Coreference Resolution in a Modular, Entity-Centered Model

Aria Haghighi
Computer Science Division
University of California, Berkeley
aria42@cs.berkeley.edu

Dan Klein
Computer Science Division
University of California, Berkeley
klein@cs.berkeley.edu

Abstract

Coreference resolution is governed by syntactic, semantic, and discourse constraints. We present a generative, model-based approach in which each of these factors is modularly encapsulated and learned in a primarily unsupervised manner. Our semantic representation first hypothesizes an underlying set of latent *entity types*, which generate specific entities that in turn render individual mentions. By sharing lexical statistics at the level of abstract entity types, our model is able to substantially reduce semantic compatibility errors, resulting in the best results to date on the complete end-to-end coreference task.

1 Introduction

Coreference systems exploit a variety of information sources, ranging from syntactic and discourse constraints, which are highly configurational, to semantic constraints, which are highly contingent on lexical meaning and world knowledge. Perhaps because configurational features are inherently easier to learn from small data sets, past work has often emphasized them over semantic knowledge.

Of course, all state-of-the-art coreference systems have needed to capture semantic compatibility to some degree. As an example of nominal headword compatibility, a "president" can be a "leader" but cannot be not an "increase." Past systems have often computed the compatibility of specific headword pairs, extracted either from lexical resources (Ng, 2007; Bengston and Roth, 2008; Rahman and Ng, 2009), web statistics (Yang et al., 2005), or surface syntactic patterns (Haghighi and Klein, 2009). While the pairwise approach has high precision, it is neither realistic nor scalable to explicitly enumerate all pairs of compatible word pairs. A more compact approach has been to rely on named-entity recognition (NER) systems to give coarse-grained entity types for each mention (Soon et al., 1999; Ng and Cardie, 2002). Unfortunately, current systems use small inventories of types and so provide little constraint. In general, coreference errors in state-of-the-art systems are frequently due to poor models of semantic compatibility (Haghighi and Klein, 2009).

In this work, we take a primarily unsupervised approach to coreference resolution, broadly similar to Haghighi and Klein (2007), which addresses this issue. Our generative model exploits a large inventory of distributional entity types, including standard NER types like PERSON and ORG, as well as more refined types like WEAPON and VEHICLE. For each type, distributions over typical heads, modifiers, and governors are learned from large amounts of unlabeled data, capturing type-level semantic information (e.g. "spokesman" is a likely head for a PERSON). Each entity inherits from a type but captures entity-level semantic information (e.g. "giant" may be a likely head for the Microsoft entity but not all ORGs). Separately from the type-entity semantic module, a log-linear discourse model captures configurational effects. Finally, a mention model assembles each textual mention by selecting semantically appropriate words from the entities and types.

Despite being almost entirely unsupervised, our model yields the best reported end-to-end results on a range of standard coreference data sets.

2 Key Abstractions

The key abstractions of our model are illustrated in Figure 1 and described here.

Mentions: A mention is an observed textual reference to a latent real-world entity. Mentions are as-

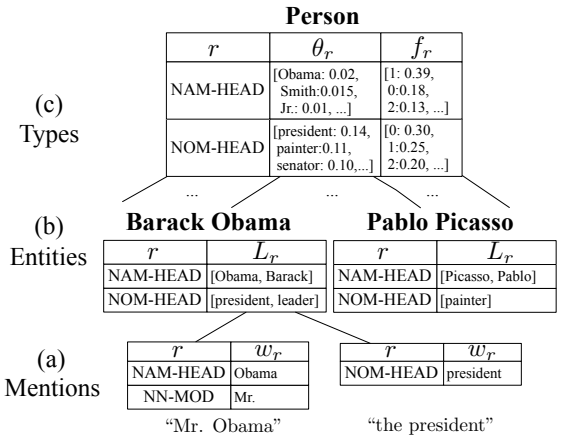

Figure 1: The key abstractions of our model (Section 2). (a) Mentions map properties (r) to words (w_r). (b) Entities map properties (r) to word lists (L_r). (c) Types map properties (r) to distributions over property words (θ_r) and the fertilities of those distributions (f_r). For (b) and (c), we only illustrate a subset of the properties.

sociated with nodes in a parse tree and are typically realized as NPs. There are three basic forms of mentions: proper (denoted NAM), nominal (NOM), and pronominal (PRO). We will often describe proper and nominal mentions together as *referring* mentions.

We represent each mention M as a collection of key-value pairs. The keys are called *properties* and the values are words. For example, the left mention in Figure 1(a) has a proper head property, denoted NAM-HEAD, with value "Obama." The set of properties we consider, denoted $\mathcal{R}$, includes several varieties of heads, modifiers, and governors (see Section 5.2 for details). Not every mention has a value for every property.

