Learning Global Features for Coreference Resolution

Sam Wiseman and **Alexander M. Rush** and **Stuart M. Shieber**
School of Engineering and Applied Sciences
Harvard University
Cambridge, MA, USA
{swiseman,srush,shieber}@seas.harvard.edu

Abstract

There is compelling evidence that coreference prediction would benefit from modeling global information about entity-clusters. Yet, state-of-the-art performance can be achieved with systems treating each mention prediction independently, which we attribute to the inherent difficulty of crafting informative cluster-level features. We instead propose to use recurrent neural networks (RNNs) to learn latent, global representations of entity clusters directly from their mentions. We show that such representations are especially useful for the prediction of pronominal mentions, and can be incorporated into an end-to-end coreference system that outperforms the state of the art without requiring any additional search.

1 Introduction

While structured, non-local coreference models would seem to hold promise for avoiding many common coreference errors (as discussed further in Section 3), the results of employing such models in practice are decidedly mixed, and state-of-the-art results can be obtained using a completely local, mention-ranking system.

In this work, we posit that global context is indeed necessary for further improvements in coreference resolution, but argue that informative *cluster*, rather than mention, level features are very difficult to devise, limiting their effectiveness. Accordingly, we instead propose to learn representations of mention clusters by embedding them sequentially using a recurrent neural network (shown in Section 4). Our model has no manually defined cluster features, but instead learns a global representation from the individual mentions present in each cluster. We incorporate these representations into a mention-ranking coreference system.

The entire model, including the recurrent neural network and the mention-ranking system, is trained end-to-end on the coreference task. We train the model as a local classifier with fixed context (that is, as a history-based model). As such, unlike several recent approaches, which may require complicated inference during training, we are able to train our model in much the same way as a vanilla mention-ranking model.

Experiments compare the use of learned global features to several strong baseline systems for coreference resolution. We demonstrate that the learned global representations capture important underlying information that can help resolve difficult pronominal mentions, which remain a persistent source of errors for modern coreference systems (Durrett and Klein, 2013; Kummerfeld and Klein, 2013; Wiseman et al., 2015; Martschat and Strube, 2015). Our final system improves over 0.8 points in CoNLL score over the current state of the art, and the improvement is statistically significant on all three CoNLL metrics.

2 Background and Notation

Coreference resolution is fundamentally a clustering task. Given a sequence $(x_n)_{n=1}^N$ of (intra-document) mentions – that is, syntactic units that can refer or be referred to – coreference resolution involves partitioning (x_n) into a sequence of clusters $(X^{(m)})_{m=1}^M$ such that all the mentions in any particular cluster

$X^{(m)}$ refer to the same underlying entity. Since the mentions within a particular cluster may be ordered linearly by their appearance in the document,[1] we will use the notation $X_j^{(m)}$ to refer to the j'th mention in the m'th cluster.

A valid clustering places each mention in exactly one cluster, and so we may represent a clustering with a vector $z \in \{1, \ldots, M\}^N$, where $z_n = m$ iff x_n is a member of $X^{(m)}$. Coreference systems attempt to find the best clustering $z^* \in \mathcal{Z}$ under some scoring function, with $\mathcal{Z}$ the set of valid clusterings.

One strategy to avoid the computational intractability associated with predicting an entire clustering z is to instead predict a single *antecedent* for each mention x_n; because x_n may not be anaphoric (and therefore have no antecedents), a "dummy" antecedent ϵ may also be predicted. The aforementioned strategy is adopted by "mention-ranking" systems (Denis and Baldridge, 2008; Rahman and Ng, 2009; Durrett and Klein, 2013), which, formally, predict an antecedent $\hat{y}_n \in \mathcal{Y}(x_n)$ for each mention x_n, where $\mathcal{Y}(x_n) = \{1, \ldots, n-1, \epsilon\}$. Through transitivity, these decisions induce a clustering over the document.

Mention-ranking systems make their antecedent predictions with a *local* scoring function $f(x_n, y_n)$ defined for any mention x_n and any antecedent $y_n \in \mathcal{Y}(x_n)$. While such a scoring function clearly ignores much structural information, the mention-ranking approach has been attractive for at least two reasons. First, inference is relatively simple and efficient, requiring only a left-to-right pass through a document's mentions during which a mention's antecedents (as well as ϵ) are scored and the highest scoring antecedent is predicted. Second, from a linguistic modeling perspective, mention-ranking models learn a scoring function that requires a mention x_n to be compatible with only *one* of its coreferent antecedents. This contrasts with mention-pair models (e.g., Bengtson and Roth (2008)), which score all pairs of mentions in a cluster, as well as with certain cluster-based models (see discussion in Culotta et al. (2007)). Modeling each mention as having a single antecedent is particularly advantageous for pronominal mentions, which we might like to model

---

[1]We assume nested mentions are ordered by their syntactic heads.

as linking to a single nominal or proper antecedent, for example, but not necessarily to all other coreferent mentions.

Accordingly, in this paper we attempt to maintain the inferential simplicity and modeling benefits of mention ranking, while allowing the model to utilize global, structural information relating to z in making its predictions. We therefore investigate objective functions of the form

$$\arg\max_{y_1, \ldots, y_N} \sum_{n=1}^{N} f(x_n, y_n) + g(x_n, y_n, z_{1:n-1}) \quad ,$$

where g is a global function that, in making predictions for x_n, may examine (features of) the clustering $z_{1:n-1}$ induced by the antecedent predictions made through y_{n-1}.

