Robust Entity Clustering via Phylogenetic Inference

Nicholas Andrews and Jason Eisner and Mark Dredze

Department of Computer Science and Human Language Technology Center of Excellence
Johns Hopkins University
3400 N. Charles St., Baltimore, MD 21218 USA
{noa,eisner,mdredze}@jhu.edu

Abstract

Entity clustering must determine when two named-entity mentions refer to the same entity. Typical approaches use a pipeline architecture that clusters the mentions using fixed or learned measures of name and context similarity. In this paper, we propose a model for cross-document coreference resolution that achieves robustness by learning similarity from unlabeled data. The generative process assumes that each entity mention arises from copying and optionally mutating an earlier name from a similar context. Clustering the mentions into entities depends on recovering this copying tree jointly with estimating models of the mutation process and parent selection process. We present a block Gibbs sampler for posterior inference and an empirical evaluation on several datasets.

1 Introduction

Variation poses a serious challenge for determining who or what a name refers to. For instance, Wikipedia contains more than 100 variations of the name Barack Obama as redirects to the U.S. President article, including:

President Obama Barack H. Obama, Jr.
Barak Obamba Barry Soetoro

To relate different names, one solution is to use specifically tailored measures of name similarity such as Jaro-Winkler similarity (Winkler, 1999; Cohen et al., 2003). This approach is brittle, however, and fails to adapt to the test data. Another option is to train a model like stochastic edit distance from known pairs of similar names (Ristad and Yianilos, 1998; Green et al., 2012), but this requires supervised data in the test domain.

Even the best model of name similarity is not enough by itself, since two names that are similar—even identical—do not necessarily corefer. Document context is needed to determine whether they may be talking about two different people.

In this paper, we propose a method for jointly (1) learning similarity between names and (2) clustering name mentions into entities, the two major components of cross-document coreference resolution systems (Baron and Freedman, 2008; Finin et al., 2009; Rao et al., 2010; Singh et al., 2011; Lee et al., 2012; Green et al., 2012). Our model is an evolutionary generative process based on the name variation model of Andrews et al. (2012), which stipulates that names are often copied from previously generated names, perhaps with mutation (spelling edits). This can deduce that rather than being names for different entities, Barak Obamba and Barock obama more likely arose from the frequent name Barack Obama as a common ancestor, which accounts for most of their letters. This can also relate seemingly dissimilar names via multiple steps in the generative process:

Taylor Swift → T-Swift → T-Swizzle

Our model learns without supervision that these all refer to the same entity. Such creative spellings are especially common on Twitter and other social media; we give more examples of coreferents learned by our model in Section 8.4.

Our primary contributions are improvements on Andrews et al. (2012) for the entity clustering task. Their inference procedure only clustered types (distinct names) rather than tokens (mentions in context), and relied on expensive matrix inversions for learning. Our novel approach features:

§4.1 A topical model of which entities from previously written text an author tends to mention from previously written text.

§4.2 A name mutation model that is sensitive to features of the input and output characters and takes a reader’s comprehension into account.

§5 A scalable Markov chain Monte Carlo sampler used in training and inference.
A minimum Bayes risk decoding procedure to pick an output clustering. The procedure is applicable to any model capable of producing a posterior over coreference decisions.

We evaluate our approach by comparing to several baselines on datasets from three different genres: Twitter, newswire, and blogs.

2 Overview and Related Work

Cross-document coreference resolution (CDCR) was first introduced by Bagga and Baldwin (1998b). Most approaches since then are based on the intuitions that coreferent names tend to have “similar” spellings and tend to appear in “similar” contexts. The distinguishing feature of our system is that both notions of similarity are learned together without supervision.

We adopt a “phylogenetic” generative model of coreference. The basic insight is that coreference is created when an author thinks of an entity that was mentioned earlier in a similar context, and mentions it again in a similar way. The author may alter the name mention string when copying it, but both names refer to the same entity. Either name may later be copied further, leading to an evolutionary tree of mentions—a phylogeny. Phylogenetic models are new to information extraction. In computational historical linguistics, Bouchard-Côté et al. (2013) have also modeled the mutation of strings along the edges of a phylogeny; but for them the phylogeny is observed and most mentions are not, while we observe the mentions only.

To apply our model to the CDCR task, we observe that the probability that two name mentions are coreferent is the probability that they arose from a common ancestor in the phylogeny. So we design a Monte Carlo sampler to reconstruct likely phylogenies. A phylogeny must explain every observed mention and most mentions are not, while we observe the mentions only.

To apply our model to the CDCR task, we observe that the probability that two name mentions are coreferent is the probability that they arose from a common ancestor in the phylogeny. So we design a Monte Carlo sampler to reconstruct likely phylogenies. A phylogeny must explain every observed name. While our model is capable of generating each name independently, a phylogeny will generally achieve higher probability if it explains similar names as being similar by mutation (rather than by coincidence). Thus, our sampled phylogenies tend to make similar names coreferent—especially long or unusual names that would be expensive to generate repeatedly, and especially in contexts that are topically similar and therefore have a higher prior probability of coreference.

