2024 27th International Conference on Computer Supported Cooperative Work in Design (CSCWD) | 979-8-3503-4918-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/CSCWD61410.2024.10579983

Promoting Named Entity Recognition with External Discriminator

[1]Cheng Li*,[1]Jiaqi Wu*,[2]Feng Yuan,[1]Yuqing Sun†
[1]School of Software, Shandong University, Jinan, China
[2]Shandong Shanda Oumasoft Co., Ltd., Jinan, China
li609172827@gmail.com, oofelvis@163.com, sun_yuqing@sdu.edu.cn, sdyuanf@sina.com

Abstract—In this paper, we propose an NER promotion method formed as an external discriminator. It learns the patterns about the contextual entity usages from the extensive web data and thus it can check whether the recognized entity by an NER model is correct. Different with the current popular methods on introducing the entity knowledge by gazetteers or labeled data, it can be used as the additional part to work with any NER method for promoting its performance. We adopt three widely adopted datasets for the empirical studies and the results show that our method significantly improves the NER performance. Besides, by using only a small proportion of labeled data, our method achieves a comparable performance against other models using the whole labeled data.

Index Terms—named entity recognition, usage pattern, interdisciplinary collaborations

I. Introduction

The Named Entity Recognition (NER) task aims to predicate the span and category of entity in a text. It can enhance the system's ability to process and understand text during human centered collaborative computing, thereby improving the efficiency and quality of collaborative work. Most NER methods are the supervised models and perform well when there are sufficient labeled data. But in many practical scenarios, there are few labeled data on some types. The performance of these NER models degraded significantly for the few-shot scenarios.

To solve this problem, the NER models are pre-trained with the entity-related knowledge, such as the sentence-level annotation [1], [2], the anchors in Wikipedia [3], [4], gazetteers [5], [6] etc. With the help of external information, the performance of NER models can improve significantly. However, the knowledge learned by a specific model can't be shared across NER models and can't be directly reused in a new model. Another approach is to explore rules for generating pseudo-labeled NER data [7]. Since these rules need to be elaborately designed by specialists for different types of entities, the process is time-consuming and is not easily applied to a new scenario. The teacher network method aims to label new data for retraining the NER model. This method is designed in the form of a discriminator to judge the correctness of the outputs of an NER model [8]. However, the discriminator is trained on the same labeled data as the NER model, making it challenging to apply to new patterns directly. Besides, the methods based on large language models often rely on prompt engineering and face difficulties in handling the gap between cross-domain generation tasks and sequence labeling tasks [9].

To solve the above shortcomings, we propose the discriminator-based NER enhancement model where the discriminator is designed to learn the patterns about the contextual entity usages of different types from the extensive external data. So, it can infer whether the marked entity in a text follows the usual way. Secondly, to take into account the semantics about different categories of entity, we adopt the description about entity as the query to the discriminator. It includes the specification text about the semantics of entity concept, as well as the positions and the name for marking the entity in the context, as shown in Fig.1(a). We use web data Wikipedia to train the discriminator. Then, it obtains more pseudo-labeled data from massive unlabeled data. The target NER model is then fine-tuned with these data, as shown in Figure 1(b).

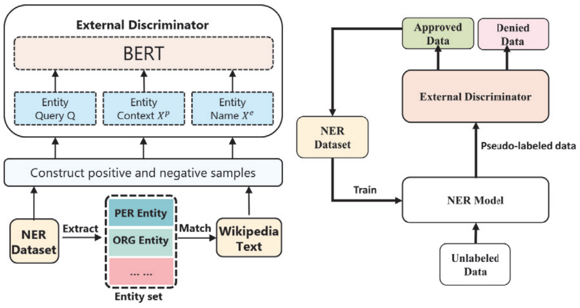
We conduct experiments on three publicly available datasets and the results show that the proposed method enhances the performance of NER model in the low-resource scenarios. Even for the NER model that has been trained with sufficient labeled data, the proposed model helps the performance improvement.

In general, our contributions are as follows: 1)We propose a method to address NER issues by learning contextual entity usages and semantics from a large amount of web data. 2)We propose the discriminator-based NER enhancement model that can be used as an additional part to promote the performance of any trained or less-trained NER model. 3)We proved that our model can effectively improve the performance of any NER model by the experimental results.

