Mention Detection Using Pointer Networks for Coreference Resolution

Cheoneum Park, Changki Lee, and Soojong Lim

A mention has a noun or noun phrase as its head and constructs a chunk that defines any meaning, including a modifier. Mention detection refers to the extraction of mentions from a document. In mentions, coreference resolution refers to determining any mentions that have the same meaning. Pointer networks, which are models based on a recurrent neural network encoder–decoder, outputs a list of elements corresponding to an input sequence. In this paper, we propose mention detection using pointer networks. This approach can solve the problem of overlapped mention detection, which cannot be solved by a sequence labeling approach. The experimental results show that the performance of the proposed mention detection approach is F1 of 80.75%, which is 8% higher than rule-based mention detection, and the performance of the coreference resolution has a CoNLL F1 of 56.67% (mention boundary), which is 7.68% higher than coreference resolution using rule-based mention detection.

Keywords: Coreference resolution, Deep learning, Mention detection, Pointer networks.

I. Introduction

Coreference resolution, which is a natural language processing (NLP) task, refers to the relationship between any words in a document that indicate the same entity. Coreference resolution is based on mentions, and the method for extracting mentions in a document is known as mention detection [1]–[3].

A mention consists of a head, which is key to a noun phrase, as well as modifiers that modifies the head. As mention detection determines the precise modifier information contained in the mention, the meaning of the mention becomes clear. It is then possible to perform NLP tasks (for example, coreference resolution and relation extraction) to which mention detection is applied correctly.

Pointer networks (Ptr-net) [4], which are based on a recurrent neural network (RNN) encoder-decoder [5], provides a location corresponding to the given input sequence as an output result. A Ptr-net can solve problems precisely for a variable output class by applying the attention mechanism [6]. In this paper, we propose that mention detection use the Ptr-net expanded from an RNN. In this way, we can solve the overlapped mentions detection problem (that is, mention detection for all mentions), which could not be solved using an existing sequence labeling problem.

To summarize, the main contributions of this paper are as follows:

1. We solve mention detection for all overlapped and single mentions using the position-based Ptr-net, which is an extended model of RNN architecture.
2. We achieve state-of-the-art performance for mention detection in Korean.
3. We help to improve coreference resolution performance by using mention detection with the Ptr-net.

Manuscript received Feb. 23, 2017; revised July 12, 2017; accepted July 19, 2017.

Cheoneum Park (parkce@kangwon.ac.kr) and Changki Lee (corresponding author, leeck@kangwon.ac.kr) are with the Department of Computer Science, Kangwon National University, Chuncheon, Rep. of Korea.

Soojong Lim (isj@etri.re.kr) is with the SW & Content Research Laboratory, ETRI, Daejeon, Rep. of Korea.

The remainder of this paper is organized as follows. We discuss related research in Section II and mention detection in Section III. In Section IV, we explain the proposed mention detection using the Ptr-net, and in Section V, we present the experimental results. Finally, we provide the conclusions of the study and discuss the scope for future research in Section VI.

II. Related Work

Mention detection has been researched extensively for coreference resolution and information extraction, among others (for example, MUC 6, MUC 7, CoNLL 2011, CoNLL 2012, SemEval, and ACE). We can show an increase in the recall of coreference resolution with more mentions because the mention is the unit of any entity (that is, the subject) in the coreference resolution. Standard mention detection was processed using rule-based [1]–[3] and statistics-based [7]–[13] approaches.

Rule-based mention detection is performed based on syntactic analysis and certain rules defined by humans [1]–[3]. First, [1], [2] extracted mentions using a left-to-right, breadth-first search based on phrase structure analysis and certain rules. Then, [3] extracted nouns or noun phrases as a head based on a dependency tree, and mention detection defined a mention boundary using a modifier and head. Rule-based mention detection can extract the correct mentions to define the appropriate boundary for a relatively short noun phrase such as a normally single noun, compound noun, or head with a short-range modifier. However, with the rule-based approach, it is difficult to extract mentions with the correct boundary when the word is in a complex sentence or has long-range modifiers. Therefore, problems may occur, for example, a cross of mentions, loss of modifier information, or accumulating errors (depending on part-of-speech tagging, syntactic analysis, and other factors) [14]–[16].

Statistics-based mention detection traditionally defines mention detection as a sequence labeling problem and extracts mentions using adapted machine learning models such as hidden markov model (HMM), maximum entropy markov model, and conditional random filed (CRF) [7]–[9]. The study in [7] transferred knowledge from rich-resource to low-resource languages via machine translation, and [8] performed mention detection using noisy input and a combination of various features. References [9], [10] showed effective performance by combining mention detection with other tasks such as relation of event extraction. In addition, [9] used the HMM model, [10] used the structured perceptron model, and [11] constructed an information network based on the interdependency of the relation and event. Statistics-based mention detection can exhibit superior performance via learning and combining various features. However, feature extraction and combination are costly and time consuming.

In order to solve the problems of rule- and statistics-based mention detection, [12], [13] proposed statistics mention detection using deep learning (for example, bidirectional gated recurrent unit (GRU) and bidirectional LSTM-CRF). In addition, [12] showed performance that was superior to the existing methods using a bidirectional GRU based on the Elman or Jordan method in an English dataset (ACE 2005). Reference [13] showed superior performance in rule-based mention detection when extracting a long-range mention boundary using the bidirectional LSTM-CRF in a Korean dataset. However, the statistics-based mention detection learned and extracted only the mention having the longest mention boundaries in a sentence because this method cannot be applied to all mentions owing to mentions overlapping in a sentence.

In this study, it is possible to detect overlapping mentions, which pose a limitation in the existing statistics-based mention detection.

III. Mention Detection

Mention is a default input unit of coreference resolution for clustering any words referring to the same object into an entity by resolving their coreference. The mention consists of a noun or noun phrase, and certain mentions exist within a sentence or phrase. We define outer and inner mentions: an outer mention exists on the outside of the mention (that is, an overlapped mention), while an inner mention is contained in the overlapped mentions, except for the outer mentions. These are processed using rule-based and statistics-based mention detection. Table 1 shows an example of mention detection.

