Extract-Select: A Span Selection Framework for Nested Named Entity Recognition with Generative Adversarial Training

Peixin Huang[1], Xiang Zhao[1,2], Minghao Hu[3]*, Yang Fang[1], Xinyi Li[1] and Weidong Xiao[1,2]
[1] National University of Defense Technology, Changsha, China
[2] Collaborative Innovation Center of Geospatial Technology, Wuhan, China
[3] Information Research Center of Military Science, Beijing, China
{huangpeixin15, xiangzhao, fangyang12, wdxiao}@nudt.edu.cn
huminghao16@gmail.com

Abstract

Nested named entity recognition (Nested NER) aims to identify named entities which may overlap. Span-based approaches regard nested NER as a two-stage task—span extraction and classification, thus having the innate ability to handle this task. However, they face the problems of error propagation, ignorance of span boundary, difficulty in long entity recognition and requirement on large-scale annotated data. In this paper, we propose *Extract-Select*, a span selection framework for nested NER, to tackle these problems. Firstly, we introduce a span selection framework in which nested entities with different entity categories would be separately extracted by the extractor, thus naturally avoiding error propagation in prior two-stage approaches. In the inference phase, the trained extractor selects final results specific to the given entity category. Secondly, we propose a hybrid selection strategy in the extractor, which not only makes full use of both span boundary and span content, but also improves the ability of long entity recognition. Thirdly, we design a discriminator to evaluate the extraction result, and train both extractor and discriminator with generative adversarial training (GAT). The use of GAT greatly alleviates the stress on the dataset size. Experimental results on four benchmark datasets demonstrate that *Extract-Select* outperforms competitive nested NER models, obtaining state-of-the-art results. The proposed model also performs well with less labeled data, proving the effectiveness of GAT.

1 Introduction

Named entity recognition (NER) aims at detecting the spans and semantic categories of entities from the text. Previous studies usually treat NER as a sequential labeling problem (Ma and Hovy, 2016; Chiu and Nichols, 2016). These studies restrict each token belonging to at most one entity mention, and hence it is unable to handle nested NER (Huang et al., 2015), where one token may belong to multiple mentions. For example in Figure 1, a LOC (i.e., Location) entity "western Canadian" is nested in another GPE (i.e., Geo-Political Entity) entity "the western Canadian province of British Columbia".

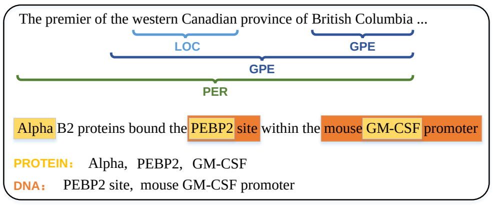
The premier of the western Canadian province of British Columbia ...
LOC
GPE
GPE
PER
Alpha B2 proteins bound the PEBP2 site within the mouse GM-CSF promoter
PROTEIN: Alpha, PEBP2, GM-CSF
DNA: PEBP2 site, mouse GM-CSF promoter

Figure 1: Examples for nested entities from ACE2005 and GENIA corpora.

Some studies seek to reconcile sequential labeling with nested NER (Alex et al., 2007; Ju et al., 2018). However, sequential labeling is naturally unsuitable for assigning multiple labels to a single token. Considering that, some studies turn to adopt the two-stage framework, including transition-based approaches (Wang et al., 2018a; Lin et al., 2019), hypergraph-based approaches (Wang and Lu, 2018; Katiyar and Cardie, 2018; Luo and Zhao, 2020) and span-based approaches (Sohrab and Miwa, 2018; Shen et al., 2021; Zhong and Chen, 2021). Among them, span-based approaches handle nested NER by extracting possible spans and classifying their categories. Although these approaches have the innate ability to cope with this task, they have the following problems: (1) Span-based approaches follow the two-stage framework, which inevitably has the problem of error propagation; (2) These approaches usually rely on span content for classification. However, span boundary information is not fully utilized, which is important for precise entity span extraction; (3) It is difficult for span-based approaches to recognize long entities because the span length in the span extraction phase is limited; (4) These approaches usually rely heavily on large size of training data for obtaining competitive results.

* Corresponding author

Findings of the Association for Computational Linguistics: ACL 2022, pages 85 - 96
May 22-27, 2022 ©2022 Association for Computational Linguistics

In this paper, we propose a novel nested NER approach, named *Extract-Select*, which is able to cope with the above shortcomings. Specifically, the proposed *Extract-Select* works as follows. Firstly, an extractor is proposed to *extract* all possible entity spans specific to a particular typed entity marker, with a novel hybrid selection strategy. Then, a discriminator is introduced to evaluate and score entity span candidates predicted by the extractor. The extractor and the discriminator are iteratively trained with generative adversarial training (GAT). In the inference phase, the iteratively trained extractor *selects* final entity spans of the given entity marker from the contexts.

Extract-Select solves the above shortcomings from the following three aspects:

- To address Problem (1), we adopt a span selection framework in *Extract-Select*, which aims to separately train the extractor for each entity category. Motivated by Zhong and Chen (2021), we design entity markers that encodes category knowledge and use it to clarify the extractor what to extract. For example in Figure 1, the nested entities "PEBP2" (type PROTEIN) and "PEBP2 site" (type DNA) would be separately extracted. As two nested entities with different categories are separately selected by the extractor in one step, the problems of error propagation in two-stage approaches can be naturally solved.
- To solve Problems (2) and (3), we design a hybrid selection strategy in our extractor. This strategy makes full use of boundary information by detecting the start and end positions of entity span, followed with span content matching. Then, the span boundary as well as content information are fully used in the training of the extractor. As this strategy detects entity spans with boundary extraction, it does not require the setting of maximum span length, thus overcoming the difficulty in long entity recognition.
- To solve Problem (4), we design a discriminator to evaluate the extractor and train the extractor and discriminator with GAT. Through min-max training, the extractor can additionally learn from the discriminator to get higher scores, meanwhile greatly reducing the demand on training data size. What is more, the well designed entity markers provide informative prior knowledge for the extractor, which also contributes to better performance with less labeled data.

