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 the severity of these drawbacks continue to progress with the growing demand for more data, the need to replace the pairwise approaches with a more expressive, highly scalable alternative is becoming increasingly urgent. In this paper we propose a novel discriminative hierarchical model that recursively structures entities into trees. These trees succinctly summarize the mentions providing a highly-compact information-rich structure for reasoning about entities and coreference uncertainty at small, large, and massive scales. The unique recursive structure of our entities allows our model to adapt to entities of various sizes, express features over entity hierarchies, and scale to massive data, making our approach a desirable new standard to replace the antiquated pairwise model.

1 Introduction

Coreference resolution, the task of clustering *mentions* into the real-world *entities* they refer to, is fundamental for high-level information extraction and data integration problems 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" that authored "Error detecting and error correcting codes" is the same" "R. Hamming" that 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, many 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, 2005; 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*? While these models have been quite successful in some domains, they also exhibit a number of undesirable characteristics. The first problem is that pairwise models lack the expressivity required to represent properties of the entities themselves. 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., 2007b; 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 by the literature, is that the pairwise

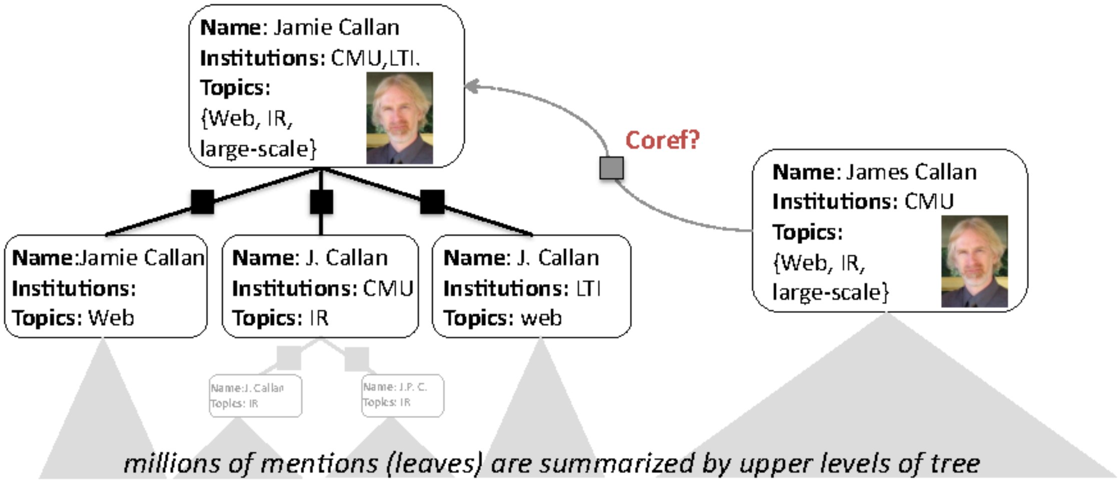
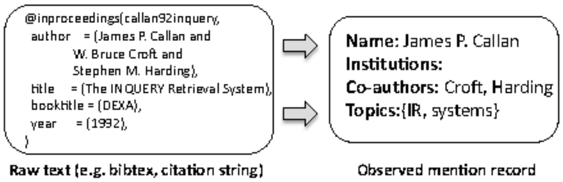


Figure 1: **Discriminative hierarchical factor graph for coreference:** Nodes summarize lower-levels of the tree. Pairwise factors (compatibility functions), indicated by black boxes, lie between each child and its parent, avoiding quadratic blow-up. Deciding whether to merge the two entities requires evaluating just a single factor, corresponding to the new child-parent relationship (indicated as the gray box).

coreference models scale poorly to voluminous collections of mentions where the expected number of mentions in each entity cluster is large. Current systems cope with this by either blocking the data 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), or employing other clever inference techniques to reduce the number of comparisons (Milch et al., 2005; Richardson and Domingos, 2006; 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 the prohibitive computational costs associated with large datasets. This dependence on entity size seems particularly wasteful because although it is common for an entity to be referred to 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 require comparisons to all contextually similar "Obama" mentions; rather we prefer a more compact representation for efficiently reasoning about them.

