2011

Large-Scale Cross-Document Coreference Using Distributed Inference and Hierarchical Models

Sameer Singh
Amarnag Subramanya
Refnando Pereira
Andrew McCallum, *University of Massachusetts - Amherst*

Large-Scale Cross-Document Coreference Using Distributed Inference and Hierarchical Models

Sameer Singh[§] **Amarnag Subramanya**[†] **Fernando Pereira**[†] **Andrew McCallum**[§]
[§] Department of Computer Science, University of Massachusetts, Amherst MA 01002
[†] Google Research, Mountain View CA 94043
sameer@cs.umass.edu, asubram@google.com, pereira@google.com, mccallum@cs.umass.edu

Abstract

Cross-document coreference, the task of grouping all the mentions of each entity in a document collection, arises in information extraction and automated knowledge base construction. For large collections, it is clearly impractical to consider all possible groupings of mentions into distinct entities. To solve the problem we propose two ideas: (a) a *distributed* inference technique that uses parallelism to enable large scale processing, and (b) a *hierarchical* model of coreference that represents uncertainty over multiple granularities of entities to facilitate more effective approximate inference. To evaluate these ideas, we constructed a labeled corpus of 1.5 million disambiguated mentions in Web pages by selecting link anchors referring to Wikipedia entities. We show that the combination of the hierarchical model with distributed inference quickly obtains high accuracy (with error reduction of 38%) on this large dataset, demonstrating the scalability of our approach.

1 Introduction

Given a collection of mentions of entities extracted from a body of text, *coreference* or *entity resolution* consists of clustering the mentions such that two mentions belong to the same cluster if and only if they refer to the same entity. Solutions to this problem are important in semantic analysis and knowledge discovery tasks (Blume, 2005; Mayfield et al., 2009). While significant progress has been made in *within*-document coreference (Ng, 2005; Culotta et al., 2007; Haghighi and Klein, 2007; Bengston and Roth, 2008; Haghighi and Klein, 2009; Haghighi and Klein, 2010), the larger problem of *cross*-document coreference has not received as much attention.

Unlike inference in other language processing tasks that scales linearly in the size of the corpus, the hypothesis space for coreference grows super-exponentially with the number of mentions. Consequently, most of the current approaches are developed on small datasets containing a few thousand mentions. We believe that cross-document coreference resolution is most useful when applied to a very large set of documents, such as all the news articles published during the last 20 years. Such a corpus would have billions of mentions. In this paper we propose a model and inference algorithms that can scale the cross-document coreference problem to corpora of that size.

Much of the previous work in cross-document coreference (Bagga and Baldwin, 1998; Ravin and Kazi, 1999; Gooi and Allan, 2004; Pedersen et al., 2006; Rao et al., 2010) groups mentions into entities with some form of greedy clustering using a pairwise mention similarity or distance function based on mention text, context, and document-level statistics. Such methods have not been shown to scale up, and they cannot exploit cluster features that cannot be expressed in terms of mention pairs. We provide a detailed survey of related work in Section 6.

Other previous work attempts to address some of the above concerns by mapping coreference to inference on an undirected graphical model (Culotta et al., 2007; Poon et al., 2008; Wellner et al., 2004; Wick et al., 2009a). These models contain pairwise factors between all pairs of mentions capturing similarity between them. Many of these models also enforce transitivity and enable features over

Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics, pages 793–803,
Portland, Oregon, June 19-24, 2011.

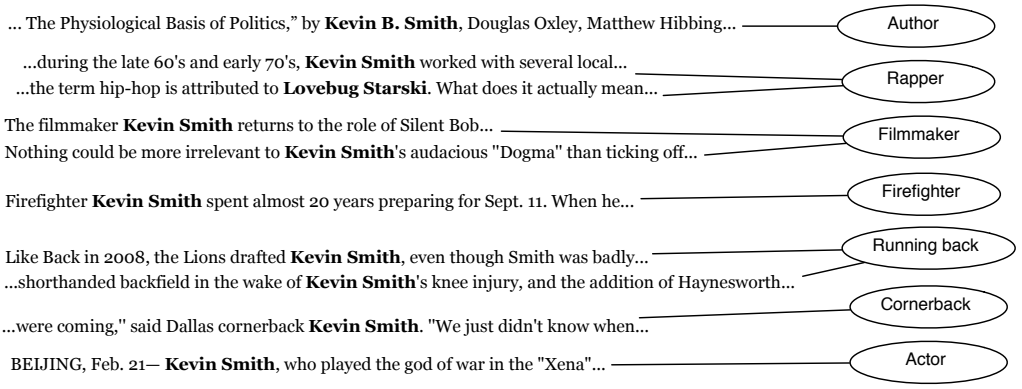

Figure 1: **Cross-Document Coreference Problem:** Example mentions of "Kevin Smith" from New York Times articles, with the true entities shown on the right.

entities by including *set-valued variables*. Exact inference in these models is intractable and a number of approximate inference schemes (McCallum et al., 2009; Rush et al., 2010; Martins et al., 2010) may be used. In particular, *Markov chain Monte Carlo* (MCMC) based inference has been found to work well in practice. However as the number of mentions grows to Web scale, as in our problem of cross-document coreference, even these inference techniques become infeasible, motivating the need for a scalable, parallelizable solution.

