Decentralized Entity-Level Modeling for Coreference Resolution

Greg Durrett, David Hall, and **Dan Klein**
Computer Science Division
University of California, Berkeley
{gdurrett,dlwh,klein}@cs.berkeley.edu

Abstract

Efficiently incorporating entity-level information is a challenge for coreference resolution systems due to the difficulty of exact inference over partitions. We describe an end-to-end discriminative probabilistic model for coreference that, along with standard pairwise features, enforces structural agreement constraints between specified properties of coreferent mentions. This model can be represented as a factor graph for each document that admits efficient inference via belief propagation. We show that our method can use entity-level information to outperform a basic pairwise system.

1 Introduction

The inclusion of entity-level features has been a driving force behind the development of many coreference resolution systems (Luo et al., 2004; Rahman and Ng, 2009; Haghighi and Klein, 2010; Lee et al., 2011). There is no polynomial-time dynamic program for inference in a model with arbitrary entity-level features, so systems that use such features typically rely on making decisions in a pipelined manner and sticking with them, operating greedily in a left-to-right fashion (Rahman and Ng, 2009) or in a multi-pass, sieve-like manner (Raghunathan et al., 2010). However, such systems may be locked into bad coreference decisions and are difficult to directly optimize for standard evaluation metrics.

In this work, we present a new structured model of entity-level information designed to allow efficient inference. We use a log-linear model that can be expressed as a factor graph. Pairwise features appear in the model as unary factors, adjacent to nodes representing a choice of antecedent (or none) for each mention. Additional nodes model entity-level properties on a per-mention basis, and structural agreement factors softly drive properties of coreferent mentions to agree with one another. This is a key feature of our model: mentions manage their partial membership in various coreference chains, so that information about entity-level properties is decentralized and propagated across individual mentions, and we never need to explicitly instantiate entities.

Exact inference in this factor graph is intractable, but efficient approximate inference can be carried out with belief propagation. Our model is the first discriminatively-trained model that both makes joint decisions over an entire document and models specific entity-level properties, rather than simply enforcing transitivity of pairwise decisions (Finkel and Manning, 2008; Song et al., 2012).

We evaluate our system on the dataset from the CoNLL 2011 shared task using three different types of properties: synthetic oracle properties, entity phi features (number, gender, animacy, and NER type), and properties derived from unsupervised clusters targeting semantic type information. In all cases, our transitive model of entity properties equals or outperforms our pairwise system and our reimplementation of a previous entity-level system (Rahman and Ng, 2009). Our final system is competitive with the winner of the CoNLL 2011 shared task (Lee et al., 2011).

2 Example

We begin with an example motivating our use of entity-level features. Consider the following excerpt concerning two famous auction houses:

> *When looking for* [*art items*], [*people*] *go to* [*Sotheby's and Christie's*] *because* [*they*]$_{\text{A}}$ *believe* [*they*]$_{\text{B}}$ *can get the best price for* [*them*].

The first three mentions are all distinct entities, *they*$_{\text{A}}$ and *they*$_{\text{B}}$ refer to *people*, and *them* refers to *art items*. The three pronouns are tricky to resolve automatically because they could at first glance resolve to any of the preceding mentions. We focus in particular on the resolution of *they*$_{\mathrm{A}}$ and *them*. In order to correctly resolve *they*$_{\mathrm{A}}$ to *people* rather than *Sotheby's and Christie's*, we must take advantage of the fact that *they*$_{\mathrm{A}}$ appears as the subject of the verb *believe*, which is much more likely to be attributed to people than to auction houses.

Binding principles prevent *them* from attaching to *they*$_{\mathrm{B}}$. But how do we prevent it from choosing as its antecedent the next closest agreeing pronoun, *they*$_{\mathrm{A}}$? One way is to exploit the correct coreference decision we have already made, *they*$_{\mathrm{A}}$ referring to *people*, since people are not as likely to have a price as art items are. This observation argues for enforcing agreement of entity-level semantic properties during inference, specifically properties relating to permitted semantic roles. Because even these six mentions have hundreds of potential partitions into coreference chains, we cannot search over partitions exhaustively, and therefore we must design our model to be able to use this information while still admitting an efficient inference scheme.

3 Models

We will first present our BASIC model (Section 3.1) and describe the features it incorporates (Section 3.2), then explain how to extend it to use transitive features (Sections 3.3 and 3.4).

Throughout this section, let x be a variable containing the words in a document along with any relevant precomputed annotation (such as parse information, semantic roles, etc.), and let n denote the number of mentions in a given document.

3.1 BASIC Model

Our BASIC model is depicted in Figure 1 in standard factor graph notation. Each mention i has an associated random variable a_i taking values in the set $\{1, \ldots, i{-}1, \text{<new>}\}$; this variable specifies mention i's selected antecedent or indicates that it begins a new coreference chain. Let $a = (a_1, ..., a_n)$ be the vector of the a_i. Note that a set of coreference chains C (the final desired output) can be uniquely determined from a, but a is not uniquely determined by C.

