Hierarchical Nested Named Entity Recognition

Zita Marinho[†*] **Afonso Mendes**[†] **Sebastião Miranda**[†] **David Nogueira**[†]
zam@priberam.com, amm@priberam.com, ssm@priberam.com, dan@priberam.com

[†]Priberam Labs, Alameda D. Afonso Henriques, 41, 2º, 1000-123 Lisboa, Portugal
[*] Instituto de Sistemas e Robótica, Instituto Superior Técnico, 1049-001 Lisboa, Portugal

Abstract

In the medical domain and other scientific areas, it is often important to recognize different levels of hierarchy in entity mentions, such as those related to specific symptoms or diseases associated with different anatomical regions. Unlike previous approaches, we build a transition-based parser that explicitly models an arbitrary number of hierarchical and nested mentions, and propose a loss that encourages correct predictions of higher-level mentions. We further propose a set of modifier classes which introduces certain concepts that change the meaning of an entity, such as absence, or uncertainty about a given disease. Our model achieves state-of-the-art results in medical entity recognition datasets, using both nested and hierarchical mentions.

1 Introduction

One of the most common studied tasks in NLP lies in extracting semantic information from unstructured text in the form of entities and detecting entity mentions across a single document, in particular where the mention is located (its span) and its corresponding classification or entity semantic type, such as person (PER), location (LOC), organization (ORG), etc. The task of entity recognition has long been studied and applied to different higher level tasks such as question answering (Abney et al., 2000), coreference resolution (Fragkou, 2017), relation extraction (Mintz et al., 2009; Miwa and Bansal, 2016; Liu et al., 2017), entity linking (Gupta et al., 2017; Guo and Barbosa, 2014) and event extraction (Feng et al., 2016). Most of the existing work in Named Entity Recognition and Classification focuses on flat mentions, usually corresponding to the longest outer mention (Ling and Weld, 2012; Marcinczuk, 2015; Leaman and Lu, 2016), or using nested mentions that can capture overlapping mentions within different nested levels (Finkel and Manning, 2009; Lu and Roth, 2015; Wang et al., 2018; Ju et al., 2018). One of the main disadvantages of using simple independent classes to model different hierarchies is that there is no information that conveys an explicit hierarchical nature, in a way that lower level classes help to disambiguate the nature of higher level classes.

The most common approach to circumvent this issue involves projecting each lower level class to an individual label throwing away all of the inherent structure of the ontology. This approach is limited, since it does not propagate information to higher level classes and it does not use common information of all children in the ontology. The ability to identify hierarchical entities is very useful in many fields, in particular in the medical domain, where we associate medication, symptoms and other pathological conditions with more specific subtypes giving a more refined classification.

Additionally, we introduce the concept of modifier classes that can alter the meaning of a given class. Often, in medical records, the doctor states either the absence or presence of a particular condition, for that purpose we created a modifier level that acts on a particular class and is associated with the degree of relevance of that class, for example in the medical domain it may identify the absence or probability of certain symptoms/diseases, or refer to their duration (chronic, acute), etc. This concept is of particular use if we consider a hierarchical model to identify where this modifier actuates.

We test our model against other state-of-the-art methods modelling nested mentions whose classification is defined by their projected lower levels. We make use of hierarchical datasets in the medical field, where these notions are of extreme importance. We evaluate our model using the GENIA (Ohta et al., 2002) dataset, a bigger and more complex proprietary medical corpus (MED18) with higher hierarchical dependencies and modifier classes. To summarize, this paper

Proceedings of the 2nd Clinical Natural Language Processing Workshop, pages 28–34
Minneapolis, Minnesota, June 7, 2019. ©2019 Association for Computational Linguistics

makes the following contributions:

- we introduce a novel Hierarchical and Nested Named Entity Recognition (HNNER) model based on a neural transition based approach (Dyer et al., 2015), that is able to handle different levels of nested mentions and hierarchy,
- we further propose a model that can learn from modifier classes, allowing to model more complex and fine grained relations, such as degree of importance/variants of each class.
- we obtain state-of-the-art performance when compared with existing nested models with lower level projected labels (corresponding to the same hierarchical levels).

