Structured Databases of Named Entities from Bayesian Nonparametrics

Jacob Eisenstein **Tae Yano** **William W. Cohen** **Noah A. Smith** **Eric P. Xing**
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213, USA
{jacobeis,taey,wcohen,nasmith,epxing}@cs.cmu.edu

Abstract

We present a nonparametric Bayesian approach to extract a structured database of entities from text. Neither the number of entities nor the fields that characterize each entity are provided in advance; the only supervision is a set of five prototype examples. Our method jointly accomplishes three tasks: (i) identifying a set of canonical entities, (ii) inferring a schema for the fields that describe each entity, and (iii) matching entities to their references in raw text. Empirical evaluation shows that the approach learns an accurate database of entities and a sensible model of name structure.

1 Introduction

Consider the task of building a set of structured records from a collection of text: for example, extracting the names of people or businesses from blog posts, where each full name decomposes into fields corresponding to *first-name*, *last-name*, *title*, etc. To instruct a person to perform this task, one might begin with a few examples of the records to be obtained; assuming that the mapping from text to records is relatively straightforward, no additional instruction would be necessary. In this paper, we present a method for training information extraction software in the same way: starting from a small table of partially-complete "prototype" records (Table 1), our system learns to add new entries and fields to the table, while simultaneously aligning the records to text.

We assume that the dimensionality of the database is unknown, so that neither the number of entries nor the number of fields is specified in advance. To accommodate this uncertainty, we apply a Bayesian model which is nonparametric along three dimensions: the assignment of text mentions to entities (making popular entries more likely while always allowing new entries); the alignment of individual text tokens to fields (encouraging the re-use of common fields, but permitting the creation of new fields); and the assignment of values to entries in the database itself (encouraging the reuse of values across entries in a given field). By adaptively updating the concentration parameter of stick-breaking distribution controlling the assignment of values to entries in the database, our model can learn domain-specific information about each field: for example, that titles are often repeated, while names are more varied.

John	McCain	Sen.			Mr.
George	Bush		W.		Mr.
Hillary	Clinton			Rodham	Mrs.
Barack	Obama	Sen.			
Sarah	Palin				

Table 1: A set of partially-complete prototype records, which constitutes the only supervision for the system.

Our system's input consists of a very small prototype table and a corpus of text which has been automatically segmented to identify names. Our desired output is a set of structured records in which each field contains a single string — not a distribution over strings, which would be more difficult to interpret. This requirement induces a tight probabilistic coupling between the assignment of text to cells in the table, so special care is required to ob-

Proceedings of EMNLP 2011, Conference on Empirical Methods in Natural Language Processing, pages 2–12,
Edinburgh, Scotland, UK, July 27–31, 2011.

tain efficient inference. Our procedure alternates between two phases. In the first phase, we perform collapsed Gibbs sampling on the assignments of string mentions to rows and columns in the table, while marginalizing the values of the table itself. In the second phase, we apply Metropolis-Hastings to swap the values of columns in the table, while simultaneously relabeling the affected strings in the text.

Our model performs three tasks: it constructs a set of entities from raw text, matches mentions in text with the entities to which they refer, and discovers general categories of tokens that appear in names (such as titles and first names). We are aware of no existing system that performs all three of these tasks jointly. We evaluate on a dataset of political blogs, measuring our system's ability to discover a set of reference entities (recall) while maintaining a compact number of rows and columns (precision). With as few as five partially-complete prototype examples, our approach gives accurate tables that match well against a manually-annotated reference list. Our method outperforms a baseline single-link clustering approach inspired by one of the most successful entries (Elmacioglu et al., 2007) in the SEMEVAL "Web People Search" shared task (Artiles et al., 2007).

2 Task Definition

In this work, we assume that a bag of M *mentions* in text have been identified. The mth mention $\boldsymbol{w}_m$ is a sequence of contiguous word tokens (its length is denoted N_m) understood to refer to a real-world *entity*. The entities (and the mapping of mentions to entities) are not known in advance. While our focus in this paper is names of people, the task is defined in a more generic way.

Formally, the task is to construct a table $\mathbf{x}$ where rows correspond to entities and columns to functional *fields*. The number of entities and the number of fields are not prespecified. $\boldsymbol{x}_{\cdot,j}$ denotes the jth column of $\mathbf{x}$, and $x_{i,j}$ is a single word type filling the cell in row i, column j. An example is Table 1, where the fields are first-name, last-name, title, middle-name, and so on. In addition to the table, we require that each mention be mapped to an entity (i.e., a row in the table). Success at this task therefore requires (i) identifying entities, (ii) discovering the internal structure of mentions (effectively canonicalizing them), and (iii) mapping mentions to entities (therefore resolving coreference relationships among mentions). Note that this task differs from previous work on knowledge base population (e.g., McNamee, 2009) because the schema is not formally defined in advance; rather, the number of fields and their meaning must be induced from just a few prototype examples.

To incorporate partial supervision, a subset of the table $\mathbf{x}$ is specified manually by an annotator. We denote this subset of "prototypes" by $\tilde{\mathbf{x}}$; for entries that are unspecified by the user, we write $\tilde{x}_{i,j} = \varnothing$. Prototypes are not assumed to provide complete information for any entity.

3 Model

We now craft a nonparametric generative story that explains both the latent table and the observed mentions. The model incorporates three nonparametric components, allowing an unbounded number of rows (entities) and columns (fields), as well as an unbounded number of values per column (field values). A plate diagram for the graphical model is shown in Figure 1.