We use a log linear model of the conditional distribution $P(a|x)$ as follows:

$$P(a|x) \propto \exp\left(\sum_{i=1}^{n} \mathbf{w}^T \mathbf{f}_A(i, a_i, x)\right)$$

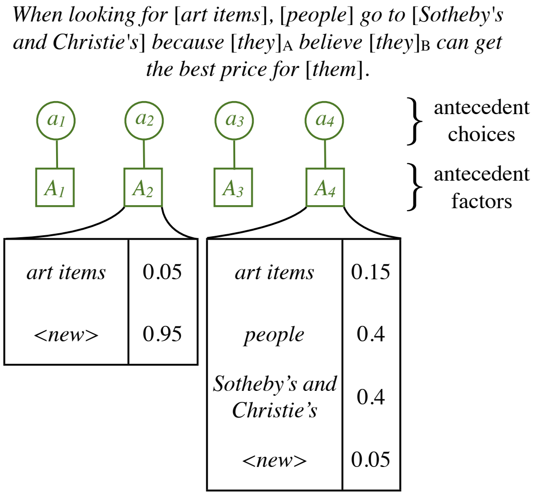

When looking for [art items], [people] go to [Sotheby's and Christie's] because [they]A believe [they]B can get the best price for [them].
a1
a2
a3
a4
antecedent choices
A1
A2
A3
A4
antecedent factors
art items 0.05
<new> 0.95
art items 0.15
people 0.4
Sotheby's and Christie's 0.4
<new> 0.05

Figure 1: Our BASIC coreference model. A decision a_i is made independently for each mention about what its antecedent mention should be or whether it should start a new coreference chain. Each unary factor A_i has a log-linear form with features examining mention i, its selected antecedent a_i, and the document context x.

where $\mathbf{f}_A(i, a_i, x)$ is a feature function that examines the coreference decision a_i for mention i with document context x; note that this feature function can include pairwise features based on mention i and the chosen antecedent a_i, since information about each mention is contained in x.

Because the model factors completely over the individual a_i, these feature functions $\mathbf{f}_A$ can be expressed as unary factors A_i (see Figure 1), with $A_i(j) \propto \exp\left(\mathbf{w}^T \mathbf{f}_A(i, j, x)\right)$. Given a setting of $\mathbf{w}$, we can determine $\hat{a} = \arg\max_a P(a|x)$ and then deterministically compute $C(a)$, the final set of coreference chains.

While the features of this model factor over coreference links, this approach differs from classical pairwise systems such as Bengtson and Roth (2008) or Stoyanov et al. (2010). Because potential antecedents compete with each other and with the non-anaphoric hypothesis, the choice of a_i actually represents a joint decision about $i{-}1$ pairwise links, as opposed to systems that use a pairwise binary classifier and a separate agglomeration step, which consider one link at a time during learning. This approach is similar to the mention-ranking model of Rahman and Ng (2009).

3.2 Pairwise Features

We now present the set of features $\mathbf{f}_A$ used by our unary factors A_i. Each feature examines the antecedent choice a_i of the current mention as well as the observed information x in the document. For each of the features we present, two conjoined versions are included: one with an indicator of the type of the current mention being resolved, and one with an indicator of the types of the current and antecedent mentions. Mention types are either NOMINAL, PROPER, or, if the mention is pronominal, a canonicalized version of the pronoun abstracting away case.[1]

Several features, especially those based on the precise constructs (apposition, etc.) and those incorporating phi feature information, are computed using the machinery in Lee et al. (2011). Other features were inspired by Song et al. (2012) and Rahman and Ng (2009).

Anaphoricity features: Indicator of anaphoricity, indicator on definiteness.

Configurational features: Indicator on distance in mentions (capped at 10), indicator on distance in sentences (capped at 10), does the antecedent c-command the current mention, are the two mentions in a subject/object construction, are the mentions nested, are the mentions in deterministic appositive/role appositive/predicate nominative/relative pronoun constructions.

Match features: Is one mention an acronym of the other, head match, head contained (each way), string match, string contained (each way), relaxed head match features from Lee et al. (2011).

Agreement features: Gender, number, animacy, and NER type of the current mention and the antecedent (separately and conjoined).

Discourse features: Speaker match conjoined with an indicator of whether the document is an article or conversation.

Because we use conjunctions of these base features together with the antecedent and mention type, our system can capture many relationships that previous systems hand-coded, especially regarding pronouns. For example, our system has access to features such as "*it* is non-anaphoric", "*it* has as its antecedent a geopolitical entity", or "*I* has as its antecedent *I* with the same speaker."

We experimented with synonymy and hypernymy features from WordNet (Miller, 1995), but these did not empirically improve performance.

3.3 TRANSITIVE Model

The BASIC model can capture many relationships between pairs of mentions, but cannot necessarily capture entity-level properties like those discussed in Section 2. We could of course model entities directly (Luo et al., 2004; Rahman and Ng, 2009), saying that each mention refers to some prior entity rather than to some prior mention. However, inference in this model would require reasoning about all possible partitions of mentions, which is computationally infeasible without resorting to severe approximations like a left-to-right inference method (Rahman and Ng, 2009).

Instead, we would like to try to preserve the tractability of the BASIC model while still being able to exploit entity-level information. To do so, we will allow *each mention* to maintain its own distributions over values for a number of properties; these properties could include gender, named-entity type, or semantic class. Then, we will require each anaphoric mention to agree with its antecedent on the value of each of these properties.

Our TRANSITIVE model which implements this scheme is shown in Figure 2. Each mention i has been augmented with a single property node $p_i \in \{1, ..., k\}$. The unary P_i factors encode prior knowledge about the setting of each p_i; these factors may be hard (*I* will not refer to a plural entity), soft (such as a distribution over named entity types output by an NER tagger), or practically uniform (e.g. the last name Smith does not specify a particular gender).

To enforce agreement of a particular property, we require a mention to have the same property value as its antecedent. That is, for mentions i and j, if $a_i = j$, we want to ensure that p_i and p_j agree. We can achieve this with the following set of structural equality factors:

$$E_{i-j}(a_i, p_i, p_j) = 1 - \mathbb{I}[a_i = j \wedge p_i \neq p_j]$$

In words, this factor is zero if both $a_i = j$ and p_i disagrees with p_j. These equality factors essentially provide a mechanism by which these priors P_i can influence the coreference decisions: if, for example, the factors P_i and P_j disagree very strongly, choosing $a_i \neq j$ will be preferred in order to avoid forcing one of p_i or p_j to take an undesirable value. Moreover, note that although a_i

[1] While this canonicalization could theoretically impair our ability to resolve, for example, reflexive pronouns, conjoining features with raw pronoun strings does not improve performance.

