Sequence-to-Nuggets: Nested Entity Mention Detection via Anchor-Region Networks

Hongyu Lin[1,3], **Yaojie Lu**[1,3], **Xianpei Han**[1,2,*], **Le Sun**[1,2]
[1]Chinese Information Processing Laboratory [2]State Key Laboratory of Computer Science
Institute of Software, Chinese Academy of Sciences, Beijing, China
[3]University of Chinese Academy of Sciences, Beijing, China
{hongyu2016,yaojie2017,xianpei,sunle}@iscas.ac.cn

Abstract

Sequential labeling-based NER approaches restrict each word belonging to at most one entity mention, which will face a serious problem when recognizing nested entity mentions. In this paper, we propose to resolve this problem by modeling and leveraging the head-driven phrase structures of entity mentions, i.e., although a mention can nest other mentions, they will not share the same head word. Specifically, we propose *Anchor-Region Networks (ARNs)*, a sequence-to-nuggets architecture for nested mention detection. ARNs first identify anchor words (i.e., possible head words) of all mentions, and then recognize the mention boundaries for each anchor word by exploiting regular phrase structures. Furthermore, we also design *Bag Loss*, an objective function which can train ARNs in an end-to-end manner without using any anchor word annotation. Experiments show that ARNs achieve the state-of-the-art performance on three standard nested entity mention detection benchmarks.

1 Introduction

Named entity recognition (NER), or more generally entity mention detection[1], aims to identify text spans pertaining to specific entity types such as *Person*, *Organization* and *Location*. NER is a fundamental task of information extraction which enables many downstream NLP applications, such as relation extraction (GuoDong et al., 2005; Mintz et al., 2009), event extraction (Ji and Grishman, 2008; Li et al., 2013) and machine reading comprehension (Rajpurkar et al., 2016; Wang et al., 2016).

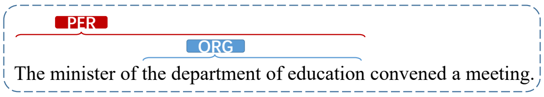

PER
ORG
The minister of the department of education convened a meeting.

Figure 1: An example of nested entity mentions. Due to the nested structure, "the","department","of" and "education" belong to both *PER* and *ORG* mentions.

Previous approaches (Zhou and Su, 2002; Chieu and Ng, 2002; Bender et al., 2003; Settles, 2004; Lample et al., 2016) commonly regard NER as a sequential labeling task, which generate label sequence for each sentence by assigning one label to each token. These approaches commonly restrict each token belonging to at most one entity mention and, unfortunately, will face a serious problem when recognizing nested entity mentions, where one token may belong to multiple mentions. For example in Figure 1, an *Organization* entity mention "the department of education" is nested in another *Person* entity mention "the minister of the department of education". Nested entity mentions are very common. For instance, in the well-known ACE2005 and RichERE datasets, more than 20% of entity mentions are nested in other mentions. Therefore, it is critical to consider nested mentions for real-world applications and downstream tasks.

In this paper, we propose a sequence-to-nuggets approach, named as *Anchor-Region Networks (ARNs)*, which can effectively detect all entity mentions by modeling and exploiting the head-driven phrase structures (Pollard and Sag, 1994; Collins, 2003) of them. ARNs originate from two observations. First, although an entity mention can nest other mentions, they will not share the same head word. And the head word of a mention can provide strong semantic evidence for its entity type (Choi et al., 2018). For example in Figure 1, although the *ORG* mention is nested in the *PER* mention, they have different head words "department" and "minister" respectively, and these head words strongly indicate their corresponding entity types to be *ORG* and *PER*. Second, entity men-

[*]Corresponding author.

[1]In entity mention detection, a mention can be either a named, nominal or pronominal reference of an entity (Katiyar and Cardie, 2018).

Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 5182–5192
Florence, Italy, July 28 - August 2, 2019. ©2019 Association for Computational Linguistics

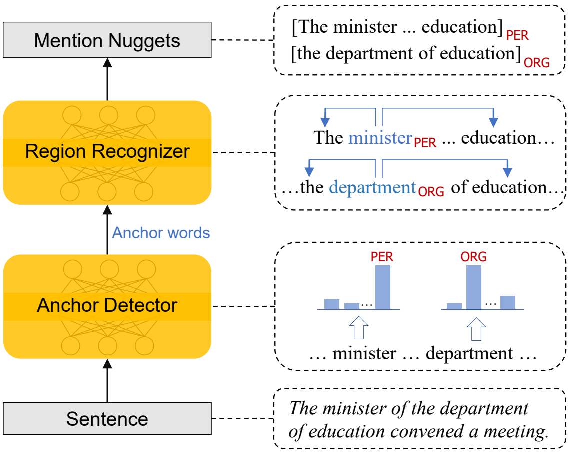

Mention Nuggets
Region Recognizer
Anchor words
Anchor Detector
Sentence
[The minister ... education] PER
[the department of education] ORG
The minister PER ... education…
…the department ORG of education…
PER
ORG
… minister … department …
The minister of the department of education convened a meeting.

