Coreference Resolution as Span Boundary Alignment

Anonymous ACL submission

Abstract

We propose a new fast and accurate solution for coreference resolution, in which the task is formulated as a span boundary alignment problem. In this solution, a mention is linked to another one via two edges modeling how likely two linked mentions point to the same entity. Specifically, for each mention, its head word (left boundary) needs to be well aligned with the head words of all other mentions that refer to the same entity, so does its tail word (right boundary). Such a "head-to-head" and "tail-to-tail" alignment strategy greatly reduces the computational complexity of coreference decisions on any pair of mentions, mitigates the error propagation problem caused by mention pruning, and encourages the sharing of features across all mentions that refer to the same entity. Experimental results show that our solution achieves close to state-of-the-art performance on the CoNLL-2012 and GAP benchmarks with much less computational cost.

1 Introduction

Coreference resolution that aims to identify all the mentions referring to the same entity in a text, is considered as an important preprocessing step for various high-level natural language processing (NLP) tasks such as document summarization, question answering, and information extraction (Chen and Ng, 2016; Falke et al., 2017; Dhingra et al., 2018). Despite the significant progress has been made on the coreference resolution in recent years, there are still some challenging problems that need to be resolved.

The first one is mention detection, which is the task of extracting possible mentions from an input text, a critical preprocessing step for the coreference resolution. Previous studies show that the results of mention detection have a significant impact on the performance of coreference resolution (Lu and Ng, 2020). However, such a pipelined solution may lead to a serious error propagation problem

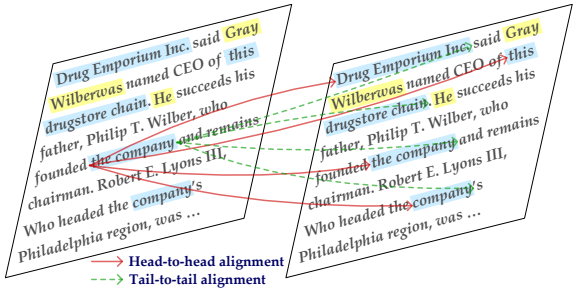

Figure 1: An example of the "head-to-head" and "tail-to-tail" alignment strategy. In this example, "*Drug Emporium Inc.*", "*this drugstore chain*", "*the company*", and "*company*" refer to the same entity. For each mention, its head word should be well aligned with the head words of the others, so does its tail word. Taking the mention of "*the company*" as an example, its head word "*the*" should be aligned with "*Drug*", "*this*", and "*company*", and its tail word "*company*" needs to align with "*Inc.*", "*chain*" and "*company*". If one of these mentions, say "*the company*", was not identified at the mention pruning stage, it has several chances (up to three times in this example) to be recovered when we align the boundary words of "*Drug Emporium Inc.*", "*this drugstore chain*", and "*company*" with their coreferent mentions.

(Clark and Manning, 2016; Wiseman et al., 2016): undetected mentions have no chance to be reconsidered and those detected incorrectly can never be corrected at the following stages. Recently, the end-to-end framework has been proposed to tackle this problem, which jointly learns to detect mentions and cluster them into groups (Lee et al., 2017, 2018; Joshi et al., 2019a). However, the computational complexity of such a solution is $O(T^4)$, where T is the length of an input text, because every possible span needs to be considered, and every pair of those spans should be evaluated for possible co-referring. To reduce this intractable complexity, the candidate spans still need to be pruned. Therefore, the mentions filtered out at the pruning stage can never be recovered, which greatly impairs the performance of models with the end-to-end framework.

Secondly, how to leverage entity-level features remains an unresolved problem. Previous stud-

ies either count on the long-term memory (LSTM) or pre-trained transformer to implicitly capture the global features (Lee et al., 2017; Zhang et al., 2018) or incorporate the features of the clusters already formed to determine whether a mention is coreferent with a preceding cluster (Lee et al., 2018; Kantor and Globerson, 2019). The former might miss out some important features for specific pairwise predictions without explicit entity-level features, while the latter may suffer from error propagation as false clusters are used to create entity-level features when making future predictions.

Recently, Wu et al. (2020) present CorefQA that provides a new solution to tackle the above-mentioned problems, in which the coreference resolution is formulated as a question answering problem. Taking a mention (pruned) and its surrounding context as a query, their model is trained to retrieve all the mentions that refer to the same entity as the query. In this way, the mentions filtered out at the pruning stage gain another chance to be "reborn". Their model achieved state-of-the-art performance on CoNLL-2012 and GAP datasets but requires considerable computational cost. For each possible mention, a query needs to be created and then answered by accessing the entire text.

