Information Sciences 496 (2019) 53–64

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

BILU-NEMH: A BILU neural-encoded mention hypergraph for mention extraction

Jerry Chun-Wei Lin[a,*], Yinan Shao[b], Philippe Fournier-Viger[c], Fujita Hamido[d]

[a] Department of Computing, Mathematics, and Physics, Western Norway University of Applied Sciences, Bergen, Norway
[b] School of Computer Science and Technology, Harbin Institute of Technology (Shenzhen), Shenzhen, China
[c] School of Natural Sciences and Humanities, Harbin Institute of Technology (Shenzhen), Shenzhen, China
[d] Faculty of Software and Information Science, Iwate Prefectural University, Iwate, Japan

ARTICLE INFO

Article history:
Received 29 July 2018
Revised 15 March 2019
Accepted 27 April 2019
Available online 3 May 2019

Keywords:
Semi-CRF
CNN
Attention mechanism
Sequence prediction
Neural network

ABSTRACT

The natural language processing (NLP) denotes a technique used to process data such as text and speech. Some of the fundamental research in NLP includes the named entity recognition, which recognizes the named entities (i.e., persons and companies) from texts, the semantic parsing, which converts a natural language utterance to a logical form, and the co-reference resolution, which extracts the nouns (including pronouns and noun phrases) pointing to the same reference body. In this paper, we focus on the mention extraction and classification, proposing a neural-encoded mention-hypergraph model named the BILU-NEMH to extract the mention entities from a content. The proposed BILU-NEMH model combines a mention hypergraph model with the encoding schema and neural network. The proposed model can effectively capture the overlapping mention entities of an unbounded length. The proposed model was verified by the experiments, and the obtained experimental results showed that the proposed model achieved better performance and greater effectiveness than the existing related models on most standard datasets.

1. Introduction

Sequence prediction represents a natural language processing task which assigns a category label to each part of an input sequence. Sequence prediction includes many sub-tasks, among which are word segmentation, named entity recognition, part-of-speech recognition, mention-extraction, and others. In this work, we mainly focus on the mention-extraction task which denotes the task of identifying and assigning the mention-entity label to each unit/sub-sequence of an input sequence. A mention is typically defined as a reference to an entity in a natural language text that can be named, nominal, or pronominal [7]. The mention task is like other traditional sequence-labeling tasks such as named entity-recognition and shallow parsing. Thesemantic tags can be appropriately assigned to text spans of an input sequence, and the text spans can be set to an arbitrary length. The mention extraction is specific because mentions can denote a representation of an overlapping or nest structure. The examples of such structures are shown in Fig. 1. In Fig. 1, on the left side, the PER (PERson) mention entity string $X_1X_2X_3$ is overlapped with string $X_2X_3X_4$, and on the right side, the PER mention entity X_1X_2 string is nested within $X_1X_2X_3X_4$ string.

* Corresponding author.
E-mail addresses: jerrylin@ieee.org (J.C.-W. Lin), shaoyn0817@gmail.com (Y. Shao), philfv@hitsz.edu.cn (P. Fournier-Viger), HFujita-799@acm.org (F. Hamido).
https://doi.org/10.1016/j.ins.2019.04.059

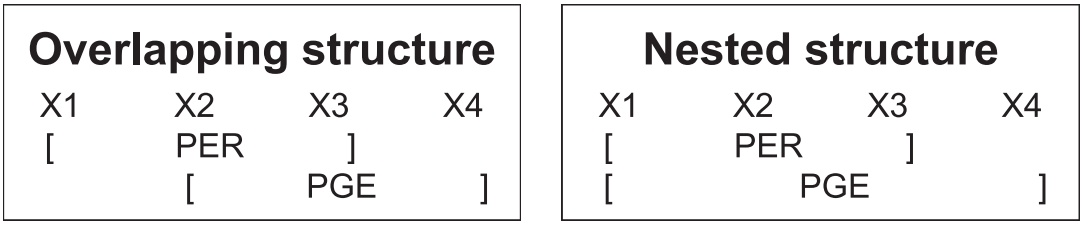

Fig. 1. The examples of the overlapping and nested structures.

In recent years, mention extraction has gained increasing attention because it has been playing a significant role in several downstream tasks, such as relation extraction [11,21] and co-reference resolution [13]. The traditional sequence-labeling models, such as the Conditional Random Fields (CRFs) and the Maximum Entropy Model (MEM), model the conditional probability over an input sequence by representing the input unit (i.e., characters or words). The segmentation models, such as the semi-Markov Random Fields (semi-MRFs), can be used to represent every text span (i.e., subsequence) directly from the input sequence. Although the above models can be easily utilized in the mention extraction process, they cannot adequately address the overlapping mention entities. Finkel and Manning [8] presented a tree-based discriminative constituency parser, which can be used to recognize the nested named entities. However, this algorithm suffers from a limitation because a tree structure requires higher time complexity. Lu and Roth [12] used a hypergraph-based model, which achieved a liner time complexity, and the nested structure occurring in the mention-extraction could be easily handled and maintained. To extract the mention entities, Muis and Lu [22] applied the notion of mention separators together with the multigraph representation. Their model can efficiently extract the mention entities of both nested and overlapping structures.

In this paper, we propose a neural network based model for mention entities extraction of both nested and overlapping structures. The proposed model can automatically extract the mention entities from the natural language texts achieving high performance. Thus, it can be used in many downstream natural language processing tasks including the information extraction and sentence classification. The proposed model named the BILU Neural-Encoded Mention Hypergraph (BILU-NEMH) combines the BILU encoding schema, a mention-hypergraph model, and a neural network to conduct the mention-extraction task. The neural network is used to compute the feature scores for given edges/hyperedges using the encoded hypergraph model. Major contributions of this work can be summarized as follows.

- A Neural-Encoded Mention hypergraph (NEMH) model capable of automatic mention-entities recognition of a nested structure from the natural language texts is proposed.
- We combine an encoding schema with a hypergraph-based model, which can capture more boundary features than the models reported in the previous work, and these features are shown to be effective in the mention-extraction task.
- The deep learning models are incorporated into our encoded mention-hypergraph model, which improves the model performance significantly.
- The experiments are conducted to demonstrate that the proposed model can achieve competitive results on the standard datasets compared to the previously reported models.

