A Neural-Encoded Mention-Hypergraph Model for Mention Recognition

Jimmy Ming-Tai Wu[1], Jerry Chun-Wei Lin[2,*], Yinan Shao[3], and Matin Pirouz[4]
[1]*College of Computer Science and Engineering*
Shandong University of Science and Technology, Qingdao, China
[2]*Department of Computing, Mathematics, and Physics*
Western Norway University of Applied Sciences, Bergen, Norway
[3]*Department of Computer Science and Technology*
Harbin Institute of Technolgy (Shenzhen), Shenzhen, China
[4]*Department of Computer Science*
California State University, Fresno, USA
wmt@wmt35.idv.tw; jerrylin@ieee.org; shaoyn0817@163.com; mpirouz@ieee.org

***Abstract*—In this paper, we propose a neural-encoded mention-hypergraph model to extract the mention entities from a content. A hypergraph model combines with an encoding schema and a neural network in the proposed model. The proposed model can effectively capture the overlapping mention entities of an unbounded length. Experimental findings demonstrated that on most standard datasets, the proposed model achieved better efficiency than existing related models.**

***Index Terms*—semi-CRF, CNN, attention mechanism, sequence prediction, neural network.**

I. Introduction

Sequence prediction includes many sub-tasks, such as word segmentation,named entity recognition, and part-of-speech recognition, among others. This work focuses mainly on the mention-extraction task, which identifies each unit/subsequence of an entry sequence, identifying and assigning the mention-entity label. A reference to the natural language text that can be named, nominal, or pronominal [7] is normally defined as a mention. The task to mention is similar to other traditional tasks such as entity recognition and shallow parsing. The semantic tags can be properly assigned to text spans of an input sequence, and arbitrary length can be used for text span. The mention extraction is specific because mention can refer to an overlap or nest structure representation.

In recent years, mention extraction has become an increasingly important part of several downstream tasks, for example the relation extraction in [11], [20] and the co-reference resolution in [13]. In the conventional sequence label models, such as the Conditional Random Fields (CRFs) and the Models of Maximum Entropy (MEM), the input unit (i.e., characters or words) models the conditional probability over an input sequence. Segments such as semi-Markov Random Fields (semi-MRFs) can be used directly from the input sequence to represent any text span. A tree-based discriminatory constituency parser was introduced by Finkel and Manning [8]. Lu and Roth [12] have used a hypergraph model which achieves a complex liner time and can easily manage and maintain the nested structure present in the mention extraction. The notion of mention separators and the multigraph representation were used by Muis and Lu [21].

In this paper, we propose a model of a neural network model based on mention entities which extract nesting and overlapping structures. The proposed model can extract these entities automatically from the natural language texts with high performance. It can therefore be used for numerous downstream processing tasks, including information collection and the classification of sentences. The proposed model combines the BILU encoding scheme, the mention-hypergraph model and the neural network to carry out the mention extraction task. The neural network model is used in the encoded hypergraph model to calculate feature scores for certain edges/hyperedges. The proposed model in this paper is capable of automatically recognizing the nestled structure of the texts in the natural language. It combines an encoding schema with a hypergraph-based model, capturing more boundary characteristics than the models reported in the last work, and this feature proves to be effective at the mention extraction task. The experiments show that, compared to the previously reported models, the proposed model is able to obtain competitive results on standard datasets.

II. Related Work

The Hidden Markov models (HMM) [3]–[5], Max-Entropy Model (MEM) [2], Conditional Random Fields (CRFs) [18], and semi-Markov Random Fields (semi-CRFs) [28] form the traditional sequence prediction models. These are linear models which can record correlations between the labels and model the distribution of the entire label sequence in the neighborhoods. A HMM, represented as a dynamic Bayesian network, was introduced by Baum et al. [3]–[5]. When applying the HMM to a sequence-prediction task, the status of the models is invisible while the outputs depending on the conditions are visible. The hierarchical model Markov hidden (HHMM) was developed by Fine et al. [10] that marks a recursive hierarchical generalization of the generic

*Corresponding author

HMM. Zhang et al. [33] have established the Chinese LeXical Analysis System (ICTCLAS), which utilizes the HHMM to include in an extensive theoretical framework, Chinese word segmentation, partial language markings, disambiguations, and unknown word recognition.

