Neural Segmental Hypergraphs for Overlapping Mention Recognition

Bailin Wang
University of Massachusetts
Amherst
bailinwang@cs.umass.edu

Wei Lu
Singapore University of Technology
and Design
luwei@sutd.edu.sg

Abstract

In this work, we propose a novel *segmental hypergraph* representation to model overlapping entity mentions that are prevalent in many practical datasets. We show that our model built on top of such a new representation is able to capture features and interactions that cannot be captured by previous models while maintaining a low time complexity for inference. We also present a theoretical analysis to formally assess how our representation is better than alternative representations reported in the literature in terms of representational power. Coupled with neural networks for feature learning, our model achieves the state-of-the-art performance in three benchmark datasets annotated with overlapping mentions.[1]

1 Introduction

One of the most crucial steps towards building a natural language understanding system is the identification of basic semantic chunks in text. Such a task is typically characterized by the named entity recognition task (Grishman, 1997; Tjong Kim Sang and De Meulder, 2003), or the more general mention recognition task, where mentions are defined as references to entities that could be named, nominal or pronominal (Florian et al., 2004). The extracted mentions can be used in various downstream tasks for performing further semantic related tasks, including question answering (Abney et al., 2000), relation extraction (Mintz et al., 2009; Liu et al., 2017), event extraction (Riedel and McCallum, 2011; Li et al., 2013), and coreference resolution (Soon et al., 2001; Ng and Cardie, 2002; Chang et al., 2013).

One popular approach to the task of mention extraction is to regard it as a sequence labeling problem, with the underlying primary assumption being that the mentions are non-overlapping spans in the text. However, as highlighted in several prior research efforts (Alex et al., 2007; Finkel and Manning, 2009; Lu and Roth, 2015), mentions may overlap with one another in practice. Thus, models based on such a simplified assumption may result in sub-optimal performance for a down-stream task when they are deployed in practice. For example, consider a phrase "*At the Seattle zoo, ...*" shown in Figure 1, the relation LOCATEDIN between the mentions "*the Seattle zoo*" (of type FACILITY) and "*Seattle*" (of type GPE: Geo-political entities) will not be extracted unless both of these two overlapping mentions could be extracted. Similarly, there are 4 mentions of the same type (PROTEIN) in the text span "... *PEBP2 alpha A1, alpha B1 ...*" taken from the biomedical domain. A downstream question answering system may fail to return the correct answer as desired, if the mention extraction system it relies on is unable to extract all these valid mentions.

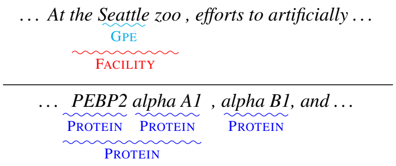
... At the Seattle zoo , efforts to artificially ...
GPE
FACILITY
... PEBP2 alpha A1 , alpha B1, and ...
PROTEIN
PROTEIN
PROTEIN
PROTEIN

Figure 1: Examples of overlapping mentions.

Various approaches to extracting overlapping mentions have been proposed in the past decade. The cascaded approach (Alex et al., 2007) builds a pipeline of sequence labeling models using conditional random fields (CRF) (Lafferty et al., 2001). However, the model is unable to handle overlapping mentions of the same type. Finkel and Manning (2009) presented a parsing based approach to nested mention extraction. Due to the chart-based parsing algorithm involved, the model has a cubic time complexity in the number of words in the sen-

[1] We make our system and code available at: http://statnlp.org/research/ie

Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 204–214
Brussels, Belgium, October 31 - November 4, 2018. ©2018 Association for Computational Linguistics

tence. A recent approach by Lu and Roth (2015) introduced a hypergraph representation for capturing overlapping mentions, which was shown fast and effective. The work was improved by Muis and Lu (2017), who proposed a sequence labeling approach that assigns tags to gaps between words. However, both approaches suffer from the *structural ambiguity* issue during inference, as we will further discuss in this paper.

We summarize our contributions as:

1. We propose a novel *segmental hypergraph* representation that is capable of modeling arbitrary combinations of (potentially overlapping) mentions in a given sentence. The model has a $\mathcal{O}(cmn)$ time complexity (m is the number of mention types, n is the number of words in a sentence, and c is the maximal number of words for each mention), and is able to capture features that cannot be captured by existing approaches.
2. Theoretically, we show that our approach based on such a new representation does not have the limitations associated with some recently proposed state-of-the-art approaches for overlapping mention extraction.
3. We show through extensive experiments on standard data that by exploiting both word-level and span-level features learned from neural networks, our model is able to achieve the state-of-the-art performance for recognizing overlapping mentions.

Our model is also general and robust. Further experiments show that our model yields competitive results when evaluated on data that does not have overlapping mentions annotated when comparing against other recently proposed state-of-the-art neural models that are capable of extracting non-overlapping mentions only.

2 Related Work

Overlapping Mention Recognition

One of the earliest research efforts on handling overlapping mentions is a rule-based approach (Zhang et al., 2004; Zhou et al., 2004; Zhou, 2006) that is evaluated on the GENIA dataset (Kim et al., 2003). The authors first detected the innermost mentions and then relied on rule-based post-processing methods to identify overlapping mentions. McDonald et al. (2005) presented a multilabel classification algorithm to model overlapping segments in a sentence systematically.

Alex et al. (2007) proposed several ways to combine multiple conditional random fields (CRF) (Lafferty et al., 2001) for such tasks. Their best results were obtained by cascading several CRF models in a specific order while each model is responsible for detecting mentions of a particular type. Outputs of one model can also serve as features to the next model. However, such an approach cannot model overlapping mentions of the same type, which frequently appear in practice.

Finkel and Manning (2009) approached this task from a parsing perspective by constructing a constituency tree, mapping each mention to a node in the tree. This approach assumes one mention is *contained by* the other when they overlap. While such an assumption largely holds in practice, it comes with a cost – the chart-based parser suffers from its cubic time complexity, making it not scalable to large datasets involving long sentences. Based on the same idea, Wang et al. (2018) proposed a scalable transition-based approach to construct a constituency forest (a collection of constituency trees).

