# Fast End-to-end Coreference Resolution for Korean

**Cheoneum Park[1], Jamin Shin[2], Sungjoon Park[3,4], Joonho Lim[5], Changki Lee[6]**
[1]AIRS Company, Hyundai Motor Group, Republic of Korea
[2]Riiid AI Research, Republic of Korea
[3]Korea Advanced Institute of Science and Technology, Republic of Korea
[4]Upstage AI Research, Republic of Korea
[5]Electronics and Telecommunications Research Institute, Republic of Korea
[6]Kangwon National University, Republic of Korea
cheoneum.park@hyundai.com, jshin49@gmail.com, sungjoon.park@kaist.ac.kr,
joonho.lim@etri.re.kr, leeck@kangwon.ac.kr

## Abstract

Recently, end-to-end neural network-based approaches have shown significant improvements over traditional pipeline-based models in English coreference resolution. However, such advancements came at a cost of computational complexity and recent works have not focused on tackling this problem. Hence, in this paper, to cope with this issue, we propose BERT-SRU-based Pointer Networks that leverages the linguistic property of head-final languages. Applying this model to the Korean coreference resolution, we significantly reduce the coreference linking search space. Combining this with Ensemble Knowledge Distillation, we maintain state-of-the-art performance 66.9% of CoNLL F1 on ETRI test set while achieving 2x speedup (30 doc/sec) in document processing time.

## 1 Introduction

Coreference resolution is one of the fundamental sub-tasks for Machine Reading Comprehension and Dialogue Systems that groups mentions of a same entity in a given sentence or document (Soon et al., 2001; Raghunathan et al., 2010; Ng, 2010; Lee et al., 2013). Recently, for English coreference resolution, span-based end-to-end trained models such as `e2e-coref` (Lee et al., 2017), `c2f-coref` (Lee et al., 2018), and `BERT-coref` (Joshi et al., 2019b) have shown to outperform previous rule-based or mention-pairing approaches.

However, such approaches suffer from the computational complexity effectively-being $O(n^4)$, where $n$ is the length of the input document. Furthermore, as coreference resolution is a very important and complicated task, most of the research efforts have been focused on how to solve the problem through better modeling, such as higher-order coreference resolution (Lee et al., 2018). Inevitably, these approaches lead to more complicated models that are more computation heavy, but there are not many studies on solving this complexity issue. Hence, this paper aims to cope with this problem by infusing relevant linguistic features into the model.

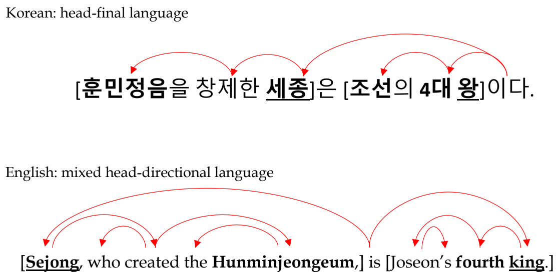
Korean: head-final language
[훈민정음을 창제한 세종]은 [조선의 4대 왕]이다.
English: mixed head-directional language
[Sejong, who created the Hunminjeongeum,] is [Joseon's fourth king.]

Figure 1: The brackets are mention boundaries, bold-faced words are the nouns, and underlined words are the heads of each mention. The red line is a dependency relation arc and Korean (top) shows the left-branching property (Dryer, 2009) where the heads are always at the end of the mention. On the other hand, the head locations for English is different across mentions.

One of the underlying reasons for such high computational complexity was the creation of $O(n^2)$ spans caused by the *mixed head directionality* of English, as shown in Figure 1. This makes it hard to locate the heads in the mentions because the head location is not deterministic. On the other hand, having deterministic head locations is a very desirable linguistic trait for solving the aforementioned computational complexity issue. This effectively reduces the search space for coreference linking as we can use only the heads of the mentions.

Korean is not only a new domain for end-to-end coreference resolution but also considered a strongly *head-final* language (Kwon et al., 2006), which motivates us to focus on Korean. In this paper, we present the first end-to-end model in Korean coreference resolution. Our model leverages such head-final properties using Pointer Net-

*Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2610–2624
November 16 - 20, 2020. ©2020 Association for Computational Linguistics

works (Vinyals et al., 2015) and achieves comparable performance to that of state-of-the-art models with a 2x speedup. Our contributions can be summarized as the following:

- First end-to-end coreference resolution model for Korean
- 2x speed up than state-of-the-art models
- Achieve state-of-the-art with Ensemble and maintain 2x speedup using Knowledge Distillation

## 2 Background

Coreference resolution is basically about linking *mention pairs* (which are often noun phrases). Essentially, this is finding *heads* of noun phrases that refer to the same entity, but the head locations within mentions are unknown. While previous rule-based approaches (Wiseman et al., 2016; Clark and Manning, 2016a,b) relied on several hand-engineered features including head-related ones, recent end-to-end methods attempt to directly model the mention distribution using span-based neural networks.

**Span-based Coreference Resolution** To elaborate, Lee et al. (2017, 2018); Joshi et al. (2019b) have formulated the task of end-to-end coreference resolution for English as a set of decisions for every possible *spans* in the document. The input is a document consisted of $n$ words and there are $S = \frac{n(n+1)}{2} = O(n^2)$ possible spans in it. For each span, the task is to assign an antecedent that refers to the same entity. Hence, as all of these spans have to be ranked against each other, the final coreference resolution search space is $\frac{S(S+1)}{2} = O(n^4)$. Finally, the entity resolutions are recovered by grouping all spans that are connected.

**Head-final Coreference Resolution** In this section, we introduce the concept of our proposed head-final coreference resolution. Head-final languages are left-branching in which the heads of mention phrases are at the end of the phrase (Dryer, 2009). This allows to easily extract accurate coreference linking between nouns across the mentions and use them for training directly. On the other hand, in English, it is impossible to know which nouns in the mentions are supposed to be linked together because the head locations are non-deterministic. Hence, using such head-final property, we can effectively reduce a *search over span candidates* to a ***search over head candidates***, which are simply the nouns. In short, this yields a coreference resolution search space of $O(n^2)$.

## 3 BERT-SRU Pointer Networks

We propose a novel model, BERT-SRU Pointer Network, that is suitable for head-final coreference resolution. This model combines bidirectional encoder representation from transformer (BERT) (Devlin et al., 2019) with bidirectional simple recurrent units (SRUs) (Lei et al., 2017) and Pointer Networks (Vinyals et al., 2015), as shown in Figure 2, to perform the head-final coreference resolution. Initially, the encoder part (which is BERT) receives morphologically analyzed texts along with their POS-tags as inputs. Then the decoder extracts the hidden state corresponding to the *head candidates* (which are all nouns) and uses them as the inputs. After that, the gated self-attention layer in decoder models head information, and the decoder outputs position corresponding to the input using the pointer networks. We use deep biaffine (Dozat and Manning, 2016) as the attention score of the pointer networks, and this model performs both the mention detection and the coreference resolution.

### 3.1 Model Inputs

To elaborate on the BERT encoder layer, we use a BERT model that is pre-trained with morphologically analyzed large-scale Korean corpus and apply byte pair encoding (BPE) (Sennrich et al., 2016) to the input morpheme sequence. When using BPE, we add a [CLS] and [SEP] token to the beginning and end of the input sequence and distinguish morphemes on the subword by attaching '_' in the last syllable of morphemes. We use features that are appropriate for Korean coreference resolution. The features are morpheme boundary, word boundary, dependency parsing, named entity recognition (NER), and candidate head distance.

#### 3.1.1 Input Text Preprocessing

The following example shows the use of morphological analysis and BPE for a given raw text. In the example below, the entity is 바카스 (Bacchus).

