End-to-end Neural Coreference Resolution

Kenton Lee[†], **Luheng He**[†], **Mike Lewis**[‡], **and Luke Zettlemoyer**[†*]

[†] Paul G. Allen School of Computer Science & Engineering, Univ. of Washington, Seattle, WA
[*] Allen Institute for Artificial Intelligence, Seattle WA
{kentonl, luheng, lsz}@cs.washington.edu

[‡] Facebook AI Research, Menlo Park, CA
mikelewis@fb.com

Abstract

We introduce the first end-to-end coreference resolution model and show that it significantly outperforms all previous work without using a syntactic parser or hand-engineered mention detector. The key idea is to directly consider all spans in a document as potential mentions and learn distributions over possible antecedents for each. The model computes span embeddings that combine context-dependent boundary representations with a head-finding attention mechanism. It is trained to maximize the marginal likelihood of gold antecedent spans from coreference clusters and is factored to enable aggressive pruning of potential mentions. Experiments demonstrate state-of-the-art performance, with a gain of 1.5 F1 on the OntoNotes benchmark and by 3.1 F1 using a 5-model ensemble, despite the fact that this is the first approach to be successfully trained with no external resources.

1 Introduction

We present the first state-of-the-art coreference resolution model that is learned end-to-end given only gold mention clusters. All recent coreference models, including neural approaches that achieved impressive performance gains (Wiseman et al., 2016; Clark and Manning, 2016b,a), rely on syntactic parsers, both for head-word features and as the input to carefully hand-engineered mention proposal algorithms. We demonstrate for the first time that these resources are not required, and in fact performance can be improved significantly without them, by training an end-to-end neural model that jointly learns which spans are entity mentions and how to best cluster them.

Our model reasons over the space of all spans up to a maximum length and directly optimizes the marginal likelihood of antecedent spans from gold coreference clusters. It includes a *span*-ranking model that decides, for each span, which of the previous spans (if any) is a good antecedent. At the core of our model are vector embeddings representing spans of text in the document, which combine context-dependent boundary representations with a head-finding attention mechanism over the span. The attention component is inspired by parser-derived head-word matching features from previous systems (Durrett and Klein, 2013), but is less susceptible to cascading errors. In our analyses, we show empirically that these learned attention weights correlate strongly with traditional headedness definitions.

Scoring all span pairs in our end-to-end model is impractical, since the complexity would be quartic in the document length. Therefore we factor the model over unary mention scores and pairwise antecedent scores, both of which are simple functions of the learned span embedding. The unary mention scores are used to prune the space of spans and antecedents, to aggressively reduce the number of pairwise computations.

Our final approach outperforms existing models by 1.5 F1 on the OntoNotes benchmark and by 3.1 F1 using a 5-model ensemble. It is not only accurate, but also relatively interpretable. The model factors, for example, directly indicate whether an absent coreference link is due to low mention scores (for either span) or a low score from the mention ranking component. The head-finding attention mechanism also reveals which mention-internal words contribute most to coreference decisions. We leverage this overall interpretability to do detailed quantitative and qualitative analyses, providing insights into the strengths and weaknesses of the approach.

Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pages 188–197
Copenhagen, Denmark, September 7–11, 2017.

2 Related Work

Machine learning methods have a long history in coreference resolution (see Ng (2010) for a detailed survey). However, the learning problem is challenging and, until very recently, hand-engineered systems built on top of automatically produced parse trees (Raghunathan et al., 2010) outperformed all learning approaches. Durrett and Klein (2013) showed that highly lexical learning approaches reverse this trend, and more recent neural models (Wiseman et al., 2016; Clark and Manning, 2016b,a) have achieved significant performance gains. However, all of these models still use parsers for head features and include highly engineered mention proposal algorithms.[1] Such pipelined systems suffer from two major drawbacks: (1) parsing mistakes can introduce cascading errors and (2) many of the hand-engineered rules do not generalize to new languages or domains. We present the first non-pipelined results, while providing further performance gains.

More generally, a wide variety of approaches for learning coreference models have been proposed. They can typically be categorized as (1) mention-pair classifiers (Ng and Cardie, 2002; Bengtson and Roth, 2008), (2) entity-level models (Haghighi and Klein, 2010; Clark and Manning, 2015, 2016b; Wiseman et al., 2016), (3) latent-tree models (Fernandes et al., 2012; Björkelund and Kuhn, 2014; Martschat and Strube, 2015), or (4) mention-ranking models (Durrett and Klein, 2013; Wiseman et al., 2015; Clark and Manning, 2016a). Our span-ranking approach is most similar to mention ranking, but we reason over a larger space by jointly detecting mentions and predicting coreference.

3 Task

We formulate the task of end-to-end coreference resolution as a set of decisions for every possible span in the document. The input is a document D containing T words along with metadata such as speaker and genre information.