2. Literature review

In this section, we introduce the frequently used sequence prediction models. The traditional models for sequence prediction include the Hidden Markov Model (HMM) [3–5], Max-Entropy Model (MEM) [2], Conditional Random Fields (CRFs) [18], and semi-Markov Random Fields (semi-CRFs) [30]. Also, these are linear models that can capture correlations between the labels in the neighborhoods, and model the distribution of the entire label sequence. Leonard et al. [3–5] introduced an HMM that can be represented as a dynamic Bayesian network. When the HMM is applied to a sequence-prediction task, the states (i.e., tags) are invisible to the models, while the outputs (i.e., words or segments), which are dependent on the states, are visible. Each state has a probability distribution of the so-called emission probabilities over the possible output tokens. Also, each state has the probability, called the transition probability, over the possible states. The sequence of tokens generated by the HMM model gives a probability distribution over all the possible label sequences. A forward-backward algorithm is used to find the unknown parameters of the HMM, and the Viterbi algorithm is used to find the most likely sequence of hidden states. Much research on sequence prediction is based on the HMMs. Fine et al. [10] designed a hierarchical hidden Markov model (HHMM), which denotes a recursive hierarchical generalization of the generic HMM. They used a systematic unsupervised approach to model a complex multi-scale structure, which appears in many natural-language texts. An efficient procedure was derived to estimate model parameters from unlabeled data. Zhang et al. [36] built the Institute of Computing Technology Chinese Lexical Analysis System (ICTCLAS), which uses an HHMM to incorporate the Chinese word segmentation, part-of-speech tagging, disambiguation, and unknown-word recognition in a comprehensive theoretical frame.

Shen et al. [29] used an HMM to perform the named entity recognition in the biomedicine field. In addition, the authors integrated some other useful features which were not used in the related work, including the deterministic features,

morphological features, POS tag features, and semantic trigger features. These features were used to capture evidence of the bio-medical named entities. They also proposed a rule-based approach and a simple algorithm to deal with the cascade phenomenon and the abbreviation problem.

Berge et al. [2] pioneered a Maximum Entropy Model (MEM) in the natural language processing.

Much research on sequence prediction has been based on the MEM. For instance, McCallum et al. [20] proposed a Maximum Entropy Markov Model (MEMM). The MEMM represents a discriminative graph model for sequence prediction. The observations were represented as plenty of overlapping features in the MEMM models. The conditional distribution of hidden states was defined using the maximum entropy framework to fit the exponential models. Yu et al. [33] applied continuous features in the MEM. They explained the reason why applying the moment constraint (MEMC) in the MEM worked well only with the binary features rather than the continuous features. They found that the weight values of the continuous features had to be functions rather than single-value scores. Thus, they proposed a spline-based solution to the MEM with the continuous functions to provide the feature scores. In their model, the optimization problem was converted to the optimization of a log-linear model in a high dimension space.

A statistical model proposed by Ratnaparkhi [26] was able to assign the POS tags to unseen texts with very high accuracy. Plenty of contextual features were used in their work. They also proved the effectiveness of the specialized features in POS recognition. A training method was also proposed to lighten the corpus consistency problem when the specialized features were applied to the model. Rosenberg et al. [27] proposed the Mixture-of-Parents Maximum Entropy Markov Model (MoP-MEMM). The proposed MoP-MEMM denotes a directed graphical model which restricts the conditional distribution of each node to a mixture of parent distribution to incorporate the long-range dependencies. The exact marginal posterior node distributions can be precisely computed by the models, which makes it possible to model the non-sequential correlations within the texts.

The CRF models were proposed by Lafferty et al. [18] in 2001. The CRF represents a strong baseline model which is commonly applied to the sequence prediction tasks. The CRF model relaxes the strong independence assumptions used in the HMMs and also avoids the problem that occurs in the MEM models which have a preference over states with a few successor states. A forward-backward algorithm is used to estimate the CRF model parameters, and a Viterbi algorithm is used for the sequence prediction. Tseng et al. [31] presented a Chinese Word Segmentation (CWS) system based on the CRF model. Their model uses a conditional random field sequence model to capture a large number of linguistic features such as character identity, as well as morphological and character reduplication features. Zhao et al. [34] regarded the CWS problem as a character-based tagging problem under a CRF framework. Instead of using the method focused only on a feature template, they considered both feature-template and tag-set selection. They demonstrated that there was a significant performance difference when different tag sets were selected. Zhao et al. [35] used a six-tag set together with the tone feature of Chinese characters and assisted segments trained on the other corpora to improve the CRF-based Chinese word-segmentation performance further. Finkel et al. [9] presented a feature-rich discriminative CRF parser for the first time. Their model was shown to be effective when applied to the full Wall Street Journal (WSJ) data. The Stochastic optimization technique, parallelization, and chart pre-filtering were their keys to success in terms of performance evaluation. Cuong et al. [32] considered the problem of incorporating the high-order dependencies between the labels or segments in the CRFs. They provided the efficient inference algorithm to handle such problems under the assumption that the number of distinct label patterns used in the features was small, and they experimentally showed that exploiting the high-order dependencies could lead to the substantial performance improvements. The Semi-Markov Conditional Random Fields (semi-CRF) was proposed by Sarawagi and Cohen [30]. The proposed semi-CRF outputs a segmentation of an input sequence x, where labels are assigned to each segment rather than to individual words. Notably, the semi-CRFs can capture features to measure the properties of each segment, and in each segment, transition features can be non-Markovian. Andrew [1] proposed a model which is capable of incorporating both CRF and semi-CRF features. A training and inference algorithm was also proposed. Compared with the conventional CRF model and the semi-CRF model, the proposed model reduced the error by 18 percent and 25 percent, respectively, in the Chinese word segmentation task. Additionally, they proposed a new feature for the model: a log-conditional odds that a given token sequence constitutes a chunk according to a generative model. Additionally, this feature can reduce the error by 13 percent. Nguyen et al. [24] incorporated the high-order semi-CRF features into the first-order semi-CRF models. An efficient algorithm was also proposed under the assumption that the higher-order semi-Markov features were sparse. They also showed that the proposed model outperformed the conventional first-order semi-CRF and high-order semi-CRF models in three sequence prediction tasks. Muis and Wei [23] proposed the weak semi-Markov CRFs for the noun-phrase chunking. In the conventional semi-CRF, the model intuitively decides the next segment length and type simultaneously, while in the weak semi-CRF, the model tries to propose a weaker variant that makes the two decisions separate by restricting each node to connect to only the nodes of the same label either in the next segment, or in the next word.