3 The Role of Global Features

Here we motivate the use of global features for coreference resolution by focusing on the issues that may arise when resolving pronominal mentions in a purely local way. See Clark and Manning (2015) and Stoyanov and Eisner (2012) for more general motivation for using global models.

3.1 Pronoun Problems

Recent empirical work has shown that the resolution of pronominal mentions accounts for a substantial percentage of the total errors made by modern mention-ranking systems. Wiseman et al. (2015) show that on the CoNLL 2012 English development set, almost 59% of mention-ranking precision errors and almost 24% of recall errors involve pronominal mentions. Martschat and Strube (2015) found a similar pattern in their comparison of mention-ranking, mention-pair, and latent-tree models.

To see why pronouns can be so problematic, consider the following passage from the "Broadcast Conversation" portion of the CoNLL development set (bc_msnbc_0000_018); below, we enclose mentions in brackets and give the same subscript to co-clustered mentions. (This example is also shown in Figure 2.)

> **DA:** um and [I]₁ think that is what's - Go ahead [Linda]₂.
> **LW:** Well and uh thanks goes to [you]₁ and to [the media]₃ to help [us]₄...So [our]₄ hat is off to all of [you]₅ as well.

This example is typical of Broadcast Conversation, and it is difficult because local systems learn to myopically link pronouns such as $[\text{you}]_5$ to other instances of the same pronoun that are close by, such as $[\text{you}]_1$. While this is often a reasonable strategy, in this case predicting $[\text{you}]_1$ to be an antecedent of $[\text{you}]_5$ would result in the prediction of an incoherent cluster, since $[\text{you}]_1$ is coreferent with the singular $[\text{I}]_1$, and $[\text{you}]_5$, as part of the phrase "all of you," is evidently plural. Thus, while there is enough information in the text to correctly predict $[\text{you}]_5$, doing so crucially depends on having access to the *history* of predictions made so far, and it is precisely this access to history that local models lack.

More empirically, there are non-local statistical regularities involving pronouns we might hope models could exploit. For instance, in the CoNLL training data over 70% of pleonastic "it" instances and over 74% of pleonastic "you" instances follow (respectively) previous pleonastic "it" and "you" instances. Similarly, over 78% of referential "I" instances and over 68% of referential "he" instances corefer with previous "I" and "he" instances, respectively.

Accordingly, we might expect non-local models with access to global features to perform significantly better. However, models incorporating non-local features have a rather mixed track record. For instance, Björkelund and Kuhn (2014) found that cluster-level features improved their results, whereas Martschat and Strube (2015) found that they did not. Clark and Manning (2015) found that incorporating cluster-level features *beyond* the pre-computed mention-pair and mention-ranking probabilities that form the basis of their agglomerative clustering coreference system did not improve performance. Furthermore, among recent, state-of-the-art systems, mention-ranking systems (which are completely local) perform at least as well as their more structured counterparts (Durrett and Klein, 2014; Clark and Manning, 2015; Wiseman et al., 2015; Peng et al., 2015).

3.2 Issues with Global Features

We believe a major reason for the relative ineffectiveness of global features in coreference problems is that, as noted by Clark and Manning (2015), cluster-level features can be hard to define. Specif-

ically, it is difficult to define discrete, fixed-length features on clusters, which can be of variable size (or shape). As a result, global coreference features tend to be either too coarse or too sparse. Thus, early attempts at defining cluster-level features simply applied the coarse quantifier predicates *all*, *none*, and *most* to the mention-level features defined on the mentions in a cluster (Culotta et al., 2007; Rahman and Ng, 2011). For example, a cluster would have the feature 'all-female=true' if all mentions in the cluster have a 'female=true' feature.

On the other extreme, Björkelund and Kuhn (2014) define certain cluster-level features by concatenating the mention-level features of a cluster's constituent mentions in order of the mentions' appearance in the document. For example, if a cluster consists, in order, of the mentions (*the president*, *he*, *he*), they would define a cluster-level "type" feature 'C-P-P=true', which indicates that the cluster is composed, in order, of a common noun, a pronoun, and a pronoun. While very expressive, these concatenated features are often quite sparse, since clusters encountered during training can be of any size.

4 Learning Global Features

To circumvent the aforementioned issues with defining global features, we propose to learn cluster-level feature representations implicitly, by identifying the state of a (partial) cluster with the hidden state of an RNN that has consumed the sequence of mentions composing the (partial) cluster. Before providing technical details, we provide some preliminary evidence that such learned representations capture important contextual information by displaying in Figure 1 the learned final states of all clusters in the CoNLL development set, projected using T-SNE (van der Maaten and Hinton, 2012). Each point in the visualization represents the learned features for an entity cluster and the head words of mentions are shown for representative points. Note that the model learns to roughly separate clusters by simple distinctions such as predominant type (nominal, proper, pronominal) and number (it, they, etc), but also captures more subtle relationships such as grouping geographic terms and long strings of pronouns.

Figure 1: T-SNE visualization of learned entity representations on the CoNLL development set (best viewed in color). Each point shows a gold cluster of size > 1. Yellow, red, and purple points represent predominantly common noun, proper noun, and pronoun clusters respectively. Captions show the head word of the mentions in a representative cluster.