In this work we first distribute MCMC-based inference for the graphical model representation of coreference. Entities are distributed across the machines such that the parallel MCMC chains on the different machines use only local proposal distributions. After a fixed number of samples on each machine, we redistribute the entities among machines to enable proposals across entities that were previously on different machines. In comparison to the greedy approaches used in related work, our MCMC-based inference provides better robustness properties.

As the number of mentions becomes large, high-quality samples for MCMC become scarce. To facilitate better proposals, we present a hierarchical model. We add *sub-entity* variables that represent clusters of similar mentions that are likely to be coreferent; these are used to propose composite jumps that move multiple mentions together. We also introduce *super-entity* variables that represent clusters of similar entities; these are used to distribute entities among the machines such that similar entities are assigned to the same machine. These additional levels of hierarchy dramatically increase the probability of beneficial proposals even with a large number of entities and mentions.

To create a large corpus for evaluation, we identify pages that have hyperlinks to Wikipedia, and extract the anchor text and the context around the link. We treat the anchor text as the mention, the context as the document, and the title of the Wikipedia page as the entity label. Using this approach, 1.5 million mentions were annotated with 43k entity labels. On this dataset, our proposed model yields a B^3 (Bagga and Baldwin, 1998) F1 score of 73.7%, improving over the baseline by 16% absolute (corresponding to 38% error reduction). Our experimental results also show that our proposed hierarchical model converges much faster even though it contains many more variables.

2 Cross-document Coreference

The problem of coreference is to identify the sets of mention strings that refer to the same underlying entity. The identities and the number of the underlying entities is not known. In *within-document* coreference, the mentions occur in a single document. The number of mentions (and entities) in each document is usually in the hundreds. The difficulty of the task arises from a large hypothesis space (exponential in the number of mentions) and challenge in resolving nominal and pronominal mentions to the correct named mentions. In most cases, named mentions

are not ambiguous within a document. In *cross-document* coreference, the number of mentions and entities is in the millions, making the combinatorics even more daunting. Furthermore, naming ambiguity is much more common as the same string can refer to multiple entities in different documents, and distinct strings may refer to the same entity in different documents.

We show examples of ambiguities in Figure 1. Resolving the identity of individuals with the *same name* is a common problem in cross-document coreference. This problem is further complicated by the fact that in some situations, these individuals may belong to the same field. Another common ambiguity is that of *alternate* names, in which the same entity is referred to by different names or *aliases* (e.g. "Bill" is often used as a substitute for "William"). The figure also shows an example of the *renaming* ambiguity – "Lovebug Starski" refers to "Kevin Smith", and this is an extreme form of alternate names. Rare *singleton* entities (like the firefighter) that may appear only once in the whole corpus are also often difficult to isolate.

2.1 Pairwise Factor Model

Factor graphs are a convenient representation for a probability distribution over a vector of output variables given observed variables. The model that we use for coreference represents mentions ($\mathbf{M}$) and entities ($\mathbf{E}$) as random variables. Each mention can take an entity as its value, and each entity takes a set of mentions as its value. Each mention also has a feature vector extracted from the observed text mention and its context. More precisely, the probability of a configuration $\mathbf{E} = \mathbf{e}$ is defined by

$$p(\mathbf{e}) \propto \exp \sum_{e \in \mathbf{e}} \left\{ \sum_{m,n \in e, n \neq m} \psi_a(m,n) + \sum_{m \in e, n \notin e} \psi_r(m,n) \right\}$$

where factor ψ_a represents *affinity* between mentions that are coreferent according to $\mathbf{e}$, and factor ψ_r represents *repulsion* between mentions that are not coreferent. Different factors are instantiated for different predicted configurations. Figure 2 shows the model instantiated with five mentions over a two-entity hypothesis.

For the factor potentials, we use cosine similarity of mention context pairs (ϕ_{mn}) such that

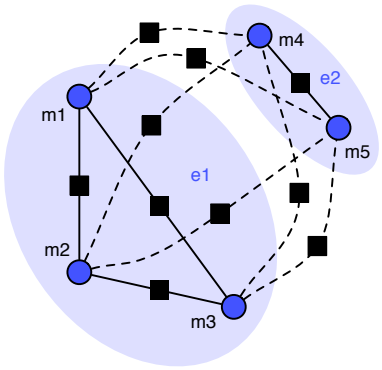

Figure 2: **Pairwise Coreference Model:** Factor graph for a 2-entity configuration of 5 mentions. Affinity factors are shown with solid lines, and repulsion factors with dashed lines.

