Latent Structures for Coreference Resolution

Sebastian Martschat and **Michael Strube**
Heidelberg Institute for Theoretical Studies gGmbH
Schloss-Wolfsbrunnenweg 35, 69118 Heidelberg, Germany
(sebastian.martschat|michael.strube)@h-its.org

Abstract

Machine learning approaches to coreference resolution vary greatly in the modeling of the problem: while early approaches operated on the mention pair level, current research focuses on ranking architectures and antecedent trees. We propose a unified representation of different approaches to coreference resolution in terms of the structure they operate on. We represent several coreference resolution approaches proposed in the literature in our framework and evaluate their performance. Finally, we conduct a systematic analysis of the output of these approaches, highlighting differences and similarities.

1 Introduction

Coreference resolution is the task of determining which mentions in a text are used to refer to the same real-world entity. The era of statistical natural language processing saw the shift from rule-based approaches (Hobbs, 1976; Lappin and Leass, 1994) to increasingly sophisticated machine learning models. While early approaches cast the problem as binary classification of mention pairs (Soon et al., 2001), recent approaches make use of complex structures to represent coreference relations (Yu and Joachims, 2009; Fernandes et al., 2014).

The aim of this paper is to devise a framework for coreference resolution that leads to a unified representation of different approaches to coreference resolution in terms of the *structure* they operate on. Previous work in other areas of natural language processing such as parsing (Klein and Manning, 2001) and machine translation (Lopez, 2009) has shown that providing unified representations of approaches to a problem deepens its understanding and can also lead to empirical improvements. By implementing popular approaches in this framework, we can highlight structural differences and similarities between them. Furthermore, this establishes a setting to systematically analyze the contribution of the underlying structure to performance, while fixing parameters such as preprocessing and features.

In particular, we analyze approaches to coreference resolution and point out that they mainly differ in the structures they operate on. We then note that these structures are not annotated in the training data (Section 2). Motivated by this observation, we develop a machine learning framework for structured prediction with latent variables for coreference resolution (Section 3). We formalize the mention pair model (Soon et al., 2001; Ng and Cardie, 2002), mention ranking architectures (Denis and Baldridge, 2008; Chang et al., 2012) and antecedent trees (Fernandes et al., 2014) in our framework and highlight key differences and similarities (Section 4). Finally, we present an extensive comparison and analysis of the implemented approaches, both quantitative and qualitative (Sections 5 and 6). Our analysis shows that a mention ranking architecture with latent antecedents performs best, mainly due to its ability to structurally model determining anaphoricity. Finally, we briefly describe how entity-centric approaches fit into our framework (Section 7).

An open source toolkit which implements the machine learning framework and the approaches discussed in this paper is available for download[1].

[1] http://smartschat.de/software

Transactions of the Association for Computational Linguistics, vol. 3, pp. 405–418, 2015. Action Editor: Mark Johnson.
Submission batch: 3/2015; Revision batch 6/2015; Published 7/2015.

2 Modeling Coreference Resolution

The aim of automatic coreference resolution is to predict a clustering of mentions such that each cluster contains all mentions that are used to refer to the same entity. However, most coreference resolution models reduce the problem to predicting coreference between pairs of mentions, and jointly or cascadingly consolidating these predictions. Approaches differ in the scope (pairwise, per anaphor, per document, ...) they employ while learning a scoring function for these pairs, and the way the consolidating is handled.

The different ways to employ the scope and to consolidate decisions can be understood as operating on *latent structures*: as pairwise links are not annotated in the data, coreference approaches create structures (either heuristically or data-driven) that guide the learning of the pairwise scoring function.

To understand this better, let us consider two examples. Mention pair models (Soon et al., 2001; Ng and Cardie, 2002) cast the problem as first creating a list of mention pairs, and deciding for each pair whether the two mentions are coreferent. Afterwards the decisions are consolidated by a clustering algorithm such as best-first or closest-first. We therefore can consider this approach to operate on a *list* of mention pairs where each pair is handled individually. In contrast, antecedent tree models (Fernandes et al., 2014; Björkelund and Kuhn, 2014) consider the whole document at once and predict a *tree* consisting of anaphor-antecedent pairs.

3 A Structured Prediction Framework

In this section we introduce a structured prediction framework for learning coreference predictors with latent variables. When devising the framework, we focus on accounting for the *latent structures* underlying coreference resolution approaches. The framework is a generalization of previous work on latent antecedents and trees for coreference resolution (Yu and Joachims, 2009; Chang et al., 2012; Fernandes et al., 2014).

3.1 Setting

In all prediction tasks, the goal is to learn a mapping f from inputs $x \in \mathcal{X}$ to outputs $y \in \mathcal{Y}_x$. A prediction task is structured if the output elements $y \in \mathcal{Y}_x$ exhibit some structure. As we work in a latent variable setting, we assume that $\mathcal{Y}_x = \mathcal{H}_x \times \mathcal{Z}_x$, and therefore $y = (h, z) \in \mathcal{H}_x \times \mathcal{Z}_x$. We call h the hidden or latent part, which is not observed in the data, and z the observed part (during training). We assume that z can be inferred from h, and that in a pair (h, z), h and z are always consistent.

We first define the input space $\mathcal{X}$ and the output spaces $\mathcal{H}_x$ and $\mathcal{Z}_x$ for $x \in \mathcal{X}$.

3.2 The Input Space $\mathcal{X}$

The input space consists of documents. We represent a document $x \in \mathcal{X}$ as follows. Let us assume that M_x is the set of mentions (expressions which may be used to refer to entities) in the document. We write $M_x = \{m_1, \ldots, m_k\}$, where the m_i are in ascending order with respect to their position in the document. We then consider $M_x^0 = \{m_0\} \cup M_x$, where m_0 precedes every $m_i \in M_x$ (Chang et al., 2012; Fernandes et al., 2014).

m_0 plays the role of a *dummy mention* for anaphoricity detection: if m_0 is chosen as the antecedent, the corresponding mention is deemed as non-anaphoric. This enables joint coreference resolution and anaphoricity determination.

3.3 The Latent Space $\mathcal{H}_x$ for an Input x

Let $x \in \mathcal{X}$ be some document. As we saw in the previous section, approaches to coreference resolution predict a latent structure which is not annotated in the data but is used to infer coreference information. Inspired by previous work on coreference (Bengtson and Roth, 2008; Fernandes et al., 2014; Martschat and Strube, 2014), we now develop a graph-based representation for these structures.