Shen et al. [27] used an HMM for identifying the named entity recognition in the field of biomedicine. In natural language processing, Berger et al. [2] have utilized a Maximum Entropy Model (MEM). The Master Entropy Markov model (MEMM) was proposed by McCallum et al. [19]. The MEMM is a discriminatory sequence prediction graph model. In the MEMM models, the observations were shown as numerous overlapping. Yu et al. used continuous features in MEM [30]. They explained why the application of the moment limit in the MEM only worked well with binary characteristics instead of with continuous characteristics. In the optimization problem of their model, a log-linear model optimization has been implemented in a large space. The POS tags were assigned with very high precision with a statistical model proposed by Ratnaparkhi [24] to unseen texts. A Maximum Entropy Entropy Model (MOP-MEMM) was proposed by Rosenberg et al. [25].

In 2001, Lafferty et al. [18] suggested the CRF models. The CRF model relieves strong assumptions of independence used in HMMs and also prevents the problem in the MEM models that prefers states with a number of countries that succeed. The Chinese Word Segmentation (CWS) system based on the CRF model was presented by Tseng et al. [29]. Zhao et al. [31] considered the CWS problem to be a character-based tagging issue. Zhao et al. [32] then used a six-tag set, combined with the Chinese character and the trained supported segments of another corpora to further improve the performance of the CRF-based Chinese language segmentation. The new feature-rich discriminatory CRF parser was first presented by Finkel et al. [9]. Their model has proven its effectiveness in full Wall Street Journal (WSJ) data.

Sarawagi and Cohen [28] suggested the Semi-Markov Conditional Random Fields (Semi-CRF). The proposed Semi-CRF produces an input sequence segment of x, in which labels are given instead of to individual words for each segment. A model that can include both CRF and Semi CRF functionality was proposed by Andrew [1]. The high-order semi-CRF features proposed by Nguyen et al. [23] are included in the first-order semi-CRF model. The weak CRFs of semi Markov, for the chunking noun, were proposed by Muis and Wei [22]. In conventional semi-CRF, the model decides intuitively on the next segment length and type simultaneously, while in the weak semi-CRF, the model tries to propose a more weak variant, separating the two decisions by restricting each node, either in the next or the next segments to the same label nodes.

In sequence prediction [6], [15], the deep learning methods show advantages. Huang et al. [15] have proposed a large number of sequence prediction deep learning models, including the LSTM, the Bi-LSTM and the LSTM-CRF model. A model was suggested by Dyer et al. [6] to represent the state of a transition-based dependency parser. Two different models were presented by Lample et al. [17], the LSTM neural network with the CRF layer, and the shift-reduce parser-based approach to build and label segments. A segmental recurrent neural network was proposed by Kong et al. [16] to denote a variant of the semi-Markov CRF. The gated recursive semi-CRFs (GR-CRF), was proposed by Zhuo et al. [34]. This model models segments directly in the input sequence and uses a gated recursive convolutional neural network to automatically extract each segment's representation. Rei et al. [26] incorporated the Out-Of Vocabulary (OOV) problem in prediction of the sequence of information on the character level.

III. Preliminaries and Problem Statement

A. Recurrent Neural Networks (RNNs)

Recurrent Networks (RNNs) indicate a type of neural network commonly used with sequence data. In practice, however, generic RNNs are difficult to train, and a long-term dependence is not determined. The neural networks of LSTM [14] refer to a RNNs variant for the problem. An LSTM unit consists of three multiplicative gates for controlling the flow of information. The structure of the LSTM cell is shown in Fig. 1.