II. Related Work

There are two forms of the outputs of NER methods: sequence-based tagging and span-based tagging. The sequence-based methods treat NER as a sequence labeling task, which often consists of a neural network and a

*Cheng Li and Jiaqi Wu contributed equally to this work.
†Yuqing Sun is the corresponding author.



(a) Stage1: Training Discriminator

(b) Stage2: Prompting NER

Fig. 1: The promotion framework for NER.

conditional random field (CRF) layer [10]. The span-based approaches extract the entity spans in a text by predicting the start and end positions [11], which are suitable for solving the nested entity problem. Most of them are the supervision methods and achieve good performances with sufficient data. But they suffer from the low-resource scenarios.

Recently, the pre-trained language models are introduced as the base neural network, such as BERT [12]. Benefiting from the general linguistic knowledge prelearned from massive data, these models are fine-tuned on the NER task and achieve a lot of progress on performance, which maintains the SOTA results. Another way to use lexical information or gazetteers as the affiliated information to help the NER models on improving the performance [13].

To solve the low-resource challenge on labeled data, the transfer learning methods introduce the entity related linguistic knowledge learned by the NER models on the cross-domain or cross-linguistic data [14], [15]. However, the learned knowledge can not be directly reused in a new NER model. Self-training has been used successfully in many fields, such as text classification [16], named entity recognition [7], [8], etc. It utilizes the trained models to obtain the pseudo-labeled data with high confidence from unlabeled data. But the noisy data are often contained in the pseudo-labeled data. How to reduce the noises remains the important concern.

III. The NER Enhancement Method

The Named Entity Recognition (NER for short) task extracts the entities $E = \{X_1^e, ..., X_l^e\}$ from a sentence $X = \{x_1, ..., x_n\}$, where X_i^e denotes an entity and x_i denotes i-th token in X. For a sequence-based NER model, the output is the category probability on each token, i.e. $P^{seq} \in \mathbb{R}^{n \times c}$, where c denotes the number of entity types. For a span-based NER model, the output is in the form of $P^{span} \in \mathbb{R}^{n \times 2}$, denoting the probability of the start or end position of an entity span. In our method, these recognized entities in the outputs are converted to the input form for the discriminator, namely the position and type of each entity.

A. The Promotion Framework for NER

The main idea of the promotion framework is to introduce a discriminator for learning the patterns about contextual entity usages in natural language using the extensive web data, which is independent of any specific NER model. When pseudo-labeled instances are obtained from a less trained NER model, the discriminator selects less noisy data for retraining. This framework, incorporating external linguistic knowledge as a neural network, helps an NER model more robust compared to traditional self-training methods.

The proposed framework consists of two stages, as shown in Fig. 1. In stage I we train the discriminator with the external data and the NER data. In stage II we use the discriminator to verify the output of an NER model and find less noisy pseudo-labeled instances for promoting.

B. The Discriminator

We take three considerations to design the discriminator. One is to introduce the web knowledge such that we can benefit not only from the extensive natural data for training the discriminator but also from the new patterns about NER. The second is to be independent of any specific NER model, which requires the discriminator to support the flexible outputs of NER model. The purpose of our discriminator is to learn this kind of natural language patterns about the entities for different types since they contain rich semantics compared with a set of labeled instances. The third is to introduce the semantics of a NER type. We adopt the query as one of the inputs for the discriminator, a short text specifying the semantics of a specific entity type. This enables the discriminator to understand the semantics of a new type of entity and quickly catch its usage.

The structure of the discriminator is shown in Fig. 1(a). The input contains three parts, the entity query Q, the entity context X^p and the entity name X^e, which are organized as the form of $\{[CLS], Q, [SEP], X^p, [SEP], X^e\}$. The query Q is a short text that specifies the concept that the queried entity belongs to. Both X^p and X^e are selected from an instance, where X^e is the name of the recognized entity and X^p is the sentence with the $< type >$ mark instead of X^e. Here, X^p shows an instance about the contextual pattern for this entity type. Through massive data training, the discriminator can learn the normal patterns about an entity type, the names and the contexts.