- **Raw text**: ”그리스 로마 신화에서 바카스라고도 불리는 술의 신” (A god of wine called Bacchus in Greek Rome mythology)
- **Morphological analysis with POS tagging**: ”그리스/NNP 로마/NNG 신화/NNG 에

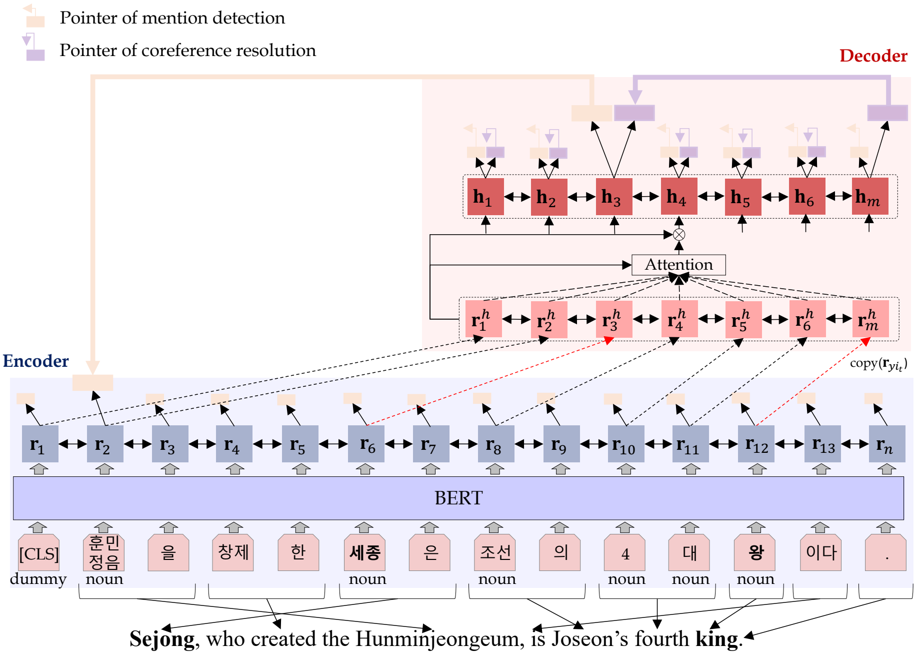
Pointer of mention detection
Pointer of coreference resolution
Decoder
$\mathbf{h}_1$ $\mathbf{h}_2$ $\mathbf{h}_3$ $\mathbf{h}_4$ $\mathbf{h}_5$ $\mathbf{h}_6$ $\mathbf{h}_m$
Attention
$\mathbf{r}_1^h$ $\mathbf{r}_2^h$ $\mathbf{r}_3^h$ $\mathbf{r}_4^h$ $\mathbf{r}_5^h$ $\mathbf{r}_6^h$ $\mathbf{r}_m^h$
Encoder
copy($\mathbf{r}_{y_{i_t}}$)
$\mathbf{r}_1$ $\mathbf{r}_2$ $\mathbf{r}_3$ $\mathbf{r}_4$ $\mathbf{r}_5$ $\mathbf{r}_6$ $\mathbf{r}_7$ $\mathbf{r}_8$ $\mathbf{r}_9$ $\mathbf{r}_{10}$ $\mathbf{r}_{11}$ $\mathbf{r}_{12}$ $\mathbf{r}_{13}$ $\mathbf{r}_n$
BERT
[CLS] 훈민정음 을 창제 한 세종 은 조선 의 4 대 왕 이다 .
dummy noun noun noun noun noun noun
Sejong, who created the Hunminjeongeum, is Joseon's fourth king.

Figure 2: Our Fast Head-final coreference resolution model for Korean. We use BERT to obtain embeddings corresponds to input tokens. Along with the five features, embeddings are used as SRU encoder inputs, and encoder outputs are fed to SRU-based decoder inputs through self-attention. Finally, decoder outputs are transformed to predict 1) mention start boundary, and 2) coreference resolution. All parameters are trained through an end-to-end manner. After training, the model could further be extended with ensemble knowledge distillation, achieving comparable performance to that of the state-of-the-art with 2x inference speed.

서/JKB 바카스/NNP 이/VCP 라고/EC 도/JX 불리/VV 는/ETM 술/NNG 의/JKG 신/NNG"

- **Applying BPE**: "그리스/NNP_ 로 마/NNG_ 신화/NNG_ 에서/JKB_ 바 카스/NNP_ 이/VCP_ 라고/EC_ 도/JX_ 불리/VV_ 는/ETM_ 술/NNG_ 의/JKG_ 신/NNG_"

If the following input text is given, morphological analysis is performed using a part-of-speech (POS)-tagger and BPE is applied. In this paper, we use the POS-tag together with the morphological analysis results to specify the POS information of each morpheme. After applying BPE, '로마/NNG' (Rome/NNG) and '바카스/NNP' (Bacchus/NNP) were divided into '로' (Ro), '마/NNG_' (me/NNG_) and '바' (Ba), '카스/NNP_' (cchus/NNP_) according to BPE dictionary matching.

### 3.1.2 Additional Input Features

In this study, We use five features for Korean coreference resolution, which are word boundary, morpheme (morp) boundary, dependency parsing, NER, and head distance. The description of each feature is as follows:

**Word boundary**: This studies the boundary feature of the coreference resolution in word units. The starting token of the word is divided into *B*, and the following token is divided into *I* tags.

**Morpheme boundary**: This reflects the morpheme boundary characteristics of the morpheme analysis results. *Morp-B* is the beginning token, and *morp-I* is the inside token of the morpheme.

**Dependency parsing**: We use the dependency parsing label as a feature to reflect the structural and semantic information of the sentences.

**NER**: We use type information for each entity appearing in the document as a feature.

**Head distance**: To use distance information between extracted candidate nouns and we measure the distance from the immediately preceding noun, the following buckets [1, 2, 3, 4, 5-7, 8-15, 16-31, 32-63, 64+] (Clark and Manning, 2016b).

## 3.2 Model Architecture

### 3.2.1 Encoder

As shown in Figure 2, each token input to the encoder gets the hidden state of BERT $\mathbf{b}_i = \mathrm{BERT}(x_i)$ from the pre-trained BERT model. The hidden state for the input feature is generated as follows: $\mathbf{h}_i^f = \mathrm{emb}_{feat}(f_i)$. We concatenate the hidden state of the BERT and the hidden state of the features to make the hidden state $\mathbf{e}_i = [\mathbf{b}_i; \mathbf{h}_i^f]$.

Then, according to equation 1, the encoder encodes $\mathbf{e}_i$ into bidirectional SRU (biSRU) to generate a hidden state $\mathbf{r}_i$.

$$\mathbf{r}_i = \text{biSRU}(\mathbf{r}_{i-1}, \mathbf{e}_i) \tag{1}$$

### 3.2.2 Decoder

In Figure 2, the input of the decoder is $\mathbf{r}_t^h = \text{copy}(\mathbf{r}_{yi_t})$ that extracts the hidden state corresponding to the head $yi_t$ from the encoded hidden state $\mathbf{r}_i$. The decoder performs biSRU(.), as shown in equation 2, to model the context information between heads.

$$\mathbf{h}_t^h = \text{biSRU}(\mathbf{h}_{t-1}^h, \mathbf{r}_t^h) \tag{2}$$

**Self-attention Module** To model the scores between similar head, we apply a gated self-matching layer (Wang et al., 2017), the equation follows as:

$$\begin{aligned}
\mathbf{h}_t &= \text{biSRU}(\mathbf{h}_{t-1}, \mathbf{g}_t) \\
\mathbf{g}_t &= \text{sigmoid}(\mathbf{W}_g[\mathbf{h}_t^h; \mathbf{c}_t]) \odot [\mathbf{h}_t^h; \mathbf{c}_t] \\
\mathbf{c}_t &= \sum_{j=1}^{m} \alpha_{t,k} \mathbf{h}_t^h \qquad (3) \\
\alpha_{t,k} &= \exp(\mathbf{h}_k^h \mathbf{W}_\alpha \mathbf{h}_{t'}^h) / \sum_j \exp(\mathbf{h}_k^h \mathbf{W}_\alpha \mathbf{h}_t^h)
\end{aligned}$$

Where $\mathbf{c}_t$ is the context vector of the whole heads. $\mathbf{g}_t$ is a hidden state generated from the additional gate. The additional gate concatenates the hidden state $\mathbf{h}_t^h$ and the context vector $\mathbf{c}_t$, and applies a sigmoid gate to convert the significant value of the two vectors to larger ones, and otherwise to smaller ones. BiSRU(.) models $\mathbf{g}_t$ with gate applied and generated $\mathbf{h}_t$.

