Deep Exhaustive Model for Nested Named Entity Recognition

Mohammad Golam Sohrab[†] and **Makoto Miwa**[†, ‡]
[†]Artificial Intelligence Research Center (AIRC),
National Institute of Advanced Industrial Science and Technology (AIST), Japan
[‡]Toyota Technological Institute, Japan
sohrab.mohammad@aist.go.jp, makoto-miwa@toyota-ti.ac.jp

Abstract

We propose a simple deep neural model for nested named entity recognition (NER). Most NER models focused on flat entities and ignored nested entities, which failed to fully capture underlying semantic information in texts. The key idea of our model is to enumerate all possible regions or spans as potential entity mentions and classify them with deep neural networks. To reduce the computational costs and capture the information of the contexts around the regions, the model represents the regions using the outputs of shared underlying bidirectional long short-term memory. We evaluate our exhaustive model on the GENIA and JNLPBA corpora in biomedical domain, and the results show that our model outperforms state-of-the-art models on nested and flat NER, achieving 77.1% and 78.4% respectively in terms of F-score, without any external knowledge resources.

1 Introduction

Named entity recognition (NER) is a task of finding entities with specific semantic types such as *Protein*, *Cell*, and *RNA* in text. NER is generally treated as a sequential labeling task, where each token is tagged with a label that corresponds to its surrounding entity. However, when entities overlap or are nested within one another, treating the task as a sequential labeling task becomes difficult because an individual token can be included in several entities and defining a label for each token can be difficult. For example, in the following phrase from the GENIA corpus (Kim et al., 2004), four levels of nested entities occur and the token "*IL-2*" is a *Protein* on its own, and it is also a part of two other *Protein*s and one *DNA*.

[[[[*IL-2*]^{Protein} *receptor*]^{Protein} *(IL-2R) alpha chain*]^{Protein} *gene*]^{DNA}

NER has drawn considerable attention as the first step towards many natural language processing (NLP) applications including relation extraction (Miwa and Bansal, 2016), event extraction (Feng et al., 2016), co-reference resolution (Fragkou, 2017; Stone and Arora, 2017), and entity linking (Gupta et al., 2017). Much work on NER, however, has ignored nested entities and instead chosen to focus on the non-nested entities, which are also referred to as flat entities. Only a few studies target the nested named entity recognition (Muis and Lu, 2017; Lu and Roth, 2015; Finkel and Manning, 2009).

Recent successes in neural networks have shown impressive performance gains on flat named entity recognition in several domains (Lample et al., 2016; Ma and Hovy, 2016; Gridach, 2017; Strubell et al., 2017). Such models achieve state-of-the-art results without requiring any hand crafted features or external knowledge resources. In contrast, fewer approaches have emphasized the nested entity recognition problem. Existing approaches to nested NER (Shen et al., 2003; Alex et al., 2007; Finkel and Manning, 2009; Lu and Roth, 2015; Xu et al., 2017; Muis and Lu, 2017) are mostly feature-based and thus suffer from heavy feature engineering. In this paper, we present a novel neural exhaustive model that reasons over all the regions within a specified maximum size. The model represents each region using the outputs of bidirectional long short-term memory (LSTM) by combining the boundary representation of a region and inside representation that simply treats all the tokens in a region equally by taking the average of LSTM outputs corresponding to tokens inside the region. It then classifies regions into their entity types or non-entity. Unlike the existing model that relies on token-level labels, our model directly employs an entity type as the label of a region. The model

Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 2843–2849
Brussels, Belgium, October 31 - November 4, 2018. ©2018 Association for Computational Linguistics

does not rely on any external knowledge resources or NLP tools like part-of-speech taggers. We evaluated our model on the GENIA and JNLPBA corpora in the biomedical domain and the model achieved 77.1% and 78.4% respectively in terms of F-score, which are the new state-of-the-art performances on the corpora.