Inspired by CorefQA (Wu et al., 2020), we formulate the coreference resolution as a span boundary alignment problem. Specifically, for a set of mentions referring to the same entity, our model is trained to align their heads (left boundary words) and tails (right boundary words) as well as possible. As shown in Figure 1, "*Drug Emporium Inc.*", "*this drugstore chain*", "*the company*", and "*company*" refer to the same entity. For each mention in this set, its head word should be well aligned with the head words of the others, so does its tail word. Taking the mention of "*the company*" as an example, its head word "*the*" should be aligned with "*Drug*", "*this*", and "*company*", and its tail word "*company*" needs to align with "*Inc.*", "*chain*" and "*company*". If one of these mentions, say "*the company*", was not identified at the mention pruning stage, it has several chances (up to three times in this example) to be recovered when we align the boundary words of "*Drug Emporium Inc.*", "*this drugstore chain*", and "*company*" with their coreferent mentions. After those "head-to-head" and "tail-to-tail" pairs have been well aligned, the coreference decision on each mention-pair can base on such "head-to-head" and "tail-to-tail" alignment scores rather than their mention-level representations. Besides, the head (or tail) representations of all possible mentions in a text are produced at a time and updated together accordingly, which greatly speeds up the training and inference time and encourages the sharing of entity-level features across all mentions referring to the same entity.

Our contributions are threefold: (1) We propose a new solution to the coreference resolution, which better leverages entity-level features and deals with the error propagation problem caused at the mention pruning stage; (2) The solution greatly reduces the computational cost by factorizing the mention-pair scores into "head-to-head" and "tail-to-tail" alignment scores; (3) Experimental results show that the proposed framework achieved close to state-of-the-art performance on two widely-used benchmarks with minimal computational cost.

2 Related Work

Coreference resolution is a long-standing challenging task which is considered as a critical step to process texts semantically for many NLP applications (Ng, 2010). Existing approaches can roughly be divided into two categories: mention-pair ranking (Bengtson and Roth, 2008; Stoyanov et al., 2010; Wiseman et al., 2015) and entity-mention models (Poon and Domingos, 2008; Björkelund and Kuhn, 2014; Clark and Manning, 2015). The former makes each coreference decision independently without taking entity-level information into consideration, while the latter addresses the lack of global information by considering whether a mention is coreferent with a previously formed cluster. However, how to better capture and leverage global entity-level information is still not well resolved.

On the other hand, traditional coreference resolution approaches usually involve a preprocessing step of mention detection, which often uses some hand-engineered mention proposal algorithms to identify possible mentions (Raghunathan et al., 2010; Clark and Manning, 2015; Wiseman et al., 2016). The errors caused in the mention detection may propagate to the following step. To tackle this problem, Lee et al. (2017) designed an end-to-end solution that takes every possible span (or a sequence of words) in a document as a candidate mention, and their model is trained to jointly perform the mention detection and coreference prediction. Zhang et al. (2018) improved such a solution by using a biaffine attention to estimate the probability of a mention-pair. To reduce the computational cost

of the end-to-end solution, Lee et al. (2018) proposed a coarse-to-fine approach that incorporates a less accurate but more efficient bilinear approximation, which leads to more aggressive mention pruning without hurting accuracy too much.

Global entity-level information can be incorporated by using joint inference algorithms (McCallum and Wellner, 2003; Poon and Domingos, 2008; Haghighi and Klein, 2010) or creating coreference clusters incrementally (Luo et al., 2004; Yang et al., 2008; Raghunathan et al., 2010). Lee et al. (2018) tried to learn an antecedent distribution for each span with a span-ranking architecture and to improve a span representation by integrating features from its possible antecedents. Kantor and Globerson (2019) refined the feature representation of a span with the information derived from the entire cluster to which it belongs. Graph neural networks (GNNs) were also introduced to make the global information be shared among mentions in both forward and backward directions (Liu et al., 2020).

Very recently, Wu et al. (2020) addressed two issues in existing coreference resolution systems (Lee et al., 2017; Zhang et al., 2018; Lee et al., 2018; Joshi et al., 2019a). One is that correct mentions might be filtered out at the mention proposal stage, and the relationship between mentions and their contexts (including their coreferent mentions) has not been well modeled. To resolve these two issues, they proposed a new approach in which the coreference resolution problem is formulated as a span prediction task, akin to the question answering. Their model achieved state-of-the-art results but at the cost of high computational complexity which presents serious scalability and performance-per-mention challenges. In this study, we present a new solution to leverage entity-level features and to recover the mentions filtered out at the mention proposal stage by factorizing mention-pair scores to "head-to-head" and "tail-to-tail" alignments, which greatly reduces the computational cost while suffering little to no performance drop.

3 Methods

We describe our solution in this section in which the coreference resolution is innovatively formulated as a span boundary alignment problem. The idea behind this solution is that the left boundary words (head) of mentions referring to the same entity should be well aligned, and so does the right boundary words (tail). In this way, span-level matching can be factorized into word-level alignments which significantly speeds up the training and inference.

3.1 Problem Definition

Given a document $D = \{x_1, \ldots, x_n\}$ consists of n words, there are $n(n+1)/2$ possible text spans. The goal is to find an antecedent y_k for each span k recognized as a mention. Formally, let $m_k = \{x_{\text{hd}(k)}, \ldots, x_{\text{tl}(k)}\}$ denotes the k-th span starting with hd(k) word and ending with tl(k) word (included), where $1 \leq k \leq N$, $N = n(n+1)/2$. A set of candidate antecedents for a span m_k is denoted as $\mathcal{Y}_{m_k} = \{\epsilon, m_1, ..., m_{k-1}\}$ that consists of all the preceding spans and a dummy antecedent (denoted as ϵ). A non-dummy antecedent indicates a coreference link between a span and one of its antecedents. If a span is linked to the dummy, we say the span is not an entity mention, or it is an mention but not coreferent with any antecedent.