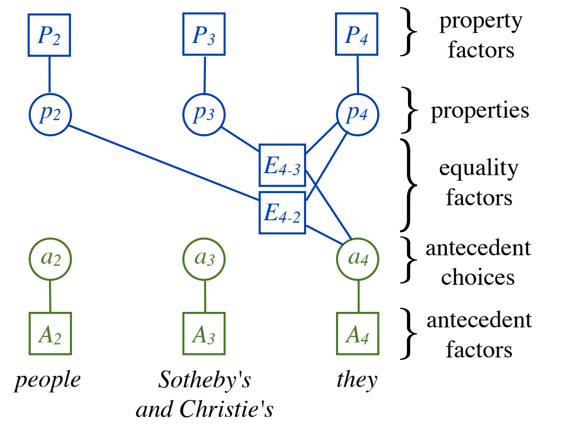


Figure 2: The factor graph for our TRANSITIVE coreference model. Each node a_i now has a property p_i, which is informed by its own unary factor P_i. In our example, a_4 strongly indicates that mentions 2 and 4 are coreferent; the factor E_{4-2} then enforces equality between p_2 and p_4, while the factor E_{4-3} has no effect.

only indicates a single antecedent, the transitive nature of the E factors forces p_i to agree with the p nodes of all other mentions likely to be in the same entity.

3.4 Property Projection

So far, our model as specified ensures agreement of our entity-level properties, but strictly enforcing agreement may not always be correct. Suppose that we are using named entity type as an entity-level property. Organizations and geo-political entities are two frequently confused and ambiguous tags, and in the gold-standard coreference chains it may be the case that a single chain contains instances of both. We might wish to learn that organizations and geo-political entities are "compatible" in the sense that we should forgive entities for containing both, but without losing the ability to reject a chain containing both organizations and people, for example.

To address these effects, we expand our model as indicated in Figure 3. As before, we have a set of properties p_i and agreement factors E_{ij}. On top of that, we introduce the notion of raw property values $r_i \in \{1, ..., k\}$ together with priors in the form of the R_i factors. The r_i and p_i could in principle have different domains, but for this work we take them to have the same domain. The P_i factors now have a new structure: they now represent a featurized projection of the r_i onto the p_i, which can now be thought of as "coreference-adapted" properties. The P_i factors are defined by $P_i(p_i, r_i) \propto \exp(\mathbf{w}^T \mathbf{f}_P(p_i, r_i))$, where $\mathbf{f}_P$ is a feature vector over the projection of r_i onto p_i. While there are many possible choices of $\mathbf{f}_P$, we choose it to be an indicator of the values of p_i and r_i, so that we learn a fully-parameterized projection matrix.[2] The R_i are constant factors, and may come from an upstream model or some other source depending on the property being modeled.

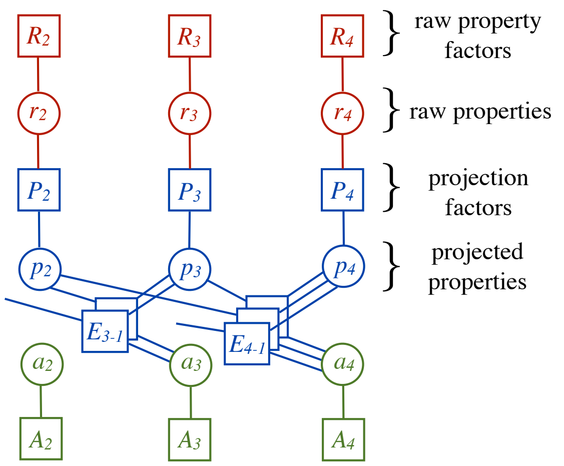


Figure 3: The complete factor graph for our TRANSITIVE coreference model. Compared to Figure 2, the R_i contain the raw cluster posteriors, and the P_i factors now project raw cluster values r_i into a set of "coreference-adapted" clusters p_i that are used as before. This projection allows mentions with different but compatible raw property values to coexist in the same coreference chain.

Our description thus far has assumed that we are modeling only one type of property. In fact, we can use multiple properties for each mention by duplicating the r and p nodes and the R, P, and E factors across each desired property. We index each of these by $l \in \{1, \ldots, m\}$ for each of m properties.

The final log-linear model is given by the following formula:

$$P(a|x) \propto \sum_{p,r} \left[\left(\prod_{i,j,l} E_{l,i-j}(a_i, p_{li}, p_{lj}) \right) \left(\prod_{i,l} R_{li}(r_{li}) \right) \exp\left(\mathbf{w}^T \sum_{i} \left(\mathbf{f}_A(i, a_i, x) + \sum_{l} \mathbf{f}_P(p_{li}, r_{li}) \right) \right) \right]$$

where i and j range over mentions, l ranges over

[2] Initialized to zero (or small values), this matrix actually causes the transitive machinery to have no effect, since all posteriors over the p_i are flat and completely uninformative. Therefore, we regularize the weights of the indicators of $p_i = r_i$ towards 1 and all other features towards 0 to give each raw cluster a preference for a distinct projected cluster.

each of m properties, and the outer sum indicates marginalization over all p and r variables.