4.1 Recurrent Neural Networks

A recurrent neural network is a parameterized non-linear function $\mathbf{RNN}$ that recursively maps an input sequence of vectors to a sequence of hidden states. Let $(\boldsymbol{m}_j)_{j=1}^J$ be a sequence of J input vectors $\boldsymbol{m}_j \in \mathbb{R}^D$, and let $\boldsymbol{h}_0 = \mathbf{0}$. Applying an RNN to any such sequence yields

$$\boldsymbol{h}_j \leftarrow \mathbf{RNN}(\boldsymbol{m}_j, \boldsymbol{h}_{j-1}; \boldsymbol{\theta}) \qquad ,$$

where $\boldsymbol{\theta}$ is the set of parameters for the model, which are shared over time.

There are several varieties of RNN, but by far the most commonly used in natural-language processing is the Long Short-Term Memory network (LSTM) (Hochreiter and Schmidhuber, 1997), particularly for language modeling (e.g., Zaremba et al. (2014)) and machine translation (e.g., Sutskever et al. (2014)), and we use LSTMs in all experiments.

4.2 RNNs for Cluster Features

Our main contribution will be to utilize RNNs to produce feature representations of entity clusters which will provide the basis of the global term g. Recall that we view a cluster $X^{(m)}$ as a sequence of mentions $(X_j^{(m)})_{j=1}^J$ (ordered in linear document order). We therefore propose to embed the state(s) of

$X^{(m)}$ by running an RNN over the cluster in order.

In order to run an RNN over the mentions we need an embedding function $\boldsymbol{h}_c$ to map a mention to a real vector. First, following Wiseman et al. (2015) define $\boldsymbol{\phi}_a(x) : \mathcal{X} \to \{0, 1\}^F$ as a standard set of local indicator features on a mention, such as its head word, its gender, and so on. (We elaborate on features below.) We then use a non-linear feature embedding $\boldsymbol{h}_c$ to map a mention x to a vector-space representation. In particular, we define

$$\boldsymbol{h}_c(x) \triangleq \tanh(\boldsymbol{W}_c \, \boldsymbol{\phi}_a(x) + \boldsymbol{b}_c) \qquad ,$$

where $\boldsymbol{W}_c$ and $\boldsymbol{b}_c$ are parameters of the embedding. We will refer to the j'th hidden state of the RNN corresponding to $X^{(m)}$ as $\boldsymbol{h}_j^{(m)}$, and we obtain it according to the following formula

$$\boldsymbol{h}_j^{(m)} \leftarrow \mathbf{RNN}(\boldsymbol{h}_c(X_j^{(m)}), \boldsymbol{h}_{j-1}^{(m)}; \boldsymbol{\theta}) \qquad ,$$

again assuming that $\boldsymbol{h}_0^{(m)} = \mathbf{0}$. Thus, we will effectively run an RNN over each (sequence of mentions corresponding to a) cluster $X^{(m)}$ in the document, and thereby generate a hidden state $\boldsymbol{h}_j^{(m)}$ corresponding to each step of each cluster in the document. Concretely, this can be implemented by maintaining M RNNs – one for each cluster – that all share the parameters $\boldsymbol{\theta}$. The process is illustrated in the top portion of Figure 2.

5 Coreference with Global Features

We now describe how the RNN defined above is used within an end-to-end coreference system.

5.1 Full Model and Training

Recall that our inference objective is to maximize the score of both a local mention ranking term as well as a global term based on the current clusters:

$$\arg\max_{y_1, \ldots, y_N} \sum_{n=1}^N f(x_n, y_n) + g(x_n, y_n, \boldsymbol{z}_{1:n-1})$$

We begin by defining the local model $f(x, y)$ with the two layer neural network of Wiseman et al. (2015), which has a specialization for the non-anaphoric case, as follows:

$$f(x, y) \triangleq \begin{cases} \boldsymbol{u}^\top \begin{bmatrix} \boldsymbol{h}_a(x) \\ \boldsymbol{h}_{p(x,y)} \end{bmatrix} + u_0 & \text{if } y \neq \epsilon \\ \boldsymbol{v}^\top \boldsymbol{h}_a(x) + v_0 & \text{if } y = \epsilon \end{cases} \qquad .$$

DA: um and [I]$_1$ think that is what's - Go ahead [Linda]$_2$.

LW: Well and thanks goes to [you]$_1$ and to [the media]$_3$ to help [us]$_4$...So [our]$_4$ hat is off to all of [you]$_5$...