Entities: An entity is a specific individual or object in the world. Entities are always latent in text. Where a mention has a single word for each property, an entity has a *list* of signature words. Formally, entities are mappings from properties $r \in \mathcal{R}$ to lists L_r of "canonical" words which that entity uses for that property. For instance in Figure 1(b), the list of nominal heads for the Barack Obama entity includes "president."

Types: Coreference systems often make a mention / entity distinction. We extend this hierarchy to include *types*, which represent classes of entities (PERSON, ORGANIZATION, and so on). Types allow the sharing of properties across entities and mediate the generation of entities in our model (Section 3.1). See Figure 1(c) for a concrete example.

We represent each type τ as a mapping between properties r and pairs of multinomials (θ_r, f_r). Together, these distributions control the lists L_r for entities of that type. θ_r is a unigram distribution of words that are semantically licensed for property r. f_r is a "fertility" distribution over the integers that characterizes entity list lengths. For example, for the type PERSON, θ_r for proper heads is quite flat (there are many last names) but f_r is peaked at 1 (people have a single last name).

3 Generative Model

We now describe our generative model. At the parameter level, we have one parameter group for the types $\boldsymbol{\tau} = (\phi, \tau_1, \ldots, \tau_t)$, where ϕ is a multinomial prior over a fixed number t of types and the $\{\tau_i\}$ are the parameters for each individual type, described in greater detail below. A second group comprises log-linear parameters $\boldsymbol{\pi}$ over discourse choices, also described below. Together, these two groups are drawn according to $P(\boldsymbol{\tau}|\boldsymbol{\lambda})P(\boldsymbol{\pi}|\sigma^2)$, where $\boldsymbol{\lambda}$ and σ^2 are a small number of scalar hyper-parameters described in Section 4.

Conditioned on the parameters $(\boldsymbol{\tau}, \boldsymbol{\pi})$, a document is generated as follows: A *semantic module* generates a sequence $\mathbf{E}$ of entities. $\mathbf{E}$ is in principle infinite, though during inference only a finite number are ever instantiated. A *discourse module* generates a vector $\mathbf{Z}$ which assigns an entity index Z_i to each mention position i. Finally, a *mention generation module* independently renders the sequence of mentions ($\mathbf{M}$) from their underlying entities. The syntactic position and structure of mentions are treated as observed, including the mention forms (pronominal, etc.). We use $\mathbf{X}$ to refer to this ungenenerated information. Our model decomposes as follows:

$$P(\mathbf{E}, \mathbf{Z}, \mathbf{M}|\boldsymbol{\tau}, \boldsymbol{\pi}, \mathbf{X}) =$$
$$P(\mathbf{E}|\boldsymbol{\tau}) \text{ [Semantic, Section 3.1]}$$
$$P(\mathbf{Z}|\boldsymbol{\pi}, \mathbf{X}) \text{ [Discourse, Section 3.2]}$$
$$P(\mathbf{M}|\mathbf{Z}, \mathbf{E}, \boldsymbol{\tau}) \text{ [Mention, Section 3.3]}$$

We detail each of these components in subsequent sections.

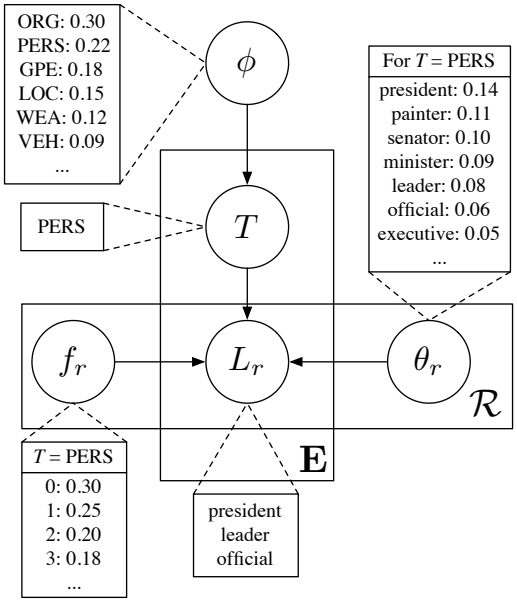

Figure 2: Depiction of the entity generation process (Section 3.1). Each entity draws a type (T) from ϕ, and, for each property $r \in \mathcal{R}$, forms a word list (L_r) by choosing a length from T's f_r distribution and then independently drawing that many words from T's θ_r distribution. Example values are shown for the person type and the nominal head property (NOM-HEAD).