3.2 Input Feature Representation

We chose to use SpanBERT (Joshi et al., 2019a) to produce the contextual word embeddings for any input text. SpanBERT only can process text within a limited length. To fit long documents into SpanBERT, we split them into multiple segments of equal length. The last segment will be extended into the same length by padding it at the end. We use non-overlapping segments, each of which is considered as an independent instance.

The boundaries of text spans play a critical role in our solution, which largely determine the results of both mention detection and alignment. Note that each word in an input text could become a head of a mention or a tail of another, and their feature representations should be derived differently. For each word, we project its contextual word embedding produced by SpanBERT to two vector spaces for the cases of being head and tail words as follows.

$$h_{x_i} = W_h \cdot e_{x_i}, \quad t_{x_i} = W_t \cdot e_{x_i} \tag{1}$$

where e_{x_i} is the contextual word embedding of i-th word in an input text, and two matrices of W_h and W_t are trainable parameters used to produce the vector representations of h_{x_i} and t_{x_i} when the word x_i is taken as head or tail word.

3.3 Span Boundary Alignment

We require that all the head and tail words are well aligned if the mentions defined by these head and tail words refer to the same entity. Some scoring functions are required to estimate how well any two words are aligned. We apply a biaffine attention

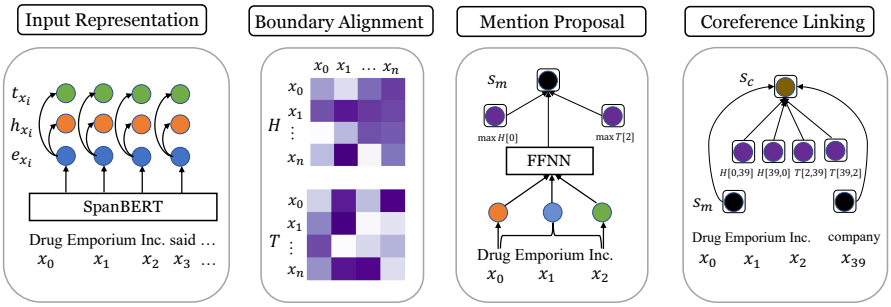

Figure 2: The main steps of our solution. A document is first fed into SpanBERT to obtain the contextual word embedding for every word in the document, and these embeddings are further projected to two vector spaces each for the case of being head or tail words. After the head and tail representations are obtained, "head-to-head" and "tail-to-tail" alignment scores are calculated and stored in two span boundary alignment matrices H and T. With these alignment matrices, the mention pruning and coreference linking can be performed. In this way, span-level matching can be factorized into word-level alignments which significantly speeds up the training and inference.

function to calculate such an alignment score for every pair of words in a document as Equation (2). For the "head-to-head" alignment, we obtain a matrix $H \in \mathbb{R}^{n \times n}$ whose element at i-th row and j-th column $H[i,j]$ is a real valued score that represents how well the words x_i and x_j are aligned as head words. Similarly, a matrix T can be obtained for the "tail-to-tail" alignment.

$$\begin{aligned} H[i,j] &= h_{x_j}^\top U_h h_{x_i} + v_h h_{x_i} \\ T[i,j] &= t_{x_j}^\top U_t t_{x_i} + v_t t_{x_i} \end{aligned} \tag{2}$$

where $U_h, U_t \in \mathbb{R}^{n \times n}$ and $v_h, v_t \in \mathbb{R}^n$ are trainable parameters to be tuned. The first term in each biaffine function models the compatibility of two words that reflects the possibility of being head or tail words, and the second term is used to model the prior likelihood of x_i as a head or a tail word.

Refining Representation To encourage the sharing of features across all mentions that refer to the same entity, we refine the head and tail representations of h_{x_i} and t_{x_i} for each word x_i by aggregating the features from the others according to their alignment scores. Taking the head representation refinement as an example, we represent each word as a node in a graph, and the node of word x_i is connected to the nodes that have top-K alignment scores with respect to word x_i. Graph neural networks are applied to aggregate the entity-level features and to update the representations.

$$\begin{aligned} \alpha_{ij}^l &= \frac{\exp^{H[i,j]}}{\sum_{r=1}^{K} \exp^{H[i,r]}} \\ a_i^l &= \sum_{j \in \mathcal{N}(i)} \alpha_{ij}^l h_{x_j}^l \\ \beta_i^l &= \text{Sigmoid}(W_f [h_{x_i}^l, a_i^l]) \\ h_{x_i}^{l+1} &= \beta_i^l \circ a_i^l + (1 - \beta_i^l) \circ h_{x_i}^l \end{aligned} \tag{3}$$

where $\mathcal{N}(i)$ is a set of word i's connected neighbors in the graph, and $h_{x_i}^l$ is the head representation of word x_i at the l-th layer. After the head and tail representations are refined with entity-level features, the matrices H and T will be updated by using Equation (2) again with the same parameters.