A key point is that the column distributions $\boldsymbol{\phi}$ range over possible values at the entity level, not over mentions in text. For example, $\boldsymbol{\phi}_2$ might be the distribution over possible last names and $\boldsymbol{\phi}_3$ the distribution over elected office titles. Note that $\boldsymbol{\phi}_2$ would contain a low value for the last name *Obama* — which indicates that few people have this last name — even though a very high proportion of mentions in our data include the string *Obama*.

The user-generated entries ($\tilde{\mathbf{x}}$) can still be treated as the outcome of the generative process: using exchangeability, we treat these entries as the first samples drawn in each column. In this work, we treat them as fully observed, but it is possible to treat them as noisy and incorporate a stochastic dependency between $x_{i,j}$ and $\tilde{x}_{i,j}$.

4 Inference

We now develop sampling-based inference for the model described in the previous section. We begin with a token-based collapsed Gibbs sampler, and then add larger-scale Metropolis-Hastings moves.

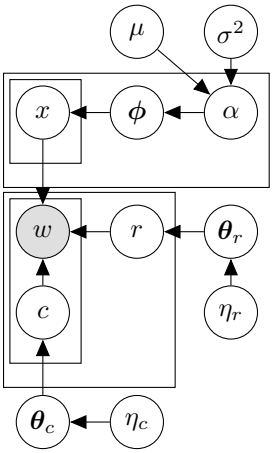

Figure 1: A plate diagram for the text-and-tables graphical model. The upper plate is the table $\boldsymbol{x}$, and the lower plate is the set of textual mentions. Notation is defined in the generative model to the right.

- **Generate the table entries**. For each column j,
  - Draw a concentration parameter α_j from a log-normal distribution, $\log \alpha_j \sim \mathcal{N}(\mu, \sigma^2)$.
  - Draw a distribution over strings from a Dirichlet process $\boldsymbol{\phi}_j \sim DP(\alpha_j, G_0)$, where the base distribution G_0 is a uniform distribution over strings in a fixed character alphabet, up to an arbitrary finite length.
  - For each row i, draw the entry $x_{i,j} \sim \boldsymbol{\phi}_j$.
- **Generate the text mentions**.
  - Draw a prior distribution over rows from a stick-breaking distribution, $\boldsymbol{\theta}_r \sim \text{Stick}(\eta_r)$.
  - Draw a prior distribution over columns from a stick-breaking distribution, $\boldsymbol{\theta}_c \sim \text{Stick}(\eta_c)$.
  - For each mention $\boldsymbol{w}_m$,
    - Draw a row in the table $r_m \sim \boldsymbol{\theta}_r$.
    - For each word token $w_{m,n}$ ($n \in \{1, \ldots, N_m\}$),
      - Draw a column in the table $c_{m,n} \sim \boldsymbol{\theta}_c$.
      - **Set** the text $w_{m,n} = x_{r_m, c_{m,n}}$.

4.1 Gibbs sampling

A key aspect of the generative process is that the word token $w_{m,n}$ is completely determined by the table $\mathbf{x}$ and the row and column indicators r_m and $c_{m,n}$: given that a token was generated by row i and column j of the table, it must be identical to the value of $x_{i,j}$. Using Bayes' rule, we can reverse this deterministic dependence: given the values for the row and column indices, the entries in the table are restricted to exact matches with the text mentions that they generate. This allows us to marginalize the unobserved entries in the table. We can also marginalize the distributions $\boldsymbol{\theta}_r$, $\boldsymbol{\theta}_c$, and $\boldsymbol{\phi}_j$, using the standard collapsed Gibbs sampling equations for Dirichlet processes. Thus, sampling the row and column indices is all that is required to explore the entire space of model configurations.

4.1.1 Conditional probability for word tokens

The conditional sampling distributions for both rows and columns will marginalize the table (besides the prototypes $\tilde{\mathbf{x}}$). To do this, we must be able to compute $P(w_{m,n} \mid r_m = i, c_{m,n} = j, \tilde{\mathbf{x}}, \boldsymbol{w}_{-(m,n)}, \boldsymbol{r}_{-m}, \boldsymbol{c}_{-(m,n)}, \alpha_j)$, which represents the probability of generating word $w_{m,n}$, given $r_m = i$ and $c_{m,n} = j$. The notation $\boldsymbol{w}_{-(m,n)}, \boldsymbol{r}_{-m}$, and $\boldsymbol{c}_{-m,n}$ represent the words, row indices, and column indices for all mentions besides $w_{m,n}$. For simplicity, we will elide these variables in much of the subsequent notation.

We first consider the case where we have a user-specified entry for the row and column $\langle i, j \rangle$ — that is, if $\tilde{x}_{ij} \neq \varnothing$. Then the probability is simply,

$$P(w_{m,n} \mid r_m = i, c_{m,n} = j, \tilde{\mathbf{x}}, \ldots) = \begin{cases} 1, & \text{if } \tilde{x}_{ij} = w_{m,n} \\ 0, & \text{if } \tilde{x}_{ij} \neq w_{m,n}. \end{cases} \tag{1}$$

Because the table cell x_{ij} is observed, we do not marginalize over it; we have a generative probability of one if the word matches, and zero otherwise. If the table cell x_{ij} is not specified by the user, then we marginalize over its possible values. For any given x_{ij}, the probability $P(w_{m,n} \mid x_{ij}, r_m = i, c_{m,n} = j)$ is still a delta function, so we have:

$$\int P(w_{m,n} \mid x_{r_m, c_{m,n}}) P(x_{r_m, c_{m,n}} \mid \ldots)\, dx_{r_m, c_{m,n}} = P(x = w_{m,n} \mid \boldsymbol{w}_{-(m,n)}, \boldsymbol{r}_{-m}, \boldsymbol{c}_{-(m,n)}, \tilde{\mathbf{x}}, \ldots)$$