**Deep Biaffine Score** To output the mention start boundary and coreference resolution, we apply elu (Clevert et al., 2015) to the last hidden state $\mathbf{h}_t$ of the decoder as shown in Dozat and Manning (2016), and create the hidden states as $\mathbf{h}_t^{men\_src}$, $\mathbf{h}_t^{coref\_src}$, $\mathbf{h}_t^{coref\_tgt}$. In this case, the hidden state to be used for the output of the mention boundary is $\mathbf{h}_i^{men\_tgt}$ based on the output hidden state $\mathbf{r}_i$ of the encoder.

$$\begin{aligned}
\mathbf{h}_t^{men\_src} &= \text{elu}(\text{FFNN}^{(men\_src)}(\mathbf{h}_t)) \\
\mathbf{h}_i^{men\_tgt} &= \text{elu}(\text{FFNN}^{(men\_tgt)}(\mathbf{r}_i)) \\
\mathbf{h}_t^{coref\_src} &= \text{elu}(\text{FFNN}^{(coref\_src)}(\mathbf{h}_t)) \\
\mathbf{h}_t^{coref\_tgt} &= \text{elu}(\text{FFNN}^{(coref\_tgt)}(\mathbf{h}_t))
\end{aligned} \tag{4}$$

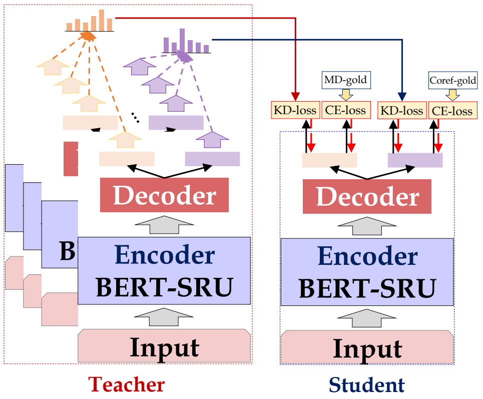

Figure 3: Ensemble knowledge distillation for BERT-SRU Pointer Networks.

We apply the deep biaffine score when performing the attention to output the mention boundary and the coreference resolution, and the equation follows as:

$$\begin{aligned}
s_{t,i}^{men} &= \mathbf{h}_i^{\top men\_tgt} \mathbf{U} \mathbf{h}_t^{men\_src} + \mathbf{w}^\top \mathbf{h}_t^{men\_src} \\
s_{t,t}^{coref} &= \mathbf{h}_t^{\top coref\_tgt} \mathbf{U} \mathbf{h}_t^{coref\_src} + \mathbf{w}^\top \mathbf{h}_t^{coref\_src}
\end{aligned} \tag{5}$$

### 3.3 Model Extension: Ensemble Knowledge Distillation

An ensemble is a model that combines the output results of several single models into a single result. When performing the ensemble, we use the method of averaging all the softmax probability distributions of single models. Meanwhile, knowledge distillation is a model compression technique to reduce the size of a single model (Hinton et al., 2015). A small student model is trained to learn the output distribution of a large teacher model using a loss function that compares the distribution, such as Kullback Leibler distance (KLD, Kullback and Leibler (1951)). As shown in Figure 3, we use the ensemble model as the distribution of the teacher model and we distill its knowledge to a single model as the student model. The loss function that we use to train the knowledge distillation is KLD, and the final loss function equation is as follows:

$$\mathcal{L}_{kd} = \sum p_T(y|x) \log (\frac{p_T(y|x)}{p_S(y|x)}) \tag{6}$$

$$\begin{aligned}
\mathcal{L} &= \alpha \mathcal{L}_{ce}^{coref} + (1-\alpha)\mathcal{L}_{ce}^{men} \\
&+ \beta(\gamma \mathcal{L}_{kd}^{coref} + (1-\gamma)\mathcal{L}_{kd}^{men})
\end{aligned} \tag{7}$$

| | Training | Dev | Test |
|---|---|---|---|
| #Document | 2,819 | 645 | 571 |
| #Sentence | 8,299 | 1,086 | 1,167 |
| #Word | 126,720 | 12,834 | 14,334 |
| #Morpheme | 295,076 | 30,396 | 34,657 |
| #Mention | 30,923 | 1,978 | 2,431 |
| #Entity | 10,416 | 799 | 931 |

Table 1: Dataset statistics of ETRI dataset for Korean coreference resolution

In equation 6, $p_T(y|x)$ is the result distribution of the teacher model, and $p_S(y|x)$ is the result distribution of the student model. Equation 7 calculates the final loss by adding the cross-entropy loss (Nasr et al., 2002) and the knowledge distillation loss of the mention detection and coreference solution. Cross-entropy losses for mention and coreference resolution are $\mathcal{L}_{ce}^{men}$ and $\mathcal{L}_{ce}^{coref}$, and knowledge distillation losses are $\mathcal{L}_{kd}^{men}$ and $\mathcal{L}_{kd}^{coref}$, respectively. Here, the weight $\alpha$ is a hyper-parameter that determines the loss reflection ratio between coreference resolution and mention boundary. $\beta$ is the weight of knowledge distillation loss. $\gamma$ determines the loss reflection ratio between coreference resolution and mention boundary in the teacher model. $\alpha$, $\beta$ and $\gamma$ all perform optimization and the values are 0.9, 0.2 and 0.9, respectively.

## 4 Experiments

**Dataset and Measures** We use the Korean coreference resolution data (Park et al., 2016) from the ETRI quiz domain of AIOpen[1]. Table 1 summarizes the dataset statistics. We use CoNLL F1 averaged MUC, B$^3$, and CEAF$_{\phi 4}$ according to the official CoNLL-2012 evaluation script. However, we evaluate coreference resolution using only heads of mentions as it is more suitable for Korean coreference resolution because the positional weighting in the script is tailored for English.

**Pre-training Korean BERT** BERT consists of a bidirectional transformer encoder with several layers. For pre-training BERT, we reuse the hyperparameters from Devlin et al. (2019). We used Wikipedia and news data (total 23.5 GB) collected from the web. After performing morpheme analysis on all input words, tokenization was done on the subwords using BPE. The dictionary consists of 30,349 BPE tokens. We used the *ETRI language analyzer* for morpheme analysis which is also available in AIOpen as a tool for Korean NLP.

**Implementation** Hyper-parameters of the BERT-SRU-based Pointer Networks model are as follows. We fine-tune all models on the ETRI Korean data for 70 epochs with a batch size of six for each GPU. The model trained on 2 GEFORCE GTX 1080 Ti GPU cards. The number of hidden layer dimensions and feature dimensions of the SRU was optimized to 800 and 1,600, respectively. We have optimized the stack of the SRU hidden layer to two. We set the dropout as 0.1. The training algorithm we used is Adam (Kingma and Ba, 2014), and Adam weight decay was set to $1 \times 10^{-2}$. The learning rate was set to $5 \times 10^{-5}$, and the linear method was used in the learning rate schedule. The maximum length of the input sequence was limited to 430 because the most extended input sequence length in the test set was 428. We used the ETRI language analyzer to obtain POS-tagging, NER, and dependency parsing features.