To evaluate our *Extract-Select*, we conduct experiments on four standard nested NER benchmarks, including ACE04, ACE05, KBP17 and GENIA datasets. Experimental results show that our model can effectively detect nested entities and achieve state-of-the-art results on the above four datasets. The ablation study on entity marker, GAT and hybrid selection strategy reveals that these components are indispensable and all of them contribute to our model. Moreover, our model only requires half amount of labeled data to achieve the same performance as baselines, indicting the effectiveness of our approach with less training data.

2 Related Work

Sequential labeling-based approaches solve the nested NER by designing suitable labeling schema. Shibuya and Hovy (2020) provide a second-best path decoding method to iteratively find nested entities. Straková et al. (2019) propose a linearized encoding scheme to model multiple named entity labels. Wang et al. (2020) design a pyramid framework to identify nested entities. Sequential labeling approaches is naturally unsuitable for nested NER.

Transition-based approaches model nested structure through state transition and construct nested entities through actions. Wang et al. (2018a) introduce a scalable transition-based model. Lin et al. (2019) propose an Anchor-Region architecture which models the head-driven phrase structures. However, these approaches rely heavily on hand-crafted features.

Hypergraph-based approaches construct hypergraphs by the structure of nested NER and decode results on hypergraphs. Muis and Lu (2017) introduce a mention hypergraph for nested NER. Wang and Lu (2018) propose a hypergraph representation, which is free from structural ambiguity. Luo and Zhao (2020) propose to capture bidirectional information interactions between hypergraph layers. However, these hypergraphs should be well designed to prevent ambiguous structure.

Span-based approaches extract entity spans and then classify their categories. Luan et al. (2019) select the most confident entity spans for classification. Fisher and Vlachos (2019) propose to merge entities and tokens into entities, and then assign labels. Shen et al. (2021) regards this task as an object detection task, locating and then labeling spans. Nevertheless, these two-stage approaches have the problem of ignorance of span boundary, difficulty in long entity recognition and error propagation.

Li et al. (2020b) formalizes NER as a machine reading comprehension (MRC) task, which uses

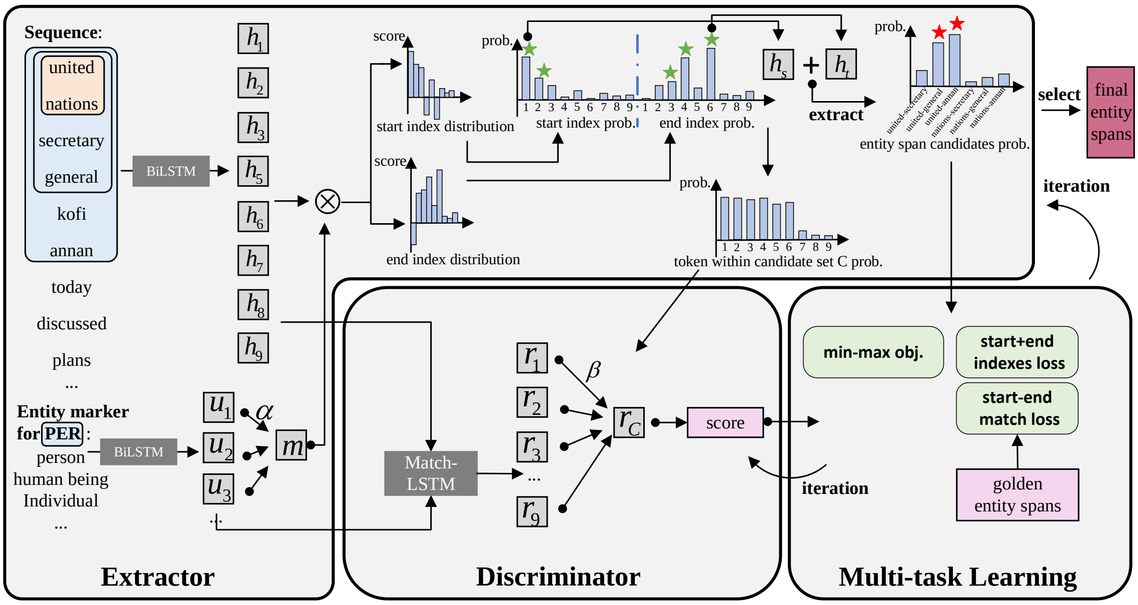

Figure 2: Overview of *Extract-Select*. It follows a span selection framework and contains an extractor which adopts a hybrid selection strategy to extract entity span candidates and a discriminator which aims to score the extractor. The extractor and discriminator are trained with multi-task learning including span boundary extraction and generative adversarial training.

BERT as backbone and extracts spans of given queries. However, this work relies heavily on training data size. Besides, it extracts entity spans based on boundary information but ignores content information. In contrast to their work, we use GAT to iteratively train extractor to get better results, and adopt hybrid selection strategy to make full use of both boundary and content information.

Generative adversarial training (GAT) gives a way to learn deep representations without extensively labeled data. It is proposed by Goodfellow et al. (2014) and is characterized by training a generator and a discriminator in competition with each other. GAT has been applied in different NLP subtasks, including dialogue generation (Li et al., 2017) and relation extraction (Qin et al., 2018). In these studies, GAT proves to be effective in reducing the usage of training data. Motivated by these work, we propose to apply GAT in NER task to reduce the demand on labeled data.

3 Problem Definition

The input of the span selection framework is a sequence $X = \{x_1, x_2, ..., x_{|X|}\}$, where $|X|$ denotes the length of the sequence. The possible entity span $x_{s,e} = \{x_s, x_{s+1}, ..., x_{e-1}, x_e\}$ is a continuous sub-string of X satisfying $s \leq e$. Let Y denote the predefined list of all entity categories and y^* be the entity marker specific to type $y \in Y$ (e.g., "LOC" has an entity marker "location"). The aim of span selection framework is to find all entities in X for each category y.

4 Extract-Select: Nested NER with GAT

4.1 Overview

In this section, we introduce *Extract-Select* in detail. As shown in Figure 2, *Extract-Select* consists of two main components: an extractor and a discriminator. In particular, given the input sequence X and the entity marker y^*, the extractor adopts a hybrid selection strategy to extract the entity span candidate set C and calculates its representation p_C, i.e., $(C, r_C) = f_E(y^*, X)$. Afterwards, the discriminator is fed with r_C to evaluate the correctness of C, i.e., $score = f_D(y^*, X, r_C)$. After iterative training of both extractor and discriminator, the extractor selects the final result (a set of entity spans), i.e., $final\ result = f_{E_{final}}(y^*, X)$.

