Phylogenetic Inference for Language

Nicholas Andrews, Jason Eisner, Mark Dredze

Department of Computer Science, CLSP, HLTCOE
Johns Hopkins University
Baltimore, Maryland 21218
noa@jhu.edu

April 23, 2013
1. Phylogenetic inference?
2. Generative model
3. A sampler sketch
4. Variational EM
5. Experiments
Language evolution: e.g. sound change\textsuperscript{1}

\[
\begin{align*}
&/d\epsilon ntis/ \\
&\quad \xrightarrow{i \rightarrow \epsilon} /d\epsilon ntes/ \\
&\quad \quad \xrightarrow{\epsilon \rightarrow j \epsilon} /d\epsilon ntes/ \\
&\quad \quad \quad \xrightarrow{s \rightarrow} /d\epsilonnti/ \\
&\quad \quad \quad \quad \text{Spanish} \\
&\quad \quad \quad \quad \quad \text{Latin} \\
&\quad \quad \quad \quad \quad \quad \text{Italian}
\end{align*}
\]

\textsuperscript{1}(Bouchard-Côté et al., 2007)
**Bibliographic entry variation:**


Paraphrase:

Papa ate the caviar
Papa devoured the caviar
Papa ate the caviar with a spoon
The caviar was devoured by papa

Active to passive
substitute "devoured"
add "with a spoon"

Papa devoured the caviar
Papa ate the caviar with a spoon
The caviar was devoured by papa
Phylogenetic inference?

One Entity, Many Names

Qaddafi, Muammar
Al-Gathafi, Muammar
al-Qadhafi, Muammar
Al Qathafi, Mu’ammar
Al Qathafi, Muammar
El Gaddafi, Moamar
El Kadhafi, Moammar
El Kazzaifi, Moamer

معمر محمد عبد السلام أبو منيار القذافي
معمر محمد أبو منيار القذافي
معمر القذافي
أبو محمد

\[\text{Spence et al, NAACL 2012}\]
In each example, there are systematic changes over time:

- **Sound change:** assimilation, metathesis, etc.
- **Bibliographic variation:** typos, abbreviations, punctuation, etc.
- **Paraphrase:** synonyms, voice change, re-arrangements, etc.
- **Name variation:** nicknames, titles, initials, etc.
In each example, there are systematic changes over time:

- **Sound change**: assimilation, metathesis, etc.
- **Bibliographic variation**: typos, abbreviations, punctuation, etc.
- **Paraphrase**: synonyms, voice change, re-arrangements, etc.
- **Name variation**: nicknames, titles, initials, etc.

This talk: name variation
1. Phylogenetic inference?

2. Generative model

3. A sampler sketch

4. Variational EM

5. Experiments
A **phylogeny** is a directed tree rooted at ♦

**Figure:** A cherry-picked fragment of a phylogeny learned by our model.
## Objects in the model

Names are mentioned in context:

![Twitter post](https://pbs.twimg.com/profile_images/860050597791124480/089776b7_58370228_785888267_8.png)

Beliebers held up infinity signs at **Justin's concert** tonight. So beautiful. [pic.twitter.com/qwVWrUJctP](https://twitter.com/)

<table>
<thead>
<tr>
<th>Observed?</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>✓</td>
<td>Name</td>
<td>Justin</td>
</tr>
<tr>
<td>✓</td>
<td>Parent</td>
<td>$x_{13}$</td>
</tr>
<tr>
<td>✓</td>
<td>Entity</td>
<td>$e_{44}$ ( = Justin Bieber)</td>
</tr>
<tr>
<td>✓</td>
<td>Type</td>
<td>PERSON</td>
</tr>
<tr>
<td>✓</td>
<td>Topic</td>
<td>6 ( = MUSIC)</td>
</tr>
<tr>
<td>✓</td>
<td>Document</td>
<td>$d_{20}$</td>
</tr>
<tr>
<td>✓</td>
<td>Language</td>
<td>ENGLISH</td>
</tr>
<tr>
<td>✓</td>
<td>Token position</td>
<td>100</td>
</tr>
<tr>
<td>✓</td>
<td>Index</td>
<td>729</td>
</tr>
</tbody>
</table>
Step 1: Sample a topic $z$ at each position in each document\(^3\) (for all documents in the corpus):

\[
\begin{array}{cccc}
Z_1 & Z_2 & Z_3 & Z_4 & Z_5 & \ldots \\
\end{array}
\]