Instead of relying on structured models, Xu et al. (2017) proposed a local classifier for each possible span. However, this local approach is unable to capture the interactions between spans. Similar to (Alex et al., 2007), Ju et al. (2018) dynamically stacked multiple flat layers which recognize mentions sequentially from innermost mentions to outermost mentions.

Our work is inspired by the model of Lu and Roth (2015), who introduced a *mention hypergraph* representation for capturing overlapping mentions. Their model was shown fast and effective, and was improved by the mention separator model (Muis and Lu, 2017). However, we note that (as also highlighted in their papers) both models suffer from the *structural ambiguity* issue during inference, which we will discuss later. Our new representation does not have this limitation.[2] Recently, Katiyar and Cardie (2018) also proposed a hypergraph-based representation based on the BILOU tagging scheme. Their model is trained greedily using neural networks by viewing the hypergraph construction procedure as a multi-label assignment process.

Neural Models for Mention Recognition

Recently, neural network based approaches to entity or mention recognition have received signifi-

[2] A model comparison can be found later in Table 1.

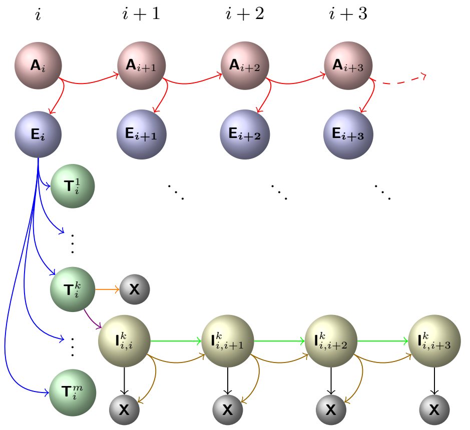

Figure 2: An example of partial segmental hypergraph (hyperedges of different types in different colors).

cant attention. They have been proven effective, even in the absence of handcrafted features. Collobert et al. (2011) used convolutional neural networks (CNN) over word sequences, paired with a CRF output layer. Huang et al. (2015) replaced the CNN with a bidirectional long short-term memory network (LSTM) (Hochreiter and Schmidhuber, 1997). Strubell et al. (2017) proposed an iterated dilated CNN to improve computational efficiency. Beyond word-level compositions, several methods incorporated character-level compositions with character embeddings, either through CNN (Chiu and Nichols, 2016; Ma and Hovy, 2016) or LSTM (Lample et al., 2016).

3 Segmental Hypergraph

A segmental hypergraph is a representation that aims at representing all possible combinations of (potentially overlapping) mentions in a given sentence. It belongs to a class of directed hypergraphs (Gallo et al., 1993), where each hyperedge e consists of a single designated parent node (*head* of e) and an ordered list of child nodes (*tail* of e). Specifically, our segmental hypergraph consists of the following 5 types of nodes:

- A_i encodes all such mentions that start with the i-th or a later word
- E_i encodes all mentions that start exactly with the i-th word
- T_i^k represents all mentions of type k starting with the i-th word
- $\mathsf{I}_{i,j}^k$ represents all mentions of type k that contain the j-th word and start with the i-th word
- X marks the end of a mention.

Hyperedges connecting these nodes are designed to indicate how the semantics of a parent node can be re-expressed in terms of its child nodes. Figure 2 gives a partial segmental hypergraph representing all combinations of mentions within the span $[i, i+3]$ consisting of 4 words. There are 4 types of hyperedges:

1. A hyperedge $\{\mathsf{A}_i \rightarrow (\mathsf{A}_{i+1}, \mathsf{E}_i)\}$ from A_i to its children implies the fact that A_i consists of those mentions that either "start exactly with the i-th word" (E_i), or "start with a word that appears strictly after the i-th word" (A_{i+1}).
2. A hyperedge $\{\mathsf{E}_i \rightarrow (\mathsf{T}_i^1, \ldots, \mathsf{T}_i^m)\}$ from E_i to its children implies that we should consider all possible types for the mentions (possibly of length 0) that start with the i-th word.
3. Two hyperedges $\{\mathsf{T}_i^k \rightarrow \mathsf{I}_{i,i}^k\}$ and $\{\mathsf{T}_i^k \rightarrow \mathsf{X}\}$ from T_i^k indicate that either there exists at least one mention starting with the i-th word (the former hyperedge), or there does not exist any such mention (the latter hyperedge).
4. Three hyperedges $\{\mathsf{I}_{i,j}^k \rightarrow \mathsf{I}_{i,j+1}^k\}$, $\{\mathsf{I}_{i,j}^k \rightarrow \mathsf{X}\}$, and $\{\mathsf{I}_{i,j}^k \rightarrow (\mathsf{I}_{i,j+1}^k, \mathsf{X})\}$ from $\mathsf{I}_{i,j}^k$ indicate the following three cases respectively: 1) both the j-th and $(j+1)$-th words belong to at least one mention that starts with the i-th word, 2) there exists one mention that starts with the i-th word and ends with the j-th word, and 3) both cases are valid.

Essentially, the complete hypergraph compactly encodes the whole search space of all possible mentions that can ever appear within a sentence, where such mentions may or may not overlap with one another. When we traverse the complete segmental hypergraph by following the directions as specified by the hyperedges, selecting only one outgoing hyperedge at a time at each node, we arrive at a *hyperpath*[3] – a rooted, directed substructure contained by the original hypergraph.

Figure 3 shows an example. Here, "*Israeli UN Ambassador*" of type PERSON is captured by the following sequence of nodes (along a hyperpath): "A_1, E_1, T_1^2, $\mathsf{I}_{1,1}^2$, $\mathsf{I}_{1,2}^2$, $\mathsf{I}_{1,3}^2$, X", while "*Israeli UN Ambassador Danny*" of type PERSON corresponds to the following node sequence: "A_1, E_1, T_1^2, $\mathsf{I}_{1,1}^2$, $\mathsf{I}_{1,2}^2$, $\mathsf{I}_{1,3}^2$, $\mathsf{I}_{1,4}^2$, X". Similarly, the following sequence "A_1, A_2, E_2, T_2^1, $\mathsf{I}_{2,2}^1$, X" represents the mention "*UN*" of type ORGANIZATION. As we can see, such node sequences together form a single hyperpath that encodes this specific combination of mentions that overlap with one another.

