TEBNER: Domain Specific Named Entity Recognition with Type Expanded Boundary-aware Network

Zheng Fang[1,2], **Yanan Cao**[1,2*], **Tai Li**[3], **Ruipeng Jia**[1,2]
Fang Fang[1,2], **Yanmin Shang**[1,2] and **Yuhai Lu**[1,2*]
[1]Institute of Information Engineering, Chinese Academy of Sciences
[2]School of Cyber Security, University of Chinese Academy of Sciences
[3]Bytedance
[1,2]{fangzheng, caoyanan, jiaruipeng, fangfang0703, shangyanmin
luyuhai }@iie.ac.cn, [3]litai.vincent@bytedance.com

Abstract

To alleviate label scarcity in Named Entity Recognition (NER) task, distantly supervised NER methods are widely applied to automatically label data and identify entities. Although the human effort is reduced, the generated incomplete and noisy annotations pose new challenges for learning effective neural models. In this paper, we propose a novel dictionary extension method which extracts new entities through the type expanded model. Moreover, we design a multi-granularity boundary-aware network which detects entity boundaries from both local and global perspectives. We conduct experiments on different types of datasets, the results show that our model outperforms previous state-of-the-art distantly supervised systems and even surpasses the supervised models.

1 Introduction

Named Entity Recognition (NER) is the task of detecting mentions from text and classifying them into predefined types. It is a fundamental task in the field of natural language processing (NLP), which can facilitate many other tasks, such as entity linking (Fang et al., 2020), machine translation (Gekhman et al., 2020), and question answering (Li et al., 2020a). However, most of existing NER methods require large amounts of manually annotated texts for training supervised models, which is difficult to implement in the specific domain because domain-expert annotation is expensive and time-consuming.

To alleviate the label scarcity problem, distant supervision methods (Fries et al., 2017; Shang et al., 2018b; Liang et al., 2020) have been applied to automatically generate labeled data and recognize entities. Given a raw corpus and a dictionary, above methods firstly label entities by exact string matching, and then use the annotated dataset to train the well-designed neural models to recognize entities. Although the human effort is reduced, the labels generated by the string matching method pose two challenges.

The first challenge is incomplete annotations. Because most of existing dictionaries have limited coverage on domain entities, just using the given dictionary will make many out-of-dictionary entities unmatched and generate a large number of false-negative labels. By analyzing several commonly used datasets (e.g., BC5CDR, NCBI, MeSH), we find that the original dictionary only covers about 50% of domain entities, which may weaken the performance of subsequent NER model. To increase the number of label entities, previous works (Fries et al., 2017; Liu et al., 2020) attempt to expand the dictionary by heuristic rules. However, these rule-based methods are difficult to migrate to other domains. So, how to extend the dictionary with more general pattern is the first problem we need to solve.

The second challenge is the difficulty of recalling new entities. Actually, even for the supervised models, new entities that have not been annotated are also difficult to be recalled (Shang et al., 2018b) because of the limited model capability. Most of previous NER methods, such as sequential label models (Chiu and Nichols, 2016; Ma and Hovy, 2016), and boundary detection models (Wang et al., 2018; Li et al., 2020b), only utilize the context information to recognize entities. However, the tight internal connection among entities and the global statistical features in domain corpus, which could contribute to identifying entities, are usually ignored by previous methods. Therefore, how to recall new entities with multi-granularity information is the second problem we need to solve.

To address two issues mentioned above, we propose a new distantly supervised method named TEBNER (Type Expanded Boundary-aware NER) in specific domains. To expand the original dictio-

[*] Corresponding authors: Yanan Cao and Yuhai Lu.

Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 198–207
November 7–11, 2021. ©2021 Association for Computational Linguistics

nary, we try to extract high-quality phrases from the raw corpus and view them as potential entities. Considering that these mined phrases lack corresponding type information and even contain noisy results, we use an entity typing model to classify and filter them based on their context information. Then these typed phrases are added to the original dictionary to resolve the incomplete annotation problem. For the purpose of recalling more new entities, we design multi-granularity boundary labeling strategies, which can capture boundary information from different perspectives. Specifically, we utilize the token interaction tagger to find the internal connection between entity tokens, the sequence labeling strategy to distinguish explicit entity boundaries in sentence and global statistical features of the whole corpus to recall potential entities. After getting the boundary results, we reuse the trained entity typing model to further classify entities and filter the noise results. In this way, we will get the trade-off between recall and precision for new entity detection.

In summary, the main contributions of this paper are listed as follows:

- We propose a novel dictionary extension method, which rely on semantic context, neither on ambiguous strings nor on artificial rules. Experiments show that our dictionary extension method significantly improves the quality of distantly supervised annotations.

- We propose a multi-granularity boundary-aware network which integrates the information at word, sentence and corpus level. Experiments show that fusing different granularity boundary results can significantly improve the recall rate of NER model.