4 Learning

Now that we have defined our model, we must decide how to train its weights $\mathbf{w}$. The first issue to address is one of the supervision provided. Our model traffics in sets of labels a which are more specified than gold coreference chains C, which give cluster membership for each mention but not antecedence. Let $\mathcal{A}(C)$ be the set of labelings a that are consistent with a set of coreference chains C. For example, if $C = \{\{1,2,3\},\{4\}\}$, then $(\text{<new>}, 1, 2, \text{<new>}) \in \mathcal{A}(C)$ and $(\text{<new>}, 1, 1, \text{<new>}) \in \mathcal{A}(C)$ but $(\text{<new>}, 1, \text{<new>}, 3) \notin \mathcal{A}(C)$, since this implies the chains $C = \{\{1,2\},\{3,4\}\}$

The most natural objective is a variant of standard conditional log-likelihood that treats the choice of a for the specified C as a latent variable to be marginalized out:

$$\ell(\mathbf{w}) = \sum_{i=1}^{t} \log \left(\sum_{a \in \mathcal{A}(C^i)} P(a|x^i) \right) \qquad (1)$$

where (x^i, C^i) is the ith labeled training example. This optimizes for the 0-1 loss; however, we are much more interested in optimizing with respect to a coreference-specific loss function.

To this end, we will use softmax-margin (Gimpel and Smith, 2010), which augments the probability of each example with a term proportional to its loss, pushing the model to assign less mass to highly incorrect examples. We modify Equation 1 to use a new probability distribution P' instead of P, where $P'(a|x^i) \propto P(a|x^i)\exp\left(l(a,C)\right)$ and $l(a,C)$ is a loss function. In order to perform inference efficiently, $l(a,C)$ must decompose linearly across mentions: $l(a,C) = \sum_{i=1}^{n} l(a_i, C)$. Commonly-used coreference metrics such as MUC (Vilain et al., 1995) and B^3 (Bagga and Baldwin, 1998) do not have this property, so we instead make use of a parameterized loss function that does and fit the parameters to give good performance. Specifically, we take

$$\begin{aligned} l(a,C) = \sum_{i=1}^{n} [&c_1 \mathbb{I}(K_1(a_i,C)) + c_2 \mathbb{I}(K_2(a_i,C)) \\ &+ c_3 \mathbb{I}(K_3(a_i,C))] \end{aligned}$$

where c_1, c_2, and c_3 are real-valued weights, K_1 denotes the event that a_i is falsely anaphoric when it should be non-anaphoric, K_2 denotes the event that a_i is falsely non-anaphoric when it should be anaphoric, and K_3 denotes the event that a_i is correctly determined to be anaphoric but linked incorrectly. These can be computed based on only a_i and C. By setting c_1 low and c_2 high relative to c_3, we can force the system to be less conservative about making anaphoricity decisions and achieve a better balance with the final coreference metrics.

Finally, we incorporate L_1 regularization, giving us our final objective:

$$\ell(\mathbf{w}) = \sum_{i=1}^{t} \log \left(\sum_{a \in \mathcal{A}(C^i)} P'(a|x^i) \right) + \lambda \|\mathbf{w}\|_1$$

We optimize this objective using AdaGrad (Duchi et al., 2011); we found this to be faster and give higher performance than L-BFGS using L_2 regularization (Liu and Nocedal, 1989). Note that because of the marginalization over $\mathcal{A}(C^i)$, even the objective for the BASIC model is not convex.

5 Inference

Inference in the BASIC model is straightforward. Given a set of weights $\mathbf{w}$, we can predict

$$\hat{a} = \arg\max_a P(a|x)$$

We then report the corresponding chains $C(a)$ as the system output.[3] For learning, the gradient takes the standard form of the gradient of a log-linear model, a difference of expected feature counts under the gold annotation and under no annotation. This requires computing marginals $P'(a_i|x)$ for each mention i, but because the model already factors this way, this step is easy.

The TRANSITIVE model is more complex. Exact inference is intractable due to the E factors that couple all of the a_i by way of the p_i nodes. However, we can compute approximate marginals for the a_i, p_i, and r_i using belief propagation. BP has been effectively used on other NLP tasks (Smith and Eisner, 2008; Burkett and Klein, 2012), and is effective in cases such as this where the model is largely driven by non-loopy factors (here, the A_i).

From marginals over each node, we can compute the necessary gradient and decode as before:

$$\hat{a} = \arg\max_a \hat{P}(a|x)$$

[3] One could use ILP-based decoding in the style of Finkel and Manning (2008) and Song et al. (2012) to attempt to explicitly find the optimal C with choice of a marginalized out, but we did not explore this option.

This corresponds to minimum-risk decoding with respect to the Hamming loss over antecedence predictions.

Pruning. The TRANSITIVE model requires instantiating a factor for each potential setting of each a_i. This factor graph grows quadratically in the size of the document, and even approximate inference becomes slow when a document contains over 200 mentions. Therefore, we use our BASIC model to prune antecedent choices for each a_i in order to reduce the size of the factor graph that we must instantiate. Specifically, we prune links between pairs of mentions that are of mention distance more than 100, as well as values for a_i that fall below a particular odds ratio threshold with respect to the best setting of that a_i in the BASIC model; that is, those for which

$$\log\left(\frac{P_{\text{BASIC}}\left(a_i|x\right)}{\max_j P_{\text{BASIC}}\left(a_i=j|x\right)}\right)$$

is below a cutoff γ.

6 Related Work

Our BASIC model is a mention-ranking approach resembling models used by Denis and Baldridge (2008) and Rahman and Ng (2009), though it is trained using a novel parameterized loss function. It is also similar to the MLN-JOINT(BF) model of Song et al. (2012), but we enforce the single-parent constraint at a deeper structural level, allowing us to treat non-anaphoricity symmetrically with coreference as in Denis and Baldridge (2007) and Stoyanov and Eisner (2012). The model of Fernandes et al. (2012) also uses the single-parent constraint structurally, but with learning via latent perceptron and ILP-based one-best decoding rather than logistic regression and BP-based marginal computation.