Figure 2: The overall architecture of ARNs. Here "minister" and "department" are detected anchor words for two mentions respectively.

tions mostly have regular phrase structures. For the two mentions in Figure 1, they share the same "DET NN of NP" structure, where the NN after DET are their head words. Based on above observations, entity mentions can be naturally detected in a sequence-to-nuggets manner by 1) identifying the head words of all mentions in a sentence; and 2) recognizing entire mention nuggets centered at detected head words by exploiting regular phrase structures of entity mentions.

To this end, we propose ARNs, a new neural network-based approach for nested mention detection. Figure 2 shows the architecture of ARNs. First, ARNs employs an *anchor detector network* to identify whether each word is a head word of an entity mention, and we refer the detected words as *anchor words*. After that, a *region recognizer network* is used to determine the mention boundaries centering at each anchor word. By effectively capturing head-driven phrase structures of entity mentions, the proposed ARNs can naturally address the nested mention problem because different mentions have different anchor words, and different anchor words correspond to different mention nuggets.

Furthermore, because the majority of NER datasets are not annotated with head words, they cannot be directly used to train our anchor detector. To address this issue, we propose *Bag Loss*, an objective function which can be used to train ARNs in an end-to-end manner without any anchor word annotation. Specifically, our Bag Loss is based on *at-least-one assumption*, i.e., each mention should have at least one anchor word, and the anchor word should strongly indicate its entity type. Based on this assumption, Bag Loss can automatically select the best anchor word within each mention during training, according to the association between words and the entity type of the mention. For example, given an *ORG* training instance "the department of education", Bag Loss will select "department" as the anchor word of this mention based on its tight correlation with type *ORG*. While other words in the mention, such as "the" and "of", will not be regarded as anchor words, because of their weak association with *ORG* type.

We conducted experiments on three standard nested entity mention detection benchmarks, including ACE2005, GENIA and TAC-KBP2017 datasets. Experiments show that ARNs can effectively detect nested entity mentions and achieve the state-of-the-art performance on all above three datasets. For better reproduction, we openly release the entire project at github.com/sanmusunrise/ARNs.

Generally, our main contributions are:

- We propose a new neural network architecture named as *Anchor-Region Networks*. By effectively modeling and leveraging the head-driven phrase structures of entity mentions, ARNs can naturally handle the nested mention detection problem and achieve the state-of-the-art performance on three benchmarks. To the best of our knowledge, this is the first work which attempts to exploit the head-driven phrase structures for nested NER.
- We design an objective function, named as *Bag Loss*. By exploiting the association between words and entity types, Bag Loss can effectively learn ARNs in an end-to-end manner, without using any anchor word annotation.
- Head-driven phrase structures are widely spread in natural language. This paper proposes an effective neural network-based solution for exploiting this structure, which can potentially benefit many NLP tasks, such as semantic role labeling (Zhou and Xu, 2015; He et al., 2017) and event extraction (Chen et al., 2015; Lin et al., 2018).

2 Related Work

Nested mention detection requires to identify all entity mentions in texts, rather than only outmost mentions in conventional NER. This raises a critical issue to traditional sequential labeling models because they can only assign one label to each token. To address this issue, mainly two kinds of methods have been proposed.

Region-based approaches detect mentions by identifying over subsequences of a sentence respectively, and nested mentions can be detected because they correspond to different subsequences. For this, Finkel and Manning (2009) regarded nodes of parsing trees as candidate subsequences. Recently, Xu et al. (2017) and Sohrab and Miwa (2018) tried to directly classify over all subsequences of a sentence. Besides, Wang et al. (2018) proposed a transition-based method to construct nested mentions via a sequence of specially designed actions. Generally, these approaches are straightforward for nested mention detection, but mostly with high computational cost as they need to classify over almost all sentence subsequences.

Schema-based approaches address nested mentions by designing more expressive tagging schemas, rather than changing tagging units. One representative direction is hypergraph-based methods (Lu and Roth, 2015; Katiyar and Cardie, 2018; Wang and Lu, 2018), where hypergraph-based tags are used to ensure nested mentions can be recovered from word-level tags. Besides, Muis and Lu (2017) developed a gap-based tagging schema to capture nested structures. However, these schemas should be designed very carefully to prevent spurious structures and structural ambiguity (Wang and Lu, 2018). But more expressive, unambiguous schemas will inevitably lead to higher time complexity during both training and decoding.

Different from previous methods, this paper proposes a new architecture to address nested mention detection. Compared with region-based approaches, our ARNs detect mentions by exploiting head-driven phrase structures, rather than exhaustive classifying over subsequences. Therefore ARNs can significantly reduce the size of candidate mentions and lead to much lower time complexity. Compared with schema-based approaches, ARNs can naturally address nested mentions since different mentions will have different anchor words. There is no need to design complex tagging schemas, no spurious structures and no structural ambiguity.

Furthermore, we also propose Bag Loss, which can train ARNs in an end-to-end manner without any anchor word annotation. The design of Bag Loss is partially inspired by multi-instance learning (MIL) (Zhou and Zhang, 2007; Zhou et al., 2009; Surdeanu et al., 2012), but with a different target. MIL aims to predict a unified label of *a bag of instances*, while Bag Loss is proposed to train ARNs whose anchor detector is required to predict the label of *each instance*. Therefore previous MIL methods are not suitable for training ARNs.