[3] Each hyperpath is a *hypertree* (Brandstädt et al., 1998).

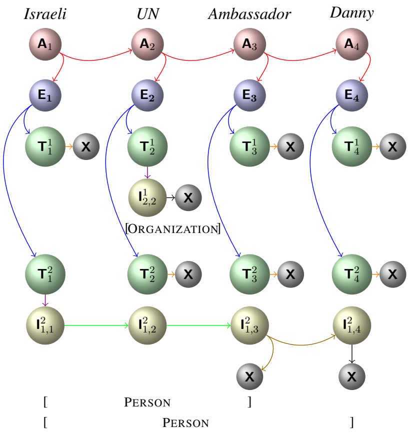

Figure 3: A specific hyperpath for encoding three mentions. For brevity, we only show two types.

More details on segmental hypergraph and hyperpaths are in the supplementary material.

Theoretical Analysis

Our proposed segmental hypergraph representation has the following theoretical property:

Theorem 3.1. (Structural Ambiguity Free) *For any sentence and its segmental hypergraph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, let $\mathcal{S}$ be the set of all possible mention combinations for the given sentence, and $\mathcal{P}$ be the set of all hyperpaths contained by $\mathcal{G}$, there is a one-to-one correspondence between elements in $\mathcal{P}$ and $\mathcal{S}$.*

Due to space, we provide a proof sketch and include more details in the supplementary material. **Proof Sketch** We note that each hyperpath is uniquely characterized by its collection of hyperedges that involve **X** nodes. These hyperedges uniquely determine the collection of mentions. Conversely, a collection of mentions can be uniquely characterized by a collection of such hyperedges, which yields a unique hyperpath. □

Note that such a theorem states that our novel representation has no *structural ambiguity*, a nice property that both mention hypergraph model of (Lu and Roth, 2015) and mention separator model of (Muis and Lu, 2017) do not hold. As the authors have mentioned in their papers, for a given sub-structure in their model, there exist multiple ways of interpreting the combination of mentions. Specifically, in both representations, the decisions on where the *beginning* and the *end* of a mention are made *locally*. Such a design will lead to the structural ambiguity as there will be multiple interpretations to the mentions given a particular collection of positions marked as beginning and end of mentions. To illustrate, consider a phrase with 4 words "*A B C D*" where there are only two overlapping mentions "*B C*" and "*A B C D*". In both of the previous approaches, their models would make local predictions and assign both "*A*" and "*B*" as left boundaries, and both "*C*" and "*D*" as right boundaries. However, based on such local predictions one could also interpret "*A B C*" as a mention – this is where the ambiguity arises. In contrast, our model enjoys the structural ambiguity free property as it uses our newly defined **I** nodes (together with **X** nodes) to jointly capture the complete boundary information of mentions. Table 1 shows a full comparison. [4]

4 Learning

We adopt a log-linear approach to model the conditional probability of each hyperpath as follows:

$$p(\boldsymbol{y}|\boldsymbol{x}) = \frac{\exp f(\boldsymbol{x}, \boldsymbol{y})}{\sum_{\boldsymbol{y}'} \exp f(\boldsymbol{x}, \boldsymbol{y}')} \tag{1}$$

where $f(\boldsymbol{x}, \boldsymbol{y})$ is the score function for any pair of input sentence $\boldsymbol{x}$ and output mention combination $\boldsymbol{y}$, which corresponds to a unique hyperpath $\mathcal{G}_{\boldsymbol{y}}$. Our objective is to minimize the negative log-likelihood of all instances in the training set $\mathcal{D}$:

$$- \sum_{(\boldsymbol{x}, \boldsymbol{y}^*) \in \mathcal{D}} \log p(\boldsymbol{y}^* | \boldsymbol{x}) \tag{2}$$

We define features over each hyperedge within the hyperpath $\mathcal{G}_{\boldsymbol{y}}$. The score function can be decomposed into the following form:

$$f(\boldsymbol{x}, \boldsymbol{y}) = \sum_{e \in \mathcal{G}_{\boldsymbol{y}}} \psi(e, \boldsymbol{x}) \tag{3}$$

where $e \in \mathcal{G}_{\boldsymbol{y}}$ denotes a hyperedge that appears within the hyperpath $\mathcal{G}_{\boldsymbol{y}}$, and $\psi(e, \boldsymbol{x})$ is a score defined over e when the input sentence is $\boldsymbol{x}$.

Apart from *word-level* features, the segmental hypergraph also allows *span-level* features to be defined. The node $\mathbf{I}_{i,j}^k$ corresponds to a particular span $[i, j]$ over which we can extract our local features. The hyperedge between **I** nodes can capture the interactions between partial mentions and hyperedge between $\mathbf{I}_{i,j}^k$ and **X** precisely represents the mention $[i, j]$ with type k. We note that such features and interactions cannot be captured by the models of (Lu and Roth, 2015) and (Muis and Lu, 2017). Such a unique property makes our segmental hypergraph model more expressive than theirs.

[4] The mention hypergraph (Lu and Roth, 2015) also suffers from the *spurious structures* issue, while we do not. We refer the readers to (Muis and Lu, 2017) for details.

	Spurious Structures	Structural Ambiguity	Only Nested Mentions	Pipeline Approach	Different Types Only	Time Complexity
Alex et al. (2007)	NO	NO	NO	YES	YES	$\mathcal{O}(mn)$
Finkel & Manning (2009)	NO	NO	YES	NO	NO	$\mathcal{O}(\lvert G \rvert n^3)$
Lu & Roth (2015)	YES	YES	NO	NO	NO	$\mathcal{O}(mn)$
Muis & Lu (2017)	NO	YES	NO	NO	NO	$\mathcal{O}(mn)$
Wang et al. (2018)	NO	NO	YES	NO	NO	$\mathcal{O}(mn)$
This work	**NO**	**NO**	**NO**	**NO**	**NO**	$\mathcal{O}(cmn)$

Table 1: Model comparison. $|G|$ is the number of rules in grammar G.

