A Generic Neural Exhaustive Approach for Entity Recognition and Sensitive Span Detection

Mohammad Golam Sohrab[1], Pham Minh Thang[1], and Makoto Miwa[1,2]

[1] Artificial Intelligence Research Center, National Institute of Advanced Industrial Science and Technlogy, 2-4-7 Aomi, Koto-ku, Tokyo, 135-0064, Japan
{sohrab.mohammad, pham.thang}@aist.go.jp
[2] Toyota Technological Institute, 2-12-1 Hisakata Tempaku-ku Nagoya, Japan
makoto-miwa@toyota-ti.ac.jp

Abstract. In this work, we present a deep exhaustive framework for the MEDDOCAN shared task. The framework employs a generic named entity recognition (NER) model that captures the underlying semantic information of texts. The key idea of our model is to enumerate all possible spans as potential entity mentions and classify them with deep neural networks. We introduce different sets of learning algorithms, including base representation(BR) average (BR-Avg), BR with attention mechanigm (BR-Attn), LSTM-Minus-based average (LM-Avg), LSTM-Minus-based attention (LM-Attn), where with or without context is used after LSTM layer (Context or None) and an ensemble approach using maximum voting of all the approaches. We evaluate our exhaustive model on two sub-tasks in the MEDDOCAN shared task in medical domain using the official evaluation script. Among the five submitted runs, the best run for each sub-task achieved the F-score of 93.12% on Sub-task 1 and the F-scores of 93.52% (strict) and 94.92% (merged) on Sub-task 2 without any external knowledge resources.

Keywords: Deep learning · NER · Exhaustive approach.

1 Introduction

The MEDDOCAN shared task [9] is an open challenge medical entity detection task that allows participants to use any methodology and knowledge sources for the clinical records with protected health information (PHI). The task allows the comparison of the participating systems using the same benchmark dataset and evaluation method. Named entity recognition has drawn considerable attentions as the first step towards many natural language processing (NLP) applications including relation extraction [10], event extraction [3], co-reference resolution [4], and entity linking [5]. Recently, deep neural networks have shown impressive performance on flat named entity recognition in several domains [8]. Such models achieved the state-of-the-art results without requiring any handcrafted features or external knowledge resources.

In this paper, we present a novel neural exhaustive model that detects flat and nested entities. The model reasons over all the regions within a specified

maximum size. The model first represents each region as the combination of the boundary and inside representations by using the outputs of bidirectional long short-term memory (LSTM). The inside representation 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 the regions into their entity types or non-entity. Unlike the existing models that rely on token-level labels, our model directly employs an entity type as the label of a region.

We evaluated our model on the MEDDOCAN task in clinical domain, which aims at named entity recognition (NER), which is officially called NER offset and entity type classification, and sensitive span detection. The best run for each sub-task achieved the F-score of 93.12% on sub-task 1 and the F-scores of 93.52% (strict) and 94.92% (merged) on sub-task 2.

2 Related Works

Sohrab et al. [11] detected the inner and outermost entities using exhaustive approach and outperformed the state-of-the-art results by achieving 77.1% in terms of F-score. Zhou et al. [14] detected nested entities in a bottom-up way. They detected the innermost flat entities and then found other NEs containing the flat entities as sub-strings 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 [2]. Recent studies show that the conditional random fields (CRFs) can produce significantly higher tagging accuracy in flat or nested (stacking flat NER to nested representation) [12] NERs. Ju et al. [6] 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 F-score.

3 Neural Exhaustive Approach

We solve the NER and sensitive span detection (SSD) tasks using a neural exhaustive approach that exhaustively consider all possible regions in a sentence using a single neural network. The model detects nested entities by enumerating all possible spans or regions. Our model is built upon a shared bidirectional LSTM (Bi-LSTM) layer. Figure 1 shows the exhaustive model to solve the entity recognition and SSD.

3.1 Embedding Layer

In the embedding layer, each word is represented by concatenating the pretrained word embedding and character-based word representations where we encode the character-level information of the word. The character-based word representations are obtained by feeding the sequence of character embeddings comprising a word to a Bi-LSTM layer and concatenate the forward and backward output representations. The character embeddings in a word is randomly initialized.