Let $N = \frac{T(T+1)}{2}$ be the number of possible text spans in D. Denote the start and end indices of a span i in D respectively by START(i) and END(i), for $1 \leq i \leq N$. We assume an ordering of the spans based on START(i); spans with the same start index are ordered by END(i).

The task is to assign to each span i an antecedent y_i. The set of possible assignments for each y_i is $\mathcal{Y}(i) = \{\epsilon, 1, \ldots, i-1\}$, a dummy antecedent ϵ and all preceding spans. True antecedents of span i, i.e. span j such that $1 \leq j \leq i-1$, represent coreference links between i and j. The dummy antecedent ϵ represents two possible scenarios: (1) the span is not an entity mention or (2) the span is an entity mention but it is not coreferent with any previous span. These decisions implicitly define a final clustering, which can be recovered by grouping all spans that are connected by a set of antecedent predictions.

4 Model

We aim to learn a conditional probability distribution $P(y_1, \ldots, y_N \mid D)$ whose most likely configuration produces the correct clustering. We use a product of multinomials for each span:

$$P(y_1, \ldots, y_N \mid D) = \prod_{i=1}^{N} P(y_i \mid D)$$
$$= \prod_{i=1}^{N} \frac{\exp(s(i, y_i))}{\sum_{y' \in \mathcal{Y}(i)} \exp(s(i, y'))}$$

where $s(i, j)$ is a pairwise score for a coreference link between span i and span j in document D. We omit the document D from the notation when the context is unambiguous. There are three factors for this pairwise coreference score: (1) whether span i is a mention, (2) whether span j is a mention, and (3) whether j is an antecedent of i:

$$s(i, j) = \begin{cases} 0 & j = \epsilon \\ s_{\text{m}}(i) + s_{\text{m}}(j) + s_{\text{a}}(i, j) & j \neq \epsilon \end{cases}$$

Here $s_{\text{m}}(i)$ is a unary score for span i being a mention, and $s_{\text{a}}(i, j)$ is pairwise score for span j being an antecedent of span i.

By fixing the score of the dummy antecedent ϵ to 0, the model predicts the best scoring antecedent if any non-dummy scores are positive, and it abstains if they are all negative.

A challenging aspect of this model is that its size is $\mathcal{O}(T^4)$ in the document length. As we will see in Section 5, the above factoring enables aggressive pruning of spans that are unlikely to belong to a coreference cluster according the mention score $s_{\text{m}}(i)$.

[1] For example, Raghunathan et al. (2010) use rules to remove pleonastic mentions of *it* detected by 12 lexicalized regular expressions over English parse trees.

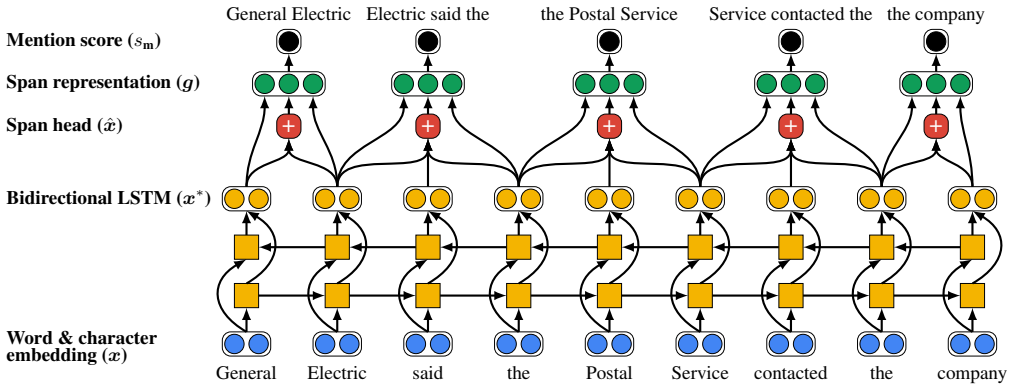

Figure 1: First step of the end-to-end coreference resolution model, which computes embedding representations of spans for scoring potential entity mentions. Low-scoring spans are pruned, so that only a manageable number of spans is considered for coreference decisions. In general, the model considers all possible spans up to a maximum width, but we depict here only a small subset.

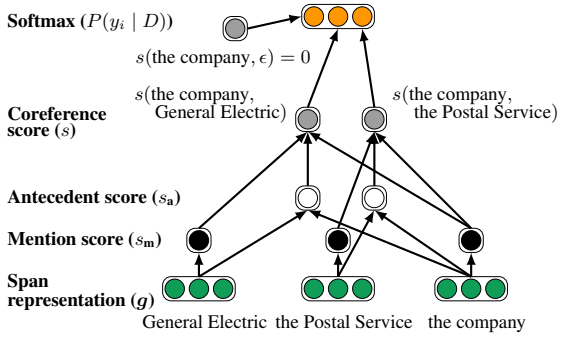

Figure 2: Second step of our model. Antecedent scores are computed from pairs of span representations. The final coreference score of a pair of spans is computed by summing the mention scores of both spans and their pairwise antecedent score.