In this paper we propose a novel hierarchical discriminative factor graph for coreference resolution that recursively structures each entity as a tree. Our hierarchical model avoids the aforementioned concerns of the pairwise approach: not only can it jointly reason about attributes of entire entities (using the power of discriminative conditional random fields), it is 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 mutually beneficial to inference, resulting in improved sampling power similar to data-augmentation methods from statistical physics (e.g., Swendsen and Wang (1987)). Moreover, each step of inference is computationally efficient because evaluating the cost of at-



```
@inproceedings(callan92inquery,
 author    = (James P. Callan and
            W. Bruce Croft and
            Stephen M. Harding),
 title     = (The INQUERY Retrieval System),
 booktitle = (DEXA),
 year      = (1992),
)
```

Name: James P. Callan
Institutions:
Co-authors: Croft, Harding
Topics:{IR, systems}

Figure 2: An example of an observed author coreference mention extracted from a BibTeX record.

taching (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 a different types of data and effectively compress entities of different scale. Second, the model also 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 semantically interesting interpretations of the entity by placing contextually similar mentions under the same subtree. Finally, if memory is limited, redundant mentions can easily be pruned by replacing subtrees with their roots.

Empirically, we demonstrate that our model is several orders of magnitudes faster than a pairwise model allowing us to perform coreference on nearly six million author mentions obtained by combining DBLP with additional BibTeX files spidered from the web.

2 Background: Pairwise Coreference

Coreference is the problem of clustering mentions into entities, and is also known as entity disambiguation/resolution and record linkage/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 (see Figure 2). The mention record may contain attributes for the first, middle, and last name of the author, as well as some contextual information such as bags of words of co-authors, titles, topics, and institutions that may occur in the citation string. The goal is to cluster these mention records into the real-world academic authors to which they refer; we use this task as a running pedagogical example.

If we let $\mathcal{M}$ be the space of observed mention records, then the traditional pairwise coreference approach models the problem 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 a log-linear combination of features and weights, 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' bag of words contexts. The corresponding real-valued weights $\boldsymbol{\theta}$ determine the impact of these features on the overall pairwise score.

Given this pairwise compatibility function, coreference is solved by clustering mentions into the entities that they refer. 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 using statistical inference (McCallum and Wellner, 2005). 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:

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

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

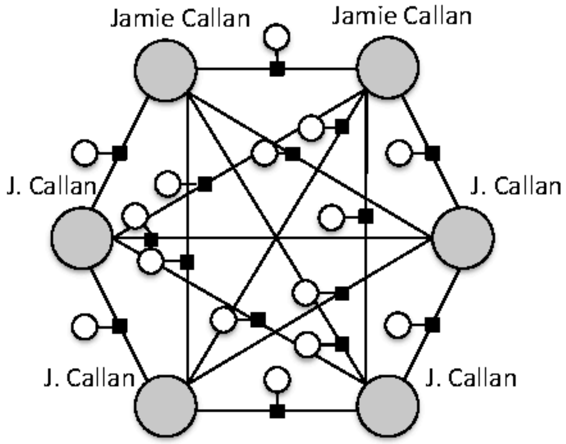


Figure 3: **Pairwise model on six mentions:** Open circles are the binary coreference decision variables, shaded circles are the observed mentions, and the black boxes are the factors of the graphical model that encode the pairwise compatibility functions.

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 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 the MCMC-based Metropolis-Hastings (MH) algorithm (Milch et al. (2005), Culotta and McCallum (2006), Poon and Domingos (2007), amongst others). 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:

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

When using MH for MPE inference, the second term $q(\mathbf{y}'|\mathbf{y})/q(\mathbf{y}|\mathbf{y}')$ is optional, and usually omitted. An important observation is that 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 only exploring the feasible space.

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.

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 which entity m refers to. 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—because of transitivity—in order to complete the move into e_j we must tie 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 size of the entities, requiring the evaluation of $2(|e_i| + |e_j| - 1)$ compatibility functions (factors).