4.2 Extractor

Given the entity type y and the input sequence $X = \{x_j\}_{j=1}^{|X|}$, the extractor aims to extract the entity span candidate set $C = \{C_1, C_2, ...C_i\}$ specific to y^* from X. Then, the extractor needs to compute a continuous latent variable p_C to represent C.

1) Sequence representation: We first represent all tokens $\{x_i\}_{i=1}^{|X|}$ in the input sentence X as a sequence of embeddings $\{w_i\}_{i=1}^{|X|}$. Each embedding w_i is the concatenation of character embedding, word embedding, contextualized word embedding and part-of-speech (POS) embedding. The character embedding is generated by a bi-directional

LSTM (BiLSTM) module with the same setting as Ju et al. (2018). For the contextualized word embedding, we obtain the context-dependent embedding for a target token with one surrounding sentence on one side. Then, the concatenation is fed into another BiLSTM to obtain the final token representation:

$$\{h_i\}_{i=1}^{|X|} = \text{BiLSTM}(\{w_i\}_{i=1}^{|X|}), \tag{1}$$

where h_i is the hidden state.

2) Entity marker representation: Entity category is an important prior knowledge as it makes the extractor know what to extract, and its effectiveness has been demonstrated in prior work (Li et al., 2020b). Besides, the usage of entity marker can avoid the error propagation issue in two-stage framework. Therefore, we propose to design an entity marker for each category, where the entity marker is its fine-grained explanation and would be used as the input of the extractor. We experiment on different types of entity marker and finally choose the combination of *Keywords* and *Synonyms*. Specifically, *Keywords* mean that entity markers are keywords describing entity type, e.g., the entity marker for type ORG is "organization", and *Synonyms* mean that entity markers are words or phrases which mean nearly the same as terms extracted from the Oxford Dictionary, e.g., the entity marker for type ORG is "institution body group company firm business corporation".

We concatenate word embeddings of *Keywords* and *Synonyms*, and feed embeddings $\{w_i\}_{i=1}^{|y^*|}$ into a BiLSTM to obtain $\{u_i\}_{i=1}^{|y^*|}$ = Bi-LSTM$(\{w_i\}_{i=1}^{|y^*|})$, where $|y^*|$ is the length of entity marker. Then, we use self attention to integrate the entity marker information:

$$\alpha_i = \frac{exp(W_a u_i)}{\sum_k exp(W_a u_k)}, \tag{2}$$

$$m = \textstyle\sum_{i=1}^{y^*} \alpha_i u_i, \tag{3}$$

where α_i is the attention weight of u_i and $W_a \in \mathbb{R}^d$ is a learned weight vector.

3) Hybrid selection: Prior span-based approaches extract spans by predicting whether each token is within the entity span with n two-class classifier, which only considers the span content. These methods need to set maximum span length parameter to avoid high computational costs. As a result, it is hard for them to identify long entities. What is more, they ignore the span boundary which is also important for entity recognition.

In view of this, we propose a hybrid selection strategy, which makes full use of both boundary and content of span. It first predicts the probability of tokens being the boundary of entity spans, and produce the entity span candidate set. Later, it uses the content of span candidates, i.e., calculates content representation of candidates, to enable the training of extractor and the golden entity spans could be selected eventually.

Given the representations of sequence and entity marker, the extractor first predicts the probabilities of token i being the boundary (i.e., start and end index) of entity spans:

$$\begin{aligned} p_s(i|y,X) &= \frac{exp(h_i W_s m)}{\sum_k exp(h_k W_s m)}, \\ p_e(i|y,X) &= \frac{exp(h_i W_e m)}{\sum_k exp(h_k W_e m)}, \end{aligned} \tag{4}$$

where $W_s, W_e \in \mathbb{R}^{d\times d}$ are learnable parameters.

In the input sequence X, there may be multiple entities of a particular category. This means that multiple start and end indexes could be predicted. To match them, we first get the indexes that might be the starting or ending positions:

$$\begin{aligned} I_s &= \{i|\arg\max(p_s(i|y,X)) = 1, i=1,..,n\}, \\ I_e &= \{j|\arg\max(p_e(j|y,X)) = 1, j=1,..,n\}. \end{aligned} \tag{5}$$

For any given start index $i_s \in I_s$ and end index $j_e \in I_e$ $(i_s \le j_e)$, we calculate the probability of entity span candidate:

$$p_{s,e}(i_s, j_e|y,X) = sigmoid(W_C \cdot concat(h_{i_s}, h_{j_e})), \tag{6}$$

where $W_C \in \mathbb{R}^{1\times 2d}$ is a learned parameter. The entity span candidate C_i is added into the candidate set C if its span probability $p_{s,e}$ is larger than a pre-defined threshold.

Then the content of candidate set C is utilized. We calculate $p_C(i|y,X)$ as the probability of the ith token appearing in C, which can be considered as the probability of the ith token within the span candidates. Specifically, $p_C(i|y,X)$ can be calculated through p_s and p_e:

$$p_C(i|y,X) = \sum_{s=1}^{i} \sum_{e=i}^{|X|} p_s(s|y,X) p_e(e|y,X), \tag{7}$$

where $p_C(i|y,X)$ also means the frequency of the ith token appearing in C. In other words, the more frequent the ith token appears in C, the higher $p_C(i|y,X)$ would be. In this way, the content information p_C can be used in candidate set scoring process to enable the training of extractor.

Finally, with both the boundary and content information, the extractor could be well trained to select the final golden entity span. This strategy does not set the maximum span length. As a result, long entity span can also be recognized.

4.3 Discriminator

After the extractor has obtained the entity span candidate set C, the discriminator aims to evaluate and score C, so as to train the extractor with GAT. This process consists of two steps as follows.