Scoring Architecture We propose an end-to-end neural architecture that computes the above scores given the document and its metadata.

At the core of the model are vector representations $\boldsymbol{g}_i$ for each possible span i, which we describe in detail in the following section. Given these span representations, the scoring functions above are computed via standard feed-forward neural networks:

$$s_{\mathrm{m}}(i) = \boldsymbol{w}_{\mathrm{m}} \cdot \mathrm{FFNN}_{\mathrm{m}}(\boldsymbol{g}_i)$$
$$s_{\mathrm{a}}(i,j) = \boldsymbol{w}_{\mathrm{a}} \cdot \mathrm{FFNN}_{\mathrm{a}}([\boldsymbol{g}_i, \boldsymbol{g}_j, \boldsymbol{g}_i \circ \boldsymbol{g}_j, \phi(i,j)])$$

where $\cdot$ denotes the dot product, $\circ$ denotes element-wise multiplication, and FFNN denotes a feed-forward neural network that computes a non-linear mapping from input to output vectors.

The antecedent scoring function $s_{\mathrm{a}}(i,j)$ includes explicit element-wise similarity of each span $\boldsymbol{g}_i \circ \boldsymbol{g}_j$ and a feature vector $\phi(i,j)$ encoding speaker and genre information from the metadata and the distance between the two spans.

Span Representations Two types of information are crucial to accurately predicting coreference links: the context surrounding the mention span and the internal structure within the span. We use a bidirectional LSTM (Hochreiter and Schmidhuber, 1997) to encode the lexical information of both the inside and outside of each span. We also include an attention mechanism over words in each span to model head words.

We assume vector representations of each word $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T\}$, which are composed of fixed pre-trained word embeddings and 1-dimensional convolution neural networks (CNN) over characters (see Section 7.1 for details)

To compute vector representations of each span, we first use bidirectional LSTMs to encode every word in its context:

$$\begin{aligned}
\boldsymbol{f}_{t,\delta} &= \sigma(\mathbf{W}_{\mathrm{f}}[\boldsymbol{x}_t, \boldsymbol{h}_{t+\delta,\delta}] + \boldsymbol{b}_{\mathrm{i}}) \\
\boldsymbol{o}_{t,\delta} &= \sigma(\mathbf{W}_{\mathrm{o}}[\boldsymbol{x}_t, \boldsymbol{h}_{t+\delta,\delta}] + \boldsymbol{b}_{\mathrm{o}}) \\
\widetilde{\boldsymbol{c}}_{t,\delta} &= \tanh(\mathbf{W}_{\mathrm{c}}[\boldsymbol{x}_t, \boldsymbol{h}_{t+\delta,\delta}] + \boldsymbol{b}_{\mathrm{c}}) \\
\boldsymbol{c}_{t,\delta} &= \boldsymbol{f}_{t,\delta} \circ \widetilde{\boldsymbol{c}}_{t,\delta} + (\mathbf{1} - \boldsymbol{f}_{t,\delta}) \circ \boldsymbol{c}_{t+\delta,\delta} \\
\boldsymbol{h}_{t,\delta} &= \boldsymbol{o}_{t,\delta} \circ \tanh(\boldsymbol{c}_{t,\delta}) \\
\boldsymbol{x}^*_t &= [\boldsymbol{h}_{t,1}, \boldsymbol{h}_{t,-1}]
\end{aligned}$$

where $\delta \in \{-1, 1\}$ indicates the directionality of each LSTM, and $\boldsymbol{x}^*_t$ is the concatenated output of the bidirectional LSTM. We use independent LSTMs for every sentence, since cross-sentence context was not helpful in our experiments.

Syntactic heads are typically included as features in previous systems (Durrett and Klein, 2013; Clark and Manning, 2016b,a). Instead of relying on syntactic parses, our model learns a task-specific notion of headedness using an attention mechanism (Bahdanau et al., 2014) over words in each span:

$$\alpha_t = \boldsymbol{w}_\alpha \cdot \text{FFNN}_\alpha(\boldsymbol{x}_t^*)$$
$$a_{i,t} = \frac{\exp(\alpha_t)}{\sum_{k=\text{START}(i)}^{\text{END}(i)} \exp(\alpha_k)}$$
$$\hat{\boldsymbol{x}}_i = \sum_{t=\text{START}(i)}^{\text{END}(i)} a_{i,t} \cdot \boldsymbol{x}_t$$

where $\hat{\boldsymbol{x}}_i$ is a weighted sum of word vectors in span i. The weights $a_{i,t}$ are automatically learned and correlate strongly with traditional definitions of head words as we will see in Section 9.2.