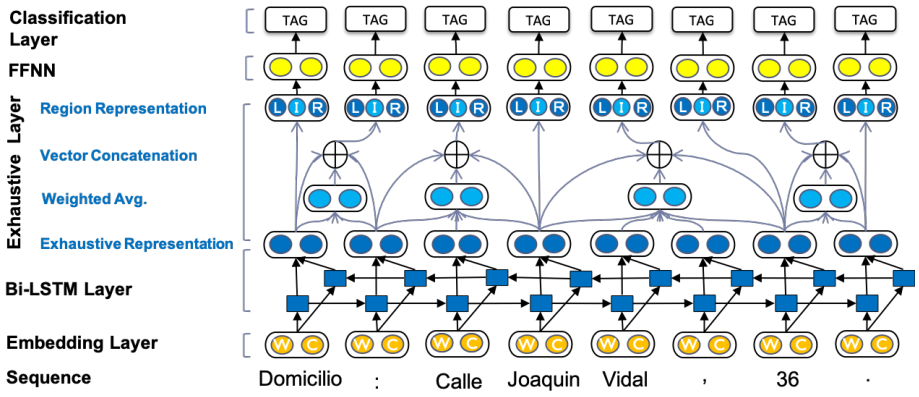

Fig. 1: A overview of the exhaustive model with base region representations.

3.2 Bidirectional LSTM Layer

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, which are introduced in the last section, are fed into a bidirectional LSTM (Bi-LSTM) layer. The Bi-LSTM layer 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.

3.3 Exhaustive Layer

The exhaustive layer enumerates all possible regions by exhaustive combination. We generate all possible regions with the sizes less than or equal to the maximum region size L, which is predefined. We use (i, k) to represent the region from i to k inclusive, where $1 \leq i < k \leq n$ and $k - i < L$.

We represent each region using the outputs of the shared underlying LSTM layer. We represent the region with two separate representations: the boundary representation for region detection and the inside representation for semantic type classification. In the latter part of this section, we first introduce the base region representations and then explain two enhancements.

Base Region Representations The boundary representation is prepared to capture the both ends of the region. We rely on the outputs of the bidirectional LSTM layer corresponding to the boundary words of a target region for this purpose. We obtain the left- and right-boundary representations $\mathbf{R}(i, k)^{[L,R]}$ of the region (i, k) as follows:

$$\mathbf{R}(i, k)^{[L,R]} = [\mathbf{h}_i; \mathbf{h}_k]\,. \tag{1}$$

The inside representation is prepared to capture its semantic type by encoding the whole semantic information of the region. In the base representation, we average the outputs of the Bi-LSTM layer in the region to treat them equally.

Using the boundary and inside representations, we obtain the left-, inside with average representation, and right-boundary $\mathbf{R}(i,k)^{[L,A,R]}$ of the region (i,k) as follows:

$$\mathbf{R}(i,k)^{[L,A,R]} = \left[\mathbf{h}_i; \frac{1}{k-i+1}\sum_{j=i}^{k}\mathbf{h}_j; \mathbf{h}_k\right]. \tag{2}$$

Region Representations using Attention Mechanism Instead of relying only on the average of the outputs of Bi-LSTM layer, we also try an attention mechanism [1] over words in each region for the task of notion of headness. Specifically, we extend the inside representations using attention mechanism as follows:

$$\alpha_t = \mathbf{w}_\alpha FFNN_\alpha\left(\overleftrightarrow{\mathbf{x}}_t\right), \tag{3}$$

$$\alpha_{i,t} = \frac{\exp(\alpha_t)}{\sum_{k=start(i)}^{end(i)}\exp(\alpha_k)}, \tag{4}$$

$$\overline{\mathbf{x}}_i = \sum_{k=start(i)}^{end(i)}\alpha_{i,t}\overleftrightarrow{\mathbf{x}}_t, \tag{5}$$

where $\overleftrightarrow{\mathbf{x}}_t$ is the concatenated output of the Bi-LSTM layer over a region. $\overline{\mathbf{x}}_i$is a weighted sum of word vectors in region (i,k). Instead of Eq. 2, we obtain left-, inside with attention-based representation, and right-boundary $\mathbf{R}(i,k)^{[L,\overline{A},R]}$ of the region (i,k) as follows:

$$\mathbf{R}(i,k)^{[L,\overline{A},R]} = \left[\mathbf{h}_i; \overline{\mathbf{x}}_i; \mathbf{h}_k\right]. \tag{6}$$