$$[\text{I}],\ \boldsymbol{h}_2^{(1)}\quad [\text{Linda}],\ \boldsymbol{h}_1^{(2)}\quad [\text{you}],\ \boldsymbol{h}_2^{(1)}\quad [\text{the media}],\ \boldsymbol{h}_1^{(3)}\quad [\text{us}],\ \boldsymbol{h}_2^{(4)}\quad [\text{our}],\ \boldsymbol{h}_2^{(4)}\quad x_n = [\text{you}]\quad \epsilon,\ \text{NA}(x_n)$$
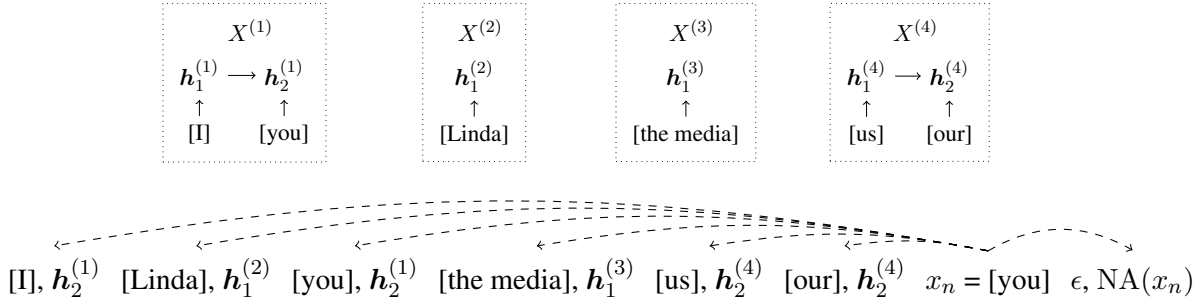
Figure 2: Full RNN example for handling the mention $x_n = $ [you]. There are currently four entity clusters in scope $X^{(1)}, X^{(2)}, X^{(3)}, X^{(4)}$ based on unseen previous decisions (y). Each cluster has a corresponding RNN state, two of which ($\boldsymbol{h}^{(1)}$ and $\boldsymbol{h}^{(4)}$) have processed multiple mentions (with $X^{(1)}$ notably including a singular mention [I]). At the bottom, we show the complete mention-ranking process. Each previous mention is considered as an antecedent, and the global term considers the antecedent clusters' current hidden state. Selecting ϵ is treated with a special case $\text{NA}(x_n)$.

Above, $\boldsymbol{u}$ and $\boldsymbol{v}$ are the parameters of the model, and $\boldsymbol{h}_\text{a}$ and $\boldsymbol{h}_\text{p}$ are learned feature embeddings of the local mention context and the pairwise affinity between a mention and an antecedent, respectively. These feature embeddings are defined similarly to $\boldsymbol{h}_\text{c}$, as

$$\boldsymbol{h}_\text{a}(x) \triangleq \tanh(\boldsymbol{W}_\text{a}\,\boldsymbol{\phi}_\text{a}(x) + \boldsymbol{b}_\text{a})$$
$$\boldsymbol{h}_\text{p}(x,y) \triangleq \tanh(\boldsymbol{W}_\text{p}\,\boldsymbol{\phi}_\text{p}(x,y) + \boldsymbol{b}_\text{p})\quad,$$

where ϕ_a (mentioned above) and ϕ_p are "raw" (that is, unconjoined) features on the context of x and on pairwise affinity between mentions x and y, respectively (Wiseman et al., 2015). Note that $\boldsymbol{h}_\text{a}$ and $\boldsymbol{h}_\text{c}$ use the same raw features; only their weights differ.

We now specify our global scoring function g based on the history of previous decisions. Define $\boldsymbol{h}_{<n}^{(m)}$ as the hidden state of cluster m before a decision is made for x_n – that is, $\boldsymbol{h}_{<n}^{(m)}$ is the state of cluster m's RNN after it has consumed all mentions in the cluster *preceding* x_n. We define g as

$$g(x_n, y_n, \boldsymbol{z}_{1:n-1}) \triangleq \begin{cases} \boldsymbol{h}_\text{c}(x_n)^\mathsf{T}\boldsymbol{h}_{<n}^{(z_{y_n})} & \text{if } y \neq \epsilon \\ \text{NA}(x_n) & \text{if } y = \epsilon \end{cases},$$

where NA gives a score for assigning ϵ based on a non-linear function of all of the current hidden states:

$$\text{NA}(x_n) = \boldsymbol{q}^\mathsf{T} \tanh\left(\boldsymbol{W}_s \begin{bmatrix} \phi_\text{a}(x) \\ \sum_{m=1}^{M} \boldsymbol{h}_{<n}^{(m)} \end{bmatrix} + \boldsymbol{b}_s\right).$$

See Figure 2 for a diagram. The intuition behind the first case in g is that in considering whether y_n is a

good antecedent for x_n, we add a term to the score that examines how well x_n matches with the mentions already in $X^{(z_{y_n})}$; this matching score is expressed via a dot-product.[2] In the second case, when predicting that x_n is non-anaphoric, we add the NA term to the score, which examines the (sum of) the current states $\boldsymbol{h}_{<n}^{(m)}$ of all clusters. This information is useful both because it allows the non-anaphoric score to incorporate information about potential antecedents, and because the occurrence of certain singleton-clusters often predicts the occurrence of future singleton-clusters, as noted in Section 3.

The whole system is trained end-to-end on coreference using backpropagation. For a given training document, let $\boldsymbol{z}^{(o)}$ be the oracle mapping from mention to cluster, which induces an oracle clustering. While at training time we do have oracle clusters, we do not have oracle antecedents (y_n), so following past work we treat the oracle antecedent as latent (Yu and Joachims, 2009; Fernandes et al., 2012; Chang et al., 2013; Durrett and Klein, 2013). We train with the following slack-rescaled, margin objective:

$$\sum_{n=1}^{N} \max_{y \in \mathcal{Y}(x_n)} \Delta(x_n, y)(1 + f(x_n, y) + g(x_n, y, \boldsymbol{z}^{(o)})$$
$$- f(x_n, y_n^\ell) - g(x_n, y_n^\ell, \boldsymbol{z}^{(o)})),$$

where the latent antecedent y_n^ℓ is defined as

---

[2]We also experimented with other non-linear functions, but dot-products performed best.

$$y_n^\ell \triangleq \argmax_{y \in \mathcal{Y}(x_n): z_y^{(o)} = z_n^{(o)}} f(x_n, y) + g(x_n, y, \boldsymbol{z}^{(o)})$$

if x_n is anaphoric, and is ϵ otherwise. The term $\Delta(x_n, y)$ gives different weight to different error types. We use a Δ with 3 different weights $(\alpha_1, \alpha_2, \alpha_3)$ for "false link" (FL), "false new" (FN), and "wrong link" (WL) mistakes (Durrett and Klein, 2013), which correspond to predicting an antecedent when non-anaphoric, ϵ when anaphoric, and the wrong antecedent, respectively.