Our TRANSITIVE model is novel; while McCallum and Wellner (2004) proposed the idea of using attributes for mentions, they do not actually implement a model that does so. Other systems include entity-level information via handwritten rules (Raghunathan et al., 2010), induced rules (Yang et al., 2008), or features with learned weights (Luo et al., 2004; Rahman and Ng, 2011), but all of these systems freeze past coreference decisions in order to compute their entities.

Most similar to our entity-level approach is the system of Haghighi and Klein (2010), which also uses approximate global inference; however, theirs is an unsupervised, generative system and they attempt to directly model multinomials over words in each mention. Their system could be extended to handle property information like we do, but our system has many other advantages, such as freedom from a pre-specified list of entity types, the ability to use multiple input clusterings, and discriminative projection of clusters.

7 Experiments

We use the datasets, experimental setup, and scoring program from the CoNLL 2011 shared task (Pradhan et al., 2011), based on the OntoNotes corpus (Hovy et al., 2006). We use the standard automatic parses and NER tags for each document. Our mentions are those output by the system of Lee et al. (2011); we also use their postprocessing to remove appositives, predicate nominatives, and singletons before evaluation. For each experiment, we report MUC (Vilain et al., 1995), B^3 (Bagga and Baldwin, 1998), and CEAF$_\text{e}$ (Luo, 2005), as well as their average.

Parameter settings. We take the regularization constant $\lambda = 0.001$ and the parameters of our surrogate loss $(c_1, c_2, c_3) = (0.15, 2.5, 1)$ for all models.[4] All models are trained for 20 iterations. We take the pruning threshold $\gamma = -2$.

7.1 Systems

Besides our BASIC and TRANSITIVE systems, we evaluate a strictly pairwise system that incorporates property information by way of indicator features on the current mention's most likely property value and the proposed antecedent's most likely property value. We call this system PAIRPROPERTY; it is simply the BASIC system with an expanded feature set.

Furthermore, we compare against a LEFTTORIGHT entity-level system like that of Rahman and Ng (2009).[5] Decoding now operates in a sequential fashion, with BASIC features computed as before and entity features computed for each mention based on the coreference decisions made thus far. Following Rahman and Ng (2009), features for each property indicate whether the cur-

[4] Additional tuning of these hyper parameters did not significantly improve any of the models under any of the experimental conditions.

[5] Unfortunately, their publicly-available system is closed-source and performs poorly on the CoNLL shared task dataset, so direct comparison is difficult.

rent mention agrees with no mentions in the antecedent cluster, at least one mention, over half of the mentions, or all of the mentions; antecedent clusters of size 1 or 2 fire special-cased features. These additional features beyond those in Rahman and Ng (2009) were helpful, but more involved conjunction schemes and fine-grained features were not. During training, entity features of both the gold and the prediction are computed using the Viterbi clustering of preceding mentions under the current model parameters.[6]

All systems are run in a two-pass manner: first, the Basic model is run, then antecedent choices are pruned, then our second-round model is trained from scratch on the pruned data.[7]

7.2 Noisy Oracle Features

We first evaluate our model's ability to exploit synthetic entity-level properties. For this experiment, mention properties are derived from corrupted oracle information about the true underlying coreference cluster. Each coreference cluster is assumed to have one underlying value for each of m coreference properties, each taking values over a domain D. Mentions then sample distributions over D from a Dirichlet distribution peaked around the true underlying value.[8] These posteriors are taken as the R_i for the Transitive model.

We choose this setup to reflect two important properties of entity-level information: first, that it may come from a variety of disparate sources, and second, that it may be based on the determinations of upstream models which produce posteriors naturally. A strength of our model is that it can accept such posteriors as input, naturally making use of this information in a model-based way.

Table 1 shows development results averaged across ten train-test splits with $m = 3$ properties, each taking one of $|D| = 5$ values. We emphasize that these parameter settings give fairly weak oracle information: a document may have hundreds of clusters, so even in the absence of noise these oracle properties do not have high discriminating power. Still, we see that all models are able to benefit from incorporating this information; however, our Transitive model outperforms both the PairProperty model and the LeftToRight model. There are a few reasons for this: first, our model is able to directly use soft posteriors, so it is able to exploit the fact that more peaked samples from the Dirichlet are more likely to be correct. Moreover, our model can propagate information backwards in a document as well as forwards, so the effects of noise can be more easily mitigated. By contrast, in the LeftToRight model, if the first or second mention in a cluster has the wrong property value, features indicating high levels of property agreement will not fire on the next few mentions in those clusters.

	Noisy Oracle			
	MUC	B^3	CEAF$_e$	Avg.
Basic	61.96	70.66	47.30	59.97
PairProperty	66.31	72.68	49.08	62.69
LeftToRight	66.49	73.14	49.46	63.03
Transitive	**67.37**	**74.05**	**49.68**	**63.70**

Table 1: CoNLL metric scores for our four different systems incorporating noisy oracle data. This information helps substantially in all cases. Both entity-level models outperform the PairProperty model, but we observe that the Transitive model is more effective than the LeftToRight model at using this information.

7.3 Phi Features

As we have seen, our Transitive model can exploit high-quality entity-level features. How does it perform using real features that have been proposed for entity-level coreference?

Here, we use hard phi feature determinations extracted from the system of Lee et al. (2011). Named-entity type and animacy are both computed based on the output of a named-entity tagger, while number and gender use the dataset of Bergsma and Lin (2006). Once this information is determined, the PairProperty and LeftToRight systems can compute features over it directly. In the Transitive model, each of the R_i factors places $\frac{3}{4}$ of its mass on the determined label and distributes the remainder uniformly among the possible options.

