A Discriminative Hierarchical Model for Fast Coreference at Large Scale

Michael Wick
University of Massachsetts
140 Governor's Drive
Amherst, MA
mwick@cs.umass.edu

Sameer Singh
University of Massachusetts
140 Governor's Drive
Amherst, MA
sameer@cs.umass.edu

Andrew McCallum
University of Massachusetts
140 Governor's Drive
Amherst, MA
mccallum@cs.umass.edu

Abstract

Methods that measure compatibility between mention pairs are currently the dominant approach to coreference. However, they suffer from a number of drawbacks including difficulties scaling to large numbers of mentions and limited representational power. As these drawbacks become increasingly restrictive, the need to replace the pairwise approaches with a more expressive, highly scalable alternative is becoming urgent. In this paper we propose a novel discriminative hierarchical model that recursively partitions entities into trees of latent sub-entities. These trees succinctly summarize the mentions providing a highly compact, information-rich structure for reasoning about entities and coreference uncertainty at massive scales. We demonstrate that the hierarchical model is several orders of magnitude faster than pairwise, allowing us to perform coreference on six million author mentions in under four hours on a single CPU.

1 Introduction

Coreference resolution, the task of clustering *mentions* into partitions representing their underlying real-world *entities*, is fundamental for high-level information extraction and data integration, including semantic search, question answering, and knowledge base construction. For example, coreference is vital for determining author publication lists in bibliographic knowledge bases such as CiteSeer and Google Scholar, where the repository must know if the "R. Hamming" who authored "Error detecting and error correcting codes" is the same" "R. Hamming" who authored "The unreasonable effectiveness of mathematics." Features of the mentions (e.g., bags-of-words in titles, contextual snippets and co-author lists) provide evidence for resolving such entities.

Over the years, various machine learning techniques have been applied to different variations of the coreference problem. A commonality in many of these approaches is that they model the problem of entity coreference as a collection of decisions between mention pairs (Bagga and Baldwin, 1999; Soon et al., 2001; McCallum and Wellner, 2004; Singla and Domingos, 2005; Bengston and Roth, 2008). That is, coreference is solved by answering a quadratic number of questions of the form "does *mention A* refer to the same entity as *mention B*?" with a compatibility function that indicates how likely A and B are coreferent. While these models have been successful in some domains, they also exhibit several undesirable characteristics. The first is that pairwise models lack the expressivity required to represent aggregate properties of the entities. Recent work has shown that these entity-level properties allow systems to correct coreference errors made from myopic pairwise decisions (Ng, 2005; Culotta et al., 2007; Yang et al., 2008; Rahman and Ng, 2009; Wick et al., 2009), and can even provide a strong signal for unsupervised coreference (Bhattacharya and Getoor, 2006; Haghighi and Klein, 2007; Haghighi and Klein, 2010).

A second problem, that has received significantly less attention in the literature, is that the pairwise coreference models scale poorly to large collections of mentions especially when the expected

Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics, pages 379–388,
Jeju, Republic of Korea, 8-14 July 2012.

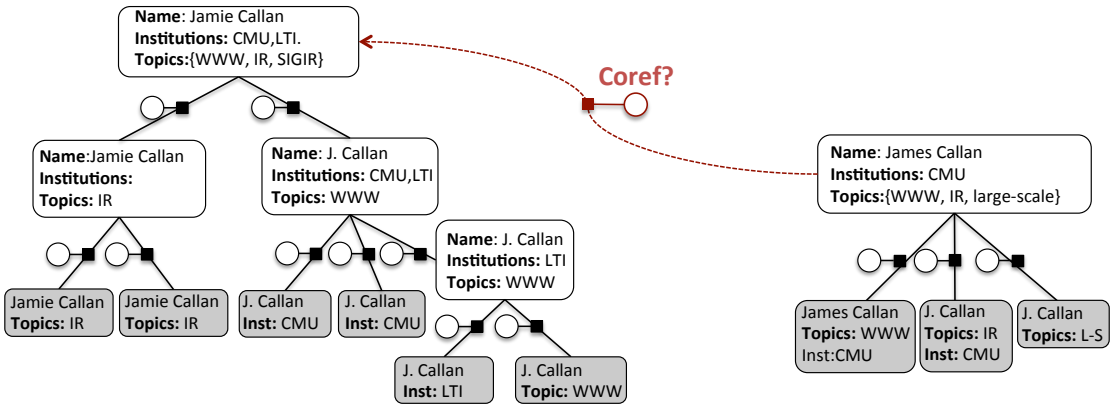


Figure 1: **Discriminative hierarchical factor graph for coreference:** Latent entity nodes (white boxes) summarize subtrees. Pairwise factors (black squares) measure compatibilities between child and parent nodes, avoiding quadratic blow-up. Corresponding decision variables (open circles) indicate whether one node is the child of another. Mentions (gray boxes) are leaves. Deciding whether to merge these two entities requires evaluating just a single factor (red square), corresponding to the new child-parent relationship.