The integral is equal to the probability of the value of the cell $x_{r_m, c_{m,n}}$ being identical to the string $w_{m,n}$, given assignments to all other variables. To compute this probability, we again must consider two cases: if the cell $x_{i,j}$ has generated some other string $w_{m',n'}$ then its value must be identical to that

string; otherwise it is unknown. More formally, for any cell $\langle i, j\rangle$, if $\exists w_{m',n'} : r_{m'} = i \wedge c_{m',n'} = j \wedge \langle m', n'\rangle \neq \langle m, n\rangle$, then $P(x_{i,j} = w_{m',n'}) = 1$; all other strings have zero probability. If $x_{i,j}$ has not generated any other entry, then its probability is conditioned on the other elements of the table $\mathbf{x}$. The known elements of this table are themselves determined by either the user entries $\tilde{\mathbf{x}}$ or the observations $\boldsymbol{w}_{-(m,n)}$. We can define these known elements as $\bar{\mathbf{x}}$, where $\bar{x}_{ij} = \varnothing$ if $\tilde{x}_{ij} = \varnothing \wedge \nexists\langle m, n\rangle : r_m = i \wedge c_{m,n} = j$. Then we can apply the standard Chinese restaurant process marginalization to obtain:

$$P(x_{ij} \mid \bar{\boldsymbol{x}}_{-(i,j)}, \alpha) = \begin{cases} \frac{\mathsf{N}(\bar{\boldsymbol{x}}_{-(i,j)}=x_{ij})}{\mathsf{N}(\bar{\boldsymbol{x}}_{-(i,j)}\neq\varnothing)+\alpha}, & \mathsf{N}(\bar{\boldsymbol{x}}_{-(i,j)} = x_{ij}) > 0 \\ \frac{\alpha}{\mathsf{N}(\bar{\boldsymbol{x}}_{-(i,j)}\neq\varnothing)+\alpha}, & \mathsf{N}(\bar{\boldsymbol{x}}_{-(i,j)} = x_{ij}) = 0 \end{cases} \quad (2)$$

In our implementation, we maintain the table $\bar{\mathbf{x}}$, updating it as we resample the row and column assignments. To construct the conditional distribution for any given entry, we first consult this table, and then compute the probability in Equation 2 for entries where $\bar{x}_{ij} = \varnothing$.

4.1.2 Sampling columns

We can now derive sampling equations for the column indices $c_{m,n}$. We first apply Bayes' rule to obtain $P(c_{m,n} \mid w_{m,n}, r_m, \ldots) \propto P(c_{m,n} \mid \boldsymbol{c}_{-(m,n)}, \eta_c) \times P(w_{m,n} \mid c_{m,n}, r_m, \tilde{\mathbf{x}}, \ldots)$. The likelihood term $P(w_{m,n} \mid c_{m,n}, \ldots)$ is defined in the previous section; we can compute the first factor using the standard Dirichlet process marginalization over $\boldsymbol{\theta}_c$. Writing $\mathsf{N}(c_{-(m,n)} = j)$ for the count of occurrences of column j in the set $\boldsymbol{c}_{-(m,n)}$, we obtain

$$P(c_{m,n} = j \mid c_{-(m,n)}, \eta_c) = \begin{cases} \frac{\mathsf{N}(c_{-(m,n)}=j)}{\mathsf{N}(c_{-(m,n)})+\eta_c}, & \text{if } \mathsf{N}(c_{-(m,n)} = j) > 0 \\ \frac{\eta_c}{\mathsf{N}(c_{-(m,n)})+\eta_c}, & \text{if } \mathsf{N}(c_{-(m,n)} = j) = 0 \end{cases} \quad (3)$$

4.1.3 Sampling rows

In principle the row indicators can be sampled identically to the columns, with the caveat that the generative probability $P(\boldsymbol{w}_m \mid r_m, \ldots)$ is a product across all N_m tokens in $\boldsymbol{w}_m$.[1] However, because of the tight probabilistic coupling between the row and column indicators, straightforward Gibbs sampling mixes slowly. Instead, we marginalize the column indicators while sampling r. Only the likelihood term is affected by this change:

$$P(\boldsymbol{w}_m \mid r_m, \boldsymbol{w}_{-m}, \boldsymbol{r}_{-m}, \ldots) = \sum_j P(c = j \mid \boldsymbol{c}_{-m}, \eta_c) P(w_{m,n} \mid c_{m,n} = j, r_m, \bar{\boldsymbol{x}}, \alpha). \quad (4)$$

The tokens are conditionally independent given the row, so we factor and then explicitly marginalize over each $c_{m,n}$. The chain rule gives the form in Equation 4, which contains terms for the prior over columns and the likelihood of the word; these are defined in Equations 2 and 3. Note that neither the inferred table $\bar{\mathbf{x}}$ nor the heldout column counts $\boldsymbol{c}_{-m}$ include counts from any of the cells in row m.