In Table 1, all mentions are extracted according to the unit of the word based on the noun and noun phrase. We define the beginning and end of any mention as the boundary, which is represented using square brackets ([ ]). In Table 1, we classify the boundary of a mention as the boundary of the inner mention (a thin square bracket) and that of the outer mention (a bold square bracket). For example, "*Yi Sun-shin, who is a military subject in*

Table 1. Example of mention detection.

Input sentence
난중일기(亂中日記)는 조선 중기의 무신(武臣) 이순신(李舜臣)이 임진왜란의 7년(1592 ~ 1598년) 동안 군중에서 썼다. 이것은 1962년 12월 20일 대한민국의 국보 제76호로 지정된 일기이다.
Nan-jung-il-gi-neun joseon jungki-ui musin yisunshin-i imjinwaeran-ui 7 nyeon (1592–1598 nyeon) dongan gunjung-eseo sseoss-da. igeot-eun 1962 nyeon 12 wol 20 il daehanminguk-ui gukbo jae 76 horo jijeong-doen ilgi-i-da.
A War Dairy is written during 7 years in the Japanese Invasion of Korea (1592–1598) by Yi Sun-shin who is a military subject in the middle of the Joseon Dynasty.
Mention detection result
[난중일기(亂中日記)는**]** **[**[조선 중기의] 무신(武臣)] 이순신(李舜臣)이**]** **[**[임진왜란의] 7년(1592 ~ 1598년) 동안**]** **[**군중에서**]** 썼다. **[**이것은**]** **[**[[1962년 12월 20일] [대한민국의] 국보] 제76호로] 지정된 일기이다.**]**
[Nan-jung-il-gi-neun**]** **[**[joseon jungki-ui] musin] yisunshin-i**]** **[**[imjinwaeran-ui] 7 nyeon (1592–1598 nyeon) dongan**]** **[**gunjung-eseo**]** sseoss-da. **[**igeot-eun**]** **[**[[[1962 nyeon 12 wol 20 il] [daehanminguk-ui] gukbo] jae 76 horo] jijeong-doen ilgi-i-da.**]**
[A War Diary] is written during [[7 years (1592–1598)] in [the Japanese Invasion of [Korea]]] by [Yi Sun-shin] [who is [a military subject in [the middle of [the Joseon Dynasty.]]]]

the middle of the Joseon Dynasty" is defined as mentions such as [*조선 중기의 무신(武臣) 이순신(李舜臣)*] ([*Yi Sun-shin, who is a military subject in the middle of the Joseon Dynasty*]), [*조선 중기의 무신(武臣)*] ([*who is a military subject in the middle of the Joseon Dynasty*]), [*조선 중기*] ([*the Joseon Dynasty*]), where the longest mention, [*조선 중기의 무신(武臣) 이순신(李舜臣)*] ([*Yi Sun-shin, who is a military subject in the middle of the Joseon Dynasty*]), is the outer mention and others are the inner mentions in the overlapped mentions.

The sequence labeling problem occurs with overlapped target classes such as [B-B-B, I-I-I, O-I-I, O-O-I], when the output sequence for the inner and outer mentions is represented using the BIO tags (B: beginning, I: inside, O: outside). Training is therefore challenging because of the increasing number of target classes when the training is processed using all of the overlapping classes.

The output class of the mention detection using the Ptr-net proposed in this study is the position corresponding to the input sequence, not the BIO tags. This model can be trained with overlapped mentions (that is, the structure of the inner and outer mentions) because the structure of the training data for the Ptr-net consists of the indexes for the beginning and end boundaries of the mention (see Section VI-1).

IV. Mention Detection with Pointer Networks

1. Pointer Networks

The Ptr-net, which is a model based on RNN, outputs a list of elements corresponding to the input sequence. With this model, it is possible to train the output class for variable lengths using the attention mechanism. The Ptr-net generates the hidden state of the encoder and trains the attention weight for which position to watch in the input sequence by inputting the hidden state of the decoder that has been generated with the hidden state of the encoder thus far.

$$\vec{h}_s = GRU\left(E(x_s), \vec{h}_{s-1}\right),$$
$$\overleftarrow{h}_s = GRU\left(E(x_s), \overleftarrow{h}_{s+1}\right),$$
$$h_s^{enc} = [\vec{h}_s; \overleftarrow{h}_s].$$

In the formula, the hidden states of the forward and backward networks over an input sequence are $\vec{h}_s$ and $\overleftarrow{h}_s$, respectively. $E(x_s)$ is the word-embedding function for word x_s at a time s in the input sequence, while h_s^{enc} is a concatenation of the two vectors $\vec{h}_s$ and $\overleftarrow{h}_s$, and contains the summaries of both the preceding and the following inputs.

The decoding process using the soft-attention mechanism is defined as follows:

$$h_t^{\text{dec}} = GRU(h_t^{\text{enc}}, h_{(t-1)}^{\text{dec}})$$
$$a_t(s) = \frac{\exp(\text{score}(h_t^{\text{dec}}, h_s^{\text{enc}}))}{\sum_{s'} \exp(\text{score}(h_t^{\text{dec}}, h_{s'}^{\text{enc}}))},$$

where

$$\text{score}\left(h_t^{\text{dec}}, h_s^{\text{enc}}\right) = \begin{cases} v_t^T \tanh\left(W_a\left[h_t^{\text{dec}}; h_s^{\text{enc}}\right]\right), concat \\ v_t^T \tanh\left(W_a\left[h_t^{\text{dec}}; h_t^{\text{enc}}; h_s^{\text{enc}}\right]\right), concat2, \end{cases}$$
$$y_t = \text{argmax}_{s'}(a_t(s')).$$