- We conduct extensive experiments on three benchmark datasets and our TEBNER model achieves the best performance with dictionaries only and no human efforts. On several datasets, our approach is even better than the supervised models.

2 Related Work

As a fundamental task, named entity recognition (NER) has drawn much attention of researchers. Most previous approaches model the NER problem as a sequence labeling task and use popular architectures like NN-CRF (Ma and Hovy, 2016; Chiu and Nichols, 2016). Recently, to recognize nested entities, many studies also propose to detect each entity boundaries individually (Wang et al., 2018; Zheng et al., 2019; Li et al., 2020b). With the burgeoning popularity of pre-training methods, large-scale language models such as ELMo (Peters et al., 2018) and BERT (Devlin et al., 2019) are also utilized in NER task, yielding state-of-the-art performances. However, all of above supervised models require a great quantity of manually annotated data, which are usually labor-intensive and time-consuming.

To alleviate human efforts, distant supervision methods are widely used in NER task (Shang et al., 2018b; Cao et al., 2019; Liu et al., 2020; Xue et al., 2020; Liang et al., 2020; Lison et al., 2020). For example, Shang et al. (Shang et al., 2018b) marks out-of-dictionary phrases as potential entities with a special "unknown" type and propose a neural model AutoNER with a token interaction tagger. However, these untyped phrases are less helpful to identify new entities. HAMNER (Liu et al., 2020) expands the dictionary with headwords and design a span-level model, which predicts entity boundaries by an entity classification model. But for sentences with complex structures, it is difficult to detect boundaries by just using entity type information. Unlike these works, TEBNER annotates phrases with semantic context and distinguishes entity boundaries by fusing multi-granularity information, which can generate labels with high precision and recall rate.

3 Problem Definition

Formally, given a sequence of words $X = [x_1, x_2, ..., x_n]$, we denote an entity as $e_t = [x_i, ..., x_j](0 \leq i \leq j \leq n)$, where $< i, j >$ represents its boundary and t indicates the entity type. Specifically, entity types include pre-defined types (e.g., Disease, Chemical) and *none* type which denotes non-entity. In distant supervision NER task, we only need a dictionary D as input in addition to the original text. Each dictionary entry contains the surface name and the entity type. In the training phase, we use dictionary matching method to generate annotations on the training corpus.

4 The Proposed Method

The overall structure of our TEBNER model is shown in Figure 1. The proposed framework mainly includes two parts: Dictionary Extender

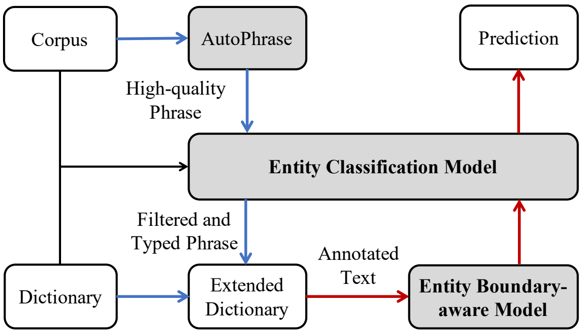

Figure 1: The process of our distant supervision method. Blue arrows show the dictionary expansion procedures, including high-quality phrase extraction, entity classification, and entity filtering. Red arrows show the entity recognization procedures, including annotation generation, entity boundary detection and entity type prediction.

which enriches entities by assigning types to extracted high-quality phrases, Entity Recognizer which identifies entity boundaries by fusing multi-granularity information and predicts entity types through the trained classifier. In the following, we will introduce the technical details of these modules.

4.1 Dictionary Extender

As the training annotations in the distantly supervised NER task are only generated from the entity dictionary, the coverage and quality of the dictionary become the key factors to improve the model performance, especially for the neural network. Therefore, we use a dictionary extender module to generate high-quality labels with high coverage to the target corpus. Our dictionary extender mainly consists of three parts: high-quality phrase extraction, entity classification and entity filtering.

High-quality phrase extraction As in previous work(Shang et al., 2018b; Liu et al., 2020), we utilize the AutoPhrase (Shang et al., 2018a) to extract high-quality phrase from domain corpus. AutoPhrase is a distantly supervised phrase mining tool, which generates frequent phrase candidates according to popularity requirement and estimates phrase quality based on features about concordance and informativeness. The main input to the tool is a corpus and a dictionary, and the output is a ranked list of phrases with decreasing quality score. To obtain the high-quality phrases, we only select phrases with score higher than a certain threshold (e.g., 0.5 for multi-word phrase and 0.9 for single-word phrase). After getting the out-of-dictionary phrases, we treat them as potential entities and propose an entity classification model to predict phrase types.