\(^3\)This is just like latent Dirichlet allocation (LDA).
Step 1: Sample a topic $z$ at each position in each document\(^3\) (for all documents in the corpus):

Step 2: Sample either (1) a context word or (2) a named-entity type at each position, conditioned on the topic:

---

\(^3\)This is just like latent Dirichlet allocation (LDA).
Step 3: For the $n$th named-entity mention $y$, pick a parent $x$:

1. Pick $\diamond$ with probability $\frac{\alpha}{n+\alpha}$

```
\diamond
---
PERSON_n
```
**Step 3:** For the $n$th named-entity mention $y$, pick a parent $x$:

1. Pick ♦ with probability $\frac{\alpha}{n+\alpha}$

   ![Diagram](diamond.png)

   $\downarrow$

   PERSON$_n$

2. Pick a previous mention with probability proportional to $\exp (\phi \cdot f(x, y))$:

   ![Diagram](triangle.png)

   $\downarrow$

   PERSON$_n$

**Features of $x$ and $y$:** topic, entity type, language
Step 4: Generate a name conditioned on the selected parent

1. If the parent is ♦, generate a name from scratch

   ♦

   ↓

   Justin Bieber
**Step 4:** Generate a name conditioned on the selected parent

1. If the parent is ♦, generate a name from scratch

   ♦

   Justin Bieber

2. Otherwise:

   Justin Bieber

   [COPY with probability $1 - \mu$]
**Step 4:** Generate a name conditioned on the selected parent

1. If the parent is ♦, generate a name from scratch
   
   ♦
   
   Justin Bieber

2. Otherwise:
   
   Justin Bieber
   
   Justin Bieber
   
   **COPY** with probability $1 - \mu$

   Justin Bieber
   
   J.B.

   **MUTATE** with probability $\mu$
Name variation as mutations

“Mutations” capture different types of name variation:

1. **Transcription errors:** Barack → barack
2. **Misspellings:** Barack → Barrack
3. **Abbreviations:** Barack Obama → Barack O.
4. **Nicknames:** Barack → Barry
5. **Dropping words:** Barack Obama → 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 $\theta$ determine the probability of being in different regions, and of the different character operations
Example: Mutating a name

Mr. Robert Kennedy

Mr. Bobby Kennedy

Example mutation

Mr. _ Robert _ Kennedy$
Mr. _ [Beginning of edit region

Beginning of edit region
Example: Mutating a name

Mr. Robert Kennedy

Mr. Bobby Kennedy

Example mutation

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

1 substitution operation: (R, B)
Example: Mutating a name

Mr. Robert Kennedy

Example mutation

Mr. Robert Kennedy

2 copy operations: (ε, o), (ε, b)
Example: Mutating a name

Mr. Robert Kennedy

Mr. Bobby Kennedy

Example mutation

Mr. _Robert_ _Kennedy$
Mr. _[Bob$

3 deletion operations: (e,ε), (r,ε), (t, ε)
Example: Mutating a name

Mr. Robert Kennedy

Mr. Bobby Kennedy

Example mutation

Mr. _Robert_ Kennedy
Mr. _Bobby_ Kennedy

2 insertion operations: (ε,b), (ε,y)
Example: Mutating a name

Mr. Robert Kennedy

\[ \text{Mr. Robert Kennedy} \]

Mr. Bobby Kennedy

\[ \text{Mr. Bobby Kennedy} \]

Example mutation

\[ \text{Mr. Robert Kennedy} \]

\[ \text{Mr. Bobby Kennedy} \]

End of edit region
Example: Mutating a name

Mr. Robert Kennedy

Mr. Bobby Kennedy

Example mutation

Mr. _ Robert _ Kennedy $
Mr. _[Bobby]_ Kennedy $
Outline

1. Phylogenetic inference?
2. Generative model
3. A sampler sketch
4. Variational EM
5. Experiments
The latent variables in the model are:

- The spanning tree over tokens \( p \)
- The token permutation \( i \)
- The topics of all named-entity and context tokens \( z \)

Inference requires marginalizing over the latent variables:

\[
\Pr_{\phi, \theta}(x) = \sum_{p, i, z} \Pr_{\phi, \theta}(x, z, i, p)
\]
The latent variables in the model are

- The spanning tree over tokens \( p \)
- The token permutation \( i \)
- The topics of all named-entity and context tokens \( z \)

