

GLARA: Graph-based Labeling Rule Augmentation for Weakly Supervised Named Entity Recognition

Xinyan Zhao* Haibo Ding† Zhe Feng†

*School of Information, University of Michigan, MI 48109, USA

† Bosch Research and Technology Center, Sunnyvale, CA 94085, USA

zhaoxy@umich.edu, {haibo.ding, zhe.feng2}@us.bosch.com

Abstract

Instead of using expensive manual annotations, researchers have proposed to train named entity recognition (NER) systems using heuristic labeling rules. However, devising labeling rules is challenging because it often requires a considerable amount of manual effort and domain expertise. To alleviate this problem, we propose GLARA, a graph-based labeling rule augmentation framework, to learn new labeling rules from unlabeled data. We first create a graph with nodes representing candidate rules extracted from unlabeled data. Then, we design a new graph neural network to augment labeling rules by exploring the semantic relations between rules. We finally apply the augmented rules on unlabeled data to generate weak labels and train a NER model using the weakly labeled data. We evaluate our method on three NER datasets and find that we can achieve an average improvement of +20% F1 score over the best baseline when given a small set of seed rules.

1 Introduction

Named entity recognition (NER) models often need to be trained with many manual labels to perform well. Due to the high cost of manual annotations, collecting labeled data to train NER models is challenging for real-world applications. Recently, researchers have proposed to collect weak labels using heuristic rules, which are called labeling rules (Bach et al., 2017; Fries et al., 2017; Ratner et al., 2020; Safranchik et al., 2020). This kind of methods typically first ask domain experts to write labeling rules for a NER task, then use these manual rules to generate labeled data and train a NER model with the weakly labeled data. The advantage of these methods is that they do not require manual annotations. However, during our study, we find that writing labeling rules is also challenging for domain-specific tasks. Devising accurate

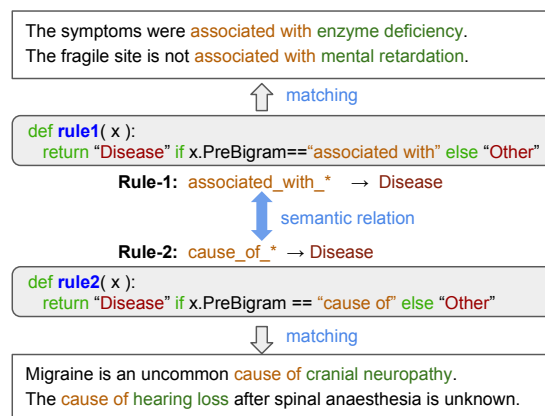


Figure 1: Two examples of rules for recognizing Diseases. PreBigram means the 2 tokens on the left of a candidate entity.

rules often demands a significant amount of manual effort because it requires developers to have deep domain expertise and a thorough understanding of the target data.

To alleviate manual effort on writing labeling rules, we propose GLARA, a Graph-based Labeling Rule Augmentation framework to automatically learn new rules from unlabeled data with a handful of seed rules. Our work is motivated by the intuition that if two rules can accurately label the same type of entities, then they are semantically related via the entities matched by them. Therefore, we can acquire new labeling rules based on their semantic relatedness with the rules that we have already known. For example, Figure 1 shows two example rules for labeling Disease entities. If we know that rule1 "associated with * -> Disease" is an accurate rule for labeling diseases, and rule1 is semantically related to rule2 "cause of * -> Disease", then we can derive that rule2 can be another accurate rule for labeling diseases.

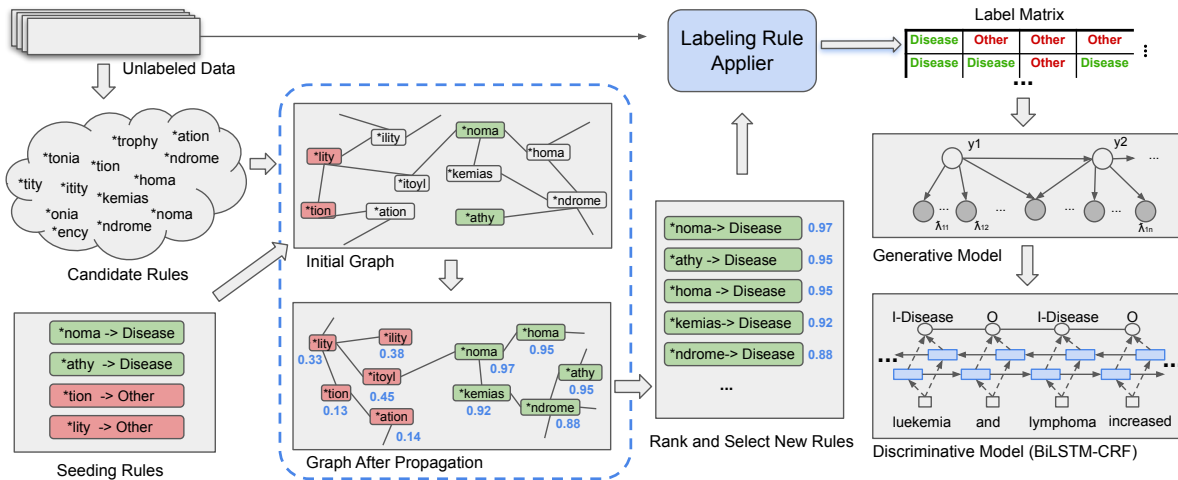


Figure 2: An example workflow of GLARA framework that using suffix rules to recognize Diseases. Given unlabeled data and seeding rules, (1) we first extract candidate rules from unlabeled data, (2) then build a graph of rules and learn new rules by propagating the labeling information from seeding rules to other rules. (3) Next we apply selected rules on unlabeled data and obtain a label matrix. (4) Finally, we estimate the noisy labels using a generative model, and (5) train a final NER model with noisy labels. “*noma \rightarrow Disease” denotes that if a word’s suffix is “noma” then it will be labeled as Disease.

To augment labeling rules, we first define six types of rules and extract all possible rules from unlabeled data as candidate rules. Then, for each rule type, we build a graph by connecting rules of this type based on their semantic similarities. In our work, we compute a rule’s embedding as the average of contextual embeddings of entities matched by the rule, and compute the similarities between rules using their embeddings. We learn new rules using a graph neural network model from a small set of seed rules. Next, we train a discriminative NER model using the weak labels generated by both the seeding and learned rules. To obtain weak training labels, we first obtain a label matrix by applying all the augmented rules on each token in unlabeled data. Then, we estimate the labels of unlabeled instances using the LinkedHMM model (Safranchik et al., 2020). We evaluate our framework on three datasets. In our experiments, we first show that our method can achieve better results than baselines when abundant rules are available. We also demonstrate that we can achieve an average improvement of +20% F1 when only a small set of rules are available.

We summarize our major contributions as:

- We propose a new Graph-based Rule Labeling Rule Augmentation (GLARA)¹ framework,

¹The code is available at <https://github.com/zhaoxy92/GLaRA>.

which can effectively learn new labeling rules from unlabeled data automatically.

- We propose a new graph neural network to estimate rules’ labeling confidence with a new class distance-based loss function.
- We define six types of labeling rules, which have been proven to be effective on three named entity recognition tasks.

2 The GLARA Framework

Our goal is to build a NER system with a small set of manually selected seeding rules and unlabeled data. Our key idea is to first augment labeling rules using graph neural networks, based on the hypothesis that semantically similar rules should have similar abilities to recognize entities. Then, we train a NER model using the weak training data labeled by the augmented rules.

