A Non-Parametric Model for the Discovery of Inflectional Paradigms from Plain Text using Graphical Models over Strings

Markus Dreyer
Center for Language and Speech Processing (CLSP)
Human Language Technology Center of Excellence (HLTCOE)
Johns Hopkins University (JHU)

Dissertation Defense, November 5, 2010
Motivation

- Rich morphology

**English text**

```
break
break
jump
jump
break
break

break
```

**German text**

```
brichst
brecht
springst
springe
break
break
breche
breche
brichst
breche
```
Motivation

• **Analyzing** text:
  • lack of generalization
  • data sparseness

• **Generating** text:
  • need to generate correct forms
  • produce correctly inflected text
• In NLP, we often **need** to analyze or generate text in such languages.

• So there is a need for a general **morphology model** that knows **how to inflect words**.

• Since annotations are always expensive, it would be best to **learn from unannotated text**.
Motivation

So how do you inflect a verb?
You look it up in such a table, for example:

<table>
<thead>
<tr>
<th>Inflectional Paradigm</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>infinitive</strong></td>
</tr>
<tr>
<td><strong>treffen</strong></td>
</tr>
<tr>
<td><strong>1st</strong></td>
</tr>
<tr>
<td><strong>treffe</strong></td>
</tr>
<tr>
<td><strong>2nd</strong></td>
</tr>
<tr>
<td><strong>trifft</strong></td>
</tr>
<tr>
<td><strong>3rd</strong></td>
</tr>
<tr>
<td><strong>trifft</strong></td>
</tr>
</tbody>
</table>

But creating such supervised data is expensive.
• This talk is about a comprehensive model for inflectional morphology.

• **Main goal:**
  
  • Given some *unannotated text*, can we *learn how to inflect* the verbs of a language (incl. irregularities and exceptions)?
  
  • Discover the *inflectional paradigms* (tables) of a language, using minimal supervision
Motivation

1. Identify the different lexemes in text

**German text**

<table>
<thead>
<tr>
<th>Tokens</th>
</tr>
</thead>
<tbody>
<tr>
<td>brichst</td>
</tr>
<tr>
<td>brecht</td>
</tr>
<tr>
<td>springst</td>
</tr>
<tr>
<td>brechen</td>
</tr>
<tr>
<td>springe</td>
</tr>
<tr>
<td>breche</td>
</tr>
<tr>
<td>brichst</td>
</tr>
<tr>
<td>breche</td>
</tr>
</tbody>
</table>

**Types**

<table>
<thead>
<tr>
<th>Paradigm</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
<tr>
<td>infinitive</td>
</tr>
<tr>
<td>1st</td>
</tr>
<tr>
<td>2nd</td>
</tr>
<tr>
<td>3rd</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>singular</th>
<th>plural</th>
</tr>
</thead>
<tbody>
<tr>
<td>present</td>
<td>past</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Motivation

1. Identify the different lexemes in text

**German text**

- brichst
- brecht
- springst
- brechen
- springe
- breche
- brichst
- breche

**Paradigm**

<table>
<thead>
<tr>
<th></th>
<th>singular</th>
<th>plural</th>
<th>singular</th>
<th>plural</th>
</tr>
</thead>
<tbody>
<tr>
<td>infinitive</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1st</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2nd</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3rd</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Tokens**

- brichst
- brecht
- springst
- brechen
- springe
- breche
- brichst
- breche

**Types**
1. Identify the different lexemes in text

**German text**

<table>
<thead>
<tr>
<th>Tokens</th>
<th>Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>brichst</td>
<td></td>
</tr>
<tr>
<td>brecht</td>
<td></td>
</tr>
<tr>
<td>springst</td>
<td></td>
</tr>
<tr>
<td>brechen</td>
<td></td>
</tr>
<tr>
<td>springe</td>
<td></td>
</tr>
<tr>
<td>breche</td>
<td></td>
</tr>
<tr>
<td>brichst</td>
<td></td>
</tr>
<tr>
<td>breche</td>
<td></td>
</tr>
</tbody>
</table>

**Paradigm**

<table>
<thead>
<tr>
<th>Paradigm</th>
<th>singular</th>
<th>plural</th>
<th>singular</th>
<th>plural</th>
</tr>
</thead>
<tbody>
<tr>
<td>infinitive</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1st</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2nd</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3rd</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>present</td>
<td></td>
<td></td>
<td></td>
<td>past</td>
</tr>
</tbody>
</table>
Motivation

2. Place each form of a lexeme into its paradigm

**German text**

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>brichst</td>
<td>brecht</td>
<td>brechen</td>
<td>springst</td>
<td>breche</td>
<td>breche</td>
</tr>
</tbody>
</table>

**Paradigm**

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>infinitive</strong></td>
<td>brichen</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>1st</strong></td>
<td>breche</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>2nd</strong></td>
<td>brichst</td>
<td>brecht</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>3rd</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Tokens**

<table>
<thead>
<tr>
<th>singular</th>
<th>plural</th>
</tr>
</thead>
<tbody>
<tr>
<td>present</td>
<td>past</td>
</tr>
</tbody>
</table>

**Types**
2. Sort each lexeme into a paradigm

**German text**

<table>
<thead>
<tr>
<th>Tokens</th>
<th>Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>brichst</td>
<td>infinitive</td>
</tr>
<tr>
<td>breche</td>
<td>1st</td>
</tr>
<tr>
<td>springst</td>
<td>2nd</td>
</tr>
<tr>
<td>brech</td>
<td>3rd</td>
</tr>
<tr>
<td>brechen</td>
<td>singular</td>
</tr>
<tr>
<td>breche</td>
<td>plural</td>
</tr>
</tbody>
</table>

**Paradigm**

<table>
<thead>
<tr>
<th>1st</th>
<th>2nd</th>
<th>3rd</th>
<th>4th</th>
</tr>
</thead>
<tbody>
<tr>
<td>breche</td>
<td>brech</td>
<td>brechst</td>
<td>brecht</td>
</tr>
<tr>
<td>brechen?</td>
<td>brachten?</td>
<td>brachten?</td>
<td>brechts?</td>
</tr>
<tr>
<td>brachte?</td>
<td>brach?</td>
<td>brachst?</td>
<td>brachte?</td>
</tr>
<tr>
<td>brichtest?</td>
<td>bracht?</td>
<td>brach?</td>
<td>brachte?</td>
</tr>
<tr>
<td>bracht?</td>
<td>brachten?</td>
<td>brach?</td>
<td>brachte?</td>
</tr>
<tr>
<td>bracht?</td>
<td>brachten?</td>
<td>brech?</td>
<td>brechte?</td>
</tr>
</tbody>
</table>

present | past
Motivation

2. Sort each lexeme into a paradigm

German text

<table>
<thead>
<tr>
<th>Tokens</th>
<th>Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>brichst</td>
<td>brichst</td>
</tr>
<tr>
<td>brecht</td>
<td>brecht</td>
</tr>
<tr>
<td>springst</td>
<td>springst</td>
</tr>
<tr>
<td>brechen</td>
<td>brechen</td>
</tr>
<tr>
<td>springe</td>
<td>springe</td>
</tr>
<tr>
<td>breche</td>
<td>breche</td>
</tr>
</tbody>
</table>

Paradigm

- **infinitive**
  - 1st
  - 2nd
  - 3rd

- **present**
  - singular
  - plural
- **past**
  - singular
  - plural
## Motivation