We adopt the light-weight neural network for the discriminator, where the pretrained BERT [12] is used as the backbone.

$$\mathcal{L} = BCE(\sigma(\mathrm{WH})) \tag{1}$$

C. Training Discriminator

The training data include two parts: the NER training dataset G^N and its extension on Wikipedia denoted by G^U. As shown in Fig. 1(a), we extract all entities in G^N

Algorithm 1 Discriminator-based Promoting

```
Require: NER training data $G^N$; unlabeled data $G^U$; pre-trained discriminator $F^D$;
Ensure: Enhanced NER model $F^N$;
for $epoch = 1$ to $n$ do
    Train NER model $F^N$ with $G^N$;
    for $x^U$ in $G^U$ do
        $\mathrm{p}^N = F^N(x^U)$; $y^U = argmax(\mathrm{p}^N)$;
        Construct discriminator input $x^D$ based on $(x^U, y^U)$;
        $p^D = F^D(x^D)$;
        Set confidence threshold $\delta$;
        if $min(\mathrm{p}^N) > \delta^N$ and $p^D > \delta^D$ then
            add $(x^U, y^U)$ to $G$
        end if
    end for
end for
```

and then use them to label the Wikipedia text[1] with the matching anchors for constructing the dataset G^U. In this process, the confusion cases are deleted, namely the same name with different entity types. Finally, the instances in G^N and G^U are reconstructed in the form of the discriminator input, as discussed in section III-B.

We also construct the negative samples by two ways: (1)Boundary error: the entity boundary is shifted, either by expanding the boundary or shrinking it. The corresponding X^p and X^e are changed. (2)Type error: the entity type is replaced with another entity type. The corresponding X^p and Q are changed. The weight for negative samples is set lower than for positive. We use the NER validation dataset to measure the performance of the discriminator in terms of the precision.

D. Promoting NER

As shown in Fig. 1(b), after pre-training the discriminator, we use a trained NER model F^N to label the unlabeled data. Then the discriminator is used to filter the less noisy pseudo-labeled instances. In this process, the output of the NER model is converted to the input format for the discriminator. If a sentence contains multiple entities, we generate multiple instances for the discriminator. Only all entities in a sentence are justified correctly by the discriminator, it is added to G^N.

The promoting process is shown in Alg.1, where p^N is the probability output vector of a NER model, and p^D is the probability by the discriminator. After each round of retraining F^N, we use two thresholds δ^N and δ^D as the confidence scores for filtering the pseudo-labeled instances. The function $min(\mathrm{p}^N)$ computes the lowest vague. We would also compare $min(\mathrm{p}^N)$ with $mean(\mathrm{p}^N)$ in the experiments.

IV. Experiments

A. Experimental Settings

We choose two representative scenarios to evaluate the effectiveness of our method by quantifying how much it promotes the NER performance. (1) low-resource on label data: we only use a proportion of the labeled NER instances as the training set. (2) sufficient label data: we use all the NER training data and 1 million Wikipedia unlabeled data for enhancement.

Our hyper-parameter settings are shown in table I. We choose Adam as the optimizer. In pre-training the discriminator, it is initialized with the publicly available pre-trained Bert-base model. After each round of obtaining new data, we train 50 epochs for the NER model. The early stopping rounds for self-training are set to 3. All experimental results are averaged on three tests. The threshold values δ^N and δ^D in the experiment are set to 0.99.

TABLE I: Hyper-parameter Settings.

Hyper-parameter	Test Values	Best
Batch size	8, 16, 32, 64	16
Learning rate	5e-5, 1e-5, 5e-6, 1e-6	5e-6
Dropout	0.1, 0.2, 0.3, 0.4, 0.5	0.1
Threshold	0.5, 0.7, 0.9, 0.99	0.99

B. Datasets and Metrics

Experiments are conducted on three widely used NER datasets: MSRA [17], Chinese OntoNotes 4.0 [18], [19] and Conll03 [20]. We use 20210301 Chinese Wikipedia dumps[2] to construct the dataset as unlabeled data. We remove the sentences with lengths less than 30 and greater than 200. We evaluate the performance of NER models with entity-level micro-averaged precision, recall and F1 score.

C. Comparison Methods

The comparison methods include two categories according to whether they use additional data.

The methods without additional data include: for English NER, the traditional baseline LSTM-CNN-CRF [10], the BERT based token-level sequence tagging model BERT-Tagger [12], and the SOTA machine reading comprehension based model BERT-MRC [11]. for Chinese NER, the lexical information incorporated Lattice [13] and the glyph information based Glyce-BERT [21].