**Head Candidates** In general, pointer networks' target outputs align with those of the decoder inputs. For our head-final coreference resolution, we set the inputs of the decoder as the list of head candidates. These head candidates are all nouns of the source document and they is extracted using the POS-tags. By doing so, we can effectively reduce the computational complexity to $O(n^2)$, as the search for coreference links is only done between the head candidates.

**Attention Masking** In coreference resolution, the antecedent at position $i$ comes before the head at $j$, where $i <= j$. Similarly, in mention detection, the beginning boundary of a mention always appears before the head. Accordingly, when calculating the attention score, we perform attention masking to prevent attention from being calculated for the element that is later than the $j$-th position.

## 5 Results

In this section, we show our experimental results for Korean coreference resolution. We denote the **BERT-SRU-based ptr-net** as our model for head-based coreference resolution. The performance of the models is measured and compared using the CoNLL Average F1-score.

[1] http://aiopen.etri.re.kr/

| Model | Word Embedding | CoNLL Avg. F1 | Doc/ sec | Time complexity |
|---|---|---|---|---|
| e2e-coref (Lee et al., 2017) | NNLM | 59.4 | 24 | $O(n^4)$ |
| c2f-coref (Lee et al., 2018) | ELMo | 60.2 | 23 | $O(n^4)$ |
| BERT-coref (Joshi et al., 2019b) | BERT | **67.0** | 15 | $O(n^4)$ |
| BERT-SRU ptr-net (Google) | BERT | 63.5 | 28 | $O(n^2)$ |
| BERT-SRU ptr-net (single) | BERT | 66.2 | **30** | $O(n^2)$ |
| BERT-SRU ptr-net (KD) | BERT | 66.9 | **30** | $O(n^2)$ |
| BERT-SRU ptr-net (ensemble) | BERT | 68.6 | - | $O(n^2)$ |

Table 2: Experimental results on the test set of the Korean data from ETRI wiseQA. The first column shows which word embedding method is used. The CoNLL Avg. F1 is the main evaluation metric that is averaged by the F1 of MUC, $B^3$, and CEAF$_{\phi_4}$ (Full results are in the Appendix). Based on the head-final trait of Korean, the coreference resolution score is calculated based on the head candidates. In the second column, we use NNLM and ELMo pre-trained in Korean (Lee et al., 2014; Park et al., 2019b). The third column (Doc/sec) is the number of documents processing per second. The final column shows a time complexity for each model.

### 5.1 Coreference Resolution

Table 2 compares our model with several previous systems for the Korean coreference resolution. We calculate the averaged F1 score of MUC, $B^3$, CEAF$_{\phi_4}$, according to the official CoNLL−2012 evaluation scripts. We evaluate performance using only the head, the last word of mention. Our main baselines are the span-ranking models from (Lee et al., 2017, 2018; Joshi et al., 2019b) Korean word vector representation, and they are denoted as `e2e-coref`, `c2f-coref`, `BERT-coref`, respectively. We extend the original Tensorflow implementations of `e2e-coref` , `c2f-coref`[2] and `BERT-coref`[3] for Korean coreference resolution.

The `e2e-coref` shows average F1 of 59.4 and `c2f-coref` from (Lee et al., 2018) uses *second-order* span representations achieves a slightly higher performance of 60.2 F1 for head-based Korean coreference resolution. Our proposed model achieves 66.2 of CoNLL F1, which is 6.8 and 6.0 points higher than `e2e-coref` and `c2f-coref`, respectively. However, this improvement is most likely due to the usage of BERT because `BERT-coref` also shows a significantly higher performance (67.0 F1) than the other two baselines, and its main difference with `c2f-coref` is the usage of BERT.

Meanwhile, by ensembling 10 models, we achieve state-of-the-art performance in this Korean dataset with F1 of 68.6, which is 2.4 points higher than our single model and 1.6 points more than `BERT-coref`. However, as ensembling models is notoriously expensive in terms of inference time and memory usage, we also provide a knowledge distilled model of the ensemble that solves this problem which is referred to as BERT-SRU ptr-net (KD). This distilled model has the same size as the single model while having 0.7 points higher in F1, and only 0.1 point difference with the best single model, `BERT-coref`. It is noteworthy that not only our ensemble KD model can achieve similar performance to `BERT-coref` without using any *higher-order* modeling, it also has a 2x faster document processing speed (30 vs 15 doc/sec) due to the much smaller computational complexity.

We also compare the usage of different pre-trained BERT embeddings. Table 2 shows that our pre-trained version is more suitable for this task than Google's multilingual BERT[4] (BERT-SRU ptr-net (Google)).

### 5.2 Ensemble Knowledge Distillation

**Ensemble** We perform an ensemble using ten single models with different random seeds on the dev set. The lowest performance among the 10-models is 70.04% F1, and the average F1 score is 70.37% and Std. deviation is 0.253, both of which still outperforms the 68.62 F1 of the Korean `BERT-coref` from Joshi et al. (2019b). We perform a maximum score ensemble and an average score of the ensemble for 10-models. The maximum score ensemble is 72.26% F1, and the average score of the ensemble is 72.23% F1. But we choose the average score of the ensemble because the average ensemble is 1.28% higher than the maximum score ensemble in the test set.

**Knowledge Distillation** We optimize the weight option $\beta$ of knowledge distillation, such that we

[2] https://github.com/kentonl/e2e-coref

[3] https://github.com/mandarjoshi90/coref

[4] https://github.com/google-research/bert

| Feature | Avg. F1 | Δ |
|---|---|---|
| BERT-SRU ptr-net (single) | 70.83 | – |
| – morp boundary | 70.03 | −0.80 |
| – dependency parsing | 69.71 | −1.12 |
| – NER | 69.63 | −1.20 |
| – head distance | 69.56 | −1.27 |
| – word boundary | 69.23 | −1.60 |

Table 3: Feature ablation study of Korean coreference resolution on dev set.

| Component | Avg. F1 | Δ |
|---|---|---|
| BERT-SRU ptr-net (single) | 70.83 | – |
| – attention masking | 70.17 | −0.66 |
| – head target class | 68.44 | −2.39 |
| – mention detection module | 70.09 | −0.74 |
| – self-attention module | 69.89 | −0.94 |

Table 4: Component ablation study on the dev set.

apply $\beta$ only to knowledge distillation loss term as $\mathcal{L} = \mathcal{L}_{ce} + \beta\mathcal{L}_{kd}$ in equation 7. The optimized $\beta$ is 0.2, and it is meaningful to apply the loss to Korean coreference resolution.

## 6 Analysis

**Feature ablation study** We perform feature ablation to understand the effect of each feature on Korean coreference resolution. Table 3 compares the ablation performance of each feature. Removing the morp boundary deteriorates the average F1 score by 0.8%. Also, dependency parsing or NER feature decreases 1.12, 1.20 F1 score, respectively. If the head distance feature is removed, the F1 score is reduced by 1.27%. Among all the features, the word boundary has the most significant difference from the other features.

**Component ablation study** To understand the effect of different components on the model, we perform components ablation study on the dev set, as illustrated in Table 4. We apply attention masking to consider only true antecedents when calculating the attention score in the decoder of the pointer networks and define the head candidate list (nouns) as the target class to reduce candidates of the target class. Removing this attention mask decreases the average F1 score by 0.66 points. When we define the target class as the entire input document, it deteriorates the F1 score significantly by 2.39 points. These two methods combined make the most contribution to our model.

