Name Phylogeny

A Generative Model of String Variation

Nicholas Andrews, Jason Eisner and Mark Dredze

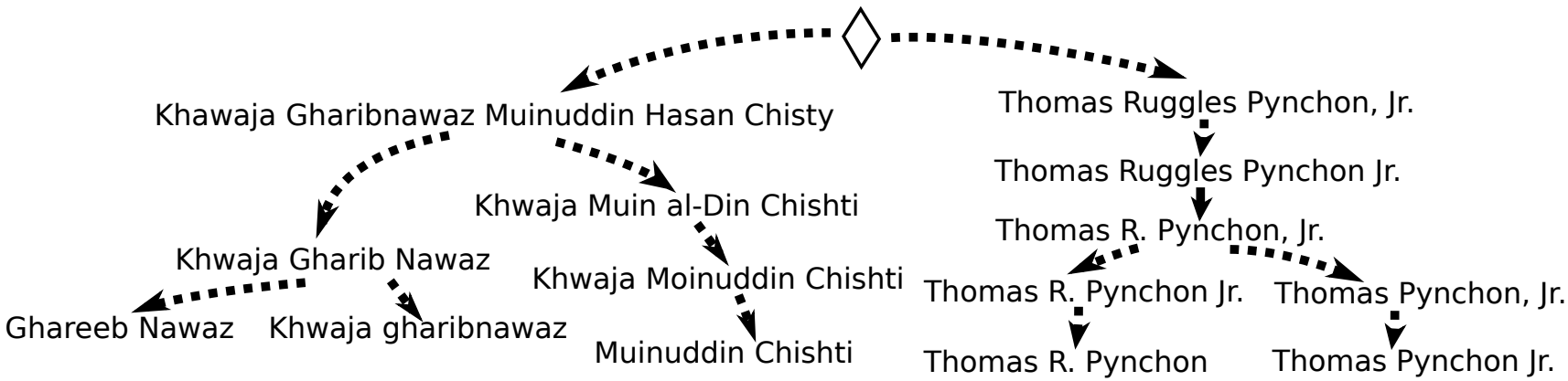
Department of Computer Science,
Johns Hopkins University

EMNLP 2012 – Thursday, July 12

Outline

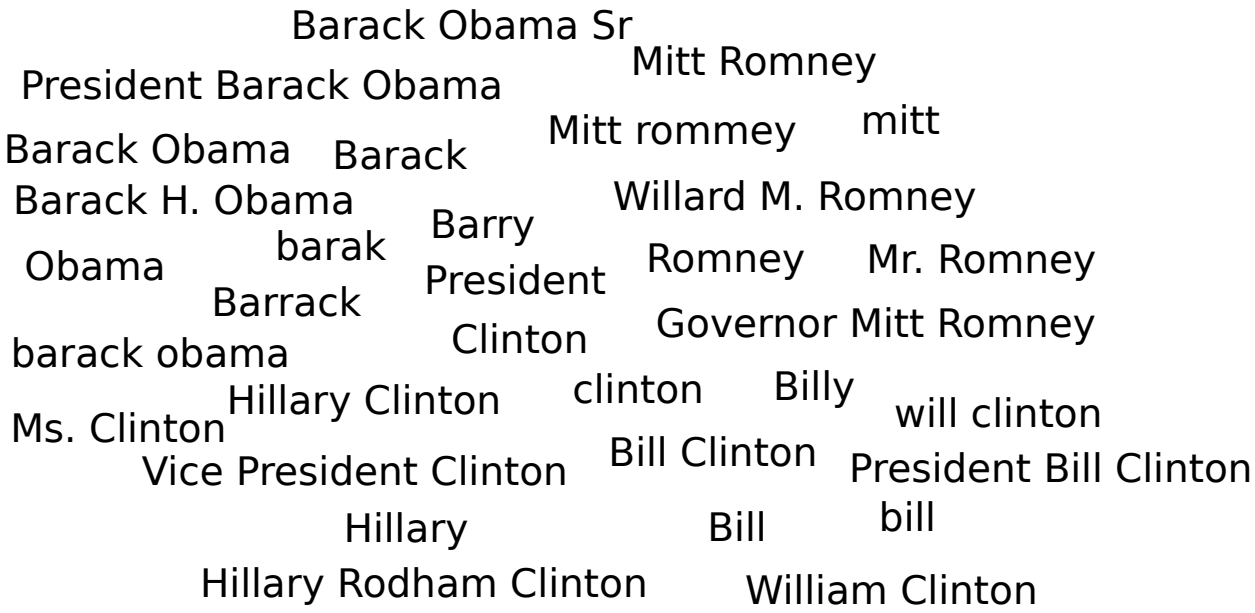
What's a name phylogeny?

A fragment of a "name phylogeny" learned by our model

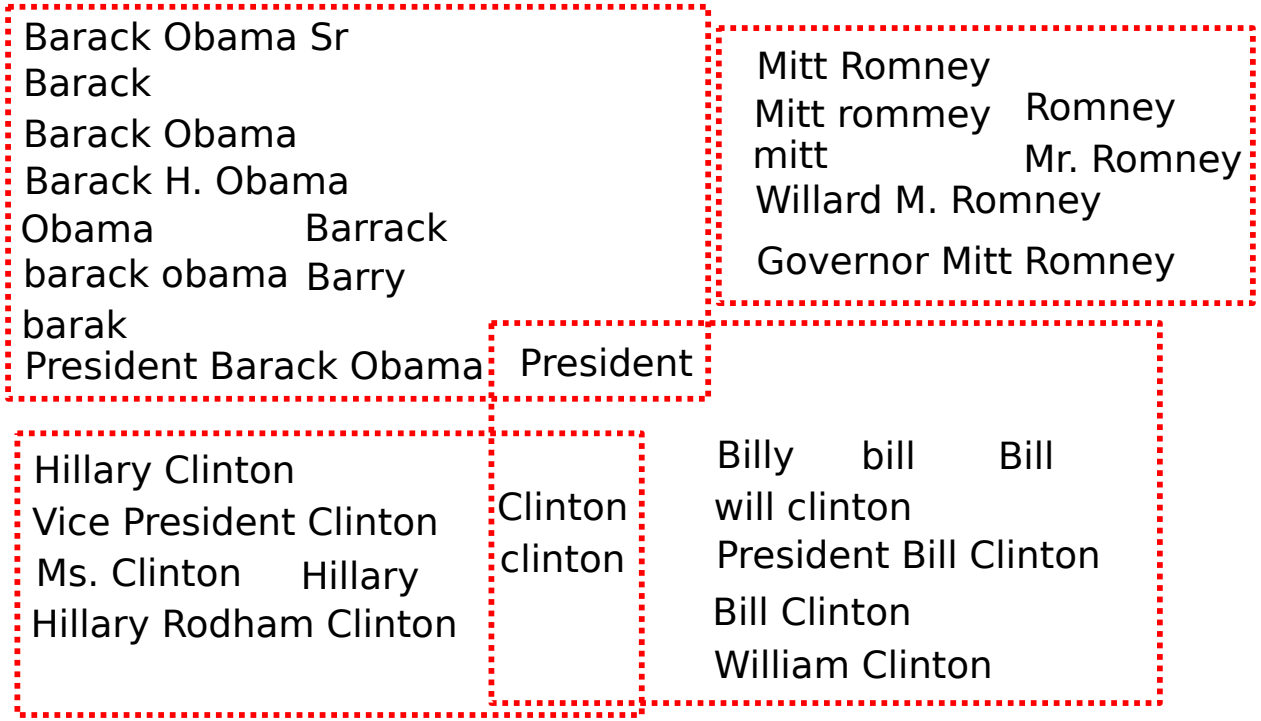


- Each edge corresponds to a "mutation"