We selected the following baseline to enhance BERT-Tagger using additional data: a pre-trained model called CoFEE-BERT [3] for coarse-grained entity recognition, which utilizes extensive anchor texts from Wikipedia, along with self-training [22].

Besides, we design the variant of our model self-training with threshold, which is a selection strategy on pseudo-labeled data by self-training, which removes the instances with low probabilities. Since a sentence contains more than one word, there are two choices to use the threshold: the minimum strategy and the mean strategy. By empirical examination, shown in Table II, the minimum strategy is better than the mean strategy and is chosen for all experiments in this paper.

[1] https://dumps.wikimedia.org

[2] https://dumps.wikimedia.org/zhwiki/20210301/zhwiki-20210301-pages-articles-multistream.xml.bz2

TABLE II: Comparison of Different Filtering Strategies.

Methods	P	R	F1
Minimum strategy $min(\mathrm{p}^N)$	89.26	89.97	89.61
Mean strategy $mean(\mathrm{p}^N)$	88.74	89.62	89.17

D. Results for the Low-resource Scenario

We first evaluate the effectiveness of the proposed promotion method in the low-resource scenario. For each dataset, we use 10%, 25% and 50% training data, respectively, and remove the labels from the remaining data as the unlabeled data. The results are shown in Table III. Compared with other enhancement methods, our method achieves the best performance of BERT-Tagger on different sizes of training data on all datasets.

Specially, on MSRA and OntoNotes, for an NER model trained by a part of labeled data, the performance enhanced by our method outperforms itself with 100% labeled data for training. This indicates that there are some low-quality instances in the labeled data, which are filtered out by our discriminator.

For Conll03, the NER model after promoted by our method is not superior than that with 100% labeled data. So we check the labeled data and find that most sentences on Conll03 are shorter than the other two datasets. The average sentence length for Conll03 is 13.6, while for MSRA and OntoNotes 4.0 are 46.8 and 31.3, respectively. Meanwhile, there are a large number of instances in Conll03 without context. Therefore, the linguistics knowledge about entity usage that the discriminator has learned is not applicable for these instances.

E. Results on Sufficient Data Scenario

Then we verify whether and how much our method promotes the performance of a trained NER model. The unlabeled data used in this part are from Wikipedia. We choose 1 million sentences with the length up to 250 in Wikipedia as the unlabeled data in our method. The CoFFEE-BERT method uses the same settings as the original paper.

As the result shown in Table IV, our method enhances the BERT-Tagger model on three datasets. Specially, on MSRA and OntoNote 4.0, BERT-Tagger promoted by our method outperforms the SOTA model BERT-MRC. Compared to CoFEE-BERT, we use less additional unlabeled data to achieve better results. On Conll03, our method still improves the performance, while the performances degrade by the other two self-training methods since our method has learned the stable pattern about the entities. The reason that BERT-MRC achieves the top performance should thank to its BERT-large architecture since the other methods use BERT-base.

F. Ablation Experiments on Discriminator

We performed ablation experiments on the core elements of the discriminator, namely the input-form, network structure and data resources. We adopt the sentence-level precision on the NER validation dataset to measure the effectiveness of discriminator. It measures the ratio of the correct instances to the amount of pseudo-labeled data accepted by the discriminator. The higher, the better.

As shown in Table VI, every part contributes positively to the performance. The entity query is particularly important for the discriminator since it connects the semantics in concept definition and the practical usages in context. The entity name is also useful for the discriminator to learn the entity pattern, which helps distinguish the type of different entities in the same contextual patterns. Besides, the external data resources also introduce more contexts about entity usages and help the discriminator learn the patterns.

V. Model Analysis

A. The Cross-domain NER Capability

We tested the cross-domain capability of the discriminator on datasets MSRA and OntoNotes 4.0, where only two entity types are the same. The discriminator is directly cross-domain reused without any retraining on new types. The experimental results in Table V show that the discriminator still works well for the cross-domain applications and is much better than the self-training method. This illustrates that the discriminator has learnt the linguistics knowledge about entity usage patterns, which are transferable.