3 Anchor-Region Networks for Nested Entity Mention Detection

Given a sentence, Anchor-Region Networks detect all entity mentions in a two-step paradigm. First, an *anchor detector network* identifies anchor words and classifies them into their corresponding entity types. After that, a *region recognizer network* is applied to recognize the entire mention nugget centering at each anchor word. In this way, ARNs can effectively model and exploit head-driven phrase structures of entity mentions: the anchor detector for recognizing possible head words and the region recognizer for capturing phrase structures. These two modules are jointly trained using the proposed Bag Loss, which learns ARNs in an end-to-end manner without using any anchor word annotation. This section will describe the architecture of ARNs. And Bag Loss will be introduced in the next section.

3.1 Anchor Detector

An anchor detector is a word-wise classifier, which identifies whether a word is an anchor word of an entity mention of specific types. For the example in Figure 1, the anchor detector should identify that "minister" is an anchor word of a *PER* mention and "department" is an anchor word of an *ORG* mention.

Formally, given a sentence $x_1, x_2, ..., x_n$, all words are first mapped to a sequence of word representations $\boldsymbol{x_1}, \boldsymbol{x_2}, ..., \boldsymbol{x_n}$ where $\boldsymbol{x_i}$ is a combination of word embedding, part-of-speech embedding and character-based representation of word x_i following Lample et al. (2016). Then we obtain a context-aware representation $\boldsymbol{h_i^A}$ of each word x_i using a bidirectional LSTM layer:

$$\begin{aligned}\overrightarrow{\boldsymbol{h_i^A}} &= \mathrm{LSTM}(\boldsymbol{x_i}, \overrightarrow{\boldsymbol{h_{i-1}^A}})\\ \overleftarrow{\boldsymbol{h_i^A}} &= \mathrm{LSTM}(\boldsymbol{x_i}, \overleftarrow{\boldsymbol{h_{i+1}^A}})\\ \boldsymbol{h_i^A} &= [\overrightarrow{\boldsymbol{h_i^A}}; \overleftarrow{\boldsymbol{h_i^A}}]\end{aligned} \tag{1}$$

The learned representation $\boldsymbol{h_i^A}$ is then fed into a multi-layer perceptron(MLP) classifier, which computes the scores $\boldsymbol{O_i^A}$ of the word x_i being an anchor word of specific entity types (or *NIL* if this word is not an anchor word):

$$\boldsymbol{O_i^A} = \mathrm{MLP}(\boldsymbol{h_i^A}) \tag{2}$$

where $\boldsymbol{O_i^A} \in R^{|C|}$ and $|C|$ is the number of entity types plus one *NIL* class. Finally a softmax layer is used to normalize $\boldsymbol{O_i^A}$ to probabilities:

$$P(c_j|x_i) = \frac{e^{O_{ij}^A}}{\sum_{k=1}^{|C|} e^{O_{ik}^A}} \tag{3}$$

where O_{ij}^A is the j^{th} element in $\boldsymbol{O_i^A}$, $P(c_j|x_i)$ is the probability of word x_i being an anchor word of class c_j. Note that because different mentions will not share the same anchor word, the anchor detector can naturally solve nested mention detection problem by recognizing different anchor words for different mentions.

3.2 Region Recognizer

Given an anchor word, ARNs will determine its exact mention nugget using a region recognizer network. For the example in Figure 1, the region recognizer will recognize that "the minister of the department of education" is the mention nugget for anchor word "minister" and "the department of education" is the mention nugget for anchor word "department". Inspired by the recent success of pointer networks (Vinyals et al., 2015; Wang and Jiang, 2016), this paper designs a pointer-based architecture to recognize the mention boundaries centering at an anchor word. That is, our region recognizer will detect the mention nugget "the department of education" for anchor word "department" by recognizing "the" to be the left boundary and "education" to be the right boundary.

Similar to the anchor detector, a bidirectional LSTM layer is first applied to obtain the context-aware representation $\boldsymbol{h_i^R}$ of word x_i. For recognizing mention boundaries, local features commonly play essential roles. For instance, a noun before a verb is an informative boundary indicator for entity mentions. To capture such local features, we further introduce a convolutional layer upon $\boldsymbol{h_i^R}$:

$$\boldsymbol{r_i} = \tanh(\boldsymbol{W}\boldsymbol{h_{i-k:i+k}^R} + \boldsymbol{b}) \tag{4}$$

where $\boldsymbol{h_{i-k:i+k}^R}$ is the concatenation of vectors from $\boldsymbol{h_{i-k}^R}$ to $\boldsymbol{h_{i+k}^R}$, $\boldsymbol{W}$ and $\boldsymbol{b}$ are the convolutional kernel and the bias term respectively. k is the (one-side) window size of convolutional layer. Finally, for each anchor word x_i, we compute its left mention boundary score L_{ij} and right mention boundary score R_{ij} at word x_j by

$$\begin{aligned} L_{ij} &= \tanh(\boldsymbol{r_j^T}\boldsymbol{\Lambda_1}\boldsymbol{h_i^R} + \boldsymbol{U_1}\boldsymbol{r_j} + b_1) \\ R_{ij} &= \tanh(\boldsymbol{r_j^T}\boldsymbol{\Lambda_2}\boldsymbol{h_i^R} + \boldsymbol{U_2}\boldsymbol{r_j} + b_2) \end{aligned} \tag{5}$$

In the above two equations, the first term within the $\tanh$ function computes the score of word x_j serving as the left/right boundary of a mention centering at word x_i. And the second term models the possibility of word x_j itself serving as the boundary universally. After that, we select the best left boundary word x_j and best right boundary word x_k for anchor word x_i, and the nugget $\{x_j, ..., x_i, ..., x_k\}$ will be a recognized mention.