A valid latent structure for the document x is a labeled directed graph $h = (V, A, L_A)$ where

- the set of nodes are the mentions, $V = M_x^0$,
- the set of edges A consists of links between mentions pointing back in the text,

$$A \subseteq \{(m_j, m_i) \mid j > i\} \subseteq M_x \times M_x^0.$$

- $L_A \colon A \to L$ assigns a label $\ell \in L$ to each edge. L is a finite set of labels, for example signaling coreference or non-coreference.

We split h into subgraphs (called *substructures* from now on), which we notate as $h = h_1 \oplus \ldots \oplus h_n$,

with $h_i = (V_i, A_i, L_{A_i}) \in \mathcal{H}_{x,i}$, where $\mathcal{H}_{x,i}$ is the latent space for an input x restricted to the mentions appearing in h_i. h_i encodes coreference decisions for a subset of mentions in x.

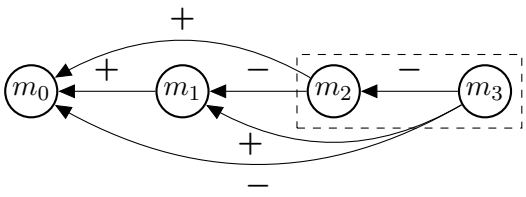

Figure 1: Graph-based representation of the mention pair model. The dashed box shows one *substructure* of the structure.

Figure 1 depicts a graph that captures the latent structure underlying the mention pair model. Mention pairs are represented as node connected by an edge. The edge either has label "+" (if the mentions are coreferent) or "−" (otherwise). As the mention pair model considers each mention pair individually, each edge is one substructure of the latent structure (expressed via the dashed box). We describe this representation in more detail in Section 4.1.

3.4 The Observed Output Space $\mathcal{Z}_x$ for an Input x

Let $x \in \mathcal{X}$ be some document. The observed output space consists of all functions $e_x \colon M_x \to \mathbb{N}$ that map mentions to entity identifiers. Two $m_i, m_j \in M_x$ are coreferent if and only if $e_x(m_i) = e_x(m_j)$. e_x is inferred from the latent structure, e.g. by taking the transitive closure over coreference decisions.

This representation corresponds to the way coreference is annotated in corpora.

3.5 Linear Models

Let us write $\mathcal{H} = \cup_{x \in \mathcal{X}} \mathcal{H}_x$ for the full latent space (analogously $\mathcal{Z}$). Our goal is to learn the mapping $f \colon \mathcal{X} \to \mathcal{H} \times \mathcal{Z}$. We assume that the mapping is parametrized by a weight vector $\theta \in \mathbb{R}^d$, and therefore write $f = f_\theta$. We restrict ourselves to *linear models*. That is,

$$f_\theta(x) = \operatorname*{arg\,max}_{(h,z) \in \mathcal{H}_x \times \mathcal{Z}_x} \langle \theta, \phi(x, h, z) \rangle,$$

where $\phi \colon \mathcal{X} \times \mathcal{H} \times \mathcal{Z} \to \mathbb{R}^d$ is a joint feature function for inputs and candidate outputs.

Since $h = h_1 \oplus \ldots \oplus h_n$, we have

$$\begin{aligned} f_\theta(x) &= \operatorname*{arg\,max}_{(h,z) \in \mathcal{H}_x \times \mathcal{Z}_x} \langle \theta, \phi(x, h, z) \rangle \\ &= \bigoplus_{i=1}^{n} \operatorname*{arg\,max}_{(h_i,z) \in \mathcal{H}_{x,i} \times \mathcal{Z}_x} \langle \theta, \phi(x, h_i, z) \rangle. \end{aligned}$$

In this paper, we only consider feature functions which factor with respect to the edges in $h_i = (V_i, A_i, L_{A_i})$, i.e. $\phi(x, h_i, z) = \sum_{a \in A_i} \phi(x, a, z)$. Hence, the features examine properties of mention pairs, such as head word of each mention, number of each mention, or the existence of a string match. We describe the feature set used for all approaches represented in our framework in Section 5.2.

3.6 Decoding

Given an input $x \in \mathcal{X}$ and a weight vector $\theta \in \mathbb{R}^d$, we obtain the prediction by solving the $\arg\max$ equation described in the previous subsection. This can be viewed as searching the output space $\mathcal{H}_x \times \mathcal{Z}_x$ for the highest scoring output pair (h, z).

The details of the search procedure depend on the space $\mathcal{H}_x$ of latent structures and the factorization into substructures. For the structures we consider in this paper, the maximization can be solved exactly via greedy search. For structures with complex constraints like transitivity, more complex or even approximate search methods need to be used (Klenner, 2007; Finkel and Manning, 2008).

3.7 Learning

We assume a supervised learning setting with latent variables, i.e., we have a training set of documents

$$\mathcal{D} = \left\{ \left(x^{(i)}, z^{(i)} \right) \mid i = 1, \ldots, m \right\}$$

at our disposal. Note that the latent structures are not encoded in this training set.

In principle we would like to directly optimize for the evaluation metric we are interested in. Unfortunately, the evaluation metrics used in coreference do not allow for efficient optimization based on mention pairs, since they operate on the entity level. For example, the CEAF_e metric (Luo, 2005) needs to compute optimal entity alignments between gold and system entities. These alignments do not factor with respect to mention pairs. We therefore have to use some surrogate loss.

Algorithm 1 Structured latent perceptron with cost-augmented inference.

Input: Training set $\mathcal{D}$, a cost function c, number of epochs n.

function PERCEPTRON($\mathcal{D}$, c, n)
  set $\theta = (0, \ldots, 0)$
  for epoch $= 1, \ldots, n$ **do**
    for $(x, z) \in \mathcal{D}$ **do**
      for each substructure **do**
        $\hat{h}_{\text{opt},i} = \underset{h_i \in \text{const}(\mathcal{H}_{x,z,i})}{\arg\max} \langle \theta, \phi(x, h_i, z) \rangle$
        $(\hat{h}_i, \hat{z}) = \underset{(h_i, z) \in \mathcal{H}_{x,i} \times \mathcal{Z}_x}{\arg\max} (\langle \theta, \phi(x, h_i, z) \rangle + c(x, h_i, \hat{h}_{\text{opt},i}, z))$
        if $\hat{h}_i$ does not partially encode z **then**
          set $\theta = \theta + \phi(x, \hat{h}_{\text{opt},i}, z) - \phi(x, \hat{h}_i, \hat{z})$

Output: A weight vector θ.