$\psi_a(m,n) = \phi_{mn} - b$ and $\psi_r(m,n) = -(\phi_{mn} - b)$, where b is the bias. While one can certainly make use of a more sophisticated feature set, we leave this for future work as our focus is to scale up inference. However, it should be noted that this approach is agnostic to the particular set of features used. As we will note in the next section, we do not need to calculate features between all pairs of mentions (as would be prohibitively expensive for large datasets); instead we only compute the features as and when required.

2.2 MCMC-based Inference

Given the above model of coreference, we seek the *maximum a posteriori* (MAP) configuration:

$$\begin{aligned} \hat{\mathbf{e}} &= \arg\max_{\mathbf{e}} p(\mathbf{e}) \\ &= \arg\max_{\mathbf{e}} \sum_{e \in \mathbf{e}} \left\{ \sum_{m,n \in e, n \neq m} \psi_a(m,n) + \sum_{m \in e, n \notin e} \psi_r(m,n) \right\} \end{aligned}$$

Computing $\hat{\mathbf{e}}$ exactly is intractable due to the large space of possible configurations.[1] Instead, we employ MCMC-based optimization to discover the MAP configuration. A proposal function q is used to propose a change $\mathbf{e}'$ to the current configuration $\mathbf{e}$. This jump is accepted with the following Metropolis-Hastings acceptance probability:

$$\alpha(\mathbf{e}, \mathbf{e}') = \min\left(1, \left(\frac{p(\mathbf{e}')}{p(\mathbf{e})}\right)^{1/t} \frac{q(\mathbf{e})}{q(\mathbf{e}')}\right) \quad (1)$$

[1] Number of possible entities is Bell(n) in the number of mentions, i.e. number of partitions of n items

where t is the annealing temperature parameter.

MCMC chains efficiently explore the high-density regions of the probability distribution. By slowly reducing the temperature, we can decrease the entropy of the distribution to encourage convergence to the MAP configuration. MCMC has been used for optimization in a number of related work (McCallum et al., 2009; Goldwater and Griffiths, 2007; Changhe et al., 2004).

The proposal function moves a randomly chosen mention l from its current entity e_s to a randomly chosen entity e_t. For such a proposal, the log-model ratio is:

$$\log \frac{p(\mathbf{e}')}{p(\mathbf{e})} = \sum_{m \in e_t} \psi_a(l, m) + \sum_{n \in e_s} \psi_r(l, n) - \sum_{n \in e_s} \psi_a(l, n) - \sum_{m \in e_t} \psi_r(l, m) \quad (2)$$

Note that since only the factors between mention l and mentions in e_s and e_t are involved in this computation, the acceptance probability of each proposal is calculated efficiently.

In general, the model may contain arbitrarily complex set of features over pairs of mentions, with parameters associated with them. Given labeled data, these parameters can be *learned* by Perceptron (Collins, 2002), which uses the MAP configuration according to the model ($\hat{\mathbf{e}}$). There also exist more efficient training algorithms such as SampleRank (McCallum et al., 2009; Wick et al., 2009b) that update parameters *during* inference. However, we only focus on inference in this work, and the only parameter that we set manually is the bias b, which indirectly influences the number of entities in $\hat{\mathbf{e}}$. Unless specified otherwise, in this work the initial configuration for MCMC is the *singleton* configuration, i.e. all entities have a size of 1.

This MCMC inference technique, which has been used in McCallum and Wellner (2004), offers several advantages over other inference techniques: (a) unlike message-passing-methods, it does not require the full ground graph, (b) we only have to examine the factors that lie within the changed entities to evaluate a proposal, and (c) inference may be stopped at any point to obtain the current best configuration. However, the super exponential nature of the hypothesis space in cross-doc coreference renders this algorithm computationally unsuitable for large scale coreference tasks. In particular, fruitful proposals (that increase the model score) are extremely rare, resulting in a large number of proposals that are not accepted. We describe methods to speed up inference by 1) evaluating multiple proposal simultaneously (Section 3), and 2) by augmenting our model with hierarchical variables that enable better proposal distributions (Section 4).

3 Distributed MAP Inference

The key observation that enables distribution is that the acceptance probability computation of a proposal only examines a few factors that are **not** common to the previous and next configurations (Eq. 2). Consider a pair of proposals, one that moves mention l from entity e_s to entity e_t, and the other that moves mention l' from entity e'_s to entity e'_t. The set of factors to compute acceptance of the first proposal are factors between l and mentions in e_s and e_t, while the set of factors required to compute acceptance of the second proposal lie between l' and mentions in e'_s and e'_t. Since these set of factors are completely disjoint from each other, and the resulting configurations do not depend on each other, these two proposals are *mutually-exclusive*. Different orders of evaluating such proposals are equivalent, and in fact, these proposals can be proposed and evaluated concurrently. This mutual-exclusivity is not restricted only to pairs of proposals; a set of proposals are mutually-exclusive if no two proposals require the same factor for evaluation.