number of mentions in each entity cluster is also large. Current systems cope with this by either dividing the data into blocks to reduce the search space (Hernández and Stolfo, 1995; McCallum et al., 2000; Bilenko et al., 2006), using fixed heuristics to greedily compress the mentions (Ravin and Kazi, 1999; Rao et al., 2010), employing specialized Markov chain Monte Carlo procedures (Milch et al., 2006; Richardson and Domingos, 2006; Singh et al., 2010), or introducing shallow hierarchies of sub-entities for MCMC block moves and super-entities for adaptive distributed inference (Singh et al., 2011). However, while these methods help manage the search space for medium-scale data, evaluating each coreference decision in many of these systems still scales linearly with the number of mentions in an entity, resulting in prohibitive computational costs associated with large datasets. This scaling with the number of mentions per entity seems particularly wasteful because although it is common for an entity to be referenced by a large number of mentions, many of these coreferent mentions are highly similar to each other. For example, in author coreference the two most common strings that refer to Richard Hamming might have the form "R. Hamming" and "Richard Hamming." In newswire coreference, a prominent entity like Barack Obama may have millions of "Obama" mentions (many occurring in similar semantic contexts). Deciding whether a mention belongs to this entity need not involve comparisons to all contextually similar "Obama" mentions; rather we prefer a more compact representation in order to efficiently reason about them.

In this paper we propose a novel hierarchical discriminative factor graph for coreference resolution that recursively structures each entity as a tree of latent sub-entities with mentions at the leaves. Our hierarchical model avoids the aforementioned problems of the pairwise approach: not only can it jointly reason about attributes of entire entities (using the power of discriminative conditional random fields), but it is also able to scale to datasets with enormous numbers of mentions because scoring entities does not require computing a quadratic number of compatibility functions. The key insight is that each node in the tree functions as a highly compact information-rich summary of its children. Thus, a small handful of upper-level nodes may summarize millions of mentions (for example, a single node may summarize all contextually similar "R. Hamming" mentions). Although inferring the structure of the entities requires reasoning over a larger state-space, the latent trees are actually beneficial to inference (as shown for shallow trees in Singh et al. (2011)), resulting in rapid progress toward high probability regions, and mirroring known benefits of auxiliary variable methods in statistical physics (such as Swendsen and Wang (1987)). Moreover,

each step of inference is computationally efficient because evaluating the cost of attaching (or detaching) sub-trees requires computing just a single compatibility function (as seen in Figure 1). Further, our hierarchical approach provides a number of additional advantages. First, the recursive nature of the tree (arbitrary depth and width) allows the model to adapt to different types of data and effectively compress entities of different scales (e.g., entities with more mentions may require a deeper hierarchy to compress). Second, the model contains compatibility functions at all levels of the tree enabling it to simultaneously reason at multiple granularities of entity compression. Third, the trees can provide split points for finer-grained entities by placing contextually similar mentions under the same subtree. Finally, if memory is limited, redundant mentions can be pruned by replacing subtrees with their roots.

Empirically, we demonstrate that our model is several orders of magnitude faster than a pairwise model, allowing us to perform efficient coreference on nearly six million author mentions in under four hours using a single CPU.

2 Background: Pairwise Coreference

Coreference is the problem of clustering mentions such that mentions in the same set refer to the same real-world entity; it is also known as entity disambiguation, record linkage, and de-duplication. For example, in author coreference, each mention might be represented as a record extracted from the author field of a textual citation or BibTeX record. The mention record may contain attributes for the first, middle, and last name of the author, as well as contextual information occurring in the citation string, co-authors, titles, topics, and institutions. The goal is to cluster these mention records into sets, each containing all the mentions of the author to which they refer; we use this task as a running pedagogical example.

Let $\mathcal{M}$ be the space of observed mention records; then the traditional pairwise coreference approach scores candidate coreference solutions with a compatibility function $\psi : \mathcal{M} \times \mathcal{M} \rightarrow \Re$ that measures how likely it is that the two mentions refer to the same entity.[1] In discriminative log-linear models, the function ψ takes the form of weights $\boldsymbol{\theta}$ on features $\boldsymbol{\phi}(m_i, m_j)$, i.e., $\psi(m_i, m_j) = \exp(\boldsymbol{\theta} \cdot \boldsymbol{\phi}(m_i, m_j))$. For example, in author coreference, the feature functions $\boldsymbol{\phi}$ might test whether the name fields for two author mentions are string identical, or compute cosine similarity between the two mentions' bags-of-words, each representing a mention's context. The corresponding real-valued weights $\boldsymbol{\theta}$ determine the impact of these features on the overall pairwise score.