2. Sort each lexeme into a paradigm

<table>
<thead>
<tr>
<th>Paradigm</th>
<th>1st</th>
<th>2nd</th>
<th>3rd</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>springe</td>
<td>springst</td>
<td>breche</td>
</tr>
<tr>
<td></td>
<td>brecht</td>
<td>brechen</td>
<td>brechten</td>
</tr>
<tr>
<td></td>
<td>springe</td>
<td>breche</td>
<td>brecht</td>
</tr>
</tbody>
</table>

### Tokens

- brichst
- brecht
- springst
- brechen
- springe
- breche
- brichst
- breche

### Types

<table>
<thead>
<tr>
<th>Paradigm</th>
<th>singular</th>
<th>plural</th>
<th>singular</th>
<th>plural</th>
<th>present</th>
<th>past</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Motivation

2. Sort each lexeme into a paradigm

German text

<table>
<thead>
<tr>
<th>Infinitive</th>
<th>Springen?</th>
<th>Sprengen?</th>
</tr>
</thead>
<tbody>
<tr>
<td>1st Person Singular</td>
<td>springe</td>
<td>?</td>
</tr>
<tr>
<td>1st Person Plural</td>
<td>springst</td>
<td>?</td>
</tr>
<tr>
<td>2nd Person Singular</td>
<td>sprengt?</td>
<td>?</td>
</tr>
<tr>
<td>2nd Person Plural</td>
<td>sprengt?</td>
<td>?</td>
</tr>
<tr>
<td>3rd Person Singular</td>
<td>springen?</td>
<td>?</td>
</tr>
<tr>
<td>3rd Person Plural</td>
<td>springen?</td>
<td>?</td>
</tr>
</tbody>
</table>

Tokens

Types

Present

Past
Motivation

German text

<table>
<thead>
<tr>
<th>Tokens</th>
<th>Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>breichst</td>
<td>brecht</td>
</tr>
<tr>
<td>springst</td>
<td>brechen</td>
</tr>
<tr>
<td>springe</td>
<td>breche</td>
</tr>
</tbody>
</table>

Paradigm

<table>
<thead>
<tr>
<th>Paradigm</th>
<th>1st</th>
<th>2nd</th>
<th>3rd</th>
</tr>
</thead>
<tbody>
<tr>
<td>infinitive</td>
<td>springen?</td>
<td>sprengen?</td>
<td>springten?</td>
</tr>
<tr>
<td>1st</td>
<td>springe</td>
<td>springst</td>
<td>springt?</td>
</tr>
<tr>
<td>2nd</td>
<td>sprengen?</td>
<td>sprengt?</td>
<td>springten?</td>
</tr>
<tr>
<td>3rd</td>
<td>sprengen?</td>
<td>sprengt?</td>
<td>springten?</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>singular</th>
<th>plural</th>
<th>singular</th>
<th>plural</th>
</tr>
</thead>
<tbody>
<tr>
<td>present</td>
<td>past</td>
<td>present</td>
<td>past</td>
</tr>
</tbody>
</table>
Motivation

German text:

- brichst
- brecht
- springst
- brechen
- springe
- breche
- bricht?
- brecht?
- brichen?
- brechen?
- brichte?
- brach?
- brichtest?
- brachst?
- brichte?
- brach?
- brichten?
- brachen?
- brichtet?
- bracht?
- brichten?
- brachen?
- säufen
- saufe
- säufig
- sauft
- säufst
- sauft?
Motivation

• Similar to information extraction tasks:
  • Find information in text,
  • Put it in database,
  • Make deductions,
  • Find more information in text,
  • iterate ...
Motivation

In order to perform this morphological knowledge discovery, we define a **probability distribution** over a **text corpus** and its (hidden) inflectional paradigms:

\[ p(\text{Tokens}, \text{Types}) \]
Overview

1. String pairs

2. Multiple strings (paradigms)

3. Text and paradigms
Overview

1. String pairs
   - Dreyer, Smith & Eisner, 2008

2. Multiple strings (paradigms)
   - Dreyer & Eisner, 2009

3. Text and paradigms
   - Dreyer & Eisner, in prep
Overview

1. **String pairs**

2. **Multiple strings (paradigms)**

3. **Text and paradigms**
String pair problems are common in NLP:

Morphology:

<table>
<thead>
<tr>
<th>infinitive</th>
<th>brechen</th>
</tr>
</thead>
<tbody>
<tr>
<td>1st</td>
<td>breche?</td>
</tr>
</tbody>
</table>
String Pairs

String pair problems are common in NLP:

Transliteration
john hardt - yue han ha te
patrick johnson - pa te li ke yue han xun
frederick william mulley - fu lei de li ke wei lian ma li

Pronunciation
Sternanisöl - /ˈstɛrnəniːsɔ̃l/ 
loophole - /ˈluːp, həʊl/

Morphology:
infinitive | brechen
----------|----------
1st       | breche?

Spelling correction
Honululu - Honolulu
braek - break
• We want to build a **probability model** over string pairs.

• Such a model can produce k-best output, can be **plugged** into bigger models later, etc.

• We would like to make use of **flexible features**, be able to look at **linguistic properties** of the strings,

• and **train** the parameters from data.
String Pairs

\[
\text{Pr}(s_1,s_2) = \frac{1}{Z} F(s_1,s_2)
\]

\[
\text{Pr}(s_2 \mid s_1) = \frac{1}{Z} F(s_1,s_2)
\]

- Function \( F \) evaluates how well the two strings go together.
- It looks at properties ("features") of the string pair and assigns some score.
String Pairs

\[ s_1 = \text{breaking} \]
\[ s_2 = \text{broke} \]
String Pairs

\[ s_1 = \#\text{breaking}\# \]
\[ s_2 = \#\text{broke}\# \]
String Pairs

#breaking#
#broke#
String Pairs

#breaking#
#brεokeεεε#
String Pairs

#breaking#
#broεkeεεε#

#breaking#
#broεkeεεε#

#breaking#
#brεεokeεεε#

#breakεing#
#brεεεokeεεε#

#breakεing#
#broεkeεεε#

...
String Pairs

\[ s_1 = \#\text{breaking}\# \]
\[ s_2 = \#\text{broke}\# \]
String Pairs

\[ F(s_1, s_2) = \]

\[ s_1 = \text{#breaking#} \]
\[ s_2 = \text{#broke#} \]
String Pairs

\[ F(s_1, s_2) = \]

\[ s_1 = \#breaking\#
\]

\[ s_2 = \#broke\# \]
String Pairs

\[ F(s_1, s_2) = \exp \sum_i \theta_i f_i(\text{breaking#}) \]

\[ s_1 = \text{breaking#} \]
\[ s_2 = \text{broke#} \]
String Pairs

\[ F(s_1, s_2) = \]
\[ \exp \sum_i \theta_i f_i(\#breaking\#) \]
\[ + \exp \sum_i \theta_i f_i(\#broke\#) \]
String Pairs

\[
F(s_1, s_2) = \exp \sum_i \theta_i f_i(s_1) \exp \sum_i \theta_i f_i(s_2) + \exp \sum_i \theta_i f_i(s_1) \exp \sum_i \theta_i f_i(s_2) + \exp \sum_i \theta_i f_i(s_1) \exp \sum_i \theta_i f_i(s_2)
\]

\[s_1 = \text{#breaking#}\]
\[s_2 = \text{#broke#}\]
F(s₁,s₂) =

\[\exp \sum_i \theta_i f_i(\#\text{breaking}\#)\]

\[+ \exp \sum_i \theta_i f_i(\#\text{broke}\#)\]

\[+ \exp \sum_i \theta_i f_i(\#\text{breaking}\#)\]

\[+ \exp \sum_i \theta_i f_i(\#\text{broke}\#)\]
String Pairs

\[ F(s_1, s_2) = \]

\[ = \exp \sum_i \theta_i f_i(\text{breaking#}) \]
\[ + \exp \sum_i \theta_i f_i(\text{broke#}) \]
\[ + \exp \sum_i \theta_i f_i(\text{breaking#}) \]
\[ + \exp \sum_i \theta_i f_i(\text{broke#}) \]
\[ + \ldots \]
Pr(s₁, s₂) = \frac{1}{Z} F(s₁, s₂) + \exp \sum_i \theta_i f_i(#breaking#)