In the above equation, h_t^{dec} is the hidden state of the decoder, with h_t^{enc} and the previous hidden state of the decoder as inputs. The value $a_t(s)$, the attention weight, is the value normalized by softmax to the result value of the function $\text{score}\left(h_t^{\text{dec}}, h_s^{\text{enc}}\right)$. This weight determines the position corresponding to the input sequence. The $\text{score}\left(h_t^{\text{dec}}, h_s^{\text{enc}}\right)$ function, which is a method between *concat* and *concat*2, calculates the alignment score. The value *concat* is a method of calculating the score by concatenating h_t^{dec} and h_s^{enc}, while *concat*2 is a method of calculating the score by concatenating h_t^{dec}, h_t^{enc}, and h_s^{enc}. We calculate the result of the optimized output sequence by using a beam search.

https://doi.org/10.4218/etrij.17.0117.0140

Figure 1 represents the result of mention detection using a pointing structure for the following sentence: "*세종대왕은 조선의 4대 군주이며 언어학자이다* (*Sejongde awang-eun joseon-ui gunju-i-myeo eoneohakja-i-da*)." Each mention consists of the boundary of the beginning mention ([) and that of the end mention (]), where the boundary of the end mention determines the boundary of the beginning mention. For example, the mention [*세종대왕/NNP 은/JX* (*sejongdeawang-eun*)] has the boundary of the end mention as the index of "*은/JX* (*eun*)" and the boundary of the beginning mention as the index of "*세종대왕/NNP* (*sejongdeawang*)." Other mentions have the same pointing structure. The Korean sentence consists of morphemes, which are expressed as "lemma/type."

In Fig. 2, which shows the structure of the Ptr-net model for *concat* and *concat*2 of the function score$\left(h_t^{\text{dec}}, h_s^{\text{enc}}\right)$, the input for the encoder is a sentence, and the input sequence is X = [*세종대왕/NNP* (*sejongdeawang*), *은/JX* (*eun*), *조선/NNP* (*joseon*), *의/JKG* (*ui*), *4/SN*, *대/NNB* (*dea*), *군주/NNG* (*gunju*), *이/VCP* (*i*), *며/EC* (*myeo*), *언어학자/NNG* (*eoneohakja*), *이/VCP* (*i*), *다/EC* (*da*), *<EOS>*]. <EOS> is the end symbol of the sentence. The input for the decoder is assumed to be the last boundary of the mentions. Y_{input}, which is the input $[h_0^{\text{enc}}, h_1^{\text{enc}}, h_2^{\text{enc}}, \ldots, h_s^{\text{enc}}]$ for the decoder, is the hidden state list of the encoder (the hidden states of the positions for *세종대왕/NNP* (*sejongdeawang*), *은/JX* (*eun*), *조선/NNP* (*joseon*), *의/JKG* (*ui*), *4/SN*, *대/NNB* (*dea*), *군주/NNG* (*gunju*), *이/VCP* (*i*), *며/EC* (*myeo*), *언어학자/NNG* (*eoneohakja*), *이/VCP* (*i*), *다/EC* (*da*)).

The output sequence of the decoder corresponding to the input consists of the position indexes that are represented by the beginning boundaries of all mentions, such as Y_{output} = [12, 0, 12, 2, 12, 4, 12, 12, 2, 12, 12, 2] [indexes for *<EOS>*, *세종대왕/NNP* (*sejongdeawang*), *<EOS>*, *조선/NNP* (*joseon*), *<EOS>*, *4/SN*, *<EOS>*, *<EOS>*, *조선/NNP* (*joseon*), *<EOS>*, *<EOS>*, *조선/NNP* (*joseon*)]. The input of each decoder points to the

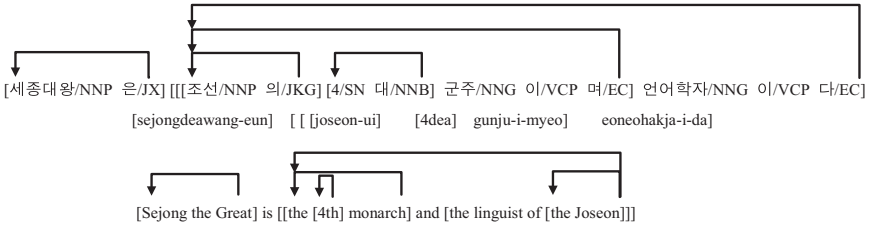

Fig. 1. Pointing structure of mention boundary.