3 Hierarchical Coreference

Instead of only capturing a single notion of coreference between mention pairs, we can imagine more

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

specific notions of coreference over multiple 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 may be further divided based on a yet another measure of similarity. In this section, we describe a model that captures arbitrarily deep hierarchies over such layers of coreference decisions, facilitating efficient inference and rich representations for entity-level reasoning.

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 as before with a set of attributes as shown earlier in Figure 2. Each internal node of the tree also 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 defining its mentions. More 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 hierarchical notion 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 intuition for why this may be the case 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).

We now describe our modified Metropolis-Hastings inference algorithm. In classic MH, a sample is generated by choosing to accept or reject a single proposal (according to Equation 2). However, because inference in our model must also infer the structure of the entity trees, it considers multiple proposals. For each sample we make k proposals and sample one according to its model ratio score (the first term in Equation 2). More specifically, for each MH step, we first randomly select two sub-trees headed by node-records r_i and r_j from the current coreference hypothesis. If r_i and r_j are part of different clusters, we propose several alternate merge operations: (also illustrated in Figure 4):

- **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** - similar to merge-left, but instead takes all the children of r_j and moves them to be children of r_i and then deletes the vacuous head r_j.
- **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 determined using a canonicalization function that takes the attributes of r_i and r_j as input and chooses among them. The bags of words for r^p are combined by accumulating the word counts

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.

Computing the model ratio for many of these proposals requires only a constant number of compatibility functions. On the other hand, for evaluating proposals in the pairwise model, we must compute 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 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 note that if a decomposable similarity metric like cosine distance is used, then we do not have to re-evaluate compatibilities with the children when bags of word counts are incrementally updated.

4 Experiments

We evaluate the performance on the problem of author coreference resolution.

4.1 Author Coreference

Analysis of bibliographic data is important as the knowledge generated by the scientific communities continues grow. Author coreference forms a major sub-component to understand and provide scientific data to researchers, funding agencies, and governments, by comprehensively identifying the contributions of individual scientists. 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.

To gather the data for this task we spider the web for BibTeX files and collect 4394 .bib files containing 817,193 entries. We extract 1,322,985 author mentions with records containing the first, middle, last names, and construct bags of words from the tokens in paper titles, topics in paper titles (by running latent Dirichlet allocation (Blei et al., 2003)), and last names of co-authors; we intend to release this dataset to the community. In addition we include 2833 labeled mentions from the REXA dataset (Culotta et al., 2007a) for accuracy evaluation. We also include ∼5 million mentions from DBLP.

4.2 Models and Inference

Due to the paucity of labeled training data, we construct the compatibility functions manually by specifying their log scores. The pairwise compatibility functions punish a mismatch of first, middle, and last name, (-8); reward a match $(+2)$; and reward for initials matching $(+1)$. Additionally, we use the cosine similarity (shifted and scaled between -4 and 4) between title tokens, topics, and co-author last names. These compatibility functions become 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 $1/(|\text{number of children}-8|+1)$, 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.

We initialize the MCMC procedures to the singleton configuration for each model, and run the MH algorithm described in Section 2.2 for the pairwise model and run modified MH algorithm described in

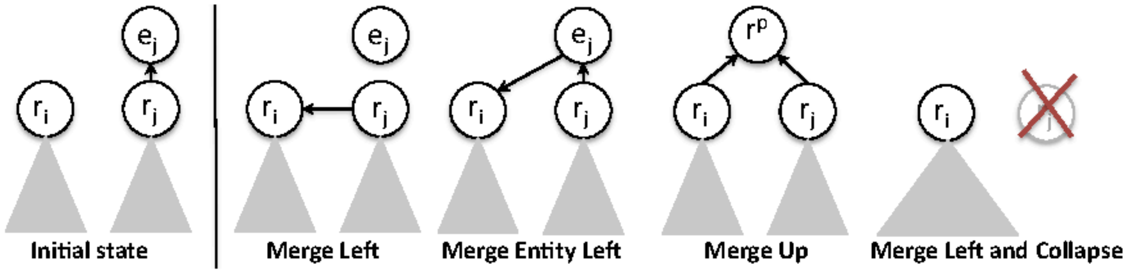


Figure 4: Example coreference proposals for the case where r_i and r_j are initially in different clusters.