Inference requires marginalizing over the latent variables:

\[
Pr_{\phi, \theta}(x) = \sum_{p, i, z} Pr_{\phi, \theta}(x, z, i, p)
\]

This sum is intractable to compute 😞
The latent variables in the model are

- The spanning tree over tokens $p$
- The token permutation $i$
- The topics of all named-entity and context tokens $z$

Inference requires marginalizing over the latent variables:

$$
Pr_{\phi, \theta}(x) = \sum_{p, i, z} Pr_{\phi, \theta}(x, z, i, p)
$$

$$
\approx \frac{1}{N} \sum_{n=1}^{N} Pr_{\phi, \theta}(x, z_n, i_n, p_n)
$$

But we can sample from the posterior! 😊
Key idea: sampling \((p, i, z)\) jointly is hard, but sampling from the conditional for each variable is easy(ier)
A block sampler

**Key idea:** sampling \((p, i, z)\) jointly is hard, but sampling from the conditional for each variable is easy(ier)

**Procedure:**

- Initialize \((p, i, z)\).
- For \(n = 1\) to \(N\):
  1. Resample a permutation \(i\) given all other variables.
  2. Resample the topic vector \(z\), similarly.
  3. Resample the phylogeny \(p\), similarly.
  4. Output the current sample \((p, i, z)\).

Steps 1 and 2 are Metropolis-Hastings proposals.
Step 1: Run belief propagation with messages $M_{ij}$ directed from the leaves to the root ♦
**Step 1:** Run belief propagation with messages $M_{ij}$ directed from the leaves to the root.

\[ \diamond \]

\[ M_{yx} \quad y \quad M_{zx} \]

**Step 2:** Sample topics $z$ from $\diamond$ downwards proportional to the belief at each vertex, conditioned on previously sampled topics.
Sampling permutations

(a) Compatible with both \((x, y)\) and \((y, x)\).

(b) Compatible with a single permutation: \((x, y)\).
Each edge between non-root vertices yields a constraint on possible permutations:

**Example**

\[ \Diamond \]

\[ \downarrow \]

\[ x \]

\[ \rightarrow \]

\[ y \]

\[ \rightarrow \]

\[ z \]

yields two constraints: \( x \prec y \) and \( x \prec z \).
Each edge between non-root vertices yields a constraint on possible permutations:

Example

\[
\begin{array}{c}
\diamond \\
\downarrow \\
x \\
\downarrow \\
y & z
\end{array}
\]

yields two constraints: \( x \prec y \) and \( x \prec z \).

Sampling uniformly from the set of permutations respecting these constraints is a simple recursive procedure:
Conditioned on topics and a permutation of the tokens, sample a parent $x$ for each mention $y$ with probability:

$$\propto \underbrace{\Pr_\phi(x, y)}_{\text{affinity model}} \cdot \underbrace{\Pr_\theta(x.n, y.n)}_{\text{transducer model}}$$

No cycles, since the mention permutation $i$ is known.
1. Phylogenetic inference?
2. Generative model
3. A sampler sketch
4. Variational EM
5. Experiments
A simplified model

The sampler is still running 😞
A simplified model

The sampler is still running 😊

We report experiments from our EMNLP 2012 paper + followup experiments, which use a simpler model:

- **No context/topics:** only the transducer parameters $\theta$ need to be estimated
- **Type-level inference and supervision:** vertices in the phylogeny represent distinct name types rather than name tokens
**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 $\theta$
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
Edge weights

- **New names**: edges from ⨾ to a name \( x \):

  \[
  \delta(x \mid ⨾) = \alpha \cdot p(x \mid ⨾)
  \]

- **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

\[
\mathbb{E} \prod_y \delta(y \mid pa(y)) \approx \prod_y \mathbb{E} \delta(y \mid pa)
\]
Inference via EM

Iterate until convergence:

1. **E-step**: Given $\theta$, compute a distribution over name phylogenies.

2. **M-step**: Re-estimate transducer parameters $\theta$ 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
1. Phylogenetic inference?
2. Generative model
3. A sampler sketch
4. Variational EM
5. Experiments
We collected a corpus of *Wikipedia redirect strings* used as examples of names variations

- Filtered down to a subset 77489 people from English Wikipedia (Examples in the next slide!)