For learning, we iteratively adjust our model’s parameters to better explain our samples. That is, we do unsupervised training via Monte Carlo EM.

What is learned? An important component of a CDCR system is its model of name similarity (Winkler, 1999; Porter and Winkler, 1997), which is often fixed up front. This role is played in our system by the name mutation model, which we take to be a variant of stochastic edit distance (Ristad and Yianilos, 1996). Rather than fixing its parameters before we begin CDCR, we learn them (without supervision) as part of CDCR, by training from samples of reconstructed phylogenies.

Name similarity is also an important component of within-document coreference resolution, and efforts in that area bear resemblance to our approach. Haghighi and Klein (2010) describe an “entity-centered” model where a distance-dependent Chinese restaurant process is used to pick previous coreferent mentions within a document. Similarly, Durrett and Klein (2013) learn a mention similarity model based on labeled data. Our cross-document setting has no observed mention ordering and no observed entities: we must sum over all possibilities, a challenging inference problem.

The second major component of CDCR is context-based disambiguation of similar or identical names that refer to the same entity. Like Kozareva and Ravi (2011) and Green et al. (2012) we use topics as the contexts, but learn mention topics jointly with other model parameters.

3 Generative Model of Coreference

Let \( \mathbf{x} = (x_1, \ldots, x_N) \) denote an ordered sequence of distinct named-entity mentions in documents \( \mathbf{d} = (d_1, \ldots, d_D) \). We assume that each document has a (single) known language, and that its mentions and their types have been identified by a named-entity recognizer. We use the object-oriented notation \( x.v \) for attribute \( v \) of mention \( x \).

Our model generates an ordered sequence \( \mathbf{x} \) although we do not observe its order. Thus each mention \( x \) has latent position \( x.i \) (e.g., \( x_{729}.i = 729 \)). The entire corpus, including these entities, is generated according to standard topic model assumptions; we first generate a topic distribution for a document, then sample topics and words for the document (Blei et al., 2003). However, any topic may generate an entity type, e.g. PERSON, which is then replaced by a specific name: when PERSON is generated, the model chooses a previous mention of any person and copies it, perhaps mutating its name.\(^1\)

\(^1\)We make the closed-world assumption that the author is
a name for a new person, though the name itself may not be new.

If all previous mentions were equally likely, this would be a Chinese Restaurant Process (CRP) in which frequently mentioned entities are more likely to be mentioned again (“the rich get richer”). We refine that idea by saying that the current topic, language, and document influence the choice of which previous mention to copy, similar to the distance-dependent CRP (Blei and Frazier, 2011). This will help distinguish multiple John Smith entities if they tend to appear in different contexts.

Formally, each mention $x$ is derived from a parent mention $x.p$ where $x.p.i < x.i$ (the parent came first), $x.e = x.p.e$ (same entity) and $x.n$ is a copy or mutation of $x.p.n$. In the special case where $x$ is the first mention of $x.e$, $x.p$ is the special symbol $\diamond$, $x.e$ is a newly allocated entity of some appropriate type, and the name $x.n$ is generated from scratch.

Our goal is to reconstruct mappings $p$, $i$, $z$ that specify the latent properties of the mentions $x$. The mapping $p : x \mapsto x.p$ forms a phylogenetic tree on the mentions, with root $\diamond$. Each entity corresponds to a subtree that is rooted at some child of $\diamond$. The mapping $i : x \mapsto x.i$ gives an ordering consistent with that in the sense that $(\forall x) x.p.i < x.i$. Finally, the mapping $z : x \mapsto x.z$ specifies, for each mention, the topic that generated it. While $i$ and $z$ are not necessary for creating coref clusters, they are needed to produce $p$.

4 Detailed generative story

Given a few constants that are referenced in the main text, we assume that the corpus $d$ was generated as follows.

First, for each topic $z = 1, \ldots, K$ and each language $\ell$, choose a multinomial $\beta_{z,\ell}$ over the word vocabulary, from a symmetric Dirichlet prior with concentration parameter $\eta$. Then set $m = 0$ (entity count), $i = 0$ (mention count), and for each document index $d = 1, \ldots, D$:

1. Choose the document’s length $L$ and language $\ell$. (The distributions used to choose these are unimportant because these variables are always observed.)
2. Choose its topic distribution $\psi_d$ from an asymmetric Dirichlet prior with parameters $m$ (Wallach et al., 2009).
3. For each token position $k = 1, \ldots, L$:
   a. Choose a topic $z_{dk} \sim \psi_d$.
   b. Choose a word conditioned on the topic and language, $w_{dk} \sim \beta_{z_{dk},\ell}$.
   c. If $w_{dk}$ is a named entity type (PERSON, PLACE, ORG, . . .) rather than an ordinary word, then increment $i$ and:
      i. create a new mention $x$ with $x.e.t = w_{dk}$ $x.d = d$ $x.\ell = \ell$
         $x.i = i$ $x.z = z_{dk}$ $x.k = k$
      ii. Choose the parent $x.p$ from a distribution conditioned on the attributes just set (see §4.1).
   d. If $x.p = \diamond$, increment $m$ and set $x.e = \phantom{new entity e.m}$. Else set $x.e = x.p.e$.
   e. Choose $x.n$ from a distribution conditioned on $x.p.n$ and $x.\ell$ (see §4.2).