2 Related Work

Named entity recognition and classification has long been a popular task in NLP (Zhou and Su, 2002; McDonald et al., 2005; Ratinov and Roth, 2009; Wang et al., 2013). The first contribution on detecting nested mentions was proposed by Shen et al. (2003); Zhang et al. (2004); GuoDong (2004) and relied mostly on rule-based models. Later Finkel and Manning (2009) introduced a constituency parser as the first model-based approach for nested recognition, followed by work of Alex et al. (2007) using models based on linear-Conditional Random Fields (CRFs). Lu and Roth (2015); Muis and Lu (2017) handcrafted features to extract nested mentions without modelling their hidden dependencies using mention hypergraphs, that can capture nested dependencies with unbounded lengths.

With the success of neural based approaches for NER (Collobert et al., 2011; Chiu and Nichols, 2016; Ma and Hovy, 2016), several work has been done in classifying nested mentions: Ju et al. (2018) dynamically modeled each nested layer as a Long-Short-Term-Memory (LSTM)-CRF layer (Lample et al., 2016), requiring the knowledge of the number of nested overlapps to be known a priori. Katiyar and Cardie (2018) proposed a recurrent neural network to extract features to learn an hypergraph structure of nested mentions, using a BILOU encoding scheme. This required the creation of additional hyperarcs whenever a nested mention is encountered. More recently Wang et al. (2018) used a model based on a shift reduce parser that builds a forest structure for nested mentions. This neural approach can only be applied to classify nested mentions of different spans, meaning a single span cannot correspond to different mentions.

All of the proposed approaches so far, allow nested mentions classification but have never attempted to model explicit hierarchical and nested structures. Furthermore, our proposed model architecture is more expressive since it allows the same sequence of words to correspond to distinct mentions possibly with different hierarchical or nested levels.

3 Hierarchical Nested Named Entity Recognition (HNNER)

For a given input sequence of words $\{w_1, w_2, \ldots, w_n\}$ our model generates a sequence of actions that identifies nested and hierarchical mentions simultaneously.

Our transition-based model allows for several mentions to start and end at a given location in the sequence. We make use of an additional stack to store temporarily the terms corresponding to each mention, which we denote as *word stack*. The system state s is represented by a stack of words S containing all the temporary words pertaining to a mention (the word stack), a buffer of words to be parsed B, and a stack of actions corresponding to all mentions to be parsed M (the *mention stack*) and an output buffer that encode the entity mentions and other words O. Initially, we define the starting state as $s_0 = [M = \emptyset, S = \emptyset, B = \{w_1, \ldots, w_n\}], O = \emptyset$.

At each state, we apply an action a_n and change the state of the system s_n: by adding elements or resetting the word stack and moving the resulting mention to the output buffer, popping the top most word of the buffer and adding or popping actions from the mention stack. We consider four types of possible system actions $a \in \mathcal{A}$:

- OUT pops the top element of the buffer, and moves it to the output unaltered,
- SHIFT shifts the top element of the buffer to the word stack,
- TRANSITION(a) indicates the start of a mention, adds action label a to the mention stack,
- REDUCE(a) indicates the end of a mention and pops all elements of the mention stack until the last recorded transition and inserts the resulting mention (encoded as the output of an LSTM) in the output buffer. Since we only allow reductions of actions that remain in the top of the mention stack, we transition first to longer mentions, whenever more than one mention starts at the same point in the word sequence.