3.1 Semantic Module

The semantic module is responsible for generating a sequence of entities. Each entity E is generated independently and consists of a type indicator T, as well as a collection $\{L_r\}_{r \in \mathcal{R}}$ of word lists for each property. These elements are generated as follows:

Entity Generation

Draw entity type $T \sim \phi$

For each mention property $r \in \mathcal{R}$,

Fetch $\{(f_r, \theta_r)\}$ for τ_T

Draw word list length $|L_r| \sim f_r$

Draw $|L_r|$ words from $w \sim \theta_r$

See Figure 2 for an illustration of this process. Each word list L_r is generated by first drawing a list length from f_r and then independently populating that list from the property's word distribution θ_r.[1] Past work has employed broadly similar distributional models for unsupervised NER of proper mentions (Collins and Singer, 1999; Elsner et al., 2009). However, to our knowledge, this is the first work to incorporate such a model into an entity reference process.

[1] There is one exception: the sizes of the proper and nominal head property lists are jointly generated, but their word lists are still independently populated.

3.2 Discourse Module

The discourse module is responsible for choosing an entity to evoke at each of the n mention positions. Formally, this module generates an entity assignment vector $\mathbf{Z} = (Z_1, \ldots, Z_n)$, where Z_i indicates the entity index for the ith mention position. Most linguistic inquiry characterizes NP anaphora by the pairwise relations that hold between a mention and its antecedent (Hobbs, 1979; Kehler et al., 2008). Our discourse module utilizes this pairwise perspective to define each Z_i in terms of an intermediate "antecedent" variable A_i. A_i either points to a previous antecedent mention position ($A_i < i$) and "steals" its entity assignment or begins a new entity ($A_i = i$). The choice of A_i is parametrized by affinities $s_{\boldsymbol{\pi}}(i, j; \mathbf{X})$ between mention positions i and j. Formally, this process is described as:

Entity Assignment

For each mention position, $i = 1, \ldots, n$,

Draw antecedent position $A_i \in \{1, \ldots, i\}$:

$$P(A_i = j|X) \propto s_{\boldsymbol{\pi}}(i, j; X)$$

$$Z_i = \begin{cases} Z_{A_i}, \text{ if } A_i < i \\ K + 1, \text{ otherwise} \end{cases}$$

Here, K denotes the number of entities allocated in the first i-1 mention positions. This process is an instance of the sequential distance-dependent Chinese Restaurant Process (DD-CRP) of Blei and Frazier (2009). During inference, we variously exploit both the A and Z representations (Section 4).

For nominal and pronoun mentions, there are several well-studied anaphora cues, including centering (Grosz et al., 1995), nearness (Hobbs, 1978), and deterministic constraints, which have all been utilized in prior coreference work (Soon et al., 1999; Ng and Cardie, 2002). In order to combine these cues, we take a log-linear, feature-based approach and parametrize $s_{\boldsymbol{\pi}}(i, j; X) = \exp\{\boldsymbol{\pi}^\top \mathbf{f}_{\mathbf{X}}(i, j)\}$, where $\mathbf{f}_{\mathbf{X}}(i, j)$ is a feature vector over mention positions i and j, and $\boldsymbol{\pi}$ is a parameter vector; the features may freely condition on $\mathbf{X}$. We utilize the following features between a mention and an an-

tecedent: tree distance, sentence distance, and the syntactic positions (subject, object, and oblique) of the mention and antecedent. Features for starting a new entity include: a definiteness feature (extracted from the mention's determiner), the top CFG rule of the mention parse node, its syntactic role, and a bias feature. These features are conjoined with the mention form (nominal or pronoun). Additionally, we restrict pronoun antecedents to the current and last two sentences, and the current and last three sentences for nominals. Additionally, we disallow nominals from having direct pronoun antecedents.

In addition to the above, if a mention is in a deterministic coreference configuration, as defined in Haghighi and Klein (2009), we force it to take the required antecedent. In general, antecedent affinities learn to prefer close antecedents in prominent syntactic positions. We also learn that new entity nominals are typically indefinite or have SBAR complements (captured by the CFG feature).

In contrast to nominals and pronouns, the choice of entity for a proper mention is governed more by entity frequency than antecedent distance. We capture this by setting $s_{\boldsymbol{\pi}}(i, j; \mathbf{X})$ in the proper case to 1 for past positions and to a fixed α otherwise. [2]

3.3 Mention Module

Once the semantic module has generated entities and the discourse model selects entity assignments, each mention M_i generates word values for a set of observed properties R_i:

Mention Generation

For each mention M_i, $i = 1, \ldots, n$

Fetch $(T, \{L_r\}_{r \in \mathcal{R}})$ from E_{Z_i}

Fetch $\{(f_r, \theta_r)\}_{r \in \mathcal{R}}$ from τ_T

For $r \in R_i$:

$w \sim (1 - \alpha_r)\textsc{Uniform}(L_r) + (\alpha_r)\theta_r$

For each property r, there is a hyper-parameter α_r which interpolates between selecting a word from the entity list L_r and drawing from the underlying type property distribution θ_r. Intuitively, a small value of α_r indicates that an entity prefers to re-use a small number of words for property r. This is typically the case for proper and nominal heads as well as modifiers. At the other extreme, setting α_r to 1 indicates the property isn't particular to the entity itself, but rather only on its type. We set α_r to 1 for pronoun heads as well as for the governor of the head properties.