Notice that the tokens $w_{dk}$ in document $d$ are exchangeable: by collapsing out $\psi_d$, we can regard them as having been generated from a CRP. Thus, for fixed values of the non-mention tokens and their topics, the probability of generating the mention sequence $x$ is proportional to the product of the probabilities of the choices in step 3 at the positions $dk$ where mentions were generated. These choices generate a topic $x.z$ (from the CRP for document $d$), a type $x.e.t$ (from $\beta_{z,x.t}$), a parent mention (from the distribution over previous mentions), and a name string (conditioned on the parent’s name if any). §5 uses this fact to construct an MCMC sampler for the latent parts of $x$.

4.1 Sub-model for parent selection

To select a parent for a mention $x$ of type $x.e.t$, a simple model (as mentioned above) would be a CRP: each previous mention of the same type is selected with probability proportional to 1, and $\diamond$ is
selected with probability proportional to $\alpha_t > 0$. A larger choice of $\alpha_t$ results in smaller entity clusters, because it prefers to create new entities of type $t$ rather than copying old ones.

We modify this story by re-weighting $\diamond$ and previous mentions according to their relative suitability as the parent of $x$:

$$
\Pr(\phi|x.p) = \frac{\exp(\phi \cdot f(x.p, x))}{Z(x)}
$$

(1)

where $x.p$ ranges over $\diamond$ and all previous mentions of the same type as $x$, that is, mentions $p$ such that $p.i < x.i$ and $p.e.t = x.e.t$. The normalizing constant $Z(x) \equiv \sum_p \exp(\phi \cdot f(x.p, x))$ is chosen so that the probabilities sum to 1.

This is a conditional log-linear model parameterized by $\phi$, where $\phi_k \sim N(0, \sigma_k^2)$. The features $f$ are extracted from the attributes of $x$ and $x.p$. Our most important feature tests whether $x.p.z = x.z$. This binary feature has a high weight if authors mainly choose mentions from the same topic. To model which (other) topics tend to be selected, we also have a binary feature for each parent topic $x.p.z$ and each topic pair $(x.p.z, x.z)$.

\section{Sub-model for name mutation}

Let $x$ denote a mention with parent $p = x.p$. As in Andrews et al. (2012), its name $x.n$ is a stochastic transduction of its parent’s name $p.n$. That is,

$$
\Pr(\theta|x.n | p.n) = \text{Pr}(\phi|x.p) \Pr(g|x.p, x) \Pr(\theta|x.n, g, x)
$$

(2)

is given by the probability that applying a random sequence of edits to the characters of $p.n$ would yield $x.n$. The contextual probabilities of different edits depend on learned parameters $\theta$.

(2) is the total probability of all edit sequences that derive $x.n$ from $p.n$. It can be computed in time $O(|x.n| \cdot |p.n|)$ by dynamic programming.

The probability of a single edit sequence, which corresponds to a monotonic alignment of $x.n$ to $p.n$, is a product of individual edit probabilities of the form $\Pr(\theta|g|x.p, x)$, which is conditioned on the next input character $\hat{a}$. The edit $(\hat{a})$ replaces input $a \in \epsilon, \hat{a}$ with output $b \in \epsilon \cup \Sigma$ (where $\epsilon$ is the empty string and $\Sigma$ is the alphabet of language $x.l$). Insertions and deletions are the cases where respectively $a = \epsilon$ or $b = \epsilon$—we do not allow both at once. All other edits are substitutions. When $\hat{a}$ is the special end-of-string symbol $\#$, the only allowed edits are the insertion $(\hat{a})$ and the substitution $(\hat{a} b)$. We define the edit probability using a locally normalized log-linear model:

$$
\Pr(\theta|g|x.p, x) = \frac{\exp(\theta \cdot f(\hat{a}, a, b))}{\sum_{a', b'} \exp(\theta \cdot f(\hat{a}, a', b'))}
$$

(3)

We use a small set of simple feature functions $f$, which consider conjunctions of the attributes of the characters $\hat{a}$ and $b$: character, character class (letter, digit, etc.), and case (upper vs. lower).

More generally, the probability (2) may also be conditioned on other variables such as on the languages $p.l$ and $x.l$—this leaves room for a transliteration model when $x.l \neq p.l$—and on the entity type $x.t$. The features in (3) may then depend on these variables as well.

Notice that we use a locally normalized probability for each edit. This enables faster and simpler training than the similar model of Dreyer et al. (2008), which uses a globally normalized probability for the whole edit sequence.