The above span information is concatenated to produce the final representation $\boldsymbol{g}_i$ of span i:

$$\boldsymbol{g}_i = [\boldsymbol{x}^*_{\text{START}(i)}, \boldsymbol{x}^*_{\text{END}(i)}, \hat{\boldsymbol{x}}_i, \phi(i)]$$

This generalizes the recurrent span representations recently proposed for question-answering (Lee et al., 2016), which only include the boundary representations $\boldsymbol{x}^*_{\text{START}(i)}$ and $\boldsymbol{x}^*_{\text{END}(i)}$. We introduce the soft head word vector $\hat{\boldsymbol{x}}_i$ and a feature vector $\phi(i)$ encoding the size of span i.

5 Inference

The size of the full model described above is $\mathcal{O}(T^4)$ in the document length T. To maintain computation efficiency, we prune the candidate spans greedily during both training and evaluation.

We only consider spans with up to L words and compute their unary mention scores $s_\text{m}(i)$ (as defined in Section 4). To further reduce the number of spans to consider, we only keep up to λT spans with the highest mention scores and consider only up to K antecedents for each. We also enforce non-crossing bracketing structures with a simple suppression scheme.[2] We accept spans in decreasing order of the mention scores, unless, when considering span i, there exists a previously accepted span j such that $\text{START}(i) < \text{START}(j) \leq \text{END}(i) < \text{END}(j) \vee \text{START}(j) < \text{START}(i) \leq \text{END}(j) < \text{END}(i)$.

Despite these aggressive pruning strategies, we maintain a high recall of gold mentions in our experiments (over 92% when $\lambda = 0.4$).

For the remaining mentions, the joint distribution of antecedents for each document is computed in a forward pass over a single computation graph. The final prediction is the clustering produced by the most likely configuration.

6 Learning

In the training data, only clustering information is observed. Since the antecedents are latent, we optimize the marginal log-likelihood of all correct antecedents implied by the gold clustering:

$$\log \prod_{i=1}^{N} \sum_{\hat{y} \in \mathcal{Y}(i) \cap \text{GOLD}(i)} P(\hat{y})$$

where $\text{GOLD}(i)$ is the set of spans in the gold cluster containing span i. If span i does not belong to a gold cluster or all gold antecedents have been pruned, $\text{GOLD}(i) = \{\epsilon\}$.

By optimizing this objective, the model naturally learns to prune spans accurately. While the initial pruning is completely random, only gold mentions receive positive updates. The model can quickly leverage this learning signal for appropriate credit assignment to the different factors, such as the mention scores s_m used for pruning.

Fixing score of the dummy antecedent to zero removes a spurious degree of freedom in the overall model with respect to mention detection. It also prevents the span pruning from introducing noise. For example, consider the case where span i has a single gold antecedent that was pruned, so $\text{GOLD}(i) = \{\epsilon\}$. The learning objective will only correctly push the scores of non-gold antecedents lower, and it cannot incorrectly push the score of the dummy antecedent higher.

This learning objective can be considered a span-level, cost-insensitive analog of the learning objective proposed by Durrett and Klein (2013). We experimented with these cost-sensitive alternatives, including margin-based variants (Wiseman et al., 2015; Clark and Manning, 2016a), but a simple maximum-likelihood objective proved to be most effective.

[2] The official CoNLL-2012 evaluation only considers predictions without crossing mentions to be valid. Enforcing this consistency is not inherently necessary in our model.

	MUC			B^3			$CEAF_{\phi_4}$			
	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1	Avg. F1
Our model (ensemble)	**81.2**	**73.6**	**77.2**	**72.3**	**61.7**	**66.6**	**65.2**	**60.2**	**62.6**	**68.8**
Our model (single)	78.4	73.4	75.8	68.6	61.8	65.0	62.7	59.0	60.8	67.2
Clark and Manning (2016a)	79.2	70.4	74.6	69.9	58.0	63.4	63.5	55.5	59.2	65.7
Clark and Manning (2016b)	79.9	69.3	74.2	71.0	56.5	63.0	63.8	54.3	58.7	65.3
Wiseman et al. (2016)	77.5	69.8	73.4	66.8	57.0	61.5	62.1	53.9	57.7	64.2
Wiseman et al. (2015)	76.2	69.3	72.6	66.2	55.8	60.5	59.4	54.9	57.1	63.4
Clark and Manning (2015)	76.1	69.4	72.6	65.6	56.0	60.4	59.4	53.0	56.0	63.0
Martschat and Strube (2015)	76.7	68.1	72.2	66.1	54.2	59.6	59.5	52.3	55.7	62.5
Durrett and Klein (2014)	72.6	69.9	71.2	61.2	56.4	58.7	56.2	54.2	55.2	61.7
Björkelund and Kuhn (2014)	74.3	67.5	70.7	62.7	55.0	58.6	59.4	52.3	55.6	61.6
Durrett and Klein (2013)	72.9	65.9	69.2	63.6	52.5	57.5	54.3	54.4	54.3	60.3