Alignment Loss The model will be trained to increase the alignment scores for those word pairs that are the head (or tail) words of mentions referring to the same entity and to decrease the scores for the others. Thus, the loss function for word alignment can be defined as follows.

$$\begin{aligned} L_{\text{align}}(k,g) =& y_{kg} \log \hat{y}_{kg} + (1 - y_{kg}) \log(1 - \hat{y}_{kg}) \\ \hat{y}_{kg} =& \frac{1}{2}(\text{Sigmoid}(H[\text{hd}(k), \text{hd}(g)]) \\ &+ \text{Sigmoid}(T[\text{tl}(k), \text{tl}(g)])) \end{aligned} \tag{4}$$

where $y_{kg} = 1$ if two span m_k and m_g are coreferent mentions, and $y_{kg} = 0$ otherwise.

3.4 Mention Proposal

Like (Lee et al., 2017), we consider all possible spans up to a maximum length of Q words. To improve computational efficiency, we prune the candidate spans at the training and inference time. To obtain the scores for mention pruning, we take both the boundary and internal information into account. The spans with scores higher than a given threshold will be selected as candidate mentions. For each span m_k, its feature representation is derived from its constituent words.

$$\begin{aligned} \varphi_i &= \text{FFNN}_\varphi(e_{x_i}) \\ \psi_{ik} &= \frac{\exp(\varphi_i)}{\sum_{r=\text{hd}(k)}^{\text{tl}(k)} \exp(\varphi_r)} \\ e_{m_k} &= \sum_{i \in \{\text{hd}(k), \ldots, \text{tl}(k)\}} \varphi_{ik} e_{x_i} \\ s_d(m_k) &= \text{FFNN}_m([e_{x_{\text{hd}(k)}}, e_{x_{\text{tl}(k)}}, e_{m_k}, \phi(m_k)]) \end{aligned} \tag{5}$$

where FFNN_φ and FFNN_m are two feed-forward networks used to estimate how important a word is to a span and how likely a span is a mention. $\phi(m_k)$ is the length of span m_k.

Mention Pruning We want to filter out the spans that are unlikely to be mentions and those said to be *singleton mentions*, which have no other coreferent mentions in the document. To prune the singleton mentions, we consider whether or not a candidate mention has other coreferent mentions. Therefore, we estimate the the possibility of whether a span in question is aligned with others as follows.

$$\begin{aligned} s_h(m_k) &= \max_{j=\{1,\ldots,n\}, j\neq \text{hd}(k)} H[\text{hd}(k), j] \\ s_t(m_k) &= \max_{j=\{1,\ldots,n\}, j\neq \text{tl}(k)} T[\text{tl}(k), j] \\ s_m(m_k) &= s_d(m_k) + \gamma(s_h(m_k) + s_t(m_k)) \end{aligned} \tag{6}$$

where is γ a hyperparameter that governs the importance of two terms.

We only consider up to λn spans with the highest scores, where n is the length of an input document and $\lambda \in [0, 1]$. The loss for the mention detection is defined as follows.

$$L_{\text{detect}}(k) = t_k \log \hat{t}_k + (1 - t_k) \log(1 - \hat{t}_k) \tag{7}$$

where $\hat{t}_k = \text{Sigmoid}(s_m(m_k))$, and $t_k = 1$ if m_k is a truth mention, and $t_k = 0$ otherwise.

3.5 Coreference Linking

Given a mention-pair of m_k and m_g proposed at the mention proposal stage, the coreference linking module assigns a score $s(k, g)$ to this pair, which indicates how likely m_k and m_g refer to the same entity. In our solution, such mention-pair scores can be factorized into word-level "head-to-head" and "tail-to-tail" alignment scores as follows.

$$\begin{aligned} s_a(k|g) &= H[\text{hd}(k), \text{hd}(g)] + T[\text{tl}(k), \text{tl}(g)] \\ s_a(g|k) &= H[\text{hd}(g), \text{hd}(k)] + T[\text{tl}(g), \text{tl}(k)] \\ s_a(m_k, m_g) &= s_a(k|g) + s_a(g|k) \\ s_c(m_k, m_g) &= s_m(m_k) + s_m(m_g) + s_a(m_k, m_g) \end{aligned} \tag{8}$$

At the coreference linking stage, we recalculate the span boundary alignment scores using the refined representations produced by graph neural networks to obtain more accurate alignment results.