When $p = \diamond$, we are generating a new name $x.n$. We use the same model, taking $\diamond. n$ to be the empty string (but with $\#$ rather than $\#$ as the end-of-string symbol). This yields a feature-based unigram language model (whose character probabilities may differ from usual insertion probabilities because they see $\#$ as the lookahead character).

\textbf{Pragmatics.} We can optionally make the model more sophisticated. Authors tend to avoid names $x.n$ that readers would misinterpret (given the previously generated names). The edit model thinks that $\Pr(\theta|g|\text{CIA} | \diamond)$ is relatively high (because CIA is a short string) and so is $\Pr(\theta|g|\text{Chuck's Ice Art})$. But in fact, if CIA has already been frequently used to refer to the Central Intelligence Agency, then an author is unlikely to use it for a different entity.

To model this pragmatic effect, we multiply our definition of $\Pr(\theta|x.n | p.n)$ by an extra factor $\Pr(x.e | x)^\gamma$, where $\gamma \geq 0$ is the effect strength.\footnote{Currently we omit the step of renormalizing this deficient model. Our training procedure also ignores the extra factor. Here $\Pr(x.e | x)$ is the probability that a reader correctly identifies the entity $x.e$. We take this to be the probability that a reader who knows our sub-models would guess some parent...}
having the correct entity (or ♦ if x is a first mention): \( \sum_{p',\xi' = x} w(p', x) / \sum_{p'} w(p', x) \). Here \( p' \) ranges over mentions (including ♦) that precede x in the ordering \( i \), and \( w(p', x) \)—defined later in sec. 5.3—is proportional to the posterior probability that \( x.p = p' \), given name \( x.n \) and topic \( x.z \).6

5 Inference by Block Gibbs Sampling

We use a block Gibbs sampler, which from an initial state \((p_0, i_0, z_0)\) repeats these steps:

1. Sample the ordering \( i \) from its conditional distribution given all other variables.
2. Sample the topic vector \( z \) likewise.
3. Sample the phylogeny \( p \) likewise.
4. Output the current sample \( s_t = (p, i, z) \).

It is difficult to draw exact samples at steps 1 and 2. Thus, we sample \( i \) or \( z \) from a simpler proposal distribution, but correct the discrepancy using the Independent Metropolis-Hastings (IMH) strategy: with an appropriate probability, reject the proposed new value and instead use another copy of the current value (Tierney, 1994).

5.1 Resampling the ordering \( i \)

We resample the ordering \( i \) of the mentions \( x \), conditioned on the other variables. The current phylogeny \( p \) already defines a partial order on \( x \), since each parent must precede its children. For instance, phylogeny (a) below requires \( ♦ ≺ x \) and \( ♦ ≺ y \). This partial order is compatible with 2 total orderings, \( ♦ ≺ x ≺ y \) and \( ♦ ≺ y ≺ x \). By contrast, phylogeny (b) requires the total ordering \( ♦ ≺ x ≺ y \).

\[
\begin{align*}
\text{(a)} & \quad x \quad y \\
\text{(b)} & \quad x \quad ♦ \quad y
\end{align*}
\]

We first sample an ordering \( i_0 \) (the ordering of mentions with parent ♦, i.e. all mentions) uniformly at random from the set of orderings compatible with the current \( p \). (We provide details about this procedure in Appendix A.)7 However, such orderings are not in fact equiprobable given the other variables—some orderings better explain why that phylogeny was chosen in the first place, according to our competitive parent selection model (§4.1).

To correct for this bias using IMH, we accept the proposed ordering \( i_0 \) with probability

\[
a = \min \left( 1, \frac{\Pr(p, i, z, x | \theta, \phi) \Psi(x.p, z, x | \theta, \phi)}{\Pr(p, i_0, z, x | \theta, \phi) \Psi(x.p, z, x | \theta, \phi)} \right)
\]

(4)

where \( i \) is the current ordering. Otherwise we reject \( i_0 \) and reuse \( i \) for the new sample.

5.2 Resampling the topics \( z \)

Each context word and each named entity is associated with a latent topic. The topics of context words are assumed exchangeable, and so we resample them using Gibbs sampling (Griffiths and Steyvers, 2004).

Unfortunately, this is prohibitively expensive for the (non-exchangeable) topics of the named mentions \( x \). A Gibbs sampler would have to choose a new value for \( x.z \) with probability proportional to the resulting joint probability of the full sample. This probability is expensive to evaluate because changing \( x.z \) will change the probability of many edges in the current phylogeny \( p \). (Equation (1) puts \( x \) in competition with other parents, so every mention \( y \) that follows \( x \) must recompute how happy it is with its current parent \( y.p \))

Rather than resampling one topic at a time, we resample \( z \) as a block. We use a proposal distribution for which block sampling is efficient, and use IMH to correct the error in this proposal distribution.