Table 1: Results on the test set on the English data from the CoNLL-2012 shared task. The final column (Avg. F1) is the main evaluation metric, computed by averaging the F1 of MUC, B^3, and $CEAF_{\phi_4}$. We improve state-of-the-art performance by 1.5 F1 for the single model and by 3.1 F1.

7 Experiments

We use the English coreference resolution data from the CoNLL-2012 shared task (Pradhan et al., 2012) in our experiments. This dataset contains 2802 training documents, 343 development documents, and 348 test documents. The training documents contain on average 454 words and a maximum of 4009 words.

7.1 Hyperparameters

Word representations The word embeddings are a fixed concatenation of 300-dimensional GloVe embeddings (Pennington et al., 2014) and 50-dimensional embeddings from Turian et al. (2010), both normalized to be unit vectors. Out-of-vocabulary words are represented by a vector of zeros. In the character CNN, characters are represented as learned 8-dimensional embeddings. The convolutions have window sizes of 3, 4, and 5 characters, each consisting of 50 filters.

Hidden dimensions The hidden states in the LSTMs have 200 dimensions. Each feed-forward neural network consists of two hidden layers with 150 dimensions and rectified linear units (Nair and Hinton, 2010).

Feature encoding We encode speaker information as a binary feature indicating whether a pair of spans are from the same speaker. Following Clark and Manning (2016b), the distance features are binned into the following buckets [1, 2, 3, 4, 5-7, 8-15, 16-31, 32-63, 64+]. All features (speaker, genre, span distance, mention width) are represented as learned 20-dimensional embeddings.

Pruning We prune the spans such that the maximum span width $L = 10$, the number of spans per word $\lambda = 0.4$, and the maximum number of antecedents $K = 250$. During training, documents are randomly truncated to up to 50 sentences.

Learning We use ADAM (Kingma and Ba, 2014) for learning with a minibatch size of 1. The LSTM weights are initialized with random orthonormal matrices as described in Saxe et al. (2013). We apply 0.5 dropout to the word embeddings and character CNN outputs. We apply 0.2 dropout to all hidden layers and feature embeddings. Dropout masks are shared across timesteps to preserve long-distance information as described in Gal and Ghahramani (2016). The learning rate is decayed by 0.1% every 100 steps. The model is trained for up to 150 epochs, with early stopping based on the development set.

All code is implemented in TensorFlow (Abadi et al., 2015) and is publicly available. [3]

7.2 Ensembling

We also report ensemble experiments using five models trained with different random initializations. Ensembling is performed for both the span pruning and antecedent decisions.

At test time, we first average the mention scores $s_m(i)$ over each model before pruning the spans.

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

	Avg. F1	Δ
Our model (ensemble)	69.0	+1.3
Our model (single)	67.7	
– distance and width features	63.9	-3.8
– GloVe embeddings	65.3	-2.4
– speaker and genre metadata	66.3	-1.4
– head-finding attention	66.4	-1.3
– character CNN	66.8	-0.9
– Turian embeddings	66.9	-0.8

Table 2: Comparisons of our single model on the development data. The 5-model ensemble provides a 1.3 F1 improvement. The head-finding attention, features, and all word representations contribute significantly to the full model.

Given the same pruned spans, each model then computes the antecedent scores $s_a(i, j)$ separately, and they are averaged to produce the final scores.

8 Results

We report the precision, recall, and F1 for the standard MUC, B^3, and $CEAF_{\phi_4}$ metrics using the official CoNLL-2012 evaluation scripts. The main evaluation is the average F1 of the three metrics.

8.1 Coreference Results

Table 1 compares our model to several previous systems that have driven substantial improvements over the past several years on the OntoNotes benchmark. We outperform previous systems in all metrics. In particular, our single model improves the state-of-the-art average F1 by 1.5, and our 5-model ensemble improves it by 3.1.

The most significant gains come from improvements in recall, which is likely due to our end-to-end setup. During training, pipelined systems typically discard any mentions that the mention detector misses, which for Clark and Manning (2016a) consists of more than 9% of the labeled mentions in the training data. In contrast, we only discard mentions that exceed our maximum mention width of 10, which accounts for less than 2% of the training mentions. The contribution of joint mention scoring is further discussed in Section 8.3

8.2 Ablations

To show the importance of each component in our proposed model, we ablate various parts of the architecture and report the average F1 on the development set of the data (see Figure 2).