4 Model Learning with Bag Loss

This section describes how to train ARNs using existing NER datasets. The main challenge here is that current NER corpus are not annotated with anchor words of entity mentions, and therefore they cannot be directly used to train the anchor detector. To address this problem, we propose *Bag Loss*, an objective function which can effectively learn ARNs in an end-to-end manner, without using any anchor word annotation.

Intuitively, one naive solution is to regard all words in a mention as its anchor words. However, this naive solution will inevitably result in two severe problems. First, a word may belong to different mentions when nested mentions exist. Therefore this naive solution will lead to ambiguous and noisy anchor words. For the example in Figure 1, it is unreasonable to annotate the word "department" as an anchor word of both *PER* and *ORG* mentions, because it has little association to *PER* type although the *PER* mention also contains it. Second, many words in a mention are just function words, which are not associated with its entity type. For example, words "the","of" and "education" in "the department of education" are not associated with its type *ORG*. Therefore annotating them as anchor words of the *ORG* mention will introduce remarkable noise.

To resolve the first problem, we observe that a word can only be the anchor word of the innermost mention containing it. This is because a mention nested in another mention can be regarded as a replaceable component, and changing it will not affect the structure of outer mentions. For the case in Figure 1, if we replace the nested mention "the department of education" by other *ORG* mention(e.g., changing it to "State"), the type of the

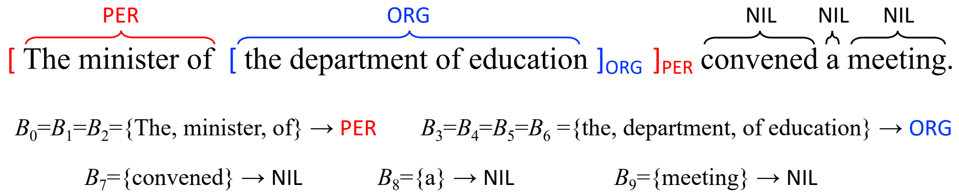

PER
ORG
NIL
NIL
NIL
[The minister of [the department of education]ORG]PER convened a meeting.
B0=B1=B2={The, minister, of} → PER
B3=B4=B5=B6 ={the, department, of education} → ORG
B7={convened} → NIL
B8={a} → NIL
B9={meeting} → NIL

Figure 3: An illustration of bags. B_i represents the bag where word x_i is in. This sentence forms five bags, two of which correspond to two entity mentions and three of which correspond to *NIL*.

outer mention will not change. Therefore, words in a nested mention should not be regarded as the anchor word of outer mentions, and therefore a word can only be assigned as the anchor word of the innermost mention containing it.

To address the second problem, we design *Bag Loss* based on the *at-least-one assumption*, i.e., for each mention at least one word should be regarded as its anchor word. Specifically, we refer to all words belonging to the same innermost mention as a *bag*. And the type of the bag is the type of that innermost mention. For example, in Figure 3,{the, minister, of} will form a *PER* bag, and {the, department, of education} will form an *ORG* bag. Besides, each word not covered by any mention will form a one-word bag with *NIL* type. So there are three *NIL* bags in Figure 3, including {convened}, {a} and {meeting}.

Given a bag, Bag Loss will make sure that at least one word in each bag will be selected as its anchor word, and be assigned to the bag type. While other words in that bag will be classified into either the bag type or *NIL*. Bag Loss selects anchor words according to their associations with the bag type. That is, only words highly related to the bag type (e.g., "department" in "the department of education") will be trained towards the bag type, and other irrelevant words (e.g., "the" and "of" in the above example) will be trained towards *NIL*.

Bag Loss based End-to-End Learning. For ARNs, each training instance is a tuple $x = (x_i, x_j, x_k, c_i)$, where $x_j, ..., x_k$ is an entity mention with left boundary x_j and right boundary x_k. c_j is its entity type and word x_i is a word in this mention's bag[2]. For each instance, Bag loss considers two situations: 1) If x_i is its anchor word, the loss will be the sum of the anchor detector loss (i.e., the loss of correctly classifying x_i into its bag type c_i) and the region recognizer loss (i.e., the loss of correctly recognizing the mention boundary x_j and x_k); 2) If x_i is not its anchor word, the loss will be only the anchor detector loss (i.e., correctly classifying x_i into *NIL*). The final loss for this instance is a weighted sum of the loss of these two situations, where the weight are determined using the association between word x_i and the bag type c_i compared with other words in the same bag. Formally, Bag Loss is written as:

$$\begin{aligned}\mathcal{L}(x_i;\theta) = \; & \omega_i \cdot [-\log P(c_i|x_i) + L^R(x_i;\theta)] \\ & + (1-\omega_i)\cdot[-\log P(NIL|x_i)]\end{aligned} \tag{6}$$

where $-\log P(c_i|x_i)$ is the anchor detector loss. $\mathcal{L}^R(x_i;\theta) = \mathcal{L}^{left}(x_i;\theta) + \mathcal{L}^{right}(x_i;\theta)$ is the loss for the region recognizer measuring how preciously the region recognizer can identify the boundaries centered at anchor word x_i. We define $\mathcal{L}^{left}(x_i;\theta)$ using max-margin loss:

$$\mathcal{L}^{left}(x_i;\theta) = \begin{cases} 0, & c_i = NIL \\ \max(0, \gamma - L_{ij} + \max\limits_{t\neq j} L_{it}), & c_i \neq NIL \end{cases} \tag{7}$$

where γ is a hyper-parameter representing the margin, and $\mathcal{L}^{right}(x_i;\theta)$ is similarly defined.