Using this insight, we introduce the following approach to distributed cross-document coreference. We divide the mentions and entities among multiple machines, and propose moves of mentions between entities assigned to the same machine. These jumps are evaluated exactly and accepted without communication between machines. Since acceptance of a mention's move requires examining factors that lie between other mentions in its entity, we ensure that all mentions of an entity are assigned the same machine. Unless specified otherwise, the distribution is performed randomly. To enable exploration of the complete configuration space, rounds of sampling are interleaved by *redistribution* stages, in which the entities are redistributed among the machines (see Figure 3). We use MapReduce (Dean and Ghe-

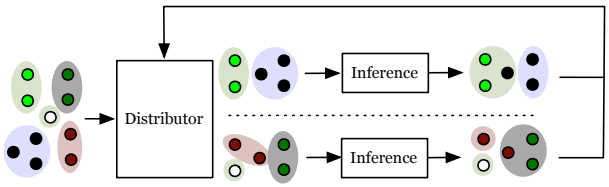

Figure 3: **Distributed MCMC-based Inference**: Distributor divides the entities among the machines, and the machines run inference. The process is repeated by the redistributing the entities.

mawat, 2004) to manage the distributed computation.

This approach to distribution is equivalent to inference with all mentions and entities on a single machine with a restricted proposer, but is faster since it exploits independencies to propose multiple jumps simultaneously. By restricting the jumps as described above, the acceptance probability calculation is exact. Partitioning the entities and proposing local jumps are restrictions to the single-machine proposal distribution; redistribution stages ensure the equivalent Markov chains are still irreducible. See Singh et al. (2010) for more details.

4 Hierarchical Coreference Model

The proposal function for MCMC-based MAP inference presents changes to the current entities. Since we use MCMC to reach high-scoring regions of the hypothesis space, we are interested in the changes that improve the current configuration. But as the number of mentions and entities increases, these *fruitful* samples become extremely rare due to the blowup in the possible space of configurations, resulting in rejection of a large number of proposals. By distributing as described in the previous section, we propose samples in parallel, improving chances of finding changes that result in better configurations. However, due to random redistribution and a naive proposal function within each machine, a large fraction of proposals are still wasted. We address these concerns by adding *hierarchy* to the model.

4.1 Sub-Entities

Consider the task of proposing moves of mentions (within a machine). Given the large number of mentions and entities, the probability that a randomly picked mention that is moved to a random entity results in a better configuration is extremely small. If such a move is accepted, this gives us evidence that the mention did not belong to the previous entity, and we should also move similar mentions from the previous entity simultaneously to the same entity. Since the proposer moves only a single mention at a time, a large number of samples may be required to discover these fruitful moves. To enable *block* proposals that move similar mentions simultaneously, we introduce latent *sub-entity* variables that represent groups of similar mentions within an entity, where the similarity is defined by the model. For inference, we have stages of sampling sub-entities (moving individual mentions) interleaved with stages of entity sampling (moving all mentions within a sub-entity). Even though our configuration space has become larger due to these extra variables, the proposal distribution has also improved since it proposes composite moves.

4.2 Super-Entities

Another issue faced during distributed inference is that random redistribution is often wasteful. For example, if dissimilar entities are assigned to a machine, none of the proposals may be accepted. For a large number of entities and machines, the probability that similar entities will be assigned to the same machine is extremely small, leading to a larger number of wasted proposals. To alleviate this problem, we introduce *super-entities* that represent groups of similar entities. During redistribution, we ensure all entities in the same super-entity are assigned to the same machine. As for sub-entities above, inference switches between regular sampling of entities and sampling of super-entities (by moving entities). Although these extra variables have made the configuration space larger, they also allow more efficient distribution of entities, leading to useful proposals.

4.3 Combined Hierarchical Model

Each of the described levels of the hierarchy are similar to the initial model (Section 2.1): mentions/sub-entities have the same structure as the entities/super-entities, and are modeled using similar factors. To represent the "context" of a sub-entity we take the union of the bags-of-words of the constituent mention contexts. Similarly, we take the union of sub-

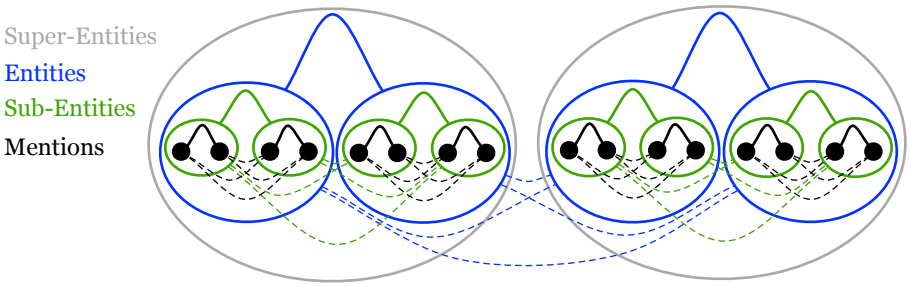

Figure 4: **Combined Hierarchical Model** with factors instantiated for a hypothesis containing 2 super-entities, 4 entities, and 8 sub-entities, shown as colored circles, over 16 mentions. Dotted lines represent repulsion factors and solid lines represent affinity factors (the color denotes the type of variable that the factor touches). The boxes on factors were excluded for clarity.

entity contexts to represent the context of an entity. The factors are instantiated in the same manner as Section 2.1 except that we change the bias factor b for each level (increasing it for sub-entities, and decreasing it for super-entities). The exact values of these biases indirectly determines the number of predicted sub-entities and super-entities.

