Unsupervised Ranking Model for Entity Coreference Resolution

Xuezhe Ma and Zhengzhong Liu and Eduard Hovy
Language Technologies Institute
Carnegie Mellon University
Pittsburgh, PA 15213, USA
{xuezhem, liu}@cs.cmu.edu, ehovy@cmu.edu

Abstract

Coreference resolution is one of the first stages in deep language understanding and its importance has been well recognized in the natural language processing community. In this paper, we propose a generative, unsupervised ranking model for entity coreference resolution by introducing resolution mode variables. Our unsupervised system achieves 58.44% F1 score of the CoNLL metric on the English data from the CoNLL-2012 shared task (Pradhan et al., 2012), outperforming the Stanford deterministic system (Lee et al., 2013) by 3.01%.

1 Introduction

Entity coreference resolution has become a critical component for many Natural Language Processing (NLP) tasks. Systems requiring deep language understanding, such as information extraction (Wellner et al., 2004), semantic event learning (Chambers and Jurafsky, 2008; Chambers and Jurafsky, 2009), and named entity linking (Durrett and Klein, 2014; Ji et al., 2014) all benefit from entity coreference information.

Entity coreference resolution is the task of identifying mentions (i.e., noun phrases) in a text or dialogue that refer to the same real-world entities. In recent years, several supervised entity coreference resolution systems have been proposed, which, according to Ng (2010), can be categorized into three classes — mention-pair models (McCarthy and Lehnert, 1995), entity-mention models (Yang et al., 2008a; Haghighi and Klein, 2010; Lee et al., 2011) and ranking models (Yang et al., 2008b; Durrett and Klein, 2013; Fernandes et al., 2014) — among which ranking models recently obtained state-of-the-art performance. However, the manually annotated corpora that these systems rely on are highly expensive to create, in particular when we want to build data for resource-poor languages (Ma and Xia, 2014). That makes unsupervised approaches, which only require unannotated text for training, a desirable solution to this problem.

Several unsupervised learning algorithms have been applied to coreference resolution. Haghighi and Klein (2007) presented a mention-pair non-parametric fully-generative Bayesian model for unsupervised coreference resolution. Based on this model, Ng (2008) probabilistically induced coreference partitions via EM clustering. Poon and Domingos (2008) proposed an entity-mention model that is able to perform joint inference across mentions by using Markov Logic. Unfortunately, these unsupervised systems’ performance on accuracy significantly falls behind those of supervised systems, and are even worse than the deterministic rule-based systems. Furthermore, there is no previous work exploring the possibility of developing an unsupervised ranking model which achieved state-of-the-art performance under supervised settings for entity coreference resolution.

In this paper, we propose an unsupervised generative ranking model for entity coreference resolution. Our experimental results on the English data from the CoNLL-2012 shared task (Pradhan et al., 2012) show that our unsupervised system outperforms the Stanford deterministic system (Lee et al., 2013) by 3.01% absolute on the CoNLL official metric. The
contributions of this work are (i) proposing the first unsupervised ranking model for entity coreference resolution. (ii) giving empirical evaluations of this model on benchmark data sets. (iii) considerably narrowing the gap to supervised coreference resolution accuracy.

2 Unsupervised Ranking Model

2.1 Notations and Definitions

In the following, \( D = \{ m_0, m_1, \ldots, m_n \} \) represents a generic input document which is a sequence of coreference mentions, including the artificial root mention (denoted by \( m_0 \)). The method to detect and extract these mentions is discussed later in Section 2.6. Let \( C = \{ c_1, c_2, \ldots, c_n \} \) denote the coreference assignment of a given document, where each mention \( m_i \) has an associated random variable \( c_i \) taking values in the set \( \{ 0, i, \ldots, i - 1 \} \); this variable specifies \( m_i \)'s selected antecedent \( c_i \in \{ 1, 2, \ldots, i - 1 \} \), or indicates that it begins a new coreference chain \( (c_i = 0) \).

2.2 Generative Ranking Model

The following is a straightforward way to build a generative model for coreference:

\[
P(D, C) = P(D|C)P(C) = \prod_{j=1}^{n} P(m_j|m_{c_j}) \prod_{j=1}^{n} P(c_j|j) (1)
\]

where we factorize the probabilities \( P(D|C) \) and \( P(C) \) into each position \( j \) by adopting appropriate independence assumptions that given the coreference assignment \( c_j \) and corresponding coreferent mention \( m_{c_j} \), the mention \( m_j \) is independent with other mentions in front of it. This independence assumption is similar to that in the IBM 1 model on machine translation (Brown et al., 1993), where it assumes that given the corresponding English word, the aligned foreign word is independent with other English and foreign words. We do not make any independent assumptions among different features (see Section 2.4 for details).