In addition, we share a hidden layer to perform coreference resolution and detection of mention start boundary together. When mention detection module is removed, the F1 score is reduced by 0.74. Finally, removing the self-attention module of the decoder results in a difference of 0.94 F1. Accordingly, it can be seen that all components of the proposed model are contribute meaningfully to the Korean coreference resolution task.

**Qualitative Analysis** Our qualitative analysis in Figure 4 highlights the strengths of our model. Figure 4 shows examples first in Korean and then its English translated version. In Example 1, we can see that the removal of the mention detection (*w/o MD*) module from our model does not properly link the entity to 레오나르도 다빈치 (*Leonardo da Vinci*). When training using BERT embedding without fine-tuned Korean BERT, it does not find 엘리자베타 (*Elisabeta*) as an entity to resolve. On the other hand, our model distinguishes various entity information and performs coreference resolution correctly on all entities. From example 2, our model even finds 물체 *(object)* entity links missing from the ground truth, demonstrating the robustness of our model.

Meanwhile, pronouns and determiner phrases are the most substantial part of coreference resolution. In example 3, our model can successfully predict that the pronouns and the determiner phrases such as 이 사자성어 (*This idiom*), 이 말 (*this*), 무엇 (*What*) are linked to an entity as 어려운 기회 (*challenging opportunity*). Furthermore, in Korean documents, foreign languages such as Chinese characters and English frequently appear. Our model reflects the contextual information and can successfully perform coreference to foreign languages. In Example 4, Persian token exists in the vocabulary of BERT and the model can successfully resolve the coreference between the two foreign words. In addition, the model can also detect relatively long and complex noun phrases, such as 낙타나 말 등에 짐을 싣고 떼지어 다니면서 특산물을 파고 사는 상인의 집단 (*a group of merchants carrying loads of troops on camels and horses and selling specialties*).

**Weaknesses and Future Works** As shown from the results, head-final coreference resolution, which reflects the linguistic characteristics of Korean, has a significant computational advantage over span-based coreference resolution. However, our method

| | | |
|---|---|---|
| 1 | Truth | [[이탈리아의 화가]$^{0}$ 레오나르도 다빈치]$^{0}$가 [[[[피렌체의 부호]$^{1}$ 프란체스코 데 조콘다]$^{1}$의 부인]$^{2}$ 엘리자베타]$^{2}$를 그린 초상화.<br>A portrait of [[Italian painter]$^{0}$ Leonardo da Vinci]$^{0}$ depicting [Elisabeta, [wife of [[Florence's rich]$^{1}$ Francesco de Joconda]$^{1}$]$^{2}$]$^{2}$. |
| | Ours | [[이탈리아의 화가]$^{0}$ 레오나르도 다빈치]$^{0}$가 [[[[피렌체의 부호]$^{1}$ 프란체스코 데 조콘다]$^{1}$의 부인]$^{2}$ 엘리자베타]$^{2}$를 그린 초상화.<br>A portrait of [[Italian painter]$^{0}$ Leonardo da Vinci]$^{0}$ depicting [Elisabeta, [wife of [[Florence's rich]$^{1}$ Francesco de Joconda]$^{1}$]$^{2}$]$^{2}$. |
| | w/o MD | 이탈리아의 화가 레오나르도 다빈치가 피렌체의 [부호]$^{1}$ 프란체스코 데 [조콘다]$^{1}$의 [부인]$^{2}$ [엘리자베타]$^{2}$를 그린 초상화.<br>A portrait of Italian painter Leonardo da Vinci depicting [Elisabeta, [wife of [[Florence's rich]$^{1}$ Francesco de Joconda]$^{1}$]$^{2}$]$^{2}$. |
| | BERT emb. | [[이탈리아의 화가]$^{0}$ 레오나르도 다빈치]$^{0}$가 [[피렌체의 부호]$^{1}$ 프란체스코 데 조콘다]$^{1}$의 부인 엘리자베타를 그린 초상화.<br>A portrait of [[Italian painter]$^{0}$ Leonardo da Vinci]$^{0}$ depicting Elisabeta, wife of [[Florence's rich]$^{1}$ Francesco de Joconda]$^{1}$. |
| 2 | Truth | [도플러효과를 이용한 기구]$^{0}$인 [이것]$^{0}$은 움직이는 물체에 초음파 등을 쏘아 물체의 속도를 측정한다.<br>[An instrument using the Doppler effect]$^{0}$, [it]$^{0}$ measures the speed of an object by shooting an ultrasonic wave on a moving object. |
| | Ours | [도플러효과를 이용한 기구]$^{0}$인 [이것]$^{0}$은 [움직이는 물체]$^{1}$에 초음파 등을 쏘아 [물체]$^{1}$의 속도를 측정한다.<br>[An instrument using the Doppler effect]$^{0}$, [it]$^{0}$ measures the speed of [an object]$^{1}$ by shooting an ultrasonic wave on [a moving object]$^{1}$. |
| 3 | Ours | ['좀처럼 만나기 어려운 기회']$^{0}$를 뜻하는 [이 사자성어]$^{0}$는 중국 동진 시대의 학자인 원굉이 '현명한 군주와 지모가 뛰어난 신하가 만나는 기회는 천년에 한번쯤이다'라고 한 데서 유래했다. [이 말]$^{0}$은 [무엇]$^{0}$일까?<br>[This idiom]$^{0}$, which means [‘challenging opportunity,']$^{0}$ comes from Won Auk, a scholar from the East China era, who said, "A chance to meet a wise monarch and a brilliant servant is once every millennium." [What]$^{0}$ does [this]$^{0}$ mean? |
| 4 | Ours | [대상(隊商)]$^{0}$은 [낙타나 말 등에 짐을 싣고 떼지어 다니면서 특산물을 팔고 사는 상인의 집단]$^{0}$을 뜻하며 [[캐러밴(영어: caravan)]$^{0}$ 또는 카라반(페르시아어: کاروان)]$^{0}$이라고도 부른다.<br>[A caravan (隊商)]$^{0}$ is [a group of merchants carrying loads of troops on camels and horses and selling specialties]$^{0}$, also called [caravans]$^{0}$ or [caravans (کاروان)]$^{0}$. |

Figure 4: Qualitative Analysis: Examples of predictions from the development data. Example 1 and 2 describe the coreference entities predicted in our model. Each row of examples 3 to 4 depicts a single coreference entity predicted by our model. Square brackets refer to mentions, and underline refers to the head. The superscript in the mention is the entity number.

| Document length | # Docs | Avg. F1 |
|---|---|---|
| 0-32 | 175 | 71.98 |
| 33-64 | 248 | 75.48 |
| 65-96 | 151 | 72.91 |
| 97-128 | 49 | 66.96 |
| 128+ | 22 | 52.32 |

Table 5: Performance on the Korean ETRI dev set generally drops as the document length increases.

can only be applied to languages that are either *strongly head-initial* (head is at the beginning of mention) or *strongly head-final*, and English is a mixture of those two. In future works, a search for *English linguistic traits* could alleviate the computational complexity issue of this task in English or other mixed head-directional languages.

Furthermore, as shown in Table 5, it is clear that the coreference resolution performance significantly decreases when the document length increases. Although this is partly due to the Korean dataset being relatively small and non-uniform regarding document length, we believe the choice of BERT size is also relevant. Recent studies have shown that larger BERT might better encode longer contexts (Joshi et al., 2019a). By using the `BERT-large` model (we use `BERT-base`) in Joshi et al. (2019b), coreference resolution improves overall performance, *especially for long documents*. In future works, we would like to explore BERT variants that are good at larger contexts.