Besides, ω_i in Equation (6) measures the correlation between word x_i and the bag type c_i. Compared with other words in the same bag, a word x_i should have larger w_i if it has a tighter association with the bag type. Therefore, ω_i can be naturally defined as:

$$\omega_i = [\frac{P(c_i|x_i)}{\max_{x_t \in B_i} P(c_i|x_t)}]^\alpha. \tag{8}$$

where B_i denotes the bag x_i belonging to, i.e., all words that share the same innermost mention with x_i. α is a hyper-parameter controlling how likely a word will be regarded as an anchor word rather than regarded as *NIL*. $\alpha = 0$ means that all words are annotated with the bag type. And $\alpha \to +\infty$ means that Bag Loss will only choose the word with highest $P(c_i|x_i)$ as anchor word, while all other words in the same bag will be regarded as *NIL*. Consequently, Bag Loss guarantees that

[2] For words not in any mention, we define $x_j = x_k = x_i$ and c_i = *NIL*, but their boundary will not be considered during optimization according to Equation (7).

at least one anchor word (the one with highest $P(c_i|x_i)$, and its corresponding w_i will be 1.0) will be selected for each bag. For other words that are not associated with the type (the ones with low $P(c_i|x_i)$), Bag Loss can make it to automatically learn towards *NIL* during training.

5 Experiments

5.1 Experimental Settings

We conducted experiments on three standard English entity mention detection benchmarks with nested mentions: ACE2005, GENIA and TAC-KBP2017 (KBP2017) datasets. For ACE2005 and GENIA, we used the same setup as previous work (Ju et al., 2018; Wang et al., 2018; Wang and Lu, 2018; Katiyar and Cardie, 2018). For KBP2017, we evaluated our model on the 2017 English evaluation dataset (LDC2017E55), using previous RichERE annotated datasets (LDC2015E29, LDC2015E68, LDC2016E31 and LDC2017E02) as the training set except 20 randomly sampled documents reserved as development set. Finally, there were 866/20/167 documents for KBP2017 train/dev/test set. In ACE2005, GENIA and KBP2017, there are 22%, 10% and 19% mentions nested in other mentions respectively. We used Stanford CoreNLP toolkit (Manning et al., 2014) to preprocess all documents for sentence splitting and POS tagging. Adadelta update rule (Zeiler, 2012) is applied for optimization. Word embeddings are initialized with pretrained 200-dimension Glove (Pennington et al., 2014) vectors[3]. Hyper-parameters are tuned on the development sets[4] apart from α in Equation (8), which will be further discussed in Section 5.4.

5.2 Baselines

We compare ARNs with following baselines[5]:

- **Conventional CRF models**, including *LSTM-CRF* (Lample et al., 2016) and *Multi-CRF*. LSTM-CRF is a classical baseline for NER, which doesn't consider nested mentions so only outmost mentions are used for training. Multi-CRF is similar to LSTM-CRF but learns one model for each entity type, and thus is able to recognize nested mentions if they have different types.
- **Region-based methods**, including *FOFE* (Xu et al., 2017), *Cascaded-CRF* (Ju et al., 2018) and a transition model (refered as *Transition*) proposed by Wang et al. (2018). FOFE directly classifies over all sub-sequences of a sentence and thus all potential mentions can be considered. Cascaded-CRF uses several stacked CRF layers to recognize nested mentions at different levels. Transition constructs nested mentions through a sequence of actions.
- **Hypergraph-based methods**, including the *LSTM-Hypergraph (LH)* model (Katiyar and Cardie, 2018) and the *Segmental Hypergraph (SH)* by Wang and Lu (2018). LH used an LSTM model to learn features and then decode them into a hypergraph. SH further considered the transition between labels to alleviate labeling ambiguity, which is the state-of-the-art in both ACE2005 and GENIA[6] datasets.

Besides, we also compared the performance of ARNs with the best system in TAC-KBP 2017 Evaluation (Ji et al., 2017). The same as all previous studies, models are evaluated using micro-averaged Precision(P), Recall(R) and F1-score. To balance time complexity and performance, Wang and Lu (2018) proposed to restrict the maximum length of mentions to 6, which covers more than 95% mentions. So we also compared to baselines where the maximum length of mention is restricted or unrestricted. Besides, we also compared the decoding time complexity of different methods.

5.3 Overall Results

Table 1 shows the overall results on ACE2005, GENIA and KBP2017 datasets. From this table, we can see that:

1) **Nested mentions have a significant influence on NER performance and are required to be specially treated.** Compared with LSTM-CRF and Multi-CRF baselines, all other methods dealing with nested mentions achieved significant F1-score improvements. So it is critical to take nested mentions into consideration for real-world applications and downstream tasks.