2 Neural Exhaustive Model

The proposed model exhaustively considers all possible regions in a sentence using a single neural network; we thus call the model neural exhaustive model. Our model is built upon a shared bidirectional LSTM layer. The model enumerates all possible regions or spans that can include all the nested entities. It then represent the regions by using the outputs of the LSTM layer and detect the entities from the regions. The number of possible regions depend on the predefined maximum size. In this section, we describe the architecture of our neural exhaustive model in detail, which is summarized in Figure 1.

2.1 Word Representation

We represent each word by concatenating word embeddings and character-based word representations. Pre-trained word embeddings are used to initialize word embeddings (Chiu et al., 2016). For the character-based word representations, we encode the character-level information of each word following the successes of Ma and Hovy (2016) and Lample et al. (2016) that utilized character embeddings for the flat NER task. The embedding of each character in a word is randomly initialized. We feed the sequence of character embeddings comprising a word to a bidirectional LSTM layer and concatenate the forward and backward output representations to obtain the word representations.

2.2 Exhaustive Combination using LSTM

Given an input sentence sequence $X = \{x_1, x_2, ...x_n\}$, where x_i denotes the i-th word and n denotes the number of words in the sentence sequence, the distributed embeddings of words and characters are fed into a bidirectional LSTM layer that computes the hidden vector sequence in forward $\overrightarrow{\mathbf{h}} = \left\{\overrightarrow{\mathbf{h}_1}, \overrightarrow{\mathbf{h}_2}, \ldots, \overrightarrow{\mathbf{h}_n}\right\}$ and backward $\overleftarrow{\mathbf{h}} = \left\{\overleftarrow{\mathbf{h}_1}, \overleftarrow{\mathbf{h}_2}, \ldots, \overleftarrow{\mathbf{h}_n}\right\}$ manners. We concatenate the forward and backward outputs as $\mathbf{h}_i = \left[\overrightarrow{\mathbf{h}_i}; \overleftarrow{\mathbf{h}_i}\right]$, where $[;]$ denotes concatenation.

With the LSTM output $\mathbf{h}_i$, our exhaustive model shares the underlying representations of all possible regions by exhaustive combination. We generate all possible regions with the sizes less than or equal to the maximum region size L. We use a $region(i, j)$ to represent the region from i to j inclusive, where $1 \leq i < j \leq n$ and $j - i < L$.

2.3 Region Representation and Classification

We represent the region by separating the region into the boundary and inside representations. The boundary representation is important to capture the contexts surrounding the region. We simply rely on the outputs of the bidirectional LSTM layer corresponding to the boundary words of a target region for this purpose. For the inside representation, we simply average the outputs of the bidirectional LSTM layer in the region to treat them equally. We include the outputs for the boundary words to guarantee that the inside representation has corresponding outputs. In summary, we obtain the representation $\mathbf{R}(i, j)$ of the $region(i, j)$ as follows:

$$\mathbf{R}(i, j) = \left[\mathbf{h}_i; \frac{1}{j - i + 1} \sum_{k=i}^{j} \mathbf{h}_k; \mathbf{h}_j\right]. \quad (1)$$

We then feed the representation of each segmented region to a rectified linear unit (ReLU) as an activation function. Finally, the output of the activation layer is passed to a softmax output layer to classify the region into a specific entity type or non-entity.

The exhaustive model represents all possible regions based on maximum entity length and classify all of them. The overall number of classifications for each sentence in the exhaustive model is in $O(lmn)$, where l is a total number of words in the sentence, m is the maximum entity length and n is the total number of possible entity types. Finkel and Manning (2009) and Alex et al. (2007) proposed featured-based approaches for handling nested NER. The time complexity of their models are expensive, i.e., cubic in the number of the words in the sentence. The exhaustive approach is fast since we run the LSTM once and the classifications can be performed in parallel on the combinations created from the LSTM outputs.