B. Understanding Model Enhancements

To understand how our method improves a lot on NER models, we analyze the instances justified by the discriminator, shown in Table VII. With an increasing threshold for the discriminator, the precision increases and thus the error samples in the new labeled data are removed, namely a higher $TN/(TN + FP)$ score. On the setting threshold=0.99, the discriminator removes 53.8%, 77.9% and 64.9% wrong samples on Conll03, OntoNotes 4.0 and MSRA, respectively. The ratio is much higher on OntoNotes 4.0 than the other datasets, which leads to a larger enhancement during the self-training process. Thus the threshold can be set to a higher score to ensure the quality of the newly labeled data.

C. Impact of Unlabeled Data Resources

We compare the effects by different data resources for self-training NER, where 10% labeled data in each set are used for training the NER model and the remaining data after removing labels are for the unlabeled data. Meanwhile, we select an equal amount of Wikipedia unlabeled corpus. During each iteration of self-training, we add at most 1000 pseudo-labeled data.

As the results shown in Fig. 2, the use of external data leads to very good results on MSRA, which due to its similar text style to Wikipedia and thus benefits from the introduced instances on the contextual entity usages.

TABLE III: The Promotion Effectiveness on the Low-resource Scenario (F1 scores).

Methods	MSRA			OntoNotes 4.0			Conll03		
	10%	25%	50%	10%	25%	50%	10%	25%	50%
BERT-Tagger [12]	91.29	92.76	94.03	77.44	79.33	79.90	86.88	90.09	90.99
+ self-training [22]	91.99	93.31	94.26	78.68	79.66	80.45	88.57	90.25	91.11
+ self-training with threshold	92.99	94.26	94.73	78.97	80.36	80.55	89.61	90.58	91.15
+ ours	93.57	94.60	94.85	81.04	81.23	81.28	90.21	91.12	91.81

TABLE IV: The Promotion Effectiveness on the Sufficient Labeled Data Scenario.

Methods	MSRA			OntoNotes 4.0			Conll03		
	P	R	F1	P	R	F1	P	R	F1
LSTM-CNN-CRF [10]	-	-	-	-	-	-	91.35	91.06	91.21
Lattice [13]	93.57	92.79	93.18	76.35	71.56	73.88	-	-	-
Glyce-BERT [21]	95.57	95.51	95.54	81.87	81.40	80.62	-	-	-
BERT-MRC [11]	96.18	95.12	95.75	82.98	81.25	82.11	92.33	94.61	93.04
BERT-Tagger [12]	94.97	94.62	94.80	79.25	81.38	80.30	92.08	92.75	92.42
w/ CoFEE-BERT [3]	-	-	-	80.27	80.64	80.46	-	-	-
w/ self-training [22]	95.34	95.19	95.27	79.74	82.18	80.95	91.36	92.15	91.75
w/ self-training with threshold	95.33	95.26	95.30	81.86	81.47	81.66	91.73	92.51	92.11
w/ ours	95.84	95.77	95.81	84.19	82.86	83.52	92.83	93.16	93.00

TABLE V: Evaluation on the Discriminator For Cross-domain NER. The model is trained by 10% labeled data and enhanced by the remaining 90% unlabeled data.

Datasets	Methods	F1
MSRA	BERT-Tagger	91.29
	w/ self-training	91.99
	w/ MSRA Discriminator	93.57
	w/ OntoNotes Discriminator	93.20
OntoNotes 4.0	BERT-Tagger	77.44
	w/ self-training	78.68
	w/ OntoNotes Discriminator	81.04
	w/ MSRA Discriminator	81.02

TABLE VI: The Ablation Test on the Discriminator.

Datasets	Model	Precision
Conll03	Discriminator	95.79
	w/o Entity query Q	90.15
	w/o Entity name X^e	93.28
	w/o External matched data	95.13

Comparatively, on both Conll03 and OntoNotes 4.0, using unlabeled data from the same dataset for self-training NER is better than the external data.

D. Case Study

In this section, we show some examples to illustrative how the discriminator recognized the incorrect NER sentences in Table VIII. Taking the first instance as an example, there is a difference between the NER label and the discriminator justification. From the point of linguistic view, an ”Organization” entity appearing after ”founded” is more frequently used than a ”Miscellaneous” entity. That is the pattern on entity usage that our discriminator devotes to learn. Comparatively, these errors usually happen in NER models with a high probability on label.

E. Interpretability

To understand why the discriminator works well, we analyze the attention weights of the last layer of its network in a visualization way. We choose a difficult sentence as an example ”Henson and Meinel founded the L5 Society in 1975” . The organization ”L5 Society” was previously recognized as miscellaneous by an NER model, which is corrected by the discriminator.