Overview Figure 2 shows an example workflow of GLARA framework using suffix rules to recognize Disease entities. Our framework consists of five major components. (1) *Rule extractor*: We define six types of rules and extract all possible rules from unlabeled data as candidate rules. (2) *Rule augmentation*: For each rule type, we first build a graph of rules by connecting rules based on their semantic similarities. Given a small set of

manual seeding rules, we learn new rules by propagating the labeling confidence from seeding rules to other rules. (3) *Rule Applier*: We obtain a label matrix by applying all the augmented rules on each token in the unlabeled data. (4) *Generative Model*: We estimate the labels of unlabeled instances using a generative model. (5) *Discriminative Model*: We train a final discriminative NER model using the weak labels produced by the generative model.

2.1 Candidate Rule Extraction

As demonstrated in previous work on NER (Zhou and Su, 2002), lexical and contextual clues are strong indicators for entity recognition. Therefore, we define and extract six types of rules: SurfaceForm, Prefix, Suffix, PreNgram, PostNgram, and DependencyRule to recognize entities by considering their lexical, contextual, and syntax information.

Given an unlabeled sentence, we first extract all noun phrases (NPs) using a set of Part-of-Speech (POS) patterns, as candidate entity mentions. The POS patterns include “JJ? NN+” (JJ denotes an adjective, and NN denotes a noun) and top 15 most frequent POS patterns of the entity mentions in the development sets. Then, we extract all six types of rules from the unlabeled data as candidate rules for each candidate entity mention. Specifically, we extract the surface form of each candidate entity mention as a SurfaceForm rule. If the mention is a single token, we extract its first and last m characters as Prefix and Suffix rules, respectively. For PreNgram rule, we extract leading n words of a candidate entity as inclusive PreNgram rule; meanwhile, we also extract n words on the left of candidate as exclusive PreNgram rule. PostNgram rules are created similarly from the right context. Also, for each multi-token entity candidate, we first extract the dependency relations of the first token and the second last token, respectively, and then each dependency is combined with the last token as Dependency rules of this mention. We treat m and n as hyperparameters in our work.

Figure 3 show some example rules that extracted for the candidate entity mention “Alzheimer’s disease” and how they are used for labeling entities.

2.2 Labeling Rule Augmentation

We aim to learn new labeling rules from seed rules by exploiting the semantic relations between rules. Specifically, we first build a graph with rules as nodes. Then, we initialize the graph with both

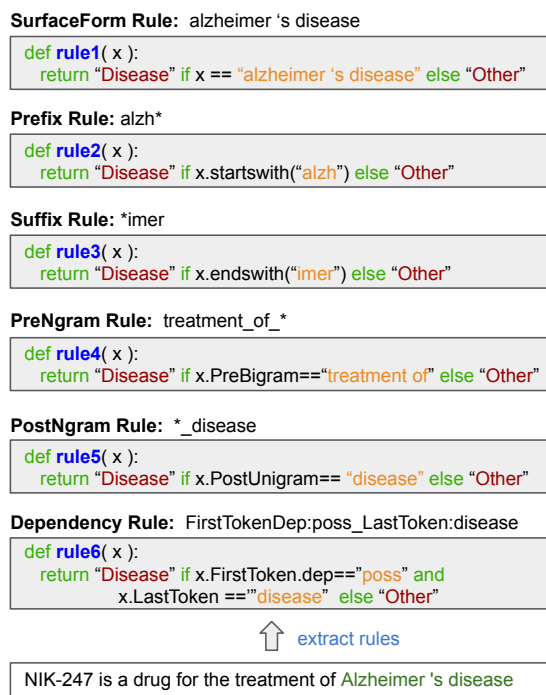


Figure 3: Example rules extracted for an entity mention, and how these rules will be used to label entities.

manually selected positive and negative seeding rules. Positive seeding rules are those that can be used to predict a target entity type. Negative ones are used to predict instances of the “Other” class. Next, we estimate the representations of rules by optimizing a graph neural network model. Finally, we compute the distances of a rule to the centroids of positive and negative seeding rules, respectively, and then select rules close to positive centroid as new labeling rules.

Graph of Rules For each type rules, we create a graph $G = (\mathcal{V}_u, \mathcal{V}_s^{pos}, \mathcal{V}_s^{neg}, \mathcal{A})$ with this type of rules as nodes, where \mathcal{V}_u denotes the candidate rules extracted in the previous step, \mathcal{V}_s^{pos} and \mathcal{V}_s^{neg} denotes the positive and negative seeding rules, respectively. \mathcal{A} is the adjacency matrix of nodes. In our graph, each node (i.e., rule) is connected with the top 10 semantically similar nodes. The similarity between two rules is computed as the cosine similarity using their embeddings.

Rule Embeddings We estimate semantic relatedness between rules with their embeddings, which are computed using pre-trained contextual embedding models. Specifically, we first apply the pre-trained ELMo (Peters et al., 2018) model on all unlabeled sentences to obtain contextual embeddings of candidate entity mentions. Then, we compute

the embedding of a rule as the average of the embeddings of all the candidate entities that can be matched by this rule. For example, the embedding of the prefix rule “*noma” is calculated as the average of the embeddings of all candidate mentions that ends with “noma”.

Graph Propagation Model Given a graph of rules and seeding rules, we formulate the problem of learning new labeling rules (i.e., positive rules) as a graph-based semi-supervised node classification task that aims to classify candidate rules (which could be treated as unlabeled nodes in the graph) as positive or negative.

Based on the intuition that semantically similar rules should predict entity labels similarly. We propose a graph neural network model to propagate labeling information from seeding nodes to other nodes based on the recent work on Graph Attention Network (Veličković et al., 2017). Specifically, given the input embedding h_i of node i and its neighbors \mathcal{N}_i , we first compute an attention weight for each connected pair (i, j) as,

$$\alpha_{ij} = \frac{\exp(f(A^T[Wh_i, Wh_j]))}{\sum_{k \in \mathcal{N}_i} \exp(f(A^T[Wh_i, Wh_k]))} \quad (1)$$

where W is a parameter matrix, and f is the LeakyReLU activation function. We then recompute the embedding of i :

$$h_i^* = \alpha_{i,i}Wh_i + \sum_{j \in \mathcal{N}_i} Wh_j \quad (2)$$

To keep model stable, we apply multi-head attention mechanism to obtain K attentional states for each node, the average of which is used as final node representation, i.e., $h_i^{*T} = \frac{1}{K} \sum_k h_i^{kT}$.

The objective of our model is defined as follows:

$$\mathcal{L}_{total} = \mathcal{L}_{sup} + \mathcal{L}_{reg} + \mathcal{L}_{dist} \quad (3)$$

where $\mathcal{L}_{sup} = -(y_i \log(p_i)) + (1 - y_i) \log(1 - p_i)$

$$\mathcal{L}_{reg} = \sum_{i,j \in \mathcal{N}_i} \|h_i - h_j\|_2$$

$$\mathcal{L}_{dist} = dist(h_{pos}, h_{neg})$$

where \mathcal{L}_{sup} is the supervision loss computed on both positive and negative seeding rule nodes, \mathcal{L}_{reg} is the regularization that encourages connected nodes to share similar representations, and \mathcal{L}_{dist} aims to maximize the distance between positive and negative seeding nodes. $dist(\cdot)$ computes the cosine similarity between the centroids of the positive and negative seeds. p_i is the probability of a node being classified as positive, and h_{pos} and

h_{neg} are the average embeddings of positive and negative nodes, respectively.