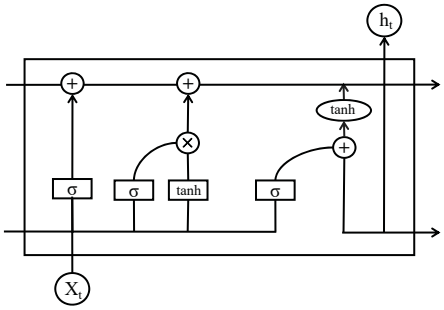

Fig. 1: The long short-term memory cell.

An LSTM unit is updated *t* by:

$$i_t = \sigma(W_i h_{t-1} + U_i x_t + b_i), \tag{1}$$

$$f_t = \sigma(W_g h_{t-1} + U_f x_f + b_g), \tag{2}$$

$$c_t^{'} = tanh(W_c h_{t-1} + U_c x_t + b_c), \tag{3}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot c_t^{'}, \tag{4}$$

$$o_t^{'} = \sigma(W_o h_{t-1} + U_o x_t + b_o), \tag{5}$$

$$h_t = o_t \odot tanh(c_t), \tag{6}$$

where x_t shows the input of an LSTM unit at time step *t*, σ indicates the sigmoid function, and $\odot$ refers to the product operation; further, h_t shows the hidden state computed by an LSTM unit at time step *t*, U_i, U_f, U_c, and U_o are the corresponding weight matrices of different gates, W_i, W_f, W_c, and W_o are the corresponding weight matrices of a hidden state h_t; and b_i, b_f, b_c, and b_o denote the bias vectors; f_t denotes the forget gate which controls how much information

to forget, and c_t denotes the cell state which controls how much information to update. The hidden h_t indicates the final result that can be regarded as an input word vector.

In this work, the bidirectional LSTM (Bi-LSTM) [4] is used, which can both access (left) and future (right) contexts when extracting a hidden condition at time step *t*. The Bi-LSTM appears in Fig. 2. The round nodes on the ground bottom mark the input vectors, the dark-square nodes on the top mark the output vectors and the rectangular nodes on the middle mark the LSTM units already displayed in the Fig. 1. Bi-LSTM basically inputs two separate LSTM neural networks, forward and backwards each sequence and at every stage combines the output of these two LSTM neural networks to form the final output.

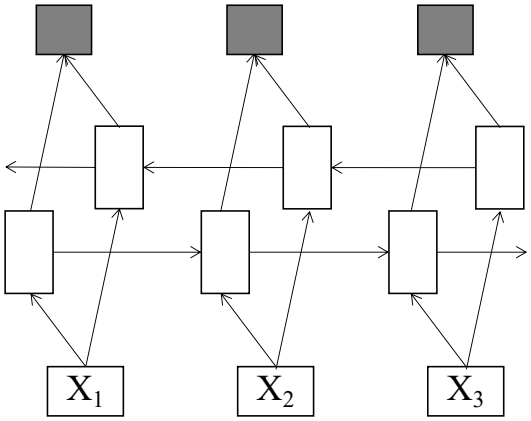

Fig. 2: A Bi-LSTM network structure.

B. Hypergraph

Each hyperedge is a parent node and a listed children nodes in the proposed model. In this work, a hypergraph models the conditional probability of a possible output sequence *s* over an input sequence *x* by:

$$p(s|x) = \frac{1}{Z(x)} exp\{W \cdot G(x,s)\}, \tag{7}$$

where $G(x,s)$ denotes the feature function, W is the weight vector which can be adjusted during the training phase, and $Z(x)$ is the normalization factor of all the possible segmentations i.e., *s* over *x*. Here, a dynamic programming technique is used to efficiently measure $Z(x)$. To discover the best label sequence in a hypergraph, we can suggest that α_j shows the best label sequence ends with the *j*-th input, (m,n,y) denotes a label sequence starting at the *m*-th position, which at the *n*-th position has a label *y*. Then, α_j can be thus recursively calculated by:

$$\alpha_j = max_y \psi(j-1,j,y) + \alpha_{j-1}, \tag{8}$$

where $\psi(j-1,j,y)$ is the feature value defined over the edge $(j-1,\ j,\ y)$.