Since these two levels of hierarchy operate at separate granularities from each other, we combine them into a single hierarchical model that contains both sub- and super-entities. We illustrate this hierarchical structure in Figure 4. Inference for this model takes a round-robin approach by fixing two of the levels of the hierarchy and sampling the third, cycling through these three levels. Unless specified otherwise, the initial configuration is the *singleton* configuration, in which all sub-entities, entities, and super-entities are of size 1.

5 Experiments

We evaluate our models and algorithms on a number of datasets. First, we compare performance on the small, publicly-available "John Smith" dataset. Second, we run the automated *Person-X* evaluation to obtain thousands of mentions that we use to demonstrate accuracy and scalability improvements. Most importantly, we create a large labeled corpus using links to Wikipedia to explore the performance in the large-scale setting.

5.1 John Smith Corpus

To compare with related work, we run an evaluation on the "John Smith" corpus (Bagga and Baldwin, 1998), containing 197 mentions of the name "John Smith" from New York Times articles (labeled to obtain 35 true entities). The bias b for our approach is set to result in the correct number of entities. Our model achieves B^3 F1 accuracy of 66.4% on this dataset. In comparison, Rao et al. (2010) obtains 61.8% using the model most similar to ours, while their best model (which uses sophisticated topic-model features that do not scale easily) achieves 69.7%. It is encouraging to note that our approach, using only a subset of the features, performs competitively with related work. However, due to the small size of the dataset, we require further evaluation before reaching any conclusions.

5.2 Person-X Evaluation

There is a severe lack of labeled corpora for cross-document coreference due to the effort required to evaluate the coreference decisions. Related approaches have used automated *Person-X* evaluation (Gooi and Allan, 2004), in which unique person-name strings are treated as the true entity labels for the mentions. Every mention string is replaced with an "X" for the coreference system. We use this evaluation methodology on 25k person-name mentions from the New York Times corpus (Sandhaus, 2008) each with one of 50 unique strings. As before, we set the bias b to achieve the same number of entities. We use 1 million samples in each round of inference, followed by random redistribution in the flat model, and super-entities in the hierarchical model. Results are averaged over five runs.

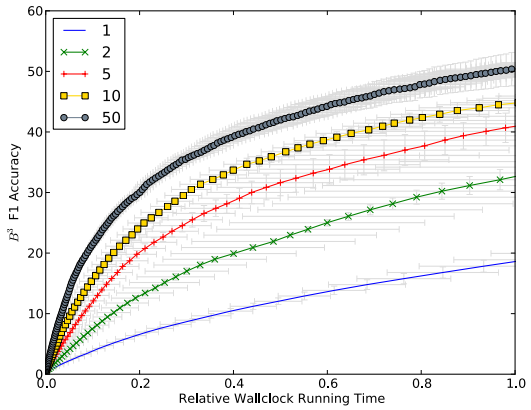

Figure 5: **Person-X Evaluation of Pairwise model**: Performance as number of machines is varied, averaged over 5 runs.

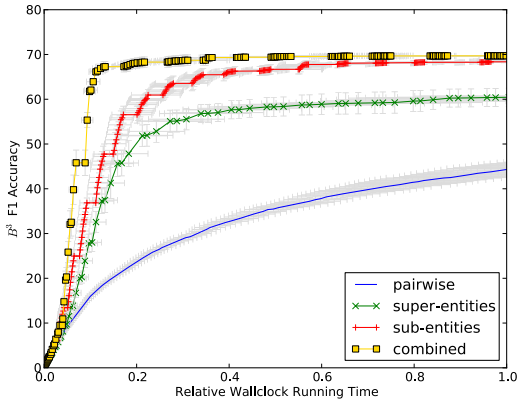

Figure 6: **Person-X Evaluation of Hierarchical Models**: Performance of inference on hierarchical models compared to the pairwise model. Experiments were run using 50 machines.

Number of Entities	43,928
Number of Mentions	1,567,028
Size of Largest Entity	6,096
Average Mentions per Entity	35.7
Variance of Mentions per Entity	5191.7

Table 1: **Wikipedia Link Corpus Statistics**. Size of an entity is the number of mentions of that entity.