Our proposal distribution is an undirected graphical model whose random variables are the topics \( z \) and whose graph structure is given by the current phylogeny \( p \):

\[
Q(z) \propto \prod_{x \neq ♦} \Psi_z(x.z) \Psi_{x,p,x}(x.p.z, x.z)
\]

(5)

\( Q(z) \) is an approximation to the posterior distribution over \( z \). As detailed below, a proposal can be sampled from \( Q(z) \) in time \( O(|z|K^2) \) where \( K \) is the number of topics, because the only interactions among topics are along the edges of the tree \( p \). The unary factor \( \Psi_z \) gives a weight for each possible value of \( x.z \), and the binary factor \( \Psi_{x,p,x} \) gives a weight for each possible value of the pair \( (x.p.z, x.z) \).

The \( \Psi_z(x.z) \) factors in (5) approximate the topic model’s prior distribution over \( z \). \( \Psi_{x,p,x} \) is proportional to the probability that a Gibbs sampling step for an ordinary topic model would choose this value of \( x.z \). This depends on whether—in the
current sample—x.z is currently common in x’s document and x.t is commonly generated by x.z. It ignores the fact that we will also be resampling the topics of the other mentions.

The \( \Psi_{x,p,x} \) factors in (5) approximate \( \Pr(p \mid z, i) \) (up to a constant factor), where \( p \) is the current phylogeny. Specifically, \( \Psi_{x,p,x} \) approximates the probability of a single edge. It ought to be given by (1), but we use only the numerator of (1), which avoids modeling the competition among parents.

We sample from \( Q \) using standard methods, similar to sampling from a linear-chain CRF by running the backward algorithm followed by forward sampling. Specifically, we run the sum-product algorithm from the leaves up to the root \( \hat{\phi} \). Then we sample from the root down to the leaves, computing the following for each topic at node \( x \) using the procedure described in §5.1.

\[
\beta_x(z) = \sum_{y \in \text{children}(x)} \Psi_{x,y}(z, z') \cdot \beta_y(z')
\]

Then we sample from the root down to the leaves, first sampling \( \hat{\phi}, z \) from \( \beta_{\hat{\phi}} \), then at each \( x \neq \hat{\phi} \) sampling the topic \( x.z \) to be \( z \) with probability proportional to \( \Psi_{x,p,x}(x.p, z, z) \cdot \beta_x(z) \).

Again we use IMH to correct for the bias in \( Q \): we accept the resulting proposal \( \hat{z} \) with probability

\[
\min \left( 1, \frac{\Pr(p, i, \hat{z}, x \mid \theta, \phi) \cdot Q(z)}{\Pr(p, i, z, x \mid \theta, \phi) \cdot Q(\hat{z})} \right)
\]

While \( \Pr(p, i, \hat{z}, x \mid \theta, \phi) \) might seem slow to compute because it contains many factors (1) with different denominators \( Z(x) \), one can share work by visiting the mentions \( x \) in their order \( i \). Most summands in \( Z(x) \) were already included in \( Z(x') \), where \( x' \) is the latest previous mention having the same attributes as \( x \) (e.g., same topic).

### 5.3 Resampling the phylogeny \( p \)

It is easy to resample the phylogeny. For each \( x \), we must choose a parent \( x.p \) from among the possible parents \( p \) (having \( p.i < x.i \) and \( p.e.t = x.e.t \)). Since the ordering \( i \) prevents cycles, the resulting phylogeny \( p \) is indeed a tree.

Given the topics \( z \), the ordering \( i \), and the observed names, we choose an \( x.p \) value according to its posterior probability. This is proportional to \( w(x.p, x) \approx \Pr(x.p \mid x) \cdot \Pr(x.n \mid x.p.n) \), independent of any other mention’s choice of parent. The two factors here are given by (1) and (2) respectively. As in the previous section, the denominators \( Z(x) \) in the \( \Pr(x.p \mid x) \) factors can be computed efficiently with shared work.

With the pragmatic model (section 4.2), the parent choices are no longer independent; then the samples of \( p \) should be corrected by IMH as usual.

### 5.4 Initializing the sampler

The initial sampler state \( (z_0, p_0, i_0) \) is obtained as follows. (1) We fix topics \( z_0 \) via collapsed Gibbs sampling (Griffiths and Steyvers, 2004). The sampler is run for 1000 iterations, and the final sampler state is taken to be \( z_0 \). This process treats all topics as exchangeable, including those associated with named entities.(2) Given the topic assignment \( z_0 \), initialize \( p_0 \) to the phylogeny rooted at \( \hat{\phi} \) that maximizes \( \sum_x \log w(x.p, x) \). This is a maximum rooted directed spanning tree problem that can be solved in time \( O(n^2) \) (Tarjan, 1977). The weight \( w(x.p, x) \) is defined as in section 5.3—except that since we do not yet have an ordering \( i \), we do not restrict the possible values of \( x.p \) to mentions \( p \) with \( p.i < x.p.i \). (3) Given \( p_0 \), sample an ordering \( i_0 \) using the procedure described in §5.1.