For each state of the system s_n we consider the subset of all possible valid actions $\mathcal{A}(a_{n-1}, s_n)$,

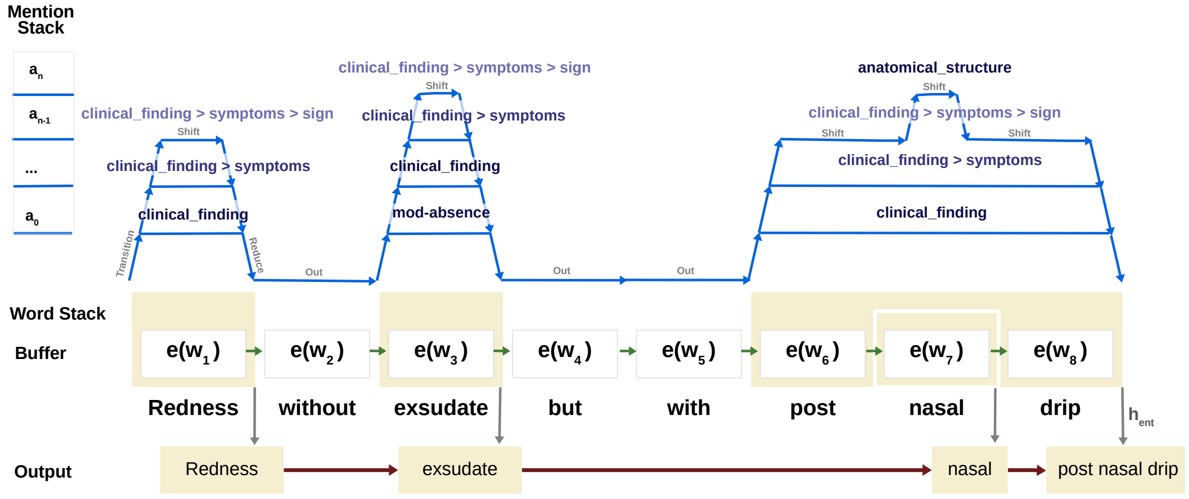

Figure 1: Transition-shift-reduce mechanism for hierarchical nested mention recognition. *Transition* is indicated by arrows pointing upwards, *Reduce* by downward arrows, *Out* horizontal arrows when mention stack is empty, and *Shift* action when non-empty. Different levels of the mention stack indicate the number of nested layers, while mention color indicates the hierarchical level (darker blue for level 0 and lighter as we go up in the hierarchy).

that depends on the previous action generated and the current parser state, in particular the mention stack. We consider a simple set of rules: for hierarchical mentions we only allow transitions to lower levels in the hierarchy if the upper levels exist in the mention buffer, meaning transitions of the form TRANSITION($a > b$) where the symbol $>$ indicates that b is a lower level hierarchy of class a and is only admitted if TRANSITION(a) exists in the mention stack. Our model allows an arbitrary number of hierarchies since, without knowing this number beforehand; we only allow reductions of the top most element in the mention stack, this step requires an ordering of nested mentions from longer to shorter spanning windows; we also only allow SHIFT actions if the mention stack is non-empty.

A mention containing a single word requires three actions to be considered: TRANSITION(a), SHIFT and REDUCE(a). Using this approach, we can model consecutive transitions of different mentions, multiple hierarchical as well as nested mentions, as long as they remain without overlaps.[1] For modifier classes, we model each individual modifier as a top level class. Figure 2 provides an example of a sequence of hierarchical and nested mentions. The terminal state is achieved when the word buffer is empty and all the elements of the mention stack have been reduced.

4 HNNER Model

Our transition-based model draws inspiration from the transition based parser proposed by Dyer et al. (2015). For a given sequence of input words $W = \{w_1, \dots, w_N\}$ we represent each word as a low dimensional vector $e(w_n) \in \mathbb{R}^{d_w}$ for each word in the vocabulary $w_n \in [V]$. To better capture morphological and orthographic features of words, we consider each word vector the product of concatenating a fixed word lookup embedding $l(w_n)$ with its learned character sequence representation $c(w_n)$, such that $e(w_n) = [l(w_n); c(w_n)]$. We compute the character embeddings using a bidirectional LSTM following work of Ma and Hovy (2016); Lample et al. (2016). We initialize character embeddings randomly, while each word embedding is retrieved from a pretrained look-up representation. For words out-of-vocabulary we consider the word's character based representation and we train a representation of the unknown word embedding.