Entity classification The entity classification model is used to classify and filter the mined phrases and candidate entities. It is trained on annotated corpus generated by the original dictionary matching. Considering that some mined phrases are not real entities, we further add non entities to the training corpus to help the classification model recognize noisy entities. Specifically, we label the phrases in the corpus with lower scores (e.g., less than 0.3) as *none* entity type. We use pre-trained language model BERT as the backbone, which has been proven to be able to capture rich language information from text. Given an entity $e_t = [x_i, ..., x_j]$ and its context, we construct the input of each entity as:

$$[CLS]\ \ ctxt_l\ \ [x_i] \ldots [x_j]\ \ ctxt_r\ \ [SEP]$$

where $[x_i]$, $[x_j]$ denote the token at the beginning and the end of the entity respectively. $[x_i] \ldots [x_j]$ are the word-piece tokens of the entity. $ctxt_l$ and $ctxt_r$ denote the context before and after the entity respectively. After getting the BERT output of each token in the sentence, we concatenates the representations of $[CLS]$, $[x_i]$, $[x_j]$ and input them into a fully-connected layer. Then the representation of the entity will be sent into a softmax layer to predict the type label:

$$V_h = V_{[cls]} \oplus V_{[x_i]} \oplus V_{[x_j]} \tag{1}$$

$$V_e = Relu(W_t^1 V_h + b_t^1) \tag{2}$$

$$P(y_t|e_t) = softmax(W_t^2 V_e + b_t^2) \tag{3}$$

where W_t^1, W_t^2 and b_t^1, b_t^2 are trainable parameters, $P(y_t|e_t)$ denotes the probability of entity e_t being predicted to the type t. Generally, we compute the loss of type label prediction as follows:

$$L_{type} = -\sum_{i=1}^{n} y_i \log(P(y_i|e_i)) \tag{4}$$

After training the entity typing model, we backtrack and reconstruct the annotation data by out-of-dictionary phrases, and assign each phrase with the highest probability type.

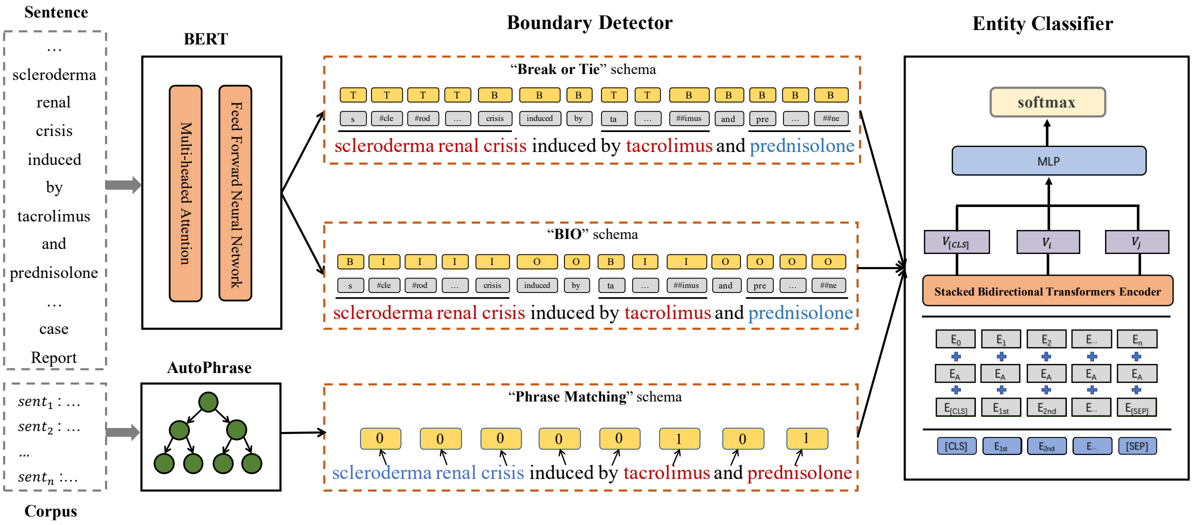
Sentence
…
scleroderma
renal
crisis
induced
by
tacrolimus
and
prednisolone
…
case
Report
BERT
Multi-headed Attention
Feed Forward Neural Network
$sent_1$: …
$sent_2$: …
…
$sent_n$:…
Corpus
AutoPhrase
Boundary Detector
"Break or Tie" schema
T T T T B B B T T B B B B B
s #cle #rod … crisis induced by ta … ##imus and pre … ##ne
scleroderma renal crisis induced by tacrolimus and prednisolone
"BIO" schema
B I I I I O O B I I O O O O
s #cle #rod … crisis induced by ta … ##imus and pre … ##ne
scleroderma renal crisis induced by tacrolimus and prednisolone
"Phrase Matching" schema
0 0 0 0 0 1 0 1
scleroderma renal crisis induced by tacrolimus and prednisolone
Entity Classifier
softmax
MLP
$V_{[CLS]}$
V_i
V_j
Stacked Bidirectional Transformers Encoder
E_0 E_1 E_2 $E_{...}$ E_n
E_A E_A E_A E_A E_A
$E_{[CLS]}$ E_{1st} E_{2nd} $E_{...}$ $E_{[SEP]}$
[CLS] E_{1st} E_{2nd} $E_{...}$ [SEP]