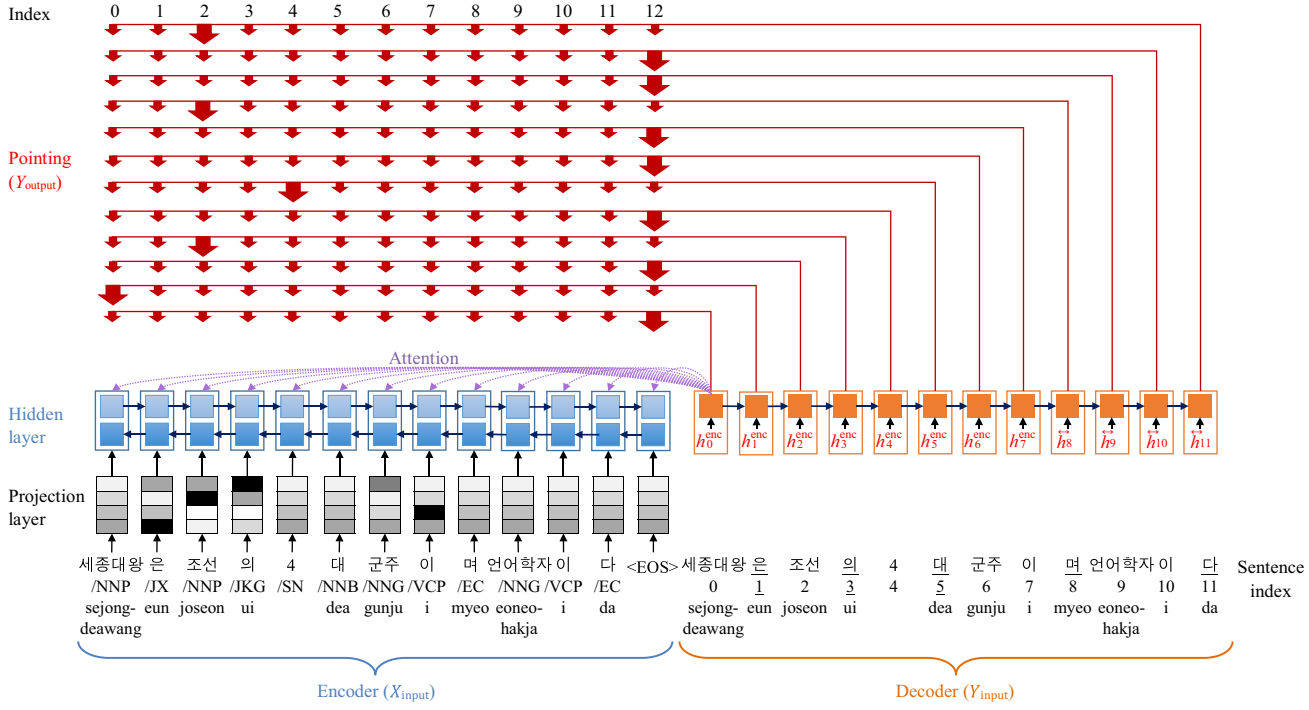

Fig. 2. Structure of pointer networks model.

beginning boundary of the input mention as a result of the output. (If the input is not a mention, it points to the position of <*EOS*>).

In the input sequence in Fig. 2, we represent the position with the highest alignment score using a bold arrow. For example, the output result of "*세종대왕/NNP* (*sejongdeawang*)" inputted as the beginning of the decoder is the index 12 (<*EOS*>), which is the highest score in the input sequence. The next decoding predicts the index 0 [*세종대왕/NNP* (*sejongdeawang*)] of the output result with "*은/JX*" as input. The output result of a decoder that is inputted as "*조선/NNP*" is the index 12 (<*EOS*>), and the pointer network prediction ends once all inputs of the decoder have been predicted.

2. Input Criterion of Mention Detection

We defined the four input criteria based on morphemes in Table 2 and experimented accordingly because the morpheme is the input criterion when mention detection using the Ptr-net is processed. The input criterion is the position of the morpheme that represents the eojeol[1] in the eojeol of the mention boundary position. Table 2 lists the input criteria of the morpheme, which represents the eojeol of the decoder for the three mentions. The input sentence in Table 2 is changed to a morphological representation because the input method proposed in this paper is based on the morphemes (each morpheme is expressed as "lemma/type:position_index" in Table 2).

V. Experiment Result

We conducted experiments on the Korean and English portions of the ETRI data for Korean, and SemEval 2010 for English. The ETRI data consist of quiz and news domains, as shown in Table 3.

The experiments in this study for mention detection using the Ptr-net were conducted as follows. Korean word embedding is used to train the Sejong corpus for 100,000 words using the neural network language model [17]. We used 50-dimension word embedding from SENNA [18]. In order to measure the performance of the mention detection, we converted the output results of the Ptr-net to BIO tags. We evaluated the results of the mention detection using the F1 measure.

In the experiments, we found the optimized performance that followed the input criterion, and performed a comparative experiment for the training method and methods of the scoring function (*concat* and *concat*2). We performed the optimization for hyperparameters (for example, activation function,

Table 2. Input criterion with word of beginning boundary and head for mention.

Input sentence
[이곳에는] [사람 얼굴 형상의 거대한 석상이] 있는데 [무엇일까?] [Igot-e-neun] [saram eolgul hyeongsang-ui geodaehan seoksang-i] itt-neun-dae [mu-eot-il-kka?] [What] is there [a huge statue of [a human face]] in [this place]?
Morpheme information of 3 mentions for input sentence
Mention 1: [이곳/NP:0 에/JKB:1 는/JX:2] Mention 2: [사람/NNG:3 얼굴/NNG:4 형상/NNG:5 의/JK G: 6 거대/NNG:7 하/XSA:8 ㄴ /ETM:9 석상/NNG:10 이/JKS: 11] Mention 3: [무엇/NP:14 이/VCP:15 ㄹ 까/EF:16 ?/SF:17]

Input criterion	Decoder input structure
*Md*0	The beginning morpheme index of each inputted eojeol. [이곳/NP:0 → 이곳/NP:0] [석상/NNG:10 → 사람/NNG:3] [무엇/NP:14 → 무엇/NP:14] [igot/NP:0 → igot/NP:0] [seoksang/NNG:10 → saram/NNG:3] [mu-eot/NP:14 → mu-eot/NP:14]
*Md*1	The end morpheme index of each inputted eojeol. [는/JX:2 → 는/JX:2] [이/JKS:11 → 사람/NNG:3] [?/SF:17 → ?/SF:17] [neun/JX:2 → neun/JX:2] [i/JKS:11 → saram/NNG:3] [?/SF:17 → ?/SF:17]
*Md*2	The end morpheme index if the eojeol is the head of the mention (containing the josa* and symbol), the beginning morpheme index if the eojeol is the beginning boundary of the mention. [는/JX:2 → 이곳/NP:0] [이/JKS:11 → 사람/NNG:3] [?/SF:17 → 무엇/NP:14] [neun/JX:2 → igot/NP:0] [i/JKS:11 → saram/NNG:3] [?/SF:17 → mu-eot/NP:14]
*Md*3	The end morpheme index if the eojeol is the head of the mention (removing the josa and symbol), the beginning morpheme index if the eojeol is the beginning boundary of the mention. [이곳/NP:0 → 이곳/NP:0] [석상/NNG:10 → 사람/NNG:3] [무엇/NP:14 → 무엇/NP:14] [igot/NP:0 → igot/NP:0] [seoksang/NNG:10 → saram/NNG:3] [mu-eot/NP:14 → mu-eot/NP:14]

*Josa is a part-of-speech tag in Korean that affects the meaning of a word or its grammatical relation to another word.

[1] Eojeol is a word, which consists of a sentence. The eojeol is the smallest unit of sentence components and the unit of spacing.

https://doi.org/10.4218/etrij.17.0117.0140

Table 3. Corpus statistics.