Region Representations using LSTM-Minus We also employ LSTM-Minus [13] for the boundary representation. The left-boundary computed as the representation of the previous word of the region subtracted from the representation of the last word of the current region. Similarly, the right-boundary computed as the representation of the next word of the region subtracted from the representation of the first word of the current region. We obtain the representation $\mathbf{R}(i,k)^{[\overline{L},\overline{R}]}$ of the region (i,k) by concatenating the left- and right-boundary based on LSTM-Minus and it is computed as follows:

$$\mathbf{R}(i,k)^{[\overline{L},\overline{R}]} = \left[\mathbf{h}_k - \mathbf{h}_{i-1}; \mathbf{h}_i - \mathbf{h}_{k+1}\right]. \tag{7}$$

The above region or span information is concatenated with average embeddings of the region (i,k) to produce the LSTM-Minus-based representations as:

$$\mathbf{R}(i,k)^{[\overline{L},A,\overline{R}]} = \left[\mathbf{h}_k - \mathbf{h}_{i-1}; \frac{1}{k-i+1}\sum_{j=i}^{k}\mathbf{h}_j; \mathbf{h}_i - \mathbf{h}_{k+1}\right]. \tag{8}$$

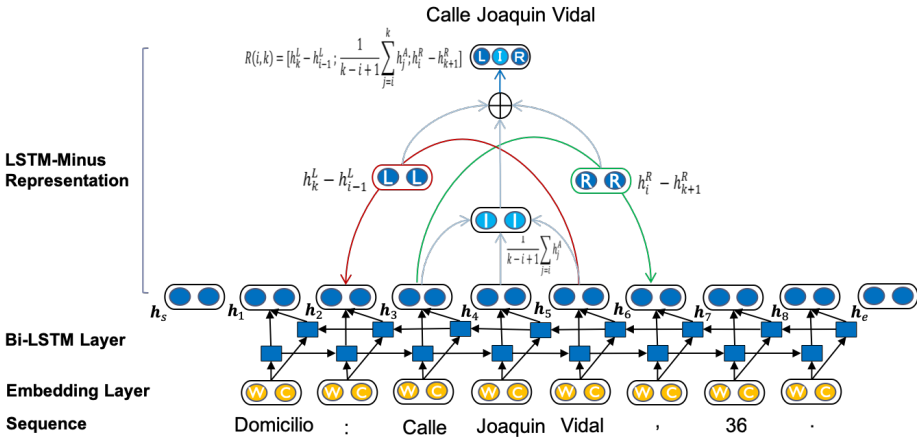

Fig. 2: LSTM-Minus based region representations.

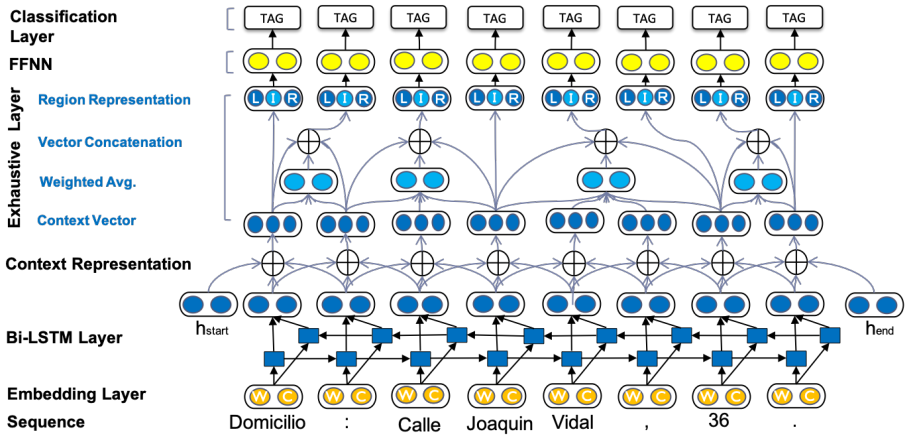

Fig. 3: An overview of the exhaustive model with contextual region representations.

