1} with $\{\overline{\mathcal{B}}_b^1, \overline{\mathcal{B}}_e^1\}$ by minimizing $\mathcal{L}$;

    Update θ_{s2} with $\{\overline{\mathcal{B}}_b^2, \overline{\mathcal{B}}_e^2\}$ by minimizing $\mathcal{L}$;

  else

    $\mathcal{X}_B \leftarrow \{(\boldsymbol{x}_i)\}_{i=1}^{B}$

    $\overline{\mathcal{B}}_b \cup \overline{\mathcal{B}}_e \leftarrow \text{Guess}(f(\theta_{t1}), f(\theta_{t2}), \mathcal{X}_B)$;

    Update θ_{s1}, θ_{s2} with $\{\overline{\mathcal{B}}_b, \overline{\mathcal{B}}_e\}$ by minimizing $\mathcal{L}$

  end if

  $\theta_{t1} \leftarrow \text{EMA}(\alpha, \theta_{t1})$, $\theta_{t2} \leftarrow \text{EMA}(\alpha, \theta_{t2})$

  $t \leftarrow t + 1$

end while

/ Post-hoc entity pathway finetuning */*

while $t < T_1 + T_2$ **do**

  Get a batch $\mathcal{B} = \{(\boldsymbol{x}_i, \tilde{\boldsymbol{y}}_i)\}_{i=1}^{B} \subset \mathcal{D}$;

  $\mathcal{X}_B \leftarrow \{(\boldsymbol{x}_i)\}_{i=1}^{B}$, $\tilde{\mathcal{Y}}_B \leftarrow \{(\tilde{\boldsymbol{y}}_i)\}_{i=1}^{B}$

  $\overline{\mathcal{B}}_e^1 \leftarrow \text{EntitySel}(f(\theta_{t1}), \mathcal{X}_B, \tilde{\mathcal{Y}}_B)$;

  $\overline{\mathcal{B}}_e^2 \leftarrow \text{EntitySel}(f(\theta_{t2}), \mathcal{X}_B, \tilde{\mathcal{Y}}_B)$;

  Finetuning θ_{s1} with $\overline{\mathcal{B}}_e^1$

  Finetuning θ_{s2} with $\overline{\mathcal{B}}_e^2$

  $\theta_{t1} \leftarrow \text{EMA}(\alpha, \theta_{t1})$, $\theta_{t2} \leftarrow \text{EMA}(\alpha, \theta_{t2})$

  $t \leftarrow t + 1$

end while

is theoretically restricted. One potential solution is to estimate a small validation set to remedy this problem, but we leave it as our future work. Second, while the imbalance between entity labels is located in Figure 1, our framework does not particularly integrate special components to explicitly overcome this problem. We believe it is not hard to draw inspiration from the recent achievement in long-tailed learning to further improve the NER performance. Lastly, since DesERT ensembles two sets of teacher-student networks as previous works did [7, 8], we should train peer-student models simultaneously and utilize the predictions from dual-teacher models iteratively, thus resulting in relatively higher training costs. We hope future efforts are made in alleviating the cost of network ensembling.

F Ethics Statement

While distant supervision is deemed a cheap way to collect and curated training data, the off-the-shelf and external knowledge bases steering the autonomous annotation procedure may include bias and unfairness. Indeed, if one trains the model by these biased labels, it may unpleasantly yield unfair and biased predictions on the basis of characteristics like race, gender, disabilities, LGBTQ, or political orientation. Therefore, when deploying our DesERT framework, it is recommended to equip it with some auxiliary tools for labeling censorship so as to improve overall fairness and ethical standard. Grounded on this, we would suggest regarding our DesERT framework as an auxiliary weakly-supervised annotation tool for assisting human annotations.

References

[1] Baixu Chen, Junguang Jiang, Ximei Wang, Pengfei Wan, Jianmin Wang, and Mingsheng Long. Debiased self-training for semi-supervised learning. In *Advances in Neural Information Processing Systems*, 2022.

[2] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[3] Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. Emergent abilities of large language models. *Trans. Mach. Learn. Res.*, 2022, 2022.

[4] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *NeurIPS*, 2022.

[5] Yejin Bang, Samuel Cahyawijaya, Nayeon Lee, Wenliang Dai, Dan Su, Bryan Wilie, Holy Lovenia, Ziwei Ji, Tiezheng Yu, Willy Chung, Quyet V. Do, Yan Xu, and Pascale Fung. A multitask, multilingual, multimodal evaluation of chatgpt on reasoning, hallucination, and interactivity. *CoRR*, abs/2302.04023, 2023.

[6] Junyuan Xie, Ross B. Girshick, and Ali Farhadi. Unsupervised deep embedding for clustering analysis. In *ICML*, volume 48 of *JMLR Workshop and Conference Proceedings*, pages 478–487. JMLR.org, 2016.

[7] Xinghua Zhang, Bowen Yu, Tingwen Liu, Zhenyu Zhang, Jiawei Sheng, Xue Mengge, and Hongbo Xu. Improving distantly-supervised named entity recognition with self-collaborative denoising learning. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 1518–1529, Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.

[8] Junnan Li, Richard Socher, and Steven C. H. Hoi. Dividemix: Learning with noisy labels as semi-supervised learning. In *ICLR*. OpenReview.net, 2020.