When the learning process is finished, each rule is associated with a new embedding representation, i.e., h_i^{*T} . For each rule, we first compute its cosine distances to the centroids of positive and negative seeding nodes using their embeddings, respectively. Then, we rank all rules by the difference of two distances and select the top M rules closest to the centroid of the positive seeding rules as new labeling rules.

2.3 Generative Model for Label Estimation

After the rule learning process, we apply both the newly learned rules and the seeding rules on unlabeled data to produce a matrix of labels.² Since one token can be potentially matched by several different rules, the resulting labels can have conflicts. Therefore, we use the LinkedHMM (Safranchik et al., 2020) model to combine these labels into one label for each token. Briefly, the main idea of LinkHMM is to treat the true label of a token as a latent random variable and estimate its value by relating it to the label outputs from different labeling rules. The estimated labels can be used to train final discriminative NER models.

2.4 Discriminative NER Model

After the training of the generative model (i.e., the LinkedHMM in our work) is completed, each token in the unlabeled data is associated with a weak label. Each weak label is a probability distribution over all entity classes, which can be used to train a discriminative NER model. One advantage of training a discriminative NER model is that it can use other token features while the generative model can only use labeling rules’ outputs as inputs. Therefore, even if a token is not matched by any labeling rules, it can still be predicted correctly by the discriminative model.

In our work, we use BiLSTM-CRF (Huang et al., 2015) as our discriminative model. The model first uses BiLSTM to generate a state representation for each token in a sequence. The CRF layer then predicts each token by maximizing the expected likelihood of the entire sequence based on the estimated labels.

²In our work, we also applied the linking rules developed by (Safranchik et al., 2020) on unlabeled data, so the label matrix also contains the results from linking rules.

3 Experimental Setup

In this section, we present the details of our experimental setup. First, we evaluate our method on three NER datasets, which we refer to NCBI, BC5CDR, and LaptopReview. Then, we describe the baseline methods compared in our experiments. We also give a detailed description of the seed rules used in our experiments.

3.1 Datasets

We evaluated our method on three datasets. Details of each dataset are described below.

NCBI (Doğan et al., 2014) contains 793 PubMed abstracts with 6,892 `Disease` mentions and is split into 593 train, 100 dev and 100 test data.

BC5CDR (Li et al., 2016) has 1,500 PubMed articles with 5,818 `Disease` and 3,116 `Chemical` mentions. It is split into train, dev, and test sets with 500 articles each.

LaptopReview (Pontiki et al., 2016) contains sentences regarding laptop `AspectTerms`, including 3,048 training and 800 test sentences. Following (Safranchik et al., 2020), we hold out 20% of training data as dev set.

Note that we use all training data as our unlabeled data by removing the manual annotations.

3.2 Compared Methods

We compare our method with state-of-the-art weakly supervised methods using dictionaries or heuristic rules as supervision. In this section, we briefly describe these baseline methods.

AutoNER (Shang et al., 2018) is a distantly supervised method, which automatically builds a neural named entity recognition model using dictionaries as weak supervision.

Snorkel (Ratner et al., 2020) is a general machine learning framework that can train classifiers using heuristic rules. By default, it uses a Naive Bayes generative model to denoise labeling rules by predicting each token’s label independently.

SwellShark (Fries et al., 2017) is an extension of Snorkel that was developed for biomedical NER. Same as Snorkel, it uses a naive Bayes generative model to denoise manual labeling rules. It also requires a special entity candidate generator to detect entity spans accurately before predicting their entity labels. In our experiments, we report both results using simple noun phrases as candidates, and that generated using extra expert effort.

LinkedHMM (Safranchik et al., 2020) is a framework for training sequence tagging models using weak supervision from manual rules. Besides using labeling rules, it can also use linking rules that indicate whether two consecutive tokens have the same label.

RuleType	NCBI		BC5CDR (Disease)		BC5CDR (Chem)		Laptop	
	S	M	S	M	S	M	S	M
Positive seeding rules								
Surface	875	0	632	K0	1.5	M0	274	0
Suffix	12	5	14	2	7	9	0	5
Prefix	0	7	2	13	0	8	0	5
PreNgram [†]	5	9	0	9	1	8	6	5
PostNgram [†]	4	6	5	10	2	9	0	5
DepRule	5	8	5	7	0	6	0	4
Negative seeding rules								
Surface	34	0	34	6	34	0	34	3
Suffix	0	10	0	18	0	22	0	5
Prefix	0	10	0	12	0	8	0	8
PreNgram [†]	0	15	0	18	0	16	0	9
PostNgram [†]	0	10	0	19	0	10	0	7
DepRule	0	8	0	13	0	5	0	4

Table 1: Number of positive and negative seed rules for each dataset. S denotes the number of seed rules by (Safranchik et al., 2020) and M denotes the rules we manually selected. [†] denotes that corresponding rule type includes both inclusive and exclusive rules.

3.3 Seed Rules

In our experiments, we used all the labeling rules³ developed by Safranchik et al. (2020) as part of positive seeding rules. Besides, we manually selected another small set of labeling rules as our input because: (1) we defined six types of rules as described in Section 2.1, but some types of rules were not used in Safranchik et al. (2020) (e.g., prefix rules are not used in NCBI), (2) our method requires negative seed rules, which are used to identify terms that are not entities, to initiate its learning process. To automatically learn new labeling rules, we use both labeling rules from (Safranchik et al., 2020) and our manually selected rules as seeding rules. Numbers of both positive and negative seeding rules used in our experiments are shown in Table 1.⁴ Our manually selected seed rules are

³In our experiments, we used the linked rules from (Safranchik et al., 2020) to obtain weak labels, which are another type of rules that can be used to vote if the two consecutive tokens have the same label, to train our LinkedHMM model. However, our work only focused on augmenting labeling rules, and did not augment linking rules.

⁴Note that in previous work a whole lexicon or dictionary is counted as one rule. However, in our work we count each term in a lexicon or dictionary as one surface form rule.

Method	Human Effort	NCBI			BC5CDR			LaptopReview		
		P	R	F1	P	R	F1	P	R	F1
Supervised	Full Annotations	85.2	89.2	87.2	87.2	88.0	87.5	83.5	82.2	82.9
Snorkel	Saf.'s Manual Rules	-	-	68.7	-	-	83.2	-	-	60.0
SwellShark	Fries' Manual Rules	64.7	69.7	67.1	85.0	83.5	84.2	-	-	-
	+Special Candidate Generator	81.6	80.1	80.8	86.1	82.4	84.2	-	-	-
AutoNER	Dictionaries	79.4	72.0	75.5	89.0	81.0	84.8	72.3	59.8	65.4
LinkedHMM	Saf.'s Manual Rules	-	-	78.7	-	-	85.9	-	-	68.2
	Our Seed Rules	89.4	70.7	79.0	88.0	84.5	86.2	82.7	59.8	69.4
GLARA	Our Seed Rules	89.9	73.2	80.2 \pm .2	88.2	84.6	86.3\pm.3	82.4	64.2	72.2\pm1.2

Table 2: Micro F1 performance of baselines and our method on test sets. Our results are the mean and std across 5 random runs. Saf.'s Manual Rules are the rules developed by Safranchik et al. (2020), and Fries' Manual Rules are those used in (Fries et al., 2017).

included in Appendix C.