We employ a *structured latent perceptron* (Sun et al., 2009) extended with *cost-augmented inference* (Crammer et al., 2006) to learn the parameters of the models we discuss. While this restricts us to a particular objective to optimize, it comes with various advantages: the implementation is simple and fast, we can incorporate error functions via cost-augmentation, the structures are plug-and-play if we provide a decoder, and the (structured) perceptron with cost-augmented inference has exhibited good performance for coreference resolution (Chang et al., 2012; Fernandes et al., 2014).

To describe the algorithm, we need some additional terminology. Let (x, z) be a training example. Let $(\hat{h}, \hat{z}) = f_\theta(x)$ be the prediction under the model parametrized by θ. Let $\mathcal{H}_{x,z}$ be the space of all latent structures for an input x that are consistent with a coreference output z. Structures in $\mathcal{H}_{x,z}$ provide substitutes for gold structures in training. Some approaches restrict $\mathcal{H}_{x,z}$, for example by learning only from the closest antecedent of a mention (Denis and Baldridge, 2008). Hence, we consider the *constrained* space $\text{const}(\mathcal{H}_{x,z}) \subseteq \mathcal{H}_{x,z}$, where const is a function that depends on the approach in focus.

$$\hat{h}_{\text{opt}} = \underset{h \in \text{const}(\mathcal{H}_{x,z})}{\arg\max} \langle \theta, \phi(x, h, z) \rangle$$

is the optimal constrained latent structure under the current model which is consistent with z. We write $\hat{h}_i$ and $\hat{h}_{\text{opt},i}$ for the ith substructure of the latent structure.

To estimate θ, we iterate over the training data. For each input, we compute the optimal constrained prediction consistent with the gold information, $\hat{h}_{\text{opt,i}}$. We then compute the optimal prediction $(\hat{h}_i, \hat{z})$, but also include the cost function c in our maximization problem. This favors solutions with high cost, which leads to a large margin approach.

If $\hat{h}_i$ does not partially encode the gold data, we update the weight vector. This is repeated for a given number of epochs[2]. Algorithm 1 gives a more formal description.

4 Latent Structures

In the previous section we developed a machine learning framework for coreference resolution. It is flexible with respect to

- the latent structure $h \in \mathcal{H}_x$ for an input x,
- the substructures of $h \in \mathcal{H}_x$,
- the constrained space of latent structures consistent with a gold solution $\text{const}(H_{x,z})$, and
- the cost function c and its factorization.

In this paper, we focus on giving a unified representation and in-depth analysis of prevalent coreference models from the literature. Future work should investigate devising and analyzing novel representations for coreference resolution in the framework.

We express three main coreference models in our framework, the mention pair model (Soon et al., 2001), the mention ranking model (Denis and Baldridge, 2008; Chang et al., 2012) and antecedent trees (Yu and Joachims, 2009; Fernandes et al., 2014; Björkelund and Kuhn, 2014). We characterize each approach by the latent structure it operates on during learning and inference (we assume that all approaches we consider share the same features). Furthermore, we also discuss the factorization into substructures and typical cost functions used in the literature.

4.1 Mention Pair Model

We first consider the mention pair model. In its original formulation, it extracts mention pairs from the

[2]We also shuffle the data before each epoch and use averaging (Collins, 2002).

data and labels these as positive or negative. During testing, all pairs are extracted and some clustering algorithm such as closest-first or best-first is applied to the list of pairs. During training, some heuristic is applied to help balancing positive and negative examples. The most popular heuristic is to take the closest antecedent of an anaphor as a positive example, and all pairs in between as negative examples.

Latent Structure. In our framework, we can represent the mention pair model as a labeled graph. In particular, let the set of edges be all backward-pointing edges, i.e. $A = \{(m_j, m_i) \mid j > i\}$. In the testing phase, we operate on the whole set A. During training, we consider only a subset of edges, as defined by the heuristic used by the approach.

The labeling function maps a pair of mentions to a positive ("+") or a negative label ("−") via

$$L_A(m_j, m_i) = \begin{cases} + & m_j, m_i \text{ are coreferent,} \\ - & \text{otherwise.} \end{cases}$$

One such graph is depicted in Figure 1 (Section 3).

A clustering algorithm (like closest-first or best-first) is then employed to infer the coreference information from this latent structure.

Substructures. In the mention pair model, the parts of the substructures are the individual edges: each pair of mentions is considered as an instance from which the model learns and which the model predicts individually.

Cost Function. As discussed above, mention pair approaches employ heuristics to resample the training data. This is a common method to introduce cost-sensitivity into classification (Elkan, 2001; Geibel and Wysotzk, 2003). Hence, mention pair approaches do not use cost functions in addition to the resampling.

4.2 Mention Ranking Model

The mention ranking model captures competition between antecedents: for each anaphor, the highest-scoring antecedent is selected. For training, this approach needs gold antecedents to compare to. There are two main approaches to determine these: first, they are heuristically extracted similarly to the mention pair model (Denis and Baldridge, 2008; Rahman and Ng, 2011). Second, latent antecedents are employed (Chang et al., 2012): in such models, the highest-scoring preceding coreferent mention of an anaphor under the current model is selected as the gold antecedent.

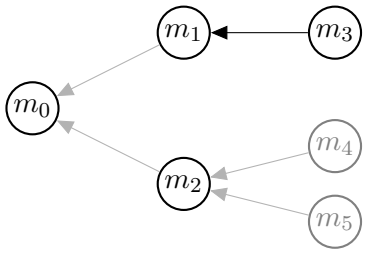

Figure 2: Latent structure underlying the mention ranking and the antecedent tree approach. The black nodes and arcs represent one substructure for the mention ranking approach.

Latent Structure. The mention ranking approach can be represented as an unlabeled graph. In particular, we allow any graph with edges $A \subseteq \{(m_j, m_i) \mid j > i\}$ such that for all j there is exactly one i with $(m_j, m_i) \in A$ (each anaphor has exactly one antecedent). Figure 2 shows an example graph.

We can represent heuristics for creating training data by constraining the latent structures consistent with the gold information $\mathcal{H}_{x,z}$. Again, the most popular heuristic is to consider the closest antecedent of a mention as the gold antecedent during training (Denis and Baldridge, 2008). This corresponds to constraining $\mathcal{H}_{x,z}$ such that $\text{const}(\mathcal{H}_{x,z}) = \{h\}$ with $h = (V, A, L_A)$ and $(m_j, m_i) \in A$ if and only if m_i is the closest antecedent of m_j. When learning from latent antecedents, the unconstrained space $\mathcal{H}_{x,z}$ is considered.

To infer coreference information from this latent structure, we take the transitive closure over all anaphor-antecedent decisions encoded in the graph.