Coreference can be solved by introducing a set of binary coreference decision variables for each mention pair and predicting a setting to their values that maximizes the sum of pairwise compatibility functions. While it is possible to independently make pairwise decisions and enforce transitivity *post hoc*, this can lead to poor accuracy because the decisions are tightly coupled. For higher accuracy, a graphical model such as a conditional random field (CRF) is constructed from the compatibility functions to jointly reason about the pairwise decisions (McCallum and Wellner, 2004). We now describe the pairwise CRF for coreference as a factor graph.

2.1 Pairwise Conditional Random Field

Each mention $m_i \in \mathcal{M}$ is an observed variable, and for each mention pair (m_i, m_j) we have a binary coreference decision variable y_{ij} whose value determines whether m_i and m_j refer to the same entity (i.e., 1 means they are coreferent and 0 means they are not coreferent). The pairwise compatibility functions become the factors in the graphical model. Each factor examines the properties of its mention pair as well as the setting to the coreference decision variable and outputs a score indicating how likely the setting of that coreference variable is. The joint probability distribution over all possible settings to the coreference decision variables ($\mathbf{y}$) is given as a product of all the pairwise compatibility factors:

$$Pr(\mathbf{y}|\mathbf{m}) \propto \prod_{i=1}^{n} \prod_{j=1}^{n} \psi(m_i, m_j, y_{ij}) \qquad (1)$$

Given the pairwise CRF, the problem of coreference is then solved by searching for the setting of the coreference decision variables that has the highest probability according to Equation 1 subject to the

[1] We can also include an *incompatibility* function for when the mentions are not coreferent, e.g., $\psi : \mathcal{M} \times \mathcal{M} \times \{0, 1\} \rightarrow \Re$

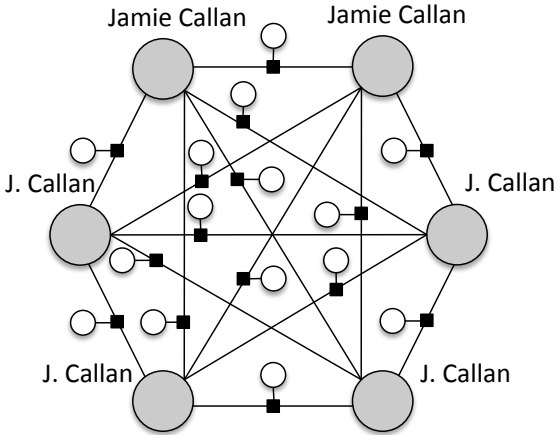


Figure 2: **Pairwise model on six mentions:** Open circles are the binary coreference decision variables, shaded circles are the observed mentions, and the black boxes are the factors of the graphical model that encode the pairwise compatibility functions.

constraint that the setting to the coreference variables obey transitivity;[2] this is the maximum probability estimate (MPE) setting. However, the solution to this problem is intractable, and even approximate inference methods such as loopy belief propagation can be difficult due to the cubic number of deterministic transitivity constraints.

2.2 Approximate Inference

An approximate inference framework that has successfully been used for coreference models is Metropolis-Hastings (MH) (Milch et al. (2006), Culotta and McCallum (2006), Poon and Domingos (2007), amongst others), a Markov chain Monte Carlo algorithm traditionally used for marginal inference, but which can also be tuned for MPE inference. MH is a flexible framework for specifying customized local-search transition functions and provides a principled way of deciding which local search moves to accept. A proposal function q takes the current coreference hypothesis and proposes a new hypothesis by modifying a subset of the decision variables. The proposed change is accepted with probability α:

$$\alpha = \min\left(1, \frac{Pr(\mathbf{y}')}{Pr(\mathbf{y})}\frac{q(\mathbf{y}|\mathbf{y}')}{q(\mathbf{y}'|\mathbf{y})}\right) \qquad (2)$$

[2]We say that a full assignment to the coreference variables $\mathbf{y}$ obeys transitivity if $\forall\, ijk\ y_{ij} = 1 \wedge y_{jk} = 1 \implies y_{ik} = 1$

When using MH for MPE inference, the second term $q(\mathbf{y}|\mathbf{y}')/q(\mathbf{y}'|\mathbf{y})$ is optional, and usually omitted. Moves that reduce model score may be accepted and an optional temperature can be used for annealing. The primary advantages of MH for coreference are (1) only the compatibility functions of the changed decision variables need to be evaluated to accept a move, and (2) the proposal function can enforce the transitivity constraint by exploring only variable settings that result in valid coreference partitionings.