The exhaustive model classify each region independently unlike word-level taggers. This makes the model flexible so that it can incorporate

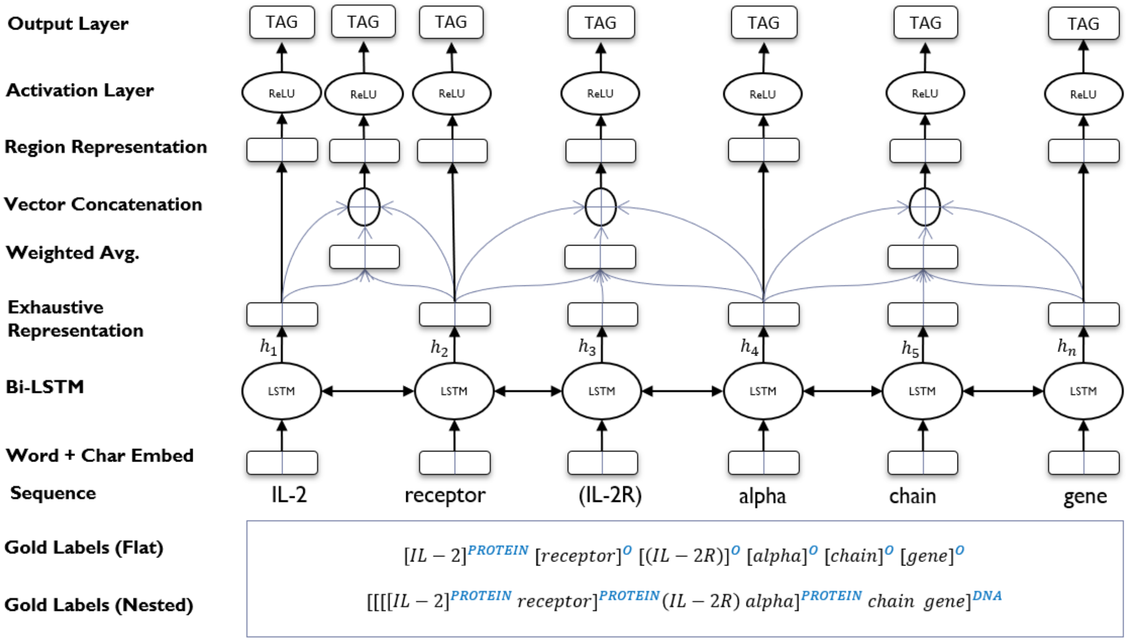

Figure 1: Architecture of the proposed neural exhaustive model. The model considers all possible regions up to a maximum size, but we depict here only a small subset for brevity. "*IL-2*", "*IL-2 receptor*", "*IL-2 receptor (IL-2R) alpha*", and "*IL-2 receptor (IL-2R) alpha chain gene*" are nested entities.

phrase-level dictionary information directly and we can tune biases for each type unlike CRF. We leave this evaluation to our future work.

3 Experimental Settings

We evaluated our exhaustive model on GENIA[1] (Kim et al., 2003) and JNLPBA[2] (Kim et al., 2004) datasets to provide empirical evidence for the effectiveness of our model both in nested and flat NER. Table 1 shows the statistics of GENIA dataset.

Our model was implemented in Chainer[3] deep learning framework. We employed pre-trained word embeddings that were trained on MEDLINE abstracts (Chiu et al., 2016), which included 200-dimensional embeddings of 2,231,686 vocabulary. We used ADAM (Kingma and Ba., 2015) for learning with a mini-batch size of 100. We used the same hyper-parameters in all the experiments; we set the dimension of word embedding to 200, the dimension of character embedding to 25, the hidden layer size to 200, the gradient clipping to 5, and the ADAM hyper-parameters to its default values (Kingma and Ba., 2015).