1) Candidate set representation: To represent the entity span candidate set C, we propose to encode the information of both entity type y and sequence X into p_C. Specifically, we build entity-aware sequence representation with Match-LSTM (Wang et al., 2018b), by matching the entity marker up with the sequence:

$$\{r_i\}_{i=1}^{|X|} = \text{Match-LSTM}(\{h_i\}_{i=1}^{|X|}, \{u_i\}_{i=1}^{|y^*|}). \tag{8}$$

Then the representation of entity span candidate set r_C is calculated with r_i and the probability $p_C(i|y,X)$ through weighted sum:

$$\begin{aligned} \beta_i &= \frac{p_C(i|y,X)}{\Sigma_{k=1}^{|X|} p_C(k|y,X)}, \\ r_C &= \Sigma_{k=1}^{|X|} \beta_k r_k. \end{aligned} \tag{9}$$

2) Candidate set scoring: The score f_D of entity span candidate set C can be calculated:

$$f_D(C) = sigmoid(W_D r_C), \tag{10}$$

where $W_D \in \mathbb{R}^d$ is a learned weight vector and the score $f_D \in [0,1]$ would be used to iteratively train the extractor to get higher span candidate set score from the discriminator through min-max training in the GAT process.

4.4 Multi-task Learning with GAT

In the training process, we train the extractor by multi-task learning, and train the extractor and discriminator together with GAT.

The first task is to train the extractor by minimizing the negative log probabilities of the true start and end indexes of the golden entity span:

$$\ell_E^{boundary} = -\log p_s(s|y,X) - \log p_e(e|y,X), \tag{11}$$

where s and e denote the start and end indexes of the golden entity in the sequence X.

The second is to train the extractor by minimizing the start-end index matching loss:

$$\ell_E^{span} = -\log p_{s,e}. \tag{12}$$

The third is to train the extractor and discriminator together with GAT. We train the extractor to obtain a higher score from the discriminator:

$$\ell_E^{GAT} = \log(1 - f_D(y,X,p_C)). \tag{13}$$

Meanwhile, we train the discriminator to maximize $\log f_D(y,X,p_C)$ and minimize $\log(1 - f_D(y,X,p_C))$:

$$\ell_D^{GAT} = z \log f_D(C) + (1-z)\log(1-f_D(C)), \tag{14}$$

where $z \in \{1,0\}$ denotes whether the golden entity appears in the entity span candidate set or not.

The overall training objective ℓ_E of the extractor is defined as follows:

$$\ell_E = \gamma_1 \ell_E^{boundary} + \gamma_2 \ell_E^{span} + (1-\gamma_1-\gamma_2)\ell_E^{GAT}, \tag{15}$$

where $\gamma_1,\gamma_2 \in [0,1]$ are learnable hyper-parameters to control the contributions towards the overall training objective.

In each training iteration, we use the extractor to select the new entity span candidate set through Eq.(4)-(6). The new entity span candidate set would then be scored by the discriminator. Such training procedure will be conducted iteratively so that the extractor can select spans having high score from discriminator. In the inference phase, the trained extractor would first select the start and end indexes and then match the start indexes with end indexes, getting the final results. The pseudo-code of the training procedure is given in Appendix A.

Our model differs from *BERT-MRC* in the following ways: (1) Different from *BERT-MRC* which uses bert-based machine reading comprehension model as the backbone, we design an *Extract-Select* model which iteratively trains the extractor to select the golden entity spans specific to the entity marker. (2) We propose a hybrid selection strategy for better entity span selection. This strategy makes full use of both boundary and content information of the span. However, *BERT-MRC* only conducts entity decoding based on the span boundary. (3) We propose to incorporate GAT in our model to train the extractor to learn additional information from the discriminator, meanwhile greatly reducing the demand on training data size. (4) We use easily-obtained entity markers to achieve competitive performance, avoiding complex query designing.

5 Experimental Setup

5.1 Benchmarks and Evaluations

We evaluate *Extract-Select* on four NER benchmarks — ACE2004[1], ACE2005[2], GENIA[3], and KBP2017[4]. Please refer to Appendix B.1 for the statistics and the detailed processing of the datasets, and refer to Appendix B.2 for implementation details.

[1] https://catalog.ldc.upenn.edu/LDC2005T09

[2] https://catalog.ldc.upenn.edu/LDC2006T06

[3] http://www.geniaproject.org/genia-corpus

[4] https://catalog.ldc.upenn.edu/LDC2017D55

Model	ACE2004			ACE2005			GENIA			KBP2017		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Transition[5]	74.9	71.8	73.3	74.5	71.5	73.0	78.0	70.2	73.9	74.7	67.0	70.1
Seg-Graph	78.0	72.4	75.1	76.8	72.3	74.5	77.0	73.3	75.1	79.2	66.5	72.3
Merge-Label	-	-	-	75.1	74.1	74.6	-	-	-	-	-	-
ARN	-	-	-	76.2	73.6	74.9	75.8	73.9	74.8	77.7	71.8	74.6
Second-Path	83.73	81.91	82.81	82.98	82.42	82.70	78.07	76.45	77.25	-	-	-
Seq2seq[6]	-	-	84.33	82.58	84.29	83.42	79.92	76.55	78.20	-	-	-
BiFlat-Graph	-	-	-	75.0	75.2	75.1	77.4	74.6	76.0	77.1	74.3	75.6
Pyramid	86.08	86.48	86.28	83.95	85.39	84.66	79.45	78.94	79.19	-	-	-
BERT-MRC	85.05	86.32	85.98	87.16	86.59	86.88	85.18	81.12	83.75	82.33	77.61	80.97
Locate-Label	87.44	87.38	87.41	86.09	87.27	86.67	80.19	80.89	80.54	85.46	82.67	84.05
Extract-Select	**88.26**	**88.53**	**88.39**	**87.15**	**88.37**	**87.76**	**83.64**	**84.41**	**84.02**	83.76	**85.87**	**84.80**

Table 1: Results for nested NER tasks. Bold indicates the best scores

As for the evaluation metrics, we use strict evaluation that an entity is considered correct when both span and category are correctly predicted. We use span-level micro-averaged Precision (P), Recall (R) and F1 scores (F1) for evaluation.

5.2 Baselines

We choose the following models as baselines.

Sequential labeling-based models. *Second-Path* (Shibuya and Hovy, 2020) regards the tag sequence as a path and searches for results with the second-best path decoding. *Seq2seq* (Straková et al., 2019) views the nested NER as a sequence-to-sequence problem. *Pyramid* (Wang et al., 2020) is based on BERT and decodes nested mentions by its length in a bottom-up manner.