C. Neural Hypergraph-Based Model

The past inputs of LSTM layer and the other user-specific sparse features of hypergraph layers can be used efficiently by a neural hypergraph-oriented network. Here is an example of a transition to a tag feature called $[A]$, in which each of $[A]_{i,j}$ is a model of a *i*-th tag transition score from a *i*-th to a *j*-th tag. The transition matrix must be noted as being regardless of the location.

The neural network outputs the matrix with scores $f_\theta([x]_i^T)$, which can be defined as the neural features. Element $[f_\theta]_{i,t}$ of the matrix denotes the score of the *i*-th tag at the *t*-th word in a sentence $[x]_i^T$, measured by the neural networks with parameter θ. The score of a sentence $[x]_i^T$, which can be labeled with the label path $[i]_i^T$ that is measured by the sum of transition and network scores as:

$$s([x]_i^T, [i]_i^T, \theta) = \Sigma_{t=1}^T([A]_{[i]_{t-1},[i]_t} + [f_\theta]_{[i]_t,t}), \tag{9}$$

where $[A]$ denotes the tag transition matrix, and $[f_\theta]$ denotes the hypergraph features.

Thus, the problem statement can be defined as follows. Given an input sequence $x = (x_1, \ldots, x_k)$ of length *k*, $x_{a:b}$ denotes its subsequence $(x_a, \ldots, x_b)$, which holds $a \leq b \leq k$. A mention entity is defined as (u,v,y), which indicates that the sub-sequence $x_{u:v}$ is relevant to the mention entity label *y*. With an input sequence *x*, the mention-extraction problem is defined to extract all the mention entities from an input sequence *x*, in which the mentions can be defined as overlapped or nested structure.

IV. Proposed Mention Hypergraph Model

The mention hypergraph model [12] is used to encode mentions of different types and lengths using nodes and directed hyperedges. A hypergraph of partial mention is shown in Fig. 3. This hypergraph contains all the possible label paths of the input sentences. The following five types of nodes are used in this model:

- A_k indicates a mention which starts at *k* or later.
- E_k indicates a mention whose left boundary is at position *k*.
- T_k^j indicates a mention of *j* type whose left boundary is at position *k*.
- I_k^j indicates a mention of *j* type which contains a position *k*.
- *X* indicates the mention end.

We adjust the parameters in the training phase to maximize the log likelihood of the graph presented in Fig. 4.

A. Encoded-Mention Hypergraph Model

The proposed model is the encoded mention hypergraph mode, which has more nodes and edge links. A part of the hypergraph model encoded with the designed model is shown in Fig. 5. In the proposed model, the following eight node types are used:

- A_k indicates a mention which starts at position *k* or later.
- E_k indicates a mention whose left boundary is at position *k*.

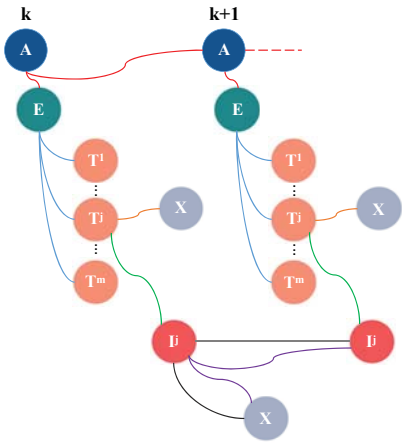

Fig. 3: The (partial) mention hypergraph model.

- T_k^j indicates a mention of type j whose left boundary is at position k.
- B_k^j indicates a mention of type j starting at position k.
- I_k^j indicates a mention of type j covering position k.
- L_k^j indicates a mention of type j ending at position k.
- U_k^j indicates a mention of type j of a unit length at position k.
- X indicates the mention end.

The parameters are adjusted during the training phase so that the log likelihood of the graph in the Fig. 6 is maximized.