4.2 Column swaps

Suppose that during initialization, we encounter the string *Barry Obama* before encountering *Barack Obama*. We would then put *Barry* in the first-name column, and put *Barack* in some other column for nicknames. After making these initial decisions, they would be very difficult to undo using Gibbs sampling — we would have to first shift all instances of *Barry* to another column, then move an instance of *Barack* to the first-name column, and then move the instances of *Barry* to the nickname column. To rectify this issue, we perform sampling on the table itself, swapping the columns of entries in the table, while simultaneously updating the relevant column indices of the mentions.

In the proposal, we select at random a row t and indices i and j. In the table, we will swap $x_{t,i}$ with $x_{t,j}$; in the text we will swap the values of each $c_{m,n}$ whenever $r_m = t$ and $c_{m,n} = i$ or j. This proposal is symmetric, so no Hastings correction is required. Because we are simultaneously updating the table and the column indices, the generative likelihood of the words is unchanged; the only changes

[1] This relies on the assumption that the values of $\{c_{m,n}\}$ are mutually independent given $\boldsymbol{c}_{-m}$. Future work might apply a more structured model of the ways that fields are combined when mentioning an entity. For example, a first-order Markov model could learn that family names often follow given names, but the reverse rarely occurs (in English).

in the overall likelihood come from the column indices and the values of the cells in the table. Letting $\mathbf{x}^*, \mathbf{c}^*$ indicate the state of the table and column indices after the proposed move, we will accept with probability,

$$P_{\text{accept}}(\mathbf{x} \to \mathbf{x}^*) = \min\left(1, \frac{P(\mathbf{c}^*)P(\mathbf{x}^*)}{P(\mathbf{c})P(\mathbf{x})}\right) \quad (5)$$

We first consider the ratio of the table probabilities, $\frac{P(\mathbf{x}^*|\alpha)}{P(\mathbf{x}|\alpha)}$. Recall that each column of $\mathbf{x}$ is drawn from a Dirichlet process; appealing to exchangeability, we can treat the row t as the last element drawn, and compute the probabilities $P(x_{t,i} \mid \boldsymbol{x}_{-(t,i)}, \alpha_i)$, with $\boldsymbol{x}_{-(t,i)}$ indicating the elements of the column i excluding row t. This probability is given by Equation 2. For a swap of columns i and j, we compute the ratio:

$$\frac{P(x_{t,i} \mid \boldsymbol{x}_{-(t,j)}, \alpha_j)P(x_{t,j} \mid \boldsymbol{x}_{-(t,i)}, \alpha_i)}{P(x_{t,i} \mid \boldsymbol{x}_{-(t,i)}, \alpha_i)P(x_{t,j} \mid \boldsymbol{x}_{-(t,j)}, \alpha_j)} \quad (6)$$

Next we consider the ratio of the column probabilities, $\frac{P(\mathbf{c}^*)}{P(\mathbf{c})}$. Again we can apply exchangeability, $P(\mathbf{c}) = P(\{\boldsymbol{c}_m : r_m = t\} \mid \{\boldsymbol{c}_{m'} : r_{m'} \neq t\})P(\{\boldsymbol{c}_{m'} : r_{m'} \neq t\})$. The second term $P(\{\boldsymbol{c}_{m'} : r_{m'} \neq t\})$ is unaffected by the move, and so is identical in both the numerator and denominator of the likelihood ratio; probabilities from columns other than i and j also cancel in this way. The remaining ratio can be simplified to,

$$\left(\frac{P(c = j \mid \mathbf{c}_{-t}, \eta_c)}{P(c = i \mid \mathbf{c}_{-t}, \eta_c)}\right)^{\mathsf{N}(r=t \wedge c=i) - \mathsf{N}(r=t \wedge c=j)} \quad (7)$$

where the counts $\mathsf{N}()$ are from the state of the sampler before executing the proposed move. The probability $P(c = i \mid \mathbf{c}_{-t}, \eta_c)$ is defined in Equation 3, and the overall acceptance ratio for column swaps is the product of (6) and (7).

4.3 Hyperparameters

The concentration parameters η_r and η_c help to control the number of rows and columns in the table, respectively. These parameters are updated to their maximum likelihood values using gradient-based optimization, so our overall inference procedure is a form of Monte Carlo Expectation-Maximization (Wei and Tanner, 1990).

The concentration parameters α_j control the diversity of each column in the table: if α_j is low then we expect a high degree of repetition, as with titles; if α_j is high then we expect a high degree of diversity. When the sampling procedure adds a new column, there is very little information for how to set its concentration parameter, as the conditional likelihood will be flat. Consequently, greater care must be taken to handle these priors appropriately.

We place a log-normal hyperprior on the column concentration parameters, $\log \alpha_j \sim \mathcal{N}(\mu, \sigma^2)$. The parameters of the log-normal are shared across columns, which provides additional information to constrain the concentration parameters of newly-created columns. We then use Metropolis-Hastings to sample the values of each α_j, using the joint likelihood,

$$P(\alpha_j, \bar{\boldsymbol{x}}^{(j)} \mid \mu, \sigma^2) \propto \frac{\exp(-(\log \alpha_j - \mu)^2)\alpha_j^{k_j}\Gamma(\alpha_j)}{2\sigma^2\Gamma(n_j + \alpha_j)},$$

where $\bar{\boldsymbol{x}}^{(j)}$ is column j of the inferred table, n_j is the number of specified entries in column j of the table $\bar{\boldsymbol{x}}$ and k_j is the number of unique entries in the column; see Rasmussen (2000) for a derivation. After repeatedly sampling several values of α_j for each column in the table, we update μ and σ^2 to their maximum-likelihood estimates.