Transition-based models. *Transition* (Wang et al., 2018a) constructs forests for nested mentions through an action sequence. *ARN* (Lin et al., 2019) builds the Anchor-Region networks by using the head-driven structures of nested entities.

Hypergraph-based models. *Seg-Graph* (Wang and Lu, 2018) utilizes a segmental hypergraph representation for the modeling of nested mentions. *BiFlat-Graph* (Luo and Zhao, 2020) constructs a hypergraph module and uses the representation of it to improve inner entity predictions.

Span-based models. *Merge-Label* (Fisher and Vlachos, 2019) first merges tokens and entities to form nested structures and then labels them. *Locate-Label* (Shen et al., 2021) is based on BERT and generates span proposals by filtering and doing regression on seed spans. *BERT-MRC* Li et al. (2020b) formulates NER as a MRC task.

6 Results and Discussions

6.1 Overall Evaluation

Table 1 presents the performance of *Extract-Select* as well as the above baselines on four datasets. From the table, we observe that: (1) *Extract-Select* can effectively deal with nested NER, achieving the state-of-the-art performance. Specifically, *Extract-Select* gains at least 0.98%, 0.88%, 0.27%, 0.75% F1 scores improvements on ACE2004, ACE2005, GENIA and KBP2017, respectively. This verifies the effectiveness of our span selection architecture. (2) *Extract-Select* brings much higher recall value improvements than other methods, especially on KBP2017 and GENIA datasets. We notice that KBP2017 and GENIA contain much more entities than the other two datasets and the number of entities on test set of KBP2017 is over four times more than that of ACE2005. *Extract-Select* has significant advantages on such dataset, proving the effectiveness of GAT. (3) Compared with most of the baselines, *Extract-Select* can well balance precision and recall, maintaining precision value with high recall improvement. The reason may be that entity markers highlight the category information, clarifying which to extract. (4) With conventional word embeddings, *Extract-Select* method performs better that those BERT-based models (e.g., *Locate-Label* and *Pyramid*), which further proves the advantage of span selection framework. We also evaluate our model on two flat NER datasets, as shown in Appendix C.

6.2 Ablation Study

We then conduct ablation study to elucidate the effectiveness of main components of our *Extract-Select* method. Likewise, we only present the results on ACE2005. We compare *Extract-Select* with the following three internal baselines:

w/o EntityMarker: To verify the effectiveness of entity marker, this variation removes entity marker representations and only uses the index (i.e., "one", "two", *et al.*) of entity category for span selection.

w/o GAT: To evaluate the effectiveness of GAT, this variation only retains the extractor. Extractor

[5] *Transition* (Wang et al., 2018a) did not report precision and recall scores. Instead, Wang and Lu (2018) reported these scores for *Transition*.

[6] *Seq2seq* (Straková et al., 2019) did not report precision and recall scores. We use the reported F1 scores in this article.

Model	ACE2005		
	P	R	F1
w/o EntityMarker	85.82	86.03	85.92
w/o GAT	85.75	85.12	85.43
w/o HybridSelect	84.96	87.21	86.07
Extract-Select	**87.15**	**88.37**	**87.76**

Table 2: Results of internal baselines on the test set of ACE2005.

is trained by Eq.(11) and Eq.(12):

$$\ell_E = \lambda \ell_E^{boundary} + (1-\lambda)\ell_E^{span}, \tag{16}$$

where $\lambda \in [0,1]$ is a hyper-parameter that controls the weights of two tasks.

w/o HybridSelect: To verify the effectiveness of hybrid selection strategy, this variation only considers the boundary information and leaves out the content. Specifically, we still calculate the probabilities of tokens being start and end indexes of entity spans, but use the boundaries of entity candidates to represent the candidate set. Thus, Eq.(9) can be rewritten as follows: $r_{C_j} = r_{j_s} + r_{j_e}$, where j_s and j_e are the start and end indexes of the j^{th} entity span candidate C_j, obtained through Eq.(5). And $r_{j_{s(e)}}$ is obtained through Eq.(8).

In the training process, the extractor is trained with the policy gradient method. Thus, the training objective of the extractor in Eq.(13) can be modified as follows:

$$\nabla \ell_E^{GAT} \approx -\sum_{j=1}^{k} [f_D(y,X,C_j) \nabla(log p_s(j_s|y,X)) \\ + log p_e(j_e|y,X))], \tag{17}$$

Results are shown in Table 2. From the table, we find that: (1) *Extract-Select* outperforms three internal baselines on the test set of ACE2005. Compared with *w/o GAT*, the F1 scores of full model improve by up to 2.33%, which means it is useful to introduce the discriminator to train the extractor through min-max training. (2) *w/o HybridSelect* suffers from much more precision decrease than recall compared to full model. The reason may be that the policy gradient adopted in *w/o HybridSelect* produces noise when sampling the span candidates, whereas ours can avoid such noise by training the extractor using back-propagation. Such intuition reveals the effectiveness of our hybrid selection strategy in enabling the extractor to be trained by back-propagation from the discriminator. (3) Experimental results also demonstrate that entity markers are effective. This allows the model to take advantage of the prior knowledge of categories, improving the F1 score by 1.84% on ACE2005. (4) *w/o EntityMarker* shows significant or comparable performance improvements compared to the baselines presented in Table 1. This validates the effectiveness of our span selection framework.

Setting	**ACE2005** (F1)
Keywords	87.12
Synonyms	87.34
Wikipedia	86.71
w/o EntityMarker	85.92
Extract-Select	**87.76**

Table 3: Results of the model with different entity markers on ACE2005 dataset.

6.3 Analysis of Entity Marker

To explore the influence of using different types of entity marker in *Extract-Select*, we investigate the performance of our model with different entity marker settings. Three experimental settings are considered: *Keywords*, *Synonyms*, *Wikipedia*. *Keywords* means the entity marker is the keyword describing the category, whereas *Synonyms* represents entity markers as synonymous words of keywords that are extracted from the Oxford Dictionary. *Wikipedia* means entity markers are constructed using the Wikipedia definition. For example, the entity marker for type ORG is “an entity comprising multiple people, such as an institution or an association”. Besides, we also include *w/o EntityMarker*, for which entity markers are replaced with the position index of the category.