[2] As Blei and Frazier (2009) notes, when marginalizing out the A_i in this trivial case, the DD-CRP reduces to the traditional CRP (Pitman, 2002), so our discourse model roughly matches Haghighi and Klein (2007) for proper mentions.

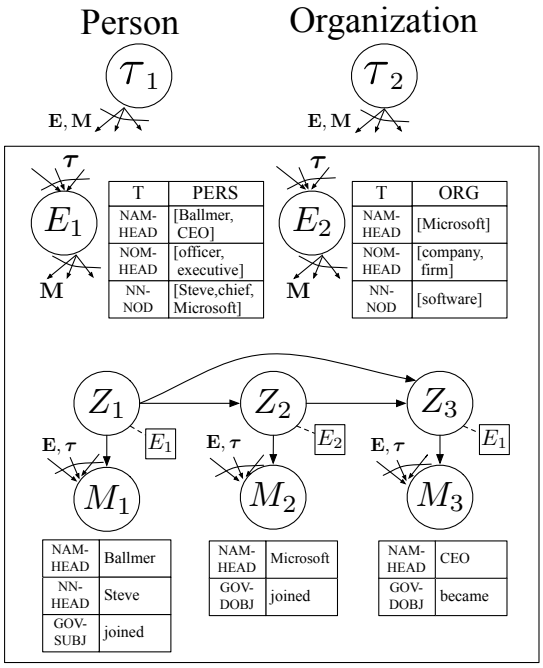

Figure 3: Depiction of the discourse module (Section 3.2); each random variable is annotated with an example value. For each mention position, an entity assignment (Z_i) is made. Conditioned on entities (E_{Z_i}), mentions (M_i) are rendered (Section 3.3). The $\overset{\mathbf{Y}}{\curlywedge}$ symbol denotes that a random variable is the parent of all $\mathbf{Y}$ random variables.

4 Learning and Inference

Our learning procedure involves finding parameters and assignments which are likely under our model's posterior distribution $P(\mathbf{E}, \mathbf{Z}, \boldsymbol{\tau}, \boldsymbol{\pi} | \mathbf{M}, \mathbf{X})$. The model is modularized in such a way that running EM on all variables simultaneously would be very difficult. Therefore, we adopt a variational approach which optimizes various subgroups of the variables in a round-robin fashion, holding approximations to the others fixed. We first describe the variable groups, then the updates which optimize them in turn.

Decomposition: We decompose the entity vari-

ables $\mathbf{E}$ into types, $\mathbf{T}$, one for each entity, and word lists, $\mathbf{L}$, one for each entity and property. We decompose the mentions $\mathbf{M}$ into *referring* mentions (propers and nominals), $\mathbf{M}^r$, and pronominal mentions, $\mathbf{M}^p$ (with sizes n_r and n_p respectively). The entity assignments $\mathbf{Z}$ are similarly divided into $\mathbf{Z}^r$ and $\mathbf{Z}^p$ components. For pronouns, rather than use $\mathbf{Z}^p$, we instead work with the corresponding antecedent variables, denoted $\mathbf{A}^p$, and marginalize over antecedents to obtain $\mathbf{Z}^p$.

With these variable groups, we would like to approximation our model posterior $P(\mathbf{T}, \mathbf{L}, \mathbf{Z}^r, \mathbf{A}^p, \boldsymbol{\tau}, \boldsymbol{\pi} | \mathbf{M}, \mathbf{X})$ using a simple factored representation. Our variational approximation takes the following form:

$$Q(\mathbf{T}, \mathbf{L}, \mathbf{Z}^r, \mathbf{A}^p, \boldsymbol{\tau}, \boldsymbol{\pi}) = \delta_r(\mathbf{Z}^r, \mathbf{L}) \left(\prod_{k=1}^{n} q_k(T_k) \right) \left(\prod_{i=1}^{n_p} r_i(A_i^p) \right) \delta_s(\boldsymbol{\tau}) \delta_d(\boldsymbol{\pi})$$

We use a mean field approach to update each of the RHS factors in turn to minimize the KL-divergence between the current variational posterior and the true model posterior. The δ_r, δ_s, and δ_d factors place point estimates on a single value, just as in hard EM. Updating these factors involves finding the value which maximizes the model (expected) log-likelihood under the other factors. For instance, the δ_s factor is a point estimate of the type parameters, and is updated with:[3]

$$\delta_s(\boldsymbol{\tau}) \leftarrow \operatorname*{argmax}_{\boldsymbol{\tau}} \mathbb{E}_{Q_{-\delta_s}} \ln P(\mathbf{E}, \mathbf{Z}, \mathbf{M}, \boldsymbol{\tau}, \boldsymbol{\pi}) \quad (1)$$

where $Q_{-\delta_s}$ denotes all factors of the variational approximation except for the factor being updated. The r_i (pronoun antecedents) and q_k (type indicator) factors maintain a soft approximation and so are slightly more complex. For example, the r_i factor update takes the standard mean field form:

$$r_i(A_i^p) \propto \exp\{\mathbb{E}_{Q_{-r_i}} \ln P(\mathbf{E}, \mathbf{Z}, \mathbf{M}, \boldsymbol{\tau}, \boldsymbol{\pi})\} \quad (2)$$

We briefly describe the update for each additional factor, omitting details for space.