	Avg. F1	Δ
Our model (joint mention scoring)	67.7	
w/ rule-based mentions	66.7	-1.0
w/ oracle mentions	85.2	+17.5

Table 3: Comparisons of of various mention proposal methods with our model on the development data. The rule-based mentions are derived from the mention detector from Raghunathan et al. (2010), resulting in a 1 F1 drop in performance. The oracle mentions are from the labeled clusters and improve our model by over 17.5 F1.

Features The distance between spans and the width of spans are crucial signals for coreference resolution, consistent with previous findings from other coreference models. They contribute 3.8 F1 to the final result.

Word representations Since our word embeddings are fixed, having access to a variety of word embeddings allows for a more expressive model without overfitting. We hypothesis that the different learning objectives of the GloVe and Turian embeddings provide orthogonal information (the former is word-order insensitive while the latter is word-order sensitive). Both embeddings contribute to some improvement in development F1.

The character CNN provides morphological information and a way to backoff for out-of-vocabulary words. Since coreference decisions often involve rare named entities, we see a contribution of 0.9 F1 from character-level modeling.

Metadata Speaker and genre indicators many not be available in downstream applications. We show that performance degrades by 1.4 F1 without them, but is still on par with previous state-of-the-art systems that assume access to this metadata.

Head-finding attention Ablations also show a 1.3 F1 degradation in performance without the attention mechanism for finding task-specific heads. As we will see in Section 9.4, the attention mechanism should not be viewed as simply an approximation of syntactic heads. In many cases, it is beneficial to pay attention to multiple words that are useful specifically for coreference but are not traditionally considered to be syntactic heads.

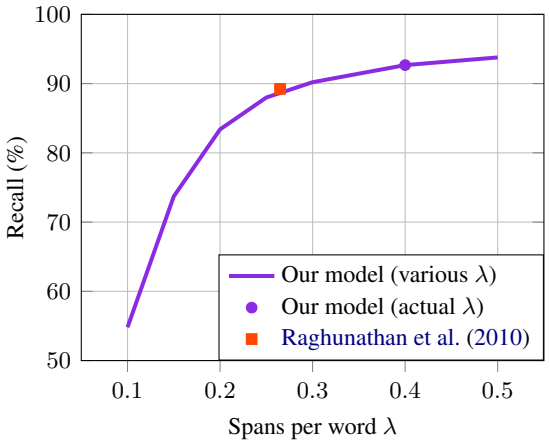

Figure 3: Proportion of gold mentions covered in the development data as we increase the number of spans kept per word. Recall is comparable to the mention detector of previous state-of-the-art systems given the same number of spans. Our model keeps 0.4 spans per word in our experiments, achieving 92.7% recall of gold mentions.

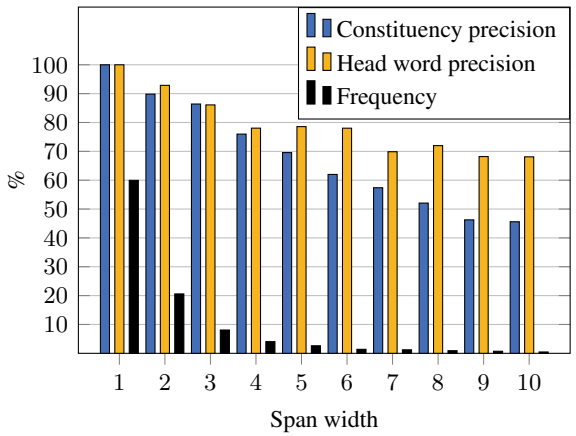

Figure 4: Indirect measure of mention precision using agreement with gold syntax. Constituency precision: % of unpruned spans matching syntactic constituents. Head word precision: % of unpruned constituents whose syntactic head word matches the most attended word. Frequency: % of gold spans with each width.

8.3 Comparing Span Pruning Strategies

To tease apart the contributions of improved mention scoring and improved coreference decisions, we compare the results of our model with alternate span pruning strategies. In these experiments, we use the alternate spans for both training and evaluation. As shown in Table 3, keeping mention candidates detected by the rule-based system over predicted parse trees (Raghunathan et al., 2010) degrades performance by 1 F1. We also provide oracle experiment results, where we keep exactly the mentions that are present in gold coreference clusters. With oracle mentions, we see an improvement of 17.5 F1, suggesting an enormous room for improvement if our model can produce better mention scores and anaphoricity decisions.

9 Analysis

To highlight the strengths and weaknesses of our model, we provide both quantitative and qualitative analyses. In the following discussion, we use predictions from the single model rather than the ensembled model.

9.1 Mention Recall

The training data only provides a weak signal for spans that correspond to entity mentions, since singleton clusters are not explicitly labeled. As a by product of optimizing marginal likelihood, our model automatically learns a useful ranking of spans via the unary mention scores from Section 4.