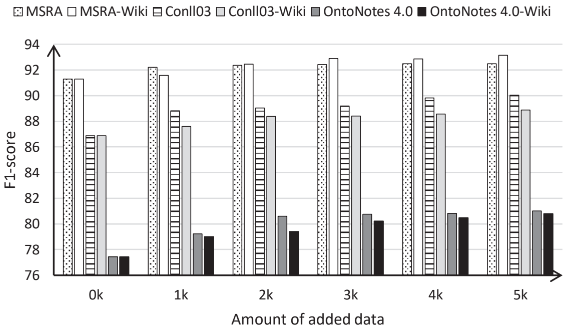
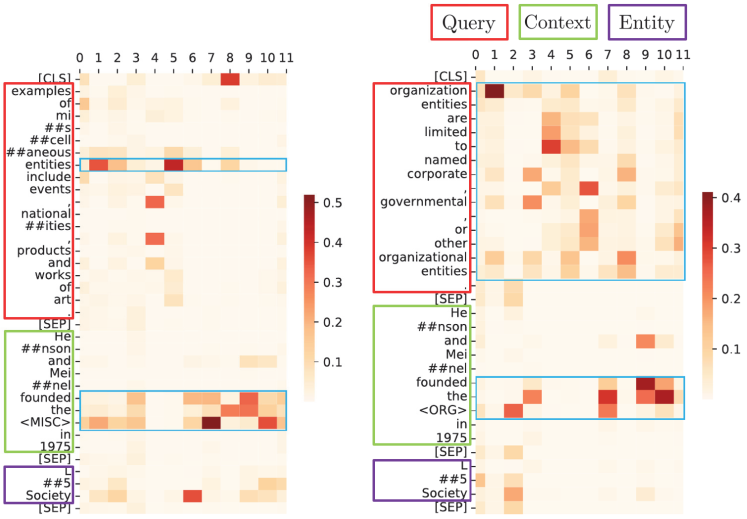


Fig. 2: Impact of unlabeled data resources.



(a) On the ”Miscellaneous” query

(b) On the ”Organization” query

Fig. 3: Visualization of the attention weights of the output tokens in the discriminator.

The heat maps in Fig. 3 (a) and (b) show the attentions of the discriminator on two queries miscellaneous and

TABLE VII: Discriminator Analysis on Three NER Validation Datasets. Invalid cases: 1. output without entity; 2. output with logical errors, e.g. E-ORG after B-PER. Metrics are measured on the valid outputs.

Dataset	Dev size	Invalid outputs	Threshold δ^D/δ^N	TP	TN	FN	FP	TP/ (TP+FN)	TN/ (TN+FP)	Precision
Conll03	3466	946	0.99	2050	105	275	90	88.17%	53.8%	95.79
			0.5	2141	75	184	120	92%	38.4%	94.69
OntoNotes 4.0	4301	1817	0.99	917	687	686	194	57.2%	77.9%	82.53
			0.5	1458	213	145	668	90.9%	24.1%	68.57
MSRA	4636	2641	0.99	1549	161	198	87	88.6%	64.9%	94.68
			0.5	1684	57	63	191	96.3%	22.9%	89.81

TABLE VIII: The Incorrect NER Instances.

Incorrect NER Instances	Errors Description
Henson and Meinel founded the L5 Society in 1975.	NER Output: ”L5 Society” – ”Miscellaneous” Ground Truth: ”L5 Society” – ”Organization”
The ground was opened in 1926, when Maccabi Tel Aviv moved into the ground from their previous stadium.	NER Output: ”Tel Aviv” – ”Organization” Ground Truth: ”Maccabi Tel Aviv” – ”Organization”

organization from the first row of Table VIII, respectively. The horizontal axis represents the attention heads of BERT, and the vertical axis represents the input tokens after WordPiece. Comparing two figures, the discriminator pays more attention to the entity query of ”Organization” than ”Miscellaneous”. Specially, the corresponding words in the query such as ”organization ”, ”founded” are highlighted, which are obviously likely to be the correct type of ”Organization” for this context. This shows that the discriminator has learned entity usage patterns, and the query effectively guides the discriminator in comprehending various types.