Note that in training we use the oracle clusters $\boldsymbol{z}^{(o)}$. Since these are known a priori, we can precompute all the hidden states $\boldsymbol{h}_j^{(m)}$ in a document, which makes training quite simple and efficient. This approach contrasts in particular with the work of Björkelund and Kuhn (2014) — who also incorporate global information in mention-ranking — in that they train against latent *trees*, which are not annotated and must be searched for during training. On the other hand, training on oracle clusters leads to a mismatch between training and test, which can hurt performance.

5.2 Search

When moving from a strictly local objective to one with global features, the test-time search problem becomes intractable. The local objective requires $O(n^2)$ time, whereas the full clustering problem is NP-Hard. Past work with global features has used integer linear programming solvers for exact search (Chang et al., 2013; Peng et al., 2015), or beam search with (delayed) early update training for an approximate solution (Björkelund and Kuhn, 2014). In contrast, we simply use greedy search at test time, which also requires $O(n^2)$ time.[3] The full algorithm is shown in Algorithm 1. The greedy search algorithm is identical to a simple mention-ranking system, with the exception of line 11, which updates

---

[3]While beam search is a natural way to decrease search error at test time, it may fail to help if training involves a *local* margin objective (as in our case), since scores need not be calibrated across local decisions. We accordingly attempted to train various locally normalized versions of our model, but found that they underperformed. We also experimented with training approaches and model variants that expose the model to its own predictions (Daumé III et al., 2009; Ross et al., 2011; Bengio et al., 2015), but found that these yielded a negligible performance improvement.

Algorithm 1 Greedy search with global RNNs

1: **procedure** GREEDYCLUSTER($x_1, \dots, x_N$)
2:     Initialize clusters $X^{(1)} \dots$ as empty lists, hidden states $\boldsymbol{h}^{(0)}, \dots$ as $\mathbf{0}$ vectors in $\mathbb{R}^D$, $\boldsymbol{z}$ as map from mention to cluster, and cluster counter $M \leftarrow 0$
3:     **for** $n = 2 \dots N$ **do**
4:       $y^* \leftarrow \argmax_{y \in \mathcal{Y}(x_n)} f(x_n, y) + g(x_n, y_n, \boldsymbol{z}_{1:n})$
5:       $m \leftarrow z_{y^*}$
6:       **if** $y^* = \epsilon$ **then**
7:         $M \leftarrow M + 1$
8:         $m \leftarrow M$
9:       append x_n to $X^{(m)}$
10:      $z_n \leftarrow m$
11:      $\boldsymbol{h}^{(m)} \leftarrow \mathbf{RNN}(\boldsymbol{h}_c(x_n), \boldsymbol{h}^{(m)})$
12:     **return** $X^{(1)}, \dots, X^{(M)}$

---

the current RNN representation based on the previous decision that was made, and line 4, which then uses this cluster representation as part of scoring.

6 Experiments

6.1 Methods

We run experiments on the CoNLL 2012 English shared task (Pradhan et al., 2012). The task uses the OntoNotes corpus (Hovy et al., 2006), consisting of 3,493 documents in various domains and formats. We use the experimental split provided in the shared task. For all experiments, we use the Berkeley Coreference System (Durrett and Klein, 2013) for mention extraction and to compute features ϕ_a and ϕ_p.

Features We use the raw BASIC+ feature sets described by Wiseman et al. (2015), with the following modifications:

- We remove all features from ϕ_p that concatenate a feature of the antecedent with a feature of the current mention, such as bi-head features.

- We add true-cased head features, a current speaker indicator feature, and a 2-character genre (out of {bc,bn,mz,nw,pt,tc,wb}) indicator to ϕ_p and ϕ_a.

- We add features indicating if a mention has a substring overlap with the current speaker (ϕ_p and ϕ_a), and if an antecedent has a substring overlap with a speaker distinct from the current mention's speaker (ϕ_p).

System	MUC			B^3			CEAF$_e$			CoNLL
	P	R	F$_1$	P	R	F$_1$	P	R	F$_1$	
B&K (2014)	74.3	67.46	70.72	62.71	54.96	58.58	59.4	52.27	55.61	61.63
M&S (2015)	76.72	68.13	72.17	66.12	54.22	59.58	59.47	52.33	55.67	62.47
C&M (2015)	76.12	69.38	72.59	65.64	56.01	60.44	59.44	52.98	56.02	63.02
Peng et al. (2015)	-	-	72.22	-	-	60.50	-	-	56.37	63.03
Wiseman et al. (2015)	76.23	69.31	72.60	66.07	55.83	60.52	59.41	54.88	57.05	63.39
This work	77.49	69.75	**73.42**	66.83	56.95	**61.50**	62.14	53.85	**57.70**	**64.21**

Table 1: Results on CoNLL 2012 English test set. We compare against recent state of the art systems, including (in order) Bjorkelund and Kuhn (2014), Martschat and Strube (2015), Clark and Manning (2015), Peng et al. (2015), and Wiseman et al. (2015). F$_1$ gains are significant ($p < 0.05$ under the bootstrap resample test (Koehn, 2004)) compared with Wiseman et al. (2015) for all metrics.