Substructures. The distinctive feature of the mention ranking approach is that it considers each anaphor in isolation, but all candidate antecedents at once. We therefore define substructures as follows. The jth substructure is the graph h_j with nodes $V_j = \{m_0, \ldots, m_j\}$ and $A_j = \{(m_j, m_i) \mid \text{there is } i \text{ with } j > i \text{ s.t. } (m_j, m_i) \in A\}$. A_j contains the antecedent decision for m_j. One such substructure encoding the antecedent decision for m_3 is colored black in Figure 2.

Cost Function. Cost functions for the mention ranking model can reward the resolution of specific classes. The most sophisticated cost function was proposed by Durrett and Klein (2013), who distinguish between three errors: finding an antecedent for a non-anaphoric mention, misclassifying an anaphoric mention as non-anaphoric, and finding a wrong antecedent for an anaphoric mention. We will use a variant of this cost function in our experiments (described in Section 5.3).

4.3 Antecedent Trees

Finally, we consider antecedent trees. This structure encodes all antecedent decisions for all anaphors. In our framework they can be understood as an extension of the mention ranking approach to the document level. So far, research did not investigate constraints on the space of latent structures consistent with the gold annotation.

Latent Structure. Antecedent trees are based on the same structure as the mention ranking approach.

Substructures. In the antecedent tree approach, the latent structure does not factor in parts: the whole graph encoding all antecedent information for all mentions is treated as an instance.

Cost Function. The cost function from the mention ranking model naturally extends to the tree case by summing over all decisions. Furthermore, in principle we can take the structure into account. However, we are not aware of any approaches which go beyond (variations of) Hamming loss (Hamming, 1950).

5 Experiments

We now evaluate model variants based on different latent structures on a large benchmark corpus. The aim of this section is to compare popular approaches to coreference only in terms of the structure they operate on, fixing preprocessing and feature set. In Section 6 we complement this comparison with a qualitative analysis of the influence of the structures on the output.

5.1 Data and Evaluation Metrics

The aim of our evaluation is to assess the effectiveness and competitiveness of the models implemented in our framework in a realistic coreference setting, i.e. without using gold information such as gold mentions. As all models we consider share the same preprocessing and features, this allows for a fair comparison of the individual structures.

We train, evaluate and analyze the models on the English data of the CoNLL-2012 shared task on multilingual coreference resolution (Pradhan et al., 2012). The shared task organizers provide the training/development/ test split. We use the 2802 training documents for training the models, and evaluate and analyze the models on the development set containing 343 documents. The 349 test set documents are only used for final evaluation.

We work in a setting that corresponds to the shared task's *closed track* (Pradhan et al., 2012). That is, we make use of the automatically created annotation layers (parse trees, NE information, ...) shipped with the data. As additional resources we use only WordNet 3.0 (Fellbaum, 1998) and the number/gender data of Bergsma and Lin (2006).

For evaluation we follow the practice of the CoNLL-2012 shared task and employ the reference implementation of the CoNLL scorer (Pradhan et al., 2014) which computes the popular evaluation metrics MUC (Vilain et al., 1995), B^3 (Bagga and Baldwin, 1998), $CEAF_e$ (Luo, 2005) and their average. The average is the metric for ranking the systems in the CoNLL shared tasks on coreference resolution (Pradhan et al., 2011; Pradhan et al., 2012).

5.2 Features

We employ a rich set of features frequently used in the literature (Ng and Cardie, 2002; Bengtson and Roth, 2008; Björkelund and Kuhn, 2014). The set consists of the following features:

- the mention type (name, def. noun, indef. noun, citation form of pronoun, demonstrative) of anaphor, antecedent and both,
- gender, number, semantic class, named entity class, grammatical function and length in words of anaphor, antecedent and both,
- semantic head, first/last/preceding/next token of anaphor, antecedent and both,
- distance between anaphor and antecedent in sentences,
- modifier agreement,
- whether anaphor and antecedent embed each other,
- whether there is a string match, head match or

an alias relation,

- whether anaphor and antecedent have the same speaker.

If the antecedent in the pair under consideration is m_0, i.e. the dummy mention, we do not extract any feature (Chang et al., 2012).

State-of-the-art models greatly benefit from feature conjunctions. Approaches for building such conjunctions include greedy extension (Björkelund and Kuhn, 2014), entropy-guided induction (Fernandes et al., 2014) and linguistically motivated heuristics (Durrett and Klein, 2013). We follow Durrett and Klein (2013) and conjoin every feature with each mention type feature.

5.3 Model Variants

We now consider several instantiations of the approaches discussed in the previous section in order of increasing complexity. These instantiations correspond to specific coreference models proposed in the literature. With the framework described in this paper, we are able to give a unified account of representing and learning these models. We always train on automatically predicted mentions.

We start with the mention pair model. To create training graphs, we employ a slight modification of the closest pair heuristic (Soon et al., 2001), which worked best in preliminary experiments. For each mention m_j which is in some coreference chain and has an antecedent m_i, we add an edge to m_i with label "+". For all k with $i < k < j$, we add an edge from m_j to m_k with label "−". If m_j does not have an antecedent, we add edges from m_j to m_k with label "−" for all $0 < k < j$. Compared to the heuristic of Soon et al. (2001), who only learn from anaphoric mentions, this improves precision. During testing, if for a mention m_j no pair (m_j, m_i) is deemed as coreferent, we consider the mention as not anaphoric. Otherwise, we employ *best-first clustering* and take the mention in the highest scoring pair as the antecedent of m_j (Ng and Cardie, 2002).

The mention ranking model tries to improve the mention pair model by capturing the competition between antecedents. We consider two variants of the mention ranking model, where each employs dummy mentions for anaphoricity determination. The first variant *Closest* (Denis and Baldridge, 2008) constrains the latent structures consistent with the gold annotation: for each mention, the closest antecedent is chosen as the gold antecedent. If the mention does not have any antecedent, we take the dummy mention m_0 as the antecedent. The second variant *Latent* (Chang et al., 2012) aims to learn from more meaningful antecedents by dropping the constraints, and therefore selecting the best-scoring antecedent (which may also be m_0) under the current model during training.

We view the antecedent tree model (Fernandes et al., 2014) as a natural extension of the mention ranking model. Instead of predicting an antecedent for each mention, we predict an entire tree of anaphor-antecedent pairs. This should yield more consistent entities. As in previous work we only consider the latent variant.