Figure 5 shows accuracy compared to relative wallclock running time for distributed inference on the flat, pairwise model. Speed and accuracy improve as additional machines are added, but larger number of machines lead to diminishing returns for this small dataset. Distributed inference on our hierarchical model is evaluated in Figure 6 against inference on the pairwise model from Figure 5. We see that the individual hierarchical models perform much better than the pairwise model; they achieve the same accuracy as the pairwise model in approximately 10% of the time. Moreover, distributed inference on the combined hierarchical model is both faster and more accurate than the individual hierarchical models.

5.3 Wikipedia Link Corpus

To explore the application of the proposed approach to a larger, realistic dataset, we construct a corpus based on the insight that links to Wikipedia that appear on webpages can be treated as mentions, and since the links were added manually by the page author, we use the destination Wikipedia page as the entity the link refers to.

The dataset is created as follows: First, we crawl the web and select hyperlinks on webpages that link to an English Wikipedia page.[2] The anchors of these links form our set of *mentions*, with the surrounding block of clean text (obtained after removing markup, etc.) around each link being its context. We assign the title of the linked Wikipedia page as the *entity* label of that link. Since this set of mentions and labels can be noisy, we use the following filtering steps. All links that have less than 36 words in their block, or whose anchor text has a large string edit distance from the title of the Wikipedia page, are discarded. While this results in cases in which "President" is discarded when linked to the "Barack Obama" Wikipedia page, it was necessary to reduce noise. Further, we also discard links to Wikipedia pages that are concepts (such as "`public_domain`") rather than entities. All entities with less than 6 links to them are also discarded.

Table 1 shows some statistics about our automatically generated data set. We randomly sampled 5% of the entities to create a development set, treating the remaining entities as the test set. Unlike the John Smith and Person-X evaluation, this data set also contains non-person entities such as organizations and locations.

For our models, we augment the factor potentials with mention-string similarity:

[2] e.g. `http://en.wikipedia.org/Hillary_Clinton`

$$\psi_{a/r}(m,n) = \pm\left(\phi_{mn} - b + w\text{STREQ}(m,n)\right)$$

where STREQ is 1 if mentions m and n are string identical (0 otherwise), and w is the weight to this feature.[3] In our experiments we found that setting $w = 0.8$ and $b = 1e-4$ gave the best results on the development set.

Due to the large size of the corpus, existing cross-document coreference approaches could not be applied to this dataset. However, since a majority of related work consists of using clustering after defining a similarity function (Section 6), we provide a baseline evaluation of clustering with *Subsquare* (Bshouty and Long, 2010), a scalable, distributed clustering method. Subsquare takes as input a weighted graph with mentions as nodes and similarity between mentions used as edge weights. Subsquare works by stochastically assigning a vertex to the cluster of one its neighbors if they have significant neighborhood overlap. This algorithm is an efficient form of approximate spectral clustering (Bshouty and Long, 2010), and since it is given the same distances between mentions as our models, we expect it to get similar accuracy. We also generate another baseline clustering by assigning mentions with identical strings to the same entity. This mention-string clustering is also used as the initial configuration of our inference.

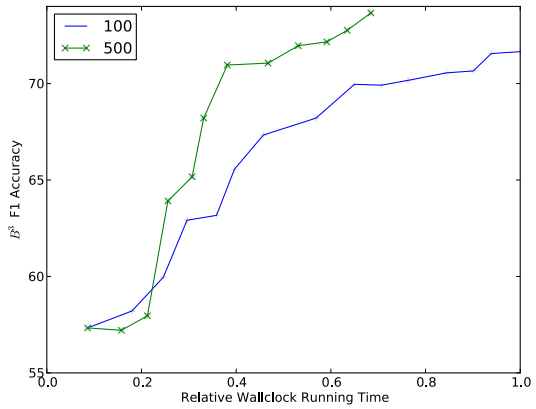

Figure 7: **Wikipedia Link Evaluation**: Performance of inference for different number of machines ($N = 100, 500$). Mention-string match clustering is used as the initial configuration.

[3]Note that we do not use mention-string similarity for John Smith or Person-X as the mention strings are all identical.

Method	Pairwise		B^3 Score	
	P/ R	F1	P/ R	F1
String-Match	30.0 / 66.7	41.5	82.7 / 43.8	57.3
Subsquare	38.2 / 49.1	43.0	87.6 / 51.4	64.8
Our Model	44.2 / 61.4	**51.4**	89.4 / 62.5	**73.7**

Table 2: **F1 Scores on the Wikipedia Link Data**. The results are significant at the 0.0001 level over Subsquare according to the difference of proportions significance test.

Inference is run for 20 rounds of 10 million samples each, distributed over N machines. We use $N = 100, 500$ and the B^3 F1 score results obtained set for each case are shown in Figure 7. It can be seen that $N = 500$ converges to a better solution faster, showing effective use of parallelism. Table 2 compares the results of our approach (at convergence for $N = 500$), the baseline mention-string match and the Subsquare algorithm. Our approach significantly outperforms the competitors.