- We add a single centered, rescaled document position feature to each mention when learning h_c. We calculate a mention x_n's rescaled document position as $\frac{2n-N-1}{N-1}$.

These modifications result in there being approximately 14K distinct features in ϕ_a and approximately 28K distinct features in ϕ_p, which is far fewer features than has been typical in past work.

For training, we use document-size minibatches, which allows for efficient pre-computation of RNN states, and we minimize the loss described in Section 5 with AdaGrad (Duchi et al., 2011) (after clipping LSTM gradients to lie (elementwise) in $(-10, 10)$). We find that the initial learning rate chosen for AdaGrad has a significant impact on results, and we choose learning rates for each layer out of $\{0.1, 0.02, 0.01, 0.002, 0.001\}$.

In experiments, we set $h_a(x)$, $h_c(x)$, and $h^{(m)}$ to be $\in \mathbb{R}^{200}$, and $h_p(x, y) \in \mathbb{R}^{700}$. We use a single-layer LSTM (without "peep-hole" connections), as implemented in the `element-rnn` library (Léonard et al., 2015). For regularization, we apply Dropout (Srivastava et al., 2014) with a rate of 0.4 before applying the linear weights u, and we also apply Dropout with a rate of 0.3 to the LSTM states before forming the dot-product scores. Following Wiseman et al. (2015) we use the cost-weights $\alpha = \langle 0.5, 1.2, 1 \rangle$ in defining Δ, and we use their pre-training scheme as well. For final results, we train on both training and development portions of the CoNLL data. Scoring uses the official CoNLL 2012 script (Pradhan et al., 2014; Luo et al., 2014). Code for our system is available at `https://github.com/swiseman/nn_coref`. The

	MUC	B^3	CEAF$_e$	CoNLL
MR	73.06	62.66	58.98	64.90
Avg, OH	73.30	63.06	58.85	65.07
RNN, GH	73.63	63.23	59.56	65.47
RNN, OH	74.26	63.89	59.54	65.90

Table 2: F$_1$ scores of models described in text on CoNLL 2012 development set. Rows in grey highlight models using oracle history.

system makes use of a GPU for training, and trains in about two hours.

6.2 Results

In Table 1 we present our main results on the CoNLL English test set, and compare with other recent state-of-the-art systems. We see a statistically significant improvement of over 0.8 CoNLL points over the previous state of the art, and the highest F$_1$ scores to date on all three CoNLL metrics.

We now consider in more detail the impact of global features and RNNs on performance. For these experiments, we report MUC, B^3, and CEAF$_e$ F$_1$-scores in Table 2 as well as errors broken down by mention type and by whether the mention is anaphoric or not in Table 3. Table 3 further partitions errors into FL, FN, and WL categories, which are defined in Section 5.1. We typically think of FL and WL as representing precision errors, and FN as representing recall errors.

Our experiments consider several different settings. First, we consider an oracle setting (referred to as RNN, OH in tables), in which the model receives $z_{1:n-1}^{(o)}$, the oracle partial clustering of all mentions preceding x_n in the document,

	Non-Anaphoric (FL)		
	Nom. HM	Nom. No HM	Pron.
MR	1061	130	1075
Avg, OH	983	140	1011
RNN, GH	914	125	893
RNN, OH	913	130	842
# Mentions	9.0K	22.2K	3.1K

	Anaphoric (FN + WL)		
Model	Nom. HM	Nom. No HM	Pron.
MR	665+326	666+56	533+796
Avg, OH	781+300	641+60	578+744
RNN, GH	767+303	648+57	664+727
RNN, OH	750+289	648+52	611+686
# Mentions	4.7K	1.0K	7.3K

Table 3: Number of "false link" (FL) errors on non-anaphoric mentions (top) and number of "false new" (FN) and "wrong link" (WL) errors on anaphoric mentions (bottom) on CoNLL 2012 development set. Mentions are categorized as nominal or proper with (previous) head match (Nom. HM), nominal or proper with no head match (Nom. No HM), and pronominal. Models are described in the text, and rows in grey highlight models using oracle history.

Figure 3: Cluster predictions of greedy RNN model; co-clustered mentions are of the same color, and intensity corresponds to $\boldsymbol{h}_c(x_n)^\top \boldsymbol{h}_{<k}^{(i)}$, where $k = j+1$, $i \in \{1, 2\}$, and $x_n =$ "his." See text for full description.

and is therefore not forced to rely on its own past predictions when predicting x_n. This provides us with an upper bound on the performance achievable with our model. Next, we consider the performance of the model under a greedy inference strategy (RNN, GH), as in Algorithm 1. Finally, for baselines we consider the mention-ranking system (MR) of Wiseman et al. (2015) using our updated feature-set, as well as a non-local baseline with oracle history (Avg, OH), which averages the representations $\boldsymbol{h}_c(x_j)$ for all $x_j \in X^{(m)}$, rather than feed them through an RNN; errors are still backpropagated through the $\boldsymbol{h}_c$ representations during learning.