[3] http://nlp.stanford.edu/data/glove.6B.zip

[4] The hyper-parameter configures are openly released together with our source code at github.com/sanmusunrise/ARNs.

[5] As Wang and Lu (2018) reported, neural network-based baselines significantly outperform all non-neural methods. So we only compared with neural network-based baselines.

[6] Even Sohrab and Miwa (2018) reported a higher performance on GENIA, their experimental settings are obviously different from other baselines. As they didn't release their dataset splits and source code, we are unable to compare it with listed baselines.

Model	ACE2005			GENIA			KBP2017			Time Complexity
	P	R	F1	P	R	F1	P	R	F1	
LSTM-CRF (Lample et al., 2016)	70.3	55.7	62.2	75.2	64.6	69.5	71.5	53.3	61.1	$O(mn)$
Multi-CRF	69.7	61.3	65.2	73.1	64.9	68.8	69.7	60.8	64.9	$O(mn)$
FOFE(c=6) (Xu et al., 2017)	76.5	66.3	71.0	75.4	67.8	71.4	81.8	62.0	70.6	$O(mn^2)$
FOFE(c=n) (Xu et al., 2017)	76.9	62.0	68.7	74.0	65.5	69.5	79.1	62.5	69.8	$O(mn^2)$
Transition (Wang et al., 2018)	74.5	71.5	73.0	78.0	70.2	73.9	74.7	67.0	70.1	$O(mn)$
Cascaded-CRF (Ju et al., 2018)	74.2	70.3	72.2	78.5	71.3	74.7	-	-	-	-
LH (Katiyar and Cardie, 2018)	70.6	70.4	70.5	79.8	68.2	73.6	-	-	-	$O(mn)$
SH(c=6) (Wang and Lu, 2018)	75.9	70.0	72.8	76.8	71.8	74.2	73.3	65.8	69.4	$O(cmn)$
SH(c=n) (Wang and Lu, 2018)	76.8	72.3	74.5	77.0	73.3	**75.1**	79.2	66.5	72.3	$O(mn^2)$
KBP2017 Best (Ji et al., 2017)	-	-	-	-	-	-	72.6	73.0	72.8	-
Anchor-Region Networks (c=6)	75.2	72.5	73.9	75.2	73.3	74.2	76.2	71.5	73.8	$O(mn + ck)$
Anchor-Region Networks (c=n)	76.2	73.6	**74.9**	75.8	73.9	74.8	77.7	71.8	**74.6**	$O(mn + nk)$

Table 1: Overall experiment results on ACE2005, GENIA and KBP2017 datasets. c is the maximum length of mention and n refers to the length of sentence. For time complexity, m denotes the number of class and k denotes the average number of anchor words in each sentence($k << n$). The time complexity of Cascaded-CRF depends on datasets so is not listed here.

2) **Our Anchor-Region Networks can effectively resolve the nested mention detection problem, and achieved the state-of-the-art performance in all three datasets.** On ACE2005 and GENIA, ARNs achieved the state-of-the-art performance on both the restricted and the unrestricted mention length settings. On KBP2017, ARNs outperform the top-1 system in the 2017 Evaluation by a large margin. This verifies the effectiveness of our new architecture.

3) **By modeling and exploiting head-driven phrase structure of entity mentions, ARNs reduce the computational cost significantly.** ARNs only detect nuggets centering at detected anchor words. Note that for each sentence, the number of potential anchor words k is significantly smaller than the sentence length n. Therefore the computational cost of our region recognizer is significantly lower than that of traditional region-based methods which perform classification on all sub-sequences, as well as hypergraph-based methods which introduced structural dependencies between labels to prevent structural ambiguity (Wang and Lu, 2018). Furthermore, ARNs are highly parallelizable if we replace the BiLSTM context encoder with other parallelizable context encoder architecture (e.g., Transformer (Vaswani et al., 2017)).

5.4 Effects of Bag Loss

In this section, we investigate effects of Bag Loss by varying the values of hyper-parameter α in Equation (8) on the system performance. Figure 4 shows the F1 curves on both ACE2005 and KBP2017 datasets when α varies. We can see that:

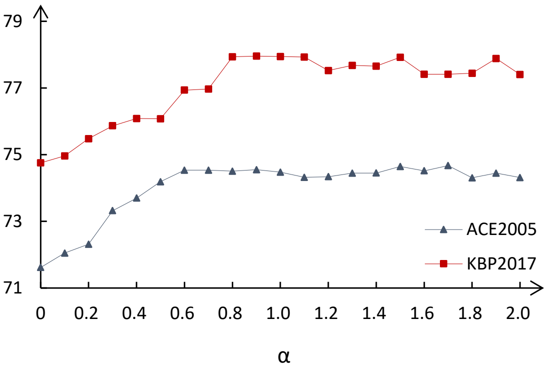


Figure 4: The F1-score w.r.t. different α in Bag Loss on development sets. When $\alpha = 0$, the model ablates Bag Loss and will treat all words in the same innermost mention as anchor words during training.