For the mention ranking model and for antecedent trees we use a cost function similar to previous work (Durrett and Klein, 2013; Fernandes et al., 2014). For a pair of mentions (m_j, m_i), we consider

$$c_{\text{pair}}(m_j, m_i) = \begin{cases} \lambda & i > 0 \text{ and} \\ & m_j, m_i \text{ are not coreferent,} \\ 2\lambda & i = 0 \text{ and } m_j \text{ is anaphoric,} \\ 0 & \text{otherwise,} \end{cases}$$

where $\lambda > 0$ will be tuned on development data.

Let $\hat{h}_i = (V_i, A_i, L_{A_i})$. c_{pair} is extended to a cost function for the whole latent structure $\hat{h}_i$ by

$$c(x, \hat{h}_i, \hat{h}_{\text{opt},i}, z) = \sum_{(m_j, m_k) \in A_i} c_{\text{pair}}(m_j, m_k).$$

The use of such a cost function is necessary to learn reasonable weights, since most automatically extracted mentions in the data are not anaphoric.

5.4 Experimental Setup

We evaluate the models on the development and the test sets. When evaluating on the test set, we train on the concatenation of the training and development set. After preliminary experiments with the ranking model with closest antecedents on the development set, we set the number of perceptron epochs to 5 and set $\lambda = 100$ in the cost function.

We assess statistical significance of the difference in F_1 score for two approaches via an approximate randomization test (Noreen, 1989). We say an improvement is *statistically significant* if $p < 0.05$.

Model	MUC			B^3			$CEAF_e$			Average F_1
	R	P	F_1	R	P	F_1	R	P	F_1	
CoNLL-2012 English development data										
Fernandes et al. (2014)	64.88	74.74	69.46	51.85	65.35	57.83	51.50	57.72	54.43	60.57
Björkelund and Kuhn (2014)	68.58	73.04	70.74	57.97	62.28	60.03	54.57	59.23	56.80	62.52
Mention Pair	66.68	71.71	69.10	53.57	62.44	57.67	52.56	53.87	53.21	59.99
Ranking: Closest	67.85	76.66	71.99*	55.33	65.45	59.97*	53.16	61.28	56.93*	62.96
Ranking: Latent	68.02	76.73	**72.11**$^{\diamond\times}$	55.61	66.91	**60.74**$^{\dagger\diamond}$	54.48	61.36	**57.72**$^{\dagger\diamond\times}$	**63.52**
Antecedent Trees	65.91	77.92	71.41	52.72	67.98	59.39	52.13	60.82	56.14	62.31
CoNLL-2012 English test data										
Fernandes et al. (2014)	65.83	75.91	70.51	51.55	65.19	57.58	50.82	57.28	53.86	60.65
Björkelund and Kuhn (2014)	67.46	74.30	70.72	54.96	62.71	58.58	52.27	59.40	55.61	61.63
Mention Pair	67.16	71.48	69.25	51.97	60.55	55.93	51.02	51.89	51.45	58.88
Ranking: Closest	67.96	76.61	72.03*	54.07	64.98	59.03*	51.45	59.02	54.97*	62.01
Ranking: Latent	68.13	76.72	**72.17**$^{\diamond}$	54.22	66.12	**59.58**$^{\dagger\diamond}$	52.33	59.47	**55.67**$^{\dagger\diamond}$	**62.47**
Antecedent Trees	65.79	78.04	71.39	50.92	67.76	58.15	50.55	58.34	54.17	61.24

Table 1: Results of different systems and model variants on CoNLL-2012 English development and test data. Models below the dashed lines are implemented in our framework. The best F_1 score results for each dataset and metric are boldfaced. * indicates significant improvements in F_1 score of *Ranking: Closest* compared to *Mention Pair*; † indicates significant improvements of *Ranking: Latent* compared to *Ranking: Closest*; ◇ indicates significant improvements of *Ranking: Latent* compared to *Antecedent Trees*; × indicates significant improvements of *Ranking: Latent* compared to Björkelund and Kuhn (2014). We do not perform significance tests on differences in average F_1 since this measure constitutes an average over other F_1 scores.

5.5 Results

Table 1 shows the result of all model configurations discussed in the previous section on CoNLL'12 English development and test data. In order to put the numbers into context, we also report the results of Björkelund and Kuhn (2014), who present a system that implements an antecedent tree model with non-local features. Their system is the highest-performing system on the CoNLL data which operates in a *closed track* setting. We also compare with Fernandes et al. (2014), the winning system of the CoNLL-2012 shared task (Pradhan et al., 2012)[3]. Both systems were trained on training data for evaluating on the development set, and on the concatenation of training and development data for evaluating on the test set.

Despite its simplicity, the mention pair model yields reasonable performance. The gap to Björkelund and Kuhn (2014) is roughly 2.8 points in average F_1 score on test data.

Compared to the mention pair model, the variants of the mention ranking model improve the results for all metrics, largely due to increased precision. Switching from regarding the closest antecedent as the gold antecedent to latent antecedents yields an improvement of roughly 0.5 points in average F_1. All improvements of the mention ranking model with closest antecedents compared to the mention pair model are statistically significant. Furthermore, with the exception of the differences in MUC F_1, all improvements are significant when switching from closest antecedents to latent antecedents.

The mention ranking model with latent an-

[3]We do not compare with the system of Durrett and Klein (2014) since it uses Wikipedia as an additional resource, and therefore does not work under the *closed track* setting. Its performance is 61.71 average F_1 (71.24 MUC F_1, 58.71 B^3 F_1 and 55.18 $CEAF_e$ F_1) on CoNLL-2012 English test data.

	Recall			Precision		
Model	**Errors**	**Max**	**% of Max**	**Errors**	**Max**	**% of Max**
Mention Pair	4867	14609	33%	4187	13585	31%
Ranking: Closest	4695		32%	3336	12932	26%
Ranking: Latent	4671		32%	3357	12951	26%
Antecedent Trees	4979		34%	3042	12358	25%

Table 2: Overview of recall and precision errors.

tecedents outperforms the state-of-the-art system by Björkelund and Kuhn (2014) by more than 0.8 points average F_1. These results show the competitiveness of a simple mention ranking architecture. Regarding the individual F_1 scores compared to Björkelund and Kuhn (2014), the improvements in the MUC and $CEAF_e$ metrics on development data are statistically significant. The improvements on test data are not statistically significant.

Using antecedent trees yields higher precision than using the mention ranking model. However, recall is much lower. The performance is similar to the antecedent tree models of Fernandes et al. (2014) and Björkelund and Kuhn (2014).

6 Analysis

The numbers discussed in the previous section do not give insights into where the models make different decisions. Are there specific linguistic classes of mention pairs where one model is superior to the other? How do the outputs differ? How can these differences be explained by different structures employed by the models?