B. Neural-Encoded Mention Hypergraph Model

For a particular edge/hyperedge at k, word k^{th} embedding indicates the input of the neural network, which through the linear/nonlinear transformation, outputs the scores for all the mentioned types. We use the following function to combine the neural network with the encoded mention hypergraph model as:

$$p(s|x) = \frac{1}{Z(x)} exp\{w_1 G(x,s) + w_2 N(x,s)\}, \qquad (10)$$

where $G(x,s)$ indicates the hypergraph feature score, w_1 is the related weight of the encoded mention hypergraph, w_2 is corresponding to the neural network features, $N(x,s)$ represents the neural feature score measured by the Bi-LSTM neural network, and $Z(x)$ indicates the normalization factor of all the possible label sequences over x. In the neural encoded mention hypergraph training, we use the maximum conditional likelihood estimation. The log-like feature is for training set$\{(x_i, s_i)\}$ as:

$$L_D(W) = \Sigma_{i \epsilon D} log\ p(s|x). \qquad (11)$$

The W parameter is chosen to maximize the log-likelihood $L_D(W)$. The algorithm for training is given in the Algorithm 1. In similarity with the classic CRF model training algorithm, the proposed neural hypergrapher model initially transfers the neural network (i.e., the neural Bi-LSTM) and then combines the calculated feature scores with the hypergraph spare features. These features are used to update the hypergraph (i.e., sparse features and neural network features) in the forward and backward directions. Finally the updated features of the neural network are used to update the neural network parameters using the backward algorithm.

C. Features

This section briefly presents all the hypergraph features used to measure $G(x,s)$ by 10, which was inspired by Finkel et al. [9]. In particular, the following features defined by the input are considered:

- **Word features:** Words with a window size of 3 appear around the current position.
- **POS tag features (if available):** POS tags with window size 3 appearing around the current position.
- **Word *n*-grams features**: Word n-grams with a window size of n = 2, 3, 4 (contains current position).
- **POS *n*-gram features (if available):** POS tags with a window size of n = 2, 3, 4 (contains current position).
- **Bag of words features**: Word bags with a size of 5 windows.
- **Word pattern features:** The word pattern features include: All-Capital, All-Digits, All-Alphanumeric, Contain-Digits, Contains-Hyphen, Initial-Capital, Punctuation, Roman-Number, and URLs.

V. Experimental Evaluation

The empirical assessment of our proposed model is presented in this section. The testing of the **ACE2004** and **ACE2005** datasets was carried out after previous works [12], [21]. In Table I, the parameters of the dataset are summarized, where the number in the brackets is the corresponding mention entity number. Since the performance of the semi-CRF model is mainly related to the hyperparameter max span length n, n has been set at 6 and ∞ in experiments. In addition, it should be noted that F in brackets shows the mention penalty feature introduced to optimize the value of F.

A. ACE2004 Dataset

In the experiments, 80% of data was used for model training, 10% for developmental set (dev set) and the remainder 10% for evaluation set (test set). The best performances are marked with the bold and underlined font from Tables II to III.

The proposed model produced significantly better results as the other templates in Table II, regardless of the F1-score was optimized or not. A better performance than the other model tested is achieved with the designed model. In addition, when optimization F was adopted, the designed model yielded the best performance. In particular, our approach was much more accurate and retracted when the BILU encoding scheme has been used, leading to an improved F1 score. These results largely demonstrated the efficacy of the proposed hypergraph model of neural encoding. Furthermore, the semi-CRF baseline produced, as expected, relatively lower results than the other models, as it could not predict the mentions overlapping.