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.

4.3 Comparison to Pairwise Model

In Figure 5a we plot the number of samples 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, which is expected because the cost of evaluating a proposal requires evaluating much fewer factors.

Next, we plot coreference F1 accuracy over time and show in Figure 6a 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 takes a larger number of samples to reach a similar level of accuracy due to the larger state space (Figure 5b).

4.4 Massive-Scale Experiments

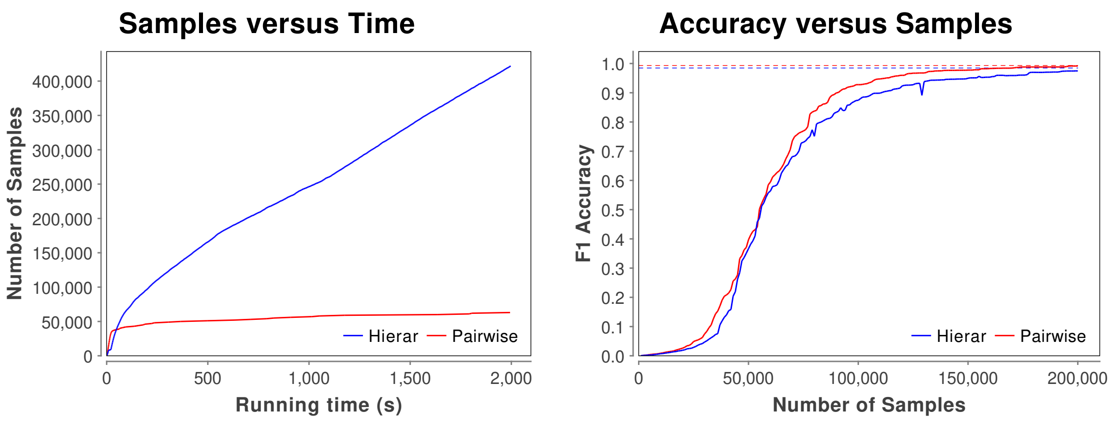
In order to demonstrate the scalability of the hierarchical model, we run it on nearly 5 million author mentions from DBLP. In under two hours (6,700 seconds), we achieve an accuracy of 80%, and in under three hours (10,600 seconds), we achieve an accuracy of over 90%. Finally, we combine DBLP with our spidered dataset 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

The hierarchical coreference model of Singh et al. (2011) 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 in this model is fixed and shallow. In contrast, our model can be arbitrarily deep and wide. Additionally, their model contains pairwise factors and may suffer from the quadratic curse.

The work of Rao et al. (2010) uses streaming clustering for large-scale coreference. However, the greedy nature of the approach means errors can never be revisited. Further, they summarize entities on just a single level by averaging the mention feature vectors. We are able to provide richer entity compression, the ability to revisit errors, and scale to larger data.

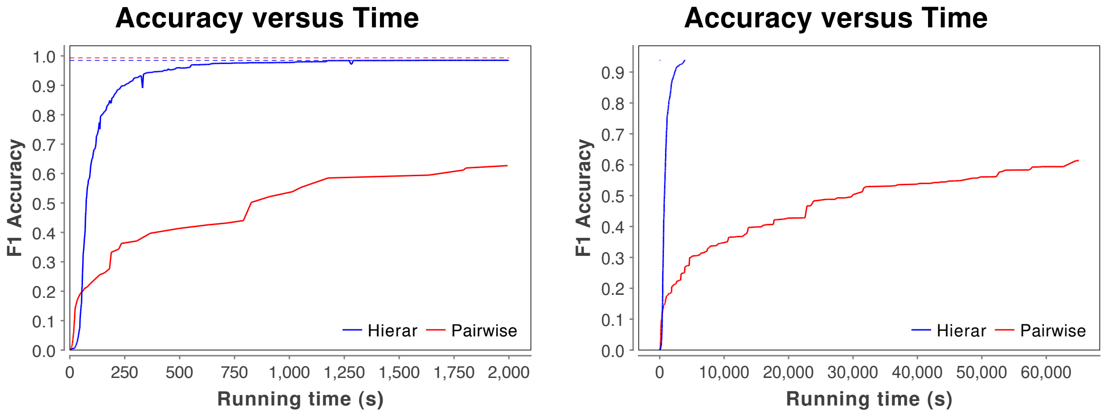
Our hierarchical model provides the advantages of recently proposed entity-based coreference systems that are known to provide higher accuracy (Haghighi and Klein, 2007; Culotta et al., 2007b;