The deep learning methods show advantages in sequence prediction [6,15]. Huang et al. [15] proposed many deep learning models for sequence prediction, including the LSTM model, the Bi-LSTM model, and the LSTM-CRF model. Dyer et al. [6] proposed a model which is able to represent the state of a transition-based dependency parser. Three types of parser states can be captured: (i) unbounded buffer of incoming words; (ii) actions taken by a parser; (iii) the complete contents of the stack of the partially-built tree fragments, including their internal structures. A conventional forward and backward algorithm is used to train and infer the model.

Lample et al. [17] introduced two models, an LSTM neural network with a CRF layer, and a transition-based approach to construct and label the segments similar to the shift-reduce parser. Kong et al. [16] proposed a segmental recurrent neural

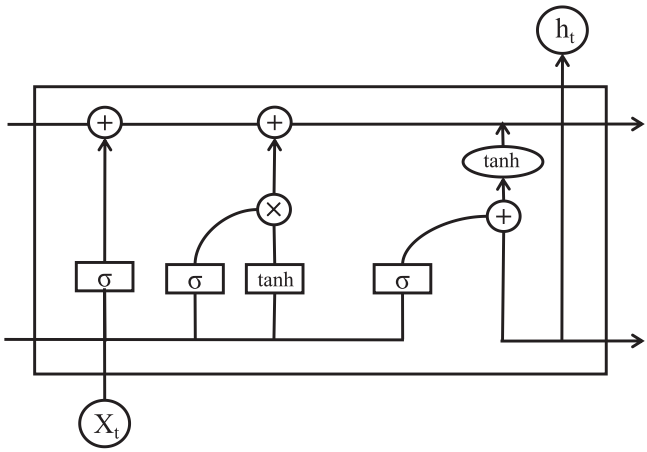

Fig. 2. A long short-term memory cell.

network, which denotes a variant of the semi-Markov CRF. Given an input sequence, the segmental recurrent neural network models a joint probability distribution over the segmentation of input and labels of the segments. Liu et al. [19] proposed a neural semi-Markov CRF, which composes the embedding of both input units and segments. They conducted the experiments on the NER and Chinese Word Segmentation (CWS) tasks. Their model achieved the state-of-the-art performance on the CWS benchmark dataset, and the competitive results on the CoNLL03 dataset. Zhuo et al. [37] proposed the gated recursive semi-CRFs (grSemi-CRFs). This model directly models the segments in the input sequence and uses a gated recursive convolutional neural network to extract the representations of each segment automatically. Ma and Hovy [25] proposed a CNN-LSTM-CRF model that benefits from both word-level and character-level representations. They evaluated their model on two datasets for two sequence-labeling tasks, namely the Part-Of-Speech Tagging (POS Tag) and the NER, and the obtained results showed that proposed model achieved the state-of-the-art performance on both datasets. Rei et al. [28] incorporated the character-level information to address the Out-Of-Vocabulary (OOV) problem in sequence prediction. They investigated the character-level extensions to the conventional LSTM-CRF structure models. Another character-level LSTM was adopted to encode the character-level information. The encoded character-level information was combined with the pre-trained word embeddings obtained by an attention mechanism, enabling the model to dynamically decide how much information to use from the word-level or character-level component. They evaluated their models on a range of sequence-labeling datasets and demonstrated an improvement in model robustness.

Finkel and Manning [8] proposed a tree-based discriminative constituency parser for mention extraction to recognize the nested named entities. Due to the tree structure, the proposed algorithm suffers from high time-complexity. Lu and Roth [12] used a hypergraph-based model to address the nested structure occurring in mention extraction. Muis and Lu [22] used a notion of mention separators, together with the multigraph representation to extract the mention entities. Their model can extract the mention entities form both nested and overlapping structures, and the model time complexity performance is mostly acceptable.

3. Preliminaries and problem statement

3.1. Recurrent neural networks

Recurrent Neural Networks (RNNs) denote a neural network type which is commonly used with sequential data. The RNNs take a sequence of vectors as an input, while the output is another sequence of vectors that represents the information about the input sequence at each time step. However, in practice, generic RNNs are difficult to train and fail to determine a long-term dependency. The LSTM neural networks [14] denote a variant of RNNs designed to deal with the mentioned issue. An LSTM unit is composed of three multiplicative gates to control the information flow. Fig. 2 illustrates the structure of an LSTM cell.

An LSTM unit is updated t by:

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

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

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

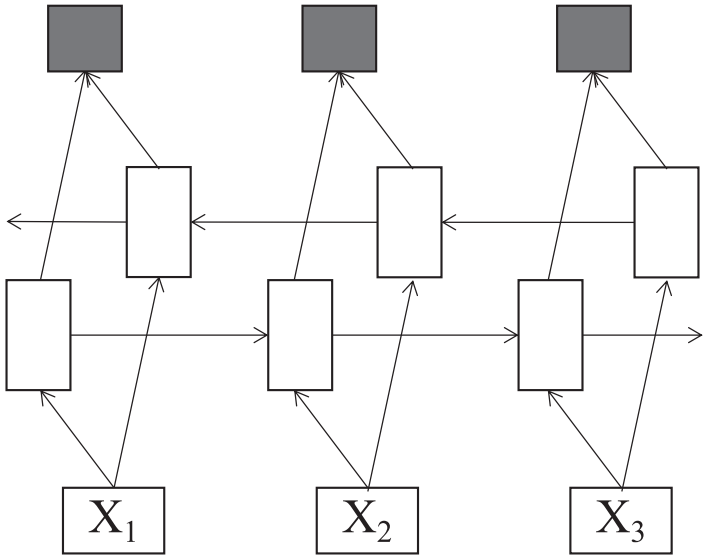

Fig. 3. The Bi-LSTM network structure.