ETRI dataset			
Domain	Train	Dev	Test
QA	1,935	240	240
News	130	10	10
SemEval dataset			
N/A	Train	Dev	Test
N/A	229	39	85

Table 4. Comparison of mention detection following input criterion (%, dev).

Input criterion	Precision	Recall	F1
*Md*0	76.86	75.90	76.38
*Md*1	73.25	72.26	72.75
*Md*2	79.65	78.63	79.14
*Md*3	79.26	79.14	**79.20**

Table 5. Comparison of mention detection following learning method (%, dev).

Cost function	Precision	Recall	F1
Momentum	79.26	79.14	**79.20**
RMSprop	79.27	77.28	78.26

Table 6. Comparison of mention detection following activation function (%, dev).

Activation function	Precision	Recall	F1
[tanh, tanh]	79.26	79.14	**79.20**
[tanh, sigm]	77.49	79.05	78.26
[tanh, ReLU]	77.89	76.48	77.18
[sigm, tanh]	78.71	78.63	78.67
[sigm, sigm]	78.63	77.45	78.03
[sigm, ReLU]	79.92	78.27	79.08

Table 7. Performance of mention detection following attention scoring method (%, dev).

Attention scoring method	Precision	Recall	F1
concat	79.26	79.14	79.20
*concat*2	79.45	79.05	**79.25**

dropout, and hidden layers of sizes), and we performed the comparative experiment for the mention detection using the proposed Ptr-net and the existing methods. We followed the training methodology where the Ptr-net starts with a learning rate of 0.05. If there is no performance improvement, the learning rate decreases by 50%.

Table 8. Optimization of drop-out for *concat*2 (%, dev).

Activation scoring method	Drop-out	Precision	Recall	F1
*concat*2	0.0	79.45	79.05	79.25
	0.1	79.48	80.72	**80.10**
	0.3	80.74	78.68	79.70
	0.5	80.12	78.49	79.30
	0.7	78.25	76.33	77.28

Table 9. Optimization dimension of hidden layer for *concat*2 (%).

Activation scoring method	Dimension of $[[\overleftrightarrow{h_s}, h_t]$	F1 (%, dev)	F1 (%, test)
*concat*2	[100, 50]	80.10	N/A
	[200, 100]	**80.51**	79.89
	[400, 200]	79.76	N/A
	[800, 400]	80.39	N/A
	[1,600, 800]	80.30	N/A

Table 10. Comparison of mention detection following dimension of word embedding (%, dev).

Dimension of word embedding	Precision	Recall	F1
50	81.49	79.55	**80.51**
100	79.89	80.67	80.28
200	80.33	78.83	79.57

Table 11. Improvement of performance for mention detection using pretraining (%).

Drop-out	Dev			Test
	Precision	Recall	F1	F1
0.0	80.37	80.16	80.27	N/A
0.1	80.61	80.04	80.32	N/A
0.3	80.99	79.48	80.23	N/A
0.5	81.13	80.01	**80.57**	80.07
0.7	80.83	79.87	80.35	N/A

We measured the performance of rule-based coreference resolution based on a model showing the optimal results of the mention detection using CoNLL F1 [1]–[3] (see Table 14).

Tables 4–12 show the tuning of the hyperparameters in the experiments. We use a hyperparameter that shows the

Table 12. Experiment results for mention detection (%, test).

Model	Long boundary F1	All boundary F1
Rule-based MD [5]	44.08	72.42
Bi-LSTM CRF based MD	**76.24**	N/A
Pointer Networks based MD	73.23	**80.07**

optimal performance in each experiment. The initial hyperparameters are as follows: the learning method is *momentum*, the activation function is [tanh, tanh] (that is, the activation function of the hidden layer and of the attention layer), the attention score function is *concat* with a drop-out of 0, hidden layers have a size of [100, 50] (that is, $[h_s^{\text{enc}}, h_t^{\text{dec}}]$), and the dimension of word embedding is 50.

Table 4 shows the experiment following the input criteria of the Ptr-net and using the development dataset. In the experimental results, *Md*3 showed the best performance, with an F1 of 79.20% in the input criteria. In the comparison experiments in Tables 5 through 12, we use the *Md*3 input criterion with the best performance in Table 4 as input.

Table 5 shows a performance comparison of the momentum and RMSprop for the training. The results show that momentum achieves a performance of approximately 0.94% higher than RMSprop.

Table 6 shows the results of a combination of activation functions, which each contain a hidden layer and an attention layer. The most effective combination of activation functions is [tanh, tanh], with a performance of an F1 of 79.20%. The next best combination of activation functions, at 79.08%, is [sigm, ReLU].

Table 7 shows the results of the comparison experiment for alignment scoring, where *concat*2 shows an F1 of 79.25%, which is 0.05% higher than *concat*.

Table 8 summarizes the drop-out optimizations for *concat2*, which shows superior performance in Table 7. The F1 value shows the best performance at 80.10% for the development dataset when the drop-out is 0.1.

Table 9 shows that the number of hidden layer units is optimized for a drop-out of 0.1, which shows optimal performance in Table 8. The F1 value shows the best performance at 80.51% when the number of hidden layer units is [200, 100] for the development dataset, while another F1 value shows 79.89% for the test dataset.

Table 10 optimizes the word embedding dimension using the hyperparameters showing the best performance in the previous experiments. The results indicate that the word embedding has the best performance, with an F1 of 80.51% at 50 dimensions.

Table 11 shows that performance improves when fine-tuning is performed with the optimal hyperparameters obtained in the previous experiments, using the pretraining model. Drop-out is set to between 0.1 and 0.5, and shows higher performance in Table 8. When the drop-out is 0.4, the best performance has an F1 of 80.57% in the development dataset shown. At that time, the F1 of the test dataset is 80.07%, which shows an increase of approximately 0.18% compared with the pretrained performance of 79.89% (see Table 9).