$$\begin{aligned} H_f[\text{hd}(k), \text{hd}(g)] &= \text{FFNN}_h(h^L_{x_{\text{hd}(k)}}, h^L_{x_{\text{hd}(g)}}, \phi(m_k, m_g)) \\ T_f[\text{tl}(k), \text{tl}(g)] &= \text{FFNN}_t(t^L_{x_{\text{tl}(k)}}, t^L_{x_{\text{tl}(g)}}, \phi(m_k, m_g)) \end{aligned} \tag{9}$$

where $h^L_{x_{\text{hd}(k)}}$ and $t^L_{x_{\text{tl}(k)}}$ are refined head and tail representations produced by GNNs for words $x_{\text{hd}(k)}$ and $x_{\text{tl}(k)}$ respectively, L is the number of GNN's layers, and $\phi(m_k, m_g)$ are a set of features derived from the attributes of speaker, genre, and the distance between a pair of mentions. The score of coreference resolution $s_r(m_k, m_g)$ can be divided into two parts: word alignment score $s_c(m_k, m_g)$ and pseudo-coreference linking score $s_f(m_k, m_g)$:

$$\begin{aligned} s_f(m_l, m_g) &= H_f[\text{hd}(k), \text{hd}(g)] + T_f[\text{tl}(k), \text{tl}(g)] \\ s_r(m_k, m_g) &= s_c(m_k, m_g) + s_f(m_k, m_g) \end{aligned} \tag{10}$$

3.6 Training and Inference

There are two main stages in our solution: mention pruning and coreference linking. Although they can be trained in a joint manner, to make the training process more stable and speed up the training time, we chose to use a two-stage training strategy. The loss functions L_{align} and L_{detect} are first applied to warm-up a model, and then the loss $L_{cluster}$ defined below is used to train the model jointly. For each mention m_k recommended at the mention proposal stage, the model is trained to optimize the marginal log-likelihood over all the antecedents as follows.

$$\begin{aligned} L_{\text{cluster}}(m_k) &= -\log \sum_{m_g \in \mathcal{Y}_{m_k} \cap \text{GOLD}(m_k)} p(m_g) \\ p(m_g) &= \frac{\exp^{s_r(m_k, m_g)}}{\sum_{m'_g \in \mathcal{Y}_{m_k}} \exp^{s_r(m_k, m'_g)}} \end{aligned} \tag{11}$$

where $\text{GOLD}(m_k)$ is a set of mentions that m_k is coreferent with. If m_k is not a mention or does not have any antecedent, then $\text{GOLD}(m_K) = \{\epsilon\}$.

At the inference time, for each possible mention m_k, we take the span m_g with the highest score of $s_r(m_k, m_g)$ as its antecedent, and those mentions that are only linked to the dummy ϵ will be abandoned. The final results of coreference resolution can be easily derived from such mention-antecedent pairs.

4 Experiments

4.1 Experimental Settings

4.1.1 Datasets

We evaluated the proposed solution and an implemented model, named Mention Boundary Alignment (MBA), by comparing it to eight strong competitors on two datasets. One is the English portion of CONLL-2012 shared task (Pradhan et al., 2012), which is widely used for coreference resolution evaluation. Another is GAP dataset (Webster

Table 1: Results on the test set of the English CoNLL-2012 shared task. The scores of "MUC", "B_3", "$CEAF_{\phi4}$" and their average F1-scores (indicated by "Avg. F1") are used as the evaluation metrics. The symbols of "P", "R" and "F1" shown in the second row denote the precision, recall, and F1-scores respectively.

Model	MUC			B_3			$CEAF_{\phi4}$			Avg. F1
	P	R	F1	P	R	F1	P	R	F1	
E2E-Coref	78.4	73.4	75.8	68.6	61.8	65.0	62.7	59.0	60.8	67.2
E2E-Coref + MD	79.4	73.8	76.5	69.0	62.3	65.5	64.9	58.3	61.4	67.8
C2F-Coref	81.4	79.5	80.4	72.2	69.5	70.8	68.2	67.1	67.6	73.0
C2F-Coref + RL	85.4	77.9	81.4	77.9	66.4	71.7	70.6	66.3	68.4	73.8
EE + BERT-large	82.6	84.1	83.4	73.3	76.1	74.7	72.4	71.1	71.8	76.6
C2F-Coref (BERT-base)	80.2	82.4	81.3	69.6	73.8	71.6	69.0	68.6	68.8	73.9
C2F-Coref (SpanBERT-base)	83.5	85.5	84.5	75.0	76.2	75.6	73.8	72.2	73.0	77.7
CorefQA (SpanBERT-base)	85.2	87.4	86.3	78.7	76.5	77.6	76.0	75.6	75.8	**79.9**
MBA (SpanBERT-base)	84.0	85.8	84.9	75.6	76.6	76.1	73.9	72.7	73.3	78.1

et al., 2018) consisting of manually labeled ambiguous pronoun-name pairs extracted from Wikipedia snippets. We used the standard splits of the English CoNLL-2012 dataset for training, development, and testing. Following the protocol established in (Webster et al., 2018; Joshi et al., 2019b), we used the coreference resolvers trained on the CoNLL-2012 dataset to test (without fine-tuning) their performance on the GAP test set[1].

4.1.2 Evaluation Metrics

Strictly following the evaluation convention established in the CoNLL-2012 shared task, we use the link-based MUC (Vilain et al., 1995), mention-based B3 (Bagga and Baldwin, 1998), entity-based CEAF (Luo, 2005), and their unweighted average as evaluation metrics. The scores of these metrics are calculated by using an official toolkit of the CoNLL-2012 evaluation scripts[2].