Figure 2: The two-stage pipeline framework. In Stage I, "Break or Tie" schema, "BIO" schema and "Phrase Matching" schema are utilized to identify entity boundaries at the word, sentence and corpus level separately, where the entity marked in red represents the result correctly predicted by the model. On the contrary, the entity marked in blue indicates that it is not recognized correctly. In Stage II, the entity classification model trained in the dictionary expansion procedure is used to predict entity types.

Entity filtering We first filter the phrases that are predicted as *none* type by the entity classification model. Moreover, there are some phrases predicted as several different types. We skip the phrases that are identified as multiple categories during the entity recognizer module training. As for phrases with consistent results, we add them to original dictionary to improve the entity coverage. Finally, the extended dictionary will be used to generate more annotations for the training of entity recognizer.

4.2 Entity Recognizer

Previous works (Ma and Hovy, 2016; Shang et al., 2018b; Cao et al., 2019; Wang et al., 2020) always jointly model the entity boundary detection and classification tasks. In general, it is effective for the supervised model using manual annotation data. However, there are many labeling errors in the data annotated by distantly supervised method. The joint learning method can easily lead to overfitting of the NER model, which makes it difficult to identify unlabeled entities. With this in mind, we utilize a pipeline framework which learns entity boundary and entity type separately. The overall structure of our framework is shown in Figure 2.

Our entity recognizer contains two components: boundary detector and entity classifier. As the final results in NER are only generated from the boundary detector, we will recall candidate entities as comprehensively as possible to ensure that the target entity can be input into the entity classifier. To achieve this goal, three kinds of tagging schemas are utilized to identify entity boundaries at the word, sentence and corpus level. In the following, we will describe each tagging schema in detail.

"Break or Tie" Tagging Schema To capture boundary information at word granularity, we construct a token interaction tagger to distinguish whether two adjacent tokens are tied in the same entity mention or not. The key motivation is that domain entities usually contain the specific words, modeling the connection between adjacent tokens can help to find new entities with the same domain words. For example, in the field of biology, many disease entities may contain kidney, lung and other organ nouns; in the field of finance, many institutional entities may contain insurance, bank, etc. Inspired by this insituition, we utilize a "Break or Tie" tagging schema to recognize domain entities. As shown in Figure 2, (i) **T** (Tie) indicates that both of the two adjacent tokens belong to the same entity. (ii) **B** (Break) means that the ties between adjacent tokens are broken into two parts.

Specifically, we build a binary classifier to distinguish whether current token is connected to the next one in the sentence. Given the output from BERT, the representation of i-th token and $i+1$-th token are concatenated to a new feature vector, which is referred as $V_i^{'}$ for the i-th token. The model predicts the probability of each token being connected with next one as follows:

$$P(c_i|V_i^{'}) = \frac{exp(c_i^T V_i^{'})}{\sum_{c_k \in C} exp(c_k^T V_i^{'})} \tag{5}$$

where c_i is the label between the i-th and its next tokens, and C is the set of connection modes (e.g., "Tie" and "Break"). To train the model, we adopt the following cross entropy loss function:

$$L_{word} = -\frac{1}{n}\sum_{x}\left[y \ln y^{'} + (1-y)\ln(1-y^{'})\right] \tag{6}$$

where $y \in \{0, 1\}$ represents the label of each token, and $y^{'} \in [0, 1]$ indicates the predicted result of our model. In the inference stage, we connect the tokens between every two consecutive "Break" to form a candidate entity.

"BIO" Tagging Schema It is worth noting that above token interaction tagger cannot detect entities which just contain single token. Moreover, besides the tight internal connection between words, context information is also important for identifying entities. Therefore, we follow the sequence labeling framework using "BIO" tagging scheme to detect entities at sentence level. Concretely, we tag the beginning token of an entity by "**B**", the other token of this entity by "**I**", and the non-entity tokens by "**O**". As shown in Figure 2, we first tokenize each word in sentence and pass it through BERT Transformer stacks. Then we use a Dense layer with the softmax activation function as the entity classifier to get probability of the labels from the contextualized representation. Similar to the token interaction tagger mentioned above, we use the cross entropy loss function in the model training process and connect the tokens between "B" and "O" to form a candidate entity in the inference process.