The results of our model with different input entity markers on ACE2005 are presented in Table 3. From the table, we find that our *Extract-Select* (with Keywords+Synonyms as entity marker) achieves the highest F1 scores. In all settings, *w/o EntityMarker* that do not contain any entity information underperforms the others, indicting that meaningful prior knowledge contributes to superior performance. We also observe that *Wikipedia* underperforms *Keywords* and *Synonyms*. The reason may be that descriptive words from *Wikipedia* may not precisely describe entity categories compared to other settings.

6.4 Analysis of Training Data

Since entity markers encode useful prior knowledge and the min-max training also learns from unlabeled data, we expect that the proposed model works better with less training data. We test our model, *w/o EntityMarker* and *w/o GAT* on randomly sub-sampled labeled data of ACE2005 training set. As shown in Figure 3, the performance of three models drops with the decline of training sample size. However, our full model only requires half amount of training data to achieve comparable performance with two internal base-

Sentence 1	The US Supreme Court will hear arguments from both sides on Friday and Florida ’ s Leon County Circuit Court will consider the arguments on disputed state ballots on Saturday .

Gold Label	ORG:{The US Supreme Court, both sides, Florida ’ s Leon County Circuit Court};
	GPE:{Florida, Leon County, state}

Locate-Label	ORG:{The US Supreme Court, Florida ’ s Leon County Circuit Court}; GPE:{US, Florida, Leon County, state};PER: {both sides}

Extract-Select	ORG:{The US Supreme Court, both sides, Florida ’ s Leon County Circuit Court};
	GPE:{US, Florida, Leon County, state}

Sentence 2	Separatists have fought since 1975 for independence in Aceh , which is rich in oil and gas and has a population of about 4 . 1 million people .
Gold Label	PER:{Separatists, a population of about 4 . 1 million people}; GEP:{which, Aceh , which is rich in oil and gas and has a population of about 4 . 1 million people}
Locate-Label	PER:{Separatists, about 4 . 1 million people, a population of about 4 . 1 million people} ; GEP: {which, Aceh , which is rich in oil and gas and has a population of about 4 . 1 million peoople}
Extract-Select	PER:{Separatists, a population of about 4 . 1 million people}; GEP:{which, Aceh , which is rich in oil and gas and has a population of about 4 . 1 million people}
Sentence 3	united nations secretary general kofi annan today discussed plans for the summit with the host , egyptian president hosni mubarak .
Gold Label	ORG:{united nations}; PER:{united nations secretary general, united nations secretary general kofi annan, the host, egyptian president, egyptian president hosni mubarak}; GPE:{egyptian}
Locate-Label	ORG:{united nations}; PER:{united nations secretary general, united nations secretary general kofi annan, the host, egyptian president, egyptian president hosni mubarak}; GPE:{egyptian}
Extract-Select	ORG:{united nations}; PER:{united nations secretary general, united nations secretary general kofi annan, secretary general kofi annan, the host, egyptian president, egyptian president hosni mubarak} ; GPE:{egyptian}

Table 4: Examples of predicted results of our model and Locate-Label. Blue highlights indicate wrong predictions by Locate-Label, red highlights indicate wrong predictions by our model, colored words indicate wrongly predicted entity references.

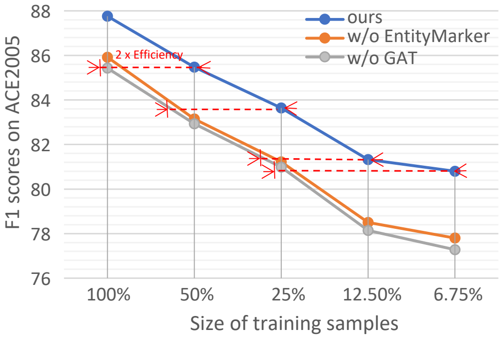

Figure 3: Effect of varying training sample size on ACE2005.

Model	ACE2004 (F1)			
	$1 \leq L \leq 5$	$5 < L \leq 10$	$L > 10$	ALL
support	2719	219	97	3035
Locate-Label-reg	88.43	66.12	37.11	85.18
Locate-Label	88.55	82.78	61.72	87.41
Extract-Select	**89.52**	**84.06**	**66.20**	**88.39**

Table 5: A comparison of recognition F1 score on entities of different lengths, we divide the entities into three groups: $1 \leq L \leq 5$, $5 < L \leq 10$, and $L > 10$, where L denotes entity length. Support denotes the number of entities in each length group on the test set of ACE2004.

lines (removing the entity marker and the GAT, respectively). Besides, the observation from the vertical line reviews that the degree of performance decrease of the full model is much less than that of other two internal baselines. In sum, the above observations indicate that both the entity marker and the GAT contributes to the better performance when less training data is given.

6.5 Analysis of Long Entity Recognition

To illustrate the performance of model on entities of different lengths, we divide the entities into three groups according to their lengths. We compare *Extract-Select* with two-stage models: *Locate-Label* (Shen et al., 2021) which adopts boundary regressors to enable long entity recognition and *Locate-Label-reg* which is a two-stage baseline. The results are shown in Table 5. We notice that the F1 score of *Locate-Label-reg* has a sharp decrease for long entities ($L > 10$) by 29.09% compared to our model. This may because *Locate-Label-reg* set maximum span length in span extraction, limiting the ability of recognizing long entities. *Locate-Label* faces a large F1 score decrease (5.77% and 21.06%) when the entity length increases from $1 \leq L \leq 5$ to $5 < L \leq 10$ and from $5 < L \leq 10$ to $L > 10$, respectively. Compared with them, *Extract-Select* maintains a good performance when the entity length increases, with only 5.46% and 17.86% F1 score decrease. This verifies that our model is more effective in recognizing long entities.

6.6 Case Study

Examples of predictions are shown in Table 4. The first part illustrates that *Extract-Select* has the ability of resolving ambiguous entity references, as span selection framework separately extracts entities for each category rather than conducts multi-classification for every entity. As shown in the fourth line, *Extract-Select* accurately recognizes the reference phrase “both sides” as ORG category, whereas *Locate-Label* incorrectly classifies it into PER category due to the ambiguity. The second part reveals that *Extract-Select* can recognize long entities well. As shown in the second part, the long entities “Aceh , which is rich in oil and gas and has a population of about 4 . 1 million people” of GEP category can be extracted. However, this framework may also incorrectly recall some entities, especially for entities with multi-level nested structures. For example in the third part, the multi-level nested entities “secretary general kofi annan” is incorrectly recognized. The reason may be that the extracted multi-level nested entities confuses the discriminator.