A commonly used proposal distribution for coreference is the following: (1) randomly select two mentions (m_i, m_j), (2) if the mentions (m_i, m_j) are in the same entity cluster according to $\mathbf{y}$ then move one mention into a singleton cluster (by setting the necessary decision variables to 0), otherwise, move mention m_i so it is in the same cluster as m_j (by setting the necessary decision variables). Typically, MH is employed by first initializing to a singleton configuration (all entities have one mention), and then executing the MH for a certain number of steps (or until the predicted coreference hypothesis stops changing).

This proposal distribution always moves a single mention m from some entity e_i to another entity e_j and thus the configuration $\mathbf{y}$ and $\mathbf{y}'$ only differ by the setting of decision variables governing to which entity m refers. In order to guarantee transitivity and a valid coreference equivalence relation, we must properly remove m from e_i by untethering m from each mention in e_i (this requires computing $|e_i| - 1$ pairwise factors). Similarly—again, for the sake of transitivity—in order to complete the move into e_j we must coref m to each mention in e_j (this requires computing $|e_j|$ pairwise factors). Clearly, all the other coreference decision variables are independent and so their corresponding factors cancel because they yield the same scores under $\mathbf{y}$ and $\mathbf{y}'$. Thus, evaluating each proposal for the pairwise model scales linearly with the number of mentions assigned to the entities, requiring the evaluation of $2(|e_i| + |e_j| - 1)$ compatibility functions (factors).

3 Hierarchical Coreference

Instead of only capturing a single coreference clustering between mention pairs, we can imagine multiple levels of coreference decisions over different

granularities. For example, mentions of an author may be further partitioned into semantically similar sets, such that mentions from each set have topically similar papers. This partitioning can be recursive, i.e., each of these sets can be further partitioned, capturing candidate splits for an entity that can facilitate inference. In this section, we describe a model that captures arbitrarily deep hierarchies over such layers of coreference decisions, enabling efficient inference and rich entity representations.

3.1 Discriminative Hierarchical Model

In contrast to the pairwise model, where each entity is a flat cluster of mentions, our proposed model structures each entity recursively as a tree. The leaves of the tree are the observed mentions with a set of attribute values. Each internal node of the tree is latent and contains a set of unobserved attributes; recursively, these *node records* summarize the attributes of their child nodes (see Figure 1), for example, they may aggregate the bags of context words of the children. The root of each tree represents the entire entity, with the leaves containing its mentions. Formally, the coreference decision variables in the hierarchical model no longer represent pairwise decisions directly. Instead, a decision variable $y_{r_i,r_j} = 1$ indicates that node-record r_j is the parent of node-record r_i. We say a node-record *exists* if either it is a mention, has a parent, or has at least one child. Let R be the set of all existing node records, let r^p denote the parent for node r, that is $y_{r,r^p} = 1$, and $\forall r' \neq r^p, y_{r,r'} = 0$. As we describe in more detail later, the structure of the tree and the values of the unobserved attributes are determined during inference.

In order to represent our recursive model of coreference, we include two types of factors: pairwise factors ψ_{pw} that measure compatibility between a child node-record and its parent, and unit-wise factors ψ_{rw} that measure compatibilities of the node-records themselves. For efficiency we enforce that parent-child factors only produce a non-zero score when the corresponding decision variable is 1. The unit-wise factors can examine compatibility of settings to the attribute variables for a particular node (for example, the set of topics may be too diverse to represent just a single entity), as well as enforce priors over the tree's breadth and depth. Our recursive hierarchical model defines the probability of a configuration as:

$$Pr(\mathbf{y}, R|\mathbf{m}) \propto \prod_{r \in R} \psi_{\text{rw}}(r)\psi_{\text{pw}}(r, r^p) \qquad (3)$$

3.2 MCMC Inference for Hierarchical models

The state space of our hierarchical model is substantially larger (theoretically infinite) than the pairwise model due to the arbitrarily deep (and wide) latent structure of the cluster trees. Inference must simultaneously determine the structure of the tree, the latent node-record values, as well as the coreference decisions themselves.

While this may seem daunting, the structures being inferred are actually beneficial to inference. Indeed, despite the enlarged state space, inference in the hierarchical model is substantially faster than a pairwise model with a smaller state space. One explanatory intuition comes from the statistical physics community: we can view the latent tree as auxiliary variables in a data-augmentation sampling scheme that guide MCMC through the state space more efficiently. There is a large body of literature in the statistics community describing how these auxiliary variables can lead to faster convergence despite the enlarged state space (classic examples include Swendsen and Wang (1987) and slice samplers (Neal, 2000)).