Problem: organizing disorganized collections of strings

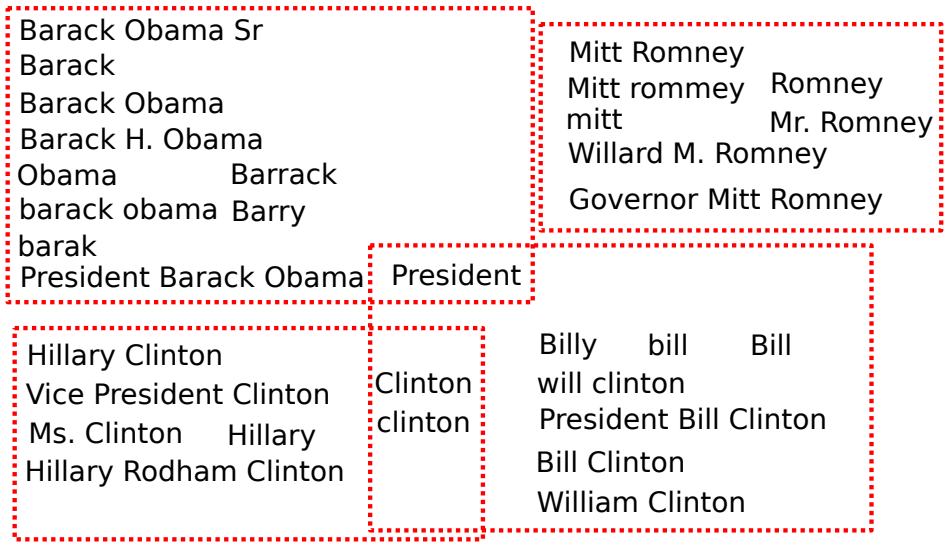


Problem: organizing disorganized collections of strings



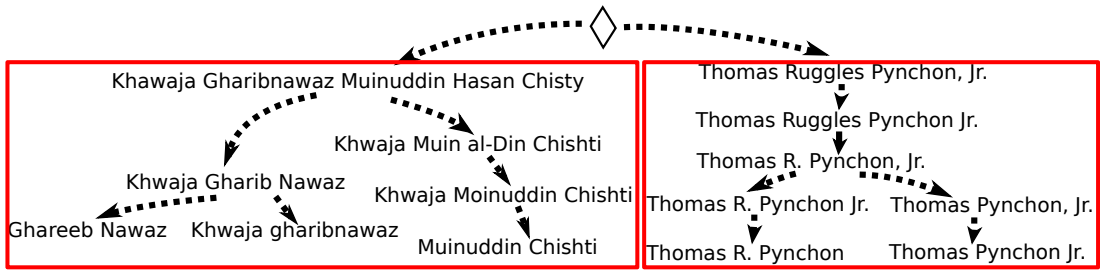
Challenges

- **Name variation:** the same entity may have different names, and a good measure of "similarity" between strings may not be available **(This work)**
- **Disambiguation:** different entities may have names in common, requiring the use of context to disambiguate between them

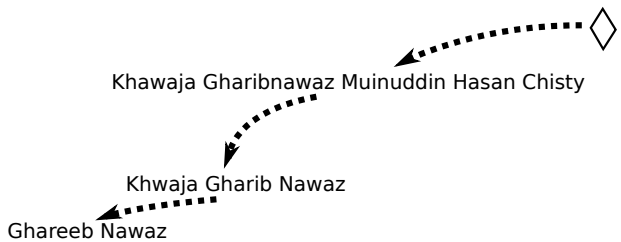


How does a name phylogeny help?

1. Organizes name variants into connected components (clusters)



2. Align names as "mutations" of one another



3. We can estimate a mutation model given a phylogeny, and a mutation model gives a distribution over phylogenies ($\rightarrow$ EM)

Outline

Generative Model

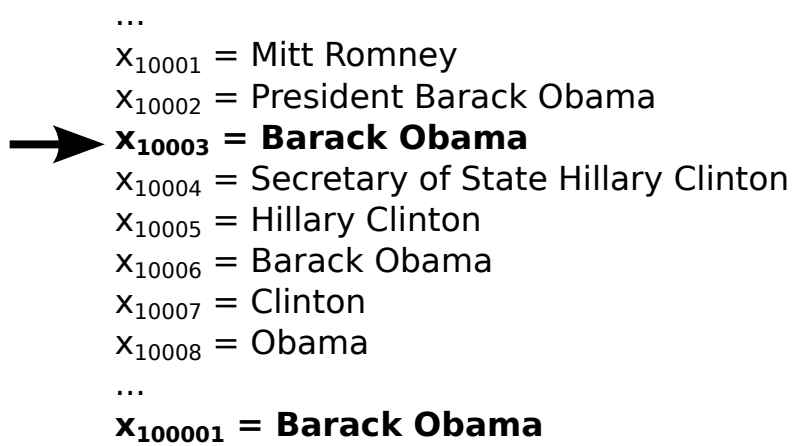
We propose a generative model for string variation explaining the reasons for name variation.

...
x_{10001} = Mitt Romney
x_{10002} = President Barack Obama
x_{10003} = Barack Obama
x_{10004} = Secretary of State Hillary Clinton
x_{10005} = Hillary Clinton
x_{10006} = Barack Obama
x_{10007} = Clinton
x_{10008} = Obama
...

What are the sources of variation for names?

Copying a previous mention

We can COPY a name seen before.

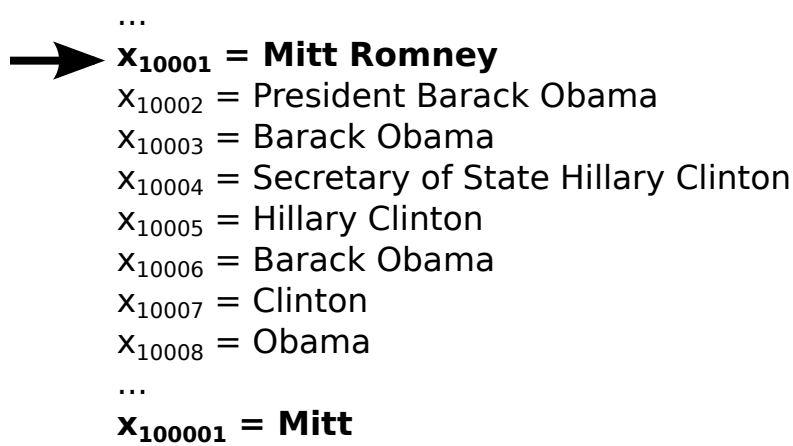


Procedure:

- Select a previous name mention uniformly at random
- Decide to COPY it with probability $1 - \mu$

Mutating a previous mention

We can MUTATE a name seen before.



Procedure:

- Select a previous name mention uniformly at random
- Decide to MUTATE it with probability μ
- Sample a mutation from $p(\cdot \mid \text{Mitt Romney})$

Generating a new name

We can generate a NEW name.

Procedure:

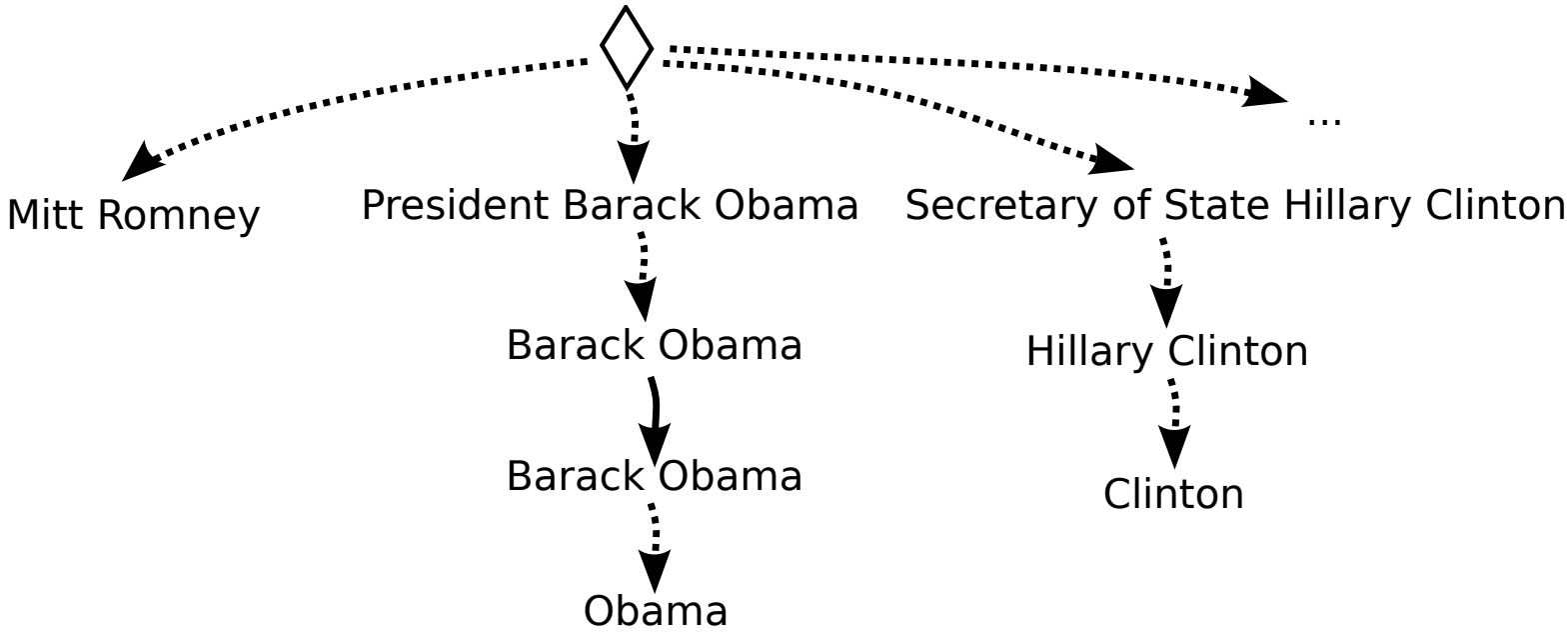
- Select $\diamondsuit$ with probability proportional to α (a "pseudocount")
- Sample a new name from $p(\cdot \mid \diamondsuit)$
  - A character language model

Generative model summary

To generate the next name mention:

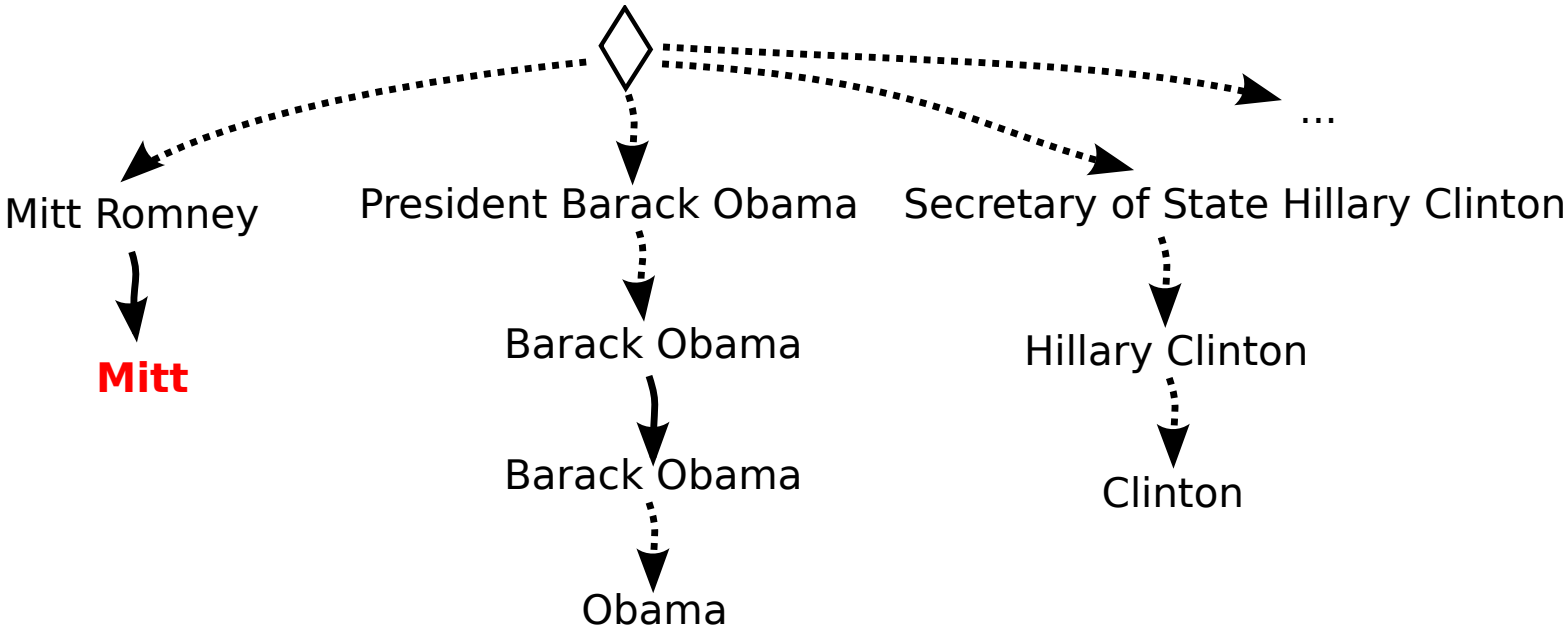
1. Pick an existing name mention w with probability $1/(\alpha + k)$
   - 1.1 Copy w verbatim with probability $1 - \mu$
   - 1.2 Mutate w with probability μ
2. Decide to talk about a new entity with probability $\alpha/(\alpha + k)$
   - 2.1 Generate a name for it

Generative model in action



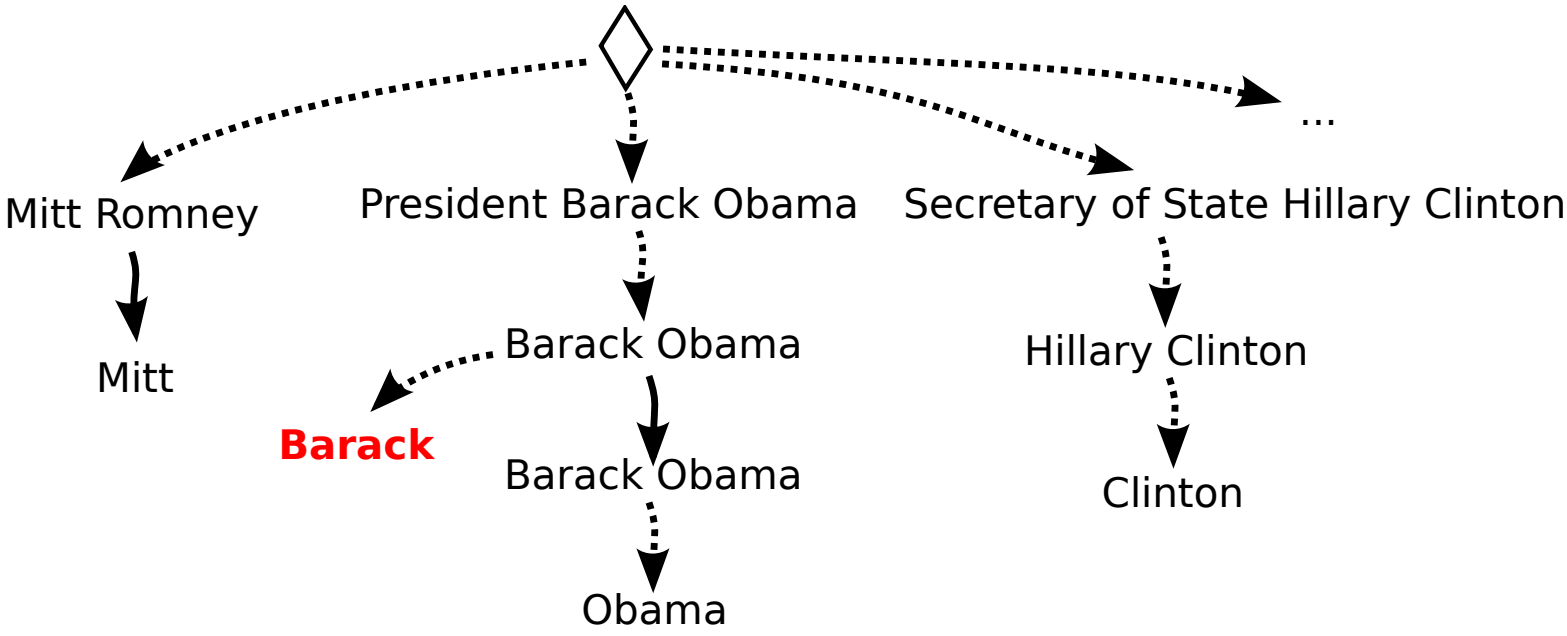
x_{10001} = Mitt Romney
x_{10002} = President Barack Obama
x_{10003} = Barack Obama
x_{10004} = Secretary of State Hillary Clinton
x_{10005} = Hillary Clinton
x_{10006} = Barack Obama
x_{10007} = Clinton
x_{10008} = Obama