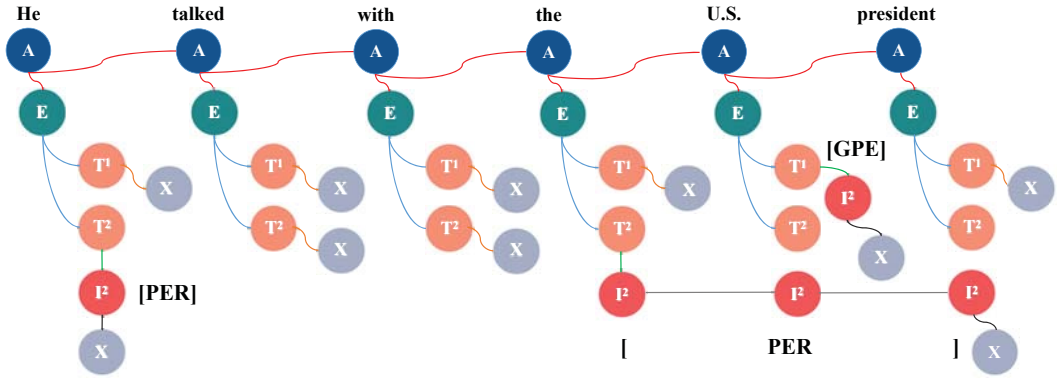

Fig. 4: The mention hypergraph model.

Algorithm 1: The training procedure of the developed neural-encoded mention hypergraph model

```
for each epoch do
    for each batch do
        neural network forward pass for neural network state;
        encoded mention hypergraph forward and backward pass;
        neural network backward pass: backward pass for neural network ;
        update the parameters.
```

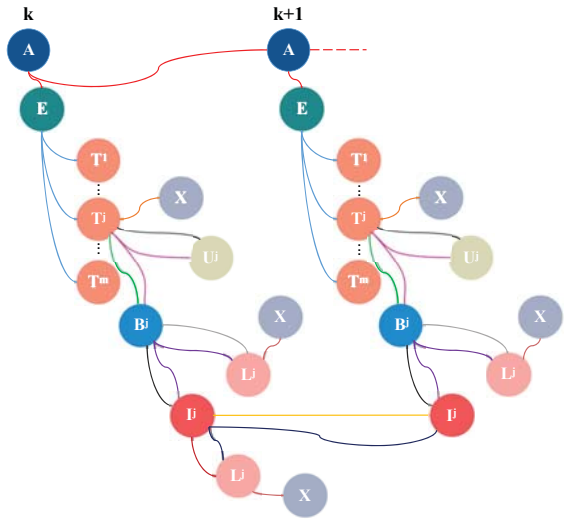

Fig. 5: The (partial) encoded mention hypergraph model.

TABLE I: Characteristics of the used datasets.

	#Train-sent	#Dev-sent	#Test-sent
ACE2004	6,799 (22,207)	829 (2,511)	879 (3,031)
ACE2005	7,336 (24,687)	958 (3,217)	1,047 (3,027)

B. ACE2005 Dataset

Similarly, the experiments were carried out using the ACE2005 dataset. All the documents from *bc*, *bn*, *nw*, and *wl* are then considered. As before, 80% of the data were used for training, the remaining 10% was used for model evaluation and 10% for development. The best performance on developing and testing systems is marked with the bold, highlighted font in table III.

TABLE II: Compared results on ACE2004.

	Test Set		
	Precision	Recall	F-Value
CRF (BIO)	70.0	40.3	51.2
CRF (BILOU)	71.8	40.8	52.1
semi-CRF (n=6)	76.1	41.4	53.6
semi-CRF (n=∞)	66.7	42.0	51.5
Proposed model	82.28	53.78	65.04
Proposed model (F)	76.41	58.66	**66.37**

TABLE III: Compared results on ACE2005.

	Test Set		
	Precision	Recall	F-Value
CRF (BIO)	67.6	43.7	53.1
CRF (BILOU)	69.5	44.5	54.2
semi-CRF (n=6)	72.8	45.0	55.6
semi-CRF (n=∞)	67.5	46.1	54.8
Proposed model	80.75	54.87	65.34
Proposed model (F)	73.96	59.76	**66.11**

The proposed model was significantly better than the other models, as shown in Table III. The F values of the designed model were better than that of the other models. Furthermore, our model can handle about higher F-value for nested structures compared to the baseline semi-CRF model.