In order to answer these questions, we need to perform a qualitative analysis of the differences in system output for the approaches. To do so, we employ the error analysis method presented in Martschat and Strube (2014). In this method, recall errors are extracted via comparing spanning trees of reference entities with system output. Edges in the spanning tree missing from the output are extracted as errors. For extracting precision errors, the roles of reference and system entities are switched. To define the spanning trees, we follow Martschat and Strube (2014) and use a notion based on Ariel's accessibility theory (Ariel, 1990) for reference entities, while we take system antecedent decisions for system entities.

6.1 Overview

We extracted all errors of the model variants described in the previous section on CoNLL-2012 English development data.

Table 2 gives an overview of all recall and precision errors. For each model variant the table shows the number of recall and precision errors, and the maximum number of errors[4]. The numbers confirm the findings obtained from Table 1: the ranking models beat the mention pair model largely due to fewer precision errors.

The antecedent tree model outputs more precise entities by establishing fewer coreference links: it makes fewer decisions and fewer precision errors than the other configurations, but at the expense of an increased number of recall errors.

The more sophisticated models make consistently fewer linking decisions than the mention pair model. We therefore hypothesize that the improvements in the numbers mainly stem from improved anaphoricity determination. The mention pair model handles anaphoricity determination implicitly: if for a mention m_j no pair (m_j, m_i) is deemed as coreferent, the model does not select an antecedent for m_j[5]. Since the mention ranking model allows to include the search for the best antecedent during prediction, we can explicitly model the anaphoricity decision, via including the dummy mention during search.

We now examine the errors in more detail to investigate this hypothesis. To do so, we will investi-

[4]For recall, the maximum number of errors is the number of errors made by a system that assigns each mention to its own entity. For precision, the maximum number of errors is the total number of anaphor-antecedent decisions made by the model.

[5]Initial experiments which included the dummy mention during learning for the mention pair model yielded worse results. This is arguably due to the large number of non-anaphoric mentions, which causes highly imbalanced training data.

Model	Name/noun			Anaphor pronoun			Remaining
	Both name	Mixed	Both noun	I/you/we	he/she	it/they	
Upper bound	3579	948	2063	2967	1990	2471	591
Mention Pair	815	657	1074	394	373	1005	549
Ranking: Closest	879	637	1221	348	247	806	557
Ranking: Latent	857	647	1158	370	251	822	566
Antecedent Trees	911	686	1258	441	247	863	572

Table 3: Recall errors of model variants on CoNLL-2012 English development data.

Model	Name/noun						Anaphor pronoun						Remaining	
	Both name		Mixed		Both noun		I/you/we		he/she		it/they			
	err.	corr.	err.	corr.	err.	corr.	err.	corr.	err.	corr.	err.	corr.	err.	corr.
Mention Pair	885	2673	83	79	1055	1098	836	2479	289	1546	864	1408	175	115
Ranking: Closest	587	2620	93	96	494	960	873	2521	324	1692	844	1510	121	97
Ranking: Latent	640	2664	92	102	567	1038	862	2461	318	1692	835	1594	42	43
Antecedent Trees	595	2628	57	82	442	924	836	2398	318	1691	757	1557	37	36

Table 4: Precision errors (err.) and correct links (corr.) of model variants on CoNLL-2012 English development data.

gate error classes, and compare the models in terms of how they handle these error classes. This is a practice common in the analysis of coreference resolution approaches (Stoyanov et al., 2009; Martschat and Strube, 2014). We distinguish between errors where both mentions are a proper name or a common noun, errors where the anaphor is a pronoun and the remaining errors.

Tables 3 and 4 summarize recall and precision errors for subcategories of these classes[6]. We now compare individual models.

6.2 Mention Ranking vs. Mention Pair

For pairs of proper names and pairs of common nouns, employing the ranking model instead of the mention pair model leads to a large decrease in precision errors, but an increase in recall errors. For pronouns and *mixed* pairs, we can observe decreases in recall errors and slight increases in precision errors, except for *it/they*, where both recall precision errors decrease.

We can attribute the largest differences to determining anaphoricity: in 82% of all precision errors between two proper names made by the mention pair model, but not by the ranking model, the mention appearing later in the text is non-anaphoric. The ranking model correctly determines this. Similar numbers hold for common noun pairs.

While most nouns and names are not anaphoric, most pronouns are. Hence, determining anaphoricity is less of an issue here. From the resolved *it/they* recall errors of the ranking model compared to the mention pair model, we can attribute 41% to better antecedent selection: the mention pair model decided on a wrong antecedent. The ranking model, however, was able to leverage the competition between the antecedents to decide on a correct antecedent. The remaining 59% stem from selecting a correct antecedent for pronouns that were classified as non-anaphoric by the mention pair model. We observe similar trends for the other pronoun classes.

Overall, the majority of error reduction can be attributed to improved determination of anaphoricity, which can be modeled *structurally* in the mention ranking model (we do not use any features when a dummy mention is involved, therefore non-anaphoricity decisions always get the score 0). However, for pronoun resolution, where there are

[6]For the pronoun subcategories, we map each pronoun to its canonical form. For example, we map *him* to *he*.

many competing compatible antecedents for a mention, the model is able to learn better weights by leveraging the competition. These findings suggest that extending the mention pair model to explicitly determine anaphoricity should improve results especially for non-pronominal coreference.

6.3 Latent Antecedent vs. Closest Antecedent

Using latent instead of closest antecedents leads to fewer recall errors and more precision errors for non-pronominal coreference. Pronoun resolution recall errors slightly increase, while precision errors slightly decrease.

While these changes are minor, there is a large reduction in the *remaining* precision errors. Most of these correspond to predictions which are considered very difficult, such as links between a proper name anaphor and a pronoun antecedent (Bengtson and Roth, 2008). Via latent antecedents, the model can avoid learning from the most unreliable pairs.

6.4 Antecedent Trees vs. Ranking

Compared to the ranking model with latent antecedents, the antecedent tree model commits consistently more recall errors and fewer precision errors. This is partly due to the fact that the antecedent tree model also predicts fewer links between mentions than the other models. The only exception is *he/she*, where there is not much of a difference.

The only difference between the ranking model with latent antecedents and the antecedent tree model is that weights are updated document-wise for antecedent trees, while they are updated per anaphor for the ranking model. This leads to more precise predictions, at the expense of recall.