Inference in this model is efficient, because we can compute \( c_j \) separately for each mention:

\[
c_j^* = \arg\max_{c_j} P(m_j|m_{c_j})P(c_j|j)
\]

The model is a so-called ranking model because it is able to identify the most probable candidate antecedent given a mention to be resolved.

2.3 Resolution Mode Variables

According to previous work (Haghighi and Klein, 2009; Ratinov and Roth, 2012; Lee et al., 2013), antecedents are resolved by different categories of information for different mentions. For example, the Stanford system (Lee et al., 2013) uses string-matching sieves to link two mentions with similar text and precise-construct sieve to link two mentions which satisfy special syntactic or semantic relations such as apposition or acronym. Motivated by this, we introduce resolution mode variables \( \Pi = \{ \pi_1, \ldots, \pi_n \} \), where for each mention \( j \) the variable \( \pi_j \in \{ \text{str, prec, attr} \} \) indicates in which mode the mention should be resolved. In our model, we define three resolution modes — string-matching (\( \text{str} \)), precise-construct (\( \text{prec} \)), and attribute-matching (\( \text{attr} \)) — and \( \Pi \) is deterministic when \( D \) is given (i.e. \( P(\Pi|D) \) is a point distribution). We determine \( \pi_j \) for each mention \( m_j \) in the following way:

\[
\bullet \ \pi_j = \text{str}, \text{ if there exists a mention } m_i, i < j \text{ such that the two mentions satisfy the String Match sieve, the Relaxed String Match sieve, or the Strict Head Match sieve in the Stanford multi-sieve system (Lee et al., 2013)}.
\]

\[
\bullet \ \pi_j = \text{prec}, \text{ if there exists a mention } m_i, i < j \text{ such that the two mentions satisfy the Speaker Identification sieve, or the Precise Constructs sieve}.
\]

\[
\bullet \ \pi_j = \text{attr}, \text{ if there is no mention } m_i, i < j \text{ satisfies the above two conditions}.
\]

Now, we can extend the generative model in Eq. 1 to:

\[
P(D, C) = P(D, C, \Pi) = \prod_{j=1}^{n} P(m_j|m_{c_j}, \pi_j)P(c_j|\pi_j, j)P(\pi_j|j)
\]

where we define \( P(\pi_j|j) \) to be uniform distribution.

We model \( P(m_j|m_{c_j}, \pi_j) \) and \( P(c_j|\pi_j, j) \) in the fol-
### Table 1: Feature set for representing a mention under different resolution modes. The *Distance* feature is for parameter $q$, while all other features are for parameter $t$.

**Algorithm 1: Learning Model with EM**

1. **Initialization:** Initialize $\theta_0 = \{t_0, q_0\}$
2. for $t = 0$ to $T$
   3. set all counts $c(\ldots) = 0$
   4. for each document $D$
      5. for $j = 1$ to $n$
         6. for $k = 0$ to $j - 1$
            7. $L_{jk} = \frac{t(m_j|m_k, \pi)}{\sum_{i=0}^{\pi} t(m_j|m_i, \pi) q(i|\pi_j, j)}$
            8. $c(m_j, m_k, \pi_j) += L_{jk}$
            9. $c(m_k, \pi_j) += L_{jk}$
            10. $c(k, j, \pi_j) += L_{jk}$
            11. $c(j, \pi_j) += L_{jk}$
   12. // Recalculate the parameters
      13. $t(m|m', \pi) = \frac{c(m|m', \pi)}{c(m, \pi)}$
      14. $q(k, j, \pi) = \frac{c(k, j, \pi)}{c(j, \pi)}$

following way:

$$P(m_j|m_{c_j}, \pi_j) = t(m_j|m_{c_j}, \pi_j)$$

where $\theta = \{t, q\}$ are parameters of our model. Note that in the attribute-matching mode ($\pi_j = attr$) we model $P(c_j|\pi_j, j)$ with parameter $q$, while in the other two modes, we use the uniform distribution. It makes sense because the position information is important for coreference resolved by matching attributes of two mentions such as resolving pronoun coreference, but not that important for those resolved by matching text or special relations like two mentions referring the same person and matching by the name.