Furthermore, the LSTM-Minus based representation using attention can be considered as:

$$\mathbf{R}(i,k)^{[\overline{L},\overline{A},\overline{R}]} = [\mathbf{h}_k - \mathbf{h}_{i-1}; \overline{\mathbf{x}}_i; \mathbf{h}_i - \mathbf{h}_{k+1}]\,. \tag{9}$$

3.4 Contextual Region Representations

With the LSTM output $\mathbf{h}_i$, we introduce a context level representation from bidirectional LSTM layer. The idea of this approach is to capture the surrounding LSTM output of a target region (i,k) by concatenating vector output of previous $\mathbf{h}_{i-1}$, current $\mathbf{h}_i$, and next index $\mathbf{h}_{k+1}$ of LSTM output. With contextual

region representations, we can further generate new representation from Eqs. 1-9. Figure 3 shows an architecture of contextual level integration. 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.

4 Experimental Settings

4.1 Evaluation Settings

We evaluated our exhaustive model on MEDDOCAN[3] dataset to provide empirical evidence for the effectiveness of the exhaustive model both in NER and SSD. Our model is implemented in Chainer[4] deep learning framework. We generated task specific word embeddings by merging the raw text of training, development, and test (including background set) sets, which included 200-dimensional embeddings of 77,559 vocabulary. We used Adam [7] for learning with a mini-batch size of 10. 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 [7]. We employed the official MEDDOCAN evaluation script[5] to evaluate our system performances for both tasks.

4.2 Data Pre-processing

We read text directly from input text files. We learn and detect spans using the neural exhaustive approach from Bi-LSTM layer, creating all possible combination from beginning to end of a given sequence. Unlike the traditional NER models, our model is independent from traditional 'BIO' tagging scheme, where 'B', 'I', and 'O' are stands for 'Begin', 'Inside', and 'Outside' of named entities, respectively. Thus, each text and annotation files are processed by several simple rules only for tokenization. After tokenization, each text with mapping annotation files are passed to deep neural approach for mention detection, classification, and sensitive token detection. Note that the offsets are restored to the original offsets in evaluation.

5 Results and Discussions

In order to evaluate the performance of NER and sensitive token detection, we conduct experiments on different sets of learning algorithms, including base representation(BR) average (BR-Avg), BR attention (BR-Attn), LSTM-Minus-based average (LM-Avg), LSTM-Minus-based attention (LM-Attn), where with

[3] http://temu.bsc.es/meddocan/index.php/data/
[4] https://chainer.org/
[5] https://github.com/PlanTL-SANIDAD/MEDDOCAN-CODALAB-Evaluation-Script

Table 1: Test Set: Performances using strict evaluation on Sub-task 1 and strict and merged evaluations on Sub-task 2

	Sub-task 1				Sub-task 2					
Learning Model	Strict				Strict			Merged		
	P(%)	R(%)	F(%)	Leak(%)	P(%)	R(%)	F(%)	P(%)	R(%)	F(%)
Ensemble	95.68	90.69	**93.12**	0.070	96.09	91.08	**93.52**	97.70	92.03	94.78
BR-Attn-None	94.12	89.33	91.66	0.080	92.64	92.03	92.33	94.91	92.81	93.85
LM-Attn-Context	92.55	90.23	91.38	0.073	93.57	91.22	92.38	96.23	92.24	94.20
BR-Avg-None	91.00	91.59	91.29	0.063	91.96	92.56	92.26	95.95	93.91	**94.92**
LM-Attn-None	93.25	88.61	90.87	0.086	94.75	89.93	92.28	96.92	90.91	93.82