Generative model in action



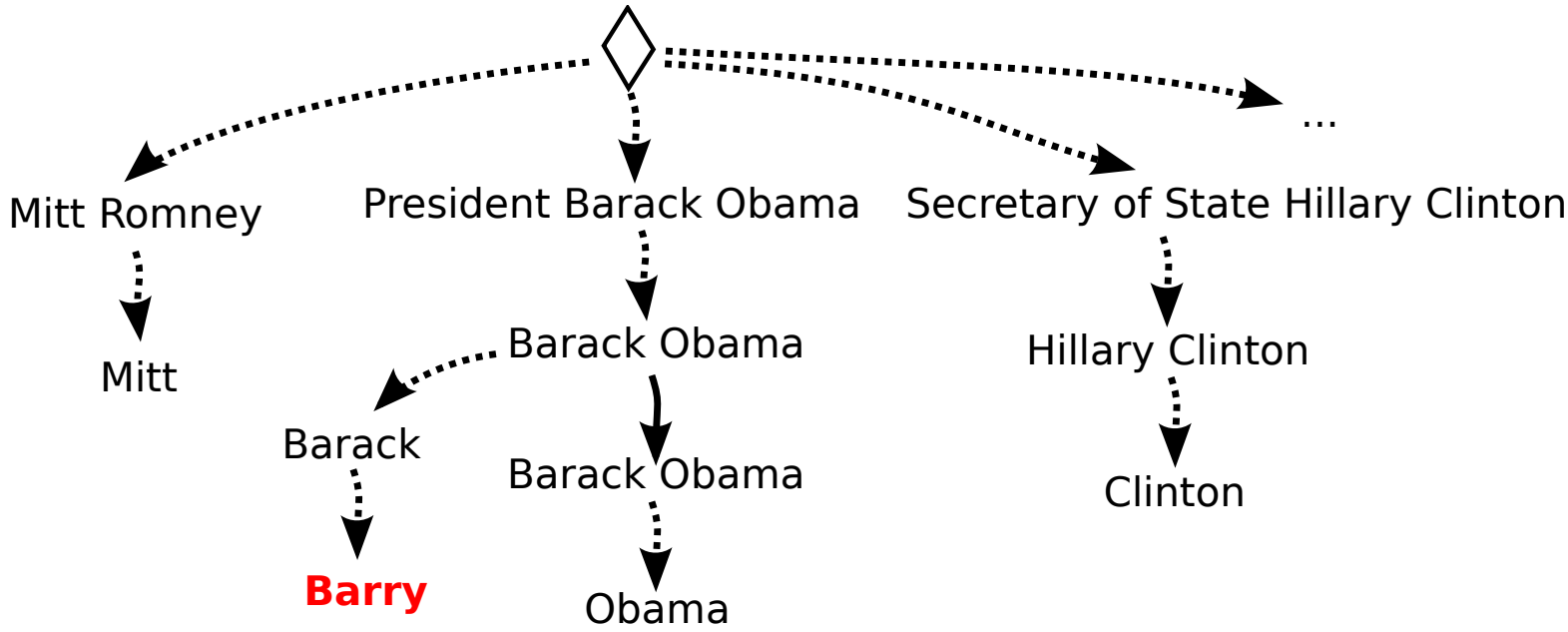
x_{10001} = Mitt Romney
x_{10002} = President Barack Obama
x_{10003} = Barack Obama
x_{10004} = Secretary of State Hillary Clinton
x_{10005} = Hillary Clinton
x_{10006} = Barack Obama
x_{10007} = Clinton
x_{10008} = Obama
x_{10009} = Mitt

Generative model in action



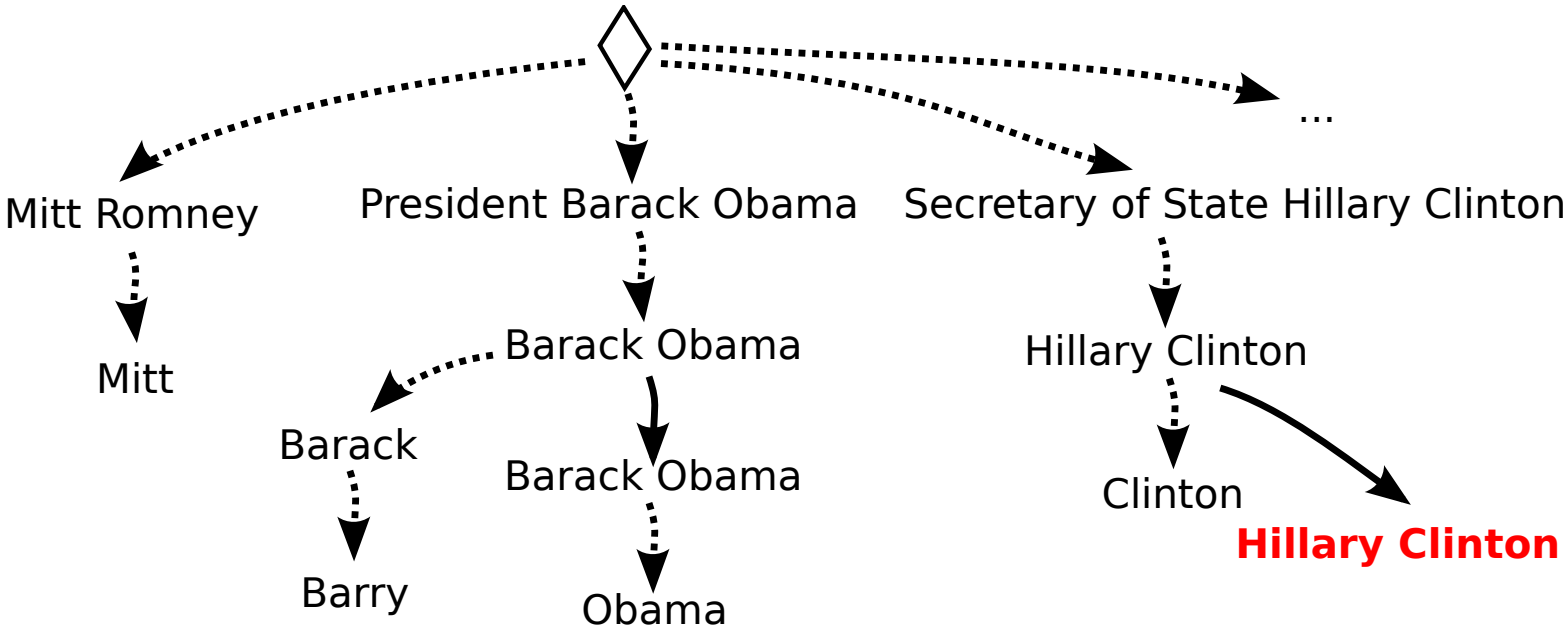
x_{10001} = Mitt Romney
x_{10002} = President Barack Obama
x_{10003} = Barack Obama
x_{10004} = Secretary of State Hillary Clinton
x_{10005} = Hillary Clinton
x_{10006} = Barack Obama
x_{10007} = Clinton
x_{10008} = Obama
x_{10009} = Mitt
x_{10010} = Barack