We associate an LSTM with the word stack LSTM$_S(\{e(w_j)\}_{w_j \in S})$ whose inputs correspond to the words shifted from the buffer, another with the mention stack LSTM$_M(\{a_n\}_{a_n \in M})$ with inputs from mentions that the system initialized, and a last LSTM that models the output of the system LSTM$_O(\{e(o_n)\}_{o_n \in O})$, whose inputs correspond to the latest state of the word LSTM or the word embeddings, depending on whether the word is in the word stack or not, respectively. We start by filling the input buffer $B_0 = [w_n, w_n - 1, \dots, w_0]$ with the sequence of word embeddings

[1] We consider only non overlapping mentions disregarding any occurrences of the form TRANSITION(a)- SHIFT- TRANSITION(b)- SHIFT- REDUCE(a)- SHIFT- REDUCE(b).

Models	GENIA flat NER P	R	F1
Finkel et al. (2004)	71.62	68.56	70.06
GuoDong (2004)	75.99	69.42	72.55
HNNER	**76.11**	**69.43**	**72.62**

Table 1: Results on JNLPBA with flat mentions.

to be parsed in reverse order, and leave the first word at the top of the buffer. For a given state of the system $s_i = [M, S, B, O]$ we compute the system state representation $\boldsymbol{p}_i$ for each action i as a nonlinear transformation of the last LSTM state of the word stack $\boldsymbol{h}_w \in \mathbb{R}^{d_w}$, the last LSTM state of the mention stack $\boldsymbol{h}_m \in \mathbb{R}^{d_m}$ and the top most element of the buffer $\boldsymbol{b}_n \in \mathbb{R}^{d_w}$ and the last element of the output LSTM $\boldsymbol{o}_n \in \mathbb{R}^{d_o}$:

$$\boldsymbol{p}_i = \tanh(W[\boldsymbol{h}_w; \boldsymbol{h}_m; \boldsymbol{b}_n; \boldsymbol{o}_n] + \boldsymbol{b}),$$

with the bias $\boldsymbol{b} \in \mathbb{R}^k$ and linear weights $W \in \mathbb{R}^{(2d_w+d_m+d_o)\times k}$.

The system state $\boldsymbol{p}_i$ contains all the information required to make predictions about the current action of the parser $a_i \in \mathcal{A}$, according to a set of possible valid actions that we compute with simple rules $\mathcal{V}(a_{n-1}, s_n)$. Namely, we consider only as viable actions: SHIFT actions if it follows after a TRANSITION; REDUCE actions can only be applied in the reverse order of the previously applied TRANSITIONS; OUT actions are only allowed if there is no action to be reduced, and hierarchies must respect their parent transitions, meaning TRANSITION(a¿b) is not allowed if TRANSITION(a) has not been created first. Modifier classes are considered as a separate class of labels that may be applied in any hierarchical level.

The system greedily decides the current action based on:

$$p(a_n = a \mid \boldsymbol{p}_n) = \frac{\exp \boldsymbol{\alpha}_a^\top \boldsymbol{p}_n}{\sum_{a' \in \mathcal{V}} \exp \boldsymbol{\alpha}_{a'}^\top \boldsymbol{p}_n}$$

We train our model to maximize the log-likelihood of each action in a batch of M sequences:

$$\mathcal{L} = -\sum_{i=1}^{M} \sum_{n=1}^{N} \beta^{H-L(a_n)} \log p(a_n \mid \boldsymbol{p}_n),$$

weighted by a different value for each hierarchical level $\beta < 1$, where the level of each action $L(a_n) = 0$ for the top levels and decreases as we go down in the hierarchy, and H denotes the total number of levels.