6 Related Work

Although the cross-document coreference problem is challenging and lacks large labeled datasets, its ubiquitous role as a key component of many knowledge discovery tasks has inspired several efforts.

A number of previous techniques use scoring functions between pairs of contexts, which are then used for clustering. One of the first approaches to cross-document coreference (Bagga and Baldwin, 1998) uses an idf-based cosine-distance scoring function for pairs of contexts, similar to the one we use. Ravin and Kazi (1999) extend this work to be somewhat scalable by comparing pairs of contexts only if the mentions are deemed "ambiguous" using a heuristic. Others have explored multiple methods of context similarity, and concluded that agglomerative clustering provides effective means of inference (Gooi and Allan, 2004). Pedersen et al. (2006) and Purandare and Pedersen (2004) integrate second-order co-occurrence of words into the similarity function. Mann and Yarowsky (2003) use biographical facts from the Web as features for clustering. Niu et al. (2004) incorporate information extraction into the context similarity model, and annotate a small dataset to learn the parameters. A number of other approaches include various forms of

hand-tuned weights, dictionaries, and heuristics to define similarity for name disambiguation (Blume, 2005; Baron and Freedman, 2008; Popescu et al., 2008). These approaches are greedy and differ in the choice of the distance function and the clustering algorithm used. Daumé III and Marcu (2005) propose a generative approach to supervised clustering, and Haghighi and Klein (2010) use entity profiles to assist within-document coreference.

Since many related methods use clustering, there are a number of distributed clustering algorithms that may help scale these approaches. Datta et al. (2006) propose an algorithm for distributed k-means. Chen et al. (2010) describe a parallel spectral clustering algorithm. We use the Subsquare algorithm (Bshouty and Long, 2010) as baseline because it works well in practice. Mocian (2009) presents a survey of distributed clustering algorithms.

Rao et al. (2010) have proposed an online deterministic method that uses a stream of input mentions and assigns them greedily to entities. Although it can resolve mentions from non-trivial sized datasets, the method is restricted to a single machine, which is not scalable to the very large number of mentions that are encountered in practice.

Our representation of the problem as an undirected graphical model, and performing distributed inference on it, provides a combination of advantages not available in any of these approaches. First, most of the methods will not scale to the hundreds of millions of mentions that are present in real-world applications. By utilizing parallelism across machines, our method can run on very large datasets simply by increasing the number of machines used. Second, approaches that use clustering are limited to using pairwise distance functions for which additional supervision and features are difficult to incorporate. In addition to representing features from all of the related work, graphical models can also use more complex entity-wide features (Culotta et al., 2007; Wick et al., 2009a), and parameters can be learned using supervised (Collins, 2002) or semi-supervised techniques (Mann and McCallum, 2008). Finally, the inference for most of the related approaches is *greedy*, and earlier decisions are not revisited. Our technique is based on MCMC inference and simulated annealing, which are able to escape local maxima.

7 Conclusions

Motivated by the problem of solving the coreference problem on billions of mentions from all of the newswire documents from the past few decades, we make the following contributions. First, we introduce distributed version of MCMC-based inference technique that can utilize parallelism to enable scalability. Second, we augment the model with hierarchical variables that facilitate fruitful proposal distributions. As an additional contribution, we use links to Wikipedia pages to obtain a high-quality cross-document corpus. Scalability and accuracy gains of our method are evaluated on multiple datasets.

There are a number of avenues for future work. Although we demonstrate scalability to more than a million mentions, we plan to explore performance on datasets in the billions. We also plan to examine inference on complex coreference models (such as with entity-wide factors). Another possible avenue for future work is that of learning the factors. Since our approach supports parameter estimation, we expect significant accuracy gains with additional features and supervised data. Our work enables cross-document coreference on very large corpora, and we would like to explore the downstream applications that can benefit from it.

Acknowledgments

This work was done when the first author was an intern at Google Research. The authors would like to thank Mark Dredze, Sebastian Riedel, and anonymous reviewers for their valuable feedback. This work was supported in part by the Center for Intelligent Information Retrieval, 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., in part by an award from Google, in part by The Central Intelligence Agency, the National Security Agency and National Science Foundation under NSF grant #IIS-0326249, in part by NSF grant #CNS-0958392, and in part by UPenn NSF medium IIS-0803847. 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. 1998. Entity-based cross-document coreferencing using the vector space model. In *International Conference on Computational Linguistics*, pages 79–85.

A. Baron and M. Freedman. 2008. Who is who and what is what: experiments in cross-document co-reference. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 274–283.

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

Matthias Blume. 2005. Automatic entity disambiguation: Benefits to NER, relation extraction, link analysis, and inference. In *International Conference on Intelligence Analysis (ICIA)*.

Nader H. Bshouty and Philip M. Long. 2010. Finding planted partitions in nearly linear time using arrested spectral clustering. In Johannes Fürnkranz and Thorsten Joachims, editors, *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 135–142, Haifa, Israel, June. Omnipress.