Generative model in action



x_{10001} = Mitt Romney
x_{10002} = President Barack Obama
x_{10003} = Barack Obama
x_{10004} = Secretary of State Hillary Clinton
x_{10005} = Hillary Clinton
x_{10006} = Barack Obama
x_{10007} = Clinton
x_{10008} = Obama
x_{10009} = Mitt
x_{10010} = Barack
x_{10011} = Barry

Generative model in action



x_{10001} = Mitt Romney
x_{10002} = President Barack Obama
x_{10003} = Barack Obama
x_{10004} = Secretary of State Hillary Clinton
x_{10005} = Hillary Clinton
x_{10006} = Barack Obama
x_{10007} = Clinton
x_{10008} = Obama
x_{10009} = Mitt
x_{10010} = Barack
x_{10011} = Barry
x_{10012} = Hillary Clinton

A few observations

- The proposed generative model is clearly naive
  - No model of discourse or of name structure
- The pseudocount α controls the likelihood of new names
- We assume a low mutation probability μ, so that most names are COPIED from earlier frequent names

Outline

Name variation as mutations

"Mutations" capture different types of name variation:

1. **Transcription errors:** Barack $\rightarrow$ barack
2. **Misspellings:** Barack $\rightarrow$ Barrack
3. **Abbreviations:** Barack Obama $\rightarrow$ Barack O.
4. **Nicknames:** Barack $\rightarrow$ Barry
5. **Dropping words:** Barack Obama $\rightarrow$ Barack

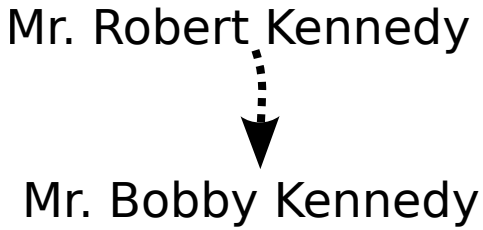
Mutation via probabilistic finite-state transducers

The mutation model is a **probabilistic finite-state transducer** with four character operations: COPY, SUBSTITUTE, DELETE, INSERT

- Character operations are conditioned on the right input character
- Latent regions of contiguous edits
- Back-off smoothing

Transducer parameters $\boldsymbol{\theta}$ determine the probability of being in different regions, and of the different character operations

Example: Mutating a name

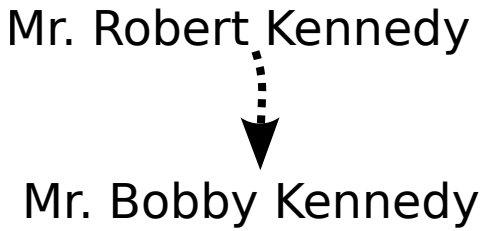


Example mutation

```
M r . _ R o b e r t     _ K e n n e d y $
M r . _[
```

Beginning of edit region

Example: Mutating a name

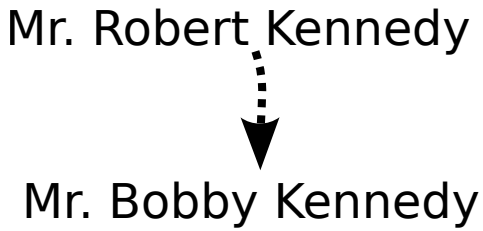


Example mutation

```
M r . _ R o b e r t     _ K e n n e d y $
M r . _ B
```

1 substitution operation: (R, B)

Example: Mutating a name

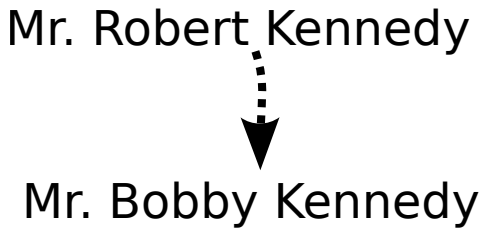


Example mutation

```
M r . _ R o b e r t     _ K e n n e d y $
M r . _[B o b
```

2 copy operations: (ε, o), (ε, b)

Example: Mutating a name

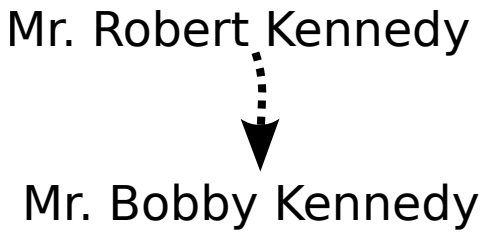


Example mutation

```
M r . _ R o b e r t     _ K e n n e d y $
M r . _[B o b
```

3 deletion operations: (e,ε), (r,ε), (t, ε)

Example: Mutating a name

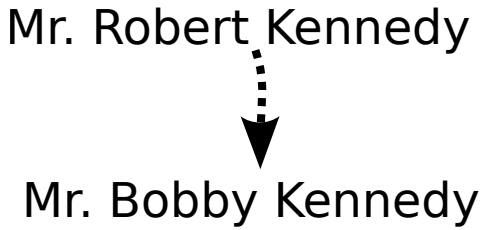


Example mutation

```
M r . _ R o b e r t     _ K e n n e d y$
M r . _[B o b       b y
```

2 insertion operations: (ε,b), (ε,y)

Example: Mutating a name

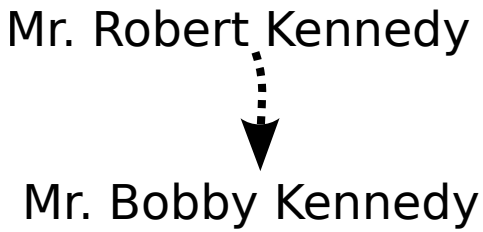


Example mutation

```
M r . _ R o b e r t     _ K e n n e d y $
M r . _[B o b       b y]
```

End of edit region

Example: Mutating a name



Example mutation

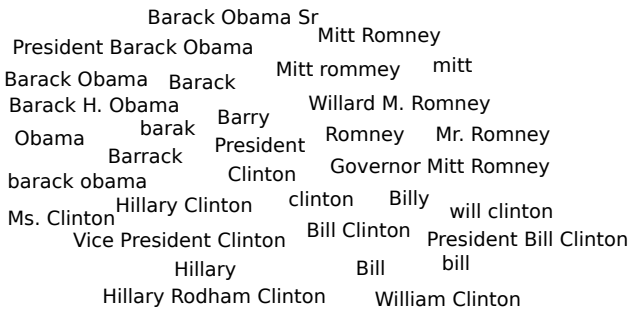
```
M r . _ R o b e r t     _ K e n n e d y $
M r . _[B o b       b y]_ K e n n e d y $
```

Outline

Inference

Input: An unaligned corpus of names ("bag-of-words")

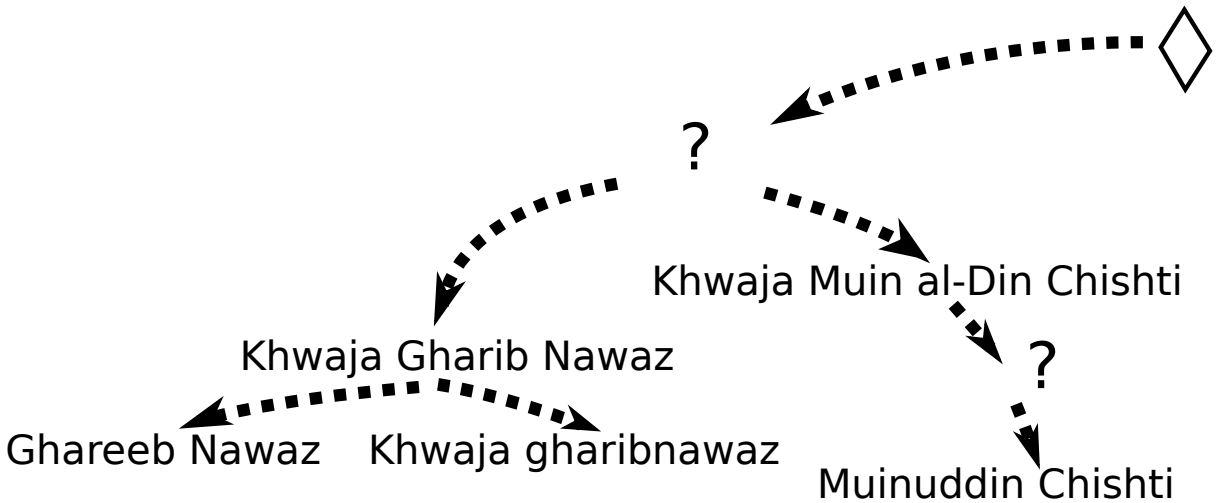
- The order in which the tokens were generated is unknown
- No "inputs" or "outputs" are known for the mutation model