F(s₁, s₂) = \exp \sum_i \theta_i f_i(#breaking#)

String Pairs
\[ \exp \sum_i \theta_i f_i(\text{breaking}) \]
\[ \exp \sum_i \theta_i f_i(\text{breaking#}) \]
\[ \exp \sum_i \theta_i f_i \left( \text{breaking} \right) \]
$\exp \sum_{i} \theta_{i} f_{i}(\text{#breaking#})$
\[ \exp \sum_i \theta_i f_i \]
\[ \exp \sum_i \theta_i f_i(\text{breaking}) \]
full window

vowels, consonants
eak
oεk

full window

VVC
VεC

vowels, consonants

???
oεk

target language
full window

vowels, consonants

target language

“collapsed”
full window
vowels, consonants
target language
subst, del, ins, ident

“collapsed”
Also add versions of these features that are backed off to bigrams!
String Pairs

- To compute such feature-based scores for two string variables $S_1$ and $S_2$, we construct a **weighted finite-state transducer** $F$

- It can assign a **score** to any string pair $s_1, s_2$

$$\Pr(s_1, s_2) = \frac{1}{Z} F(s_1, s_2)$$
Background: Finite-state machines

What is a **finite-state acceptor (FSA)**?
An automaton with a finite number of states and arcs. Can be used to assign a score to any **string**.

What is a **finite-state transducer (FST)**?
Same as FSA, but used to assign score to any **string pair** (e.g. evaluating how well they go together).
• Specific kind of **grammar** that describes and scores one or more strings

• **Closure** properties under many useful operations (we will use composition, intersection, projection)

• **Useful** for many tasks in natural language processing
String Pairs

\[ S_1 = \text{brchen} \]
\[ S_2 = \text{bracht} \]
String Pairs

$S_1 = \text{finite-state transducer}$

$S_1 = \text{brechen}$

$S_2 = \text{bracht}$

$S_1 = \text{finite-state transducer}$

$S_2 = \text{bracht}$
String Pairs

String Pair $S_1 = \text{brechen}$

Finite-state transducer

String Pair $S_2 = \text{bracht}$
String Pairs

$S_1 = \text{finite-state transducer}$

arc have weights, determined by their features

$S_2 = \text{finite-state transducer}$
String Pairs

Finite-state transducer

Arcs have weights, determined by their features

Transducer $F$ computes score by looking at all alignments.

$S_1 = \text{brechten}$

$S_2 = \text{brachent}$
String Pairs

Sum over all paths in the finite-state transducer

\[ S_1 = \text{brechen} \]

\[ S_2 = \text{bracht} \]

Transducer F computes score by looking at all alignments

There are arcs in the finite-state transducer with weights determined by their features.

\[ F(S_1, S_2) = 13.26 \]
• The alignment between the string pair is a latent variable.

• We add more latent variables to the model:
  • Change regions
  • Conjugations classes

For details, see my thesis, and Dreyer, Smith & Eisner, 2008
Inflection (on German verbs)

13SIA-13SKE  2PIE-13PKE  2PKE-z  rP-pA

Moses3 (baseline)  FST (this talk)  FST (+latent)

Lemmatization

- Basque
- English
- Irish
- Tagalog

Wicentowski (2002)
This talk

String Pairs

**Transliteration** competition, NEWS 2009

Accuracy on English-to-Russian

- U Alberta: 61.3
- NICT: 60.5
- This talk (basic features): 60.0
- IBM: 54.5
- U Tokyo: 54.5
- UIUC: 50.0

This talk uses basic features.
Presented a novel, well-defined **probability model** over string pairs (or single strings)

**General** enough to model many string-to-string problems in NLP (and neighboring disciplines)

Achieved high-scoring results in **different tasks** (inflection, lemmatization, transliteration) in **multiple languages** (*German, Basque, English, Irish, Tagalog, Russian*)
• **Linguistic properties** and soft constraints can be expressed and learned (*prefer certain vowel/consonant sequences, prefer identities, ...*)

• **Arbitrary-length output** is handled elegantly (eliminates need for limiting structure insertion)

• Much information does not need to be annotated; it is inferred as **hidden variables** (alignments, conjugation classes, regions)
Overview

1. String pairs

2. Multiple strings (paradigms)

3. Text and paradigms
• We’ve seen how to model 2 strings, using feature-based finite-state machines

• But we have bigger goals ...
Multiple Strings

infinitive | brechen
---|---
1st | breche?
## Inflectional paradigms

<table>
<thead>
<tr>
<th>infinitive</th>
<th>brechen</th>
</tr>
</thead>
<tbody>
<tr>
<td>1st</td>
<td>breche?</td>
</tr>
<tr>
<td>2nd</td>
<td>brichst</td>
</tr>
<tr>
<td>singular</td>
<td></td>
</tr>
<tr>
<td>plural</td>
<td></td>
</tr>
<tr>
<td>present</td>
<td></td>
</tr>
<tr>
<td>past</td>
<td></td>
</tr>
</tbody>
</table>
Example applications

Inflectional paradigms
### Inflectional paradigms

<table>
<thead>
<tr>
<th>infinitive</th>
<th>brechen</th>
</tr>
</thead>
<tbody>
<tr>
<td>1st</td>
<td>breche</td>
</tr>
<tr>
<td>2nd</td>
<td>bricht</td>
</tr>
<tr>
<td>3rd</td>
<td>bricht</td>
</tr>
<tr>
<td>singular</td>
<td></td>
</tr>
<tr>
<td>plural</td>
<td></td>
</tr>
<tr>
<td>present</td>
<td></td>
</tr>
<tr>
<td>past</td>
<td></td>
</tr>
</tbody>
</table>
## Inflectional paradigms

<table>
<thead>
<tr>
<th></th>
<th>singular</th>
<th>plural</th>
<th>singular</th>
<th>plural</th>
</tr>
</thead>
<tbody>
<tr>
<td>present</td>
<td></td>
<td></td>
<td>past</td>
<td></td>
</tr>
</tbody>
</table>

**Example applications**

- **predict**
- **brechen**

For example:

<table>
<thead>
<tr>
<th>1st</th>
<th>2nd</th>
<th>3rd</th>
</tr>
</thead>
<tbody>
<tr>
<td>breche</td>
<td>brichst</td>
<td>bricht</td>
</tr>
<tr>
<td>brechen</td>
<td>brecht</td>
<td>brechen</td>
</tr>
<tr>
<td>brach</td>
<td>brachst</td>
<td>bracht</td>
</tr>
<tr>
<td>brach</td>
<td>brach</td>
<td>brachen</td>
</tr>
</tbody>
</table>

**Infinitive**: brechen
## Example applications

### Inflectional paradigms

<table>
<thead>
<tr>
<th></th>
<th>1st</th>
<th>2nd</th>
<th>3rd</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>brechen</td>
<td>brecht</td>
<td>brechen</td>
</tr>
<tr>
<td></td>
<td>brechen</td>
<td>brecht</td>
<td>brechen</td>
</tr>
<tr>
<td>singular</td>
<td>present</td>
<td>singular</td>
<td>present</td>
</tr>
<tr>
<td>plural</td>
<td></td>
<td>plural</td>
<td></td>
</tr>
</tbody>
</table>

The table shows the inflectional paradigms for the verb "brechen". The table includes forms for the present and past tenses in both singular and plural. The arrows indicate the predicted forms based on the infinitive "brechen".
### Example applications

#### Inflectional paradigms

<table>
<thead>
<tr>
<th>1st</th>
<th>2nd</th>
<th>3rd</th>
<th>1st</th>
<th>2nd</th>
<th>3rd</th>
</tr>
</thead>
<tbody>
<tr>
<td>breche</td>
<td>brecht</td>
<td>bricht</td>
<td>brechen</td>
<td>brecht</td>
<td>bracht</td>
</tr>
</tbody>
</table>

- **Infinitive**: brechen
- **Present**: predict
- **Past**: predict

<table>
<thead>
<tr>
<th>Form</th>
<th>Singular</th>
<th>Plural</th>
</tr>
</thead>
<tbody>
<tr>
<td>Present</td>
<td>brechen</td>
<td>brachen</td>
</tr>
<tr>
<td>Past</td>
<td>brach</td>
<td>brach</td>
</tr>
</tbody>
</table>
## Example applications

### Inflectional paradigms

<table>
<thead>
<tr>
<th></th>
<th>singular</th>
<th>plural</th>
<th>singular</th>
<th>plural</th>
</tr>
</thead>
<tbody>
<tr>
<td>present</td>
<td>brechen</td>
<td>brecht</td>
<td>brach</td>
<td>brachen</td>
</tr>
<tr>
<td>past</td>
<td>brechen</td>
<td>brecht</td>
<td>brach</td>
<td>brachen</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>1st</th>
<th>2nd</th>
<th>3rd</th>
</tr>
</thead>
<tbody>
<tr>
<td>infinitive</td>
<td>brechen</td>
<td>brichst</td>
<td>bricht</td>
</tr>
<tr>
<td>predict</td>
<td>predict</td>
<td>predict</td>
<td>predict</td>
</tr>
<tr>
<td>reinforce</td>
<td>reinforce</td>
<td>reinforce</td>
<td>reinforce</td>
</tr>
</tbody>
</table>

#### Example

- Infinitive: `brechen`
- 1st person singular: `breche`
- 2nd person singular: `bricht`
- 3rd person singular: `bricht`
- 1st person plural: `brechen`
- 2nd person plural: `bricht`
- 3rd person plural: `bricht`
Example applications

Transliteration (using phonology)

ice cream

アイスクリーム

English orthography

Japanese orthography

日本人

English phonology

Japanese phonology
2 Example applications

Spelling correction

Misspelling

egg sample

Correct spelling

example

Pronunciation
... and all other tasks where word forms and representations interact:

- Cognate modeling
- Multiple-string alignment
- System combination
Multiple Strings

• Let’s build a general **probability model over multiple strings**

• It extends the string-pair model we saw in the last part.

• We will later be able to use it to learn how to inflect verbs.
\[
\Pr(s_1, s_2) = \frac{1}{Z} \times F_1(s_1, s_2)
\]
Factor Graph:

\[ \Pr(s_1, s_2) = \frac{1}{Z} \times F_1(s_1, s_2) \]

Random variable, ranges over any string
Pr(s₁, s₂) = \frac{1}{Z} x F₁(s₁, s₂)

Factor Graph:

Random variable, ranges over any string

Potential function, can score any string pair

Random variable, ranges over any string
Model. *Factor graph examples*

\[
\Pr(s_1, s_2) = \frac{1}{Z} \times F_1(s_1, s_2)
\]
Model. *Factor graph examples*

Factor Graph:

\[
\text{Pr}(s_1, s_2, s_3) = \frac{1}{Z} \times F_1(s_1, s_2) \times F_2(s_1, s_3)
\]
Model. 

**Factor graph examples**

\[
\Pr(s_1, s_2, s_3, s_4) = \frac{1}{Z} \\
\times F_1(s_1, s_2) \\
\times F_2(s_1, s_3) \\
\times F_3(s_1, s_4)
\]

**Factor Graph:**

- Variables: \(S_1, S_2, S_3, S_4\)
- Factors: \(F_1, F_2, F_3\)
2 Model. Factor graph examples

Factor Graph:

\[
\Pr(s_1, s_2, s_3, s_4) = \frac{1}{Z} \times F_1(s_1, s_2) \times F_2(s_1, s_3) \times F_3(s_1, s_4) \times F_4(s_2, s_3)
\]
Model. *Factor graph examples*

Pr($s_1, s_2, s_3, s_4$) = $\frac{1}{Z}$

$x$ $F_1(s_1, s_2)$

$x$ $F_2(s_1, s_3)$

$x$ $F_3(s_1, s_4)$

$x$ $F_4(s_2, s_3)$

$x$ $F_5(s_3, s_4)$
Model. *Factor graph examples*

**Factor Graph:**

\[
\text{Pr}(s_1, s_2, s_3, s_4) = \frac{1}{Z} \times F_1(s_1, s_2) \times F_2(s_1, s_3) \times F_3(s_1, s_4) \times F_4(s_2, s_3) \times F_5(s_3, s_4) \times F_6(s_2, s_4)
\]
Model. Factor graph examples

Factor Graph:

\[ \Pr(s_1, s_2, s_3, s_4) = \frac{1}{Z} \times F_1(s_1, s_2) \times F_2(s_1, s_3) \times F_3(s_1, s_4) \times F_4(s_2, s_3) \times F_5(s_3, s_4) \times F_6(s_2, s_4) \]

Potential function, can score any string pair
Factor Graph:

Each potential function $F$ is computed by a finite-state transducer.
Factor Graph examples

$$\Pr(s_1, s_2, s_3, s_4) = \frac{1}{Z} \times F_1(s_1, s_2) \times F_2(s_1, s_3) \times F_3(s_1, s_4) \times F_4(s_2, s_3) \times F_5(s_3, s_4) \times F_6(s_2, s_4)$$

A formal description of such a model ...
• It is formally an undirected **graphical model** (a.k.a. Markov Random Field, MRF),
• in which the **variables** are **string-valued**, and the **factors** (potential functions) are **finite-state transducers**.

Dreyer & Eisner, 2009
To model multiple strings and their various interactions, I ...

- use many finite-state transducers,
- have each of them look at a different string pair,
- plug them together into a big network,
- and coordinate them to predict all strings jointly (also: train the transducers jointly).
• Model \(k\) strings with a \(k\)-tape finite-state machine?