4.1.3 The Choice of Hyper-parameters

We tuned the hyper-parameters by trying only a few different settings on the validation sets. The dimensionality of head and tail representations was set to the same size as the hidden layers of SpanBERT. The maximum length of candidate mentions was set to 30 words, the ratio λ used to prune possible mentions to 0.4, and the length of segments for splitting long documents to 384 as Joshi et al. (2019a). We used the Adam optimizer for the training and used a learning rate of 0.1×10^{-6} to update the weights of SpanBERT and another rate of 0.2×10^{-5} to update the other parameters. A two-layer graph neural network ($L = 2$) was used to refine the head and tail representations by aggregating the features from their top-5 neighbors.

[1]This is motivated by the fact that there are only $4,000$ name-pronoun pairs in the GAP dataset, which was not created for full-scale training.

[2]http://conll.cemantix.org/2012/software.html

Table 2: Results on the test set of GAP dataset. We reported F1-scores on masculine (indicated by "M") and feminine (indicated by "F") examples as well as their bias factors (indicated by "B") calculated by F1-scores on feminine examples divided by those on masculine ones and overall F1-scores (indicated by "O") on all the test examples. †The results of CorefQA were excerpted from the numbers reported by Wu et al. (2020).

Model	M	F	B	O
E2E-Coref	67.2	62.2	0.92	64.7
C2F-Coref	75.8	71.1	0.94	73.5
C2F-Coref (BERT-base)	84.4	81.2	0.96	82.8
C2F-Coref (SpanBERT-base)	88.5	84.1	0.95	86.3
CorefQA (SpanBERT-large) †	88.9	86.1	0.97	**87.5**
MBA (SpanBERT-base)	88.5	84.9	0.96	86.7

Table 3: Inference speed (the number of samples per second) of the three most competitive models evaluated on the test set of two datasets. All the three models evaluated were built upon the SpanBERT-base.

Model	Inference Speed	
	CoNLL-2012	GAP
CorefQA	0.11	0.58
C2F-Coref	4.95	19.70
MBA	**5.04** (×**45.82** ↑)	**20.69** (×**35.67** ↑)

4.2 Baseline Models

The eight representative models with the end-to-end framework were used for comparison.

- E2E-Coref is the first coreference resolution model built with the end-to-end framework (Lee et al., 2017).
- E2E-Coref + MD extends E2E-Coref by using a biaffine attention to estimate the probability of each possible mention-pair (Zhang et al., 2018).
- C2F-Coref also extends E2E-Coref by combining

a coarse-to-fine pruning strategy with a higher-order inference mechanism (Lee et al., 2018).

- C2F-Coref + RL further extents C2F-Coref by introducing reinforcement learning to directly optimize the evaluation metrics of coreference resolution (Fei et al., 2019).
- EE (BERT-large) refines the vector representation of a mention by taking all the other mentions that most likely refer to the same entity into account (Kantor and Globerson, 2019).
- C2F-Coref (BERT-base) extends C2F-Coref by replacing the LSTM-based encoder with a pre-trained transformer (Joshi et al., 2019b).
- C2F-Coref (SpanBERT-base) uses SpanBERT to obtain representations of spans in a document for coreference resolution (Joshi et al., 2019a).
- CorefQA (SpanBERT-base) formulates the problem of coreference resolution as a span prediction task, like question answering (Wu et al., 2020).

4.3 Experimental Results

We report in Table 1 the results of our model and other competitors on the CoNLL-2012 benchmark dataset. We only give the results yielded by the models built on the base version of SpanBERT, which greatly reduces the computational resources required for training. As we can see from Table 1, our model (indicating by "MBA") outperforms all the other competitors except CorefQA. Although our model slightly performs worse than CorefQA, it performs comparably and runs more than 45 times faster than CorefQA on the CoNLL-2012 shared dataset and about 35 times faster on the GAP dataset (see Table 3 for details). Although CorefQA achieved the highest average F1-scores on the two datasets, yet at the cost of great computational time and resources. For each possible mention, a query must be constructed and answered by processing the entire document again and again with a large architecture (i.e., BERT or SpanBERT), which makes it hard for CorefQA to scale to long documents as the number of candidate mentions increases. As shown in Table 3, MBA runs much faster on the CoNLL-2021 shared task than on the GAP dataset, comparing to CorefQA model because the length of documents in the former dataset (about 454 words) is more than that in the latter (around 71 words) on average.

The results on the GAP are reported in Tables 2 and 3, and we found similar trends as that on the CoNLL-2012 shared task. The results on the two benchmark datasets show that MBA achieves consistently both higher performance and lower computational cost over the competitors except CoreQA. The two issues raised by Wu et al. (2020) are well addressed in our proposed solution. Firstly, the mentions filtering out at the mention pruning stage will be given at least one chance (usually multiple chances) to be recovered when we try to align the boundary words with their coreferent mentions. Such an alignment strategy enables us to perform a more aggressive pruning, which helps to speed up the training and inference time. Secondly, our solution encourages the sharing of features among the mentions referring to the same entity by forcing their head and tail words to be aligned with each other in the form of mutually refined vector representations via graph neural networks.