Updating type parameters $\delta_s(\boldsymbol{\tau})$: The type parameters $\boldsymbol{\tau}$ consist of several multinomial distributions which can be updated by normalizing expected counts as in the EM algorithm. The prior $P(\boldsymbol{\tau}|\boldsymbol{\lambda})$ consists of several finite Dirichlet draws for each multinomial, which are incorporated as pseudocounts.[4] Given the entity type variational posteriors $\{q_k(\cdot)\}$, as well as the point estimates of the $\mathbf{L}$ and $\mathbf{Z}^r$ elements, we obtain expected counts from each entity's attribute word lists and referring mention usages.

Updating discourse parameters $\delta_d(\boldsymbol{\pi})$: The learned parameters for the discourse module rely on pairwise antecedent counts for assignments to nominal and pronominal mentions.[5] Given these expected counts, which can be easily obtained from other factors, the update reduces to a weighted maximum entropy problem, which we optimize using LBFGS. The prior $P(\boldsymbol{\pi}|\sigma^2)$ is a zero-centered normal distribution with shared diagonal variance σ^2, which is incorporated via L2 regularization during optimization.

Updating referring assignments and word lists $\delta_r(\mathbf{Z}^r, \mathbf{L})$: The word lists are usually concatenations of the words used in nominal and proper mentions and so are updated together with the assignments for those mentions. Updating the $\delta_r(\mathbf{Z}^r, \mathbf{L})$ factor involves finding the referring mention entity assignments, $\mathbf{Z}^r$, and property word lists $\mathbf{L}$ for instantiated entities which maximize $\mathbb{E}_{Q_{-\delta_r}} \ln P(\mathbf{T}, \mathbf{L}, \mathbf{Z}^r, \mathbf{A}^p, \mathbf{M}, \boldsymbol{\tau}, \boldsymbol{\pi})$. We actually only need to optimize over $\mathbf{Z}^r$, since for any $\mathbf{Z}^r$, we can compute the optimal set of property word lists $\mathbf{L}$. Essentially, for each entity we can compute the L_r which optimizes the probability of the referring mentions assigned to the entity (indicated by $\mathbf{Z}^r$). In practice, the optimal L_r is just the set of property words in the assigned mentions. Of course enumerating and scoring all $\mathbf{Z}^r$ hypotheses is intractable, so we instead utilize a left-to-right sequential beam search. Each partial hypothesis is an assignment to a prefix of mention positions and is scored as though it were a complete hypothesis. Hypotheses are extended via adding a new mention to an existing entity or creating a new one. For our experiments, we limited the number of hypotheses on the beam to the top fifty and did not notice an improvement in model score from increasing beam size.

[3] Of course during learning, the argmax is performed over the entire document collection, rather than a single document.

[4] See software release for full hyper-parameter details.

[5] Propers have no learned discourse parameters.

Updating pronominal antecedents $r_i(A_i^p)$ and entity types $q_k(T_k)$: These updates are straightforward instantiations of the mean-field update (2).

To produce our final coreference partitions, we assign each referring mention to the entity given by the δ_r factor and each pronoun to the most likely entity given by the r_i.

4.1 Factor Staging

In order to facilitate learning, some factors are initially set to fixed heuristic values and only learned in later iterations. Initially, the assignment factors δ_r and $\{r_i\}$ are fixed. For δ_r, we use a deterministic entity assignment $\mathbf{Z}^r$, similar to the Haghighi and Klein (2009)'s SYN-CONSTR setting: each referring mention is coreferent with any past mention with the same head or in a deterministic syntactic configuration (appositives or predicative nominatives constructions).[6] The $\{r_i\}$ factors are heuristically set to place most of their mass on the closest antecedent by tree distance. During training, we proceed in stages, each consisting of 5 iterations:

Stage	Learned	Fixed	B^3All
1	$\delta_s, \delta_d, \{q_k\}$	$\{r_i\}$,δ_r	74.6
2	$\delta_s, \delta_d, \{q_k\}, \delta_r$	$\{r_i\}$	76.3
3	$\delta_s, \delta_d, \{q_k\}, \delta_r, \{r_i\}$	–	78.0

We evaluate our system at the end of stage using the B^3All metric on the A05CU development set (see Section 5 for details).