6.5 Summary

Our analysis shows that the mention ranking model mostly improves precision over the mention pair model. For non-pronominal coreference, the improvements can be mainly attributed to improved anaphoricity determination. For pronoun resolution, both anaphoricity determination and capturing antecedent competition lead to improved results. Employing latent antecedents during training mainly helps in resolving very difficult cases. Due to the update strategy, employing antecedent trees leads to a more precision-oriented approach, which significantly improves precision at the expense of recall.

7 Beyond Pairwise Predictions

In this paper we concentrated on representing and analyzing the most prevalent approaches to coreference resolution, which are based on predicting whether pairs of mentions are coreferent. Hence, we choose graphs as latent structures and let the feature functions factor over edges in the graph, which correspond to pairs of mentions.

However, entity-based approaches (Rahman and Ng, 2011; Stoyanov and Eisner, 2012; Lee et al., 2013, inter alia) obtain coreference chains by predicting whether *sets* of mentions are coreferent, going beyond pairwise predictions. While a detailed discussion of such approaches is beyond the scope of this paper, we now briefly describe how we can generalize the proposed framework to accommodate for such approaches.

When viewing coreference resolution as prediction of latent structures, entity-based models operate on structures that relate sets of mentions to each other. This can be expressed by *hypergraphs*, which are graphs where edges can link more than two nodes. Hypergraphs have already been used to model coreference resolution (Cai and Strube, 2010; Sapena, 2012).

To model entity-based approaches, we extend the valid latent structures to labeled directed hypergraphs. These are tuples $h = (V, A, L_A)$, where

- the set of nodes are the mentions, $V = M_x^0$,
- the set of edges $A \subseteq 2^V \times 2^V$ consists of directed hyperedges linking two sets of mentions,
- $L_A \colon A \rightarrow L$ assigns a label $\ell \in L$ to each edge. L is a finite set of labels.

For example, the entity-mention model (Yang et al., 2008) predicts coreference in a left-to-right fashion. For each anaphor m_j, it considers the set

$$E_j \subseteq 2^{\{m_0,\ldots,m_{j-1}\}}$$

of preceding partial entities that have been established so far (such as $e = \{m_1, m_3, m_6\}$). In terms of our framework, substructures for this approach are hypergraphs with hyperedges $(\{m_j\}, e)$ for $e \in E_j$, encoding the decision to which partial entity m_j refers.

The definitions of features and the decoding problem carry over from the graph-based framework (we drop the edge factorization assumption for features). Learning requires adaptations to cope with the dependency between coreference decisions. For example, for the entity-mention model, establishing that an anaphor m_j refers to a partial entity e influences the search space for decisions for anaphors m_k with $k > j$. We leave a more detailed discussion to future work.

8 Related Work

The main contributions of this paper are a framework for representing coreference resolution approaches and a systematic comparison of main coreference approaches in this framework.

Our representation framework generalizes approaches to coreference resolution which employed specific latent structures for representation, such as latent antecedents (Chang et al., 2012) and antecedent trees (Fernandes et al., 2014). We give a unified representation of such approaches and show that seemingly disparate approaches such as the mention pair model also fit in a framework based on latent structures.

Only few studies systematically compare approaches to coreference resolution. Most previous work highlights the improved expressive power of the presented model by a comparison to a mention pair baseline (Culotta et al., 2007; Denis and Baldridge, 2008; Cai and Strube, 2010).

Rahman and Ng (2011) consider a series of models with increasing expressiveness, ranging from a mention pair to a cluster-ranking model. However, they do not develop a unified framework for comparing approaches, and their analysis is not qualitative. Fernandes et al. (2014) compare variations of antecedent tree models, including different loss functions and a version with a fixed structure. They only consider antecedent trees and also do not provide a qualitative analysis. Kummerfeld and Klein (2013) and Martschat and Strube (2014) present a large-scale qualitative comparison of coreference systems, but they do not investigate the influence of the latent structures the systems operate on. Furthermore, the systems in their studies differ in terms of mention extraction and feature sets.

9 Conclusions

We observed that many approaches to coreference resolution can be uniformly represented by the latent structure they operate on. We devised a framework that accounts for such structures, and showed how we can express the mention pair model, the mention ranking model and antecedent trees in this framework.

An evaluation of the models on CoNLL-2012 data showed that all models yield competitive results. While antecedent trees give results with the highest precision, a mention ranking model with latent antecedent performs best, obtaining state-of-the-art results on CoNLL-2012 data.

An analysis based on the method of Martschat and Strube (2014) highlights the strengths of the mention ranking model compared to the mention pair model: it is able to structurally model anaphoricity determination and antecedent competition, which leads to improvements in precision for non-pronominal coreference resolution, and in recall for pronoun resolution. The effect of latent antecedents is negligible and has a large effect only on very difficult cases of coreference.

The flexibility of the framework, toolkit and analysis methods presented in this paper helps researchers to devise, analyze and compare representations for coreference resolution.

Acknowledgments

This work has been funded by the Klaus Tschira Foundation, Heidelberg, Germany. The first author has been supported by a HITS PhD scholarship. We thank the anonymous reviewers and our colleagues Benjamin Heinzerling, Yufang Hou and Nafise Moosavi for feedback on earlier drafts of this paper. Furthermore, we are grateful to Anders Björkelund for helpful comments on cost functions.

References

Mira Ariel. 1990. *Accessing Noun Phrase Antecedents*. Routledge, London, U.K.; New York, N.Y.

Amit Bagga and Breck Baldwin. 1998. Algorithms for scoring coreference chains. In *Proceedings of the 1st International Conference on Language Resources and Evaluation,* Granada, Spain, 28–30 May 1998, pages 563–566.

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,* Waikiki, Honolulu, Hawaii, 25–27 October 2008, pages 294–303.

Shane Bergsma and Dekang Lin. 2006. Bootstrapping path-based pronoun resolution. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics,* Sydney, Australia, 17–21 July 2006, pages 33–40.

Anders Björkelund and Jonas Kuhn. 2014. Learning structured perceptrons for coreference resolution with latent antecedents and non-local features. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers),* Baltimore, Md., 22–27 June 2014, pages 47–57.

Jie Cai and Michael Strube. 2010. End-to-end coreference resolution via hypergraph partitioning. In *Proceedings of the 23rd International Conference on Computational Linguistics,* Beijing, China, 23–27 August 2010, pages 143–151.

Kai-Wei Chang, Rajhans Samdani, Alla Rozovskaya, Mark Sammons, and Dan Roth. 2012. Illinois-Coref: The UI system in the CoNLL-2012 shared task. In *Proceedings of the Shared Task of the 16th Conference on Computational Natural Language Learning,* Jeju Island, Korea, 12–14 July 2012, pages 113–117.