"Phrase Matching" Tagging Schema Most of the previous distantly supervised models (Shang et al., 2018b; Cao et al., 2019; Liu et al., 2020) use deep neural network to identify entity boundaries at the word or sentence level. However, the statistical features of domain entities in the corpus are often ignored by them. For example, the part-of-speech tagging, the term frequency, and the probabilities of an entity in quotes and brackets are all helpful to identify boundaries. Therefore, we use a phrase mining tool which can capture multiple features to extract high-quality phrases and detect candidate entities from the testing data set through exact string matching.

Algorithm 1 The Process of TEBNER

```
Input: Domain Corpus, Dictionary $D$
Output: The named entities $\Gamma = \{e_1, ..., e_N\}$ in test dataset $\mathcal{T}$
Extract high-quality phrases $P$ by AutoPhrase
Based on $D$, detect all entities by string matching, and extract noise entities from corpus.
for $epoch \leftarrow 1$ to $n$ do
    Update $W_t^1, W_t^2$ and $b_t^1, b_t^2$ w.r.t. $L_{type}$
end for
Extend $D$ to $D_{ext}$ by assigning types to $P$
Reconstruct the annotation data $\mathcal{S} = \{S_1, ..., S_m\}$ by $D_{ext}$
for $S$ in $\mathcal{S}$ do
    Update $\theta_w$ of token interaction model $M_w$
    Update $\theta_s$ of sequence label model $M_s$
end for
for $S$ in $\mathcal{T}$ do
    Predict candidate entities $\mathbf{E}_{word}$ by $M_w$
    Predict candidate entities $\mathbf{E}_{sent}$ by $M_s$
    Get candidate entities $\mathbf{E}_{corp}$ by phrases $P$
    Combine $\mathbf{E}_{word}$, $\mathbf{E}_{sent}$, $\mathbf{E}_{corp}$ as $\mathbf{E}_{all}$
    Assign types to $\mathbf{E}_{all}$
end for
```

Finally, the results of the above three methods are fused and input into the entity classification model (mentioned in 4.1). In this stage, the entity classification model is trained on annotated corpus labeled by the extended dictionary. If the candidate entity does belong to a given entity type (not *none*), we output it as the final result. The details of the overall process of our model are presented in Algorithm 1.

5 Experiments

To evaluate the effectiveness of our method, we conduct experiments on a series of popular distantly supervised NER datasets which are also used by (Fries et al., 2017; Shang et al., 2018b; Liu et al., 2020). The results show that our model achieves the state-of-the-art performance.

5.1 Experiment Setup

Dataset We train and evaluate our model on three benchmark datasets. The statistics of the datasets are shown in Table 1.

Dataset	BC5CDR	NCBI-Disease	LaptopReview
Domain	Biomedical	Biomedical	Technical
Entity Num	28,787	6,881	3,012
Dictionary Size	2,482	931	272
Phrase Num	6,877	2,728	1,399

Table 1: The statistical results on experimental datasets. There are duplicate items in the entity set.

- **BC5CDR** (Li et al., 2016) consists of 1,500 PubMed articles, including 15,935 Chemical and 12,852 Disease mentions. It is split into three subsets: 500 each for the training, development and test sets.
- **NCBI-Disease** (Dogan et al., 2014) consists of 793 PubMed abstracts, including 6,881 Disease mentions. It is separated into three subsets: 593 for training, 100 for validation and 100 for testing.
- **LaptopReview** (Pontiki et al., 2014) consists of 3,845 review sentences, including 3,012 AspectTerm mentions. As in previous work (Giannakopoulos et al., 2017; Liu et al., 2020), it is split into three subsets: 2,445 for training, 600 for validation and 800 for testing.

Dictionary and High-Quality Phrase For a fair comparison with the previous methods, we use the same dictionary and high-quality phrase as (Shang et al., 2018b; Liu et al., 2020). Specifically, for the BC5CDR and NCBI-Disease datasets, the dictionary is a combination of both the MeSH database [1] and the CTD Chemical and Disease vocabularies [2]. For the LaptopReview dataset, the dictionary is crawled from the public website [3]. Moreover, the phrase mining tool AutoPhrase is pre-trained on a same domain text and then applied to small datasets. In the biomedical domain, it is pre-trained on the titles and abstracts of 686,568 PubMed papers. In the laptop review domain, it is pre-trained on Amazon laptop review dataset (Wang et al., 2011).

Training Details In the boundary detection and entity classification models, we use a fine-tuned BERT to encode entity context. During the fine-tuning process, we use "biobert-base-cased-v1.1" (Lee et al., 2020) and ”bert-base-cased” (Devlin et al., 2019) as our pre-trained models for biomedical and technical domain separately. We set a maximum sentence length of 256 tokens. The dimension of hidden representations is set to 768, the learning rate is set to 3e-5, the probability of dropout is set to 0.15, and the AdamW (Loshchilov and Hutter, 2019) is utilized as optimizer. The multi-layer perceptron in the entity classifier has a depth of 2 and a hidden size of 256. All above modules are trained on Nvidia Tesla V100 GPU and implemented in the PyTorch framework.