Table 12 shows the experimental results of mention detection for the method proposed in this paper, as well as the rule-based and Bi-LSTM-CRF model-based methods. Our method shows an F1 of 73.23%, which is approximately 3.01% lower than that of Bi-LSTM-CRF in the case of a long mention boundary. However, our method is approximately 29.15% higher than the rule-based mention detection and can detect not only long boundary mentions but also all mentions when the Ptr-net is used. Our model shows an F1 of 80.07% for all mentions, which is approximately 7.65% higher than that of the rule-based mention detection.

Table 13 compares the results of applying the proposed mention detection method to English with the performance of other English mention detections [10]–[12]. The Korean data used in the proposed model is measured based on the optimal hyperparameters updated thus far, and for the English data (SemEval 2010). We measure the performance after optimization using the same method as the hyperparameter optimization performed previously.

We retrain the drop-out at 0.5 after pretraining with the following hyperparameters: the input criteria is the direction from the end of a mention to the beginning of a mention (therefore, the same as *Md*0 because English does

Table 13. Comparison of English mention detection (%, test).

MD model (long boundary)	Precision	Recall	F1
Joint system [10]	85.20	76.90	80.80
Joint system [11]	85.10	77.30	81.00
BIDIRECT [12]	83.70	81.80	**82.70**
Ptr_net based mention detection	76.33	74.40	75.36
MD model (all boundary)	**Precision**	**Recall**	**F1**
Ptr_net based mention detection	81.03	78.40	79.71

https://doi.org/10.4218/etrij.17.0117.0140

not have morphemes), the learning method is *momentum*, the activation functions are [tanh, tanh], the attention scoring method is *concat*2, the drop-out is 0.1, the dimension of the hidden layer is [100, 100], and the word-embedding dimension is 50.

The compared mention detection uses the ACE 2005 dataset, and performance is shown for the joint system [10], [11] and BIDIRECT [12] using the BILOU tags (B: beginning, I: inside, L: last, O: outside, U: unit) for long boundary mentions. Reference [10] uses the joint extraction system for entity mentions and relations based on a pipeline, and [11] uses an information network to unify the outputs of three information extraction tasks (entity mentions, relations, and events) using a structured perceptron. [12] uses mention detection with a bidirectional RNN.

The mention detection for English shows higher performance than our model in Table 14. However, [10], [11] were designed for feature extraction and combination by a person, and with [10]–[12] it is impossible to process overlapped mentions because BILOU tags are used.

Table 14 shows the performance results of the coreference resolution based on mention detection using the Ptr-net as proposed in this paper. In the case of coreference resolution based on mention detection using the Ptr-net, the performance of CoNLL, with an F1 of 56.67%, is 7.68% higher than that of rule-based mention detection. Thus, it can be seen that the method proposed in this paper helps to improve coreference resolution performance.

Table 14. Experiment results for coreference resolution following mention detection approach (%).

MD model		Precision	Recall	CoNLL F1
Rule-based MD	M.B.	51.64	46.63	48.99
Ptr_net-based MD	M.B.	59.34	54.23	**56.67**

Figure 3 shows the results for cases where only long boundary mentions are detected, and all mentions when all overlapped mentions are detected. Each mention is separated by square brackets ([ ]); the subscripts in the brackets indicate the indexes of the mentions, and the superscripts indicate the indexes of the entities. An entity is a cluster of words that refer to each other. The arrows represent the coreference between two mentions.

As shown in Fig. 3(a), if only the mentions with only a long boundary are detected for the input sentence, coreference resolution is not performed because it does not use the overlapped mentions in the long boundary mention. On the other hand, if all mentions including the overlapped mentions are detected as shown in Fig. 3(b), the coreference resolution can be performed because all candidates are detected for performing the coreference resolution. For example, all mentions of "[[선한 영]$^{3}_{4}$ 스펜타 마이뉴와]$^{3}_{3}$ [[악한 영]$^{5}_{6}$ 앙그라 마이뉴의]$^{5}_{5}$ ([[*seonhan young*]$^{3}_{4}$ *spentas mainyu-woa*]$^{3}_{3}$ [[*akhan young*]$^{5}_{6}$ *angra mainyu-ui*]$^{5}_{5}$ / [[*the good spirit*]$^{3}_{4}$ *Spentas mainyu,*]$^{3}_{3}$ *and* [[*the evil spirit*]$^{5}_{6}$ *Angra mainyu.*]$^{5}_{5}$)," which were not detected in the mention detection with only long boundaries, were detected. As a result, [선한 영]$^{3}_{4}$ ([*seonhan young*]$^{3}_{4}$ / [*the good spirit*]$^{3}_{4}$) and [선한 영 스펜타 마이뉴와]$^{3}_{3}$ ([*seonhan young spentas mainyu-woa*]$^{3}_{3}$ / [*the good spirit Spentas mainyu,*]$^{3}_{3}$) are referred to each other, and [악한 영]$^{5}_{6}$ ([*akhan young*]$^{5}_{6}$ / [*the evil*