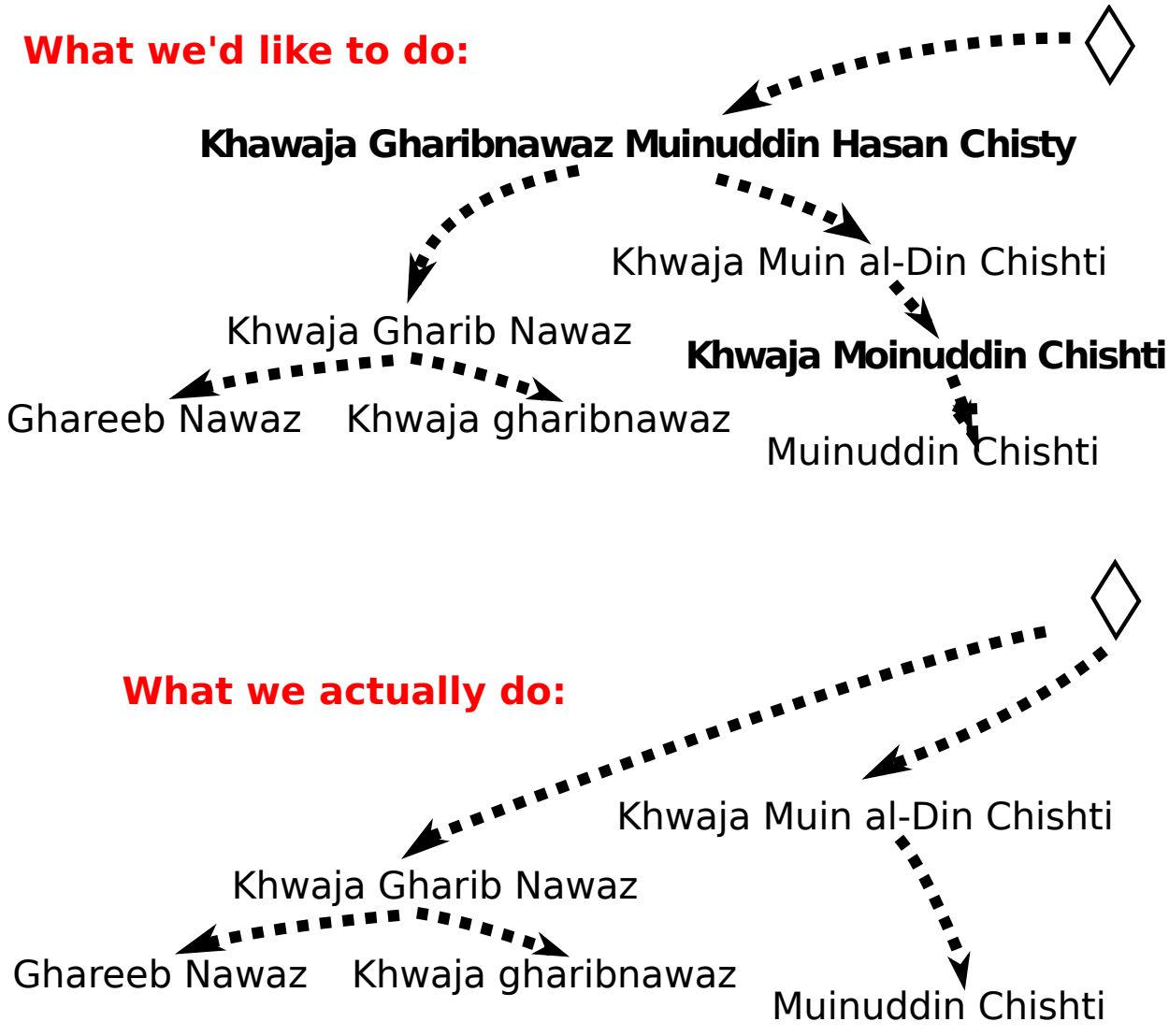
Output: A distribution over name phylogenies parametrized by transducer parameters $\boldsymbol{\theta}$

Observed vs unobserved names

Could there be latent forms in the phylogeny?

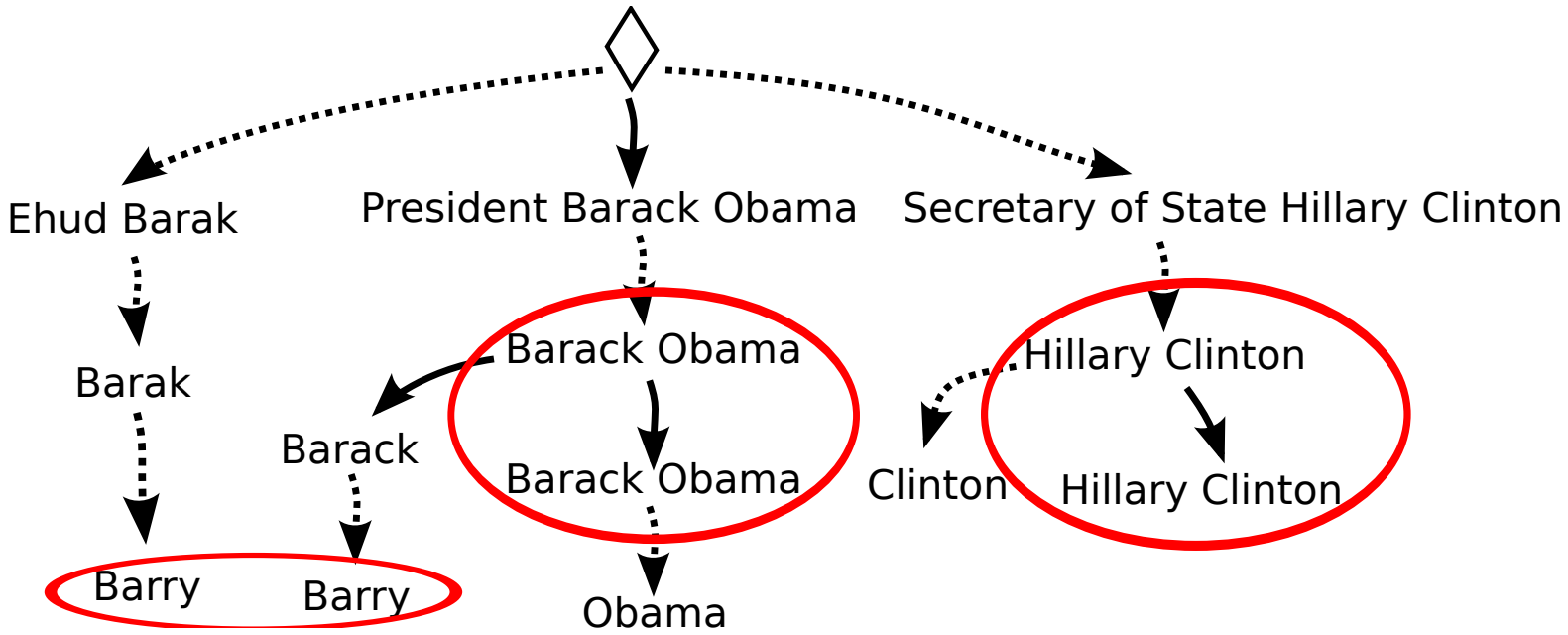


Observed vs unobserved names



Type phylogeny vs token phylogeny

The generative model is over **tokens** (name mentions)