5.2 Comparing with Previous Work

Baselines We compare TEBNER with a series of NER models which report state-of-the-art results on the test datasets. There are two types of baselines methods, including supervised model (BiLSTM-CRF, ELMo-NER, BERT-NER) and distantly supervised model (Dictionary Match, SwellShark, AutoNER, HAMNER).

- **BiLSTM-CRF** (Lample et al., 2016) adopts bi-directional LSTM with character-based representations to produce token embeddings, which are fed into a CRF layer to predict token labels.
- **ELMo-NER** (Liu et al., 2020) uses pre-trained word embeddings, a character-based CNN representation, two BiLSTM layers with ELMo to train the NER model.
- **BERT-NER** adopts BERT-base model with sequence labeling framework to perform token-level prediction.
- **Dictionary Match** recognizes entities by performing string matching with given dictionary. It can be viewed as the baseline of distantly supervised model to test the improvement of other methods over the distant supervision itself.
- **SwellShark** (Fries et al., 2017) is a distantly supervised method designed for the biomedical domain. It needs regular expressions, and hand-tuning for special cases.
- **AutoNER** (Shang et al., 2018b) uses a BiLSTM network to learn connection between adjacent tokens and extracts high-quality phrases to reduce false-negative labels.

[1] https://www.nlm.nih.gov/mesh/download mesh.html
[2] http://ctdbase.org/downloads/
[3] https://www.computerhope.com/jargon.htm

Method	Description	BC5CDR			NCBI-Disease			LaptopReview		
		Pre	Rec	F1	Pre	Rec	F1	Pre	Rec	F1
BiLSTM-CRF	Gold Annotations	88.84	85.16	86.96	86.11	85.49	85.80	84.80	66.51	74.55
ELMo-NER		88.17	88.39	88.28	85.34	90.94	88.05	85.14	80.58	82.80
BERT-NER		87.24	90.75	88.96	85.90	90.10	87.95	79.26	82.42	80.81
Dictionary Match	source	**95.94**	50.82	66.44	**92.20**	49.64	64.54	**82.43**	39.45	53.36
	extended	92.57	68.86	78.97	90.07	67.08	76.90	69.54	60.40	64.65
SwellShark	Regex Design (RD)	84.98	83.49	84.23	64.7	69.7	67.1	-	-	-
	RD + Case Tuning	86.11	82.39	84.21	81.6	80.1	80.8	-	-	-
AutoNER	+ELMo	83.08	82.16	82.70	76.98	74.65	75.78	68.72	59.39	63.70
	+BERT	82.89	83.17	83.03	80.24	87.97	83.93	62.75	62.19	62.47
HAMNER	-	86.01	86.34	86.17	82.03	83.56	82.79	74.02	62.02	67.46
TEBNER		88.05	**90.36**	**89.19**	86.32	**91.35**	**88.77**	70.82	**80.89**	**75.52**

Table 2: Performance comparison of supervised and distantly supervised NER on three test datasets.

- **HAMNER** (Liu et al., 2020) is the best distantly supervised method in the past. It extends the dictionary with headword-based matching and infers the entity spans with an entity typing model.

Results The comparative results on three benchmark datasets are shown in Table 2. We observe that TEBNER achieves the best performance on all datasets. It should be emphasized that we use the same dictionary and phrases as the AutoNER and HAMNER. Due to the differences in data processing methods, our dictionary matching results are slightly different from them. Although SwellShark is designed for the biomedical domain and utilizes much more expert effort, TEBNER can easily surpasses it without human effort. Since the original AutoNER model uses all the raw texts for training (i.e., the training dataset is the union of the training, development, and test sets), Liu et al. (Liu et al., 2020) retrained the model with ELMo. To make a fair comparison with them, we use the AutoNER+ELMo (trained on the training set only) results reported in (Liu et al., 2020), which are slightly lower than original results in (Shang et al., 2018b). Moreover, we also train the AutoNER model with BERT and report the evaluation results in our paper. Compared with our proposed model that integrates multi-granularity boundary information, AutoNER only focuses on the ties between adjacent tokens and has poor performance on benchmark datasets. In particular, TEBNER outperforms the previous state-of-the-art method HAMNER by $\{3.02\%, 5.98\%, 8.06\%\}$ in terms of F1 score and surpasses the supervision model on the BC5CDR and NCBI datasets, which demonstrates the significant superiority of our proposed model.