Further, evaluating each proposal during inference in the hierarchical model is substantially faster than in the pairwise model. Indeed, we can replace the linear number of factor evaluations (as in the pairwise model) with a constant number of factor evaluations for most proposals (for example, adding a subtree requires re-evaluating only a single parent-child factor between the subtree and the attachment point, and a single node-wise factor).

Since inference must determine the structure of the entity trees in addition to coreference, it is advantageous to consider multiple MH proposals per sample. Therefore, we employ a modified variant of MH that is similar to multi-try Metropolis (Liu et al., 2000). Our modified MH algorithm makes k proposals and samples one according to its model ratio score (the first term in Equation 2) normalized across all k. More specificaly, for each MH step, we first randomly select two subtrees headed by node-

records r_i and r_j from the current coreference hypothesis. If r_i and r_j are in different clusters, we propose several alternate merge operations: (also in Figure 3):

• **Merge Left** - merges the entire subtree of r_j into node r_i by making r_j a child of r_i

• **Merge Entity Left** - merges r_j with r_i's root

• **Merge Left and Collapse** - merges r_j into r_i then performs a collapse on r_j (see below).

• **Merge Up** - merges node r_i with node r_j by creating a new parent node-record variable r^p with r_i and r_j as the children. The attribute fields of r^p are selected from r_i and r_j.

Otherwise r_i and r_j are subtrees in the same entity tree, then the following proposals are used instead:

• **Split Right** - Make the subtree r_j the root of a new entity by detaching it from its parent

• **Collapse** - If r_i has a parent, then move r_i's children to r_i's parent and then delete r_i.

• **Sample attribute** - Pick a new value for an attribute of r_i from its children.

Computing the model ratio for all of coreference proposals requires only a constant number of compatibility functions. On the other hand, evaluating proposals in the pairwise model requires evaluating a number of compatibility functions equal to the number of mentions in the clusters being modified.

Note that changes to the attribute values of the node-record and collapsing still require evaluating a linear number of factors, but this is only linear in the number of child nodes, not linear in the number of mentions referring to the entity. Further, attribute values rarely change once the entities stabilize. Finally, we incrementally update bags during coreference to reflect the aggregates of their children.

4 Experiments: Author Coreference

Author coreference is a tremendously important task, enabling improved search and mining of scientific papers by researchers, funding agencies, and governments. The problem is extremely difficult due to the wide variations of names, limited contextual evidence, misspellings, people with common names, lack of standard citation formats, and large numbers of mentions.

For this task we use a publicly available collection of 4,394 BibTeX files containing 817,193 entries.[3] We extract 1,322,985 author mentions, each containing first, middle, last names, bags-of-words of paper titles, topics in paper titles (by running latent Dirichlet allocation (Blei et al., 2003)), and last names of co-authors. In addition we include 2,833 mentions from the REXA dataset[4] labeled for coreference, in order to assess accuracy. We also include $\sim$5 million mentions from DBLP.

4.1 Models and Inference

Due to the paucity of labeled training data, we did not estimate parameters from data, but rather set the compatibility functions manually by specifying their log scores. The pairwise compatibility functions punish a string difference in first, middle, and last name, (-8); reward a match $(+2)$; and reward matching initials $(+1)$. Additionally, we use the cosine similarity (shifted and scaled between -4 and 4) between the bags-of-words containing title tokens, topics, and co-author last names. These compatibility functions define the scores of the factors in the pairwise model and the parent-child factors in the hierarchical model. Additionally, we include priors over the model structure. We encourage each node to have eight children using a per node factor having score $1/(|\text{number of children}–8|+1)$, manage tree depth by placing a cost on the creation of intermediate tree nodes -8 and encourage clustering by placing a cost on the creation of root-level entities -7. These weights were determined by just a few hours of tuning on a development set.