Table 2 shows results when adding entity-level phi features on top of our Basic pairwise system (which already contains pairwise features) and on top of an ablated Basic system without pairwise

[6] Using gold entities for training as in Rahman and Ng (2009) resulted in a lower-performing system.

[7] We even do this for the Basic model, since we found that performance of the pruned and retrained model was generally higher.

[8] Specifically, the distribution used is a Dirichlet with $\alpha = 3.5$ for the true underlying cluster and $\alpha = 1$ for other values, chosen so that 25% of samples from the distribution did not have the correct mode. Though these parameters affect the quality of the oracle information, varying them did not change the relative performance of the different models.

Phi Features				
	MUC	B^3	CEAF$_e$	Avg.
Basic	61.96	70.66	47.30	59.97
LeftToRight	61.34	70.41	**47.64**	59.80
Transitive	**62.66**	**70.92**	46.88	**60.16**
Phi Features (ablated Basic)				
Basic-Phi	59.45	69.21	46.02	58.23
PairProperty	61.88	70.66	47.14	59.90
LeftToRight	61.42	70.53	**47.49**	59.81
Transitive	**62.23**	**70.78**	46.74	**59.92**

Table 2: CoNLL metric scores for our systems incorporating phi features. Our standard Basic system already includes phi features, so no results are reported for PairProperty. Here, our Transitive system does not give substantial improvement on the averaged metric. Over a baseline which does not include phi features, all systems are able to incorporate them comparably.

phi features. Our entity-level systems successfully captures phi features when they are not present in the baseline, but there is only slight benefit over pairwise incorporation, a result which has been noted previously (Luo et al., 2004).

7.4 Clustering Features

Finally, we consider mention properties derived from unsupervised clusterings; these properties are designed to target semantic properties of nominals that should behave more like the oracle features than the phi features do.

We consider clusterings that take as input pairs (n, r) of a noun head n and a string r which contains the semantic role of n (or some approximation thereof) conjoined with its governor. Two different algorithms are used to cluster these pairs: a NaiveBayes model, where c generates n and r, and a Conditional model, where c is generated conditioned on r and then n is generated from c. Parameters for each can be learned with the expectation maximization (EM) algorithm (Dempster et al., 1977), with symmetry broken by a small amount of random noise at initialization.

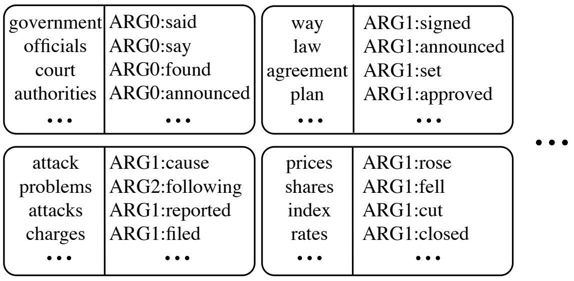
Similar models have been used to learn subcategorization information (Rooth et al., 1999) or properties of verb argument slots (Yao et al., 2011). We choose this kind of clustering for its relative simplicity and because it allows pronouns to have more informed properties (from their verbal context) than would be possible using a model that makes type-level decisions about nominals only. Though these specific cluster features are novel to coreference, previous work has used similar types of fine-grained semantic class information (Hendrickx and Daelemans, 2007; Ng, 2007; Rahman and Ng, 2010). Other approaches incorporate information from other sources (Ponzetto and Strube, 2006) or compute heuristic scores for real-valued features based on a large corpus or the web (Dagan and Itai, 1990; Yang et al., 2005; Bansal and Klein, 2012).

We use four different clusterings in our experiments, each with twenty clusters: dependency-parse-derived NaiveBayes clusters, semantic-role-derived Conditional clusters, SRL-derived NaiveBayes clusters generating a NoVerb token when r cannot be determined, and SRL-derived NaiveBayes clusters with all pronoun tuples discarded. Examples of the latter clusters are shown in Figure 4. Each clustering is learned for 30 iterations of EM over English Gigaword (Graff et al., 2007), parsed with the Berkeley Parser (Petrov et al., 2006) and with SRL determined by Senna (Collobert et al., 2011).

Table 3 shows results of modeling these cluster properties. As in the case of oracle features, the PairProperty and LeftToRight systems use the modes of the cluster posteriors, and the Transitive system uses the posteriors directly as the R_i. We see comparable performance from incorporating features in both an entity-level framework and a pairwise framework, though the Transi-

Clusters				
	MUC	B^3	CEAF$_e$	Avg.
Basic	61.96	70.66	47.30	59.97
PairProperty	62.88	70.71	**47.45**	60.35
LeftToRight	61.98	70.19	45.77	59.31
Transitive	**63.34**	**70.89**	46.88	**60.37**

Table 3: CoNLL metric scores for our systems incorporating clustering features. These features are equally effectively incorporated by our PairProperty system and our Transitive system.



government officials court authorities ...	ARG0:said ARG0:say ARG0:found ARG0:announced ...

way law agreement plan ...	ARG1:signed ARG1:announced ARG1:set ARG1:approved ...

attack problems attacks charges ...	ARG1:cause ARG2:following ARG1:reported ARG1:filed ...

prices shares index rates ...	ARG1:rose ARG1:fell ARG1:cut ARG1:closed ...

...

Figure 4: Examples of clusters produced by the NaiveBayes model on SRL-tagged data with pronouns discarded.