VI. Conclusion

In this paper, we propose an external discriminator-based promotion method for NER. Different from the current methods on introducing the external knowledge on entities by gazetteers or labeled data, we learn the usage patterns of entities in language by the external data and form the knowledge as the neural discriminator. By determining whether the marked entity in a context is correct usage, it helps the NER model find less noisy pseudo-label data for promoting the performance. We use the widely adopted datasets to verify our method. The results on both scenarios of low-resource and sufficient label show that our method significantly outperforms baselines.

VII. Acknowledgment

This work was supported by the National Natural Science Foundation of China (62376138) and the Innovative Development Joint Fund Key Projects of Shandong NSF (ZR2022LZH007).

References

[1] C. Kruengkrai, T. H. Nguyen, and S. A. M. et al., “Improving low-resource named entity recognition using joint sentence and token labeling,” in ACL, 2020, pp. 5898–5905.

[2] B. Patra and J. R. A. Moniz, “Weakly supervised attention networks for entity recognition,” in EMNLP, 2019, pp. 6267–6272.

[3] M. Xue, B. Yu, and Z. Z. et al., “Coarse-to-fine pre-training for named entity recognition,” in EMNLP, 2020, pp. 6345–6354.

[4] J. Chen, Q. Liu, and H. L. et al., “Few-shot named entity recognition with self-describing networks,” in ACL, 2022, pp. 5711–5722.

[5] S. Rijhwani, S. Zhou, and G. N. et al., “Soft gazetteers for low-resource named entity recognition,” in ACL, 2020, pp. 8118–8123.

[6] B. Fetahu, A. Fang, and O. R. et al., “Dynamic gazetteer integration in multilingual models for cross-lingual and cross-domain named entity recognition,” in NAACL, 2022, pp. 2777–2790.

[7] W. Liao and S. Veeramachaneni, “A simple semi-supervised algorithm for named entity recognition,” in NAACL, 2009, pp. 58–65.

[8] Z. Li, D. Feng, and D. L. et al., “Learning to select pseudo labels: a semi-supervised method for named entity recognition,” Frontiers Inf. Technol. Electron. Eng., pp. 903–916, 2020.

[9] W. Shuhe, S. Xiaofei, and L. X. et al., “Gpt-ner: Named entity recognition via large language models,” arXiv preprint arXiv:2304.10428, 2023.

[10] M. Xuezhe and H. Eduard, “End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF,” in ACL, 2016, pp. 1064–1074.

[11] L. Xiaoya, F. Jingrong, and M. Y. et al., “A unified MRC framework for named entity recognition,” in ACL, 2020, pp. 5849–5859.

[12] J. Devlin, M. Chang, and K. L. et al., “BERT: pre-training of deep bidirectional transformers for language understanding,” in ACL, 2019, pp. 4171–4186.

[13] Y. Zhang and J. Yang, “Chinese NER using lattice LSTM,” in ACL, 2018, pp. 1554–1564.

[14] C. Jia, X. Liang, and Y. Zhang, “Cross-domain NER using cross-domain language modeling,” in ACL, 2019, pp. 2464–2474.

[15] J. T. Zhou, H. Zhang, and D. J. et al., “Dual adversarial neural transfer for low-resource named entity recognition,” in ACL, 2019, pp. 3461–3471.

[16] S. Mukherjee and A. H. Awadallah, “Uncertainty-aware self-training for few-shot text classification,” in NeurIPS, 2020, pp. 21 199–21 212.

[17] G. Levow, “The third international chinese language processing bakeoff: Word segmentation and named entity recognition,” in COLING, 2006, pp. 108–117.

[18] R. Weischedel, M. Palmer, and M. M. et al., “Ontonotes release 4.0,” in LDC2011T03, Philadelphia, Penn.: Linguistic Data Consortium., 2011.

[19] C. Wanxiang, W. Mengqiu, and M. C. D. et al., “Named entity recognition with bilingual constraints,” in NAACL, 2013, pp. 52–62.

[20] E. F. Tjong Kim Sang and F. De Meulder, “Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition,” in NAACL, 2003, pp. 142–147.

[21] Y. Meng, W. Wu, and F. W. et al., “Glyce: Glyph-vectors for chinese character representations,” in NeurIPS, 2019, pp. 2742–2753.

[22] J. Huang, C. Li, and K. S. et al., “Few-shot named entity recognition: An empirical baseline study,” in EMNLP, 2021, pp. 10 408–10 423.