1) **Bag Loss is effective for anchor word selection during training.** In Figure 4, setting α to 0 significantly undermines the performance. Note that setting α to 0 is the same as ablating Bag Loss, i.e., the model will treat all words in the same innermost mention as anchor words. This result further verifies the necessity of Bag Loss. That is, because not all words in a mention are related to its type, it will introduce remarkable noise by regarding all words in mentions as anchor words.

2) **Bag Loss is not sensitive to α when it is larger than a threshold.** In Figure 4, our systems achieve nearly the same performance when $\alpha > 0.8$. We find that this is because our model can predict anchor word in a very sharp probability distribution, so slight change of α does not make a big difference. Therefore, in all our

Type	Most Frequent Anchor Words
PER	I, you, he, they, we, people, president, Mandela, family, officials
ORG	government, Apple, they, its, Nokia, company, Microsoft, military, party, bank
FAC	building, home, prison, house, store, factories, factory, school, streets, there
GPE	country, China, U.S., US, Cyprus, our, state, countries, Syria, Russia
LOC	world, moon, areas, space, European, Europe, area, region, places, border
NIL	the, a, of, 's, in, and, to, his, who, former

Table 2: The top-10 most frequent anchor words of each type on KBP2017 datasets. Line *NIL* shows most frequent words that appears in a mention but are not regarded as anchor words.

experiments we empirically set $\alpha = 1$ without special declaration. This also verified that Bag Loss can discover head-driven phrase structure steadily without using anchor word annotations.

5.5 Further Discussion on Bag Loss and Marginalization-based Loss

One possible alternative solution for Bag Loss is to regard the anchor word as a hidden variable, and obtain the likelihood of each mention by marginalizing over all words in the mention nugget with

$$P(c, x_j, x_k) = \sum_{x_i} P(x_i, c) P(x_j, x_k | x_i, c). \qquad (9)$$

For $P(x_i, c)$, if we assume that the prior for each word being the anchor word is equal, it can be refactorized by

$$P(x_i, c) = P(c|x_i) P(x_i) \propto P(c|x_i). \qquad (10)$$

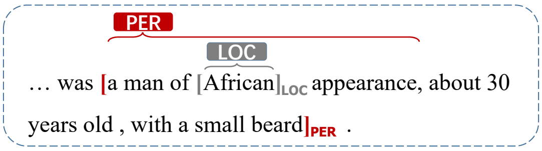
However, we find that this approach does not work well in practice. This may because that, as we mentioned above, the prior probability of each word being the anchor word should not be equal. Words with highly semantic relatedness to the types are more likely to be the anchor word. Furthermore, this marginalization-based training object can only guarantee that words being regarded as the anchor words are trained towards the mention type, but will not encourage the other irrelevant words in the mention to be trained towards *NIL*. Therefore, compared with Bag Loss, the marginalization-based solution can not achieve the promising results for ARNs training.

5.6 Analysis on Anchor Words

To analyze the detected anchor words, Table 2 shows the most common anchor words for all entity types. Besides, words that frequently appear in a mention but being recognized as *NIL* are also presented. We can see that the top-10 anchor words of each type are very convincing: all these words are strong indicators of their entity types. Besides, we can see that frequent *NIL* words in entity mentions are commonly function words, which play significant role in the structure of mention nuggets (e.g., "the" and "a" often indicates the start of an entity mention) but have little semantic association with entity types. This supports our motivation and further verifies the effectiveness of Bag Loss for anchor word selection.

	ACE2005	GENIA	KBP2017
Anchor Detector	82.9	82.7	83.0
Entire ARNs	74.9	74.8	74.6
Δ	8.0	7.9	8.4

Table 3: F1-scores gap between the anchor detector and the entire ARNs (anchor + region).



… was [a man of [African]$_{LOC}$ appearance, about 30 years old , with a small beard]$_{PER}$.

Figure 5: A representative error case of ARNs, where the right boundary of the *PER* mention is misclassified. Braces above the sentence indicate the output of ARNs, and brackets in the sentence represent the golden annotation. We find that the majority of errors occur because of the long-term dependencies stemming from postpositive attributive and attributive clauses.

5.7 Error Analysis

This section conducts error analysis on ARNs. Table 3 shows the performance gap between the anchor detector and the entire ARNs. We can see that there is still a significant performance gap from the anchor detector to entire ARNs. That is, there exist a number of mentions whose anchor words are correctly detected by the anchor detector but their boundaries are mistakenly recognized by the region recognizer. To investigate the reason behind this above performance gap, we analyze these cases and find that most of these errors stem from the existence of postpositive attributive and attributive clauses. Figure 5 shows an error case stemming from postpositive attributive. These cases are quite difficult for neural networks because long-term dependencies between clauses need to be carefully considered. One strategy to handle these cases is to introduce syntactic knowledge, which we leave as future work for improving ARNs.

6 Conclusions and Future Work

This paper proposes Anchor-Region networks, a sequence-to-nuggets architecture which can naturally detect nested entity mentions by modeling and exploiting head-driven phrase structures of entity mentions. Specifically, an anchor detector is first used to detect the anchor words of entity mentions and then a region recognizer is designed to recognize the mention boundaries centering at each anchor word. Furthermore, we also propose Bag Loss to train ARNs in an end-to-end manner without using any anchor word annotation. Experiments show that ARNs achieve the state-of-the-art performance on all three benchmarks.