We initialize the MCMC procedures to the singleton configuration (each entity consists of one mention) for each model, and run the MH algorithm described in Section 2.2 for the pairwise model and multi-try MH (described in Section 3.2) for the hierarchical model. We augment these samplers using canopies constructed by concatenating the first initial and last name: that is, mentions are only selected from within the same canopy (or block) to reduce the search space (Bilenko et al., 2006). During the course of MCMC inference, we record the pairwise F1 scores of the labeled subset. The source code for our model is available as part of the FACTORIE package (McCallum et al., 2009, `http:`

[3] `http://www.iesl.cs.umass.edu/data/bibtex`

[4] `http://www2.selu.edu/Academics/Faculty/aculotta/data/rexa.html`

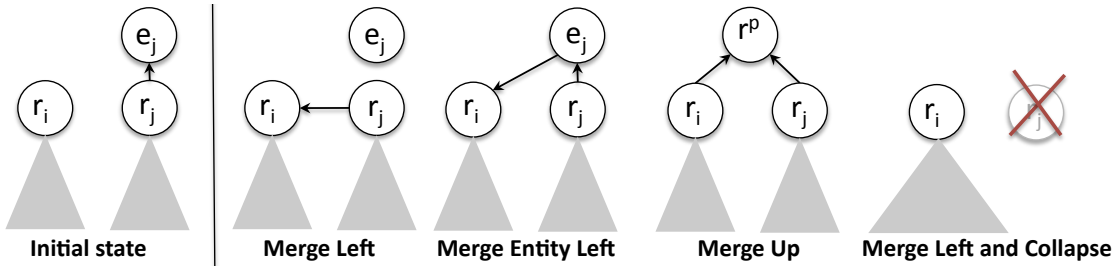


Figure 3: Example coreference proposals for the case where r_i and r_j are initially in different clusters.

//factorie.cs.umass.edu/).

4.2 Comparison to Pairwise Model

In Figure 4a we plot the number of samples completed over time for a 145k subset of the data. Recall that we initialized to the singleton configuration and that as the size of the entities grows, the cost of evaluating the entities in MCMC becomes more expensive. The pairwise model struggles with the large cluster sizes while the hierarchical model is hardly affected. Even though the hierarchical model is evaluating up to four proposals for each sample, it is still able to sample much faster than the pairwise model; this is expected because the cost of evaluating a proposal requires evaluating fewer factors. Next, we plot coreference F1 accuracy over time and show in Figure 5a that the prolific sampling rate of the hierarchical model results in faster coreference. Using the plot, we can compare running times for any desired level of accuracy. For example, on the 145k mention dataset, at a 60% accuracy level the hierarchical model is 19 times faster and at 90% accuracy it is 31 times faster. These performance improvements are even more profound on larger datasets: the hierarchical model achieves a 60% level of accuracy 72 times faster than the pairwise model on the 1.3 million mention dataset, reaching 90% in just 2,350 seconds. Note, however, that the hierarchical model requires more samples to reach a similar level of accuracy due to the larger state space (Figure 4b).

4.3 Large Scale Experiments

In order to demonstrate the scalability of the hierarchical model, we run it on nearly 5 million author mentions from DBLP. In under two hours (6,700 seconds), we achieve an accuracy of 80%, and in under three hours (10,600 seconds), we achieve an accuracy of over 90%. Finally, we combine DBLP with BibTeX data to produce a dataset with almost 6 million mentions (5,803,811). Our performance on this dataset is similar to DBLP, taking just 13,500 seconds to reach a 90% accuracy.

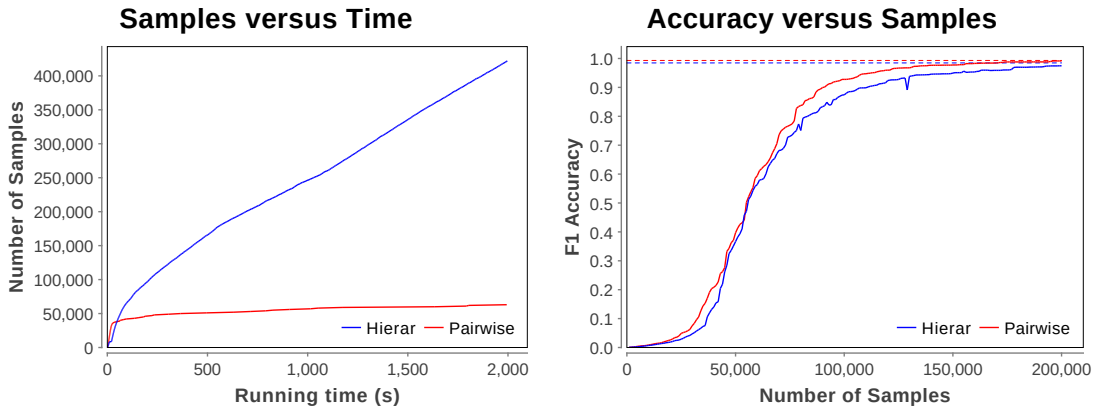
5 Related Work

Singh et al. (2011) introduce a hierarchical model for coreference that treats entities as a two-tiered structure, by introducing the concept of sub-entities and super-entities. Super-entities reduce the search space in order to propose fruitful jumps. Sub-entities provide a tighter granularity of coreference and can be used to perform larger block moves during MCMC. However, the hierarchy is fixed and shallow. In contrast, our model can be arbitrarily deep and wide. Even more importantly, their model has pairwise factors and suffers from the quadratic curse, which they address by distributing inference.