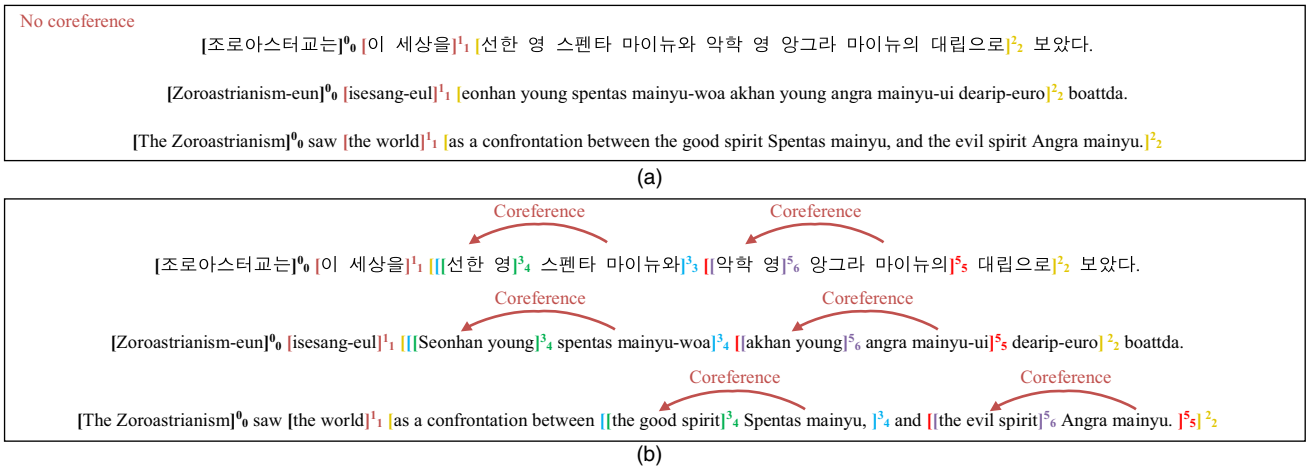

Fig. 3. Difference for mention detection between only long boundaries and all boundaries: (a) result of mention detection for only long boundary mentions (no coreference) and (b) result of mention detection for all boundary mentions (it is available for coreference resolution).

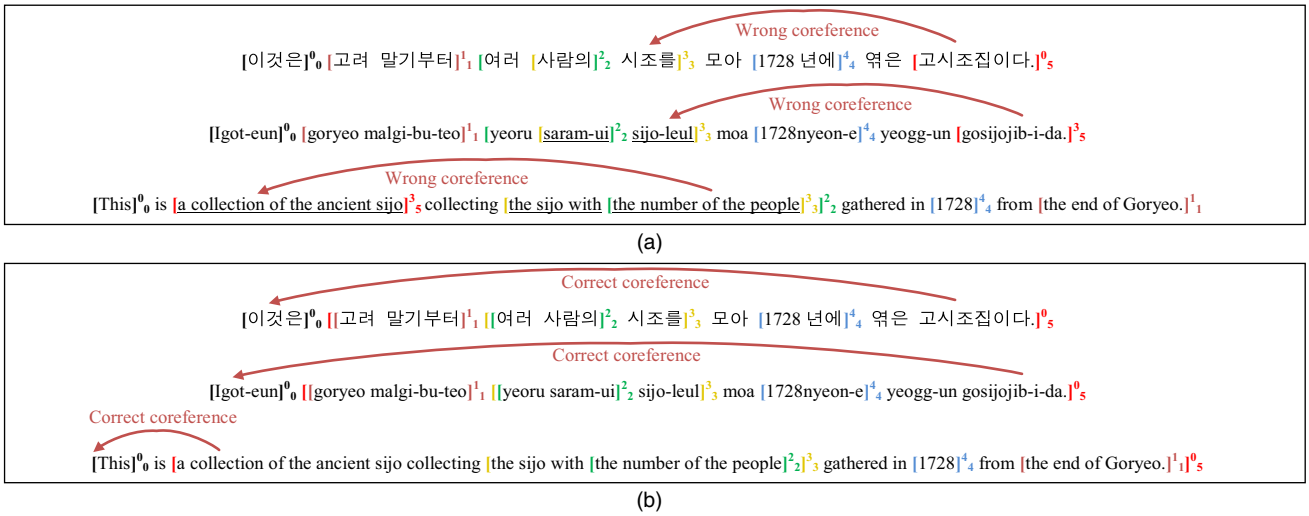

Fig. 4. Comparison of coreference resolution: (a) result of coreference resolution using rule-based mention detection and (b) result of coreference reoslution based on mention detection using pointer networks.

spirit]$^{5}_{6}$) and [*악한 영 앙그라 마이뉴의*]$^{5}_{5}$ ([*akhan young angra mainyu-ui*]$^{5}_{5}$ / [*the evil spirit Angra mainyu.*]$^{5}_{5}$) are referred each other.

Figure 4 shows the results of coreference resolution using rule-based mention detection and coreference resolution using our Ptr-net based mention detection.

In the case of coreference resolution using the rule-based mention detection, as in Fig. 4(a), mentions may be those that have lost modifier information such as [*고시조집이다.*]$^{3}_{5}$ ([*gosijojib-i-da.*]$^{3}_{5}$ / [*a collection of the ancient sijo*]$^{3}_{5}$), or occur at the crossing of mentions such as [*여러* [*사람의*]$^{2}_{2}$ *시조를*]$^{3}_{3}$ ([*yeoru* [*saram-ui*]$^{2}_{2}$ *sijo-leul*]$^{3}_{3}$ / [*the sijo with* [*the number of the people*]$^{3}_{3}$]$^{2}_{2}$). In such a case, the mistaken coreference resolution [referring to [*사람의 시조를*]$^{3}_{3}$ ([*saram-ui sijo-leul*]$^{3}_{3}$ / [*the sijo with the number of the people*]$^{3}_{3}$] and [*고시조집이다.*]$^{3}_{5}$ ([*gosijojib-i-da.*]$^{3}_{5}$/[*a collection of the ancient sijo*]$^{3}_{5}$)] is caused by the absence of a modifier. However, in the case of coreference resolution based on mention detection using the Ptr-net [Fig. 4(b)], the boundary of all mentions can be correctly defined. Therefore, the above problems are solved, and the correct coreference resolution [referring to [*이것은*]$^{0}_{0}$ and [*고려 ~ 고시조집이다.*]$^{0}_{5}$ ([*goryeo ~ gosijojib-i-da.*]$^{0}_{5}$ / [*a collection ~* [*the end of Goryeo.*]$^{1}_{1}$]$^{0}_{5}$)] is performed.

VI. Conclusion

In this study, we performed mention detection including overlapped mentions using the Ptr-net, and we applied the method to coreference resolution. Moreover, we performed experiments that use the two input methods of the Ptr-net (method for all morphemes and method for minimum morphemes), the input criteria of a boundary word for the beginning and head of a mention, and two scoring functions (*concat* and *concat*2), among others. Our model demonstrated superior performance, with an F1 of 80.07%, when hyperparameters were set as mention detection using the Ptr-net, *Md*3 of the input criterion, and *concat*2 for the scoring function. Furthermore, coreference resolution based on our model showed a CoNLL F1 of 56.67%, which is 7.68% higher than the performance of the rule-based coreference resolution. Therefore, it can be seen that the proposed method shows better performance than the rule-based mention detection and is helpful for solving coreference resolution.