**2.4 Features**

In this section, we describe the features we use to represent mentions. Specifically, as shown in Table 1, we use different features under different resolution modes. It should be noted that only the *Distance* feature is designed for parameter $q$, all other features are designed for parameter $t$.

**2.5 Model Learning**

For model learning, we run EM algorithm (Dempster et al., 1977) on our Model, treating $D$ as observed data and $C$ as latent variables. We run EM with 10 iterations and select the parameters achieving the best performance on the development data. Each iteration takes around 12 hours with 10 CPUs parallelly. The best parameters appear at around the 5th iteration, according to our experiments. The detailed derivation of the learning algorithm is shown in Appendix A. The pseudo-code is shown is Algorithm 1. We use uniform initialization for all the parameters in our model.

Several previous work has attempted to use EM for entity coreference resolution. Cherry and Bergsma (2005) and Charniak and Elsner (2009) applied EM for pronoun anaphora resolution. Ng (2008) probabilistically induced coreference partitions via EM clustering. Recently, Moosavi and Strube (2014) proposed an unsupervised model uti-
Table 3: F1 scores of different evaluation metrics for our model, together with two deterministic systems and one unsupervised system as baseline (above the dashed line) and seven supervised systems (below the dashed line) for comparison on CoNLL 2012 development and test datasets.

<table>
<thead>
<tr>
<th></th>
<th>CoNLL'12 English development data</th>
<th>CoNLL'12 English test data</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MUC</td>
<td>B³</td>
</tr>
<tr>
<td>MIR</td>
<td>65.39</td>
<td>54.89</td>
</tr>
<tr>
<td>Stanford</td>
<td>64.96</td>
<td>54.49</td>
</tr>
<tr>
<td>Multigraph</td>
<td>66.22</td>
<td>56.41</td>
</tr>
<tr>
<td>Our Model</td>
<td>67.89</td>
<td>57.83</td>
</tr>
<tr>
<td>IMS</td>
<td>67.15</td>
<td>55.19</td>
</tr>
<tr>
<td>Latent-Tree</td>
<td>69.46</td>
<td>57.83</td>
</tr>
<tr>
<td>Berkeley</td>
<td>70.44</td>
<td>59.10</td>
</tr>
<tr>
<td>LaSO</td>
<td>70.74</td>
<td>60.03</td>
</tr>
<tr>
<td>Latent-Strc</td>
<td>72.11</td>
<td>60.74</td>
</tr>
<tr>
<td>Model-Stack</td>
<td>72.59</td>
<td>61.98</td>
</tr>
<tr>
<td>Non-Linear</td>
<td>72.74</td>
<td>61.77</td>
</tr>
</tbody>
</table>

Evaluation Metrics. We evaluate our model on three measures widely used in the literature: MUC (Vilain et al., 1995), B³ (Bagga and Baldwin, 1998), and Entity-based CEAF (CEAFe) (Luo, 2005). In addition, we also report results on another two popular metrics: Mention-based CEAF (CEAFm) and BLANC (Recasens and Hovy, 2011). All the results are given by the latest version of CoNLL-2012 scorer¹.

3.2 Results and Comparison

Table 3 illustrates the results of our model together as baseline with two deterministic systems, namely Stanford: the Stanford system (Lee et al., 2011) and Multigraph: the unsupervised multigraph system (Martschat, 2013), and one unsupervised system, namely MIR: the unsupervised system using most informative relations (Moosavi and Strube, 2014). Our model outperforms the three baseline systems on all the evaluation metrics. Specifically, our model achieves improvements of 2.93% and 3.01% on CoNLL F1 score over the Stanford system, the winner of the CoNLL 2011 shared task, on the CoNLL 2012 development and test sets, respectively. The improvements on CoNLL F1 score over the Multigraph model are 1.41% and 1.77% on the development and test sets, respectively. Comparing