The frequency of each variation is estimated using the *Google crosswiki dataset*\(^5\)

- Dictionary of anchor strings linking to English Wikipedia articles
- Collected “by crawling a reasonably large approximation of the entire web”

---

\(^5\)Spitkovsky and Chang, 2012
Example Wikipedia redirects

Ho Chi Minh
Ho chi mihn
Ho-Chi Minh
Ho Chih-minh
<table>
<thead>
<tr>
<th>Name</th>
<th>Redirects</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ho Chi Minh</td>
<td>Ho chi mihn, Ho-Chi Minh, Ho Chih-minh</td>
</tr>
<tr>
<td>Guy Fawkes</td>
<td>Guy fawkes, Guy faux, Guy foxe</td>
</tr>
</tbody>
</table>
Example Wikipedia redirects

<table>
<thead>
<tr>
<th>Name</th>
<th>Alternative Names</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ho Chi Minh</td>
<td>Ho chi mihn, Ho-Chi Minh, Ho Chih-minh</td>
</tr>
<tr>
<td>Guy Fawkes</td>
<td>Guy fawkes, Guy faux, Guy foxe</td>
</tr>
<tr>
<td>Bill Gates</td>
<td>Lord Billy, William Gates III, William H. Gates</td>
</tr>
</tbody>
</table>
Example Wikipedia redirects

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

Guy Fawkes
Guy fawkes
Guy faux
Guy foxe

Bill Gates
Lord Billy
William Gates III
William H. Gates

Billll Clinton
William J. Blythe IV
William Clinton
President Clinton
Type-level supervision is incorporated by tagging vertices with unique IDs and enforcing that they agree from parent to child:
Experiment 1: Evaluating the transducer

Procedure:

- At train time:
  1. Estimate the transducer parameters $\theta$

- At test time:
  1. For each name $x$ in the test set, rank all other names $y$ by the transducer probability

$$
Pr_\theta(y \mid x)
$$

2. Compute the mean reciprocal rank (MRR) over all names
Experiment 1: Evaluating the transducer
Experiment 2: Evaluating the phylogeny

Step 1: Estimate $\theta$ via EM on the training corpus
Step 2: Find the highest scoring tree $^6$

Input: “bag of words.”

$^6 O(m \log n)$ for graphs of $n$ vertices and $m$ edges
Experiment 2: Evaluating the phylogeny

**Step 1:** Estimate $\theta$ via EM on the **training** corpus

**Step 2:** Find the highest scoring tree

---

**Input:** “bag of words.”

**Output:** 1-best tree

---

$O(m \log n)$ for graphs of $n$ vertices and $m$ edges
**Step 3:** Attach each name in the *test* corpus to its most likely parent in the 1-best tree
Experiment 2: Evaluating the phylogeny

**Step 3:** Attach each name in the test corpus to its most likely parent in the 1-best tree

\[ \propto c(\text{William H. Gates}) \cdot \Pr_\theta(\text{Mr. Clinton} \mid \text{William H. Gates}) \]

- **name frequency**
- **transducer probability**

Diagram:
- William H. Gates
- President Bill Clinton
- Guy Fawkes
- Bill Gates
- Lord Billy
- Guido Fawkes
- Mr. Clinton
Step 3: Attach each name in the test corpus to its most likely parent in the 1-best tree.

\[ \propto c(\text{Bill Gates}) \cdot \Pr(\text{Mr. Clinton} \mid \text{Bill Gates}) \]

- William H. Gates
- President Bill Clinton
- Guy Fawkes
- Guido Fawkes
- Bill Gates
- Lord Billy
- Mr. Clinton

name frequency \cdot transducer probability
Experiment 2: Evaluating the phylogeny

Step 3: Attach each name in the test corpus to its most likely parent in the 1-best tree

\[ \alpha \cdot c(\text{President Bill Clinton}) \cdot \Pr_\theta(\text{Mr. Clinton} \mid \text{President Bill Clinton}) \]

- name frequency
- transducer probability

Mr. Clinton

William H. Gates

President Bill Clinton

Guy Fawkes

Lord Billy

Guido Fawkes

Bill Gates
Step 3: Attach each name in the test corpus to its most likely parent in the 1-best tree

\[ \propto c(\text{Lord Billy}) \cdot \Pr_\theta(\text{Mr. Clinton} \mid \text{Lord Billy}) \]