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

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

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

where x_t denotes the input of an LSTM unit at time step t, σ denotes the sigmoid function, and $\odot$ is the product operation; further, h_t denotes 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 state h_t denotes the final output which can be considered as a vector representation of the input word.

In this work, we use a bidirectional LSTM (Bi-LSTM) [4] which can access both past (left) and future (right) contexts when extracting a hidden state at time step t. The Bi-LSTM is shown in Fig. 3. The round nodes at the bottom denote the input vectors, the dark-square nodes at the top denote the output vectors, and the rectangular nodes in the middle denote the LSTM units previously shown in Fig. 2. Basically, the Bi-LSTM inputs each sequence forward and backward to two separate LSTM neural networks, and at each time step, it concatenates the output of these two LSTM neural networks to form the final output.

3.2. Hypergraph

A hypergraph represents a generalization of a conventional graph whose edges (i.e., hyperedges) can connect two or more nodes, while in a conventional CRF model, an edge can connect only one node. In the proposed model, each hyperedge consists of a parent node and a list of child nodes. 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 (the features we use in this work are given in Section 4.3), W is the weight vector which is adjusted during the training phase, and $Z(x)$ is the normalization factor of all the possible segmentations s over x. Here, a dynamic programming technique is used to compute $Z(x)$ efficiently. To find the best label sequence in a hypergraph, assume that α_j indicates that 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 is 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 $(j-1, j, y)$.

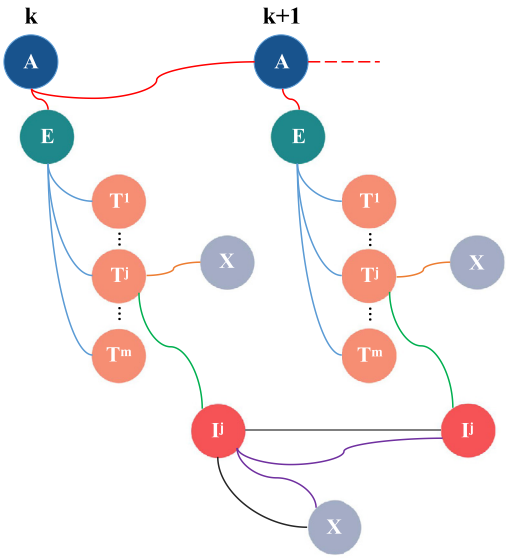

Fig. 4. The (partial) mention hypergraph model.

3.3. Neural hypergraph-based model

A neural hypergraph-based network can efficiently use the past input features in the LSTM layer and the other user-specified sparse features (e.g., the transition features or n-gram features) in the hypergraph layer. Many features are considered in this work, but for simplicity, here, we take the tag-transition feature as an example, considering a tag transition matrix $[A]$, where each $[A]_{i,j}$ models the transition score from the ith tag to the jth tag. It should be noted that the transition matrix is independent of the position. The neural network outputs the matrix consisted of scores $f_\theta([x]_i^T)$, which are called the neural features. Element $[f_\theta]_{i,t}$ of the matrix denotes the score of the ith tag at the tth word in sentence $[x]_i^T$, computed by the neural networks with parameter θ. The score of a sentence $[x]_i^T$ which is labeled with the label path $[i]_i^T$ is computed by the sum of transition and network scores, which is given by:

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

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

Problem statement: Given an input sequence $x = (x_1, \ldots, x_k)$ of length k, let $x_{a:\,b}$ denote its subsequence $(x_a, \ldots, x_b)$, where $a \le b \le k$. A mention entity is defined as a triad (u, v, y), which means the sub-sequence $x_{u:\,v}$ is associated with the mention entity label y. Given an input sequence x, the mention-extraction problem is defined as a problem of extracting all the mention entities from an input sequence x, where the mentions can be overlapped or nested.

4. Proposed mention hypergraph model

In this section, we briefly introduce the mention hypergraph model and our modified BILU neural-encoded mention hypergraph model. The mention hypergraph model [12] uses nodes and directed hyperedges to encode mentions of different types and lengths. A hypergraph represents a generalization of a conventional graph which uses hypheredges to connect two or more nodes. A partial mention hypergraph is displayed in Fig. 4, where it can be seen that this hypergraph contains all the possible label paths of the input sentences. In this model, the five following node types are used:

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

As shown in Fig. 4, each A_k node is connected to A_{k+1} and E_k nodes, which explains why the set of mentions that start at k or later represents a union of a set of mentions that start at position $(k+1)$ or later and those that start at position k. Each E_k node is connected to $(T_k^1, T_k^2, \ldots, T_k^m)$ through a hyperedge, where m denotes a number of mention types in texts. This hyperedge indicates that mentions starting at position k must be of m type. Each T_k^j node has two edges. It can be connected to: (1) an I_k^j node, indicating that a mention of type T^j starts at position k; or (2) an X node, indicating that no

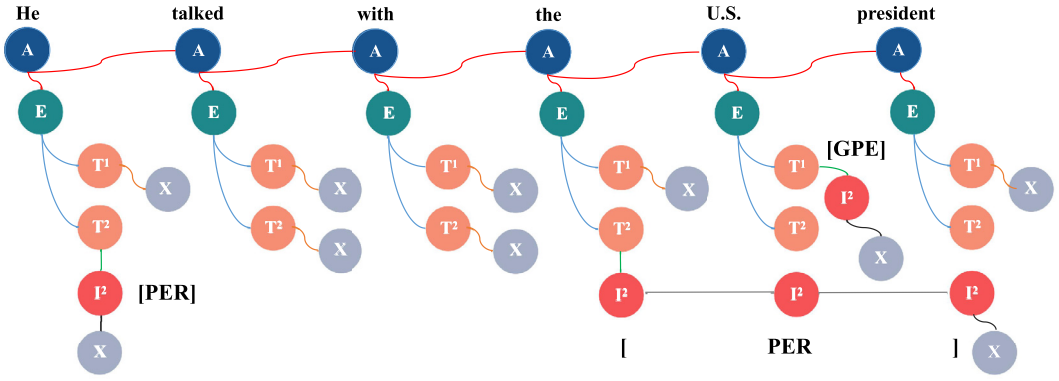

Fig. 5. The mention hypergraph model.