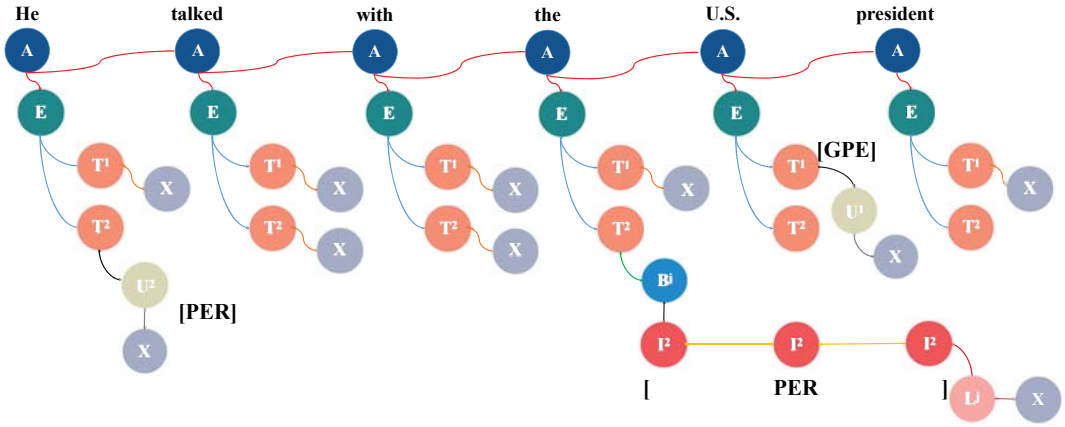

Fig. 6: The designed mention hypergraph model.

VI. Conclusion

A Neural-Encoded Mention Hypergraph (NEMH) is suggested in this paper for a mention recognition task, using the BLIU encoding schema. The empirical findings showed that most datasets were able to deliver higher performance than the traditional models on most datasets.

References

[1] G. Andrew, "A hybrid Markov/semi-Markov conditional random field for sequence segmentation," *The Conference on Empirical Methods in Natural Language Processing*, pp. 465–472, 2006

[2] A. L. Berger, S. A. D. Pietra and V. J. D. Pietra, A maximum entropy approach to natural language processing," *Computational Linguistics*, pp. 39–71, 1996

[3] L. E. Baum and T. Petrie, "Statistical inference for probabilistic functions of finite state Markov chains," *The Annals of Mathematical Statistics*, 37(6), pp. 1554–1563, 1966

[4] L. E. Baum, T. Petrie, G. Soules and N. Weiss, "A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains," *The Annals of Mathematical Statistics*, 41(1), pp. 164–171, 1970

[5] L. E. Baum, "An inequality and associated maximization technique in statistical estimation of probabilistic functions of a Markov process," *Inequalities*, 3, pp. 1–8, 1972

[6] C. Dyer, M. Ballesteros, W. Ling, A. Matthews, and N. A. Smith, "Transition based dependency parsing with stack long short term memory," *The Association for Computational Linguistics*, pp. 334–343, 2015

[7] R. Florian, H. Hassan, A. Ittycheriah, H. Jing, N. Kambhatla, X. Luo, H Nicolov, and S. Roukos, "A statistical model for multilingual entity detection and tracking," *The Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 1–8, 2004

[8] J. R. Finkel and C. D. Manning, "Nested named entity recognition," *The Conference on Empirical Methods in Natural Language Processing*, pp. 141–150, 2009

[9] J. R. Finkel, A. Kleeman and C. D. Manning, "Efficient, feature-based, conditional random field parsing," *The Association for Computational Linguistics*, pp. 959–967, 2008

[10] S. Fine, Y. Singer and N. Tishby, "The hierarchical hidden Markov model: analysis and applications," *Kluwer Academic Publishers*, 1998

[11] P. Gupta and B. Andrassy, "Table filling multi-task recurrent neural network for joint entity and relation extraction," *The International Conference on Computational Linguistics*, pp. 2537–2547, 2016

[12] W. Lu and D. Roth, "Joint mention extraction and classification with mention hypergraphs," *The Conference on Empirical Methods in Natural Language Processing*, pp. 857–867, 2015

[13] S. Guo, M. W. Chang and E. Kiciman, "To link or not to link? a study on end-to-end Tweet entity linking," *The Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 1020–1030, 2013

[14] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, 9(8), pp. 1735–1780, 1997