4.1 Softmax-Margin Training

Inspired by (Mohit et al., 2012), we consider the softmax-margin (Gimpel and Smith, 2010) in our model. The function $\psi(e, \boldsymbol{x})$ is defined as follows:

$$\psi(e, \boldsymbol{x}) = \phi(e, \boldsymbol{x}) + \Delta(e, \mathcal{G}_{\boldsymbol{y}^*}) \tag{4}$$

where $\phi(e, \boldsymbol{x})$ is a feature function, and $\Delta(e, \mathcal{G}_{\boldsymbol{y}^*})$ is the cost function that defines the margin:

$$\Delta(e, \mathcal{G}_{\boldsymbol{y}^*}) = \begin{cases} \beta & \mathrm{TX}[e] \wedge e \notin \mathcal{G}_{\boldsymbol{y}^*} \\ 1 & \mathrm{TI}[e] \wedge e \notin \mathcal{G}_{\boldsymbol{y}^*} \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

Here, $\boldsymbol{y}^*$ is the gold mention combination, and $\mathrm{TX}[e]$ and $\mathrm{TI}[e]$ are indicator functions that return true if e is between T and X and between T and I respectively, and false otherwise. We set $\beta \geq 1$ such that the cost function will assign more penalty to false negatives than to false positives.

4.2 Feature Representation

We use two bidirectional LSTMs to learn word-level and span-level feature representations that can be used in our approach, resulting in our *neural segmental hypergraph* model. We first map the i-th word in a sentence to its pre-trained word embedding $\mathbf{e}_i$, and its POS tag to its embedding $\mathbf{p}_i$ if it exists. The final representation for i-th word is the concatenation of them: $\mathbf{v}_i = [\mathbf{e}_i, \mathbf{p}_i]$. Next, we use the a bidirectional LSTM to capture context-specific information for each word, resulting in the word-level features:

$$\mathbf{h}_i^w = [\mathrm{biLSTM}_1(\mathbf{v}_0, ..., \mathbf{v}_n)]_i \tag{6}$$

Such representations are then used as inputs to a second LSTM to generate span-level features: Inspired by (Kong et al., 2016), we compute all possible span embeddings efficiently with time complexity $\mathcal{O}(cn)$ using dynamic programming, with n being the number of words in the input $\boldsymbol{x}$ and c being the maximal length of a mention.

$$\mathbf{h}_{i:j}^s = \mathrm{biLSTM}_2(\mathbf{h}_i^w, ..., \mathbf{h}_j^w) \tag{7}$$

Recall that there are 4 types of hyperedges in our hypergraph, over which we can define the score functions. Since every valid mention hyperpath contains the first and second kind of hyperedges, defining scores over such hyperedges are unnecessary as their scores would serve as a constant factor that can be eliminated in the overall loss function of the log-linear model. Thus we only need to define the score functions on the latter two types of hyperedges. For hyperedges that only involve two nodes, we use a linear layer to compute their scores:

$$\phi(\{\mathsf{T}_i^k \to \mathsf{X}\}, \boldsymbol{x}) = \mathbf{W}_{\mathrm{TX}}^{(k)} \cdot \mathbf{h}_i^w \tag{8}$$

$$\phi(\{\mathsf{T}_i^k \to \mathsf{I}_{i,j}^k\}, \boldsymbol{x}) = \mathbf{W}_{\mathrm{TI}}^{(k)} \cdot \mathbf{h}_i^w \tag{9}$$

$$\phi(\{\mathsf{I}_{i,j}^k \to \mathsf{I}_{i,j+1}^k\}, \boldsymbol{x}) = \mathbf{W}_{\mathrm{II}}^{(k)} \cdot [\mathbf{h}_{i:j}^s, \mathbf{h}_{i:j+1}^s] \tag{10}$$

$$\phi(\{\mathsf{I}_{i,j}^k \to \mathsf{X}\}, \boldsymbol{x}) = \mathbf{W}_{\mathrm{IX}}^{(k)} \cdot \mathbf{h}_{i:j}^s \tag{11}$$

where matrices $\mathbf{W}_{\mathrm{TX}}, \mathbf{W}_{\mathrm{TI}} \in \mathbb{R}^{d_1 \times m}$, $\mathbf{W}_{\mathrm{II}} \in \mathbb{R}^{2d_2 \times m}$, $\mathbf{W}_{\mathrm{IX}} \in \mathbb{R}^{d_2 \times m}$, with superscript (k) referring to the k-th column of the matrix, d_1 is the dimension of $\mathbf{h}^w$, d_2 is the dimension of $\mathbf{h}^s$, and m is the number of mention types.

For the hyperedges that involve more than two nodes, the score is computed as follows:

$$\begin{aligned} &\phi(\{\mathsf{I}_{i,j}^k \to (\mathsf{X}, \mathsf{I}_{i,j+1}^k)\}, \boldsymbol{x}) \\ &= \mathbf{W}_{\mathrm{II}}^{\prime(k)} \cdot [\mathbf{h}_{i:j}^s, \mathbf{h}_{i:j+1}^s] + \mathbf{W}_{\mathrm{IX}}^{\prime(k)} \cdot \mathbf{h}_{i:j}^s \end{aligned} \tag{12}$$

where $\mathbf{W'}_{\mathrm{II}} \in \mathbb{R}^{2d_2 \times m}$, $\mathbf{W'}_{\mathrm{IX}} \in \mathbb{R}^{d_2 \times m}$. Note that in this work, we set $\mathbf{W'}_{\mathrm{II}} = \mathbf{W}_{\mathrm{II}}$ and $\mathbf{W'}_{\mathrm{IX}} = \mathbf{W}_{\mathrm{IX}}$ to reduce the number of free parameters.

Learning uses stochastic gradient descent with the update rule of Adam (Kingma and Ba, 2014) and a gradient clipping of 3.0. Dropout (Srivastava et al., 2014) for input vectors $\mathbf{v}$ and ℓ_2 regularization are used to reduce overfitting; both are tuned during the development process.