• Factored model more powerful:

😊 Encode swaps and other useful models

☹ Encode undecidable models
Model

Comparison with k-tape FSM

- Model k strings with a k-tape finite-state machine?
- \( >26^k \) arcs, intractable!

Multiple-sequence alignment

- Factored model more powerful:
  - ☻ Encode swaps and other useful models
  - ☹ Encode undecidable models
Inference. Overview

Factor Graph:

\[
\begin{align*}
S_1 & \quad F_1 \quad F_2 \quad F_3 \\
F_4 & \quad S_2 \\
S_3 & \quad F_5 \\
S_4 & \quad F_6
\end{align*}
\]
Inference. Overview

- Run Belief Propagation (BP)
Inference. Overview

- Run **Belief Propagation (BP)**
- BP is a message-passing algorithm, a generalization of forward-backward.
• Run **Belief Propagation (BP)**
• BP is a message-passing algorithm, a generalization of forward-backward.
• BP computes marginals
Inference. Overview

- Run **Belief Propagation (BP)**
- BP is a message-passing algorithm, a generalization of forward-backward.
- BP computes marginals

In our version of BP, all messages and beliefs are finite-state machines, which is novel.
Example:

- **Inference. Multiple strings**

![Diagram](image)
Example:

(whole distribution)

0.20 bracht
0.13 brechtet
0.08 brachtet...

Inference. *Multiple strings*
Example:

(whole distribution)

0.20 bracht
0.13 brechtet
0.08 brachtet
...
Example:

```
(whole distribution)
0.20 bracht
0.13 brechtet
0.08 brachtet
...
```

```
0.27 brachen
0.07 brechten
...
```

```
0.09 brach
0.03 brech
0.02 brich
...
```
Example:

Inference. Multiple strings

(whole distribution)

0.20 bracht
0.13 brechtet
0.08 brachtet
...

0.27 brachen
0.07 brechten
...

0.09 brach
0.03 brech
0.02 brich
...

0.23 brachten
0.18 brachen
0.11 brechten
...

0.12 brachen
0.07 brechen
0.01 brichen
...
Inference. Multiple strings

Example:

(whole distribution)

0.20 bracht
0.13 brechtet
0.08 brachtet
...

0.23 brachten
0.18 brachen
0.11 brechten
...

0.27 brachen
0.07 brechten
...

0.09 brach
0.03 brech
0.02 brich
...

0.12 brachen
0.07 brechen
0.01 brichen
...

Inference. *Multiple strings*

Example:

- S1
  - F1: 0.27 brachen
  - F2: 0.07 brechten

- S2
  - F4: 0.23 brachten
  - F5: 0.18 brachen
  - F5: 0.11 brechten

- S3
  - 0.23 brachten
  - 0.18 brachen
  - 0.11 brechten

- S4
  - 0.12 brachen
  - 0.07 brechen
  - 0.01 brichen

(whole distribution)

- F1: 0.20 bracht
- F2: 0.13 brechtet
- F3: 0.08 brachtet

- F4: 0.09 brach
- F5: 0.03 brech
- F5: 0.02 brich
Inference. *Multiple strings*

Example:

Decoding output for $S_3$ (consensus): \textbf{brachen}
Inference. Multiple strings

Example:

- Each message is a **finite-state** acceptor
- **Intersect** all incoming messages
Inference. *Multiple strings*

Example:

Again: Usually, BP just works with simple **lookup tables** as factors and messages, **not** finite-state machines.

For example, message passing in a CRF:
Inference. *Multiple strings*

Example:

For example, message passing in a CRF:

Again: Usually, BP just works with simple **lookup tables** as factors and messages, **not** finite-state machines.

For example, message passing in a CRF:
We can also run **loopy belief propagation** on these finite-state Markov Random Fields (MRFs).

- Just **iterate** the message passing.
- Issues with **intractability**, see my thesis.
Inference. *Multiple strings*

- We can also run **loopy belief propagation** on these finite-state Markov Random Fields (MRFs).
- Just **iterate** the message passing.
- Issues with **intractability**, see my thesis.

\[ S_1 = \text{brechen} \] (observed)
Joint inference can be used to train these models from data.

Training data consists of complete or incomplete tables of forms.

We present a method to induce factor graphs in a data-driven way.

See my thesis for the approach and results.
Presented general, novel joint probability model over multiple strings

Combines NLP techniques (finite-state machines) with machine-learning techniques (graphical models)

Markov Random Field over strings (variables: string-valued, potential functions: finite-state machines)
Conclusions / Contributions

- Novel variant of belief propagation with finite-state messages
- Presented approximations
- Presented novel way of structure induction for string-based models based on edit distance
- Achieved significant improvements through staged joint training of complex factor graphs
Overview

1. String pairs
2. Multiple strings (paradigms)
3. Text and paradigms
We have seen how an inflectional paradigm ("multiple strings") can be modeled by finite-state Markov Random Fields (MRFs).

Now we will build a joint model over inflectional paradigms and a text corpus.

Goal: Learn how to inflect words in the language, using clues from the text corpus.

Pr(s₁, s₂, s₃, s₄)
Why do we want to use a text corpus?

- In **Part 2**: We learn how to inflect words, given some observed paradigms *(complete or incomplete)* that someone created as training data *(expensive supervision)* ☹️

- Here in **Part 3**: We also want to learn how to inflect words, we’ll use a few observed paradigms as well, but mainly learn from plain text *(cheap data)* ☺️
How can a text corpus help?

It can potentially fix erroneous MRF string predictions.

Intuition:
If a spelling predicted by the MRF cannot be found in the corpus, it was probably an incorrect prediction.
Decoding output for $S_3$ (consensus): **brechten**

MRF is making a mistake: **brechten** is nonsense and not found in corpus.

But the 2nd-best form, **brachen**, is frequent. It’s probably correct!
We will make such decisions using **statistical inference**, under a **probability model** that uses the finite-state MRFs, but models the text corpus as well.
What kind of probability model do we want?

- Keep tokens and types separate
- Tokens are in the text corpus
- Types are in the paradigms
- We also model abstract morphological knowledge about how forms are related
• Each paradigm contains the systematically related spellings of a lexeme (modeled by finite-state MRFs).

• Assume an unbounded number of possible lexemes and paradigms (“non-parametric”) in the text.
Text & Paradigms

**Generative story**

Model → **generate** → Data

**Inference (Sampling)**

Model ← **learn** ← Data
Generative story

Model \hspace{0.5cm} \text{generate} \hspace{0.5cm} \text{Data}
To generate from our model:
First, generate the **types** of the language.
Then, use them to generate the corpus **tokens**.
(1) Generate infinitely many lexemes
(1) Generate infinitely many lexemes
(1) Generate infinitely many lexemes
(1) Generate infinitely many lexemes
Generate infinitely many lexemes
Each lexeme has a paradigm with slots for the different inflections.
(2) Each lexeme has a paradigm with slots for the different inflections