[15] Z. Huang, W. Xu and K. Yu, "Bidirectional LSTM-CRF models for sequence tagging," https://arxiv.org/pdf/1508.01991.pdf, 2015

[16] L. Kong, C. Dyer C and N. A. Smith, Segmental recurrent neural networks, https://arxiv.org/abs/1511.06018, 2016

[17] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami and C. Dyer, "Neural architectures for named entity recognition," *The Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 260–270, 2016

[18] J. D. Lafferty, A. Mccallum and F. C. N. Pereira, "Conditional random fields: probabilistic models for segmenting and labeling sequence data," *The International Conference on Machine Learning*, pp. 282–289, 2001

[19] A. McCallum, D. Freitag and F. C. N. Pereira, "Maximum entropy Markov models for information extraction and segmentation," *The International Conference on Machine Learning*, pp. 591–598, 1999

[20] M. Mintz, S. Bills, R. Snow and D. Jurafsky, "Distant supervision for relation extraction without labeled data," *The Annual Meeting of the Association for Computational Linguistics-International Joint Conference of the Asian Federation of Natural Language Processing*, pp. 1003–1011, 2009

[21] A. O. Muis and W. Lu, "Labeling gaps between words: recognizing overlapping mentions with mention separators," *The Conference on Empirical Methods in Natural Language Processing*, pp. 2598–2608, 2017

[22] A. O. Muis and W. Lu, "Weak semi-Markov CRFs for noun phrase chunking in informal text," *The Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 714–719, 2016

[23] V. C. Nguyen, N. Ye, W. S. Lee and L. C. Hai, "Semi-Markov conditional random field with high-order features," *Journal of Machine Learning Research*, 15(1), pp. 981–1009, 2014

[24] A. Ratnaparkhi, "A maximum entropy model for part-of-speech tagging," *The Conference on Empirical Methods on Natural Language Processing*, pp. 133–142, 2011

[25] D. S. Rosenberg, K. Dan, B. Taskar, "Mixture-of-parents maximum entropy Markov models," *The Conference on Uncertainty in Artificial Intelligence*, pp. 318–325, 2007

[26] M. Rei, G. K. O. Crichton and S. Pyysalo, "Attending to characters in neural sequence labeling models," https://arxiv.org/abs/1611.04361, 2016

[27] D. Shen, J. Zhang, G. Zhou, J. Su and C. L. Tan, "Effective adaptation of a hidden Markov model-based named entity recognizer for biomedical domain," *The ACL Workshop on Natural Language Processing in Biomedicine*, pp. 49–56, 2003

[28] S. Sarawagi and W. W. Cohen, "Semi-Markov conditional random fields for information extraction," *The Neural Information Processing Systems*, pp. 1185–1192, 2004

[29] H. Tseng, P. Chang, G. Andrew, D. Jurafsky and C. Manning, "A conditional random field word segmenter for sighan bakeoff 2005," pp. 168–171, 2015

[30] D. Yu, L. Deng and A. Acero, "Using continuous features in the

maximum entropy model," *Pattern Recognition Letters*, 30(14), pp. 1295–1300, 2009

[31] H. Zhao, C. N. Huang, M. Li and T. Kudo, "An improved Chinese word segmentation system with conditional random field," *The SIGHAN Workshop on Chinese Language Processing*, pp. 162–165, 2006

[32] H. Zhao, C. N. Huang, M. Li and B. L. Lu, Effective tag set selection in Chinese word segmentation via conditional random field modeling, *The Pacific Asia Conference on Language, Information and Computation*, pp. 87–94, 2006

[33] H. P. Zhang, Q. Liu, X. Q. Cheng, H. Zhang and H. K. Yu, "Chinese lexical analysis using hierarchical hidden Markov model," *Sighan Workshop on Chinese Language Processing*, 17(8), pp.63–70, 2003

[34] J. Zhuo, Y. Cao, J. Zhu, B. Zhang and Z. Nie, "Segment-level sequence modeling using gated recursive semi-Markov conditional random fields," *The Association for Computational Linguistics*, pp. 1413–1423, 2016