4.3 Character-level Representation

To make fair comparisons with recent models (Ju et al., 2018; Wang et al., 2018) that additionally incorporate character-level components in capturing orthographic and morphological features of words, we follow Lample et al. (2016) to use a bidirectional LSTM that takes the character embeddings as input. Specifically, the character-level representation $\mathbf{ch}_i$ for each word is obtained by concatenating the last hidden vectors of the forward and backward LSTMs. When this component is activated, the representation of each word is changed to: $\mathbf{v}_i = [\mathbf{e}_i, \mathbf{p}_i, \mathbf{ch}_i]$.

5 Inference

Inference can be done efficiently using a generalized inside-outside style message-passing algorithm (Baker, 1979). The partition function of (1) can be computed using the inside algorithm applied to the complete hypergraph $\mathcal{G}$, where we traverse from leaf nodes X to the root node A_1, passing messages to a parent node $\mathbf{p}$ from its child nodes:

$$\mu[\mathbf{p}] \leftarrow \log\Big(\sum_{e:h(e)\equiv\mathbf{p}} \exp\big(\psi(e,\boldsymbol{x}) + \sum_{\mathbf{c}\in\mathcal{T}(e)}\mu[\mathbf{c}]\big)\Big) \tag{13}$$

where $h(e)$ is the head of the hyperedge e, and $\mathcal{T}(e)$ is the collection of nodes that form the tail of e – they are the child nodes of $h(e)$ given e. The message passing step for the outside algorithm can be defined analogously. It can be verified that such a message passing algorithm, that is analogous to the sum-product belief propagation algorithm (Kschischang et al., 2001) used in standard graphical models, will converge after one forward and one backward pass.

For decoding, we perform the standard MAP inference on top of the complete hypergraph to find the most probable hyperpath. The resulting procedure is similar to the max-product message passing algorithm, where we consider only the feature function ϕ for constructing the messages:

$$\mu[\mathbf{p}] \leftarrow \max_{e:h(e)\equiv\mathbf{p}} \Big(\phi(e,\boldsymbol{x}) + \sum_{\mathbf{c}\in\mathcal{T}(e)} \mu[\mathbf{c}]\Big) \tag{14}$$

During inference, each node corresponds to a sum/max computation. Since one node is incident to 3 hyperedges maximally, the time complexity of inference algorithm can be implied by the number of nodes in the graph, which is $\mathcal{O}(cmn)$, where c is the maximal length for any mention. This complexity is the same as that of a zero-th order semi-Markov CRF model (Sarawagi and Cohen, 2005). Please refer to the supplementary material for a detailed explanation of the inference algorithm.

	ACE-2004		GENIA	
	Train (%)	Test (%)	Train (%)	Test (%)
# sentences	6,799	879	14,836	1,855
with o.l.	2,683 (39)	272 (42)	3,199 (22)	448 (24)
# mentions	22,207	3,031	46,473	5,600
o.l.	10,170 (46)	1,418 (47)	8,337 (18)	1,217 (22)
o.l. (st)	5,431 (24)	780 (26)	4,613 (10)	634 (11)
o.l. (st & slb)	2,188 (10)	307 (10)	2,133 (5)	287 (5)
lengh > 6	1,439 (6)	199 (7)	2,449 (5)	301 (5)
max lengh	57	43	28	19

Table 2: Statistics (ACE04, GENIA). *o.l.*: overlapping mentions, *st/slb*: same type/left boundary.

6 Experiments

6.1 Datasets

We mainly evaluate our models on the standard ACE-2004, ACE-2005 (Doddington et al., 2004), and GENIA (Kim et al., 2003) datasets with the same splits used by previous works (Lu and Roth, 2015; Muis and Lu, 2017). Sample data statistics of these datasets are listed in Table 2. [5] We can see that overlapping mentions frequently appear in such datasets. For ACE2004, over 46% of the mentions overlap with one another. GENIA focuses on biomedical entity recognition[6] and overlapping mentions are also common in it. Most mentions (over 93%) are not longer than 6 tokens which we select as maximal length (c) for the restricted models.

6.2 Baseline Approaches

We consider the following baseline models:

- CRF (LINEAR): a linear-chain CRF model. Since the linear-chain CRF cannot handle overlapping structures, we only use the outer-most mentions for learning. Specifically, every outer-most mention is labeled based on the BILOU tagging scheme, which was empirically shown to be better than the BIO scheme (Ratinov and Roth, 2009).
- CRF (CASCADED): the cascaded CRF based approach following (Alex et al., 2007). Note that this approach cannot model the overlapping mentions of the same type.
- Semi-CRF: the semi-Markov CRF model (Sarawagi and Cohen, 2005). The semi-CRF model is also only trained on the outer-most mentions. It can also capture span-level fea-

[5] See supplementary material for complete data statistics.

[6] Following previous works, we used version 3.02p which comes with annotated POS tags (Tateisi, 2004) . Following (Finkel and Manning, 2009), we collapse *DNA*, *RNA* and *protein* subtypes into *DNA*, *RNA* and *protein* respectively, keep cell line and cell type and remove mentions of other types.