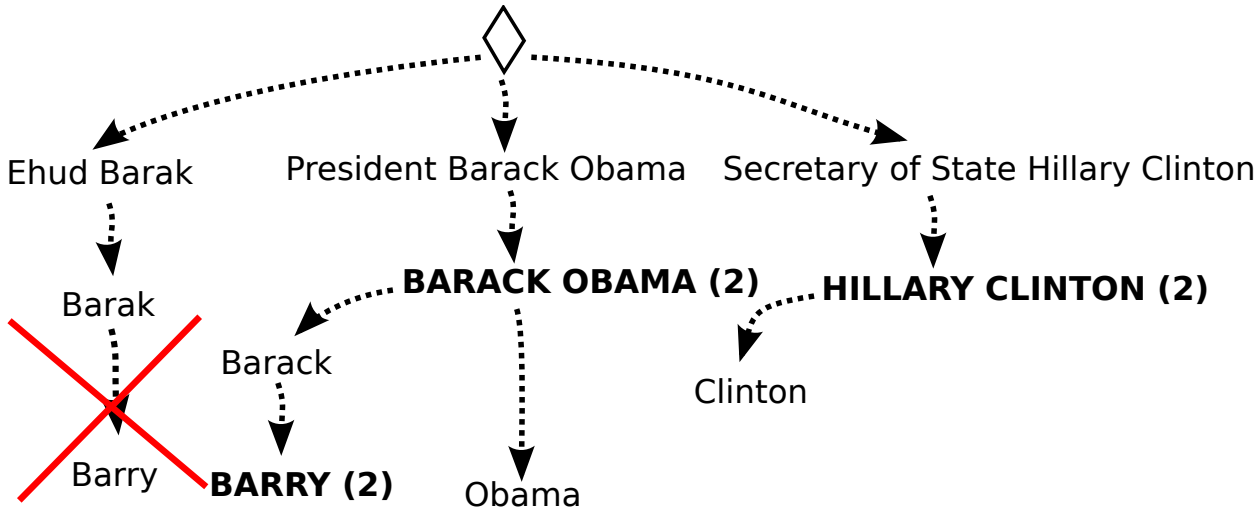


But we do **type-level** inference for the following reasons:

1. Allows faster inference
2. Allows type-level supervision

Type phylogeny vs token phylogeny

We collapse all COPY edges into a single vertex



- The first token in each collapsed vertex is a MUTATION, and the rest are COPIES
- Every edge in the phylogeny now corresponds to a mutation
- Approximation: disallow multiple tokens of the same type to be derived from mutations

Scoring phylogenies

The weight of a single phylogeny is the product of the weight of its edges

$$\prod_{y \in \mathcal{Y}} \delta(y \mid \mathsf{pa}(y))$$

What should the edge weights be?

Edge weights

- NEW NAMES: edges from $\diamondsuit$ to a name x:

$$\delta(x \mid \diamondsuit) = \alpha \cdot p(x \mid \diamondsuit)$$

- MUTATIONS: edges from a name x to a name y:

$$\delta(y \mid x) = \mu \cdot p(y \mid x) \cdot \frac{n_x}{n_y + 1}$$

Approximation: Edges weights are not *quite* edge factored. We are making an approximation of the form

$$\mathbb{E} \prod_y \delta(y \mid \mathrm{pa}(y)) \approx \prod_y \mathbb{E}\delta(y \mid \mathrm{pa})$$

Inference via EM

Iterate until convergence:

1. **E-step:** Given θ, compute a *distribution* over name phylogenies
2. **M-step:** Re-estimate transducer parameters θ given marginal edge probabilities.
   - This step sums over alignments for each (x, y) string pair using forward-backward
   - Each (x, y) pair may be viewed as a training example weighted by the marginal probability of the edge from x to y

E-step: marginalizing over latent variables

The latent variables in the model are:

1. Name phylogeny (spanning tree) relating names as inputs and/or outputs
2. Character alignments from potential input names x to output names y

We use the Matrix-Tree theorem for directed graphs (Tutte, 1984) to efficiently evaluate marginal probabilities:

1. Partition function (sum over phylogenies)
2. Edge marginals

Speed of inference

Two main slowdowns:

- The complexity of the **E-step** is dominated by the $O(n^3)$ (for n names) matrix inversion required to compute the edge marginals c_{xy}.
- The **M-step** sums over alignments for $O(n^2)$ input-output pairs

Approximation: To speed up inference, we prune edges (set $\delta(y \mid x) = 0$) for names with no trigrams in common

Outline

Data preparation

We used English Wikipedia (2011) to create lists of name variants

1. Wikipedia redirects are human-curated pages to resolve common name variants to the correct page (unambiguously)
2. We use Freebase to restrict to redirects for PERSON entities
3. We applied some further filters to remove redirects that were clearly not names (e.g. numbers)
4. We use LDC Gigaword to obtain a frequency for each name variant

Sample Wikipedia redirects

Ho Chi Minh, Ho chi mihn, Ho-Chi Minh, Ho Chih-minh

Guy Fawkes, Guy fawkes, Guy faux, Guy Falks, Guy Faukes, Guy Fawks, Guy foxe, Guy Falkes

Nicholas II of Russia, Nikolai Aleksandrovich Romanov, Nicholas Alexandrovich of Russia, Nicolas II

Bill Gates, Lord Billy, Bill Gates, BillGates, Billy Gates, William Gates III, William H. Gates

William Shakespeare, William shekspere, William shakspeare, Bill Shakespear

Bill Clinton, Billll Clinton, William Jefferson Blythe IV, Bill J. Clinton, William J Clinton

Wikipedia as supervision

We use Wikipedia name lists for **supervision** and **evaluation**

- Treat page redirects as "gold" mutations of the page title:
  Ho Chi Minh $\rightarrow$ Ho chi mihn
  Ho Chi Minh $\rightarrow$ Ho-Chi Minh
  Ho Chi Minh $\rightarrow$ Ho Chih-minh
- Each list of redirects is cluster of names belonging to the same entity
  - No ambiguous names (by construction)