Dictionary	BC5CDR		
	Pre	Rec	F1
Source	**97.40**	66.15	78.79
Extended (based HAMNER)	91.89	84.48	88.03
Extended (based KNN)	92.82	87.26	89.95
Extended (based TEBNER)	93.94	**95.32**	**94.63**

Table 3: Distantly supervised annotation quality on the training set.

5.3 Impact of Different Modules

To analyze the performance of different modules and investigate their impact on the final results, we also conduct experiments on following aspects.

Effectiveness of Dictionary Extension To evaluate the effectiveness of our dictionary extender, we compare three extension methods and report their distantly supervised annotation quality on the training set. As shown in Table 3, our dictionary extension method can greatly increase the entity recall while slightly reducing the precision. For example, on the BC5CDR dataset, our method significantly boosts the recall from 66.15% to 95.32% with an acceptable precision loss from 97.40% to

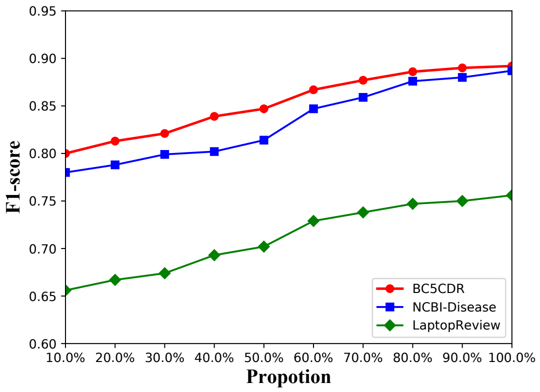

Figure 3: Influence of annotated data size on the distantly supervised model.

93.94%. Compare to the (Liu et al., 2020), our method significantly achieves 10.84% and 6.60% relative improvements on recall and F1 scores. It is worth noting that previous methods utilize complex strategies (i.e., headword matching, semantic similarity calculation, annotation weight setting) to improve the entity recall rate. Unlike them, our method mainly depends on contextual semantic information, which can be applied to any domain corpus. Moreover, we also try to extend the dictionary with a KNN model. Specifically, for each phrase, a closest entity will be recalled from the source dictionary based on the cosine similarity between the corresponding word embeddings. Then the type of the recalled entity will be assigned to the phrase. Limited by the size of the original dictionary, some entities with similar semantics but different types are easy to be recalled by KNN model, which leads to a decline in F1 score.

Influence of the number of Annotations To evaluate the robustness of our model, we study the influence of the annotated data size on the final results. Concretely, we randomly select sentences from the distant annotations and evaluate our model trained on the selected texts. From Figure 3, we can observe that increasing the size of annotations will generally improve the performance of the model, and the improvement tends to flatten out with 80% data. In particular, our model achieve 86.70% test F1 score on the BC5CDR dataset with only 60% data, which demonstrates that our TEBNER model can significantly reduce the human efforts to create NER taggers.

Model	F1(%)	
	Total	Δ
Full Model	89.19	-
– Dictionary Extension	85.30	3.89
– BERT("Break or Tie" tagging scheme)	87.77	1.42
– BERT("BIO" tagging scheme)	86.65	2.54
– Phrase Matching	86.44	2.75

Table 4: Results on the BC5CDR dataset to investigate the influence of different model components.

Ablation studies To better explore the contribution of different modules to the overall performance, we conduct the ablation studies on BC5CDR dataset. From the results shown in table 4, we can observe that: (1) Dictionary extender is a necessary component that contributes 3.89% gain of F1 to the ultimate performance, we attribute this gain to the context semantic information. (2) Removing “Break or Tie" tagging scheme degrades the performance by 1.42% F1, which shows that the connection information reflecting the interdependence between adjacent tokens is useful for NER. (3) The “BIO" tagging scheme contributes much to the overall performance, since the F1 drops by 2.54% if it is removed. (4) When we remove phrase matching result, the score drops by 2.75%, which indicates that the participation of multi-aspect statistical information is important for our model.

6 Conclusion

In this paper, we propose a new dictionary extension method and design a boundary-aware model in specific domains using distant supervision. Our dictionary extender combines phrase mining method with entity classification model, which can be easily applied to any other domain corpus. By utilizing different tagging schemes to extract candidate entities from sentence and introducing AutoPhrase tool to extract high-quality phrases from corpus, our distantly supervised NER model can detect entities from both local and global perspectives. In experiments, we evaluate our method on different domain datasets and the results demonstrate the effectiveness of our model.

Acknowledgements

This research is supported by the National Key Research and Development Program of China

(NO.2017YFC0820700) and National Natural Science Foundation of China (No.61902394). We thank all authors for their contributions and all anonymous reviewers for their constructive comments.

References

Yixin Cao, Zikun Hu, Tat-Seng Chua, Zhiyuan Liu, and Heng Ji. 2019. Low-resource name tagging learned with weakly labeled data. 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 2019*, pages 261–270.