5 Temporal Prominence

Andy Warhol predicted, "in the future, everyone will be world-famous for fifteen minutes." A model of temporal dynamics that accounts for the fleeting and fickle nature of fame might yield better performance for transient entities, like Joe the Plumber. Among several alternatives for modeling temporal dynamics in latent variable models, we choose a simple non-parametric approach: the recurrent Chinese restaurant process (RCRP; Ahmed and Xing, 2008). The core idea of the RCRP is that time is partitioned into epochs, with a unique Chinese restaurant process in each epoch. Each CRP has a prior which takes the form of pseudo-counts computed from the counts in previous epochs. We employ the simplest version of the RCRP, a first-order Markov model in which the prior for epoch t is equal to the vector of counts for epoch $t - 1$:

$$P(r_m^{(t)} = i | \boldsymbol{r}_{1\ldots m-1}^{(t)}, \boldsymbol{r}^{(t-1)}, \eta_r) \propto \begin{cases} \mathsf{N}(\boldsymbol{r}_{1\ldots m-1}^{(t)} = i) + \mathsf{N}(\boldsymbol{r}^{(t-1)} = i), & \text{if} > 0; \\ \eta_r, & \text{otherwise.} \end{cases} \tag{8}$$

The count of row i in epoch $t-1$ is written $\mathsf{N}(\boldsymbol{r}^{(t-1)} = i)$; the count in epoch t for mentions 1 to $m-1$ is written $\mathsf{N}(\boldsymbol{r}_{1\ldots m-1}^{(t)} = i)$. As before, we can apply exchangeability to treat each mention as the last in the epoch, so during inference we can replace this with the count $\mathsf{N}(\boldsymbol{r}_{-m}^{(t)})$. Note that there is *zero probability* of drawing an entity that has no counts in epochs t or $t-1$ but exists in some other epoch; the probability mass η_r is reserved for drawing a new entity, and the chance of this matching some existing entity from another epoch is vanishingly small.

During Gibbs sampling, we also need to consider the effect of $r_m^{(t)}$ on the subsequent epoch $t+1$. While space does not permit a derivation, the resulting probability is proportional to

$$P(\boldsymbol{r}^{(t+1)} | \boldsymbol{r}_{-m}^{(t)}, r_m^{(t)} = i, \eta_r) \propto \begin{cases} 1 & \text{if } \mathsf{N}(\boldsymbol{r}^{(t+1)} = i) = 0, \\ \frac{\mathsf{N}(\boldsymbol{r}^{(t+1)}=i)}{\eta_r} & \text{if } \mathsf{N}(\boldsymbol{r}_{-m}^{(t)} = i) = 0, \\ 1 + \frac{\mathsf{N}(\boldsymbol{r}^{(t+1)}=i)}{\mathsf{N}(\boldsymbol{r}_{-m}^{(t)}=i)} & \text{if } \mathsf{N}(\boldsymbol{r}_{-m}^{(t)} = i) > 0. \end{cases} \tag{9}$$

This favors entities which are frequent in epoch $t+1$ but infrequent in epoch t.

The move to a recurrent Chinese restaurant process does not affect the sampling equations for the columns c, nor the concentration parameters of the table, α. The only part of the inference procedure that needs to be changed is the optimization of the hyperparameter η_r; the log-likelihood is now the sum across all epochs, and each epoch makes a contribution to the gradient.

6 Evaluation Setup

Our model jointly performs three tasks: identifying a set of entities, discovering the set of fields, and matching mention strings with the entities and fields to which they refer. We are aware of no prior work that performs these tasks jointly, nor any dataset that is annotated for all three tasks.[2] Consequently, we focus our quantitative evaluation on what we take to be the most important subtask: identifying the entities which are mentioned in raw text. We annotate a new dataset of blog text for this purpose, and design precision and recall metrics to reward systems that recover as much of the reference set as possible, while avoiding spurious entities and fields. We also perform a qualitative analysis, noting the areas where our method outperforms string matching approaches, and where there is need for further improvement.

Data Evaluation was performed on a corpus of blogs describing United States politics in 2008 (Eisenstein and Xing, 2010). We ran the Stanford Named Entity Recognition system (Finkel et al., 2005) to obtain a set of 25,000 candidate mentions which the system judged to be names of people. We then pruned strings that appeared fewer than four times and eliminated strings with more than seven tokens (these were usually errors). The resulting dataset has 19,247 mentions comprising 45,466 word tokens, and 813 unique mention strings.

Gold standard We develop a *reference set* of 100 entities for evaluation. This set was created by sorting the unique name strings in the training set by frequency, and manually merging strings that reference the same entity. We also manually discarded strings from the reference set if they resulted from errors in the preprocessing pipeline (tokenization and named entity recognition). Each entity is represented by the set of all word tokens that appear in its references; there are a total of 231 tokens for the 100 entities. Most entities only include first and last names, though the most frequent entities have many more: for example, the entity **Barack Obama** has known names: {*Barack*, *Obama*, *Sen.*, *Mr.*}.

Metrics We evaluate the recall and precision of a system's *response* set by matching against the reference set. The first step is to create a bipartite matching between response and reference entities.[3] Using a cost function that quantifies the sim-

[2] Recent work exploiting Wikipedia disambiguation pages for evaluating cross-document coreference suggests an appealing alternative for future work (Singh et al., 2011).