Table 4: Ablation study on the development set of CoNLL-2012 dataset. The experiment results show that all the components contribute to the overall system.

Model	Avg. F1	Δ
MBA	78.1	
w/o Replacing SpanBERT	74.3	−3.8
w/o Updating alignment score	75.7	−2.4
w/o Pruning singleton	77.6	−0.5
w/o Refining representation	77.7	−0.4

5 Ablation Study and Qualitative Analysis

We performed comprehensive ablation study and some analysis on the development set of the CoNLL-2012 dataset. We remove one component at a time to understand the contribution of the removed component to the overall system by observing the changes in the performance. The results of the ablation study are reported in Table 4.

5.1 Impacts of Different Components

Replacing SpanBERT The model exhibits 3.7 degradation in F1-score after replacing the SpanBERT with a vanilla BERT, which confirms the importance of span-level pre-training for coreference resolution and is consistent with the previous findings of (Joshi et al., 2019a).

Refining Representation GNNs were employed to refine head and tail representations to encourage the sharing of features among the mentions referring to the same entity, which brings about a moderate improvement in F1-score.

Pruning Singleton At the mention pruning stage, we filter out both the spans that are most unlikely to

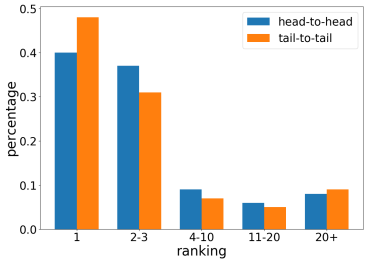

Figure 3: The percentage of correct "head-to-head" and "tail-to-tail" alignments when the words are ranked at 1, 2-3, 4-10, 11-20, and 20+ positions by the estimated alignment scores on the CoNLL-2012 development set.

Table 5: The precision, recall and F1-scores of mention detection on the CoNLL-2012 development set.

Model	Precision	Recall	F1
C2F-Coref	86.2	83.7	84.9
C2F-Coref + RL	**89.6**	82.2	85.7
C2F-Coref (SpanBERT-base)	87.2	87.5	87.4
MBA (SpanBERT-base)	87.0	**88.4**	**87.7**

be mentions and the singleton mentions by considering whether a candidate mention has other coreferent mentions. The experimental results demonstrate that distinguishing between singleton and non-singleton mentions helps to improve the performance of coreference resolution.

Updating Alignment Score We will recalculate the span boundary alignment scores using the refined feature representations produced by GNNs as Equation (9), which boosts the F1-score by a significant margin.

5.2 Analysis and Discussion

5.2.1 Alignment Matrices

We would like to know how well the alignment scores estimated by using Equation (2) reflect the ground truth of coreferent mentions. For each mention m_k, we sort its alignment scores of $H[\text{hd}(k)]$ and $T[\text{tl}(k)]$ in descending order, and compute the percentage of correct alignments when they are ranked at 1, 2-3, 4-10, 11-20, and 20+ positions respectively. As we can see in Figure 3, the alignment scores generally can be used to differentiate the correct "head-to-head" and "tail-to-tail" word alignment from the incorrect ones. More than 40% of top-1 predictions are the ground truth alignments.

5.2.2 Mention Detection

It is well known that the results of mention detection greatly impact the performance of the coreference resolution system. We individually evaluated the performance of models on this task and reported the experimental results in Table 5. Since we consider whether or not a candidate mention has other coreferent mentions by their alignment scores to prune the mentions, MDA achieved the best F1-scores in the mention detection task, comparing to existing representative models.

5.2.3 Qualitative Analysis

We show in Table 6 three examples that C2F-Coref fails to resolve, by correctly predicted by the proposed MBA. In Examples 1 and 2, "Greg Lefevre" and "the speech that ..." were not identified as candidate mentions at the mention pruning stage, but they were successfully recovered by MBA at the coreference linking step by searching for the antecedents for the mentions of "its release" and "that speech". As shown in Example 3, C2F-Coref incorrectly merged {"The followers"} and {"Some Pharisees", "They"} into the same cluster. Although it is plausible to select "They" as the antecedent of "The followers", there is a conflict between "The followers" and "Some Pharisees". Our MBA can alleviate this problem by taking the entity-level features into consideration.

Table 6: Example coreferent mentions that correctly predicted by our MBA, but incorrectly predicted by C2F-Coref (SpanBERT-base). The coreferent mentions are indicated by the same colors.

1	As Greg Lefevre reports, many stores have sold out of the game even before its release.
2	We're all getting, this news in from the speech that ... ... Might get more information from Secretary Tom Ridge when he delivers that speech over at the press ...
3	Some Pharisees came to Jesus. They tried to make him say something wrong ... The followers said to Jesus ...