## 7 Conclusion

We propose head-based coreference resolution that reflects the head-final characteristics of Korean and present a suitable BERT-SRU-based Pointer Networks model that leverages this linguistic trait. The proposed method, as the first end-to-end Korean coreference resolution model, not only achieves state-of-the-art performance in the Korean coreference resolution model through ensembling but also dramatically speeds up the document processing time compared to the conventional span-based coreference resolution. Our method achieves this result by reducing the problem of coreference resolution from a *search over span candidates* to a *search over head candidates* using the fact that we can easily extract the mention heads for a head-final language.

Moreover, our proposed method of using head-directionality to speed up coreference resolution while maintaining the best performance is valid for not only other strongly *head-final* languages like Japanese, but also for strongly *head-initial* languages as the same method of head extraction can be applied. We believe that our paper also provides an interesting and important research direction. Combining linguistic theories like head-directionality and branching with deep learning has a strong potential of more efficiently and effectively model fundamental tasks like coreference resolution.

## Acknowledgments

This work was supported by Institute for Information & Communications Technology Promotion (IITP) grants funded by the Korean government (MSIT) (2013-0-00131, Development of Knowledge Evolutionary WiseQA Platform Technology for Human Knowledge Augmented Services; the Korea Electric Power Corporation (Grant number:R18XA05)).

## References

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar. Association for Computational Linguistics.

Kevin Clark and Christopher D. Manning. 2016a. Deep reinforcement learning for mention-ranking coreference models. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2256–2262, Austin, Texas. Association for Computational Linguistics.

Kevin Clark and Christopher D. Manning. 2016b. Improving coreference resolution by learning entity-level distributed representations. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 643–653, Berlin, Germany. Association for Computational Linguistics.

Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. 2015. Fast and accurate deep network learning by exponential linear units (elus). *arXiv preprint arXiv:1511.07289*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Timothy Dozat and Christopher D Manning. 2016. Deep biaffine attention for neural dependency parsing. *arXiv preprint arXiv:1611.01734*.

Matthew S Dryer. 2009. The branching direction theory of word order correlations revisited. In *Universals of language today*, pages 185–207. Springer.

Aria Haghighi and Dan Klein. 2010. Coreference resolution in a modular, entity-centered model. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 385–393, Los Angeles, California. Association for Computational Linguistics.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network.

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

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2019a. Spanbert: Improving pre-training by representing and predicting spans. *CoRR*, abs/1907.10529.

Mandar Joshi, Omer Levy, Luke Zettlemoyer, and Daniel Weld. 2019b. BERT for coreference resolution: Baselines and analysis. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5802–5807, Hong Kong, China. Association for Computational Linguistics.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

S. Kullback and R. A. Leibler. 1951. On information and sufficiency. *Ann. Math. Statist.*, 22(1):79–86.

Nayoung Kwon, Maria Polinsky, and Robert Kluender. 2006. Subject preference in korean. In *Proceedings of the 25th west coast conference on formal linguistics*, pages 1–14. Cascadilla Proceedings Project Somerville, MA.

Changki Lee, Junseok Kim, and Jeonghee Kim. 2014. Korean dependency parsing using deep learning. In *Proceedings of the 26th Annual Conference on Human Cognitive Language Technology*, pages 87–91. SIGHCLT: Special Interest Group of Human and Cognitive Language Technology.

Heeyoung Lee, Angel Chang, Yves Peirsman, Nathanael Chambers, Mihai Surdeanu, and Dan Jurafsky. 2013. Deterministic coreference resolution based on entity-centric, precision-ranked rules. *Computational Linguistics*, 39(4):885–916.

Kenton Lee, Luheng He, Mike Lewis, and Luke Zettlemoyer. 2017. End-to-end neural coreference resolution. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 188–197, Copenhagen, Denmark. Association for Computational Linguistics.

Kenton Lee, Luheng He, and Luke Zettlemoyer. 2018. Higher-order coreference resolution with coarse-to-fine inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Asso-*

*ciation for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 687–692, New Orleans, Louisiana. Association for Computational Linguistics.

Tao Lei, Yu Zhang, and Yoav Artzi. 2017. Training rnns as fast as cnns. *CoRR*, abs/1709.02755.

G. E. Nasr, E. A. Badr, and C. Joun. 2002. Cross entropy error function in neural networks: Forecasting gasoline demand. In *Proceedings of the Fifteenth International Florida Artificial Intelligence Research Society Conference*, pages 381–384. AAAI Press.

Vincent Ng. 2010. Supervised noun phrase coreference research: The first fifteen years. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1396–1411, Uppsala, Sweden. Association for Computational Linguistics.

Vincent Ng and Claire Cardie. 2002. Identifying anaphoric and non-anaphoric noun phrases to improve coreference resolution. In *COLING 2002: The 19th International Conference on Computational Linguistics*.

Cheoneum Park, Kyoung-Ho Choi, Changki Lee, and Soojong Lim. 2016. Korean coreference resolution with guided mention pair model using deep learning. *ETRI Journal*, 38(6):1207–1217.

Cheoneum Park, Juae Kim, Hyeon-gu Lee, Reinald Kim Amplayo, Harksoo Kim, Jungyun Seo, and Changki Lee. 2019a. ThisIsCompetition at SemEval-2019 task 9: BERT is unstable for out-of-domain samples. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 1254–1261, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Cheoneum Park, Dongheon Lee, Kihoon Kim, Changki Lee, and Hyunki Kim. 2019b. Korean movie review sentiment analysis using self-attention and contextualized embedding. *Journal of KIISE*, 46(9):901–908.

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Olga Uryupina, and Yuchen Zhang. 2012. CoNLL-2012 shared task: Modeling multilingual unrestricted coreference in OntoNotes. In *Joint Conference on EMNLP and CoNLL - Shared Task*, pages 1–40, Jeju Island, Korea. Association for Computational Linguistics.

Karthik Raghunathan, Heeyoung Lee, Sudarshan Rangarajan, Nathanael Chambers, Mihai Surdeanu, Dan Jurafsky, and Christopher Manning. 2010. A multi-pass sieve for coreference resolution. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 492–501, Cambridge, MA. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

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

Rupesh Kumar Srivastava, Klaus Greff, and Jürgen Schmidhuber. 2015. Highway networks. *arXiv preprint arXiv:1505.00387*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer networks. In *Advances in Neural Information Processing Systems*, pages 2692–2700.

Wenhui Wang, Nan Yang, Furu Wei, Baobao Chang, and Ming Zhou. 2017. Gated self-matching networks for reading comprehension and question answering. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 189–198.

Yuxuan Wang, Wanxiang Che, Jiang Guo, Yijia Liu, and Ting Liu. 2019. Cross-lingual BERT transformation for zero-shot dependency parsing. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5720–5726, Hong Kong, China. Association for Computational Linguistics.

Kellie Webster, Marta Recasens, Vera Axelrod, and Jason Baldridge. 2018. Mind the gap: A balanced corpus of gendered ambiguous pronouns. *Transactions of the Association for Computational Linguistics*, 6:605–617.

Sam Wiseman, Alexander M. Rush, Stuart Shieber, and Jason Weston. 2015. Learning anaphoricity and antecedent ranking features for coreference resolution. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1416–1426, Beijing, China. Association for Computational Linguistics.

Sam Wiseman, Alexander M. Rush, and Stuart M. Shieber. 2016. Learning global features for coreference resolution. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 994–1004, San Diego, California. Association for Computational Linguistics.

Meishan Zhang, Zhenghua Li, Guohong Fu, and Min Zhang. 2019. Syntax-enhanced neural machine translation with syntax-aware word representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1151–1161, Minneapolis, Minnesota. Association for Computational Linguistics.

Rui Zhang, Cicero Nogueira dos Santos, Michihiro Yasunaga, Bing Xiang, and Dragomir Radev. 2018. Neural coreference resolution with deep biaffine attention by joint mention detection and mention clustering. *arXiv preprint arXiv:1805.04893*.