Table 2: Sub-task 1: Categorical performances on test set using strict evaluation

Label	P(%)	R(%)	F(%)	Label	P(%)	R(%)	F(%)
CORREO_elect.	97.85	91.57	94.61	Id_asegura.	99.49	99.49	99.49
Sexo_sujeto_asist.	99.13	99.13	99.13	Famil._sujeto_asist.	82.09	67.90	74.32
Edad_sujeto_asist.	98.02	95.56	96.77	Hospital	91.57	58.46	71.36
Id_titulacion_per._sani.	99.57	99.99	99.79	Id_empleo_per._sani.	0.000	0.000	0.000
Nombre_per._sani.	96.71	94.01	95.34	Numero_telefono	94.74	69.23	80.00
Fechas	97.72	98.20	97.96	Id_contacto_asist.	99.98	97.44	98.70
Pais	98.82	92.29	95.44	Profesion	0.000	0.000	0.000
Territorio	97.05	92.89	94.92	Institucion	60.00	13.43	21.95
Calle	77.19	63.92	69.93	Numero_fax	99.98	14.29	25.00
Id_sujeto_asist.	99.26	94.70	96.93	Otros_sujeto_asist.	0.000	0.000	0.000
Nombre_sujeto_asist.	91.91	99.60	95.60	Centro_salud	0.000	0.000	0.000
Overall (micro)	95.68	90.69	93.12	Overall (macro)	95.86	91.30	93.36

***Note**: We abbreviate some labels for brevity.*

or without context is used after LSTM layer (Context or None). Table 1 shows the five submitted results on NER in terms of F-score on the test sets. In strict matching, it is shown that ensemble approach using maximum voting of all the approaches, including BR-avg-None, BR-Attn-None, BR-Avg-Context, BR-Attn-Context, LM-avg-None, LM-Attn-None, LM-Avg-Context, LM-Attn-Context for NER and sensitive token detection is very effective to improve the system performance. In contrast, the BR-Avg-None shows the best performance on NER in terms of F-score when using merged matching. Table 2 shows the categorical performances on the MEDDOCAN dataset.

We show the differences in performance on the development data set to compare the possible scenarios of the given solutions and to report the best system submissions for NER and SSD. Table 3 shows the performances of different approaches on the development set in Sub-task 1 and 2. Table 3 in Sub-task 1 shows that almost all the results in different approaches are close to each other to solve the Sub-task 1. In contrast, Table 3 in Sub-task 2 shows that attention and average with different boundary representations of a region are effective both in strict and merged evaluations to detect sensitive token.

Table 3: Development Set: Performances using strict evaluation on Sub-task 1 and strict and merged evaluations on Sub-task 2

	Sub-task 1				Sub-task 2					
Learning Model	Strict				Strict			Merged		
	P(%)	R(%)	F(%)	Leak(%)	P(%)	R(%)	F(%)	P(%)	R(%)	F(%)
Ensemble	95.66	92.01	**92.74**	0.075	96.01	90.33	**93.08**	97.29	91.06	94.07
BR-Attn-None	94.03	88.76	91.32	0.084	94.83	89.52	92.10	96.88	90.84	93.57
LM-Attn-Context	92.51	90.04	91.26	0.083	93.29	90.79	92.02	95.74	91.77	93.71
BR-Avg-None	91.52	90.93	91.22	0.081	92.33	91.74	92.04	96.30	93.11	**94.68**
LM-Attn-None	93.29	88.85	90.81	0.086	89.79	92.79	91.27	92.81	93.95	93.37

6 Conclusion

This paper presented approaches of neural exhaustive and neural contextual exhaustive models model that considers all possible regions exhaustively for named entity recognition and sensitive token detection. The model obtains the representation of each region using the outputs of the underlying shared LSTM layer, and it represents the different regions by concatenating boundary and inside representations of the region. Several enhancements, namely attention mechanism, LSTM-Minus, context from base representations, and context from LSTM-Minus are investigated for the representations. It then classifies the region into an 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 flat and nested entities from the generated mention candidates of all possible regions. Among the five submitted runs, the best run for each subtask achieved the F-score of 93.12% on Sub-task 1 and the F-scores of 93.52% (strict) and 94.92% (merged) on Sub-task 2 without any external knowledge resources.