In Table 3 we see that the RNN improves performance overall, with the most dramatic improvements on non-anaphoric pronouns, though errors are also decreased significantly for non-anaphoric nominal and proper mentions that follow at least one mention with the same head. While WL errors also decrease for both these mention-categories under the RNN model, FN errors increase. Importantly, the RNN performance is significantly better than that of the Avg baseline, which barely improves over mention-ranking, even with oracle history. This suggests that modeling the sequence of mentions in a cluster is advantageous. We also note that while RNN performance degrades in both precision and recall when moving from the oracle history upper-bound to a greedy setting, we are still able to recover a significant portion of the possible performance improvement.

6.3 Qualitative Analysis

In this section we consider in detail the impact of the g term in the RNN scoring function on the two error categories that improve most under the RNN model (as shown in Table 3), namely, pronominal WL errors and pronominal FL errors. We consider an example from the CoNLL development set in each category on which the baseline MR model makes an error but the greedy RNN model does not.

The example in Figure 3 involves the resolution of the ambiguous pronoun "his," which is bracketed and in bold in the figure. Whereas the baseline MR model *incorrectly* predicts "his" to corefer with the closest gender-consistent antecedent "Justin" — thus making a WL error — the greedy RNN model correctly predicts "his" to corefer with "Mr. Kaye" in the previous sentence. To get a sense of the greedy RNN model's decision-making on this example, we color the mentions the greedy RNN model has predicted to corefer with "Mr. Kaye" in green, and the mentions it has predicted to corefer with "Justin" in blue. (Note that the model incorrectly predicts the

B: Yeah, it's not far. Through the S-bahn here. I mean it'slike twenty minutes.
A: Or something. And so, if I do it, I'd love to have you join me. [It's] a fancy wedding too.
(tc/ch/0010)

Figure 4: Magnitudes of gradients of NA score applied to bold "It's" with respect to final mention in three preceding clusters. See text for full description.

initial "I" mentions to corefer with "Justin.") Letting $X^{(1)}$ refer to the blue cluster, $X^{(2)}$ refer to the green cluster, and x_n refer to the ambiguous mention "his," we further shade each mention x_j in $X^{(1)}$ so that its intensity corresponds to $\boldsymbol{h}_c(x_n)^{\mathsf{T}}\boldsymbol{h}_{<k}^{(1)}$, where $k = j + 1$; mentions in $X^{(2)}$ are shaded analogously. Thus, the shading shows how highly g scores the compatibility between "his" and a cluster $X^{(i)}$ as each of $X^{(i)}$'s mentions is added. We see that when the initial "Justin" mentions are added to $X^{(1)}$ the g-score is relatively high. However, after "The company" is correctly predicted to corefer with "Justin," the score of $X^{(1)}$ drops, since companies are generally not coreferent with pronouns like "his."

In Figure 4 we consider an example (consisting of a telephone conversation between "A" and "B") in which the bracketed pronoun "It's" is being used pleonastically. Whereas the baseline MR model predicts "It's" to corefer with a previous "it" — thus making a FL error — the Greedy RNN model does not. In Figure 4 the final mention in three preceding clusters is shaded so its intensity corresponds to the magnitude of the gradient of the NA term in g with respect to that mention. This visualization technique is similar to the "saliency" technique of Li et al. (2016), and it attempts to gives a sense of the contribution of each preceding cluster in the calculation of the NA score.

We see that the potential antecedent "S-Bahn" has a large gradient, but also that the initial, obviously pleonastic use of "it's" has a large gradient, which may suggest that earlier, easier predictions of pleonasm can inform subsequent predictions.

7 Related Work

In addition to the related work noted throughout, we add supplementary references here. Unstructured approaches to coreference typically divide into mention-pair models, which classify (nearly) every pair of mentions in a document as coreferent or not (Soon et al., 2001; Ng and Cardie, 2002; Bengtson and Roth, 2008), and mention-ranking models, which select a single antecedent for each anaphoric mention (Denis and Baldridge, 2008; Rahman and Ng, 2009; Durrett and Klein, 2013; Chang et al., 2013; Wiseman et al., 2015). Structured approaches typically divide between those that induce a clustering of mentions (McCallum and Wellner, 2003; Culotta et al., 2007; Poon and Domingos, 2008; Haghighi and Klein, 2010; Stoyanov and Eisner, 2012; Cai and Strube, 2010), and, more recently, those that learn a latent tree of mentions (Fernandes et al., 2012; Björkelund and Kuhn, 2014; Martschat and Strube, 2015).

There have also been structured approaches that merge the mention-ranking and mention-pair ideas in some way. For instance, Rahman and Ng (2011) rank clusters rather than mentions; Clark and Manning (2015) use the output of both mention-ranking and mention pair systems to learn a clustering.

The application of RNNs to modeling (the trajectory of) the state of a cluster is apparently novel, though it bears some similarity to the recent work of Dyer et al. (2015), who use LSTMs to embed the state of a transition based parser's stack.

8 Conclusion

We have presented a simple, state of the art approach to incorporating global information in an end-to-end coreference system, which obviates the need to define global features, and moreover allows for simple (greedy) inference. Future work will examine improving recall, and more sophisticated approaches to global training.

References

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. 2015. Scheduled sampling for sequence prediction with recurrent neural networks. In *Advances in Neural Information Processing Systems*, pages 1171–1179.

9

Eric Bengtson and Dan Roth. 2008. Understanding the Value of Features for Coreference Resolution. In *Proceedings of the 2008 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. *ACL, Baltimore, MD, USA, June*.