- 1st sg
- 1st pl
- 2nd sg
- 2nd pl
- 3rd sg
- 3rd pl

- 1st sg
- 1st pl
- 2nd sg
- 2nd pl
- 3rd sg
- 3rd pl

- 1st sg
- 1st pl
- 2nd sg
- 2nd pl
- 3rd sg
- 3rd pl

0.08
0.02
0.01
(3) Each paradigm has a distribution over slot frequencies.
All paradigms generate their spellings using the same finite-state MRF parameterized by $\theta$. 
(4) All paradigms generate their spellings using the same finite-state MRF parameterized by $\theta$.

```plaintext

<table>
<thead>
<tr>
<th></th>
<th>1st sg</th>
<th>1st pl</th>
</tr>
</thead>
<tbody>
<tr>
<td>brichst</td>
<td>2nd sg</td>
<td>2nd pl</td>
</tr>
<tr>
<td>bricht</td>
<td>3rd sg</td>
<td>3rd pl</td>
</tr>
</tbody>
</table>

breche | brechen
brichst | brecht
bricht | brechen

0.08 0.02 0.01
```

All paradigms generate their spellings using the finite-state MRF parameterized by $\theta$. 

\begin{align*}
\text{breche} & \quad \text{brechen} & 0.08 \\
\text{brichst} & \quad \text{brecht} & 0.02 \\
\text{bricht} & \quad \text{brechen} & 0.01
\end{align*}
(4) All paradigms generate their spellings using the finite-state MRF parameterized by $\theta$. 
(4) All paradigms generate their spellings using the finite-state MRF parameterized by $\theta$.

$\theta$ is a “morphological grammar”

The true $\theta$ for German morphology will know, for example, “from 1st singular to plural, just append the suffix $n$”, or “between 3rd sg and pl, a vowel change is likely”.

![Diagram showing frequency of word forms]
All types of the language have been generated. Now generate the corpus tokens.
(5) Generate the corpus: *POS tags*

<table>
<thead>
<tr>
<th>Adv</th>
<th>V</th>
<th>Prep</th>
<th>V</th>
<th>N</th>
<th>V</th>
</tr>
</thead>
<tbody>
<tr>
<td>breche</td>
<td>brechen</td>
<td>treffe</td>
<td>treff</td>
<td>treffen</td>
<td>springge</td>
</tr>
</tbody>
</table>

| 0.08 | 0.02 | 0.01 |
Generate the corpus: Lexemes
(5) Generate the corpus: Inflections

<table>
<thead>
<tr>
<th>Inflections</th>
<th>Lex</th>
<th>POS</th>
</tr>
</thead>
<tbody>
<tr>
<td>treffe</td>
<td>springe</td>
<td>Prep</td>
</tr>
<tr>
<td>treffen</td>
<td>springen</td>
<td>V</td>
</tr>
<tr>
<td>triffst</td>
<td>springst</td>
<td>N</td>
</tr>
<tr>
<td>trefft</td>
<td>springt</td>
<td>V</td>
</tr>
<tr>
<td>treffen</td>
<td>springen</td>
<td>V</td>
</tr>
</tbody>
</table>

- Adv: breche, brichst, bricht
- V: brechen, treffe, treffst, trifft, treffen
- PPER: 3rd sg, 1st pl, 2nd pl, 1st pl, 2nd sg
- V: 0.08, 0.02, 0.01
Generate the corpus: Look up the spellings
Generative story

Model → generate → Data
3 Text & Paradigms
Inference (Sampling)

Model learn Data
We start with observed corpus tokens and construct the paradigms and estimate all distributions.
Text & Paradigms

- Adv: 3rd sg - brecht
- V: 1st pl - brechen
- PPER: 2nd pl - springt
- V: 1st pl - brechen
- Prep:
- N: 2nd sg - trifftst

POS: Lex, Infl, Spell

- 3rd sg: brecht, 0.08
- 1st pl: brechen, 0.02
- 2nd pl: springt, 0.01
- 1st pl: brechen
- 2nd sg: trifftst
<table>
<thead>
<tr>
<th>Adv</th>
<th>V</th>
<th>PPER</th>
<th>V</th>
<th>N</th>
<th>V</th>
<th>Prep</th>
<th>V</th>
<th>N</th>
<th>V</th>
</tr>
</thead>
<tbody>
<tr>
<td>bricht</td>
<td>brechen</td>
<td>springt</td>
<td>brechen</td>
<td>trifft</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

POS
Lex
Infl
Spell
<table>
<thead>
<tr>
<th>Adv</th>
<th>V</th>
<th>PPER</th>
<th>V</th>
<th>N</th>
<th>V</th>
<th>Prep</th>
<th>V</th>
<th>N</th>
<th>V</th>
<th>POS</th>
<th>Lex</th>
<th>Infl</th>
<th>Spell</th>
</tr>
</thead>
<tbody>
<tr>
<td>bricht</td>
<td>brechen</td>
<td>springt</td>
<td>brechen</td>
<td>triffst</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Minimal supervision: We do also observe a few seed paradigms, from which we can estimate an initial $\theta$, which parameterizes the finite-state MRFs.
Minimal supervision: We do also observe a few seed paradigms, from which we can estimate an initial $\theta$, which parameterizes the finite-state MRFs.

<table>
<thead>
<tr>
<th>Adv</th>
<th>V</th>
<th>PREP</th>
<th>V</th>
<th>N</th>
<th>V</th>
<th>Prep</th>
<th>V</th>
<th>N</th>
<th>V</th>
<th>POS</th>
<th>Lex</th>
<th>Infl</th>
<th>Spell</th>
</tr>
</thead>
<tbody>
<tr>
<td>bricht</td>
<td>brechen</td>
<td>springt</td>
<td>brechen</td>
<td>trifft</td>
<td>POS</td>
<td>Lex</td>
<td>Infl</td>
<td>Spell</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Seed paradigm

<table>
<thead>
<tr>
<th>treffe</th>
<th>treffen</th>
</tr>
</thead>
<tbody>
<tr>
<td>trifft</td>
<td>trefft</td>
</tr>
<tr>
<td>trifft</td>
<td>treffen</td>
</tr>
</tbody>
</table>
### Seed paradigm

<table>
<thead>
<tr>
<th>Adv</th>
<th>V</th>
<th>PPER</th>
<th>V</th>
<th>N</th>
<th>V</th>
<th>Prep</th>
<th>V</th>
<th>N</th>
<th>V</th>
</tr>
</thead>
<tbody>
<tr>
<td>treffe</td>
<td>treffen</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>treffst</td>
<td>trefft</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>trifft</td>
<td>treffen</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Seed paradigm

<table>
<thead>
<tr>
<th>treffe</th>
<th>treffen</th>
</tr>
</thead>
<tbody>
<tr>
<td>trifft</td>
<td>trefft</td>
</tr>
<tr>
<td>trifft</td>
<td>treffen</td>
</tr>
</tbody>
</table>

**Train initial $\theta$ values**

- "$e$ is likely to change into "$i$"
- 3rd sg ends in "t"
- from 3rd sg to 1st pl, change vowel

**“morphological grammar”**

- "$e$ is likely to change into "$i$"
- 3rd sg ends in "t"
- from 3rd sg to 1st pl, change vowel...
### Text & Paradigms

<table>
<thead>
<tr>
<th>Adv</th>
<th>V</th>
<th>PPER</th>
<th>V</th>
<th>N</th>
<th>V</th>
<th>Prep</th>
<th>V</th>
<th>N</th>
<th>V</th>
</tr>
</thead>
<tbody>
<tr>
<td>treffe</td>
<td>treffen</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>triffst</td>
<td>trefft</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>trifft</td>
<td>treffen</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **bricht**
- **brechen**
- **springt**
- **brechen**
- **triffst**
The red lexeme is completely specified and “bricht” does not fit in.
Remember: θ says, 3rd sg ends in “t”
# Text & Paradigms