- name frequency
- transducer probability
**Step 3:** Attach each name in the test corpus to its most likely parent in the 1-best tree

![Diagram]

\[ \propto c(\text{Guy Fawkes}) \cdot \Pr_\theta(\text{Mr. Clinton} \mid \text{Guy Fawkes}) \]

- **Name Frequency**
- **Transducer Probability**
Step 3: Attach each name in the test corpus to its most likely parent in the 1-best tree.

\[ \propto c(\text{Guido Fawkes}) \cdot \Pr_\theta(\text{Mr. Clinton} \mid \text{Guido Fawkes}) \]

- Name frequency
- Transducer probability

- Mr. Clinton
**Step 4:** Calculate macro-averaged precision and recall for each test name

- **William H. Gates** → **Bill Gates**
- **President Bill Clinton** → **Lord Billy**
- **Guy Fawkes** → **Guido Fawkes**

Precision = \( \frac{2}{3} \)
Recall = \( \frac{2}{2} \)

Mr. **Clinton**
Experiment 2: Evaluating the phylogeny

**Step 4:** Calculate macro-averaged precision and recall for each test name

- **William H. Gates**
  - Bill Gates

- **President Bill Clinton**
  - Lord Billy

- **Guy Fawkes**
  - Guido Fawkes

Precision = $\frac{1}{3}$
Recall = $\frac{1}{2}$

Mr. Clinton
**Step 4:** Calculate macro-averaged precision and recall for each test name

- **Precision** = \( \frac{1}{1} \)
- **Recall** = \( \frac{1}{2} \)

Mr. Clinton

Diagram:
- William H. Gates
- President Bill Clinton
- Bill Gates
- Lord Billy
- Guy Fawkes
- Guido Fawkes
We compare to two baselines:

1. **Flat tree**

   - **Flat tree:** depth $\leq 2$

   - **Unrestricted tree**

   - No latent edit regions
   - Only 3 degrees of freedom: the weights of different edit operations
We compare to two baselines:

1. **Flat tree**

   - **Flat tree**: depth ≤ 2

2. **Weak transducer**
   - No latent edit regions
   - Only 3 degrees of freedom: the weights of different edit operations
Comparison to flat tree

Full model vs. flat tree @ 0% supervision

- **Full model**
- **Baseline**

Precision vs. Recall graph showing the performance of the full model compared to the baseline.
Comparison to flat tree

Full model vs. flat tree @ 27% supervision

- Full model
- Baseline

Precision vs. Recall graph showing the performance comparison between the full model and the baseline at 27% supervision.
Comparison to flat tree

Full model vs. flat tree @ 34% supervision

- Full model
- Baseline
Comparison to flat tree

Full model vs. flat tree @ 47% supervision

- **Full model**
- **Baseline**
Comparison to flat tree

Full model vs. flat tree @ 53% supervision

- Full model
- Baseline
Comparison to flat tree

Full model vs. flat tree @ 63% supervision

- Red dots: Full model
- Blue dots: Baseline
Comparison to flat tree

Full model vs. flat tree @ 100% supervision

- **Full model**
- **Baseline**
Comparison to flat tree

Full model vs. flat tree

Recall

Precision

0%
27%
34%
47%
53%
63%
100%

Precision vs. Recall

0.0 0.2 0.4 0.6 0.8 1.0

Recall

0.0
0.2
0.4
0.6
0.8
1.0
Comparison to weak transducer

Full model vs. weak transducer @ 0% supervision

- Full model
- Baseline
Comparison to weak transducer

Full model vs. weak transducer @ 27% supervision

Full model vs. weak transducer @ 27% supervision

Full model
Baseline
Comparison to weak transducer

Full model vs. weak transducer @ 34% supervision

- Full model
- Baseline

Precision vs. Recall graph.
Comparison to weak transducer

Full model vs. weak transducer @ 47% supervision

- Full model
- Baseline
Comparison to weak transducer

Full model vs. weak transducer @ 53% supervision

- Full model
- Baseline
Comparison to weak transducer

Full model vs. weak transducer @ 63% supervision

- **Full model**
- **Baseline**
Comparison to weak transducer

Full model vs. weak transducer @ 100% supervision

- Full model
- Baseline
Comparison to weak transducer

Full model vs. weak transducer

Recall
Precision
0%
0%
27%
27%
34%
34%
47%
47%
53%
53%
63%
63%
100%
100%
The End

Thanks! Questions?