Item	Train	Dev	Test
Documents	1,599	189	212
Sentences	15,022	1,669	1,855
Split(%)	81	9	10
DNA	7,921	1061	1,283
RNA	730	140	117-
protein	29,032	2,338	3,098
cell line	3,149	340	460
cell type	6,021	563	617
outermost entity	42,462	4,020	4,942
nested level	4	3	3
entity avg. length	2.87	3.13	2.93
multi-token entity	33951	3554	4203
overall nested entity	8301	803	1202
overall entity	46,853	4,442	5,575

Table 1: Statistics of GENIA

To deeply understand the model parameters, we compared the models in different regions. We chose the maximum region size from 3, 6, 8 and 10. We also employed different region representation. We tried only the boundary representation (boundary), only the inside representation (inside), and our region representation (boundary+inside).

We employed precision, recall, and F-score to

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

[2] http://www.nactem.ac.uk/tsujii/GENIA/ERtask/report.html

[3] https://chainer.org/

Model	P(%)	R(%)	F(%)
Exhaustive Model	**93.2**	64.0	**77.1**
Ju et al. (2018)	78.5	**71.3**	74.7
Katiyar and Cardie	76.7	71.1	73.8
Muis and Lu (2017)	75.4	66.8	70.8
Lu and Roth (2015)	72.5	65.2	68.7
Finkel and Manning	75.4	65.9	70.3

Table 2: Performance comparison of the state-of-the-art nested NER models on the test dataset.

Entity Level	P(%)	R(%)	F(%)
Single-token	91.6	58.4	69.9
Multi-token	95.9	65.8	77.9
Top Level	92.7	69.8	79.3
Nested	94.3	59.3	72.7
All entities	93.2	64.0	77.1

Table 3: Performances of our model on different entity level on the test dataset.

evaluate our model. We also compared the performances for single-token v.s. multi-token entities and top-level v.s. nested entities.

4 Results and Discussions

4.1 Nested NER

Table 2 shows the comparison of our model with several previous state-of-the nested NER models on the test dataset. Our model outperforms the state-of-the-art models in terms of F-score. Our results on Table 2 is based on bidirectional LSTM with character embeddings and the maximum region size is 10.

Table 3 describes the performances of our model on different entity levels on the test dataset. The model performs well on multi-token and top-level entities. This is interesting because they are often considered difficult for sequential labeling models.

Table 4 shows the performances on the five entity types on the test dataset. We here show the performance by Finkel and Manning (2009) (F&M) for the reference. Our system performs better than their model except for the *RNA* type.

4.2 Ablation Tests

We show the differences in the performance on the development dataset to compare the possible scenarios of the proposed approach and to report the importance of each component in our exhaustive model.

Label	P(%)	R(%)	F(%)	F&M F(%)
DNA	92.6	58.7	**71.8**	65.2
RNA	98.8	57.1	72.4	**74.7**
cell line	94.6	53.1	**67.9**	64.0
cell type	88.4	70.0	**78.1**	67.1
protein	94.1	70.8	**80.8**	73.8

Table 4: Categorical performances on the GENIA test dataset.

Region	Ratio(%)	P(%)	R(%)	F(%)
size = 3	89.6	92.9	69.8	79.5
size = 6	98.9	93.6	66.7	77.5
size = 8	99.4	93.7	66.5	77.6
size = 10	100	93.5	67.6	78.2

Table 5: Performance of our model with different maximum region sizes on the development dataset. Ratio refers to the coverage ratio of entity mentions.

Setting	P(%)	R(%)	F(%)
Bi-LSTM	94.1	65.7	77.1
Bi-LSTM + Character*	93.5	67.6	78.2
Boundary*	94.1	54.3	68.5
Inside*	93.2	46.4	61.2
Boundary+Inside*	93.5	67.6	78.2

Table 6: Performance of our model with different model architectures on the development dataset. $\star$ indicates results using character embeddings.

Label	P(%)	R(%)	F(%)
DNA	95.2	56.8	71.4
RNA	96.1	61.4	75.2
cell line	86.2	44.1	58.8
cell type	96.7	61.5	75.3
protein	97.1	72.2	82.6
overall	96.4	66.8	78.4

Table 7: Categorical and overall performances of the JNLPBA test dataset.