(a) Sampling Performance

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

Figure 5: Sampling Performance Plots for 145k mentions



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

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

Figure 6: Runtime performance on two datasets

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.

Techniques such as lifted inference (Singla and Domingos, 2008) for graphical models attempt to exploit redundancy in the data, but fail to achieve any significant compression on real-world data because the observations usually negate 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 real observed data.

6 Conclusion

In this paper we presented a new hierarchical model for large scale coreference and applied it to 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 scale to six million mentions taken from DBLP and a web crawl. In future work we would like to investigate and manipulate the semantic meaning of the entity hierarchies in order to provide interpretable summaries of large datasets, such as all the entities in mentioned in New York Times.

7 Acknowledgments

This work was supported in part by the Center for Intelligent Information Retrieval and in part by IARPA via DoI/NBC contract #D11PC20152. Also, the University of Massachusetts gratefully acknowledges the support of Defense Advanced Research Projects Agency (DARPA) Machine Reading Program under Air Force Research Laboratory (AFRL) prime contract no. FA8750-09-C-0181. Any opinions, findings, and conclusion or recommendations expressed in this material are those of the authors and do not necessarily reflect the view of DARPA, AFRL, or the US government , The U.S. Government is authorized to reproduce and distribute reprint for Governmental purposes notwithstanding any copyright annotation thereon.

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. *J. Mach. Learn. Res.*, 3:993–1022, March.

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, Pallika Kanani, Robert Hall, Michael Wick, and Andrew McCallum. 2007a. Author disambiguation using error-driven machine learning with a ranking loss function. In *Sixth International Workshop on Information Integration on the Web (IIWeb-07)*, Vancouver, Canada.

Aron Culotta, Michael Wick, Robert Hall, and Andrew McCallum. 2007b. First-order probabilistic models for coreference resolution. In *Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics (HLT/NAACL)*, pages 81–88.

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*, pages 848–855, Prague, Czech Republic, June. Association for Computational Linguistics.

Aria Haghighi and Dan Klein. 2010. Coreference resolution in a modular, entity-centered model. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT '10, pages 385–393, Stroudsburg, PA, USA. Association for Computational Linguistics.

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.

Andrew McCallum and Ben Wellner. 2005. Conditional models of identity uncertainty with application to noun coreference. In Lawrence K. Saul, Yair Weiss, and Léon Bottou, editors, *NIPS17*. MIT Press, Cambridge, MA.

Andrew K. McCallum, Kamal Nigam, and Lyle Ungar. 2000. Efficient clustering of high-dimensional data sets with application to reference matching. In *Proceedings of the Sixth International Conference On Knowledge Discovery and Data Mining (KDD-2000)*, Boston, MA.

Brian Milch, Bhaskara Marthi, Stuart Russell, David Sontag, Daniel L. Ong, and Andrey Kolobov. 2005. BLOG: Probabilistic models with unknown objects. In *IJCAI*.

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 *Proceedings of the 22nd national conference on Artificial intelligence - Volume 1*, pages 913–918. AAAI Press.

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 *COLING (Posters)*, pages 1050–1058.

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*, 62:107–136.

Sameer Singh, Amarnag Subramanya, Fernando C. N. Pereira, and Andrew McCallum. 2011. Large-scale cross-document coreference using distributed inference and hierarchical models. In *ACL*, pages 793–803.

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*.

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 *ACL*, pages 843–851.