The work of Rao et al. (2010) uses streaming clustering for large-scale coreference. However, the greedy nature of the approach does not allow errors to be revisited. Further, they compress entities by averaging their mentions' features. We are able to provide richer entity compression, the ability to revisit errors, and scale to larger data.

Our hierarchical model provides the advantages of recently proposed entity-based coreference systems that are known to provide higher accuracy (Haghighi and Klein, 2007; Culotta et al., 2007; Yang et al., 2008; Wick et al., 2009; Haghighi and Klein, 2010). However, these systems reason over a single layer of entities and do not scale well.

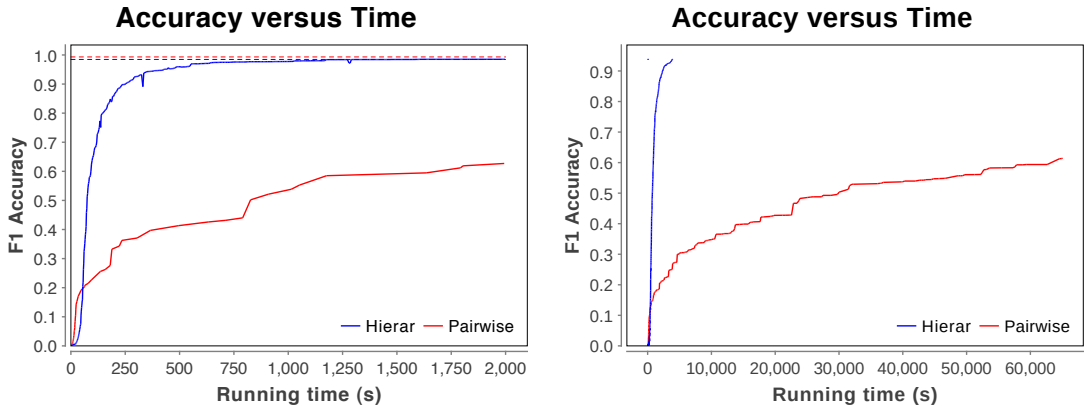
Techniques such as lifted inference (Singla and Domingos, 2008) for graphical models exploit redundancy in the data, but typically do not achieve any significant compression on coreference data be-



(a) Sampling Performance

(b) Accuracy vs. samples (convergence accuracy as dashes)

Figure 4: Sampling Performance Plots for 145k mentions



(a) Accuracy vs. time (145k mentions)

(b) Accuracy vs. time (1.3 million mentions)

Figure 5: Runtime performance on two datasets

cause the observations usually violate any symmetry assumptions. On the other hand, our model is able to compress similar (but potentially different) observations together in order to make inference fast even in the presence of asymmetric observed data.

6 Conclusion

In this paper we present a new hierarchical model for large scale coreference and demonstrate it on the problem of author disambiguation. Our model recursively defines an entity as a summary of its children nodes, allowing succinct representations of millions of mentions. Indeed, inference in the hierarchy is orders of magnitude faster than a pairwise CRF, allowing us to infer accurate coreference on six million mentions on one CPU in just 4 hours.

7 Acknowledgments

We would like to thank Veselin Stoyanov for his feedback. This work was supported in part by the CIIR, in part by ARFL under prime contract #FA8650-10-C-7059, in part by DARPA under AFRL prime contract #FA8750-09-C-0181, and in part by IARPA via DoI/NBC contract #D11PC20152. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the sponsor.

References

Amit Bagga and Breck Baldwin. 1999. Cross-document event coreference: annotations, experiments, and observations. In *Proceedings of the Workshop on Coreference and its Applications*, CorefApp '99, pages 1–8, Stroudsburg, PA, USA. Association for Computational Linguistics.

Eric Bengston and Dan Roth. 2008. Understanding the value of features for coreference resolution. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Indrajit Bhattacharya and Lise Getoor. 2006. A latent Dirichlet model for unsupervised entity resolution. In *SDM*.

Mikhail Bilenko, Beena Kamath, and Raymond J. Mooney. 2006. Adaptive blocking: Learning to scale up record linkage. In *Proceedings of the Sixth International Conference on Data Mining*, ICDM '06, pages 87–96, Washington, DC, USA. IEEE Computer Society.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent Dirichlet allocation. *Journal on Machine Learning Research*, 3:993–1022.