		ACE-2004			ACE-2005			GENIA		
		P	R	F_1	P	R	F_1	P	R	F_1
Non-Neural	CRF (LINEAR)	71.8	40.8	52.1	69.5	44.5	54.2	77.1	63.3	69.5
	CRF (CASCADED)	78.4	46.4	58.3	74.8	49.1	59.3	75.9	66.1	70.6
	Semi-CRF (c=6)	76.1	41.4	53.6	72.8	45.0	55.6	74.5	66.0	70.0
	Semi-CRF (c=n)	66.7	42.0	51.5	67.5	46.1	54.8	74.2	65.8	69.7
	Finkel and Manning (2009)	-	-	-	-	-	-	75.4	65.9	70.3
	Lu and Roth (2015)	70.0	56.9	62.8	66.3	59.2	62.5	74.2	66.7	70.3
	Muis and Lu (2017)	72.7	58.0	64.5	69.1	58.1	63.1	75.4	66.8	70.8
	SH (-NN, c=6)	69.4	57.0	62.0	70.3	55.8	62.2	77.0	66.1	71.1
	SH (-NN, c=n)	71.1	60.6	65.4	69.5	60.7	64.8	76.2	67.5	71.6
Neural	FOFE (Xu et al., 2017) (c=6)	68.2	54.3	60.5	67.4	55.1	60.6	71.2	64.3	67.6
	FOFE (Xu et al., 2017) (c=n)	57.3	46.8	51.5	56.3	44.6	49.8	63.2	59.3	61.2
	Katiyar and Cardie (2018)	73.6	71.8	72.7	70.6	70.4	70.5	**79.8**	68.2	73.6
	Ju et al. (2018) [7]	-	-	-	74.2	70.3	72.2	78.5	71.3	74.7
	Wang et al. (2018)	74.9	71.8	73.3	74.5	71.5	73.0	78.0	70.2	73.9
	SH (c=6)	79.1	67.3	72.7	75.7	69.6	72.5	76.6	71.0	73.7
	SH (c=6) + *char*	**80.1**	67.5	73.3	75.9	70.0	72.8	76.8	71.8	74.2
	SH (c=n)	77.7	72.1	74.5	76.6	71.9	74.2	76.1	72.9	74.5
	SH (c=n) + *char*	78.0	**72.4**	**75.1**	**76.8**	**72.3**	**74.5**	77.0	**73.3**	**75.1**

Table 3: Main results. SH: segmental hypergraphs (our approach).

tures defined over a complete segment. Similar to our model, semi-CRF typically comes with a length restriction (c) which indicates the maximal length of a mention.

- Finkel and Manning (2009): a parsing-based approach for recognizing nested mentions that reported results on the GENIA dataset.
- Lu and Roth (2015): the model that makes use of mention hypergraphs for recognizing overlapping mentions.
- Muis and Lu (2017): the model that makes use of mention separators to tag gaps between words for recognizing overlapping mentions.
- FOFE (Xu et al., 2017): a local classifier based on neural networks that runs on every possible span to detect mentions. The maximal mention length (c) can also be used here.
- Katiyar and Cardie (2018): a hypergraph-based model that uses LSTM for learning feature representations.
- Ju et al. (2018): a cascaded model that makes use of multiple LSTM-CRF layers to recognize mentions in an inside-out manner.
- Wang et al. (2018): a neural transition-based model that construct nested mentions through a sequence of actions.
- SH (-NN): a non-neural version of our segmental hypergraph model that excludes the LSTMs but employs handcrafted features. [8]

As discussed earlier, we also evaluate the variants of our model that takes character-level representations (+*char*).

	ACE-2004		ACE-2005		GENIA	
	(c=6)	(c=n)	(c=6)	(c=n)	(c=6)	(c=n)
SH	72.7	74.5	72.5	74.2	73.7	74.5
-D	71.5	73.1	71.3	72.9	72.1	72.8
-SM	72.0	73.3	71.8	73.5	72.4	73.3
-P	71.5	72.7	71.2	73.0	72.0	73.2

Table 4: Results of various ablations. D: dropout, SM: softmax-margin, P: pre-trained embeddings.

6.3 Training

Pre-trained embeddings GloVe (Pennington et al., 2014) of dimension 100 are used to initialize the trainable word vectors for experiments in ACE and GENIA datasets.[9] The embeddings for POS tags are initialized randomly with dimension 32. Early stopping is used based on the performance of development set. The value β used in softmax-margin is chosen from [1, 3] with step size 0.5.

6.4 Experimental Results

Main results can be found in Table 3. Using the same set of handcrafted features, our unrestricted non-neural model SH (-NN, c=n) achieves the best performance compared with other non-neural models, revealing the effectiveness of our newly proposed segmental hypergraph representation. It achieves around 1-2% gain in terms of F_1 compared with mention hypergraph of Lu and Roth (2015) and mention separator of Muis and Lu (2017), showing the necessity of eliminating structural ambiguity. CRF (LINEAR) and Semi-CRF do not perform well due to incapability of handling

[7] Note that in ACE2005, Ju et al. (2018) did their experiments with a different split than Lu and Roth (2015) which we follow as our split.

[8] To make a proper comparison, we use the same handcrafted features used by (Lu and Roth, 2015), which were proven effective in previous approaches.

[9] We also additionally tried using embeddings trained on PubMed for GENIA but the performance was comparable.

overlapping mentions. In contrast, the pipeline approach CRF (CASCADED) performs better.

Our unrestricted neural segmental hypergraph model SH (c=n) already achieves the best results among all previous models in ACE datasets, showing the effectiveness of our neural segmental hypergraph. The improvement mainly comes from its ability to recall more mentions. In GENIA, even without using external features like Brown clustering features as all non-neural models do, our neural models still get significant improvements. Compared with the non-neural SH (-NN) which has around 4.2M parameters, our neural model SH only has 1.9M parameters yet it still performs better. We empirically see that the representations learned by LSTM can better capture complex contextual dependencies in sentences. The character-level representations (+ *char*) make both restricted and unrestricted SH perform even better. Particularly, SH (c=n) + *char* achieves the best results in all datasets compared with other recent neural models (Katiyar and Cardie, 2018; Ju et al., 2018; Wang et al., 2018).

One hypothesis we may have is that, without length restriction, a model will enjoy the benefit of recalling more long mentions, but also will be exposed to more false positives. This poses a challenge for a model – whether it is capable of balancing these two factors. Empirically, we find that the length restriction (c=6) improves the precision of semi-CRF and SH at the expense of the recall, providing some evidence to support the hypothesis. However, in terms of F_1, the unrestricted semi-CRF performs worse while unrestricted SH performs better compared to their restricted counterparts. The reason is that the span-level handcrafted features that the semi-CRF relies on can be very sparse when mentions are overly long. We empirically found this issue is alleviated in the model SH (-NN), possibly due to its ability in capturing interactions between neighboring spans. Even with length restriction, SH still yields competitive results, making it attractive in processing large-scale datasets considering its linear time complexity. Furthermore, we find that as c increases, SH performs better consistently in terms of F_1. The choice of c then becomes a tradeoff between time complexity and performance. Please refer to the supplementary material for details.