The top spans, according to the mention scores, cover a large portion of the mentions in gold clusters, as shown in Figure 3. Given a similar number of spans kept, our recall is comparable to the rule-based mention detector (Raghunathan et al., 2010) that produces 0.26 spans per word with a recall of 89.2%. As we increase the number of spans per word (λ in Section 5), we observe higher recall but with diminishing returns. In our experiments, keeping 0.4 spans per word results in 92.7% recall in the development data.

9.2 Mention Precision

While the training data does not offer a direct measure of mention precision, we can use the gold syntactic structures provided in the data as a proxy. Spans with high mention scores should correspond to syntactic constituents.

In Figure 4, we show the precision of top-scoring spans when keeping 0.4 spans per word. For spans with 2–5 words, 75–90% of the predictions are constituents, indicating that the vast majority of the mentions are syntactically plausible. Longer spans, which are all relatively rare, prove more difficult for the model, and precision drops to 46% for spans with 10 words.

1	**(A fire in a Bangladeshi garment factory)** has left at least 37 people dead and 100 hospitalized. Most of the deceased were killed in the crush as workers tried to flee **(the blaze)** in the four-story building.
	A fire in **(a Bangladeshi garment factory)** has left at least 37 people dead and 100 hospitalized. Most of the deceased were killed in the crush as workers tried to flee the blaze in **(the four-story building)**.
2	We are looking for **(a region of central Italy bordering the Adriatic Sea)**. **(The area)** is mostly mountainous and includes Mt. Corno, the highest peak of the Apennines. **(It)** also includes a lot of sheep, good clean-living, healthy sheep, and an Italian entrepreneur has an idea about how to make a little money of them.
3	**(The flight attendants)** have until 6:00 today to ratify labor concessions. **(The pilots')** union and ground crew did so yesterday.
4	**(Prince Charles and his new wife Camilla)** have jumped across the pond and are touring the United States making **(their)** first stop today in New York. It's Charles' first opportunity to showcase his new wife, but few Americans seem to care. Here's Jeanie Mowth. What a difference two decades make. **(Charles and Diana)** visited a JC Penney's on the prince's last official US tour. Twenty years later here's the prince with his new wife.
5	Also such location devices, **(some ships)** have smoke floats **(they)** can toss out so the man overboard will be able to use smoke signals as a way of trying to, let the rescuer locate **(them)**.

Table 4: Examples predictions from the development data. Each row depicts a single coreference cluster predicted by our model. Bold, parenthesized spans indicate mentions in the predicted cluster. The redness of each word indicates the weight of the head-finding attention mechanism ($a_{i,t}$ in Section 4).

9.3 Head Agreement

We also investigate how well the learned head preferences correlate with syntactic heads. For each of the top-scoring spans in the development data that correspond to gold constituents, we compute the word with the highest attention weight.

We plot in Figure 4 the proportion of these words that match syntactic heads. Agreement ranges between 68-93%, which is surprisingly high, since no explicit supervision of syntactic heads is provided. The model simply learns from the clustering data that these head words are useful for making coreference decisions.

9.4 Qualitative Analysis

Our qualitative analysis in Table 4 highlights the strengths and weaknesses of our model. Each row is a visualization of a single coreference cluster predicted by the model. Bolded spans in parentheses belong to the predicted cluster, and the redness of a word indicates its weight from the head-finding attention mechanism ($a_{i,t}$ in Section 4).

Strengths The effectiveness of the attention mechanism for making coreference decisions can be seen in Example 1. The model pays attention to *fire* in the span *A fire in a Bangladeshi garment factory*, allowing it to successfully predict the coreference link with *the blaze*. For a subspan of that mention, *a Bangladeshi garment factory*, the model pays most attention instead to *factory*, allowing it successfully predict the coreference link with *the four-story building*.

The task-specific nature of the attention mechanism is also illustrated in Example 4. The model generally pays attention to coordinators more than the content of the coordination, since coordinators, such as *and*, provide strong cues for plurality.

The model is capable of detecting relatively long and complex noun phrases, such as *a region of central Italy bordering the Adriatic Sea* in Example 2. It also appropriately pays attention to *region*, showing that the attention mechanism provides more than content-word classification. The context encoding provided by the bidirectional LSTMs is critical to making informative head word decisions.