Aron Culotta and Andrew McCallum. 2006. Practical Markov logic containing first-order quantifiers with application to identity uncertainty. In *Human Language Technology Workshop on Computationally Hard Problems and Joint Inference in Speech and Language Processing (HLT/NAACL)*, June.

Aron Culotta, Michael Wick, and Andrew McCallum. 2007. First-order probabilistic models for coreference resolution. In *North American Chapter of the Association for Computational Linguistics - Human Language Technologies (NAACL HLT)*.

Aria Haghighi and Dan Klein. 2007. Unsupervised coreference resolution in a nonparametric bayesian model. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 848–855.

Aria Haghighi and Dan Klein. 2010. Coreference resolution in a modular, entity-centered model. In *North American Chapter of the Association for Computational Linguistics - Human Language Technologies (NAACL HLT)*, pages 385–393.

Mauricio A. Hernández and Salvatore J. Stolfo. 1995. The merge/purge problem for large databases. In *Proceedings of the 1995 ACM SIGMOD international conference on Management of data*, SIGMOD '95, pages 127–138, New York, NY, USA. ACM.

Jun S. Liu, Faming Liang, and Wing Hung Wong. 2000. The multiple-try method and local optimization in metropolis sampling. *Journal of the American Statistical Association*, 96(449):121–134.

Andrew McCallum and Ben Wellner. 2004. Conditional models of identity uncertainty with application to noun coreference. In *Neural Information Processing Systems (NIPS)*.

Andrew McCallum, Kamal Nigam, and Lyle Ungar. 2000. Efficient clustering of high-dimensional data sets with application to reference matching. In *International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 169–178.

Andrew McCallum, Karl Schultz, and Sameer Singh. 2009. FACTORIE: Probabilistic programming via imperatively defined factor graphs. In *Neural Information Processing Systems (NIPS)*.

Brian Milch, Bhaskara Marthi, and Stuart Russell. 2006. *BLOG: Relational Modeling with Unknown Objects*. Ph.D. thesis, University of California, Berkeley.

Radford Neal. 2000. Slice sampling. *Annals of Statistics*, 31:705–767.

Vincent Ng. 2005. Machine learning for coreference resolution: From local classification to global ranking. In *Annual Meeting of the Association for Computational Linguistics (ACL)*.

Hoifung Poon and Pedro Domingos. 2007. Joint inference in information extraction. In *AAAI Conference on Artificial Intelligence*, pages 913–918.

Altaf Rahman and Vincent Ng. 2009. Supervised models for coreference resolution. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2 - Volume 2*, EMNLP '09, pages 968–977, Stroudsburg, PA, USA. Association for Computational Linguistics.

Delip Rao, Paul McNamee, and Mark Dredze. 2010. Streaming cross document entity coreference resolution. In *International Conference on Computational Linguistics (COLING)*, pages 1050–1058, Beijing, China, August. Coling 2010 Organizing Committee.

Yael Ravin and Zunaid Kazi. 1999. Is Hillary Rodham Clinton the president? disambiguating names across documents. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 9–16.

Matthew Richardson and Pedro Domingos. 2006. Markov logic networks. *Machine Learning*, 62(1-2):107–136.

Sameer Singh, Michael L. Wick, and Andrew McCallum. 2010. Distantly labeling data for large scale cross-document coreference. *Computing Research Repository (CoRR)*, abs/1005.4298.

Sameer Singh, Amarnag Subramanya, Fernando Pereira, and Andrew McCallum. 2011. Large-scale cross-document coreference using distributed inference and hierarchical models. In *Association for Computational Linguistics: Human Language Technologies (ACL HLT)*.

Parag Singla and Pedro Domingos. 2005. Discriminative training of Markov logic networks. In *AAAI*, Pittsburgh, PA.

Parag Singla and Pedro Domingos. 2008. Lifted first-order belief propagation. In *Proceedings of the 23rd national conference on Artificial intelligence - Volume 2*, AAAI'08, pages 1094–1099. AAAI Press.

Wee Meng Soon, Hwee Tou Ng, and Daniel Chung Yong Lim. 2001. A machine learning approach to coreference resolution of noun phrases. *Comput. Linguist.*, 27(4):521–544.

R.H. Swendsen and J.S. Wang. 1987. Nonuniversal critical dynamics in MC simulations. *Phys. Rev. Lett.*, 58(2):68–88.

Michael Wick, Aron Culotta, Khashayar Rohanimanesh, and Andrew McCallum. 2009. An entity-based model for coreference resolution. In *SIAM International Conference on Data Mining (SDM)*.

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 *Association for Computational Linguistics*, pages 843–851.