### 6 Parameter Estimation

Evaluating the likelihood and its partial derivatives with respect to the parameters of the model requires marginalizing over our latent variables. As this marginalization is intractable, we resort to Monte Carlo EM procedure (Levine and Casella, 2001) which iterates the following two steps:

**E-step**: Collect samples by MCMC simulation as in §5, given current model parameters \( \theta \) and \( \phi \).

**M-step**: Improve \( \theta \) and \( \phi \) to increase

\[
L \triangleq \frac{1}{S} \sum_{s=1}^{S} \log \Pr_{\theta, \phi}(x, p_s, i_s, z_s)
\]

It is not necessary to locally maximize \( L \) at each M-step, merely to improve it if it is not already at a local maximum (Dempster et al., 1977). We improve it by a single update: at the \( t \)th M-step, we update our parameters to \( \Phi_t = (\theta_t, \phi_t) \)

\[
\Phi_t = \Phi_{t-1} + \varepsilon \Sigma_t \nabla_{\theta, \phi} L(x, \Phi_{t-1})
\]

where \( \varepsilon \) is a fixed scaling term and \( \Sigma_t \) is an adaptive learning rate given by AdaGrad (Duchi et al., 2011).

We now describe how to compute the gradient

\[
\nabla_{\theta, \phi} L
\]

The gradient with respect to the parent se-

---

\[\text{We actually do MAP-EM, which augments (7) by adding the log-likelihoods of } \theta \text{ and } \phi \text{ under a Gaussian prior.}\]
lection parameters $\phi$ is

$$\sum 1 \left( f(p, x) - \sum p' \Pr_{\phi}(p' | x) f(p', x) \right)$$

The outer summation ranges over all edges in the $S$ samples. The other variables in (9) are associated with the edge being summed over. That edge explains a mention $x$ as some parent $p$ in the context of a particular sample $(p, i_s, z_s)$. The possible parents $p'$ range over $\Diamond$ and the mentions that precede $x$ according to the ordering $i_s$, while the features $f$ and distribution $\Pr_{\phi}$ depend on the topics $z_s$.

As for the mutation parameters, let $c_{p,x}$ be the fraction of samples in which $p$ is the parent of $x$. This is the expected number of times that the string $p.n$ mutated into $x.n$. Given this weighted set of string pairs, let $c_{a,a,b}$ be the expected number of times that edit $(a,b)$ was chosen in context $a$: this can be computed using dynamic programming to marginalize over the latent edit sequence that maps $p.n$ to $x.n$, for each $(p, x)$. The gradient of $L$ with respect to $\theta$ is

$$\sum a, a, b c_{a, a, b} \left( f(\hat{a}, a, b) \right)\Pr_{\theta}(a', b' | \hat{a}) f(\hat{a}, a', b')$$

(10)

7 Consensus Clustering

From a single phylogeny $p$, we deterministically obtain a clustering $e$ by removing the root $\Diamond$. Each of the resulting connected components corresponds to a cluster of mentions. Our model gives a distribution over phylogenies $p$ (given observations $x$ and learned parameters $\Phi$)—and thus gives a posterior distribution over clusterings $e$, which can be used to answer various queries.

A traditional query is to request a single clustering $e$. We prefer the clustering $e^*$ that minimizes Bayes risk (MBR) (Bickel and Doksum, 1977):

$$e^* = \arg\min_{e'} \sum_e L(e', e) \Pr(e | x, \theta, \phi)$$

(11)

This minimizes our expected loss, where $L(e', e)$ denotes the loss associated with picking $e'$ when the true clustering is $e$. In practice, we again estimate the expectation by sampling $e$ values.

The Rand index (Rand, 1971)—unlike our actual evaluation measure—is an efficient choice of loss function $L$ for use with (11):

$$R(e', e) = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}} = \frac{\text{TP} + \text{TN}}{\binom{N}{2}}$$

where the true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) use the clustering $e$ to evaluate how well $e'$ classifies the $\binom{N}{2}$ mention pairs as coreferent or not. More similar clusterings achieve larger $R$, with $R(e', e) = 1$ iff $e' = e$. In all cases, $0 \leq R(e', e) = R(e, e') \leq 1$.

The MBR decision rule for the (negated) Rand index is easily seen to be equivalent to

$$e^* = \arg\max_{e'} \mathbb{E}[\text{TP}] + \mathbb{E}[\text{TN}]$$

$$= \arg\max_{e'} \sum_{i,j: x_i \sim x_j} s_{ij} + \sum_{i,j: x_i \neq x_j} (1 - s_{ij})$$

where $\sim$ denotes coreference according to $e'$. As explained above, the $s_{ij}$ are coreference probabilities $s_{ij}$ that can be estimated from a sample of clusterings $e$.

This objective corresponds to min-max graph cut (Ding et al., 2001), an NP-hard problem with an approximate solution (Nie et al., 2010).

8 Experiments

In this section, we describe experiments on three different datasets. Our main results are described first: Twitter features many instances of name variation that we would like our model to be able to learn. We also report the performance of different ablations of our full approach, in order to see which consistently helped across the different splits. We report additional experiments on the ACE 2008 corpus, and on a political blog corpus, to demonstrate that our approach is applicable in different settings.