<table>
<thead>
<tr>
<th>Adv</th>
<th>V</th>
<th>PPER</th>
<th>V</th>
<th>N</th>
<th>V</th>
<th>Prep</th>
<th>V</th>
<th>N</th>
<th>V</th>
<th>POS</th>
<th>Lex</th>
<th>Infl</th>
<th>Spell</th>
</tr>
</thead>
<tbody>
<tr>
<td>treffe</td>
<td>treffen</td>
<td>treffe</td>
<td>treffen</td>
<td>1st sg</td>
<td>1st pl</td>
<td>1st sg</td>
<td>1st pl</td>
<td>2nd sg</td>
<td>2nd pl</td>
<td>3rd sg</td>
<td>3rd pl</td>
<td></td>
<td></td>
</tr>
<tr>
<td>trifft</td>
<td>trefft</td>
<td>trifft</td>
<td>trefft</td>
<td>3rd sg</td>
<td>3rd pl</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Remember: θ says, 3rd sg ends in “t”
Text & Paradigms

3rd sg

<table>
<thead>
<tr>
<th>Adv</th>
<th>V</th>
<th>PPER</th>
<th>V</th>
<th>N</th>
<th>V</th>
<th>Prep</th>
<th>V</th>
<th>N</th>
<th>V</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

bricht

brechen

springt

brechen

triffst

Remember: θ says, 3rd sg ends in “t”
We immediately run finite-state-based belief propagation in this new paradigm.
### Text & Paradigms

<table>
<thead>
<tr>
<th>Adv</th>
<th>V</th>
<th>PPER</th>
<th>V</th>
<th>N</th>
<th>V</th>
<th>Prep</th>
<th>V</th>
<th>N</th>
<th>V</th>
<th>POS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>3rd sg</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>brecht</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>brechen</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>springt</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>brechen</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>triffst</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Lex</th>
<th>Infl</th>
<th>Spell</th>
</tr>
</thead>
<tbody>
<tr>
<td>treffe</td>
<td>treffen</td>
<td></td>
</tr>
<tr>
<td>triffst</td>
<td>trefft</td>
<td></td>
</tr>
<tr>
<td>trifft</td>
<td>treffen</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>W1</th>
<th>W2</th>
<th>W3</th>
<th>W4</th>
<th>W5</th>
</tr>
</thead>
<tbody>
<tr>
<td>brecht</td>
<td>brechen?</td>
<td>brechen?</td>
<td>brechen?</td>
<td>brechen?</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>w1</th>
<th>w2</th>
<th>w3</th>
<th>w4</th>
<th>w5</th>
</tr>
</thead>
<tbody>
<tr>
<td>bricht</td>
<td>springt</td>
<td>brechen</td>
<td>triffst</td>
<td>brechen</td>
</tr>
</tbody>
</table>

bricht? brechen?
Text & Paradigms

<table>
<thead>
<tr>
<th>Adv</th>
<th>V</th>
<th>PPER</th>
<th>V</th>
<th>N</th>
<th>V</th>
<th>Prep</th>
<th>V</th>
<th>N</th>
<th>V</th>
<th>PO</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

3rd sg
- bricht
- \( W_1 \)

3rd pl
- brechen
- \( W_2 \)
- springt
- \( W_3 \)
- brechen
- \( W_4 \)
- trifft
- \( W_5 \)

<table>
<thead>
<tr>
<th>W1</th>
<th>bricht</th>
<th>brechen</th>
<th>( W_2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>W1</td>
<td>bichte?</td>
<td>brichen?</td>
<td></td>
</tr>
<tr>
<td>( W_1 )</td>
<td>brechte?</td>
<td>brechen?</td>
<td></td>
</tr>
<tr>
<td>bricht?</td>
<td>brechst?</td>
<td></td>
<td></td>
</tr>
<tr>
<td>brecht?</td>
<td>brecht?</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Text & Paradigms

Adv  V  PPER  V  N  V  Prep  V  N  V

3rd sg  bricht  W1  
3rd pl  brechen  W2  springt  W3  brechen  W4  triffst  W5

treffe  treffen
triffst  trefft
trifft  treffen

briech?  brechst?
briechen?  brechen?
briche?  brecht?
briechen?  brechen?
brichst?  brecht?
## Text & Paradigms

<table>
<thead>
<tr>
<th>Adv</th>
<th>V</th>
<th>PPER</th>
<th>V</th>
<th>N</th>
<th>V</th>
<th>Prep</th>
<th>V</th>
<th>N</th>
<th>V</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>3rd sg</td>
<td>brecht</td>
<td>3rd pl</td>
<td>brechen</td>
<td>springt</td>
<td>brechen</td>
<td>trifft</td>
<td></td>
<td></td>
</tr>
<tr>
<td>triffst</td>
<td>trefft</td>
<td>bricht</td>
<td>brechen</td>
<td>2nd sg</td>
<td>2nd pl</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>trifft</td>
<td>treffen</td>
<td>3rd sg</td>
<td>3rd pl</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

POS
Lex
Infl
Spell
### Text & Paradigms

<table>
<thead>
<tr>
<th>Adv</th>
<th>V</th>
<th>PPER</th>
<th>V</th>
<th>N</th>
<th>V</th>
<th>Prep</th>
<th>V</th>
<th>N</th>
<th>V</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3rd sg</td>
<td>bricht</td>
<td></td>
<td>3rd pl</td>
<td>brechen</td>
<td></td>
<td>3rd sg</td>
<td>springt</td>
<td>brechen</td>
<td>triffst</td>
</tr>
<tr>
<td>W1</td>
<td></td>
<td></td>
<td>W2</td>
<td></td>
<td></td>
<td>W3</td>
<td></td>
<td>W4</td>
<td>W5</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>treffe</th>
<th>treffen</th>
<th>triffst</th>
<th>trefft</th>
<th>bricht</th>
<th>brechen</th>
<th>springt</th>
</tr>
</thead>
<tbody>
<tr>
<td>1st sg</td>
<td>1st pl</td>
<td>2nd sg</td>
<td>2nd pl</td>
<td>3rd pl</td>
<td></td>
<td>W3</td>
</tr>
<tr>
<td>Adv</td>
<td>V</td>
<td>PPER</td>
<td>V</td>
<td>N</td>
<td>V</td>
<td>Prep</td>
</tr>
<tr>
<td>-----</td>
<td>---</td>
<td>------</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>------</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