[3] Bipartite matchings are typical in information extraction evaluation metrics (e.g., Doddington et al., 2004).

ilarity of response and reference entities, we optimize the matching using the Kuhn-Munkres algorithm (Kuhn, 1955). For recall, the cost function counts the number of shared word tokens, divided by the number of word tokens in the reference entities; the recall is one minus the average cost of the best matching (with a cost of one for reference entities that are not matched, and no cost for unmatched response entities). Precision is computed identically, but we normalize by the number of word tokens in the response entity. Precision assigns a penalty of one to unmatched response entities and no penalty for unmatched reference entities.

Note that this metric grossly underrates the precision of all systems: the reference set is limited to 100 entities, but it is clear that our text mentions many other people. This is harsh but fair: all systems are penalized equally for identifying entities that are not present in the reference set, and the ideal system will recover the fifty reference entities (thus maximizing recall) while keeping the table as compact as possible (thus maximizing precision). However, the raw precision values have little meaning outside the context of a direct comparison under identical experimental conditions.

Systems The initial seed set for our system consists of a partial annotation of five entities (Table 1) — larger seed sets did not improve performance. We run the inference procedure described in the previous section for 20,000 iterations, and then obtain a final database by taking the intersection of the inferred tables $\bar{\mathbf{x}}$ obtained at every 100 iterations, starting with iteration 15,000. To account for variance across Markov chains, we perform three different runs. We evaluate a non-temporal version of our model (as described in Sections 3 and 4), and a temporal version with 5 epochs. For the non-temporal version, a non-parallel C implementation had a wall clock sampling time of roughly 16 hours; the temporal version required 24 hours.

We compare against a baseline that incrementally clusters strings into entities using a string edit distance metric, based on the work of Elmacioglu et al. (2007). Starting from a configuration in which each unique string forms its own cluster, we incrementally merge clusters using the single-link criterion, based on the minimum Jaccard edit distance between each pair of clusters. This yields a series of outputs that move along the precision-recall curve, with precision increasing as the clusters encompass more strings. There is prior work on heuristics for selecting a stopping point, but we compare our results against the entire precision-recall curve (Manning et al., 2008).

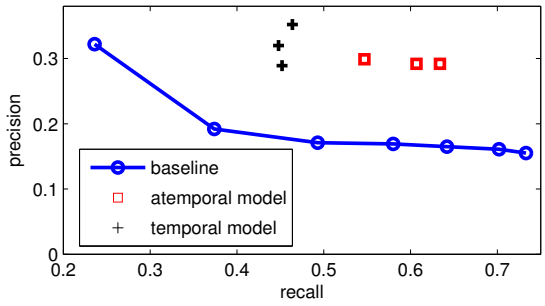

Figure 2: The precision and recall of our models, as compared to the curve defined by the incremental clustering baseline. Each point indicates a unique sampling run.

Bill	Clinton	Benazir		Bhutto
Nancy	Pelosi	Speaker		
John	Kerry	Sen.		Roberts
Martin	King	Dr.	Jr.	Luther
Bill	Nelson			

Table 2: A subset of the entity database discovered by our model, hand selected to show highlight interesting success and failure cases.

7 Results

The results of our evaluation are shown in Figure 2. All sampling runs from our models lie well beyond the precision-recall curve defined by the baseline system, demonstrating the ability to achieve reasonable recall with a far more compact database. The baseline system can achieve nearly perfect recall by creating one entity per unique string, but as it merges strings to improve precision, its recall suffers significantly. As noted above, perfect precision is not possible on this task, because the reference set covers only a subset of the entities that appear in the data. However, the numbers do measure the ability to recover the reference entities in the most compact table possible, allowing a quantitative comparison of our models and the baseline approach.

Table 2 shows a database identified by the atemporal version of our model. The most densely-populated columns in the table correspond to well-defined name parts: columns 1 and 2 are almost exclusively populated with first and last names respectively, and column 3 is mainly populated by titles. The remaining columns are more of a grab bag. Column 4 correctly captures *Jr.* for **Martin Luther King**; column 5 correctly captures *Luther*, but mistakenly contains *Roberts* (thus merging the **John Kerry** and **John Roberts** entities), and *Bhutto* (thus helping to merge the **Bill Clinton** and **Benazir Bhutto** entities).

The model successfully distinguishes some, but not all, of the entities that share tokens. For example, the model separates **Bill Clinton** from **Bill Nelson**; it also separates **John McCain** from **John Kerry** (whom it mistakenly merges with **John Roberts**). The ability to distinguish individuals who share first names is due in part to the model attributing a low concentration parameter to first names, meaning that some repetition in the first name column is expected. The model correctly identifies several titles and alternative names, including the rare title *Speaker* for **Nancy Pelosi**; however, it misses others, such as the *Senator* title for **Bill Nelson**. This may be due in part to the sample merging procedure used to generate this table, which requires that a cell contain the same value in at least 80% of the samples.

Many errors may be attributed to slow mixing. After mistakenly merging **Bhutto** and **Clinton** at an early stage, the Gibbs sampler — which treats each mention independently — is unable to separate them. Given that several other mentions of **Bhutto** are already in the row occupied by **Clinton**, the overall likelihood would benefit little from creating a new row for a single mention, though moving all such mentions simultaneously would result in an improvement. Larger scale Metropolis-Hastings moves, such as split-merge or type-based sampling (Liang et al., 2010) may help.