	MUC			B^3			CEAF_e			Avg.
	Prec.	Rec.	F_1	Prec.	Rec.	F_1	Prec.	Rec.	F_1	F_1
Basic	69.99	55.59	61.96	80.96	62.69	70.66	41.37	55.21	47.30	59.97
Stanford	61.49	59.59	60.49	74.60	68.25	71.28	47.57	49.45	48.49	60.10
Noisy Oracle										
PairProperty	76.49	58.53	66.31	84.98	63.48	72.68	41.84	59.36	49.08	62.69
LeftToRight	76.92	58.55	66.49	85.68	63.81	73.14	42.07	60.01	49.46	63.03
Transitive	76.48	60.20	*67.37	84.84	65.69	*74.05	42.89	59.01	*49.68	63.70
Phi Features										
LeftToRight	69.77	54.73	61.34	81.40	62.04	70.41	41.49	55.92	47.64	59.80
Transitive	70.27	56.54	*62.66	79.81	63.82	*70.92	41.17	54.44	46.88	60.16
Phi Features (ablated Basic)										
Basic-Phi	67.04	53.41	59.45	78.93	61.63	69.21	40.40	53.46	46.02	58.23
PairProperty	70.24	55.31	61.88	81.10	62.60	70.66	41.04	55.38	47.14	59.90
LeftToRight	69.94	54.75	61.42	81.38	62.23	70.53	41.29	55.87	47.49	59.81
Transitive	70.06	55.98	*62.23	79.92	63.52	70.78	40.90	54.52	46.74	59.92
Clusters										
PairProperty	71.77	55.95	62.88	81.76	62.30	70.71	40.98	56.35	47.45	60.35
LeftToRight	69.75	54.82	61.39	81.48	62.29	70.60	41.62	55.89	47.71	59.90
Transitive	71.54	56.83	*63.34	80.55	63.31	*70.89	40.77	55.14	46.88	60.37

Table 4: CoNLL metric scores averaged across ten different splits of the training set for each experiment. We include precision, recall, and F_1 for each metric for completeness. Starred F_1 values on the individual metrics for the Transitive system are significantly better than all other results in the same block at the $p = 0.01$ level according to a bootstrap resampling test.

	MUC			B^3			CEAF_e			Avg.
	Prec.	Rec.	F_1	Prec.	Rec.	F_1	Prec.	Rec.	F_1	F_1
Basic	68.84	56.08	61.81	77.60	61.40	68.56	38.25	50.57	43.55	57.97
PairProperty	70.90	56.26	62.73	78.95	60.79	68.69	37.69	51.92	43.67	58.37
LeftToRight	68.84	55.56	61.49	78.64	61.03	68.72	38.97	51.74	44.46	58.22
Transitive	70.62	58.06	*63.73	76.93	62.24	68.81	38.00	50.40	43.33	58.62
Stanford	60.91	62.13	61.51	70.61	67.75	69.15	45.79	44.55	45.16	58.61

Table 5: CoNLL metric scores for our best systems (including clustering features) on the CoNLL blind test set, reported in the same manner as Table 4.

tive system appears to be more effective than the LeftToRight system.

7.5 Final Results

Table 4 shows expanded results on our development sets for the different types of entity-level information we considered. We also show in in Table 5 the results of our system on the CoNLL test set, and see that it performs comparably to the Stanford coreference system (Lee et al., 2011). Here, our Transitive system provides modest improvements over all our other systems.

Based on Table 4, our Transitive system appears to do better on MUC and B^3 than on CEAF_e. However, we found no simple way to change the relative performance characteristics of our various systems; notably, modifying the parameters of the loss function mentioned in Section 4 or changing it entirely did not trade off these three metrics but merely increased or decreased them in lockstep. Therefore, the Transitive system actually substantially improves over our baselines and is not merely trading off metrics in a way that could be easily reproduced through other means.

8 Conclusion

In this work, we presented a novel coreference architecture that can both take advantage of standard pairwise features as well as use transitivity to enforce coherence of decentralized entity-level properties within coreference clusters. Our transitive system is more effective at using properties than a pairwise system and a previous entity-level system, and it achieves performance comparable to that of the Stanford coreference resolution system, the winner of the CoNLL 2011 shared task.

Acknowledgments

This work was partially supported by BBN under DARPA contract HR0011-12-C-0014, by an NSF fellowship for the first author, and by a Google fellowship for the second. Thanks to the anonymous reviewers for their insightful comments.

References

Amit Bagga and Breck Baldwin. 1998. Algorithms for Scoring Coreference Chains. In *Proceedings of the Conference on Language Resources and Evaluation Workshop on Linguistics Coreference*.

Mohit Bansal and Dan Klein. 2012. Coreference Semantics from Web Features. In *Proceedings of the Association for Computational Linguistics*.

Eric Bengtson and Dan Roth. 2008. Understanding the Value of Features for Coreference Resolution. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Shane Bergsma and Dekang Lin. 2006. Bootstrapping Path-Based Pronoun Resolution. In *Proceedings of the Conference on Computational Linguistics and the Association for Computational Linguistics*.

David Burkett and Dan Klein. 2012. Fast Inference in Phrase Extraction Models with Belief Propagation. In *Proceedings of the North American Chapter of the Association for Computational Linguistics*.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural Language Processing (Almost) from Scratch. *Journal of Machine Learning Research*, 12:2493–2537, November.

Ido Dagan and Alon Itai. 1990. Automatic Processing of Large Corpora for the Resolution of Anaphora References. In *Proceedings of the Conference on Computational Linguistics - Volume 3*.

Arthur P. Dempster, Nan M. Laird, and Donald B. Rubin. 1977. Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1):1–38.

Pascal Denis and Jason Baldridge. 2007. Joint Determination of Anaphoricity and Coreference Resolution using Integer Programming. In *Proceedings of the North American Chapter of the Association for Computational Linguistics*.

Pascal Denis and Jason Baldridge. 2008. Specialized Models and Ranking for Coreference Resolution. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

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

Eraldo Rezende Fernandes, Cícero Nogueira dos Santos, and Ruy Luiz Milidiú. 2012. Latent Structure Perceptron with Feature Induction for Unrestricted Coreference Resolution. In *Proceedings of the Joint Conference on Empirical Methods in Natural Language Proceedings and Conference on Computational Natural Language Learning - Shared Task*.