mentions of type j start at position k. Each I_k^j node has three edges. It can be connected to: (1) an I_{k+1}^j node, indicating that a mention of type j covers positions k and $(k+1)$; (2) an X node, indicating that a mention of type j ends at position k; or (3) both I_{k+1}^j and X nodes through a hyperedge, indicating that both cases 1 and 2 occur at position k. The example of a labeled sentence presented by a mention hypergraph model is shown in Fig. 5. In Fig. 5, the input sentence is: 'He talked with the U.S. president.' Thus, there are three mentions in this sentence, of which 'He' and 'the U.S. president' denote the PER mention entities, and 'U.S.' is the GPE mention entity. In Fig. 5, T^1 denotes the GPE mention entity and T^2 denotes the PER mention entity. In the first step, the proposed model uses I node to denote the word 'He' which is the PER mention entity. Since 'He' is not the GPE mention entity, the proposed model uses X node to denote the end. The word 'talked' and 'with' are not mention entities; thus, node T is directly connected to the corresponding node X. Further, 'the U.S. president' represents the PER mention entity; thus, the proposed model uses three I node to mark this mention entity. In the sentence part: 'the U.S. president', the 'U.S.' denotes the GPE mention entity, where the nested structure happens. Here, we connect the node I to node T^1 to obtain the GPE mention entity. The graph in Fig. 5 denotes the label path of the input sentence. In the training phase, we adjust the parameters to maximize the log likelihood of the graph presented in Fig. 5.

4.1. Encoded-mention hypergraph model

In the mention hypergraph model, a mention of type j is represented by a sequence of I^j nodes. Here, we try to add an encoding schema together with the additional edge connections to capture more boundary features for our hypergraph model. The proposed encoded mention hypergraph model is named the BILU (Beginning, Inside, Last, Unit length mention). More nodes and edge connections are added to the BILU-encoded hypergraph model. A part of the BILU encoded mention hypergraph model is depicted in Fig. 6. The eight following types of nodes are used in the proposed model:

- A_k denotes a mention which starts at position k or later.
- E_k denotes a mention whose left boundary is at position k.
- T_k^j denotes a mention of type j whose left boundary is at position k.
- B_k^j denotes a mention of type j starting at position k.
- I_k^j denotes a mention of type j covering position k.
- L_k^j denotes a mention of type j ending at position k.
- U_k^j denotes a mention of type j of a unit length at position k.
- X denotes the mention end.

As shown in Fig. 6, more edge connections are added. Each A_k node is connected to nodes A_{k+1} and E_k, and each E_k node is connected to $(T_k^1, T_k^2, \ldots, T_k^m)$ through a hyperedge, where m represents the number of mention types in the texts. These two nodes are the same as the nodes in the mention hypergraph model. Each T_k^j node can be connected to: (1) node U_k^j, indicating that a mention of type j has a unit length at position k; (2) node B_k^j, indicating that a mention of type j starts at position k and will continue to the next position; or (3) both U_k^j and B_k^j nodes through a hyperedge, indicating that both 1 and 2 cases occur at position k. Each B_k^j node can be connected to: (1) node I_{k+1}^j, indicating that a mention of type j starts at position k and continues to position $(k+1)$; (2) node L_{k+1}^j), indicating that a mention starts at position k and ends at position $(k+1)$; or (3) both I_{k+1}^j and L_{k+1}^j nodes through a hyperedge, indicating that both 1 and 2 cases occur at position k. Each U_k^j node can be connected only to the X node because the mention of a unit length starts and ends at the same

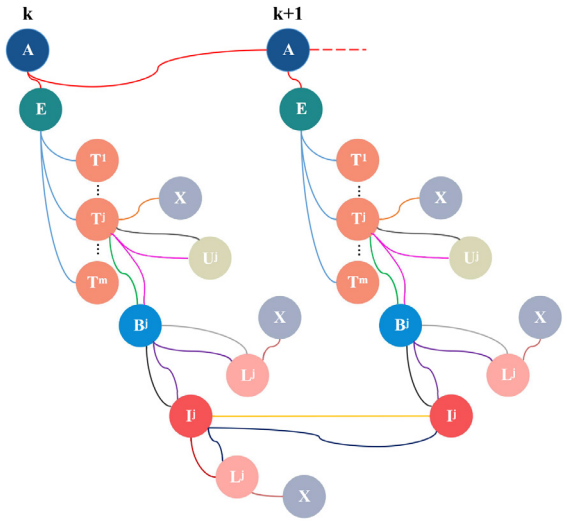

Fig. 6. The (partial) BILU-encoded mention hypergraph model.

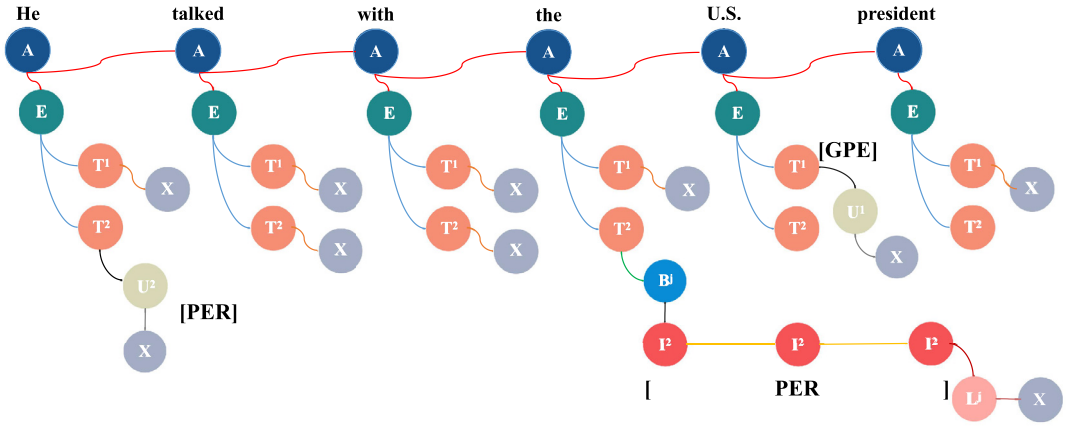

Fig. 7. The BILU-encoded mention hypergraph model.