Michael Collins. 2002. Discriminative training methods for Hidden Markov Models: Theory and experiments with perceptron algorithms. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing,* Philadelphia, Penn., 6–7 July 2002, pages 1–8.

Koby Crammer, Ofer Dekel, Joseph Keshet, Shai Shalev-Shwartz, and Yoram Singer. 2006. Online passive-aggressive algorithms. *Journal of Machine Learning Research*, 7:551–585.

Aron Culotta, Michael Wick, and Andrew McCallum. 2007. First-order probabilistic models for coreference resolution. In *Proceedings of Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics,* Rochester, N.Y., 22–27 April 2007, pages 81–88.

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,* Waikiki, Honolulu, Hawaii, 25–27 October 2008, pages 660–669.

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,* Seattle, Wash., 18–21 October 2013, pages 1971–1982.

Greg Durrett and Dan Klein. 2014. A joint model for entity analysis: Coreference, typing, and linking. *Transactions of the Association of Computational Linguistics*, 2:477–490.

Charles Elkan. 2001. The foundations of cost-sensitive learning. In *Proceedings of the 17th International Joint Conference on Artificial Intelligence,* Seattle, Wash., 4–10 August, 2001, pages 973–978.

Christiane Fellbaum, editor. 1998. *WordNet: An Electronic Lexical Database*. MIT Press, Cambridge, Mass.

Eraldo Fernandes, Cícero dos Santos, and Ruy Milidiú. 2014. Latent trees for coreference resolution. *Computational Linguistics*, 40(4):801–835.

Jenny Rose Finkel and Christopher Manning. 2008. Enforcing transitivity in coreference resolution. In *Companion Volume to the Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics,* Columbus, Ohio, 15–20 June 2008, pages 45–48.

Peter Geibel and Fritz Wysotzk. 2003. Perceptron based learning with example dependent and noisy costs. In *Proceedings of the 20th International Conference on Machine Learning,* Washington, D.C., 21–24 August 2003, pages 218–225.

Richard W. Hamming. 1950. Error detecting and error correcting codes. *Bell System Technical Journal*, 26(2):147–160.

Jerry R. Hobbs. 1976. Pronoun resolution. Technical Report 76-1, Dept. of Computer Science, City College, City University of New York.

Dan Klein and Christopher D. Manning. 2001. Parsing and hypergraphs. In *Proceedings of the Seventh International Workshop on Parsing Technologies (IWPT-2001), 17-19 October 2001, Beijing, China*, pages 123–134.

Manfred Klenner. 2007. Enforcing consistency on coreference sets. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing,* Borovets, Bulgaria, 27–29 September 2007, pages 323–328.

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, Wash., 18–21 October 2013, pages 265–277.

Shalom Lappin and Herbert J. Leass. 1994. An algorithm for pronominal anaphora resolution. *Computational Linguistics*, 20(4):535–561.

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

Adam Lopez. 2009. Translation as weighted deduction. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics,* Athens, Greece, 30 March – 3 April 2009, pages 532–540.

Xiaoqiang Luo. 2005. On coreference resolution performance metrics. In *Proceedings of the Human Language Technology Conference and the 2005 Conference on Empirical Methods in Natural Language Processing,* Vancouver, B.C., Canada, 6–8 October 2005, pages 25–32.

Sebastian Martschat and Michael Strube. 2014. Recall error analysis for coreference resolution. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing,* Doha, Qatar, 25–29 October 2014, pages 2070–2081.

Vincent Ng and Claire Cardie. 2002. Improving machine learning approaches to coreference resolution. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics,* Philadelphia, Penn., 7–12 July 2002, pages 104–111.

Eric W. Noreen. 1989. *Computer-Intensive Methods for Testing Hypotheses. An Introduction*. Wiley, New York.

Sameer Pradhan, Lance Ramshaw, Mitchell Marcus, Martha Palmer, Ralph Weischedel, and Nianwen Xue. 2011. CoNLL-2011 Shared Task: Modeling unrestricted coreference in OntoNotes. In *Proceedings of the Shared Task of the 15th Conference on Computational Natural Language Learning,* Portland, Oreg., 23–24 June 2011, pages 1–27.

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Olga Uryupina, and Yuchen Zhang. 2012. CoNLL-2012 Shared Task: Modeling multilingual unrestricted coreference in OntoNotes. In *Proceedings of the Shared Task of the 16th Conference on Computational Natural Language Learning,* Jeju Island, Korea, 12–14 July 2012, pages 1–40.

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 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers),* Baltimore, Md., 22–27 June 2014, pages 30–35.

Altaf Rahman and Vincent Ng. 2011. Narrowing the modeling gap: A cluster-ranking approach to coreference resolution. *Journal of Artificial Intelligence Research*, 40:469–521.

Emili Sapena. 2012. *A constraint-based hypergraph partitioning approach to coreference resolution*. Ph.D. thesis, Departament de Llenguatges i Sistemes Informàtics, Universitat Politècnica de Catalunya, Barcelona, Spain.

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.

Veselin Stoyanov and Jason Eisner. 2012. Easy-first coreference resolution. In *Proceedings of the 24th International Conference on Computational Linguistics,* Mumbai, India, 8–15 December 2012, pages 2519–2534.

Veselin Stoyanov, Nathan Gilbert, Claire Cardie, and Ellen Riloff. 2009. Conundrums in noun phrase coreference resolution: Making sense of the state-of-the-art. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing,* Singapore, 2–7 August 2009, pages 656–664.

Xu Sun, Takuya Matsuzaki, Daisuke Okanohara, and Jun'ichi Tsujii. 2009. Latent variable perceptron algorithm for structured classification. In *Proceedings of the 21th International Joint Conference on Artificial Intelligence,* Pasadena, Cal., 14–17 July 2009, pages 1236–1242.

Marc Vilain, John Burger, John Aberdeen, Dennis Connolly, and Lynette Hirschman. 1995. A model-theoretic coreference scoring scheme. In *Proceedings of the 6th Message Understanding Conference (MUC-6)*, pages 45–52, San Mateo, Cal. Morgan Kaufmann.

Xiaofeng Yang, Jian Su, Jun Lang, Chew Lim Tan, Ting Liu, and Sheng Li. 2008. An entity-mention model for coreference resolution with Inductive Logic Programming. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies,* Columbus, Ohio, 15–20 June 2008, pages 843–851.

Chun-Nam John Yu and Thorsten Joachims. 2009. Learning structural SVMs with latent variables. In *Proceedings of the 26th International Conference on Machine Learning,* Montréal, Québec, Canada, 14–18 June 2009, pages 1169–1176.