3.4 Implementation Details

In our experiments, we create a graph for each type of rules for each dataset and learn new rules independently with the same setup. Prefix and Suffix candidate rules are generated by considering the first and last 3 to 6 characters, and PreNgram and PostNgram candidate rules are extracted with the windows of 1 to 3 tokens.

We use a two-layer graph attention network to train our graph model. After training, we select M new rules for each type of rules, where the value of M is searched between 20 and 500 on dev sets.

For different datasets, our discriminative NER models used different pre-trained contextual models. Since NCBI and BC5CDR datasets are in the biomedical domain, we finetuned our NER model on the pretrained SciBERT embedding (Beltagy et al., 2019), while for LaptopReview data, the NER model is finetuned on the pretrained BERT_{base} (Devlin et al., 2018) embedding. More details are provided in Appendix A.

4 Experimental Results

In this section, we first compare our method with state of the art methods using all available manual rules. Then, we evaluate our method under scenarios when only limited seeding rules are available, which are very common in real-world applications. Next, we conduct an ablation study to investigate the effectiveness of different types of rules and our newly proposed loss function based on centroid distance. Finally, we also perform a quality analysis of the automatically learned labeling rules.

4.1 Results with Abundant Seeding Rules

In this subsection, we report both the result of our *generative* model with augmented rules in Table 2 and the result of our *discriminative* NER model that is trained using weak labels from the generative model in Table 3.

Table 2 shows the performance of our generative model with augmented rules and baseline generative models⁵. The *Supervised* line is the performance of the LinkedHMM model trained on fully labeled training data. The results show that our method with augmented labeling rules performed best on BC5CDR and LaptopReview datasets. We also achieved a comparative F1 score of 80.2 on NCBI with SwellShark (F1 80.8). Note that SwellShark used lots of manual effort from experts to carefully tune a particular candidate generator for a given dataset. We also notice that, with augmented rules, our method outperforms LinkedHMM by an average of 2.0 F1 points, which demonstrates the effectiveness of augmented rules.

WeakSupervision	NCBI	BC5CDR	Laptop
Snorkel	73.4 \pm 1.7	82.2 \pm .5	63.5 \pm 1.7
LinkedHMM	79.0 \pm .4	83.0 \pm .2	69.0 \pm 1.1
GLARA	80.8\pm.3	83.5\pm.6	72.3\pm1.0

Table 3: Results of discriminative models using weak labels generated by different methods. Our results are the average of five runs.

Table 3 shows the performance of our discriminative NER model (BiLSTM-CRF) trained on weak labels from our GLARA with augmented rules and baseline discriminative models using weak labels from LinkedHMM and Snorkel without augmented rules (Safranchik et al., 2020). The results show

⁵Some scores were reported in (Safranchik et al., 2020)

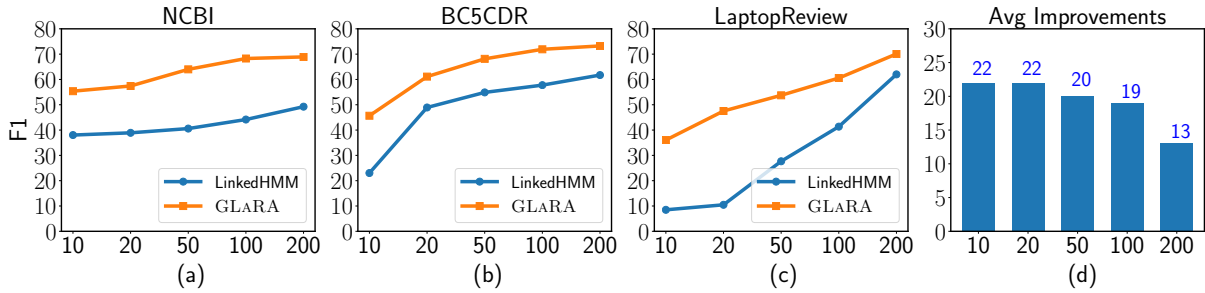


Figure 4: Results of LinkedHMM with only seeding rules and augmented rules by our GLARA method. The bottom numbers are the numbers of seeding rules used.

that our method achieved a +1.9 F1 point improvement over the second-best system. We also notice that our discriminative model performs slightly better on NCBI and Laptop, but worse on BC5CDR than the generative model, which is consistent with that from (Safranchik et al., 2020).

4.2 Results with Limited Seeding Rules

Though weakly supervised state-of-the-art methods reported in Table 2 achieved relatively good performance, they require a significant amount of manual effort from domain experts for designing and tuning labeling rules. Well-performing methods that require less manual effort are often more desirable. Therefore, we also evaluate our method with little manual effort (i.e., limited seeding rules). We conducted experiments by randomly select at most k rules for each rule type from our seed rules (Section 3.3), where $k \in \{10, 20, 50, 100, 200\}$. When a rule type has less than k seeding rules, we use all of them. Figure 4 shows the performance of the LinkedHMM model using only seeding rules and our GLARA method with augmented rules on three datasets. Figure 4(d) shows that, with automatically learned rules, our method can obtain average F1 gains of +22 points when using 10 seeding rules and +13 points when there are 200 seeding rules.

4.3 Impact of Different Types of Rules

To investigate the effectiveness of different types of labeling rules, we conducted ablation experiments to evaluate our generative model with augmented rules (i.e., GLARA) by adding each type of rules cumulatively. Results are shown in Table 4. Empty cells denote that the corresponding type of rules are not used. Results show that Prefix rules are most effective on the Laptop dataset, producing +1.5 F1 point improvement. All other rules except

PostNgram can improve the performance by +0.4 to +0.6 F1 points.

	NCBI	CDR	Laptop	Δ
LinkedHMM	78.7	85.9	68.2	
GLARA				
+DepRule	79.3	-	-	+0.6
+PostNgram	-	86.0	-	+0.1
+PreNgram	79.6	86.2	69.3	+0.5
+Prefix	-	-	70.8	+1.5
+Suffix	79.8	86.3	71.7	+0.4
+Surface	80.2	-	72.2	+0.5

Table 4: Impact of each type of rules used in our generative model. Δ denotes the average improvement achieved by adding rules cumulatively.

4.4 Effectiveness of Distance Loss \mathcal{L}_{dist}

In this work, we proposed a new graph neural network model with a new loss function, i.e., \mathcal{L}_{dist} in Eq. 3. This loss function measures the distance between the centroids of positive and negative seeding rules by computing their cosine similarity. The motivation is that the model should keep positive rules distant from negative ones during the learning process. Table 5 shows the performance of our generative model with and without the distance loss. We find that our method can obtain an average F1 gain of 1.2 points across three datasets with this new loss function.

GLARA	NCBI	BC5CDR	LaptopReview
w/ \mathcal{L}_{dist}	80.2 \pm 2	86.3 \pm 3	72.2 \pm 1.2
w/o \mathcal{L}_{dist}	79.3 \pm 1.2	86.1 \pm 1	69.7 \pm 8

Table 5: Results of our generative model using augmented rules learned by our graph neural network model with and without distance loss.