Jenny Rose Finkel and Christopher D. Manning. 2008. Enforcing Transitivity in Coreference Resolution. In *Proceedings of the Association for Computational Linguistics: Short Papers*.

Kevin Gimpel and Noah A. Smith. 2010. Softmax-Margin CRFs: Training Log-Linear Models with Cost Functions. In *Proceedings of the North American Chapter for the Association for Computational Linguistics*.

David Graff, Junbo Kong, Ke Chen, and Kazuaki Maeda. 2007. English Gigaword Third Edition. Linguistic Data Consortium, Catalog Number LDC2007T07.

Aria Haghighi and Dan Klein. 2010. Coreference Resolution in a Modular, Entity-Centered Model. In *Proceedings of the North American Chapter of the Association for Computational Linguistics*.

Iris Hendrickx and Walter Daelemans, 2007. *Adding Semantic Information: Unsupervised Clusters for Coreference Resolution*.

Eduard Hovy, Mitchell Marcus, Martha Palmer, Lance Ramshaw, and Ralph Weischedel. 2006. OntoNotes: the 90% solution. In *Proceedings of the North American Chapter of the Association for Computational Linguistics: Short Papers*.

Heeyoung Lee, Yves Peirsman, Angel Chang, Nathanael Chambers, Mihai Surdeanu, and Dan Jurafsky. 2011. Stanford's Multi-Pass Sieve Coreference Resolution System at the CoNLL-2011 Shared Task. In *Proceedings of the Conference on Computational Natural Language Learning: Shared Task*.

Dong C. Liu and Jorge Nocedal. 1989. On the Limited Memory BFGS Method for Large Scale Optimization. *Mathematical Programming*, 45(3):503–528, December.

Xiaoqiang Luo, Abe Ittycheriah, Hongyan Jing, Nanda Kambhatla, and Salim Roukos. 2004. A Mention-Synchronous Coreference Resolution Algorithm Based on the Bell Tree. In *Proceedings of the Association for Computational Linguistics*.

Xiaoqiang Luo. 2005. On Coreference Resolution Performance Metrics. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Andrew McCallum and Ben Wellner. 2004. Conditional Models of Identity Uncertainty with Application to Noun Coreference. In *Proceedings of Advances in Neural Information Processing Systems*.

George A. Miller. 1995. WordNet: A Lexical Database for English. *Communications of the ACM*, 38:39–41.

Vincent Ng. 2007. Semantic class induction and coreference resolution. In *Proceedings of the Association for Computational Linguistics*.

Slav Petrov, Leon Barrett, Romain Thibaux, and Dan Klein. 2006. Learning Accurate, Compact, and Interpretable Tree Annotation. In *Proceedings of the Conference on Computational Linguistics and the Association for Computational Linguistics*.

Simone Paolo Ponzetto and Michael Strube. 2006. Exploiting Semantic Role Labeling, WordNet and Wikipedia for Coreference Resolution. In *Proceedings of the North American Chapter of the Association of Computational Linguistics*.

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 Conference on Computational Natural Language Learning: Shared Task*.

Karthik Raghunathan, Heeyoung Lee, Sudarshan Rangarajan, Nathanael Chambers, Mihai Surdeanu, Dan Jurafsky, and Christopher Manning. 2010. A Multi-Pass Sieve for Coreference Resolution. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Altaf Rahman and Vincent Ng. 2009. Supervised Models for Coreference Resolution. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Altaf Rahman and Vincent Ng. 2010. Inducing Fine-Grained Semantic Classes via Hierarchical and Collective Classification. In *Proceedings of the International Conference on Computational Linguistics*.

Altaf Rahman and Vincent Ng. 2011. Narrowing the Modeling Gap: A Cluster-Ranking Approach to Coreference Resolution. *Journal of Artificial Intelligence Research*, 40(1):469–521, January.

Mats Rooth, Stefan Riezler, Detlef Prescher, Glenn Carroll, and Franz Beil. 1999. Inducing a Semantically Annotated Lexicon via EM-Based Clustering. In *Proceedings of the Association for Computational Linguistics*.

David A. Smith and Jason Eisner. 2008. Dependency Parsing by Belief Propagation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Yang Song, Jing Jiang, Wayne Xin Zhao, Sujian Li, and Houfeng Wang. 2012. Joint Learning for Coreference Resolution with Markov Logic. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Veselin Stoyanov and Jason Eisner. 2012. Easy-first Coreference Resolution. In *Proceedings of the International Conference on Computational Linguistics*.

Veselin Stoyanov, Claire Cardie, Nathan Gilbert, Ellen Riloff, David Buttler, and David Hysom. 2010. Coreference Resolution with Reconcile. In *Proceedings of the Association for Computational Linguistics: Short Papers*.

Marc Vilain, John Burger, John Aberdeen, Dennis Connolly, and Lynette Hirschman. 1995. A Model-Theoretic Coreference Scoring Scheme. In *Proceedings of the Conference on Message Understanding*.

Xiaofeng Yang, Jian Su, and Chew Lim Tan. 2005. Improving Pronoun Resolution Using Statistics-Based Semantic Compatibility Information. In *Proceedings of the Association for Computational Linguistics*.

Xiaofeng Yang, Jian Su, Jun Lang, Chew L. Tan, Ting Liu, and Sheng Li. 2008. An Entity-Mention Model for Coreference Resolution with Inductive Logic Programming. In *Proceedings of the Association for Computational Linguistics*.

Limin Yao, Aria Haghighi, Sebastian Riedel, and Andrew McCallum. 2011. Structured Relation Discovery Using Generative Models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.