Compared with the local approach FOFE, our global approach gives a much better performance, showing its effectiveness in capturing interactions between spans. Moreover, FOFE's performance suffers significantly in the absence of the length restriction. The reason is that it will generate much more negative training instances under this setting, which makes its learning more challenging.

	Overlapping			Non-Overlapping			*w/s*
	P	R	F_1	P	R	F_1	
Lu and Roth (2015)	68.1	52.6	59.4	64.1	65.1	64.6	503
Muis and Lu (2017)	70.4	55.0	61.8	67.2	63.4	65.2	253
Wang et al. (2018)	77.4	70.5	73.8	76.1	69.6	72.7	1445
SH (c=6)	80.2	68.3	73.8	74.8	70.0	72.3	248
SH (c=n)	80.6	73.6	76.9	75.5	71.5	73.4	157

Table 5: Results on different types of sentences (ACE05), *w/s*: # of words decoded per second.

6.5 Additional Analyses

To understand our model better, we conduct some further experiments in this section.

Ablation study

We first conduct an ablation study by removing dropout, softmax-margin and pre-trained embeddings from our model respectively. The results are shown in Table 4. The dropout and pre-trained embeddings can improve the performance of our model significantly and this behavior is consistent with previous neural models for NER (Chiu and Nichols, 2016; Lample et al., 2016). Meanwhile, our new cost function based on softmax margin training also contributes significantly to the good performance of our model across these datasets.

How well does it handle overlapping mentions?

To further understand how well our model can handle overlapping mentions, we split the test data into two portions: sentences with and without overlapping mentions. We compare our model with the two state-of-the-art models and report results on ACE-05 in Table 5.[10] In both portions, SH achieves significant improvements, especially in the portion with overlapping mentions. This observation indicates that our model can better capture the structure of overlapping mentions than these two previous models. It also helps explain why the margin of improvement is larger in ACE than in GENIA since the former has more overlapping mentions than the latter, as shown in Table 2. Compared with the model with length restriction c, the unrestricted model mainly benefits from its ability to recall more overlapping mentions.

[10] Full results are listed in the supplementary material.

Model	F_1
SH (c=6)	89.6
SH (c=6) + *char*	90.5
SH (c=n)	89.2
SH (c=n) + *char*	90.2
Collobert et al. (2011)	88.7
Chiu and Nichols (2016)	90.9
Lample et al. (2016)	90.9
Ma and Hovy (2016)	91.2
Xu et al. (2017)	90.7
Strubell et al. (2017)	90.5

Table 6: Additional results on CoNLL-2003.

Running time

Since other compared models also feature linear time complexity (see Table 1), we examine the decoding speed in terms of the number of words processed per second. We re-implement the models of Lu and Roth (2015) and Muis and Lu (2017) using the same platform as ours (PyTorch) and run them on the same machine (CPU: Intel i5 2.7 GHz). The model of (Wang et al., 2018) is also tested with the same environment. Results on ACE-05 are listed in Table 5. The length bound (c=6) makes our model much faster, resulting in a speed comparable to the model of Muis and Lu (2017). The transition-based model by (Wang et al., 2018) has the best scalability partially because of its greedy strategy for decoding.

What if the data has no overlapping mentions?

To assess the robustness of our model and understand whether it could serve as a general mention extraction model, we additionally evaluate our model on CoNLL 2003 dataset which is annotated with non-overlapping mentions only. We compared our model with recent state-of-the-art neural network based models. For a fair comparison, we used the Collobert et al. (2011) embeddings widely used by previous models, and ignored POS tag features even though they are available. Results are in Table 6. Only neural models without using external features are included. [11] By only relying on word (and character) embeddings, our model achieves competitive results compared with other state-of-the-art neural models that also do not exploit external features, yet these models are mostly designed to handle only non-overlapping mentions. The only exception is the FOFE approach by (Xu et al., 2017) as we discussed earlier.

Notes on mention interactions

The dependencies between overlapping mentions can be very beneficial. SH can capture a specific kind of interaction between neighboring spans. Such interactions happen between mentions that share the same type and the same left boundary. As we can see from the sentence in Figure 3, one mention could also serve as a pre-modifier for another mention and both could share the same type. As shown in Table 2, there are over 8% such mentions in ACE and over 4% in GENIA. Specifically, SH relies on the hyperedges between **I** nodes to capture such interactions explicitly. To verify the effectiveness of this connection, we zero the weights between **I** nodes. The ablated model only achieves around 70.0% in ACEs and 71.4% in GENIA, implying the impact of this dependency connection. On the other hand, it also reveals the potential direction of improving SH by explicitly modeling more dependencies between mentions, such as the dependencies between mentions with different types. LSTM that serves as feature representation may capture such interactions implicitly, but building the connections could still be an important aspect for improvement.

7 Conclusion and Future Work

In this work, we propose a novel neural segmental hypergraph model that is able to capture overlapping mentions. We show that our model has some theoretical advantages over previous state-of-the-art approaches for recognizing overlapping mentions. Through extensive experiments, we show that our model is general and robust in handling both overlapping and non-overlapping mentions. The model achieves the state-of-the-art results in three standard datasets for recognizing overlapping mentions. We anticipate this model could be leveraged in other similar sequence modeling tasks that involve predicting overlapping structures such as recognizing overlapping and discontinuous entities (Muis and Lu, 2016) which frequently exist in the biomedical domain.

Acknowledgements

We thank the anonymous reviewers for their valuable comments. This work was done after the first author visited Singapore University of Technology and Design. This work is supported by Singapore Ministry of Education Academic Research Fund (AcRF) Tier 2 Project MOE2017-T2-1-156.

[11] See the supplementary material for complete results.

References

Steven Abney, Michael Collins, and Amit Singhal. 2000. Answer extraction. In *Proc. of the sixth conference on applied natural language processing*.

Beatrice Alex, Barry Haddow, and Claire Grover. 2007. Recognising nested named entities in biomedical text. In *Proc. of BioNLP*.