RuleType	Top 5 learned rules
Surface	neurologic_disease, spherocytic_hemolytic_anemia, episodic_hemolytic_anemia, choroid_plexus_carcinoma, sporadic_human_renal_cell_carcinoma
Suffix	*ioma, *ophy, *ndrome, *umonia, *phoma
Prefix	mesot*, mesoth*, prost*, dyst*, scler*
PreNgram	stage_ii_*, breast/_ovarian_*, heterogeneity/_in_*, a_syndromal_*, pathology_of_*
PostNgram	*_hepatitis, *_cardiomyopathy, *_carcinoma, *_colitis, *_lymphoma
DepRule	FirstTokenDep:amod_LastToken:syndrome FirstTokenDep:compound_LastToken:sarcoma SecondLastTokenDep:amod_LastToken:dystrophy SecondLastTokenDep:conj_LastToken:dystrophy SecondLastTokenDep:nmod_LastToken:syndrome

Table 6: Examples of learned new rules for recognizing diseases on NCBI data. “_” denotes the space character.

4.5 Quality Analysis of Learned New Rules

In Table 6, we present top 5 learned rules for each rule type that we automatically learned on the NCBI data set. Each rule is formatted in the way described in Section 2.1.

Table 7 shows the analysis results of the number of new rules (#Rule) and their accuracy (ACC) on the dev sets, which were learned during the experiments presented in Section 4.1. Results show that all the learned rules have accuracy $\geq 64\%$ with an average of 76%, which justifies the quality of these new rules.

RuleType	Dataset	#Rule	ACC
Surface	NCBI	301	78%
	Laptop	154	69%
Suffix	NCBI	64	100%
	CDR(Dis)	138	90%
Prefix	Laptop	25	66%
	Laptop	16	78%
PreNgram	NCBI	156	65%
	CDR(Dis)	19	64%
PostNgram	Laptop	17	71%
	CDR(Chem)	37	82%
DepRule	NCBI	47	70%

Table 7: Quality analysis of learned new rules.

Besides, we also manually analyzed why some learned rules are not helpful to improve recognition performance. First, some of the learned rules can be overlapping. For example, both “*demia→Disease” and “*edemia→Disease” can be used to recognize Diseases. However, if

*demia is learned first, then learning *edemia rule will not help because the entities matched by these two rules have large overlaps. Second, some of the rules learned from unlabeled data may not be applied to testing data due to the mismatch between two datasets. For example, though the PostNgram rule “*_dystonia→Disease” is an accurate rule to label diseases such as “oromandibular dystonia” and “responsive dystonia”. However, we do not find any matches on test data.

5 Related Work

Training reliable NER systems usually requires large annotation efforts, which is often considered expensive and impractical in certain domains. Therefore, previous studies have been trying to reduce the manual efforts required for annotation while producing comparable performance in NER tasks by utilizing manually designed rules that are cheap and accurate.

Studies have shown that human-defined rules (i.e dictionaries) can greatly aid NER tasks, especially in the domains where the identification of entities needs domain knowledge (Cohen and Sarawagi, 2004; Savova et al., 2010; Xu et al., 2010; Eftimov et al., 2017), and they can also be used to distantly create labeled data set to benefit machine learning models (Mann and McCallum, 2010; Neelakantan and Collins, 2015; Giannakopoulos et al., 2017).

While distantly labeled data sets can be created at a low cost to boost NER tasks, the models still suffer from the noise introduced by the imperfect rules. Therefore, denoising models have been proposed to allow the model to better tolerate the im-

perfect annotations created by rules. Shang et al. (2018) proposed AutoNER that trains NER systems using only lexicons with a tie-or-break tagging denoising model. Similarly, some recent work (Liu et al., 2019; Yang et al., 2018; Cao et al., 2019) have used a partial matching module to denoise the noisily labeled data sets.

Recently, weak supervision is proposed as another form of denoising framework without using any labeled data. Weakly supervised systems use handcrafted labeling rules to create weak training instances from unlabeled data and then use a denoising generative model to approximate the posteriors of the rules. In this process, the unknown gold label is treated as latent variable by the generative model. The performance of the labeling rules could be further enhanced by training a neural network-based discriminative model by treating the posteriors as soft labels. Related studies such as Snorkel (Ratner et al., 2020), SwellShark (Fries et al., 2017), LinkedHMM (Safranchik et al., 2020), and (Lison et al., 2020) have demonstrated great success with carefully curated labeling rules.

However, manually designing those high-quality rules often require domain expertise and easy to have a low sensitivity on identifying entities. In (Lison et al., 2020), the weak training data is created by broadly collecting available labeling rules from multiple sources, which demonstrates the importance of being able to automatically find new heuristics missed by human efforts. To find new heuristic rules on the basis of a relatively limited number of manually designed rules, previous studies have tried bootstrapping by relying on the co-occurrence, context and pattern features (Thelen and Riloff, 2002; Riloff et al., 2003; Yangarber, 2003; Shen et al., 2017; Tao et al., 2015; Berger et al., 2018; Yan et al., 2019).

Recent studies on graph neural networks has opened up another possibility for learning new rules. By internally infusing the semantics of the neighboring nodes, the popular Graph Convolutional Network (GCN) (Kipf and Welling, 2016) and Graph Attention Network (GAT) (Veličković et al., 2017) have shown great success in semi-supervised node classification when the number of labeled nodes is limited. Graph neural networks have been applied for many NLP tasks such as text classification (Yao et al., 2019; Zhang et al., 2019a; Hu et al., 2019), semantic role labeling (Marcheggiani and Titov, 2017), machine translation (Beck

et al., 2018), question answering (Song et al., 2018; Saxena et al., 2020), information extraction (Liu et al., 2018; Vashishth et al., 2018; Nguyen and Grishman, 2018; Sahu et al., 2019; Sun et al., 2019; Fu et al., 2019; Zhang et al., 2019b), etc. In our work, we proposed to use graph neural networks to learn new labeling rules. Based on Graph Attention Network (Veličković et al., 2017), we designed a new graph network model with a new loss function. Experimental results demonstrated that our model performed better than the original graph attention network on learning accurate labeling rules.

6 Conclusion

In this work, we proposed a weakly supervised NER framework that automatically learns high-quality new rules from only a handful of manually designed rules with a graph-based labeling rule augmentation method (GLARA). Experiments on three NER datasets demonstrate that our model outperforms baseline systems and achieves substantially better performance when the number of manual rules is limited. In addition, we also defined six types of rules that have been demonstrated useful for recognizing entities. In the future, we plan to improve GLARA by investigating more complex rule types and rule representation methods for weakly supervised NER.

References

- Stephen H Bach, Bryan He, Alexander Ratner, and Christopher Ré. 2017. Learning the structure of generative models without labeled data. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 273–282. JMLR. org.
- Daniel Beck, Gholamreza Haffari, and Trevor Cohn. 2018. Graph-to-sequence learning using gated graph neural networks. *arXiv preprint arXiv:1806.09835*.
- Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. Scibert: A pretrained language model for scientific text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3606–3611.
- Matthew Berger, Ajay Nagesh, Joshua Levine, Mihai Surdeanu, and Helen Zhang. 2018. Visual supervision in bootstrapped information extraction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2043–2053.