5 Experiments

We considered the challenging end-to-end system mention setting, where in addition to predicting mention partitions, a system must identify the mentions themselves and their boundaries automatically. Our system deterministically extracts mention boundaries from parse trees (Section 5.2). We utilized no coreference annotation during training, but did use minimal prototype information to prime the learning of entity types (Section 5.3).

5.1 Datasets

For evaluation, we used standard coreference data sets derived from the ACE corpora:

- **A04CU**: Train/dev/test split of the newswire portion of the ACE 2004 training set[7] utilized in Culotta et al. (2007), Bengston and Roth (2008) and Stoyanov et al. (2009). Consists of 90/68/38 documents respectively.
- **A05ST**: Train/test split of the newswire portion of the ACE 2005 training set utilized in Stoyanov et al. (2009). Consists of 57/24 documents respectively.
- **A05RA**: Train/test split of the ACE 2005 training set utilized in Rahman and Ng (2009). Consists of 482/117 documents respectively.

For all experiments, we evaluated on the dev and test sets above. To train, we included the text of all documents above, though of course not looking at either their mention boundaries or reference annotations in any way. We also trained on the following much larger unlabeled datasets utilized in Haghighi and Klein (2009):

- **BLLIP**: 5k articles of newswire parsed with the Charniak (2000) parser.
- **WIKI**: 8k abstracts of English Wikipedia articles parsed by the Berkeley parser (Petrov et al., 2006). Articles were selected to have subjects amongst the frequent proper nouns in the evaluation datasets.

5.2 Mention Detection and Properties

Mention boundaries were automatically detected as follows: For each noun or pronoun (determined by parser POS tag), we associated a mention with the maximal NP projection of that head or that word itself if no NP can be found. This procedure recovers over 90% of annotated mentions on the A05CU dev set, but also extracts many unannotated "spurious" mentions (for instance events, times, dates, or abstract nouns) which are not deemed to be of interest by the ACE annotation conventions.

Mention properties were obtained from parse trees using the the Stanford typed dependency extractor (de Marneffe et al., 2006). The mention properties we considered are the mention head (annotated with mention type), the typed modifiers of the head, and the governor of the head (conjoined with

[6] Forcing appositive coreference is essential for tying proper and nominal entity type vocabulary.

[7] Due to licensing restriction, the formal ACE test sets are not available to non-participants.

System	MUC P	MUC R	MUC F1	B^3All P	B^3All R	B^3All F1	B^3None P	B^3None R	B^3None F1	Pairwise F_1 P	Pairwise F_1 R	Pairwise F_1 F1
ACE2004-STOYANOV-TEST												
Stoyanov et al. (2009)	-	-	62.0	-	-	76.5	-	-	75.4	-	-	-
Haghighi and Klein (2009)	**67.5**	61.6	64.4	77.4	69.4	73.2	77.4	67.1	71.3	58.3	44.5	50.5
THIS WORK	67.4	**66.6**	**67.0**	**81.2**	**73.3**	**77.0**	**80.6**	**75.2**	**77.3**	59.2	50.3	54.4
ACE2005-STOYANOV-TEST												
Stoyanov et al. (2009)	-	-	67.4	-	-	73.7	-	-	72.5	-	-	-
Haghighi and Klein (2009)	73.1	58.8	65.2	82.1	63.9	71.8	81.2	61.6	70.1	66.1	37.9	48.1
THIS WORK	**74.6**	**62.7**	**68.1**	**83.2**	**68.4**	**75.1**	**82.7**	**66.3**	**73.6**	**64.3**	**41.4**	**50.4**
ACE2005-RAHMAN-TEST												
Rahman and Ng (2009)	75.4	64.1	69.3	-	-	-	**54.4**	70.5	61.4	-	-	-
Haghighi and Klein (2009)	72.9	60.2	67.0	53.2	73.1	61.6	52.0	72.6	60.6	57.0	44.6	50.0
THIS WORK	**77.0**	**66.9**	**71.6**	**55.4**	**74.8**	**63.8**	54.0	**74.7**	**62.7**	**60.1**	**47.7**	**53.0**

Table 1: Experimental results with system mentions. All systems except Haghighi and Klein (2009) and current work are fully supervised. The current work outperforms all other systems, supervised or unsupervised. For comparison purposes, the B^3None variant used on A05RA is calculated slightly differently than other B^3None results; see Rahman and Ng (2009).

the mention's syntactic position). We discard determiners, but make use of them in the discourse component (Section 3.2) for NP definiteness.