7 Conclusion

This paper proposes *Extract-Select*, a span selection framework to solve nested NER. It contains an extractor which aims to extract entities specific to a particular entity category with a hybrid selection strategy, and a discriminator scoring the extractor. The extractor and discriminator are trained with GAT to reduce the demand on labeled data. Comprehensive experiments performed on four widely used nested NER datasets demonstrate the superiority of *Extract-Select*. In future, we will (1) attempt to overcome the deficiency issue and (2) investigate in discontinuous and joint NER.

Acknowledgements

This work was partially supported by NSFC under grants Nos. 61872446, 62002373 and 62006243, and the Science and Technology Innovation Program of Hunan Province under grant No. 2020RC4046.

References

Beatrice Alex, Barry Haddow, and Claire Grover. 2007. Recognising nested named entities in biomedical text. In *Biological, translational, and clinical language processing, BioNLP@ACL 2007, Prague, Czech Republic, June 29, 2007*, pages 65–72. Association for Computational Linguistics.

Billy Chiu, Gamal K. O. Crichton, Anna Korhonen, and Sampo Pyysalo. 2016. How to train good word embeddings for biomedical NLP. In *Proceedings of the 15th Workshop on Biomedical Natural Language Processing, BioNLP@ACL 2016, Berlin, Germany, August 12, 2016*, pages 166–174. Association for Computational Linguistics.

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, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

George R. Doddington, Alexis Mitchell, Mark A. Przybocki, Lance A. Ramshaw, Stephanie M. Strassel, and Ralph M. Weischedel. 2004. The automatic content extraction (ACE) program - tasks, data, and evaluation. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation, LREC 2004, May 26-28, 2004, Lisbon, Portugal*. European Language Resources Association.

Joseph Fisher and Andreas Vlachos. 2019. Merge and label: A novel neural network architecture for nested NER. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 5840–5850. Association for Computational Linguistics.

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 2672–2680.

Dou Hu and Lingwei Wei. 2020. SLK-NER: exploiting second-order lexicon knowledge for chinese NER. In *The 32nd International Conference on Software Engineering and Knowledge Engineering, SEKE 2020, KSIR Virtual Conference Center, USA, July 9-19, 2020*, pages 413–417. KSI Research Inc.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional LSTM-CRF models for sequence tagging. *CoRR*, abs/1508.01991.

Meizhi Ju, Makoto Miwa, and Sophia Ananiadou. 2018. A neural layered model for nested named entity recognition. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 1446–1459. Association for Computational Linguistics.

Arzoo Katiyar and Claire Cardie. 2018. Nested named entity recognition revisited. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 861–871. Association for Computational Linguistics.

Jiwei Li, Will Monroe, Tianlin Shi, Sébastien Jean, Alan Ritter, and Dan Jurafsky. 2017. Adversarial learning for neural dialogue generation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 2157–2169. Association for Computational Linguistics.

Xiaonan Li, Hang Yan, Xipeng Qiu, and Xuanjing Huang. 2020a. FLAT: chinese NER using flat-lattice transformer. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 6836–6842. Association for Computational Linguistics.

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, Online, July 5-10, 2020*, pages 5849–5859. Association for Computational Linguistics.

Hongyu Lin, Yaojie Lu, Xianpei Han, and Le Sun. 2019. Sequence-to-nuggets: Nested entity mention detection via anchor-region networks. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 5182–5192. Association for Computational Linguistics.

Yi Luan, Dave Wadden, Luheng He, Amy Shah, Mari Ostendorf, and Hannaneh Hajishirzi. 2019. A general framework for information extraction using dynamic span graphs. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 3036–3046. Association for Computational Linguistics.

Ying Luo and Hai Zhao. 2020. Bipartite flat-graph network for nested named entity recognition. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 6408–6418. Association for Computational Linguistics.

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, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics.

Yuxian Meng, Wei Wu, Fei Wang, Xiaoya Li, Ping Nie, Fan Yin, Muyu Li, Qinghong Han, Xiaofei Sun, and Jiwei Li. 2019. Glyce: Glyph-vectors for chinese character representations. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 2742–2753.

Aldrian Obaja Muis and Wei Lu. 2017. Labeling gaps between words: Recognizing overlapping mentions with mention separators. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 2608–2618. Association for Computational Linguistics.

Ohta, Tomoko, Tateisi, Yuka, Kim, and JinDong. 2002. The genia corpus: an annotated research abstract corpus in molecular biology domain.

Nanyun Peng and Mark Dredze. 2015. Named entity recognition for chinese social media with jointly trained embeddings. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pages 548–554. The Association for Computational Linguistics.

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, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 2227–2237. Association for Computational Linguistics.

Pengda Qin, Weiran Xu, and William Yang Wang. 2018. DSGAN: generative adversarial training for distant supervision relation extraction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 496–505. Association for Computational Linguistics.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning, CoNLL 2003, Held in cooperation with HLT-NAACL 2003, Edmonton, Canada, May 31 - June 1, 2003*, pages 142–147. ACL.

Yongliang Shen, Xinyin Ma, Zeqi Tan, Shuai Zhang, Wen Wang, and Weiming Lu. 2021. Locate and label: A two-stage identifier for nested named entity recognition. In *Proceedings of the 59th Annual*

Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021, pages 2782–2794. Association for Computational Linguistics.

Takashi Shibuya and Eduard H. Hovy. 2020. Nested named entity recognition via second-best sequence learning and decoding. *Trans. Assoc. Comput. Linguistics*, 8:605–620.

Mohammad Golam Sohrab and Makoto Miwa. 2018. Deep exhaustive model for nested named entity recognition. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 2843–2849. Association for Computational Linguistics.

Jana Straková, Milan Straka, and Jan Hajic. 2019. Neural architectures for nested NER through linearization. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 5326–5331. Association for Computational Linguistics.

Christopher Walker, Stephanie Strassel, Julie Medero, and Kazuaki Maeda. 2006. Ace 2005 multilingual training corpus ldc2006t06. *Web Download. Philadelphia: Linguistic Data Consortium*, 110(110):261–276.