- Yixin Cao, Zikun Hu, Tat-Seng Chua, Zhiyuan Liu, and Heng Ji. 2019. Low-resource name tagging learned with weakly labeled data. *arXiv preprint arXiv:1908.09659*.
- William W Cohen and Sunita Sarawagi. 2004. Exploiting dictionaries in named entity extraction: combining semi-markov extraction processes and data integration methods. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 89–98.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.
- Rezarta Islamaj Doğan, Robert Leaman, and Zhiyong Lu. 2014. Ncbi disease corpus: a resource for disease name recognition and concept normalization. *Journal of biomedical informatics*, 47:1–10.
- Tome Eftimov, Barbara Koroušić Seljak, and Peter Korošec. 2017. A rule-based named-entity recognition method for knowledge extraction of evidence-based dietary recommendations. *PLoS one*, 12(6):e0179488.
- Jason Fries, Sen Wu, Alex Ratner, and Christopher Ré. 2017. Swellshark: A generative model for biomedical named entity recognition without labeled data. *arXiv preprint arXiv:1704.06360*.
- Tsu-Jui Fu, Peng-Hsuan Li, and Wei-Yun Ma. 2019. Graphrel: Modeling text as relational graphs for joint entity and relation extraction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1409–1418.
- Athanasios Giannakopoulos, Claudiu Musat, Andreea Hossmann, and Michael Baeriswyl. 2017. Unsupervised aspect term extraction with b-lstm & crf using automatically labelled datasets. *arXiv preprint arXiv:1709.05094*.
- Linmei Hu, Tianchi Yang, Chuan Shi, Houye Ji, and Xiaoli Li. 2019. Heterogeneous graph attention networks for semi-supervised short text classification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*.
- Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional lstm-crf models for sequence tagging. *arXiv preprint arXiv:1508.01991*.
- Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.
- Jiao Li, Yueping Sun, Robin J Johnson, Daniela Sciak, Chih-Hsuan Wei, Robert Leaman, Allan Peter Davis, Carolyn J Mattingly, Thomas C Wieggers, and Zhiyong Lu. 2016. Biocreative v cdr task corpus: a resource for chemical disease relation extraction. *Database*, 2016.
- Pierre Lison, Aliaksandr Hubin, Jeremy Barnes, and Samia Touileb. 2020. Named entity recognition without labelled data: A weak supervision approach. *arXiv preprint arXiv:2004.14723*.
- Tianyu Liu, Jin-Ge Yao, and Chin-Yew Lin. 2019. Towards improving neural named entity recognition with gazetteers. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5301–5307.
- Xiao Liu, Zhunchen Luo, and Heyan Huang. 2018. Jointly multiple events extraction via attention-based graph information aggregation. *arXiv preprint arXiv:1809.09078*.
- Gideon S Mann and Andrew McCallum. 2010. Generalized expectation criteria for semi-supervised learning with weakly labeled data. *Journal of machine learning research*, 11(Feb):955–984.
- Diego Marcheggiani and Ivan Titov. 2017. Encoding sentences with graph convolutional networks for semantic role labeling. *arXiv preprint arXiv:1703.04826*.
- Arvind Neelakantan and Michael Collins. 2015. Learning dictionaries for named entity recognition using minimal supervision. *arXiv preprint arXiv:1504.06650*.
- Thien Nguyen and Ralph Grishman. 2018. Graph convolutional networks with argument-aware pooling for event detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.
- Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*.
- Maria Pontiki, Dimitrios Galanis, Haris Papageorgiou, Ion Androutsopoulos, Suresh Manandhar, Mohammad Al-Smadi, Mahmoud Al-Ayyoub, Yanyan Zhao, Bing Qin, Orphée De Clercq, et al. 2016. Semeval-2016 task 5: Aspect based sentiment analysis. In *10th International Workshop on Semantic Evaluation (SemEval 2016)*.
- Alexander Ratner, Stephen H Bach, Henry Ehrenberg, Jason Fries, Sen Wu, and Christopher Ré. 2020. Snorkel: Rapid training data creation with weak supervision. *The VLDB Journal*, 29(2):709–730.
- Ellen Riloff, Janyce Wiebe, and Theresa Wilson. 2003. Learning subjective nouns using extraction pattern bootstrapping. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003*, pages 25–32.
- Esteban Safranchik, Shiyong Luo, Stephen H Bach, Stephen H Bach, Daniel Rodriguez, Yintao Liu, Chong Luo, Haidong Shao, Cassandra Xia, Souvik Sen, et al. 2020. Weakly supervised sequence tagging from noisy rules. In *Thirty-Fourth AAAI Conference on Artificial Intelligence*.

- Sunil Kumar Sahu, Fenia Christopoulou, Makoto Miwa, and Sophia Ananiadou. 2019. Inter-sentence relation extraction with document-level graph convolutional neural network. *arXiv preprint arXiv:1906.04684*.
- Guergana K Savova, James J Masanz, Philip V Ogren, Jiaping Zheng, Sunghwan Sohn, Karin C Kipper-Schuler, and Christopher G Chute. 2010. Mayo clinical text analysis and knowledge extraction system (ctakes): architecture, component evaluation and applications. *Journal of the American Medical Informatics Association*, 17(5):507–513.
- Apoorv Saxena, Aditay Tripathi, and Partha Talukdar. 2020. Improving multi-hop question answering over knowledge graphs using knowledge base embeddings. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4498–4507.
- Jingbo Shang, Liyuan Liu, Xiang Ren, Xiaotao Gu, Teng Ren, and Jiawei Han. 2018. Learning named entity tagger using domain-specific dictionary. *arXiv preprint arXiv:1809.03599*.
- Jiaming Shen, Zeqiu Wu, Dongming Lei, Jingbo Shang, Xiang Ren, and Jiawei Han. 2017. Setexpan: Corpus-based set expansion via context feature selection and rank ensemble. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 288–304. Springer.
- Linfeng Song, Zhiguo Wang, Mo Yu, Yue Zhang, Radu Florian, and Daniel Gildea. 2018. Exploring graph-structured passage representation for multi-hop reading comprehension with graph neural networks. *arXiv preprint arXiv:1809.02040*.
- Changzhi Sun, Yeyun Gong, Yuanbin Wu, Ming Gong, Daxin Jiang, Man Lan, Shiliang Sun, and Nan Duan. 2019. Joint type inference on entities and relations via graph convolutional networks. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1361–1370.
- Fangbo Tao, Bo Zhao, Ariel Fuxman, Yang Li, and Jiawei Han. 2015. Leveraging pattern semantics for extracting entities in enterprises. In *Proceedings of the 24th International Conference on World Wide Web*, pages 1078–1088.
- Michael Thelen and Ellen Riloff. 2002. A bootstrapping method for learning semantic lexicons using extraction pattern contexts. In *Proceedings of the 2002 conference on empirical methods in natural language processing (EMNLP 2002)*, pages 214–221.
- Shikhar Vashishth, Rishabh Joshi, Sai Suman Prayaga, Chiranjib Bhattacharyya, and Partha Talukdar. 2018. Reside: Improving distantly-supervised neural relation extraction using side information. *arXiv preprint arXiv:1812.04361*.
- Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903*.
- Hua Xu, Shane P Stenner, Son Doan, Kevin B Johnson, Lemuel R Waitman, and Joshua C Denny. 2010. Medex: a medication information extraction system for clinical narratives. *Journal of the American Medical Informatics Association*, 17(1):19–24.
- Lingyong Yan, Xianpei Han, Le Sun, and Ben He. 2019. Learning to bootstrap for entity set expansion. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 292–301.
- Yaosheng Yang, Wenliang Chen, Zhenghua Li, Zhengqiu He, and Min Zhang. 2018. Distantly supervised ner with partial annotation learning and reinforcement learning. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2159–2169.
- Roman Yangarber. 2003. Counter-training in discovery of semantic patterns. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 343–350.
- Liang Yao, Chengsheng Mao, and Yuan Luo. 2019. Graph convolutional networks for text classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7370–7377.
- Chen Zhang, Qiuchi Li, and Dawei Song. 2019a. Aspect-based sentiment classification with aspect-specific graph convolutional networks. *arXiv preprint arXiv:1909.03477*.
- Yan Zhang, Zhijiang Guo, and Wei Lu. 2019b. Attention guided graph convolutional networks for relation extraction. *arXiv preprint arXiv:1906.07510*.
- GuoDong Zhou and Jian Su. 2002. Named entity recognition using an hmm-based chunk tagger. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 473–480.