5.3 Prototyping Entity Types

While it is possible to learn type distributions in a completely unsupervised fashion, we found it useful to prime the system with a handful of important types. Rather than relying on fully supervised data, we took the approach of Haghighi and Klein (2006). For each type of interest, we provided a (possibly-empty) prototype list of proper and nominal head words, as well as a list of allowed pronouns. For instance, for the PERSON type we might provide:

NAM	Bush, Gore, Hussein
NOM	president, minister, official
PRO	he, his, she, him, her, you, ...

The prototypes were used as follows: Any entity with a prototype on any proper or nominal head word attribute list (Section 3.1) was constrained to have the specified type; i.e. the q_k factor (Section 4) places probability one on that single type. Similarly to Haghighi and Klein (2007) and Elsner et al. (2009), we biased these types' pronoun distributions to the allowed set of pronouns.

In general, the choice of entity types to prime with prototypes is a domain-specific question. For experiments here, we utilized the types which are annotated in the ACE coreference data: person (PERS), organization (ORG), geo-political entity (GPE), weapon (WEA), vehicle (VEH), location (LOC), and facility (FAC). Since the person type in ACE conflates individual persons with groups of people (e.g., *soldier* vs. *soldiers*), we added the group (GROUP) type and generated a prototype specification.

We obtained our prototype list by extracting at most four common proper and nominal head words from the newswire portions of the 2004 and 2005 ACE training sets (A04CU and A05ST); we chose prototype words to be minimally ambiguous with respect to type.[8] When there are not at least three proper heads for a type (WEA for instance), we did not provide any proper prototypes and instead strongly biased the type fertility parameters to generate empty NAM-HEAD lists.

Because only certain semantic types were annotated under the arbitrary ACE guidelines, there are many mentions which do not fall into those limited categories. We therefore prototype (refinements of) the ACE types and then add an equal number of unconstrained "other" types which are automatically induced. A nice consequence of this approach is that we can simply run our model on *all* mentions, discarding at evaluation time any which are of non-prototyped types.

5.4 Evaluation

We evaluated on multiple coreference resolution metrics, as no single one is clearly superior, partic-

[8] Meaning those headwords were assigned to the target type for more than 75% of their usages.

ularly in dealing with the system mention setting. We utilized MUC (Vilain et al., 1995), B^3All (Stoyanov et al., 2009), B^3None (Stoyanov et al., 2009), and Pairwise F1. The B^3All and B^3None are B^3 variants (Bagga and Baldwin, 1998) that differ in their treatment of spurious mentions. For Pairwise F1, precision measures how often pairs of predicted coreferent mentions are in the same annotated entity. We eliminated any mention pair from this calculation where both mentions were spurious.[9]

5.5 Results

Table 1 shows our results. We compared to two state-of-the-art supervised coreference systems. The Stoyanov et al. (2009) numbers represent their THRESHOLD_ESTIMATION setting and the Rahman and Ng (2009) numbers represent their highest-performing cluster ranking model. We also compared to the strong deterministic system of Haghighi and Klein (2009).[10] Across all data sets, our model, despite being largely unsupervised, consistently outperforms these systems, which are the best previously reported results on end-to-end coreference resolution (i.e. including mention detection). Performance on the A05RA dataset is generally lower because it includes articles from blogs and web forums where parser quality is significantly degraded.

While Bengston and Roth (2008) do not report on the full system mention task, they do report on the more optimistic setting where mention detection is performed but non-gold mentions are removed for evaluation using an oracle. On this more lenient setting, they report 78.4 B^3 on the A04CU test set. Our model yields 80.3.

6 Analysis

We now discuss errors and improvements made by our system. One frequent source of error is the merging of mentions with explicitly contrasting modifiers, such as *new president* and *old president*. While it is not unusual for a single entity to admit multiple modifiers, the particular modifiers *new* and *old* are incompatible in a way that *new* and *popular* are not. Our model does not represent the negative covariance between these modifiers.

We compared our output to the deterministic system of Haghighi and Klein (2009). Many improvements arise from correctly identifying mentions which are semantically compatible but which do not explicitly appear in an appositive or predicate-nominative configuration in the data. For example, *analyst* and *it* cannot corefer in our system because *it* is not a likely pronoun for the type PERSON.

While the focus of our model is coreference resolution, we can also isolate and evaluate the type component of our model as an NER system. We test this component by presenting our learned model with boundary-annotated non-pronominal entities from the A05ST dev set and querying their predicted type variable T. Doing so yields 83.2 entity classification accuracy under the mapping between our prototyped types and the coarse ACE types. Note that this task is substantially more difficult than the unsupervised NER in Elsner et al. (2009) because the inventory of named entities is larger (7 vs. 3) and because we predict types over nominal mentions that are more difficult to judge from surface forms. In this task, the plurality of errors are confusions between the GPE (geo-political entity) and ORG entity types, which have very similar distributions.