# A Appendices

## A.1 Related Work

Traditional coreference resolution studies are divided into rule- and machine learning-based methods. In the rule-based method, Stanford's model (Lee et al., 2013), applied to multi-pass sieve using pronouns, entity attributes, named entity information, and so on. In the statistics-based, various coreference models have been proposed such as mention-pair (Ng and Cardie, 2002; Ng, 2010), mention-ranking (Wiseman et al., 2015; Clark and Manning, 2016a) and entity-level models (Haghighi and Klein, 2010; Clark and Manning, 2016b).

Lee et al. (2017) defined mentions as span representations and proposed a span ranking model based on long short-term memory (LSTM, (Hochreiter and Schmidhuber, 1997)) for all spans in the document. As span representations could reflect the contextual information from LSTM, but the other two spans are interpreted as a related entity. This phenomenon results in local consistency errors that yield erroneous coreference resolutions. Hence, Lee et al. (2018) performed the attention mechanism to resolve coreference using a high-order function. The end-to-end model of (Lee et al., 2017, 2018) showed the superior performance in English coreference resolution however, the complexity of $O(n^4)$ is considering all spans and span pairs of the document. Zhang et al. (2018) is based on the (Lee et al., 2017), which replaced the concat attention score into the biaffine attention score to calculate the conference score. Also, it performed the multi-task learning process that also calculates the loss for the mention score.

Simple recurrent units (SRU) (Lei et al., 2017) architecture solves the vanishing gradient problem that occurs when back-propagation of the recurrent neural network (RNN). SRU, which is one of RNN types such as gated recurrent unit architecture (GRU) (Cho et al., 2014) and LSTM, is less computational complexity than other RNN types because the SRU encodes hidden states using a feed-forward neural gate and recurrent cell in a layer.

Recently, a variety of downstream studies using BERT (Bidirectional Encoder Representations from Transformer, Vaswani et al. (2017); Devlin et al. (2019)) which have been pre-trained with large amounts of data, have been conducted in natural language processing tasks (Joshi et al., 2019b; Zhang et al., 2019; Park et al., 2019a; Wang et al., 2019). A BERT-coref study was also conducted in the English coreference resolution task, and a more effective SpanBERT (Joshi et al., 2019a) for coreference resolution has also been studied, with dramatic gains in GAP (Webster et al., 2018) and OntoNotes (Pradhan et al., 2012) datasets. A qualitative assessment of BERT-coref showed that BERT is significantly better at distinguishing unique entities and concepts.

## A.2 Data Format for Our Model

The following example shows input sequence, head list and decoder output format.

- **Input sequence for BERT**: "[CLS] 그리스/NNP_ 로 마/NNG_ 신화/NNG_ 에서/JKB_ 바 카스/NNP_ 이/VCP_ 라고/EC_ 도/JX_ 불리/VV_ 는/ETM_ 술/NNG_ 의/JKG_ 신/NNG_ [SEP]"
- **Heads**: "그리스/NNP, 로마/NNG, 신화/NNG, 바카스/NNP, 술/NNG, 신/NNG"
- **Heads applied by BPE**: "그리스/NNP_, 로마/NNG_, 신화/NNG_, 바, 술/NNG_, 신/NNG_"
- **Head list**: [0, 1, 2, 3, 5, 12, 14]
- **Decoder output**: [0, 0, 0, 0, 0, 5, 5]

We add [CLS] and [SEP] to match the input sequence to the BERT format. The Heads is an example of heads included in a sentence, and the Heads applied by BPE is an example of heads with BPE applied. BPE divides words into subwords. The head divided into subwords uses the first token as the representative of the head. In the example of the Heads applied by BPE, the representative of the BPE-applied head '바' (Ba) and '카스/NNP' (cchus/NNP) is '바' (Ba). The Head list is the position of the head in the sentence that matches the BERT input format, which is input to the decoder. The head list is a target class. The decoder output is a position where the coreference resolves in the head list. Since '바' (Ba) is first mention in the entity of Bacchus, '바' (Ba) outputs its own location of 5. '신/NNG' (a god) outputs position 5 because it is linked to '바' (Ba). We then change the output to word units via post-processing.

## A.3 Overall Performance

Please refer to Table 6 for full performance on all metrics, and dev set results for Table 7.

|  | MUC |  |  | $B^3$ |  |  | CEAF$_{\phi_4}$ |  |  | CoNLL |
|---|---|---|---|---|---|---|---|---|---|---|
| Model | Prec. | Rec. | F1 | Prec. | Rec. | F1 | Prec. | Rec. | F1 | Avg. F1 |
| e2e-coref (Lee et al., 2017) | 66.9 | 55.2 | 60.5 | 64.5 | 53.1 | 58.2 | 66.1 | 53.9 | 59.4 | 59.4 |
| c2f-coref (Lee et al., 2018) | 68.3 | 56.4 | 61.8 | 59.0 | 53.4 | 59.0 | 66.4 | 54.4 | 59.8 | 60.2 |
| BERT-coref (Joshi et al., 2019b) | 71.7 | 65.0 | 68.2 | 69.3 | 63.0 | 66.0 | 72.2 | 62.4 | 66.9 | 67.0 |
| BERT-SRU enc-dec (Google) | 67.7 | 61.9 | 64.6 | 65.7 | 59.8 | 62.6 | 68.5 | 58.8 | 63.3 | 63.5 |
| BERT-SRU enc-dec (single) | 67.3 | 67.3 | 67.3 | 64.8 | 65.3 | 65.1 | 69.5 | **63.5** | 66.3 | 66.2 |
| BERT-SRU enc-dec (ensemble) | **72.3** | 67.6 | **69.9** | **70.0** | 65.2 | **67.5** | **75.0** | 63.0 | **68.5** | **68.6** |
| BERT-SRU enc-dec (KD) | 68.0 | **68.2** | 68.1 | 65.6 | **66.0** | 65.8 | 71.1 | 62.9 | 66.7 | 66.9 |

Table 6: Experimental results on the test set of the Korean data from ETRI wiseQA. The final column (CoNLL Avg. F1) is the main evaluation metric, averaged by the F1 of MUC, $B^3$, and CEAF$_{\phi_4}$. Based on the Korean head-final, the coreference resolution score is calculated based on the head of the mentions.

| **Model** | **Avg. F1** | **Δ** |
|---|---|---|
| BERT-SRU ptr-net (single) | 70.83 | - |
| − fine-tuning | 64.74 | −6.09 |
| BERT-SRU ptr-net (Google) | 67.38 | −3.45 |
| `BERT-coref` | 68.62 | −2.21 |

Table 7: Dev set results. We evaluate the performance of models using different BERT.

## A.4 Optimizing Hyperparameters

We perform hyperparameter optimization on the baseline model of the BERT-SRU Pointer Networks, which is not applied to the head target class component. We optimize hyperparameters for the development set, and hyperparameter optimization proceeds for the feature embedding size, the number of RNN hidden layer dimensions, and the number of biaffine hidden layer dimensions. We set the number of dimensions to 50, 100, 200, 400, 800, 1600, respectively, to find the hyperparameters that give the best performance.

**Optimizing Dimension Size of Feature Embedding** In Table 8, we perform an optimization of the feature embedding size and our model shows the best performance when the embedding size is 1600. At this time, we could see that the overall performance improves in proportion to the size of the embedding dimension according to Table 8.

**Optimizing Size of RNN Hidden States** The optimization of the number of RNN hidden layer dimensions is as shown in Table 9, and when the hidden state size is 800, the performance is as good as Table 8. We consider that our model with the number of moderately large dimensions shows good performance because the hidden state **e** of equation 1 is that the hidden state of the BERT and the hidden state of the feature are concatenated.