Jie Cai and Michael Strube. 2010. End-to-end coreference resolution via hypergraph partitioning. In *23rd International Conference on Computational Linguistics (COLING)*, pages 143–151.

Kai-Wei Chang, Rajhans Samdani, and Dan Roth. 2013. A Constrained Latent Variable Model for Coreference Resolution. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 601–612.

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 (ACL)*, pages 1405–1415.

Aron Culotta, Michael Wick, Robert Hall, and Andrew McCallum. 2007. First-order Probabilistic Models for Coreference Resolution. In *Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics (NAACL HLT)*.

Hal Daumé III, John Langford, and Daniel Marcu. 2009. Search-based structured prediction. *Machine Learning*, 75(3):297–325.

Pascal Denis and Jason Baldridge. 2008. Specialized Models and Ranking for Coreference Resolution. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 660–669. Association for Computational Linguistics.

John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. *The Journal of Machine Learning Research*, 12:2121–2159.

Greg Durrett and Dan Klein. 2013. Easy Victories and Uphill Battles in Coreference Resolution. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1971–1982.

Greg Durrett and Dan Klein. 2014. A Joint Model for Entity Analysis: Coreference, Typing, and Linking. *Transactions of the Association for Computational Linguistics*, 2:477–490.

Chris Dyer, Miguel Ballesteros, Wang Ling, Austin Matthews, and Noah A. Smith. 2015. Transition-based dependency parsing with stack long short-term memory. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 334–343.

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.

Aria Haghighi and Dan Klein. 2010. Coreference Resolution in a Modular, Entity-centered Model. In *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 Comput.*, 9:1735–1780.

Eduard Hovy, Mitchell Marcus, Martha Palmer, Lance Ramshaw, and Ralph Weischedel. 2006. Ontonotes: the 90% Solution. In *Proceedings of the human language technology conference of the NAACL, Companion Volume: Short Papers*, pages 57–60. Association for Computational Linguistics.

Philipp Koehn. 2004. Statistical Significance Tests for Machine Translation Evaluation. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 388–395. Citeseer.

Jonathan K. Kummerfeld and Dan Klein. 2013. Error-driven Analysis of Challenges in Coreference Resolution. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, Seattle, WA, USA, October.

Nicholas Léonard, Yand Waghmare, Sagar ad Wang, and Jin-Hwa Kim. 2015. rnn: Recurrent Library for Torch. *arXiv preprint arXiv:1511.07889*.

Jiwei Li, Xinlei Chen, Eduard Hovy, and Dan Jurafsky. 2016. Visualizing and understanding neural models in nlp. In *NAACL HLT*.

Xiaoqiang Luo, Sameer Pradhan, Marta Recasens, and Eduard Hovy. 2014. An Extension of BLANC to System Mentions. *Proceedings of ACL, Baltimore, Maryland, June*.

Sebastian Martschat and Michael Strube. 2015. Latent structures for coreference resolution. *TACL*, 3:405–418.

Sebastian Martschat, Thierry Göckel, and Michael Strube. 2015. Analyzing and visualizing coreference resolution errors. In *NAACL HLT*, pages 6–10.

Andrew McCallum and Ben Wellner. 2003. Toward Conditional Models of Identity Uncertainty with Application to Proper Noun Coreference. *Advances in Neural Information Processing Systems 17*.

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.

Haoruo Peng, Kai-Wei Chang, and Dan Roth. 2015. A joint framework for coreference resolution and mention head detection. In *Proceedings of the 19th Conference on Computational Natural Language Learning (CoNLL)*, pages 12–21.

Hoifung Poon and Pedro M. Domingos. 2008. Joint unsupervised coreference resolution with markov logic. In *2008 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 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.

Sameer Pradhan, Xiaoqiang Luo, Marta Recasens, Eduard Hovy, Vincent Ng, and Michael Strube. 2014. Scoring Coreference Partitions of Predicted Mentions: A Reference Implementation. In *Proceedings of the Association for Computational Linguistics*.

Altaf Rahman and Vincent Ng. 2009. Supervised Models for Coreference Resolution. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2-Volume 2*, pages 968–977. Association for Computational Linguistics.

Altaf Rahman and Vincent Ng. 2011. Narrowing the modeling gap: A cluster-ranking approach to coreference resolution. *J. Artif. Intell. Res. (JAIR)*, 40:469–521.

Stéphane Ross, Geoffrey J. Gordon, and Drew Bagnell. 2011. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pages 627–635.

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.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958.

Veselin Stoyanov and Jason Eisner. 2012. Easy-first Coreference Resolution. In *COLING*, pages 2519–2534. Citeseer.

Ilya Sutskever, Oriol Vinyals, and Quoc VV Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems (NIPS)*, pages 3104–3112.

Laurens van der Maaten and Geoffrey E. Hinton. 2012. Visualizing non-metric similarities in multiple maps. *Machine Learning*, 87(1):33–55.

Sam Wiseman, Alexander M. Rush, Stuart M. 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 (ACL)*, pages 1416–1426.

Chun-Nam John Yu and Thorsten Joachims. 2009. Learning Structural SVMs with Latent Variables. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 1169–1176. ACM.

Wojciech Zaremba, Ilya Sutskever, and Oriol Vinyals. 2014. Recurrent neural network regularization. *CoRR*, abs/1409.2329.