As the head-driven structures are widely spread in natural language, the solution proposed in this paper can also be used for modeling and exploiting this structure in many other NLP tasks, such as semantic role labeling and event extraction.

Acknowledgments

We sincerely thank the reviewers for their insightful comments and valuable suggestions. Moreover, this work is supported by the National Natural Science Foundation of China under Grants no. 61433015, 61572477 and 61772505; the Projects of the Chinese Language Committee under Grants no. WT135-24; and the Young Elite Scientists Sponsorship Program no. YESS20160177.

References

Oliver Bender, Franz Josef Och, and Hermann Ney. 2003. Maximum entropy models for named entity recognition. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*, pages 148–151. Association for Computational Linguistics.

Yubo Chen, Liheng Xu, Kang Liu, Daojian Zeng, and Jun Zhao. 2015. Event extraction via dynamic multi-pooling convolutional neural networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, volume 1, pages 167–176.

Hai Leong Chieu and Hwee Tou Ng. 2002. Named entity recognition: a maximum entropy approach using global information. In *Proceedings of the 19th international conference on Computational linguistics-Volume 1*, pages 1–7. Association for Computational Linguistics.

Eunsol Choi, Omer Levy, Yejin Choi, and Luke Zettlemoyer. 2018. Ultra-fine entity typing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 87–96. Association for Computational Linguistics.

Michael Collins. 2003. Head-driven statistical models for natural language parsing. *Computational linguistics*, 29(4):589–637.

Jenny Rose Finkel and Christopher D Manning. 2009. Nested named entity recognition. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1-Volume 1*, pages 141–150. Association for Computational Linguistics.

Zhou GuoDong, Su Jian, Zhang Jie, and Zhang Min. 2005. Exploring various knowledge in relation extraction. In *Proceedings of the 43rd annual meeting on association for computational linguistics*, pages 427–434. Association for Computational Linguistics.

Luheng He, Kenton Lee, Mike Lewis, and Luke Zettlemoyer. 2017. Deep semantic role labeling: What works and whats next. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 473–483.

Heng Ji and Ralph Grishman. 2008. Refining event extraction through cross-document inference. *Proceedings of ACL-08: HLT*, pages 254–262.

Heng Ji, Xiaoman Pan, Boliang Zhang, Joel Nothman, James Mayfield, Paul McNamee, Cash Costello, and Sydney Informatics Hub. 2017. Overview of tackbp2017 13 languages entity discovery and linking. In *Proceedings of the Tenth Text Analysis Conference (TAC2017)*.

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, 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, Volume 1 (Long Papers)*, pages 861–871. Association for Computational Linguistics.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. *arXiv preprint arXiv:1603.01360*.

Qi Li, Heng Ji, and Liang Huang. 2013. Joint event extraction via structured prediction with global features. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 73–82.

Hongyu Lin, Yaojie Lu, Xianpei Han, and Le Sun. 2018. Nugget proposal networks for chinese event detection. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1565–1574.

Wei Lu and Dan Roth. 2015. Joint mention extraction and classification with mention hypergraphs. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 857–867.

Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*, pages 55–60.

Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2*, pages 1003–1011. Association for Computational Linguistics.

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*, pages 2608–2618. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Carl Pollard and Ivan A Sag. 1994. *Head-driven phrase structure grammar*. University of Chicago Press.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*.

Burr Settles. 2004. Biomedical named entity recognition using conditional random fields and rich feature sets. In *Proceedings of the international joint workshop on natural language processing in biomedicine and its applications*, pages 104–107. Association for Computational Linguistics.

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*, pages 2843–2849. Association for Computational Linguistics.

Mihai Surdeanu, Julie Tibshirani, Ramesh Nallapati, and Christopher D Manning. 2012. Multi-instance multi-label learning for relation extraction. In *Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning*, pages 455–465. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer networks. In *Advances in Neural Information Processing Systems*, pages 2692–2700.

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*, pages 204–214. Association for Computational Linguistic-s.

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*, pages 1011–1017. Association for Computational Linguistics.

Shuohang Wang and Jing Jiang. 2016. Machine comprehension using match-lstm and answer pointer. *arXiv preprint arXiv:1608.07905*.

Zhiguo Wang, Haitao Mi, Wael Hamza, and Radu Florian. 2016. Multi-perspective context matching for machine comprehension. *arXiv preprint arX-iv:1612.04211*.

Mingbin Xu, Hui Jiang, and Sedtawut Watcharawittayakul. 2017. A local detection approach for named entity recognition and mention detection.

In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1237–1247. Association for Computational Linguistics.

Matthew D Zeiler. 2012. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*.

GuoDong Zhou and Jian Su. 2002. Named entity recognition using an hmm-based chunk tagger. In *proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 473–480. Association for Computational Linguistics.

Jie Zhou and Wei Xu. 2015. End-to-end learning of semantic role labeling using recurrent neural networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, volume 1, pages 1127–1137.

Zhi-Hua Zhou, Yu-Yin Sun, and Yu-Feng Li. 2009. Multi-instance learning by treating instances as non-iid samples. In *Proceedings of the 26th annual international conference on machine learning*, pages 1249–1256. ACM.

Zhi-Hua Zhou and Min-Ling Zhang. 2007. Multi-instance multi-label learning with application to scene classification. In *Advances in neural information processing systems*, pages 1609–1616.