8 Related Work

Information Extraction A tradition of research in information extraction focuses on processing raw text to fill in the fields of manually-defined templates, thus populating databases of events or relations (McNamee and Dang, 2009). While early approaches focused on surface-level methods such as wrapper induction (Kushmerick et al., 1997), more recent work in this area includes Bayesian nonparametrics to select the number of rows in the database (Haghighi and Klein, 2010a). However, even in such nonparametric work, the form of the template and the number of slots are fixed in advance. Our approach differs in that the number of fields and their meaning is learned from data. Recent work by Chambers and Jurafsky (2011) approaches a related problem, applying agglomerative clustering over sentences to detect *events*, and then clustering syntactic constituents to induce the relevant fields of each event entity. As described in Section 6, our method performs well against an agglomerative clustering baseline, though a more comprehensive comparison of the two approaches is an important step for future work.

Name Segmentation and Structure A related stream of research focuses specifically on names: identifying them in raw text, discovering their structure, and matching names that refer to the same entity. We do not undertake the problem of named entity recognition (Tjong Kim Sang, 2002), but rather apply an existing NER system as a preprocessing step (Finkel et al., 2005). Typical NER systems do not attempt to discover the internal structure of names or a database of canonical names, although they often use prefabricated "gazetteers" of names and name parts as features to improve performance (Borthwick et al., 1998; Sarawagi and Cohen, 2005).

Charniak (2001) shows that it is possible to learn a model of name structure, either by using coreference information as labeled data, or by leveraging a small set of hand-crafted constraints. Elsner et al. (2009) develop a nonparametric Bayesian model of name structure using adaptor grammars, which they use to distinguish *types* of names (e.g., people, places, and organizations). Li et al. (2004) use a set of manually-crafted "transformations" of name parts to build a model of how a name might be rendered in multiple different ways. While each of these approaches bears on one or more facets of the problem that we consider here, none provides a holistic treatment of name disambiguation and structure.

Resolving Mentions to Entities The problem of resolving mentions to entities has been approach from a variety of different perspectives. There is an extensive literature on probabilistic record linkage, in which database records are compared to determine if they are likely to have the same real-world referents (e.g., Felligi and Sunter, 1969; Bilenko et al., 2003). Most approaches focus on pairwise assessments of whether two records are the same, whereas our method attempts to infer a single coherent model of the underlying relational data. Some more recent work in record linkage has explicitly formulated the task of inferring a latent relational model of a set of observed datasets (e.g., Cohen et al., 2000; Pasula et al., 2002; Bhattacharya and Getoor, 2007); however, to our knowledge, these prior models have all exploited some predefined database schema (i.e., set of columns), which our model does not require. Many of these prior models have been applied to bibliographic data, where different conventions and abbreviations lead to imperfect matches in different references to the same publication. In our task, we consider name mentions in raw text; such mentions are short, and may not offer as many redundant clues for linkage as bibliographic references.

In natural language processing, *coreference resolution* is the task of grouping entity mentions (strings), in one or more documents, based on their common referents in the world. Although much of coreference resolution has on the single document setting, there has been some recent work on cross-document coreference resolution (Li et al., 2004; Haghighi and Klein, 2007; Poon and Domingos, 2008; Singh et al., 2011). The problem we consider is related to cross-document coreference, although we take on the additional challenge of providing a canonicalized name for each referent (the corresponding table row), and in inferring a structured representation of entity names (the table columns). For this reason, our evaluation focuses on the induced table of entities, rather than the clustering of mention strings. The best coreference systems depend on carefully crafted, problem-specific linguistic features (Bengtson and Roth, 2008) and external knowledge (Haghighi and Klein, 2010b). Future work might consider how to exploit such features for the more holistic information extraction setting.

9 Conclusion

This paper presents a Bayesian nonparametric approach to recover structured records from text. Using only a small set of prototype records, we are able to recover an accurate table that jointly identifies entities and internal name structure. In our view, the main advantage of a Bayesian approach compared to more heuristic alternatives is that it facilitates incorporation of additional information sources when available. In this paper, we have considered one such additional source, incorporating temporal context using the recurrent Chinese restaurant process.

We envision enhancing the model in several other respects. One promising direction is the incorporation of name structure, which could be captured using a first-order Markov model of the transitions between name parts. In the nonparametric setting, a transition matrix is unbounded along both dimensions, and this can be handled by a hierarchical Dirichlet process (HDP; Teh et al 2006).[4] We envision other potential applications of the HDP: for example, learning "topics" of entities which tend to appear together (i.e., given a mention of Mahmoud Abbas in the American press, a mention of Benjamin Netanyahu is likely), and handling document-specific burstiness (i.e., given that an entity is mentioned once in a document, it is much more likely to be mentioned again). Finally, we would like to incorporate lexical context from the sentences in which each entity is mentioned, which might help to distinguish, say, computer science researchers who share names with former defense secretaries or professional basketball players.

Acknowledgments This research was enabled by AFOSR FA95501010247, DARPA grant N10AP20042, ONR N000140910758, NSF DBI-0546594, IIS-0713379, IIS-0915187, IIS-0811562, an Alfred P. Sloan Fellowship, and Google's support of the Worldly Knowledge project at CMU. We thank the reviewers for their thoughtful feedback.

[4] One of the reviewers proposed to draw entire column sequences from a Dirichlet process. Given the relatively small number of columns and canonical name forms, this may be a straightforward and effective alternative to the HDP.

References

Amr Ahmed and Eric P. Xing. 2008. Dynamic non-parametric mixture models and the recurrent Chinese restaurant process with applications to evolutionary clustering. In *International Conference on Data Mining*.

Javier Artiles, Julio Gonzalo, and Satoshi Sekine. 2007. The SemEval-2007 WePS evaluation: establishing a benchmark for the web people search task. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, SemEval '07, pages 64–69. Association for Computational Linguistics.