Jason P. C. Chiu and Eric Nichols. 2016. Named entity recognition with bidirectional lstm-cnns. *Trans. Assoc. Comput. Linguistics*, 4:357–370.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019*, pages 4171–4186.

Rezarta Islamaj Dogan, Robert Leaman, and Zhiyong Lu. 2014. NCBI disease corpus: A resource for disease name recognition and concept normalization. *J. Biomed. Informatics*, pages 1–10.

Zheng Fang, Yanan Cao, Ren Li, Zhenyu Zhang, Yanbing Liu, and Shi Wang. 2020. High quality candidate generation and sequential graph attention network for entity linking. In *The Web Conference 2020, WWW 2020*, pages 640–650.

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

Zorik Gekhman, Roee Aharoni, Genady Beryozkin, Markus Freitag, and Wolfgang Macherey. 2020. Kobe: Knowledge-based machine translation evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings, EMNLP 2020*, pages 3200–3207.

Athanasios Giannakopoulos, Claudiu Musat, Andreea Hossmann, and Michael Baeriswyl. 2017. Unsupervised aspect term extraction with B-LSTM & CRF using automatically labelled datasets. In *Proceedings of the 8th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis, WASSA@EMNLP 2017*, pages 180–188.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 260–270.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinform.*, pages 1234–1240.

Belinda Z. Li, Sewon Min, Srinivasan Iyer, Yashar Mehdad, and Wen-tau Yih. 2020a. Efficient one-pass end-to-end entity linking for questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020*, pages 6433–6441.

Jiao Li, Yueping Sun, Robin J. Johnson, Daniela Sciaky, Chih-Hsuan Wei, Robert Leaman, Allan Peter Davis, Carolyn J. Mattingly, Thomas C. Wiegers, and Zhiyong Lu. 2016. Biocreative V CDR task corpus: a resource for chemical disease relation extraction. *Database J. Biol. Databases Curation*.

Xiaoya Li, Jingrong Feng, Yuxian Meng, Qinghong Han, Fei Wu, and Jiwei Li. 2020b. A unified MRC framework for named entity recognition. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020*, pages 5849–5859.

Chen Liang, Yue Yu, Haoming Jiang, Siawpeng Er, Ruijia Wang, Tuo Zhao, and Chao Zhang. 2020. BOND: bert-assisted open-domain named entity recognition with distant supervision. In *The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, SIGKDD 2020*, pages 1054–1064.

Pierre Lison, Jeremy Barnes, Aliaksandr Hubin, and Samia Touileb. 2020. Named entity recognition without labelled data: A weak supervision approach. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020*, pages 1518–1533.

Shifeng Liu, Yifang Sun, Bing Li, Wei Wang, and Xiang Zhao. 2020. HAMNER: headword amplified multi-span distantly supervised method for domain specific named entity recognition. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020*, pages 8401–8408.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019*.

Xuezhe Ma and Eduard H. Hovy. 2016. End-to-end sequence labeling via bi-directional lstm-cnns-crf. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016*.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018*, pages 2227–2237.

Maria Pontiki, Dimitris Galanis, John Pavlopoulos, Harris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. 2014. Semeval-2014 task 4: Aspect based sentiment analysis. In *Proceedings of the 8th International Workshop on Semantic Evaluation, SemEval@COLING 2014*, pages 27–35.

Jingbo Shang, Jialu Liu, Meng Jiang, Xiang Ren, Clare R. Voss, and Jiawei Han. 2018a. Automated phrase mining from massive text corpora. *IEEE Trans. Knowl. Data Eng.*, 30.

Jingbo Shang, Liyuan Liu, Xiaotao Gu, Xiang Ren, Teng Ren, and Jiawei Han. 2018b. Learning named entity tagger using domain-specific dictionary. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, EMNLP 2018*, pages 2054–2064.

Bailin Wang, Wei Lu, Yu Wang, and Hongxia Jin. 2018. A neural transition-based model for nested mention recognition. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, EMNLP 2018*, pages 1011–1017.

Hongning Wang, Yue Lu, and ChengXiang Zhai. 2011. Latent aspect rating analysis without aspect keyword supervision. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2011*, pages 618–626.

Yu Wang, Yun Li, Hanghang Tong, and Ziye Zhu. 2020. HIT: nested named entity recognition via head-tail pair and token interaction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020*, pages 6027–6036.

Mengge Xue, Bowen Yu, Zhenyu Zhang, Tingwen Liu, Yue Zhang, and Bin Wang. 2020. Coarse-to-fine pre-training for named entity recognition. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020*, pages 6345–6354.

Changmeng Zheng, Yi Cai, Jingyun Xu, Ho-fung Leung, and Guandong Xu. 2019. A boundary-aware neural model for nested named entity recognition. 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 2019*, pages 357–366.