Yuan Changhe, Lu Tsai-Ching, and Druzdzel Marek. 2004. Annealed MAP. In *Uncertainty in Artificial Intelligence (UAI)*, pages 628–635, Arlington , Virginia. AUAI Press.

Wen-Yen Chen, Yangqiu Song, Hongjie Bai, Chih-Jen Lin, and Edward Y. Chang. 2010. Parallel spectral clustering in distributed systems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Michael Collins. 2002. Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithm. In *Annual Meeting of the Association for Computational Linguistics (ACL)*.

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)*.

S. Datta, C. Giannella, and H. Kargupta. 2006. K-Means Clustering over a Large, Dynamic Network. In *SIAM Data Mining Conference (SDM)*.

Hal Daumé III and Daniel Marcu. 2005. A Bayesian model for supervised clustering with the Dirichlet process prior. *Journal of Machine Learning Research (JMLR)*, 6:1551–1577.

Jeffrey Dean and Sanjay Ghemawat. 2004. Mapreduce: Simplified data processing on large clusters. *Symposium on Operating Systems Design & Implementation (OSDI)*.

Sharon Goldwater and Tom Griffiths. 2007. A fully bayesian approach to unsupervised part-of-speech tagging. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 744–751.

Chung Heong Gooi and James Allan. 2004. Cross-document coreference on a large scale corpus. In *North American Chapter of the Association for Computational Linguistics - Human Language Technologies (NAACL HLT)*, pages 9–16.

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. 2009. Simple coreference resolution with rich syntactic and semantic features. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1152–1161.

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.

Gideon S. Mann and Andrew McCallum. 2008. Generalized expectation criteria for semi-supervised learning of conditional random fields. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 870–878.

Gideon S. Mann and David Yarowsky. 2003. Unsupervised personal name disambiguation. In *North American Chapter of the Association for Computational Linguistics - Human Language Technologies (NAACL HLT)*, pages 33–40.

Andre Martins, Noah Smith, Eric Xing, Pedro Aguiar, and Mario Figueiredo. 2010. Turbo parsers: Dependency parsing by approximate variational inference. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 34–44, Cambridge, MA, October. Association for Computational Linguistics.

J. Mayfield, D. Alexander, B. Dorr, J. Eisner, T. Elsayed, T. Finin, C. Fink, M. Freedman, N. Garera, P. McNamee, et al. 2009. Cross-document coreference resolution: A key technology for learning by reading. In *AAAI Spring Symposium on Learning by Reading and Learning to Read*.

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

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

Horatiu Mocian. 2009. *Survey of Distributed Clustering Techniques*. Ph.D. thesis, Imperial College of London.

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

Cheng Niu, Wei Li, and Rohini K. Srihari. 2004. Weakly supervised learning for cross-document person name disambiguation supported by information extraction. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, page 597.

Ted Pedersen, Anagha Kulkarni, Roxana Angheluta, Zornitsa Kozareva, and Thamar Solorio. 2006. An unsupervised language independent method of name discrimination using second order co-occurrence features. In *International Conference on Intelligent Text Processing and Computational Linguistics (CICLing)*, pages 208–222.

Hoifung Poon, Pedro Domingos, and Marc Sumner. 2008. A general method for reducing the complexity of relational inference and its application to MCMC. In *AAAI Conference on Artificial Intelligence*.

Octavian Popescu, Christian Girardi, Emanuele Pianta, and Bernardo Magnini. 2008. Improving cross-document coreference. *Journées Internationales d'Analyse statistique des Données Textuelles*, 9:961–969.

A. Purandare and T. Pedersen. 2004. Word sense discrimination by clustering contexts in vector and similarity spaces. In *Conference on Computational Natural Language Learning (CoNLL)*, pages 41–48.

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.

Alexander M Rush, David Sontag, Michael Collins, and Tommi Jaakkola. 2010. On dual decomposition and linear programming relaxations for natural language processing. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1–11, Cambridge, MA, October. Association for Computational Linguistics.

Evan Sandhaus. 2008. The New York Times annotated corpus. *Linguistic Data Consortium*.

Sameer Singh, Amarnag Subramanya, Fernando Pereira, and Andrew McCallum. 2010. Distributed map inference for undirected graphical models. In *Neural Information Processing Systems (NIPS), Workshop on Learning on Cores, Clusters and Clouds*.

Ben Wellner, Andrew McCallum, Fuchun Peng, and Michael Hay. 2004. An integrated, conditional model of information extraction and coreference with application to citation matching. In *Uncertainty in Artificial Intelligence (UAI)*, pages 593–601.

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

Michael Wick, Khashayar Rohanimanesh, Aron Culotta, and Andrew McCallum. 2009b. Samplerank: Learning preferences from atomic gradients. In *Neural Information Processing Systems (NIPS), Workshop on Advances in Ranking*.