position. Each I_k^j node can be connected to: (1) node I_{k+1}^j, indicating that a mention of type j covers positions k and $(k+1)$; (2) node L_{k+1}^j, indicating that a mention of type j covers position k and ends at position $(k+1)$; or (3) both I_{k+1}^j and L_{k+1}^j nodes through a hyperedge, indicating that cases 1 and 2 occur at the same time. Each L_k^j node can be connected to only the X node since it represents the last token of a mention. The example of representing a labeled sentence using the BILU encoded mention hypergraph model is shown in Fig. 7. In Fig. 7, the input sentence is: 'He talked with the U.S. president.' There are three mentions in this sentence; 'He' and 'the U.S. president' denote the PER mention entities, and 'U.S.' denotes the GPE mention entity. In Fig. 7, T^1 denotes the GPE mention entity and T^2 denotes the PER mention entity. In Step 1, the proposed model uses U^2 node to present the word 'He' which is the PER mention entity with a unit length. Since 'He' is not the GPE mention entity, the proposed model uses X node to denote the end. The words 'talked' and 'with' are not the mention entities; thus, node T is directly connected to the corresponding X node. The part 'the U.S. president' denotes the PER mention entity; therefore, the proposed model first uses node B to connect node T^2 to mark the begin of this PER mention entity. Then, three I nodes are used to mark the inside of the mention entity, and L node is used to mark the end of the same mention entity. In the sentence part 'the U.S. president', the word 'U.S.' denotes the GPE mention entity, where the nested structure happens. Here, we connect node U to node T^1 to obtain the GPE mention entity with a unit length. The graph in Fig. 7 denotes the label path of the input sentence. In the training phase, the parameters are adjusted to maximize the log likelihood of the graph presented in Fig. 7.

4.2. Neural-encoded mention hypergraph model

The encoded hypergraph model is combined with a neural network, obtaining the Bi-LSTM-CRF model, where the Bi-LSTM is used to compute the feature scores for the edges/hyperedges in our hypergraph model. For a given edge/hyperedge at position k, word embedding of word k^{th} denotes the input of the neural network, which outputs the feature scores for all the mention types through a linear/nonlinear transformation. We combine the neural network with the encoded mention hypergraph model by using the following function:

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

where $G(x, s)$ denotes the hypergraph feature score, w_1 and w_2 are the corresponding weights of the encoded mention hypergraph and neural network features, $N(x, s)$ denotes the neural feature score computed by the Bi-LSTM neural network, and $Z(x)$ denotes the normalization factor of all the possible label sequences over x. We use the maximum conditional likelihood estimation in the neural encoded mention hypergraph training. For a training set$\{(x_i, s_i)\}$, the log-likelihood is given by:

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

The maximum likelihood training chooses parameter W such that the log-likelihood $L_D(W)$ is maximized. The training algorithm is given in Algorithm 1 . Similar to the training algorithm of the conventional CRF model, the proposed neural

Algorithm 1: The neural-encoded mention hypergraph model training procedure.

```
for each epoch do
    for each batch do
        (1) Neural network forward pass for neural network state;
        (2) Encoded mention hypergraph forward and backward pass;
        (3) Neural network backward pass: backward pass for neural network ;
        (4) Update parameters.
```

hypergraph model first forwards the neural network (i.e., the Bi-LSTM neural network), and then the computed feature scores are combined with the hypergraph spare features. These features are used to make the forward and backward pass to update the hypergraph features (i.e., sparse features and neural network features). Finally, the updated neural network features are used to make the backward algorithm for updating the neural network parameters.

4.3. Features

In this section, all the hypergraph feature used to compute $G(x, s)$ by Eq. 10 are briefly introduced. The features we use are inspired by the work of Finkel et al. [9]. Specifically, we consider the following features defined over the input:

- **Word features:** Words that appear around the current position with a window size of 3.
- **POS tag features (if available):** POS tags that appear around the current position with a window size of 3.
- **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**: Bags of words with a window size of 5.
- **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.

Also, the method of Lu and Roth [12] is employed to incorporate the following feature into our model:

- **Mention penalty:** The number of hyperedges which connect nodes T and B is called the mention penalty.

By tuning the score of this feature, the model can learn the preference of the mentions that appear in input texts.

5. Experimental evaluation

In this section, the empirical evaluation of our proposed model is presented. Following the previous work [12,22], the experiments were conducted on the standard **ACE2004, ACE2005**, and **GENIA** datasets. The parameters of the datasets are summarized in Table 1, where the number in brackets represents the corresponding mention entity number. In our experiments, several baselines models were compared with the proposed model. Since the performance of the semi-CRF model and the model proposed by Xu and Jiang [32] are largely related to the hyper-parameter max span length n, in the experiments, n was set to 6 and ∞. In the following, the models used in the comparison are respectively denoted with the authors names and year of publication. Besides, it should be noted that F value given in the brackets indicates the use of the mention penalty feature introduced to optimize F value (From Tables 2–4).

Table 1
Statistics of standard datasets.

	#Train-sent	#Dev-sent	#Test-sent
ACE2004	6799 (22,207)	829 (2511)	879 (3031)
ACE2005	7336 (24,687)	958 (3217)	1047 (3027)
GENIA	14,835 (46,465)	1855 (5014)	1855 (5600)

Table 2
Results on the ACE2004 dataset.

	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
Lu and Roth (2015)	79.2	46.8	58.9
Lu and Roth (2015) (F)	70	56.9	62.8
Xu and Jiang (2016) (c=6)	68.2	54.3	60.5
Xu and Jiang (2016) (c=n)	57.3	46.8	51.5
Muis and Lu (2017)	79.5	51.1	62.2
Muis and Lu (2017) (F)	72.7	58	64.5
BILU-NEMH	82.28	53.78	65.04
BILU-NEMH (F)	76.41	58.66	**66.37**

Table 3
Results on ACE2005 dataset.