A Hyperparameter configuration

Graph Propagation Model We use the same graph architecture to train propagation for all types of rules. The model contains 2 graph attention layers each with 3 attention heads and dropout rate is set to be 0.5. The hidden size of each layer is set as 64. We use Adam optimizer with a learning rate of 0.0001. Other hyperparameters (training epoch and number of selected new rules) are presented in Table 8.

Hyperparam	NCBI		BC5CDR		Laptop	
	epoch	#rules	epoch	#rules	epoch	#rules
SurfaceForm	50	50	50	-	50	25
Prefix	50	-	50	-	50	15
Suffix	50	25	50	25(D)	50	25
PreNgram _{inclusive}	50	25	30	-	30	
PreNgram _{exclusive}	50	-	50	25(D)	30	
PostNgram _{inclusive}	50	-	30		50	
PostNgram _{exclusive}	50	-	50	30(C)	50	
Dependency	50	25	50	-	50	25

Table 8: Summary of hyperparameters for propagation. In BC5CDR data. “D” denotes the number of selected rules for Disease and “C” denotes the that for Chemical for the corresponding rule type. “-” means the corresponding type of propagated rules are not used on our final model.

Generative Model Table 9 presents the hyperparameters used for tuning our LinkedHMM generative model, including “Initial Accuracy (estimated initial accuracy of the rules)”, “Accuracy Prior (regularization for initial accuracy)”, and “Balance Prior (the entity class distribution)”. We used grid search to find the best hyperparameters. The search ranges are created around the default settings of the LinkedHMM model on the three data sets. For training epoch, we use the default setting, 5, for all three datasets. For more details about the hyperparameters, please refer to (Safranchik et al., 2020).

Hyperparam	NCBI	CDR	Laptop
Init Acc	Search [0.75-0.95]	[0.75-0.95]	[0.75-0.95]
	Best 0.85	0.85	0.9
Acc Prior	Search [45-65]	[0-15]	[0-5]
	Best 55	5	1
Bal Prior	Search [440, 460]	[440, 460]	[0, 20]
	Best 450	450	10

Table 9: Summary of hyperparameters for training the LinkedHMM generative model on each data set. The search steps for InitialAccuracy, AccuracyPrior, and BalancePrior are 0.05, 5, and 5, respectively.

Discriminative Model Table 10 presents the hyperparameter configuration for training discriminative model (BiLSTM-CRF). BERT_{base} is used to extract word embeddings and fine-tuned with BiLSTM-CRF. All discriminative models are trained on a 11G 1080Ti GPU with training time being up to ~ 20 s/epoch. All models have ~ 110 M parameters.

B Performance on Development Data

In Table 11, we present the performance of the LinkedHMM model on development sets, with

Hyperparam		NCBI	BC5CDR	Laptop
BERT		SciBERT	SciBERT	BERT
BiLSTM	Hidden Dim	256	256	256
	Dropout	0.1	0.1	0.1
CRF		yes	yes	no
AdamW	Learning Rate	1e-4	1e-4	1e-4
	Epoch	30	30	30
Batch size		8	8	8
Max sent length		128	128	128

Table 10: Summary of hyperparameters for training discriminative model on each data set.

the additional seeding rules manually selected by us, referred as LinkedHMM-M, the performance of the following discriminative model, referred as LinkedHMM-M-D. Also, we report the performance of our models, GLARA and GLARA-D (with discriminative model) on development sets.

Model	NCBI	BC5CDR	Laptop
LinkedHMM-M	82.3	87.5	70.1
LinkedHMM-M-D	82.8 \pm .3	84.5 \pm .2	71.5 \pm .8
GLARA	83.1 \pm .2	87.4 \pm .3	72.3 \pm .8
GLARA-D	83.4 \pm .3	84.5 \pm .1	72.6 \pm .6

Table 11: Micro F1 performance on each development data set. LinkedHMM – M denotes the baseline LinkedHMM model trained with the extra manually selected seeding rules. LinkedHMM – M – D denotes the discriminative mode (LSTM-CRF) trained based on LinkedHMM – M. Similarly, GLARA – D denotes the LSTM-CRF model trained based on GLARA.

C Manually Selected Seeds for propagation

As mentioned in the paper, the baseline LinkedHMM does not have seeding rules for all types of rules. For example, negative seeding rules are missing from the baseline LinkedHMM model, except that SurfaceForm rules uses a list of stop-words as negative seeding rules. For some types of seeding rules, there are only a few available that are not good enough for training the propagation model. Therefore, for the rule types that do not have enough seeds, we manually select a small set of additional rules as seeds. Note that we keep the total number of seeding rules (including the ones from baseline system) less than 15. We report the manually selected seeding rules for NCBI, BC5CDR-Disease, BC5CDR-Chemical, and LaptopReview in Table 12, Table 13, Table 14, and Table 15, respectively.

Rules		NCBI
Surface	pos	-
	neg	-
Suffix	pos	*skott, *drich, *umour, *axia, *iridia
	neg	*ness, *nant, *tion, *ting, *enesis, *riant, *tein, *sion, *osis, *lity
Prefix	pos	carc*, myot*, tela*, ovari*, atax*, carcin*, dystro*
	neg	defi*, comp*, fami*, poly*, chro*, prot*, enzym*, sever*, develo*, varian*
exclusive PreNgram	pos	suffer_from_*, fraction_of_*, pathogenesis_of_*, cause_severe_*
	neg	-pron_*, suggest_that_*, -cell_*, presence_of_*, expression_of_*, majority_of_*, loss_of_*, associated_with_*, impair_in_*, cause_of_*, defect_in_*, family_with_*
inclusive PreNgram	pos	breast_and_ovarian_*, x_-link_*, breast_and_*, stage_iii_*, myotonic_*
	neg	enzyme_*, primary_*, non_-_*
exclusive PostNgram	pos	
	neg	*_and_the, *_cell_line, *_in_the
inclusive PostNgram	pos	*_-t, *_cell_carcinoma, *_muscular_dystrophy, *_s_disease, *_carcinoma, *_dystrophy
	neg	*_muscle, *_ataxia, *_system, *_defect, *_other_cancer, *_i, *_ii
DepRule	pos	FirstTokenDep:amod_LastToken:dystrophy, FirstTokenDep:punct_LastToken:telangiectasia, FirstTokenDep:compound_HeadSurf:t, FirstTokenDep:amod_LastToken:dysplasia SecondLastTokenDep:compound_LastToken:syndrome,
	neg	FirstTokenDep:amod_LastToken:deficienc, FirstTokenDep:amod_LastToken:deficiency FirstTokenDep:amod_LastToken:defect, FirstTokenDep:pobj_LastToken:cancer SecondLastTokenDep:compound_LastToken:cancer, SecondLastTokenDep:compound_LastToken:disease SecondLastTokenDep:appos_LastToken:t, SecondLastTokenDep:compound_LastToken:t

Table 12: Manually selected seeding rules for NCBI dataset. “-” means no seeding rules selected, and we only use the rules provided in baseline.