Datasets	GENIA train	dev	test	MED18 train	dev	test
vocabulary	74,560			51,879		
pretrained vocab.	23,813			49,782		
sentences	13,416	3,147	1,656	73,099	4,216	4,018
mentions	35,506	8738	4,492	495,148	29,522	28,458
hier. L0	17,753	4,369	2,246	230,912	13,702	13,271
hier. L1	17,753	4,369	2,246	139,665	8,353	7,933
hier. L2	–	–	–	123,291	7,372	7,132
hier. L3	–	–	–	1,200	95	122
flat actions	5			26		
hier. actions	23			531		
hier. L0	5			66		
hier. L1	18			126		
hier. L2	–			325		
hier. L3	–			14		

Table 2: Dataset description: total number of mentions, sentences, words and actions. Number of mentions and types of actions per hierarchical layer

5 Experimental Results

Datasets: We compare our HNNER model using different nested and hierarchical scenarios. First, we compare against standard baselines for flat NER using the splits and the JNLPBA dataset (Gridach, 2017), considering only flat and the topmost entities in the GENIA dataset (Ohta et al., 2002), following the same splits and entity types used by Finkel and Manning (2009). We used the GENIA dataset (Ohta et al., 2002), consisting of 2000 MEDLINE abstracts with 36 fine-grained entity categories. We also employed the same conversion to the main 5 entity types (and left the DNA and RNA subtypes the hierarchical experiments). We used pretrained word embeddings for GENIA using PUBMED dataset.[2] We further tested on a more complex medical dataset MED18, [3] comprising 3000 documents of annotated clinical reports in Portuguese. We consider 4 levels of hierarchy and 531 fine-grained entity categories. We trained word embeddings for this dataset using word2vec (Mikolov et al., 2013) on over around 10M documents of clinical records.

Table 2 in 5 shows a description of the datasets. The MED18 dataset is larger and more complex than GENIA, containing a total of 509869 mentions, 531 different hierarchical classes with 4 levels of hierarchy, while GENIA altough initialy contains 36 fine-grained classes, we only report on 23 different classes with 2 levels of hierarchy.

Models and Baselines: We evaluate our HNNER model against state-of-the-art models for

[2] Embeddings available in `http://bio.nlplab.org/#source-data`

[3] a proprietary dataset for Portuguese Medical Diagnosis

Nested Models	Nested GENIA P	R	F1
Finkel and Manning (2009)	75.4	65.9	70.3
Lu and Roth (2015)	72.5	65.2	68.7
Muis and Lu (2017)	75.4	66.8	70.8
Wang et al. (2018)	**76.0**	69.4	71.6
HNNER	74.0	**72.0**	**73.0**

Table 3: Results on GENIA with nested mentions.

nested mentions: a CRF-based constituency parser (Finkel and Manning, 2009); a nested NER model using mention hypergraphs (Lu and Roth, 2015); a multigraph representation with mention separators for overlapping mentions (Muis and Lu, 2017); a neural layered model for each nested layer (Ju et al., 2018); and a neural shift-reduce neural parser for nested mentions (Wang et al., 2018). We also, evaluated HNNER against the non-hierarchical nested version with the same number of hierarchical levels projected as a different independent class (HNNER+SUB). We train our model using Adam gradient updates (Kingma and Ba, 2014) using a learning rate of 0.001 and a batch size of 32 sentences. We employed dropout of 0.1 on all input layers (Srivastava et al., 2014). We used $\beta = 0.8$ for GENIA and $\beta = 1.0$ for MED18. For higher level datasets this value should be closer to one in order to not overshadow the effect of lower hierarchies, which are often the most frequent ones.

Results Our HNNER model obtains state-of-the-art results when compared with other flat (Table 1) and nested NER models (Table 3).

Learning hierarchical mentions explicitly using our model (HNNER) achieves better performance than using a set of projected subcategories independently, (HNNER+SUB) in Table 4. The proposed approach is still able to perform well when we deal with higher levels of hierarchy and more nested classes, which we can observe in the results using the MED18 dataset. As we progress towards higher level hierarchies the gap performance increases between projected subclasses and explicit hierarchical modeling. The performance of level $L3$ drops when compared with lower level levels, because of the scarce number of existing mentions for this level (see §5).