6 Conclusion

We proposed a new solution to the coreference resolution task, which is formulated as a problem of span boundary alignment. In this solution, the models are trained to align the heads (left boundary words) and tails (right boundary words) of the mentions referring to the same entity. From the alignment results, entity mentions and the coreference relations between them can be easily derived. In this way, the mentions filtered out in the mention proposal stage can be recovered at the coreference linking stage and the entity-level features can be well leveraged. Experimental results show that our solution delivers close to state-of-the-art performance with much less computational costs.

References

Amit Bagga and Breck Baldwin. 1998. Algorithms for scoring coreference chains. In *The first international conference on language resources and evaluation workshop on linguistics coreference*, volume 1, pages 563–566. Citeseer.

Eric Bengtson and Dan Roth. 2008. Understanding the value of features for coreference resolution. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 294–303. Association for Computational Linguistics.

Anders Björkelund and Jonas Kuhn. 2014. Learning structured perceptrons for coreference resolution with latent antecedents and non-local features. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 47–57.

Chen Chen and Vincent Ng. 2016. Joint inference over a lightly supervised information extraction pipeline: Towards event coreference resolution for resource-scarce languages. In *Thirtieth AAAI Conference on Artificial Intelligence*.

Kevin Clark and Christopher D Manning. 2015. Entity-centric coreference resolution with model stacking. 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 1405–1415.

Kevin Clark and Christopher D Manning. 2016. Improving coreference resolution by learning entity-level distributed representations. *arXiv preprint arXiv:1606.01323*.

Bhuwan Dhingra, Qiao Jin, Zhilin Yang, William W Cohen, and Ruslan Salakhutdinov. 2018. Neural models for reasoning over multiple mentions using coreference. *arXiv preprint arXiv:1804.05922*.

Tobias Falke, Christian M Meyer, and Iryna Gurevych. 2017. Concept-map-based multi-document summarization using concept coreference resolution and global importance optimization. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 801–811.

Hongliang Fei, Xu Li, Dingcheng Li, and Ping Li. 2019. End-to-end deep reinforcement learning based coreference resolution. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 660–665.

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.

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

Mandar Joshi, Omer Levy, Daniel S Weld, and Luke Zettlemoyer. 2019b. Bert for coreference resolution: Baselines and analysis. *arXiv preprint arXiv:1908.09091*.

Ben Kantor and Amir Globerson. 2019. Coreference resolution with entity equalization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 673–677.

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.

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 Association for Computational Linguistics*, pages 687–692.

Lu Liu, Zhenqiao Song, and Xiaoqing Zheng. 2020. Improving coreference resolution by leveraging entity-centric features with graph neural networks and second-order inference. *arXiv preprint arXiv:2009.04639*.

Jing Lu and Vincent Ng. 2020. Conundrums in entity reference resolution. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6620–6631.

Xiaoqiang Luo. 2005. On coreference resolution performance metrics. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 25–32.

Xiaoqiang Luo, Abe Ittycheriah, Hongyan Jing, Nanda Kambhatla, and Salim Roukos. 2004. A mention-synchronous coreference resolution algorithm based on the bell tree. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, pages 135–142.

Andrew McCallum and Ben Wellner. 2003. Toward conditional models of identity uncertainty with application to proper noun coreference.

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. Association for Computational Linguistics.

Hoifung Poon and Pedro Domingos. 2008. Joint unsupervised coreference resolution with markov logic. In *Proceedings of the 2008 conference on empirical methods in natural language processing*, pages 650–659.

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. Association for Computational Linguistics.

Karthik Raghunathan, Heeyoung Lee, Sudarshan Rangarajan, Nathanael Chambers, Mihai Surdeanu, Dan Jurafsky, and Christopher D 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.

Veselin Stoyanov, Claire Cardie, Nathan Gilbert, Ellen Riloff, David Buttler, and David Hysom. 2010. Reconcile: A coreference resolution research platform. Technical report.

Marc Vilain, John D Burger, John Aberdeen, Dennis Connolly, and Lynette Hirschman. 1995. A model-theoretic coreference scoring scheme. In *Sixth Message Understanding Conference (MUC-6): Proceedings of a Conference Held in Columbia, Maryland, November 6-8, 1995*.

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, and Stuart M Shieber. 2016. Learning global features for coreference resolution. *arXiv preprint arXiv:1604.03035*.

Sam Joshua Wiseman, Alexander Matthew Rush, Stuart Merrill Shieber, and Jason Weston. 2015. Learning anaphoricity and antecedent ranking features for coreference resolution. In *Proceedings of the 53th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.

Wei Wu, Fei Wang, Arianna Yuan, Fei Wu, and Jiwei Li. 2020. Corefqa: Coreference resolution as query-based span prediction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6953–6963.

Xiaofeng Yang, Jian Su, Jun Lang, Chew Lim Tan, Ting Liu, and Sheng Li. 2008. An entity-mention model for coreference resolution with inductive logic programming. In *Proceedings of Acl-08: Hlt*, pages 843–851.

Rui Zhang, Cícero Nogueira dos Santos, Michihiro Yasunaga, Bing Xiang, and Dragomir R. Radev. 2018. Neural coreference resolution with deep biaffine attention by joint mention detection and mention clustering. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pages 102–107.