Bailin Wang and Wei Lu. 2018. Neural segmental hypergraphs for overlapping mention recognition. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 204–214. Association for Computational Linguistics.

Bailin Wang, Wei Lu, Yu Wang, and Hongxia Jin. 2018a. A neural transition-based model for nested mention recognition. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 1011–1017. Association for Computational Linguistics.

Jue Wang, Lidan Shou, Ke Chen, and Gang Chen. 2020. Pyramid: A layered model for nested named entity recognition. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 5918–5928. Association for Computational Linguistics.

Shuohang Wang, Mo Yu, Xiaoxiao Guo, Zhiguo Wang, Tim Klinger, Wei Zhang, Shiyu Chang, Gerry Tesauro, Bowen Zhou, and Jing Jiang. 2018b. R^3: Reinforced ranker-reader for open-domain question answering. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 5981–5988. AAAI Press.

Zexuan Zhong and Danqi Chen. 2021. A frustratingly easy approach for entity and relation extraction. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 50–61. Association for Computational Linguistics.

A The Training Procedure of Extract-Select

The full pseudo-code for the learning procedure of *Extract-Select* is given in Algorithm 1.

Algorithm 1 The learning procedure of *Extract-Select*.

```
Require: An extractor E; A discriminator G; the input sequence $S$; an entity type $y \in Y$
Output: Trained extractor with multi-task leaning
Initialize E, D parameters;
Generate entity span candidate set $C$ using E for training D;
Pre-train D via min-max training by Eq.(14);
repeat
    for E-step do
        Extract a set and its representation $(C, r_C) = f_E(y, S)$ using Eq.(1)-(9);
        Compute the score of $C$ using Eq.(10);
        Compute joint objective $\ell_E$ using Eq.(11)-(13);
        Update Extractor parameters via policy gradient;
    end for
    for D-step do
        Use current E to generate entity span candidate set $C$;
        Represent $C$ and calculate its score $f_D(y, S)$ using Eq.(10);
        Train discriminator for k epochs by Eq.(14);
    end for
until Extract-Select converges
```

B Experiments on Nested NER

B.1 Datasets Processing

Dataset statistics are listed in Table 6. For ACE2004 (Doddington et al., 2004) and ACE2005 (Walker et al., 2006), we follow the same settings as Lin et al. (2019), and splitting files into training, development and test sets

	ACE2004			**ACE2005**			**GENIA**			**KBP2017**		
Model	Train	Dev	Test	Train	Dev	Test	Train	dev	Test	Train	Dev	Test
NO. sentences	6200	745	812	7194	969	1047	15022	1669	1855	10546	545	4267
NO. sent. nested entities	2712	294	388	2691	338	320	3222	328	448	2809	182	1223
NO. total entities	22204	2514	3035	24441	3200	2993	47006	4461	5596	31236	1879	12601
NO. nested entities	10149	1092	1417	9389	1112	1118	8382	818	1212	8773	605	3707
nested percentage (%)	45.71	46.69	45.61	38.41	34.75	37.35	17.83	18.34	21.66	28.09	32.20	29.42

Table 6: Statistics of the datasets used in the experiments.

Model	**CoNLL2003**		
	P	R	F1
BiLSTM-CRF	-	-	91.03
ELMo-Tagger	-	-	92.22
Bert-Tagger	-	-	92.8
Extract-Select	**92.10**	**94.03**	**93.05**
Model	**Weibo**		
	P	R	F1
SLK-NER	61.80	66.30	64.00
Glyce	67.60	67.68	67.71
FLAT	-	-	68.55
Extract-Select	**69.20**	**70.08**	**69.64**

Table 7: Results for flat NER tasks.

by 8:1:1. For GENIA (Ohta et al., 2002), we use GENIA v3.0.2 corpus, and follow the split of Wang et al. (2020), i.e., first collapse all subtypes into five types, and then split files into training, development, and test sets by 8.1:0.9:1. For KBP2017, we follow Lin et al. (2019) and evaluate the model on the 2017 English evaluation dataset, using previous RichERE annotated datasets as the training set except 20 randomly sampled documents reserved as development set. Finally, there are 866/20/167 files for training, development and test set.

B.2 Implementation Details

We initialize word embeddings of the input sequence and entity marker with 100-dimensional GLoVE vectors for extractor and discriminator. The dimensions of contextualized word embedding, POS embedding, and character embedding are 1024, 50, and 50, respectively. The hidden size is set to 1024. For GENIA dataset, we replace GLoVE vectors with word vectors pre-trained on biomedical corpus (Chiu et al., 2016), which are in 200 dimensions. During the training process, we employ the Adam Optimizer with the initial learning rate as 0.002 and the minibatch size as 64. We use a dropout rate of 0.35 in each training process. We set the threshold in Line 325 through grid search among $(0.2, 0.5, 0.8)$, and it is set to 0.5 for having the best performance.

C Experiments on Flat NER

We also choose two flat NER datasets, i.e., CoNLL2003 and Weibo, to evaluate *Extract-Select*. CoNLL2003 is an English dataset (Sang and Meulder, 2003) with four types of flat entities. We follow the data processing in Lin et al. (2019). Weibo is a Chinese dataset (Peng and Dredze, 2015) sampled from Weibo web pages. We use the same settings in Li et al. (2020a) to evaluate our model.

For English flat NER, we use several taggers as baselines: *BiLSTM-CRF* (Ma and Hovy, 2016), *ELMo-Tagger* (Peters et al., 2018), and *Bert-Tagger* (Devlin et al., 2019). For Chinese flat NER, we use the following models as baselines: *SLK-NER* (Hu and Wei, 2020) which incorporates second-order lexicon knowledge, *Glyce* (Meng et al., 2019) which combines glyph information, and *FLAT* (Li et al., 2020a) which uses phrases.

Table 7 presents comparisons between *Extract-Select* and the baselines on two flat NER datasets. On Weibo dataset, our model outperforms the baselines, improving the F1 score by 1.09%. On CoNLL2003, our model also gains comparable results, with 0.25% performance improvement compared to *Bert-Tagger*. In general, *Extract-Select* achieves good performance on not only nested NER but also flat NER.