¹http://conll.cemantix.org/2012/software.html
with the MIR model, we obtain significant improvements of 2.62% and 3.02% on CoNLL F1 score. To make a thorough empirical comparison with previous studies, Table 3 (below the dashed line) also shows the results of some state-of-the-art supervised coreference resolution systems — IMS: the second best system in the CoNLL 2012 shared task (Björkelund and Farkas, 2012); Latent-Tree: the latent tree model (Fernandes et al., 2012) obtaining the best results in the shared task; Berkeley: the Berkeley system with the final feature set (Durrett and Klein, 2013); LaSO: the structured perceptron system with non-local features (Björkelund and Kuhn, 2014); Latent-Strc: the latent structure system (Martschat and Strube, 2015); Model-Stack: the entity-centric system with model stacking (Clark and Manning, 2015); and Non-Linear: the non-linear mention-ranking model with feature representations (Wiseman et al., 2015). Our unsupervised ranking model outperforms the supervised IMS system by 1.02% on the CoNLL F1 score, and achieves competitive performance with the latent tree model. Moreover, our approach considerably narrows the gap to other supervised systems listed in Table 3.

4 Conclusion

We proposed a new generative, unsupervised ranking model for entity coreference resolution into which we introduced resolution mode variables to distinguish mentions resolved by different categories of information. Experimental results on the data from CoNLL–2012 shared task show that our system significantly improves the accuracy on different evaluation metrics over the baseline systems.

One possible direction for future work is to differentiate more resolution modes. Another one is to add more precise or even event-based features to improve the model’s performance.

Acknowledgements

This research was supported in part by DARPA grant FA8750-12-2-0342 funded under the DEFT program. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of DARPA.

References


Washington, USA, October. Association for Computational Linguistics.


**Appendix A. Derivation of Model Learning**

Formally, we iteratively estimate the model parameters $\theta$, employing the following EM algorithm:

**E-step:** Compute the posterior probabilities of $C$, $P(C|D; \theta)$, based on the current $\theta$.

**M-step:** Calculate the new $\theta'$ that maximizes the expected complete log likelihood, $E_{P(C|D, \theta)}[\log P(D, C; \theta')]$

For simplicity, we denote:

$$P(C|D; \theta) = \tilde{P}(C|D)$$
$$P(C|D; \theta') = P(C|D)$$

In addition, we use $\tau(\pi_j|j)$ to denote the probability $P(\pi_j|j)$ which is uniform distribution in our model.

Moreover, we use the following notation for convenience:

$$\theta(m_j, m_k, j, k, \pi_j) = t(m_j|m_k, \pi_j) q(k|\pi_j, j) \tau(\pi_j|j)$$

Then, we have

$$E_{\tilde{P}(C|D)}[\log P(D, C)] = \sum_{C} P(C|D) \log P(D, C)$$

$$= \sum_{C} \tilde{P}(C|D) \left( \sum_{j=1}^{n} \log t(m_j|m_c, \pi_j) + \log q(c_j|\pi_j, j) + \log \tau(\pi_j|j) \right)$$

$$= \sum_{j=1}^{n} \sum_{k=0}^{j-1} L_{jk} \left( \log t(m_j|m_k, \pi_j) + \log q(k|\pi_j, j) + \log \tau(\pi_j|j) \right)$$

Then the parameters $t$ and $q$ that maximize $E_{\tilde{P}(C|D)}[\log P(D, C)]$ satisfy that

$$t(m_j|m_k, \pi_j) = \frac{L_{jk}}{\sum_{i=0}^{j-1} L_{ij}}$$

$$q(k|\pi_j, j) = \frac{L_{jk}}{\sum_{i=0}^{j} L_{ji}}$$

where $L_{jk}$ can be calculated by

$$L_{jk} = \sum_{C, c_j=k} \tilde{P}(C|D) \frac{\prod_{i=1}^{n} \theta(m_i, m_{c_i}, c_i, \pi_i)}{\prod_{i=1}^{n} \theta(m_i, m_{c_i}, c_i, \pi_i)}$$

$$= \sum_{C, c_j=k} \tilde{P}(C|D) \frac{\prod_{i=1}^{n} \theta(m_i, m_{c_i}, c_i, \pi_i)}{\prod_{i=1}^{n} \theta(m_i, m_{c_i}, c_i, \pi_i)}$$

$$= \sum_{C, c_j=k} \tilde{P}(C|D) \frac{\prod_{i=1}^{n} \theta(m_i, m_{c_i}, c_i, \pi_i)}{\prod_{i=1}^{n} \theta(m_i, m_{c_i}, c_i, \pi_i)}$$

where $C(\neg j) = \{c_1, \ldots, c_{j-1}, c_{j+1}, \ldots, c_n\}$. The above derivations correspond to the learning algorithm in Algorithm 1.