James K Baker. 1979. Trainable grammars for speech recognition. *The Journal of the Acoustical Society of America*, 65(S1):S132–S132.

Andreas Brandstädt, Feodor Dragan, Victor Chepoi, and Vitaly Voloshin. 1998. Dually chordal graphs. *SIAM Journal on Discrete Mathematics*, 11(3):437–455.

Kai-Wei Chang, Rajhans Samdani, and Dan Roth. 2013. A constrained latent variable model for coreference resolution. In *Proc. of EMNLP*.

Jason PC Chiu and Eric Nichols. 2016. Named entity recognition with bidirectional lstm-cnns. *Transactions of the Association for Computational Linguistics*, 4:357–370.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12(Aug):2493–2537.

George R Doddington, Alexis Mitchell, Mark A Przybocki, Lance A Ramshaw, Stephanie Strassel, and Ralph M Weischedel. 2004. The automatic content extraction (ace) program-tasks, data, and evaluation. In *Proc. of LREC*.

Jenny Rose Finkel and Christopher D Manning. 2009. Nested named entity recognition. In *Proc. of EMNLP*.

R. Florian, H. Hassan, A. Ittycheriah, H. Jing, N. Kambhatla, X. Luo, N. Nicolov, and S. Roukos. 2004. A statistical model for multilingual entity detection and tracking. In *Proc. of HLT-NAACL*.

Giorgio Gallo, Giustino Longo, Stefano Pallottino, and Sang Nguyen. 1993. Directed hypergraphs and applications. *Discrete applied mathematics*, 42(2-3):177–201.

Kevin Gimpel and Noah A. Smith. 2010. Softmax-margin crfs: Training log-linear models with cost functions. In *Proc. of HLT-NAACL*.

Ralph Grishman. 1997. Information extraction: Techniques and challenges. In *International Summer School on Information Extraction*, pages 10–27. Springer.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional lstm-crf models for sequence tagging. *arXiv preprint arXiv:1508.01991*.

Meizhi Ju, Makoto Miwa, and Sophia Ananiadou. 2018. A neural layered model for nested named entity recognition. In *Proc. of NAACL-HLT*.

Arzoo Katiyar and Claire Cardie. 2018. Nested named entity recognition revisited. In *Proc. of NAACL-HLT*.

J-D Kim, Tomoko Ohta, Yuka Tateisi, and Jun'ichi Tsujii. 2003. Genia corpus—a semantically annotated corpus for bio-textmining. *Bioinformatics*, 19(suppl_1):i180–i182.

Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. In *Proc. of ICLR*.

Lingpeng Kong, Chris Dyer, and Noah A Smith. 2016. Segmental recurrent neural networks. In *Proc. of ICLR*.

Frank R Kschischang, Brendan J Frey, and H-A Loeliger. 2001. Factor graphs and the sum-product algorithm. *IEEE Transactions on information theory*, 47(2):498–519.

John Lafferty, Andrew McCallum, and Fernando CN Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. of ICML*.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *Proc. of NAACL-HLT*.

Qi Li, Heng Ji, and Liang Huang. 2013. Joint event extraction via structured prediction with global features. In *Proc. of ACL*.

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. In *Proc. of EMNLP*.

Wei Lu and Dan Roth. 2015. Joint mention extraction and classification with mention hypergraphs. In *Proc. of EMNLP*.

Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via bi-directional lstm-cnns-crf. In *Proc. of ACL*.

Ryan McDonald, Koby Crammer, and Fernando Pereira. 2005. Flexible text segmentation with structured multilabel classification. In *Proc. of HLT-EMNLP*.

Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proc. of ACL-IJCNLP*.

Behrang Mohit, Nathan Schneider, Rishav Bhowmick, Kemal Oflazer, and Noah A Smith. 2012. Recall-oriented learning of named entities in arabic wikipedia. In *Proc. of EACL*.

Aldrian Obaja Muis and Wei Lu. 2016. Learning to recognize discontiguous entities. In *Proc. of EMNLP*.

Aldrian Obaja Muis and Wei Lu. 2017. Labeling gaps between words: Recognizing overlapping mentions with mention separators. In *Proc. of EMNLP*.

Vincent Ng and Claire Cardie. 2002. Improving machine learning approaches to coreference resolution. In *Proc. of ACL*.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proc. of EMNLP*.

Lev Ratinov and Dan Roth. 2009. Design challenges and misconceptions in named entity recognition. In *Proc. of CoNLL*.

S. Riedel and A. McCallum. 2011. Fast and robust joint models for biomedical event extraction. In *Proc. of EMNLP*.

Sunita Sarawagi and William W Cohen. 2005. Semi-markov conditional random fields for information extraction. In *Proc. of NIPS*.

Wee Meng Soon, Hwee Tou Ng, and Daniel Chung Yong Lim. 2001. A machine learning approach to coreference resolution of noun phrases. *Computational linguistics*, 27(4):521–544.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *JMLR*.

Emma Strubell, Patrick Verga, David Belanger, and Andrew McCallum. 2017. Fast and accurate entity recognition with iterated dilated convolutions. In *Proc. of EMNLP*.

Yuka Tateisi. 2004. Part-of-speech annotation of biology research abstracts. In *Proc. of LREC*.

Erik F Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *Proc. of CoNLL*.

Bailin Wang, Wei Lu, Yu Wang, and Hongxia Jin. 2018. A neural transition-based model for nested mention recognition. In *Proc. of EMNLP*.

Mingbin Xu, Hui Jiang, and Sedtawut Watcharawittayakul. 2017. A local detection approach for named entity recognition and mention detection. In *Proc. of ACL*.

Jie Zhang, Dan Shen, Guodong Zhou, Jian Su, and Chew-Lim Tan. 2004. Enhancing hmm-based biomedical named entity recognition by studying special phenomena. *Journal of Biomedical Informatics*, 37(6):411–422.

Guodong Zhou. 2006. Recognizing names in biomedical texts using mutual information independence model and svm plus sigmoid. *International Journal of Medical Informatics*, 75(6):456–467.

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.