Table 5 shows the coverage ratio and the performance with different maximum region sizes. Since the average entity mention length of GENIA dataset is less than 4, the system can cover almost all the entities for the maximum sizes of 6 or more. The longer maximum region size is desirable to cover all the mentions, but it requires more computational costs. Fortunately, the performance did not degrade with the long maximum region size, despite the fact that it introduces more out-of-entity regions.

Ablations on character embeddings in Table 6 also show the importance of character embeddings. It also shows that both the boundary information and the inside information, i.e., average of the embeddings in a region, are necessary to improve the performance.

4.3 Flat NER

We evaluated our model on JNLPBA as a flat dataset, where nested and discontinuous entities are removed. Table 7 shows the performances of our model on JNLPBA dataset. We compared our result with the state-of-the-art result of Gridach (2017) which achieved 75.8% in F-score, where our model obtained 78.4% in terms of F-score.

5 Related Work

Interests in nested NER detection have increased in recent years, but it is still the case that NER models deals with only one flat level at a time. Zhou et al. (2004) detected nested entities in a bottom-up way. They detected the innermost flat entities and then found other NEs containing the flat entities as substrings using rules derived from the detected entities. The authors reported an improvement of around 3% in the F-score under certain conditions on the GENIA corpus (Collier et al., 1999). Katiyar and Cardie (2018) proposed a neural network-based approach that learns hypergraph representation for nested entities using features extracted from a recurrent neural network (RNN). The authors reported that the model outperformed the existing state-of-the-art feature-based approaches.

Recent studies show that the conditional random fields (CRFs) can significantly produce higher tagging accuracy in flat (Athavale et al., 2016) or nested (stacking flat NER to nested representation) (Son and Minh, 2017) NERs. Ju et al. (2018) proposed a novel neural model to address nested entities by dynamically stacking flat NER layers until no outer entities are extracted. A cascaded CRF layer is used after the LSTM output in each flat layer. The authors reported that the model outperforms state-of-the-art results by achieving 74.5% in terms of F-score. Finkel and Manning (2009) proposed a tree-based representation to represent each sentence as a constituency tree of nested entities. All entities were treated as phrases and represented as subtrees following the whole tree structure and used a CRF-based approach driven by entity-level features to detect nested entities. We demonstrate that the performance can be improved significantly without CRFs, by training an exhaustive neural model that learns which regions are entity mentions and how to best classify the regions.

6 Conclusion

This paper presented a neural exhaustive model that considers all possible regions exhaustively for nested NER. The model obtains the representation of each region from an underlying shared LSTM layer, and it represents the region by concatenating boundary representations of the region and inside representation that averages embeddings of words in the region. It then classifies the region into its entity type or non-entity. The model does not depend on any external NLP tools. In the experiment, we show that our model learns to detect nested named entities from the generated mention candidates of all possible regions. Our exhaustive model outperformed existing models with a significant margin in terms of F-score in both flat and nested NER.

For future work, we would like to investigate the use of region-level information. We also consider modeling the dependencies between regions.

Acknowledgments

We thank the anonymous reviewers for their valuable comments. This paper is based on results obtained from a project commissioned by the New Energy and Industrial Technology Development Organization (NEDO).

References

Beatrice Alex, Barry Haddow, and Claire Grover. 2007. Recognising Nested Named Entities in Biomedical Text. In *Proceedings of the Workshop on BioNLP 2007: Biological, Translational, and Clinical Language Processing*, pages 65–72, Stroudsburg, PA, USA. ACL.

Vinayak Athavale, Shreenivas Bharadwaj, Monik Pamecha, Ameya Prabhu, and Manish Shrivastava. 2016. *Towards Deep Learning in Hindi NER: An approach to tackle the Labelled Data Scarcity*. Cornell University Library.

Billy Chiu, Gamal 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*, pages 166–174.