Hierarchical Models	L2-GENIA P	R	F1	L3-MED18 P	R	F1
HNNER+SUB	69.3	64.5	66.8	73.2	71.7	72.5
HNNER+SUB-L0	73.5	68.4	70.9	74.4	71.3	72.8
HNNER+SUB-L1	65.1	60.6	62.8	72.7	72.7	72.7
HNNER+SUB-L2	-	-	-	72.1	72.1	72.1
HNNER+SUB-L3	-	-	-	37.5	36.9	37.2
HNNER	69.5	68.5	**70.0**	73.7	72.7	**73.2**
HNNER-L0	73.6	72.6	**73.1**	74.2	73.1	**73.6**
HNNER-L1	65.3	64.4	**64.8**	73.8	72.8	**73.3**
HNNER-L2	-	-	-	73.3	72.3	**72.8**
HNNER-L3	-	-	-	38.9	40.2	**39.5**

Table 4: Results on GENIA and MED18 with nested mentions with all the subcategories, and performance per hierarchical layer.

6 Conclusions and Future Work

We propose a hierarchical model based on a transition-based parser that is able to recognize hierarchical and nested mentions with undefined levels of complexity. We tested the performance of our model using two medical datasets GENIA and MED18, and reported state-of-the-art results on flat, nested and hierarchical datasets. We leave as future work extending this approach to more general overlapping mentions with non projective overlaps and exploiting schedule sampling techniques to make the algorithm less prone to errors during test-time.

References

Steven Abney, Michael Collins, and Amit Singhal. 2000. Answer extraction. In *Proceedings of the Sixth Conference on Applied Natural Language Processing*. Association for Computational Linguistics, Stroudsburg, PA, USA, ANLC '00, pages 296–301. https://doi.org/10.3115/974147.974188.

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*. Association for Computational Linguistics, Stroudsburg, PA, USA, BioNLP '07, pages 65–72.

Jason Chiu and Eric Nichols. 2016. Named entity recognition with bidirectional lstm-cnns. *Transactions of the Association for Computational Linguistics* 4:357–370. http://aclweb.org/anthology/Q16-1026.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *J. Mach. Learn. Res.* 12:2493–2537. http://dl.acm.org/citation.cfm?id=1953048.2078186.

Chris Dyer, Miguel Ballesteros, Wang Ling, Austin Matthews, and Noah A. Smith. 2015. Transition-based dependency parsing with stack long short-term memory. *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)* .

Xiaocheng Feng, Bing Qin, and Ting Liu. 2016. A language-independent neural network for event detection. *Science China Information Sciences* 61(9):092106. https://doi.org/10.1007/s11432-017-9359-x.

Jenny Rose Finkel, Shipra Dingare, Huy Nguyen, Malvina Nissim, Christopher D. Manning, and Gail Sinclair. 2004. Exploiting context for biomedical entity recognition: From syntax to the web. In *Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications, NLPBA/BioNLP 2004, Geneva, Switzerland, August 28-29, 2004*.

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*. Association for Computational Linguistics, Stroudsburg, PA, USA, EMNLP '09, pages 141–150.

Pavlina Fragkou. 2017. Applying named entity recognition and co-reference resolution for segmenting english texts. *Progress in Artificial Intelligence* 6. https://doi.org/10.1007/s13748-017-0127-3.

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

Zhaochen Guo and Denilson Barbosa. 2014. Robust entity linking via random walks. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*. ACM, New York, NY, USA, CIKM '14, pages 499–508. https://doi.org/10.1145/2661829.2661887.

Zhou GuoDong. 2004. Recognizing names in biomedical texts using hidden markov model and svm plus sigmoid. In *Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and Its Applications*. Association for Computational Linguistics, Stroudsburg, PA, USA, JNLPBA '04, pages 1–7. http://dl.acm.org/citation.cfm?id=1567594.1567596.