**Optimizing Size of Biaffine Hidden States** Table 10 shows the optimization of the number of biaffine hidden layer dimensions, and when the number of hidden layer dimensions is 50, the performance 69.72% of CoNLL F1 is shown as in the previous tables. We perform modeling by applying the head target class component based on the optimized hyperparameters. As a result, the performance of the single model shows 70.83% of CoNLL F1.

## A.5 Optimizing RNN types

Table 11 compares performance by RNN types such as SRU, LSTM, and GRU. We choose the RNN type suitable for Korean coreference resolution and optimize the number of layers of each RNN type. The optimal RNN type and the number of layers are 70.83% F1 with 2-layers SRU. Because the SRU uses a highway network ((Srivastava et al., 2015)), a skip connection is used to allow the gradient to directly propagate to the previous layer; the information loss is small even if the stack is deepened.

## A.6 Ensemble Knowledge Distillation

**Ensemble** Table 12 shows the performances of ten single models with different random seed and ensemble models on the dev set. We are interested in how the proposed model performs under different random initial conditions. Our model observes consistent performance regardless of 10 different initializations. The lowest performance among the 10-models is 70.04% F1, and the mean F1 score is 70.37%, both of which still outperforms the 68.62 F1 of the Korean BERT-coref from Joshi et al. (2019b). We perform a maximum score ensemble and an average score of the ensemble for 10-models. The maximum score ensemble is 72.26% F1, and the average score of the ensemble is 72.23% F1.

|  | MUC | $B^3$ | $\text{CEAF}_{\phi_4}$ | CoNLL |  |  |
|---|---|---|---|---|---|---|
| Number of dimensions | F1 | F1 | F1 | Pre. | Rec. | Avg. F1 |
| 50 | 69.71 | 66.85 | 64.60 | 63.05 | 71.65 | 67.05 |
| 100 | 69.22 | 66.86 | 65.76 | 62.66 | 72.65 | 67.28 |
| 200 | 70.14 | 67.75 | 65.85 | 65.80 | 70.21 | 67.91 |
| 400 | 70.42 | 67.93 | 66.40 | 65.38 | 71.42 | 68.25 |
| 800 | 70.56 | 67.82 | 66.32 | 65.11 | 71.69 | 68.23 |
| 1600 | **71.92** | **69.16** | **68.08** | **66.85** | **72.85** | **69.72** |

Table 8: Optimizing number of feature embedding size on the Korean ETRI dev set.

|  | MUC | $B^3$ | $\text{CEAF}_{\phi_4}$ | CoNLL |  |  |
|---|---|---|---|---|---|---|
| Number of dimensions | F1 | F1 | F1 | Pre. | Rec. | Avg. F1 |
| 50 | 69.76 | 67.68 | **69.24** | 66.48 | 71.54 | 68.89 |
| 100 | 69.72 | 66.71 | 65.16 | 64.41 | 70.25 | 67.20 |
| 200 | 69.97 | 67.14 | 65.44 | 64.27 | 71.14 | 67.52 |
| 400 | 69.26 | 67.52 | 68.63 | **66.88** | 70.17 | 68.47 |
| 800 | **71.92** | **69.16** | 68.08 | 66.85 | **72.85** | **69.72** |
| 1600 | 70.64 | 68.32 | 69.48 | 67.22 | 72.00 | 69.48 |

Table 9: Optimizing number of RNN hidden layer dimensions on the Korean ETRI dev set.

|  | MUC | $B^3$ | $\text{CEAF}_{\phi_4}$ | CoNLL |  |  |
|---|---|---|---|---|---|---|
| Number of dimensions | F1 | F1 | F1 | Pre. | Rec. | Avg. F1 |
| 50 | **71.92** | **69.16** | **68.08** | 66.85 | 72.85 | **69.72** |
| 100 | 70.94 | 68.36 | 66.77 | 67.07 | 70.48 | 68.69 |
| 200 | 70.77 | 68.22 | 66.24 | 64.44 | 72.93 | 68.41 |
| 400 | 70.97 | 68.13 | 66.56 | 63.92 | **73.92** | 68.55 |
| 800 | 70.30 | 67.39 | 65.02 | 63.43 | 72.36 | 67.57 |
| 1600 | 70.81 | 68.23 | 66.21 | **67.22** | 69.74 | 68.42 |

Table 10: Optimizing number of Biaffine hidden layer dimensions on the Korean ETRI dev set.

| RNN type | #Layer | Avg. F1 |
|---|---|---|
| SRU | 1 | 69.30 |
| SRU | 2 | **70.83** |
| GRU | 1 | 69.77 |
| GRU | 2 | 69.74 |
| LSTM | 1 | 69.31 |
| LSTM | 2 | 68.55 |

Table 11: Optimizing RNN type and the number of layers on the Korean ETRI dev set.

| Seed# | Avg. F1 | Seed# | Avg. F1 |
|---|---|---|---|
| Seed 1 | 70.04 | Seed 6 | 70.23 |
| Seed 2 | 70.31 | Seed 7 | 70.43 |
| Seed 3 | 70.45 | Seed 8 | 70.55 |
| Seed 4 | **70.83** | Seed 9 | 70.61 |
| Seed 5 | 70.11 | Seed 10 | 70.12 |

Table 12: Robustness of our model on different seeds for random initialization. The average of 10-models is 70.37%, and Std. the deviation is 0.253. Note that our official model is trained on seed 4.

But we choose the average score of the ensemble because the average ensemble is 1.28% higher than the maximum score ensemble in the test set.

**Knowledge Distillation** We optimize the weight option $\beta$ of knowledge distillation. The final loss calculated when training knowledge distillation can be divided into two methods. The first method applies $\beta$ only to the knowledge distillation loss term as $\mathcal{L} = \mathcal{L}_{ce} + \beta\mathcal{L}_{kd}$ in equation 7. The second method applies $\beta$ to both terms, such as $\mathcal{L} = (1 - \beta)\mathcal{L}_{ce} + \beta\mathcal{L}_{kd}$.

Figure 5 shows the optimization results for the hyper-parameter $\beta$ used in the knowledge distillation when training with an ensemble knowledge distillation model. The experiment uses the loss function of equation 7 with methods and optimizes $\beta$ between 0.1 and 1.0. When using the KLD, temperature (Hinton et al., 2015) is set to 5. As a result, the first method shows that the optimal performance is 71.18% F1 when $\beta$ is 0.2 on the dev set. This method improves the F1 score by 0.34% compared to the single model. When $\beta$ 0.1, F1 score is 71.06%, it is the second-best performance in the same method. In the case of the second method, when $\beta$ is 0.3 and 0.5, F1 scores are 70.66% and 71.01%, respectively, which are improved than the single model. Accordingly, we can see that knowledge distillation of $\beta$ below 0.5 is helpful for training, and it is meaningful to apply the loss of the first method to Korean coreference resolution.

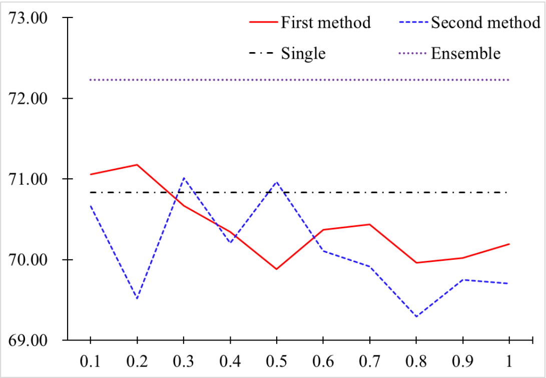

Figure 5: Hyperparameter $\beta$ optimization of knowledge distillation on dev set of Korean coreference resolution
.