Eric Bengtson and Dan Roth. 2008. Understanding the value of features for coreference resolution. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '08, pages 294–303, Stroudsburg, PA, USA. Association for Computational Linguistics.

Indrajit Bhattacharya and Lise Getoor. 2007. Collective entity resolution in relational data. *ACM Trans. Knowl. Discov. Data*, 1(1), March.

Mikhail Bilenko, William W. Cohen, Stephen Fienberg, Raymond J. Mooney, and Pradeep Ravikumar. 2003. Adaptive name-matching in information integration. *IEEE Intelligent Systems*, 18(5):16–23, September/October.

A. Borthwick, J. Sterling, E. Agichtein, and R. Grishman. 1998. Exploiting diverse knowledge sources via maximum entropy in named entity recognition. In *Sixth Workshop on Very Large Corpora New Brunswick, New Jersey. Association for Computational Linguistics.*

Nathanael Chambers and Dan Jurafsky. 2011. Template-based information extraction without the templates. In *Proceedings of ACL.*

Eugene Charniak. 2001. Unsupervised learning of name structure from coreference data. In *Proceedings of the Second Meeting of the North American Chapter of the Association for Computational Linguistics*.

William W. Cohen, Henry Kautz, and David McAllester. 2000. Hardening soft information sources. In *Proceedings of the Sixth International Conference on Knowledge Discovery and Data Mining*, pages 255–259.

George Doddington, Alexis Mitchell, Mark Przybocki, Lance Ramshaw, Stephanie Strassel, and Ralph Weischedel. 2004. The automatic content extraction (ace) program: Tasks, data, and evaluation. In *4th international conference on language resources and evaluation (LREC'04)*.

Jacob Eisenstein and Eric Xing. 2010. The CMU 2008 political blog corpus. Technical report, Carnegie Mellon University.

Ergin Elmacioglu, Yee Fan Tan, Su Yan, Min-Yen Kan, and Dongwon Lee. 2007. Psnus: Web people name disambiguation by simple clustering with rich features. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 268–271, Prague, Czech Republic, June. Association for Computational Linguistics.

Micha Elsner, Eugene Charniak, and Mark Johnson. 2009. Structured generative models for unsupervised named-entity clustering. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 164–172, Boulder, Colorado, June. Association for Computational Linguistics.

I. P. Felligi and A. B. Sunter. 1969. A theory for record linkage. *Journal of the American Statistical Society*, 64:1183–1210.

Jenny R. Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating non-local information into information extraction systems by Gibbs sampling. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, ACL '05, pages 363–370, Stroudsburg, PA, USA. Association for Computational Linguistics.

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. 2010a. An entity-level approach to information extraction. In *Proceedings of the ACL 2010 Conference Short Papers*, ACLShort '10, pages 291–295, Stroudsburg, PA, USA. Association for Computational Linguistics.

Aria Haghighi and Dan Klein. 2010b. 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*, pages 385–393, Los Angeles, California, June. Association for Computational Linguistics.

Harold W. Kuhn. 1955. The Hungarian method for the assignment problem. *Naval Research Logistic Quarterly*, 2:83–97.

Nicholas Kushmerick, Daniel S. Weld, and Robert Doorenbos. 1997. Wrapper induction for information extraction. In *Proceedings of IJCAI*.

Xin Li, Paul Morie, and Dan Roth. 2004. Identification and tracing of ambiguous names: Discriminative and generative approaches. In *Proceedings of AAAI*, pages 419–424.

Percy Liang, Michael I. Jordan, and Dan Klein. 2010. Type-Based MCMC. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 573–581, Los Angeles, California, June. Association for Computational Linguistics.

Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to Information Retrieval*. Cambridge University Press, 1 edition, July.

Paul McNamee and Hoa Trang Dang. 2009. Overview of the TAC 2009 knowledge base population track. In *Proceedings of the Text Analysis Conference (TAC)*.

Hanna Pasula, Bhaskara Marthi, Brian Milch, Stuart Russell, and Ilya Shpitser. 2002. Identity uncertainty and citation matching. In *Advances in Neural Processing Systems 15*, Vancouver, British Columbia. MIT Press.

Hoifung Poon and Pedro Domingos. 2008. Joint unsupervised coreference resolution with Markov Logic. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 650–659, Honolulu, Hawaii, October. Association for Computational Linguistics.

Carl E. Rasmussen. 2000. The Infinite Gaussian Mixture Model. In *In Advances in Neural Information Processing Systems 12*, volume 12, pages 554–560.

Sunita Sarawagi and William W. Cohen. 2005. Semi-Markov conditional random fields for information extraction. In Lawrence K. Saul, Yair Weiss, and Léon Bottou, editors, *Advances in Neural Information Processing Systems 17*, pages 1185–1192. MIT Press, Cambridge, MA.

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

Yee W. Teh, Michael I. Jordan, Matthew J. Beal, and David M. Blei. 2006. Hierarchical dirichlet processes. *Journal of the American Statistical Association*, 101:1566–1581, December.

Erik F. Tjong Kim Sang. 2002. Introduction to the CoNLL-2002 shared task: Language-independent named entity recognition. In *Proceedings of the Sixth Conference on Natural Language Learning*.

Greg C. G. Wei and Martin A. Tanner. 1990. A Monte Carlo Implementation of the EM Algorithm and the Poor Man's Data Augmentation Algorithms. *Journal of the American Statistical Association*, 85(411):699–704.