Experiment 1: Transducer log-likelihood

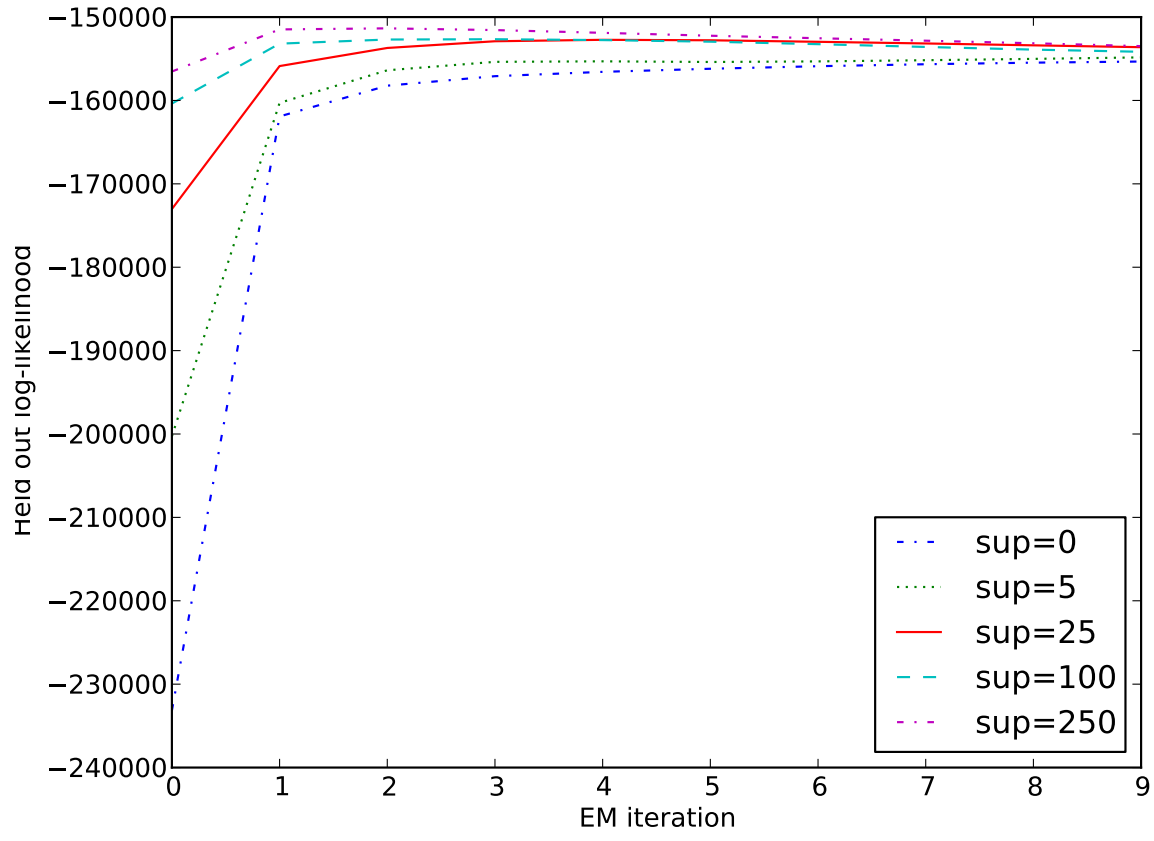
Data:

- 1500 entities (roughly 6000 names) for **train**
- 1500 **different** entities (roughly 6000 names) for **test**

Procedure:

- At train time
  1. Initialize transducer parameters $\boldsymbol{\theta}$ using different amounts of supervision (up to 250 entities)
  2. Run EM for 10 iterations to re-estimate $\boldsymbol{\theta}$
  3. $\alpha = 1.0, \mu = 0.1$
- At test time
  1. Evaluate log-likelihood of the transducer on all "gold" pairs from the test set

Experiment 1: Mutation model log-likelihood



,

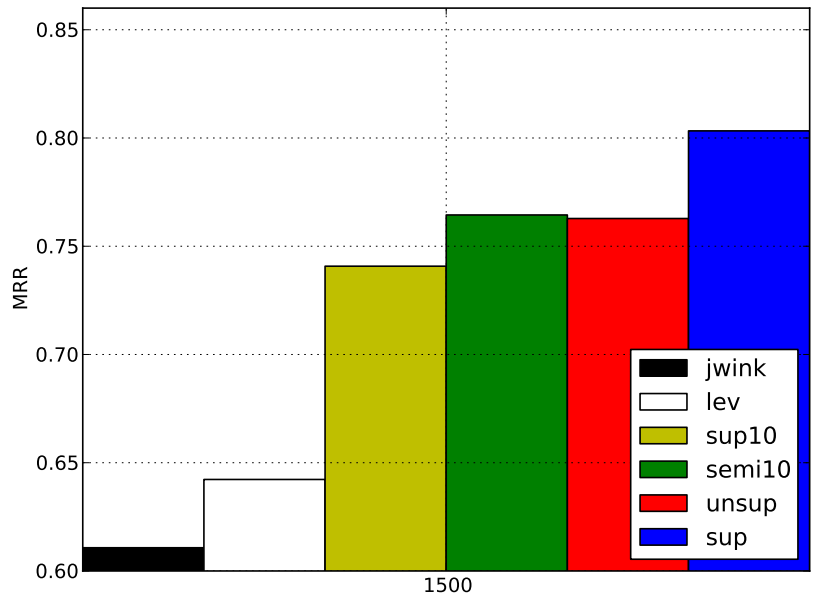
Experiment 2: Ranking

Data: same as before

Procedure:

- At train time
  1. Estimate transducer parameters $\boldsymbol{\theta}$
  2. $\alpha = 1.0, \mu = 0.1$
- At test time
  1. For each Wikipedia person page in the test set, produce a ranking of all test aliases
  2. Compute mean reciprocal rank (MRR) over all such rankings

Experiment 2: Ranking



- For each article name in the test corpus, produce a ranking of redirects
- The rankings are evaluated using mean reciprocal rank

Outline

Future Work

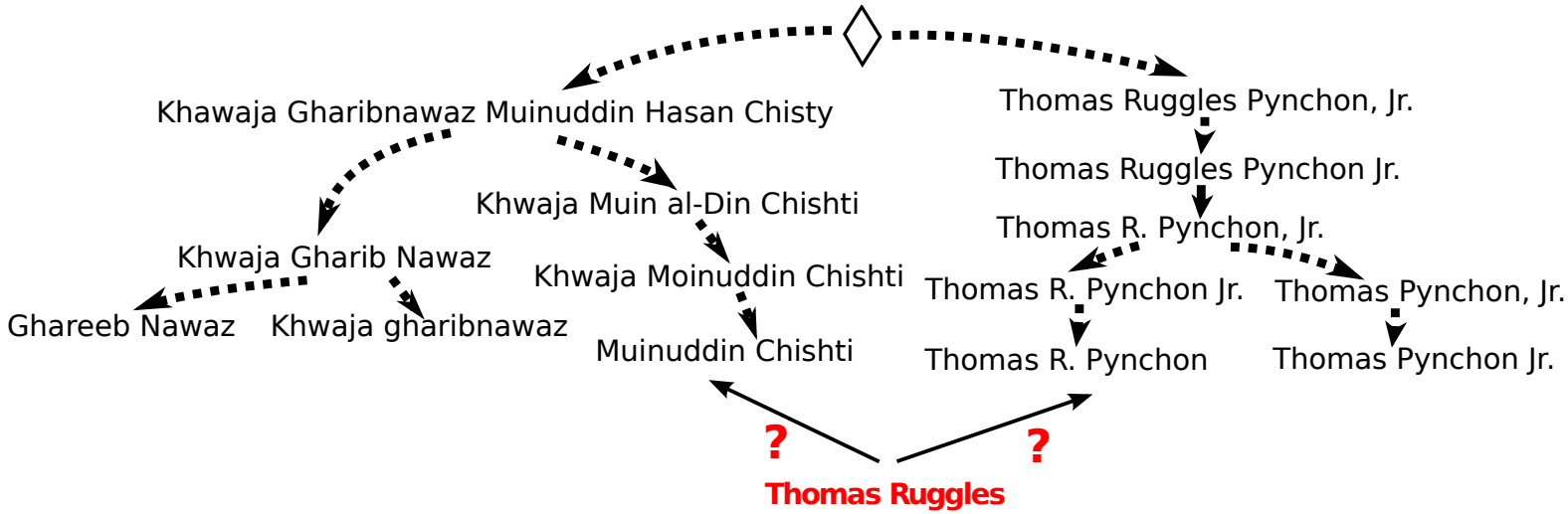
- More sophisticated mutation models
  - Incorporate internal name structure
- Incorporate context in the generative story
- Cross-lingual experiments
  - Each vertex labeled with a language, allowing systematic relationships between languages
- Other potential applications
  - Derivational morphology
  - Paraphrase
  - Transliteration
  - Historical linguistics
  - Bibliographic entry variation

Experiment 3 (preliminary): Precision/Recall

Procedure:

- At train time
  1. Estimate transducer parameters $\boldsymbol{\theta}$ using EM
  2. Find the best spanning tree given $\boldsymbol{\theta}$
- At test time
  1. Attach held-out names to the most likely vertex in the inferred spanning tree
  2. Evaluate precision and recall for the connected component

Experiment 3 (preliminary): Example attachment



- Held-out names can attach to any vertex in the tree
  - Including $\diamondsuit$
- Attachment weights given by edge weights $\delta(y|x)$

Experiment 3 (preliminary): Results