Rules		BC5CDR (Disease)
Surface	pos	-
	neg	-
Suffix	pos	*epsy, *nson
	neg	*ing, *tion, *tive, *lity, *mone, *fect, *crease, *sion, *lion, *elet, *gical, *nosis, *sive, *ment, *tory, *sionetic, *ency, *ture,
Prefix	pos	anemi*, dyski*, heada*, hypok*, hypert*, ische*, arthr*, hypox*, toxic*, arrhyt*, ischem*, hypert*, dysfunc*
	neg	symp*, resp*, funct*, inter*, decre*, prote*, neuro*, cardi*, myoca*, ventr*, decre* syst*
exclusive PreNgram	pos	to_induce_*, w_-_*, and_severe_*, suspicion_of_*, die_of_*, have_severe_*, of_persistent_*, cyclophosphamide.associate_*
	neg	seizure_and_*, symptom_and_*, dysfunction_and_*, failure_with_*, sign_of_*, lead_to_*
inclusive PreNgram	pos	parkinson.'s_*, torsade.de_*, acute.liver_*, neuroleptic_*, malignant_*, alzheimer.'s_*, congestive.heart_*, migraine_with_*, sexual_side_*, renal.cell_*, tic_-_*
	neg	renal_function_*, decrease_in_*, increase_in_*, reduction_in_*, rise_in_*, loss_of_*, chronic.liver_*, abnormality_in_*, human.immunodeficiency_*, optic.nerve_*, drug_-_*, non_-_*
exclusive PostNgram	pos	-
	neg	*_and_the, *_cell_line, *_in_the
inclusive PostNgram	pos	*.'s_disease, *_infarction, *_s_sarcoma, *_epilepticus, *_artery_disease, *_de.pointe *_insufficiency, *_with_aura, *_artery_spasm, *_s_encephalopathy
	neg	*_toxicity, *_pain, *_fever, *_function, *_blood_pressure, *_effect, *_impairment, *_loss *_event, *_protein, *_pressure, *_impair, *_phenomenon, *_system, *_side_effect *_of_disease
DepRule	pos	FirstTokenDep:poss_LastToken:disease FirstTokenDep:compound_LastToken:cancer, FirstTokenDep:amod_LastToken:dysfunction FirstTokenDep:compound_LastToken:disease, FirstTokenDep:compound_LastToken:failure FirstTokenDep:compound_LastToken:anemia, FirstTokenDep:compound_LastToken:cancer SecondLastTokenDep:pobj_LastToken:disease
	neg	FirstTokenDep:amod_LastToken:toxicity FirstTokenDep:amod_LastToken:impairment, FirstTokenDep:amod_LastToken:syndrome FirstTokenDep:amod_LastToken:complication, FirstTokenDep:amod_LastToken:symptom FirstTokenDep:amod_LastToken:damage, FirstTokenDep:amod_LastToken:disease FirstTokenDep:amod_LastToken:function, FirstTokenDep:amod_LastToken:damage FirstTokenDep:compound_LastToken:loss, SecondLastTokenDep:nmod_LastToken:b SecondLastTokenDep:conj_LastToken:arrhythmia SecondLastTokenDep:pobj_LastToken:symptom

Table 13: Manually selected seeding rules for BC5CDR (Disease) dataset. “-” means no seeding rules selected, and we only use the rules provided in baseline.

Rules	BC5CDR (Chemical)	
Surface	pos	-
	neg	-
Suffix	pos	*pine, *icin, *dine, *ridol, *athy, *zure, *mide, *fen, *phine
	neg	*ing, *tion, *tive, *tory, *inal, *ance, *duce, *atory, *mine, *line, *tin, *rate, *late, *ular, *etic, *onic, *ment, *nary, *lion, *ysis, *logue, *mone
Prefix	pos	chlor*, levo*, doxor*, lithi*, morphi*, hepari*, ketam*, potas*
	neg	meth*, hepa*, prop*, contr*, pheno*, contra*, acetyl*, dopami*
exclusive PreNgram	pos	dosage_of_*, sedation_with_*, mg_of_*, application_of_*, -_release_*, ingestion_of_*, intake_of_*
	neg	of_to_*, to_the_*, be_the_*, with_the_*, in_the_*, on_the_*, for_the_*
inclusive PreNgram	pos	external_*, mk_*, mk_-_*, cis_*, cis_-_*, nik_*, nik_-_*, ly_*, ly_-_*, puromycin_*
	neg	reduce_*, all_*, a_*, the_*, of_*, alpha_*, alpha_-_*, beta_*, beta_-_*
exclusive PostNgram	pos	*-_associate, *_-_induced
	neg	*-_related, *_that_of, *_of_the
inclusive PostNgram	pos	*_aminocaproic_acid, *_-_aminocaproic *_acid, *_retinoic_acid, *_dopa, *_tc *_-_aminopyridine, *_aminopyridine, *_-_penicillamine, *_-_dopa, *_-_aspartate, *_fu *_hydrochloride
	neg	*_drug, *_cocaine, *_calcium, *_receptor_agonist, *_blockers, *_block_agent *_inflammatory_drug
DepRule	pos	FirstTokenDep:amod_LastToken:oxide, FirstTokenDep:compound_LastToken:chloride FirstTokenDep:amod_LastToken:acid, FirstTokenDep:compound_LastToken:acid FirstTokenDep:compound_LastToken:hydrochloride SecondLastTokenDep:amod_LastToken:aminonucleoside,
	neg	FirstTokenDep:compound_LastToken:a, SecondLastTokenDep:pobj_LastToken:acid SecondLastTokenDep:pobj_LastToken:a, SecondLastTokenDep:pobj_LastToken:the SecondLastTokenDep:pobj_LastToken:a

Table 14: Manually selected seeding rules for BC5CDR (Chemical) dataset. “-” means no seeding rules selected, and we only use the rules provided in baseline.

Rules	LaptopReview	
Surface	pos	-
	neg	-
Suffix	pos	*pad, *oto, *fox, *chpad, *rams
	neg	*ion, *ness, *nant, *lly, *ary
Prefix	pos	feat*, softw*, batt*, Win*, osx*
	neg	pro*, edit*, repa*, rep*, con*, dis*, appl*, equip*
exclusive PreNgram	pos	-
	neg	in_the_*, on_the_*, for_the_*, -pron_*
inclusive PreNgram	pos	windows_*, hard_*, extended_*, touch_*, boot_*
	neg	mac_*, apple_*, a_*, launch_*, software_*
exclusive PostNgram	pos	*_and_seal, *_that_come_with
	neg	*_shut_down, *_do_not_work,
inclusive PostNgram	pos	*_x, *_xp, *_vista, *_drive, *_processing
	neg	*_screen, *_software, *_quality, *_technical, *_cut
DepRule	pos	FirstTokenDep:compound_LastToken:port, FirstTokenDep:compound_LastToken:button FirstTokenDep:nummod_LastToken:ram, FirstTokenDep:amod_LastToken:drive
	neg	FirstTokenDep:compound_LastToken:option, SecondLastTokenDep:pobj_LastToken:plan SecondLastTokenDep:nsubj_LastToken:design, SecondLastTokenDep:pobj_LastToken:plan

Table 15: Manually selected seeding rules for LaptopReview dataset. “-” means no seeding rules selected, and we only use the rules provided in baseline.