N. Collier, H. S. Park, N. Ogata, Y. Tateisi, C. Nobata, T. Ohta, T. Sekimizu, H. Imai, K. Ibushi, and Jun'ichi Tsujii. 1999. The GENIA Project: Corpus-based Knowledge Acquisition and Information Extraction from Genome Research Papers. In *Proceedings of EACL*, pages 171–172. ACL.

Xiaocheng Feng, Lifu Huang, Duyu Tang, Heng Ji, Bing Qin, and Ting Liu. 2016. A Language-Independent Neural Network for Event Detection. In *Proceedings of the 54th Annual Meeting of the ACL (Volume 2: Short Papers)*, pages 66–71, Berlin, Germany.

Jenny Rose Finkel and Manning Manning, Christopher D. 2009. Nested Named Entity Recognition. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 141–150, Singapore. ACL.

Pavlina Fragkou. 2017. Applying named entity recognition and co-reference resolution for segmenting english texts. *Progress in Artificial Intelligence*, 6(4):325–346.

Mourad Gridach. 2017. Character-level neural network for biomedical named entity recognition. *Journal of biomedical informatics*, 70:85–91.

Nitish Gupta, Sameer Singh, and Dan Roth. 2017. Entity Linking via Joint Encoding of Types, Descriptions, and Context. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2671–2680, Copenhagen, Denmark. ACL.

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, New Orleans, Louisiana. ACL.

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, New Orleans, Louisiana. ACL.

Jin-Dong Kim, Tomoko Ohta, Yoshimasa Tsuruoka, Yuka Tateisi, and Nigel Collier. 2004. Introduction to the Bio-Entity Recognition Task at JNLPBA. In *Proceedings of the international joint workshop on natural language processing in biomedicine and its applications*, pages 70–75. ACL.

Jin-Dong Kim, Yuka Ohta, Tateisi, and Junichi Tsujii. 2003. GENIA corpus– a semantically annotated corpus for bio-textmining. *Bioinformatics*, 19(suppl. 1):i180–i182.

Diederik Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *ICLR*.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural Architectures for Named Entity Recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the ACL: Human Language Technologies. ACL*, volume 1, pages 260–270, San Diego, California. ACL.

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. ACL.

Xuezhe Ma and Eduard H. Hovy. 2016. End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF. pages 1064–1074.

Makoto Miwa and Mohit Bansal. 2016. End-to-End Relation Extraction using LSTMs on Sequences and Tree Structures. In *Proceedings of the 54th Annual Meeting of the ACL*, pages 1105–1116, Berlin, Germany. ACL.

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 2598–2608, Copenhagen, Denmark. ACL.

Dan Shen, Jie Zhang, Guodong Zhou, Jian Su, and Chew-Lim Tan. 2003. Effective Adaptation of a Hidden Markov Model-based Named Entity Recognizer for Biomedical Domain. In *Proceedings of the ACL 2003 workshop on Natural language processing in biomedicine-Volume 13*, pages 49–56, Sapporo, Japan. ACL.

Nguyen Truong Son and Nguyen Le Minh. 2017. Nested Named Entity Recognition Using Multilayer Recurrent Neural Networks. In *Proceedings of PACLING 2017*, pages 16–18, Sedona Hotel, Yangon, Myanmar.

M. Stone and R. Arora. 2017. *Identifying nominals with no head match co-references using deep learning*. CoRR abs/1710.00936.

Emma Strubell, Patrick Verga, David Belanger, and Andrew McCallum. 2017. Fast and Accurate Entity Recognition with Iterated Dilated Convolutions. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2670–2680, Copenhagen, Denmark. ACL.

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 ACL (Volume 1: Long Papers)*, volume 1, pages 1237–1247.

Guodong Zhou, Jie Zhang, Jian Su, Dan Shen, and Chewlim Tan. 2004. Recognizing Names in Biomedical Texts: a Machine Learning Approach. *Bioinformatics*, 20(7):1178–1190.