Weaknesses A benefit of using neural models for coreference resolution is their ability to use word embeddings to capture similarity between words, a property that many traditional feature-based models lack. While this can dramatically increase recall, as demonstrated in Example 1, it is also prone to predicting false positive links when the model conflates paraphrasing with relatedness or similarity. In Example 3, the model mistakenly

predicts a link between *The flight attendants* and *The pilots'*. The predicted head words *attendants* and *pilots* likely have nearby word embeddings, which is a signal used—and often overused—by the model. The same type of error is made in Example 4, where the model predicts a coreference link between *Prince Charles and his new wife Camilla* and *Charles and Diana*, two non-coreferent mentions that are similar in many ways. These mistakes suggest substantial room for improvement with word or span representations that can cleanly distinguish between equivalence, entailment, and alternation.

Unsurprisingly, our model does little in the uphill battle of making coreference decisions requiring world knowledge. In Example 5, the model incorrectly decides that *them* (in the context of *let the rescuer locate them*) is coreferent with *some ships*, likely due to plurality cues. However, an ideal model that uses common-sense reasoning would instead correctly infer that a rescuer is more likely to look for *the man overboard* rather than the ship from which he fell. This type of reasoning would require either (1) models that integrate external sources of knowledge with more complex inference or (2) a vastly larger corpus of training data to overcome the sparsity of these patterns.

10 Conclusion

We presented a state-of-the-art coreference resolution model that is trained end-to-end for the first time. Our final model ensemble improves performance on the OntoNotes benchmark by over 3 F1 without external preprocessing tools used by previous systems. We showed that our model implicitly learns to generate useful mention candidates from the space of all possible spans. A novel head-finding attention mechanism also learns a task-specific preference for head words, which we empirically showed correlate strongly with traditional head-word definitions.

While our model substantially pushes the state-of-the-art performance, the improvements are potentially complementary to a large body of work on various strategies to improve coreference resolution, including entity-level inference and incorporating world knowledge, which are important avenues for future work.

Acknowledgements

The research was supported in part by DARPA under the DEFT program (FA8750-13-2-0019), the ARO (W911NF-16-1-0121), the NSF (IIS-1252835, IIS-1562364), gifts from Google and Tencent, and an Allen Distinguished Investigator Award. We also thank the UW NLP group for helpful conversations and comments on the work.

References

Martın Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, et al. 2015. TensorFlow: Large-scale Machine Learning on Heterogeneous Systems. *Software available from tensorflow.org*.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

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 *ACL*.

Kevin Clark and Christopher D. Manning. 2015. Entity-centric coreference resolution with model stacking. In *ACL*.

Kevin Clark and Christopher D. Manning. 2016a. Deep reinforcement learning for mention-ranking coreference models. In *Empirical Methods on Natural Language Processing (EMNLP)*.

Kevin Clark and Christopher D. Manning. 2016b. Improving coreference resolution by learning entity-level distributed representations. In *Association for Computational Linguistics (ACL)*.

Greg Durrett and Dan Klein. 2013. Easy victories and uphill battles in coreference resolution. In *EMNLP*.

Greg Durrett and Dan Klein. 2014. A joint model for entity analysis: Coreference, typing, and linking. *TACL*, 2:477–490.

Eraldo Rezende Fernandes, Cícero Nogueira Dos Santos, and Ruy Luiz Milidiú. 2012. Latent structure perceptron with feature induction for unrestricted coreference resolution. In *Joint Conference on EMNLP and CoNLL-Shared Task*, pages 41–48. Association for Computational Linguistics.

Yarin Gal and Zoubin Ghahramani. 2016. A theoretically grounded application of dropout in recurrent neural networks. In *Advances in Neural Information Processing Systems*, pages 1019–1027.

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. Association for Computational Linguistics.

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

Diederik Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. *arXiv preprint arXiv:1412.6980*.

Kenton Lee, Shimi Salant, Tom Kwiatkowski, Ankur Parikh, Dipanjan Das, and Jonathan Berant. 2016. Learning recurrent span representations for extractive question answering. *arXiv preprint arXiv:1611.01436*.

Sebastian Martschat and Michael Strube. 2015. Latent structures for coreference resolution. *Transactions of the Association for Computational Linguistics*, 3:405–418.

Vinod Nair and Geoffrey E Hinton. 2010. Rectified linear units improve restricted boltzmann machines. In *Proceedings of ICML*.

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.

Vincent Ng and Claire Cardie. 2002. Identifying anaphoric and non-anaphoric noun phrases to improve coreference resolution. In *Proceedings of the 19th international conference on Computational linguistics-Volume 1*, pages 1–7. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of EMNLP*.

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 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. Association for Computational Linguistics.

Andrew M Saxe, James L McClelland, and Surya Ganguli. 2013. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. *arXiv preprint arXiv:1312.6120*.

Joseph Turian, Lev Ratinov, and Yoshua Bengio. 2010. Word representations: A Simple and General Method for Semi-supervised Learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*.

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

Sam Wiseman, Alexander M. Rush, Stuart M. Shieber, and Jason Weston. 2015. Learning anaphoricity and antecedent ranking features for coreference resolution. In *ACL*.