7 Conclusion

Our model is able to acquire and exploit knowledge at either the level of individual entities ("Obama" is a "president") and entity types ("company" can refer to a corporation). As a result, it leverages semantic constraints more effectively than systems operating at either level alone. In conjunction with reasonable, but simple, factors capturing discourse and syntactic configurational preferences, our entity-centric semantic model lowers coreference error rate substantially, particularly on semantically disambiguated references, giving a sizable improvement over the state-of-the-art.[11]

Acknowledgements: This project is funded in part by the Office of Naval Research under MURI Grant No. N000140911081.

[9] Note that we are still penalized for marking a spurious mention coreferent with an annotated one.

[10] Haghighi and Klein (2009) reports on true mentions; here, we report performance on automatically detected mentions.

[11] See *nlp.cs.berkeley.edu* and *aria42.com/software.html* for software release.

References

A Bagga and B Baldwin. 1998. Algorithms for scoring coreference chains. In *Linguistic Coreference Workshop (LREC)*.

Eric Bengston and Dan Roth. 2008. Understanding the Value of Features for Corefernce Resolution. In *Empirical Methods in Natural Language Processing (EMNLP)*.

David Blei and Peter I. Frazier. 2009. Distance Dependent Chinese Restaurant Processes. http://arxiv.org/abs/0910.1022/.

Eugene Charniak. 2000. Maximum Entropy Inspired Parser. In *North American Chapter of the Association of Computational Linguistics (NAACL)*.

Michael Collins and Yoram Singer. 1999. Unsupervised Models for Named Entity Classification. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Mike Collins. 1999. *Head-Driven Statistical Models for Natural Language Parsing*. Ph.D. thesis, University of Pennsylvania.

A Culotta, M Wick, R Hall, and A McCallum. 2007. First-order Probabilistic Models for Coreference Resolution. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing (NAACL-HLT)*.

M. C. de Marneffe, B. Maccartney, and C. D. Manning. 2006. Generating Typed Dependency Parses from Phrase Structure Parses. In *LREC*.

M Elsner, E Charniak, and M Johnson. 2009. Structured generative models for unsupervised named-entity clustering. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 164–172.

Barbara J. Grosz, Aravind K. Joshi, and Scott Weinstein. 1995. Centering: A Framework for Modeling the Local Coherence of Discourse. *Computational Linguistics*, 21(2):203–225.

Aria Haghighi and Dan Klein. 2006. Prototype-Driven Learning for Sequence Models. In *HLT-NAACL*. Association for Computational Linguistics.

Aria Haghighi and Dan Klein. 2007. Unsupervised Coreference Resolution in a Nonparametric Bayesian Model. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*. Association for Computational Linguistics.

Aria Haghighi and Dan Klein. 2009. Simple Coreference Resolution with Rich Syntactic and Semantic Features. In *Proceedings of the 2009 Conference on Empirical Conference in Natural Language Processing*.

J. R. Hobbs. 1978. Resolving Pronoun References. *Lingua*, 44.

J. R. Hobbs. 1979. Coherence and Coreference. *Cognitive Science*, 3:67–90.

Andrew Kehler, Laura Kertz, Hannah Rohde, and Jeffrey Elman. 2008. Coherence and Coreference Revisited.

Vincent Ng and Claire Cardie. 2002. Improving Machine Learning Approaches to Coreference Resolution. In *Association of Computational Linguists (ACL)*.

Vincent Ng. 2005. Machine Learning for Coreference Resolution: From Local Classification to Global Ranking. In *Association of Computational Linguists (ACL)*.

Vincent Ng. 2007. Shallow semantics for coreference resolution. In *IJCAI'07: Proceedings of the 20th international joint conference on Artifical intelligence*, pages 1689–1694.

Slav Petrov, Leon Barrett, Romain Thibaux, and Dan Klein. 2006. Learning Accurate, Compact, and Interpretable Tree Annotation. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 433–440, Sydney, Australia, July. Association for Computational Linguistics.

J. Pitman. 2002. Combinatorial Stochastic Processes. In *Lecture Notes for St. Flour Summer School*.

A Rahman and V Ng. 2009. Supervised models for coreference resolution. In *Proceedings of the 2009 Conference on Empirical Conference in Natural Language Processing*.

W.H. Soon, H. T. Ng, and D. C. Y. Lim. 1999. A Machine Learning Approach to Coreference Resolution of Noun Phrases.

V Stoyanov, N Gilbert, C Cardie, and E Riloff. 2009. Conundrums in Noun Phrase Coreference Resolution: Making Sense of the State-of-the-art. In *Associate of Computational Linguistics (ACL)*.

Marc Vilain, John Burger, John Aberdeen, Dennis Connolly, and Lynette Hirschman. 1995. A model-theoretic coreference scoring scheme. In *MUC-6*.

X Yang, J Su, and CL Tan. 2005. Improving pronoun resolution using statistics-based semantic compatibility information. In *Association of Computational Linguists (ACL)*.