	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
Lu and Roth (2015)	76.9	47.7	58.9
Lu and Roth (2015) (F)	66.3	59.2	62.5
Xu and Jiang (2016) (c=6)	67.4	55.1	60.6
Xu and Jiang (2016) (c=n)	56.3	44.6	49.8
Muis and Lu (2017)	75.5	51.7	61.3
Muis and Lu (2017) (F)	69.1	58.1	63.1
BILU-NEMH	80.75	54.87	65.34
BILU-NEMH (F)	73.96	59.76	**66.11**

5.1. ACE2004 dataset

The experiments were conducted using the English portion of the ACE2004 dataset. In the experiments, 80 percent of data was used for model training, 10 percent of data was used as the developmental set (Dev-set), and the remaining 10 percent was used as the evaluation set (Test-set). The best performances are marked with the bold and underlined font from Tables 2 to 4.

As can be seen in Table 2, the proposed model yielded significantly better results than the other models, regardless of the F1-score was optimized or not. The BILU-NEMH model achieved a better performance than the other tested models. Moreover, the BILU-NEMH yielded the best performance when F optimization was adopted. Specifically, when the BILU encoding schema was used, our approach obtained much higher precision and recall, resulting in an improved F1-score. These results largely confirmed the effectiveness of the proposed neural-encoded mention hypergraph model. Additionally, as it was expected, the semi-CRF baseline yielded relatively lower results than the other models because it could not predict the mentions overlapping.

5.2. ACE2005 dataset

Similarly, the experiments were conducted using the English portion of the ACE2005 dataset. We considered all the documents from *bc, bn, nw*, and *wl*. As previously, 80 percent of the data was used for training, the other 10 percent was used for development, and the remaining 10 percent was used for model evaluation. In Table 3, the best performance on the development and test sets are marked with the bold and underlined font.

Table 4
Results on the GENIA dataset.

	Test Set		
	Precision	Recall	F-Value
CRF (BIO)	75.8	62.3	68.4
CRF (BILOU)	74.9	63.3	68.6
Semi-CRF (n=6)	76.2	61.7	68.2
Semi-CRF (n=∞)	75.4	61.1	67.5
Lu and Roth (2015)	79.2	57.39	66.6
Lu and Roth (2015) (F)	70.2	68.9	69.5
Xu and Jiang (2016) (c=6)	71.2	64.3	67.6
Xu and Jiang (2016) (c=n)	63.2	59.3	61.2
Muis and Lu (2017)	77.3	59.4	67.2
Muis and Lu (2017) (F)	71.3	67.8	69.5
BILU-NEMH	79.4	58.8	67.6
BILU-NEMH (F)	70.3	68.9	**69.6**

As shown in Table 3, the proposed model performed significantly better than the other models. After the F-value optimization, the F value of the BILU-NEMH model was better that of the other models. Although the model proposed by Muis and Lu [22] could handle the overlapping situation, its performance was worse than of our model. This could be explained by the fact that in the two datasets, most mention entities were in a nested structure rather than an overlapping structure. In addition, compared with the baseline semi-CRF model, using our model approximately 15 percent higher F-value could be handled for the nested structures.

5.3. GENIA dataset

The experiments were also conducted on the GENIA dataset (v3.02), which contained the overlapping bio-medical named entities. We followed the description given by Finkel et al. [8] to set up the experimental parameters for the GENIA dataset. Specifically, 90 percent of the data was used as training data, and the remaining 10 percent was used as evaluation data. All the DNA subtypes were collapsed into the DNA, all the RNA subtypes were collapsed into the RNA, and all the protein subtypes were collapsed into the protein. Only the cell lines and cell types were kept, and all the other entities were removed. We compared the performance of our model with the mention hypergraph [12], mention separators [22], FOFE [32], CRF models and semi-CRF models. The obtained results are shown in Table 4.

As shown in Table 4, the proposed model achieved a better F measure than the other tested models. However, it should be noted that the performance of all the models was quite similar. On the ACE2004 and ACE2005 datasets, the performance of the semi-CRF was about 13 percent poorer than that of the model that could handle nested structure mention entities. On the other hand, on the GENIA dataset, the difference in the performance between the tested models was only about 1 percent. This was because the proportion of the overlapping mentions in the GENIA dataset was quite small (18 percent), while it was higher than 40 percent in both ACE2004 and ACE2005 datasets. The obtained result is in agreement with that reported in [12,22].

6. Conclusion

In this paper, a Neural-Encoded Mention Hypergraph (NEMH) which uses the BLIU encoding schema (called the BILU-NEMH) is proposed for mention recognition task. The proposed model uses a hypergraph to recognize the overlapping mentions and achieves a time complexity of O(n) with respect to the input words. The neural network is employed to provide the features for the hypergraph model. The proposed model was evaluated by the experiments using several standard datasets. The empirical results showed that the proposed model achieved state-of-the-art performance on most datasets. In our future work, we will explore more different types of neural networks and encoding schemas. According to the previous works, different neural network structure could largely affect model performance. However, in this work, only the conventional LSTM neural network is used, but the other neural networks such as GRU and CNN will be explored in our future work. The encoding schema is another crucial point which affects the model performance, so more encoding schemas will be designed and used to enhance the model performance.

Conflict of interest

None.