For Twitter and ACE 2008, we report the standard $B^3$ metric (Bagga and Baldwin, 1998a). For the political blog dataset, the reference does not consist of entity annotations, and so we follow the evaluation procedure of Yogatama et al. (2012).

8.1 Twitter

Data. We use a novel corpus of Twitter posts discussing the 2013 Grammy Award ceremony. This is a challenging corpus, featuring many instances

---

9In our experiments, we run the clustering algorithm five times, initialized from samples chosen at random from the last 10% of the sampler run, and keep the clustering that achieved highest expected Rand score.
of name variation. The dataset consists of five splits (by entity), the smallest of which is 604 mentions and the largest is 1374. We reserve the largest split for development purposes, and report our results on the remaining four. Appendix B provides more detail about the dataset.

**Baselines.** We use the discriminative entity clustering algorithm of Green et al. (2012) as our baseline; their approach was found to outperform another generative model which produced a flat clustering of mentions via a Dirichlet process mixture model. Their method uses Jaro-Winkler string similarity to match names, then clusters mentions with matching names (for disambiguation) by comparing their unigram context distributions using the Jenson-Shannon metric. We also compare to the EXACT-MATCH baseline, which assigns all strings with the same name to the same entity.

**Procedure.** We run four test experiments in which one split is used to pick model hyperparameters and the remaining three are used for test. For the discriminative baseline, we tune the string match threshold, context threshold, and the weight of the context model prior (all via grid search). For our model, we tune only the fixed weight of the root feature, which determines the precision/recall trade-off (larger values of this feature result in more attachments to $\diamond$ and hence more entities). We leave other hyperparameters fixed: 16 latent topics, and Gaussian priors $\mathcal{N}(0, 1)$ on all log-linear parameters. For PHYLO, the entity clustering is the result of (1) training the model using EM, (2) sampling from the posterior to obtain a distribution over clusterings, and (3) finding a consensus clustering. We use 20 iterations of EM with 100 samples per E-step for training, and use 1000 samples after training to estimate the posterior. We report results using three variations of our model: PHYLO does not consider mention context (all mentions effectively have the same topic) and determines mention entities from a single sample of $p$ (the last); PHYLO+TOPIC adds context ($\S$5.2); PHYLO+TOPIC+MBR uses the full posterior and consensus clustering to pick the output clustering ($\S$7). Our results are shown in Table 1.10

<table>
<thead>
<tr>
<th></th>
<th>Test $B^3$</th>
<th>P</th>
<th>R</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXACT-MATCH</td>
<td></td>
<td>99.6</td>
<td>53.7</td>
<td>69.8</td>
</tr>
<tr>
<td>Green et al. (2012)</td>
<td></td>
<td>92.1</td>
<td>69.8</td>
<td>79.3</td>
</tr>
<tr>
<td>PHYLO</td>
<td></td>
<td>85.3</td>
<td>91.4</td>
<td>88.7</td>
</tr>
<tr>
<td>PHYLO+TOPIC</td>
<td></td>
<td>92.8</td>
<td>90.8</td>
<td>91.8</td>
</tr>
<tr>
<td>PHYLO+TOPIC+MBR</td>
<td></td>
<td>92.9</td>
<td>90.9</td>
<td>91.9</td>
</tr>
</tbody>
</table>

Table 1: Results for the Twitter dataset. Higher $B^3$ scores are better. Note that each number is averaged over four different test splits. In three out of four experiments, PHYLO+TOPIC+MBR achieved the highest F1 score; in one case PHYLO+TOPIC won by a small margin.

<table>
<thead>
<tr>
<th></th>
<th>Test $B^3$</th>
<th>P</th>
<th>R</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>PER</td>
<td>EXACT-MATCH</td>
<td>98.0</td>
<td>81.2</td>
<td>88.8</td>
</tr>
<tr>
<td>Green et al. (2012)</td>
<td></td>
<td>95.0</td>
<td>88.9</td>
<td>91.9</td>
</tr>
<tr>
<td>PHYLO+TOPIC+MBR</td>
<td></td>
<td>97.2</td>
<td>88.6</td>
<td>92.7</td>
</tr>
<tr>
<td>ORG</td>
<td>EXACT-MATCH</td>
<td>98.2</td>
<td>78.3</td>
<td>87.1</td>
</tr>
<tr>
<td>Green et al. (2012)</td>
<td></td>
<td>92.1</td>
<td>88.5</td>
<td>90.3</td>
</tr>
<tr>
<td>PHYLO+TOPIC+MBR</td>
<td></td>
<td>95.5</td>
<td>80.9</td>
<td>87.6</td>
</tr>
</tbody>
</table>

Table 2: Results for the ACE 2008 newswire dataset.

### 8.2 Newswire

**Data.** We use the ACE 2008 dataset, which is described in detail in Green et al. (2012). It is split into a development portion and a test portion. The baseline system took the first mention from each (gold) within-document coreference chain as the canonical mention, ignoring other mentions in the chain; we follow the same procedure in our experiments.11

**Baselines & Procedure.** We use the same baselines as in §8.1. On development data, modeling pragmatics as in §4.2 gave large improvements for organizations (8 points in F-measure), correcting the tendency to assume that short names like CIA were coincidental homonyms. Hence we allowed $\gamma > 0$ and tuned it on development data.12 Results are in Table 2.