In future work, we will build on the training dataset for coreference resolution and mention detection, improve the Ptr-net model to be a model such as a hierarchical RNN, and apply the model to coreference resolution and mention detection. Moreover, we will apply the Ptr-net to other boundary detection tasks such as named entity recognition, as well as NLP tasks (for example, a dependency parsing and semantic role labeling), which can be applied to the pointing concept.

Acknowledgements

This work was supported by Institute for Information & Communications Technology Promotion (IITP) grant funded by the Korean Government (MSIP) (No. 2013-0-00131, Development of Knowledge Evolutionary WiseQA Platform Technology for Human Knowledge Augmented Services). This research was supported by Basic Science Research Program through the National

Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT & Future Planning (No.NRF-2016R1C1B1014124).

References

[1] H. Lee et al., "Deterministic Coreference Resolution Based on Entity-Centric, Precision-Ranked Rules," *Comput. Linguist.*, vol. 39, no. 4, 2013, pp. 885–916.

[2] A. Haghighi and D. Klein, "Coreference Resolution in a Modular, Entity-Centered Model," *Human Language Technol. Annu. Conf. North American Chapter Assoc. Comput. Linguistics*, Los Angeles, CA, USA, June 1–6, 2010, pp. 385–393.

[3] C. Park et al., "Korean Coreference Resolution with Guided Mention Pair Model Using Deep Learning," *ETRI J.*, vol. 38, no. 6, Dec. 2016, pp. 1207–1217.

[4] O. Vinyals et al., "Pointer Networks," *Adv. Neural Inform. Process. Syst.*, vol. 12, 2015, pp. 2692–2700.

[5] K. Cho et al., "Learning Phrase Representation Using RNN Encoder-Decoder for Statistical Machine Translation," *Proc. of EMNLP '14*, 2014.

[6] D. Bahdanau et al., "Neural Machine Translation by Jointly Learning to Align and Translate," *Proc. ICLR '15*, 2015, arXiv:1409.0473.

[7] I. Zitouni and R. Florian, "Mention Detection Crossing the Language Barrier," *Proc. Conf. Empirical Methods Natural Language Process.*, Honolulu, HI, USA, Oct. 25–27, 2008, pp. 600–609.

[8] R. Florian, "Improving Mention Detection Robustness to Noisy Input," *Proc. Conf. Empirical Methods Natural Language Process*, Cambridge, MA, USA, Oct. 9–11, 2010, pp. 335–345.

[9] D. Roth and W. Yih, "Global Inference for Entity and Relation Identification via a Linear Programming Formulation," in *Introduction to Statistical Relational Learning*, Cambridge, MA, USA: MIT Press, 2007, pp. 553–580.

[10] Q. Li and H. Ji, "Incremental Joint Extraction of Entity Mentions and Relations," *Proc. Annu. Meeting Assoc. Comput. Linguistics*, Baltimore, MD, USA, June 23–25, 2014, pp. 402–412.

[11] Q. Li et al., "Constructing Information Neural Networks Using One Single Model," *Proc. Conf. Empirical Methods Natural Language Process.*, Doha, Qatar, Oct. 25–29, 2014, pp. 1846–1851.

[12] T.H. Nguyen et al., "Toward Mention Detection Robustness with Recurrent Neural Networks," arXiv preprint, 2016, arXiv:1602.07749.

[13] C. Park and C. Lee. "Mention Detection Using Bidirectional LSTM-CRF Model," *Proc. KIISE HCLT*, Rep. of Korea, 2015, pp. 224–227.

[14] A. Soraluze et al., "Mention Detection: First Steps in the Development of a Basque Coreference Resolution System," KONVENS Conf. Natural language Process., Vienna, Austria, Sept. 19–21, 2012, pp. 128–136.

[15] J.K. Kummerfeld et al., "Mention Detection: Heuristics for the OntoNotes Annotations," *Proc. Conf. Comput. Natural Language Learn.: Shared Task*, Portland, OR, USA, June 23–24, 2011, pp. 102–106.

[16] C. Park and C. Lee, "Mention Detection in the Coreference Resolution Using the Deep Learning," *Proc. KIISE and the KBS Joint Symp.*, Rep. of Korea, 2015.

[17] C. Lee, J. Kim, and J. Kim, "Korean Dependency Parsing using Deep Learning," *Proc. KIISE HCLT*, Rep. of Korea, 2014, pp. 87–91.

[18] R. Collobert et al., "Natural Language Processing (almost) from Scratch," *J. Mach. Learn. Res.*, vol. 12, 2011, pp. 2493–2537.

Cheoneum Park received his BS and MS degrees in computer science from Kangwon National University, Chuncheon, Rep. of Korea, in 2014 and 2016, respectively. He is now a PhD student at Kangwon National University. His research interests include natural language processing, question answering, machine learning, and deep learning.

Changki Lee received his BS degree in computer science from the Korea Advanced Institute of Science and Technology, Daejeon, Rep. of Korea, in 1999. He received his MS degree and PhD in computer engineering from POSTECH, Pohang, Rep. of Korea, in 2001 and 2004, respectively. From 2004 to 2012, he was a researcher with the Electronics and Technology Research Institute (ETRI), Daejeon, Rep. of Korea. Since 2012, he has been a professor of computer science at Kangwon National University, Chuncheon, Rep. of Korea. His research interests include natural language processing, machine learning, and deep learning.

Soojong Lim received his BS degree in mathematics from Yonsei University, Seoul, Rep. of Korea, in 1997. He also received his MS degree and PhD in computer science from Yonsei University, in 1998 and 2014, respectively. He is currently a principal researcher at ETRI, Daejeon, Rep. of Korea. His research interests include natural language processing, machine learning, and question answering.