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*. Association for Computational Linguistics, pages 2681–2690. https://doi.org/10.18653/v1/D17-1284.

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)*. Association for Computational Linguistics, pages 1446–1459. https://doi.org/10.18653/v1/N18-1131.

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)*. Association for Computational Linguistics, pages 861–871. https://doi.org/10.18653/v1/N18-1079.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *CoRR* abs/1412.6980. http://arxiv.org/abs/1412.6980.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. *CoRR* abs/1603.01360.

Robert Leaman and Zhiyong Lu. 2016. Taggerone: joint named entity recognition and normalization with semi-markov models. *Bioinformatics* 32(18):2839–2846.

Xiao Ling and Daniel S. Weld. 2012. Fine-grained entity recognition. In *In Proc. of the 26th AAAI Conference on Artificial Intelligence*.

Liyuan Liu, Xiang Ren, Qi Zhu, Shi Zhi, Huan Gui, Heng Ji, and Jiawei Han. 2017. Heterogeneous supervision for relation extraction: A representation learning approach. *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing* https://doi.org/10.18653/v1/d17-1005.

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*. Association for Computational Linguistics, pages 857–867. https://doi.org/10.18653/v1/D15-1102.

Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via bi-directional lstm-cnns-crf. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, pages 1064–1074. https://doi.org/10.18653/v1/P16-1101.

Michal Marcinczuk. 2015. Automatic construction of complex features in conditional random fields for named entities recognition. In *Recent Advances in Natural Language Processing, RANLP 2015, 7-9 September, 2015, Hissar, Bulgaria*. pages 413–419.

Ryan McDonald, Fernando Pereira, Seth Kulick, Scott Winters, Yang Jin, and Pete White. 2005. Simple algorithms for complex relation extraction with applications to biomedical ie. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, Stroudsburg, PA, USA, ACL '05, pages 491–498. https://doi.org/10.3115/1219840.1219901.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, Curran Associates, Inc., pages 3111–3119. http://papers.nips.cc/paper/5021-distributed-representations-of-words-and-phrases-and-their-compositionality.pdf.

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*. Association for Computational Linguistics, Stroudsburg, PA, USA, ACL '09, pages 1003–1011. http://dl.acm.org/citation.cfm?id=1690219.1690287.

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 Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, pages 1105–1116. https://doi.org/10.18653/v1/P16-1105.

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*. Association for Computational Linguistics, pages 2608–2618. https://doi.org/10.18653/v1/D17-1276.

T. Ohta, Y. Tateisi, and J.D. Kim. 2002. The genia corpus: An annotated research abstract corpus in molecular biology domain. In *the Human Language Technology Conference*.

Lev Ratinov and Dan Roth. 2009. Design challenges and misconceptions in named entity recognition. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning*. Association for Computational Linguistics, Stroudsburg, PA, USA, CoNLL '09, pages 147–155. http://dl.acm.org/citation.cfm?id=1596374.1596399.

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*. Association for Computational Linguistics, Stroudsburg, PA, USA, BioMed '03, pages 49–56. https://doi.org/10.3115/1118958.1118965.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* 15(1):1929–1958. http://dl.acm.org/citation.cfm?id=2627435.2670313.

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

Mengqiu Wang, Wanxiang Che, and Christopher D. Manning. 2013. Effective bilingual constraints for semi-supervised learning of named entity recognizers. In *Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence*. AAAI Press, AAAI'13, pages 919–925.

Jie Zhang, Dan Shen, Guodong Zhou, Jian Su, and Chew-Lim Tan. 2004. Enhancing hmm-based biomedical named entity recognition by studying special phenomena. *J. of Biomedical Informatics* 37(6):411–422. https://doi.org/10.1016/j.jbi.2004.08.005.

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*. Association for Computational Linguistics, Stroudsburg, PA, USA, ACL '02, pages 473–480. https://doi.org/10.3115/1073083.1073163.