References

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

[2] A.L. Berger, S.A.D. Pietra, V.J.D. Pietra, A maximum entropy approach to natural language processing, Comput. Linguist. 22 (1996) 39–71.
[3] L.E. Baum, T. Petrie, Statistical inference for probabilistic functions of finite state Markov chains, Ann. Math. Stat. 37 (6) (1966) 1554–1563.
[4] L.E. Baum, T. Petrie, G. Soules, N. Weiss, A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains, Ann. Math. Stat. 41 (1) (1970) 164–171.
[5] L.E. Baum, An inequality and associated maximization technique in statistical estimation of probabilistic functions of a Markov process, Inequalities 3 (1972) 1–8.
[6] C. Dyer, M. Ballesteros, W. Ling, A. Matthews, N.A. Smith, Transition based dependency parsing with stack long short term memory, in: Proceedings of the Association for Computational Linguistics, 2015, pp. 334–343.
[7] R. Florian, H. Hassan, A. Ittycheriah, H. Jing, N. Kambhatla, X. Luo, H. Nicolov, S. Roukos, A statistical model for multilingual entity detection and tracking, in: Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics, 2004, pp. 1–8.
[8] J.R. Finkel, C.D. Manning, Nested named entity recognition, in: Proceedings of the Conference on Empirical Methods in Natural Language Processing, 2009, pp. 141–150.
[9] J.R. Finkel, A. Kleeman, C.D. Manning, Efficient, feature-based, conditional random field parsing, in: Proceedings of the Association for Computational Linguistics, 2008, pp. 959–967.
[10] S. Fine, Y. Singer, N. Tishby, The Hierarchical Hidden Markov Model: Analysis and Applications, Kluwer Academic Publishers, 1998.
[11] P. Gupta, B. Andrassy, Table filling multi-task recurrent neural network for joint entity and relation extraction, in: Proceedings of the International Conference on Computational Linguistics, 2016, pp. 2537–2547.
[12] W. Lu, D. Roth, Joint mention extraction and classification with mention hypergraphs, in: Proceedings of the Conference on Empirical Methods in Natural Language Processing, 2015, pp. 857–867.
[13] S. Guo, M.W. Chang, E. Kiciman, To link or not to link? A study on end-to-end tweet entity linking, in: Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2013, pp. 1020–1030.
[14] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (8) (1997) 1735–1780.
[15] Z. Huang, W. Xu, K. Yu, Bidirectional LSTM-CRF models for sequence tagging, 2015. https://arxiv.org/pdf/1508.01991.pdf.
[16] L. Kong, C. C. Dyer, N.A. Smith, Segmental recurrent neural networks, in: Proceedings of the ICML, 2016.
[17] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, C. Dyer, Neural architectures for named entity recognition, in: Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2016, pp. 260–270.
[18] J.D. Lafferty, A. Mccallum, F.C.N. Pereira, Conditional random fields: probabilistic models for segmenting and labeling sequence data, in: Proceedings of the International Conference on Machine Learning, 2001, pp. 282–289.
[19] Y. Liu, W. Che, J. Guo, Q. Bin, T. Liu, Exploring segment representations for neural segmentation models, in: Proceedings of the International Joint Conference on Artificial Intelligence, 2016, pp. 2880–2886.
[20] A. McCallum, D. Freitag, F.C.N. Pereira, Maximum entropy Markov models for information extraction and segmentation, in: Proceedings of the International Conference on Machine Learning, 1999, pp. 591–598.
[21] M. Mintz, S. Bills, R. Snow, D. Jurafsky, Distant supervision for relation extraction without labeled data, in: Proceedings of the Annual Meeting of the Association for Computational Linguistics-International Joint Conference of the Asian Federation of Natural Language Processing, 2009, pp. 1003–1011.
[22] A.O. Muis, W. Lu, Labeling gaps between words: recognizing overlapping mentions with mention separators, in: Proceedings of the Conference on Empirical Methods in Natural Language Processing, 2017, pp. 2598–2608.
[23] A.O. Muis, W. Lu, Weak semi-markov CRFs for noun phrase chunking in informal text, in: Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics, 2016, pp. 714–719.
[24] V.C. Nguyen, N. Ye, W.S. Lee, L.C. Hai, Semi-Markov conditional random field with high-order features, J. Mach. Learn. Res. 15 (1) (2014) 981–1009.
[25] X. Ma, E. Hovy, End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF, in: Proceedings of the Association for Computational Linguistics, 2016, pp. 1064–1074.
[26] A. Ratnaparkhi, A maximum entropy model for part-of-speech tagging, in: Proceedings of the Conference on Empirical Methods on Natural Language Processing, 2011, pp. 133–142.
[27] D.S. Rosenberg, K. Dan, B. Taskar, Mixture-of-parents maximum entropy Markov models, in: Proceedings of the Conference on Uncertainty in Artificial Intelligence, 2007, pp. 318–325.
[28] M. Rei, G.K.O. Crichton, S. Pyysalo, Attending to characters in neural sequence labeling models, in: Proceedings of the International Conference on Computational Linguistics, 2016, pp. 309–318.
[29] D. Shen, J. Zhang, G. Zhou, J. Su, C.L. Tan, Effective adaptation of a hidden Markov model-based named entity recognizer for biomedical domain, in: Proceedings of the ACL Workshop on Natural Language Processing in Biomedicine, 2003, pp. 49–56.
[30] S. Sarawagi, W.W. Cohen, Semi-Markov conditional random fields for information extraction, in: Proceedings of the Neural Information Processing Systems, 2004, pp. 1185–1192.
[31] H. Tseng, P. Chang, G. Andrew, D. Jurafsky, C. Manning, A conditional random field word segmenter for sighan bakeoff 2005, 2015, pp. 168–171.
[32] M. Xu, H. Jiang, A FOFE-based local detection approach for named entity recognition and mention detection, in: Proceedings of the Association for Computational Linguistics, 2016, pp. 1237–1247.
[33] D. Yu, L. Deng, A. Acero, Using continuous features in the maximum entropy model, Pattern Recognit. Lett. 30 (14) (2009) 1295–1300.
[34] H. Zhao, C.N. Huang, M. Li, T. Kudo, An improved chinese word segmentation system with conditional random field, in: Proceedings of the SIGHAN Workshop on Chinese Language Processing, 2006, pp. 162–165.
[35] H. Zhao, C.N. Huang, M. Li, B.L. Lu, Effective tag set selection in chinese word segmentation via conditional random field modeling, in: Proceedings of the Twentieth Pacific Asia Conference on Language, Information and Computation, 2006, pp. 87–94.
[36] H.P. Zhang, Q. Liu, X.Q. Cheng, H. Zhang, H.K. Yu, Chinese lexical analysis using hierarchical hidden Markov model, in: Proceedings of the Sighan Workshop on Chinese Language Processing, volume 17, 2003, pp. 63–70.
[37] J. Zhuo, Y. Cao, J. Zhu, B. Zhang, Z. Nie, Segment-level sequence modeling using gated recursive semi-Markov conditional random fields, in: Proceedings of the Association for Computational Linguistics, 2016, pp. 1413–1423.