### 8.3 Blogs

**Data.** The CMU political blogs dataset consists of 3000 documents about U.S. politics (Yano et al., 2009). Preprocessed as described in Yogatama et al. (2012), the data consists of 10647 entity mentions.

---

10Our single-threaded implementation took around 15 minutes per fold of the Twitter corpus on a personal laptop with a 2.3 Ghz Intel Core i7 processor (including time required to parse the data files). Typical acceptance rates for ordering and topic proposals ranged from 0.03 to 0.08.

11That is, each within-document coreference chain is mapped to a single mention as a preprocessing step.

12We used only a simplified version of the pragmatic model, approximating $w(y', x)$ as 1 or 0 according to whether $y', n = x, n$. We also omitted the IMH step from section 5.3. The other results we report do not use pragmatics at all, since we found that it gave only a slight improvement on Twitter.
Unlike our other datasets, mentions are not annotated with entities: the reference consists of a table of 126 entities, where each row is the canonical name of one entity.

**Baselines.** We compare to the system results reported in Figure 2 of Yogatama et al. (2012). This includes a baseline hierarchical clustering approach, the “EEA” name canonicalization system of Eisenstein et al. (2011), as well the model proposed by Yogatama et al. (2012). Like the output of our model, the output of their hierarchical clustering baseline is a mention clustering, and therefore must be mapped to a table of canonical entity names to compare to the reference table.

**Procedure & Results** We tune our method as in previous experiments, on the initialization data used by Yogatama et al. (2012) which consists of a subset of 700 documents of the full dataset. The tuned model then produced a mention clustering on the full political blog corpus. As the mapping from clusters to a table is not fully detailed in Yogatama et al. (2012), we used a simple heuristic: the most frequent name in each cluster is taken as the canonical name, augmented by any titles from a predefined list appearing in any other name in the cluster. The resulting table is then evaluated against the reference, as described in Yogatama et al. (2012). We achieved a response score of 0.17 and a reference score of 0.61. Though not state-of-the-art, this result is close to the score of the “EEA” system of Eisenstein et al. (2011), as reported in Figure 2 of Yogatama et al. (2012), which is specifically designed for the task of canonicalization.

**8.4 Discussion**

On the Twitter dataset, we obtained a 12.6-point F1 improvement over the baseline. To understand our model’s behavior, we looked at the sampled phylogenetic trees on development data. One reason our model does well in this noisy domain is that it is able to relate seemingly dissimilar names via successive steps. For instance, our model learned to relate many variations of LL Cool J:

- Cool James
- LLCoJ
- El-EI Cool John
- LL COOL JAMES
- LLCOOLJ

In the sample we inspected, these mentions were also assigned the same topic, further boosting the probability of the configuration.

The ACE dataset, consisting of editorialized newswire, naturally contains less name variation than Twitter data. Nonetheless, we find that the variation that does appear is often properly handled by our model. For instance, we see several instances of variation due to transliteration that were all correctly grouped together, such as Megawati Soekarnoputri and Megawati Sukarnoputri. The pragmatic model was also effective in grouping common acronyms into the same entity.

We found that multiple samples tend to give different phylogenies (so the sampler is mobile), but essentially the same clustering into entities (which is why consensus clustering did not improve much over simply using the last sample). Random restarts of EM might create more variety by choosing different locally optimal parameter settings. It may also be beneficial to explore other sampling techniques (Bouchard-Côté, 2014).

Our method assembles observed names into an evolutionary tree. However, the true tree must include many names that fall outside our small observed corpora, so our model would be a more appropriate fit for a far larger corpus. Larger corpora also offer stronger signals that might enable our Monte Carlo methods to mix faster and detect regularities more accurately.

A common error of our system is to connect mentions that share long substrings, such as different PERSONs who share a last name, or different ORGANIZATIONs that contain University of. A more powerful name mutation than the one we use here would recognize entire words, for example inserting a common title or replacing a first name with its common nickname. Modeling the internal structure of names (Johnson, 2010; Eisenstein et al., 2011; Yogatama et al., 2012) in the mutation model is a promising future direction.

**9 Conclusions**

Our primary contribution consists of new modeling ideas, and associated inference techniques, for the problem of cross-document coreference resolution. We have described how writers systematically plunder (φ) and then systematically modify (θ) the work of past writers. Inference under such models could also play a role in tracking evolving memes and social influence, not merely in establishing strict coreference. Our model also provides an alternative to the distance-dependent CRP.

Our implementation is available for research use at: https://bitbucket.org/noandrews/phyloinf.
References


Alexandre Bouchard-Côté. 2014. Sequential Monte Carlo (SMC) for Bayesian phylogenetics. Bayesian phylogenetics: methods, algorithms, and applications.