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

1. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. In: ICLR 2015 (2015)
2. Collier, N., Park, H.S., Ogata, N., Tateisi, Y., Nobata, C., Ohta, T., Sekimizu, T., Imai, H., Ibushi, K., Tsujii, J.: The GENIA Project: Corpus-based Knowledge Acquisition and Information Extraction from Genome Research Papers. In: Proceedings of EACL. pp. 171–172. ACL (1999)
3. Feng, X., Huang, L., Tang, D., Ji, H., Qin, B., Liu, T.: A Language-Independent Neural Network for Event Detection. In: Proceedings of the 54th Annual Meeting of the ACL (Volume 2: Short Papers). pp. 66–71. Berlin, Germany (2016), http://anthology.aclweb.org/P16-2011.

4. Fragkou, P.: Applying named entity recognition and co-reference resolution for segmenting english texts. Progress in Artificial Intelligence **6**(4), 325–346 (2017), https://doi.org/10.1007/s13748-017-0127-3.
5. Gupta, N., Singh, S., Roth, D.: Entity Linking via Joint Encoding of Types, Descriptions, and Context. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing. pp. 2671–2680. ACL, Copenhagen, Denmark (2017), https://www.aclweb.org/anthology/D17-1284.
6. Ju, M., Miwa, M., Ananiadou, S.: 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). pp. 1446–1459. ACL, New Orleans, Louisiana (2018), http://www.aclweb.org/anthology/P16-1105
7. Kingma, D., Ba., J.: Adam: A Method for Stochastic Optimization. In: ICLR (2015)
8. Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., Dyer, C.: Neural Architectures for Named Entity Recognition. In: Proceedings of the 2016 Conference of the North American Chapter of the ACL: Human Language Technologies. ACL. vol. 1, pp. 260–270. ACL, San Diego, California (2016), http://www.aclweb.org/anthology/N16-1030.
9. Marimon, M., Gonzalez-Agirre, A., Intxaurrondo, A., Rodrguez, H., Lopez Martin, J.A., Villegas, M., Krallinger, M.: Automatic de-identification of medical texts in spanish: the meddocan track, corpus, guidelines, methods and evaluation of results. In: Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2019). vol. TBA, p. TBA. CEUR Workshop Proceedings (CEUR-WS.org), Bilbao, Spain (Sep 2019), TBA
10. Miwa, M., Bansal, M.: End-to-End Relation Extraction using LSTMs on Sequences and Tree Structures. In: Proceedings of the 54th Annual Meeting of the ACL. pp. 1105–1116. ACL, Berlin, Germany (2016), http://aclweb.org/anthology/N18-1131
11. Sohrab, M.G., Miwa, M.: Deep exhaustive model for nested named entity recognition. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. pp. 2843–2849. Association for Computational Linguistics, Brussels, Belgium (Oct-Nov 2018), https://www.aclweb.org/anthology/D18-1309
12. Son, N.T., Minh, N.L.: Nested Named Entity Recognition Using Multilayer Recurrent Neural Networks. In: Proceedings of PACLING 2017. pp. 16–18. Sedona Hotel, Yangon, Myanmar (2017)
13. Wang, W., Chang, B.: Graph-based dependency parsing with bidirectional LSTM. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 2306–2315. Association for Computational Linguistics, Berlin, Germany (Aug 2016). https://doi.org/10.18653/v1/P16-1218, https://www.aclweb.org/anthology/P16-1218
14. Zhou, G., Zhang, J., Su, J., Shen, D., Tan, C.: Recognizing Names in Biomedical Texts: a Machine Learning Approach. Bioinformatics **20**(7), 1178–1190 (2004), https://doi.org/10.1093/bioinformatics/bth060.