3rd sg

- bricht
- brechen
- springt

3rd pl

- brechen

3rd sg

- triffst

Run belief propagation!
Text & Paradigms

<table>
<thead>
<tr>
<th>Adv</th>
<th>V</th>
<th>PPER</th>
<th>V</th>
<th>N</th>
<th>V</th>
<th>Prep</th>
<th>V</th>
<th>N</th>
<th>V</th>
</tr>
</thead>
<tbody>
<tr>
<td>3rd sg</td>
<td>brecht</td>
<td>W1</td>
<td>3rd pl</td>
<td>brechen</td>
<td>W2</td>
<td>3rd sg</td>
<td>springt</td>
<td>W3</td>
<td>brechen</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>treffe</th>
<th>treffen</th>
</tr>
</thead>
<tbody>
<tr>
<td>triffst</td>
<td>trefft</td>
</tr>
<tr>
<td>trifft</td>
<td>treffen</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>V</th>
<th>N</th>
<th>V</th>
<th>POS</th>
<th>Lex</th>
<th>Infl</th>
<th>Spell</th>
</tr>
</thead>
<tbody>
<tr>
<td>treffe</td>
<td>treffen</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>triffst</td>
<td>trefft</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>trifft</td>
<td>treffen</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>V</th>
<th>N</th>
<th>V</th>
</tr>
</thead>
<tbody>
<tr>
<td>brique? breche?</td>
<td>brichen? brechen?</td>
<td></td>
</tr>
<tr>
<td>brichst? brechst?</td>
<td>blicht? brecht?</td>
<td></td>
</tr>
<tr>
<td>springe? sprengse?</td>
<td>springen? sprengen?</td>
<td></td>
</tr>
<tr>
<td>springst? sprengst?</td>
<td>springt? sprengt?</td>
<td></td>
</tr>
</tbody>
</table>
It would fit well in two of the cells:

<table>
<thead>
<tr>
<th>Adv</th>
<th>V</th>
<th>PPER</th>
<th>V</th>
<th>N</th>
<th>V</th>
<th>Prep</th>
<th>V</th>
<th>N</th>
<th>V</th>
<th>POS</th>
<th>Lex</th>
<th>Infl</th>
<th>Spell</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>treffe</td>
<td>treffen</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>triffst</td>
<td>trefft</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>trifft</td>
<td>treffen</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>brecht</td>
<td>brechen</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>brechen?</td>
<td>brechen?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>bricht</td>
<td>W1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>bricht</td>
<td>W1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>brechen</td>
<td>W2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>brechen</td>
<td>W2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>springt</td>
<td>W3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>springt</td>
<td>W3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>brechen</td>
<td>W4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>brechen</td>
<td>W4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>triffst</td>
<td>W5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>triffst</td>
<td>W5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

It would fit well in two of the cells:
Do not have to run BP now, because paradigm spellings are not changed. But frequency estimates change right away.
<table>
<thead>
<tr>
<th>Verb</th>
<th>Person</th>
<th>Infinitive</th>
<th>3rd sg</th>
<th>3rd pl</th>
<th>3rd sg</th>
<th>3rd pl</th>
</tr>
</thead>
<tbody>
<tr>
<td>treffe</td>
<td>1st sg</td>
<td>treffen</td>
<td>bricht</td>
<td>brechen</td>
<td>springt</td>
<td>brechen</td>
</tr>
<tr>
<td>triffst</td>
<td>2nd sg</td>
<td>treffen</td>
<td>bricht</td>
<td>brechen</td>
<td>springt</td>
<td>brechen</td>
</tr>
<tr>
<td>trifft</td>
<td>3rd sg</td>
<td>treffen</td>
<td>bricht</td>
<td>brechen</td>
<td>springt</td>
<td>brechen</td>
</tr>
</tbody>
</table>

- **Treffen**
- **Bricht**
- **Brechen**
- **Sprung**
- **Sprung**
- **Treffe**
- **Springe?**
- **Sprung?**
- **Sprung**
- **Spreng**
- **Spreng**
- **Treffe**
- **Breche?**
- **Brechen**
- **Sprunge?**
- **Sprung**
- **Spreng**

**Lex:**
- **Infl:**
- **Spell:**
- **POS:**
### Text & Paradigms

<table>
<thead>
<tr>
<th>Adv</th>
<th>V</th>
<th>PPER</th>
<th>V</th>
<th>N</th>
<th>V</th>
<th>Prep</th>
<th>V</th>
<th>N</th>
<th>V</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>treffe</td>
<td>treffen</td>
<td>treffe</td>
<td>treffen</td>
<td>treffe</td>
<td>treffen</td>
<td>treffe</td>
<td>treffen</td>
<td>treffe</td>
</tr>
</tbody>
</table>

#### Verb Conjugation:
- **springen** (springt)
- **springt**
- **brechen** (bricht)
- **bricht**
- **treffen** (trifft)
- **trifft**

#### Inflectional Clauses:
- **springst**
- **springt**
- **spricht**
- **spricht**
- **trifft**
- **trifft**

#### Tense:
- **3rd sg**
- **3rd pl**
- **3rd sg**
- **3rd pl**
- **2nd sg**

#### Syntax:
- **Adv**
- **V**
- **PPER**
- **V**
- **N**
- **V**
- **Prep**
- **V**
- **N**
- **V**

#### Inflection:
- **Lex**
- **Infl**
- **Spell**
We will now re-estimate $\theta$, given our new “observations” (samples). This training method is called MCEM.

<table>
<thead>
<tr>
<th>Adv</th>
<th>V</th>
<th>PPER</th>
<th>V</th>
<th>N</th>
<th>V</th>
<th>Prep</th>
<th>V</th>
<th>N</th>
<th>V</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><em><strong>bricht</strong></em></td>
<td></td>
<td><em><strong>brechen</strong></em></td>
<td></td>
<td><em><strong>springt</strong></em></td>
<td></td>
<td><em><strong>brechen</strong></em></td>
<td></td>
<td><em><strong>triffst</strong></em></td>
</tr>
</tbody>
</table>

** Paradigms **

<table>
<thead>
<tr>
<th>Adv</th>
<th>V</th>
<th>Infl</th>
<th>Spell</th>
</tr>
</thead>
<tbody>
<tr>
<td>treffe</td>
<td>treffen</td>
<td></td>
<td></td>
</tr>
<tr>
<td>triffst</td>
<td>trefft</td>
<td></td>
<td></td>
</tr>
<tr>
<td>trifft</td>
<td>treffen</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
We go over the corpus over and over again, **re-analyzing words** in the light of **newly acquired knowledge** about table frequencies, inflection frequencies and the updated “morphological grammar” $\theta$. 

<table>
<thead>
<tr>
<th>Adv</th>
<th>V</th>
<th>PPER</th>
<th>V</th>
<th>N</th>
<th>V</th>
<th>Prep</th>
<th>V</th>
<th>N</th>
<th>V</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>treffe</td>
<td>treffen</td>
<td></td>
<td></td>
<td>triffst</td>
<td>treffen</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>trefft</td>
<td>treffen</td>
<td></td>
<td></td>
<td>trifft</td>
<td>treffen</td>
<td></td>
</tr>
</tbody>
</table>

**Lex**

**Infl**

**Spell**

**POS**
We go over the corpus over and over again, **re-analyzing words** in the light of **newly acquired knowledge** about table frequencies, inflection frequencies and the updated “morphological grammar” θ.
We go over the corpus over and over again, **re-analyzing words** in the light of **newly acquired knowledge** about table frequencies, inflection frequencies and the updated “morphological grammar” $\theta$. 

We go over the corpus over and over again, **re-analyzing words** in the light of **newly acquired knowledge** about table frequencies, inflection frequencies and the updated “morphological grammar” $\theta$. 

<table>
<thead>
<tr>
<th>Adv</th>
<th>V</th>
<th>PPER</th>
<th>V</th>
<th>N</th>
<th>V</th>
<th>Prep</th>
<th>V</th>
<th>N</th>
<th>V</th>
</tr>
</thead>
<tbody>
<tr>
<td>2nd pl</td>
<td>brecht</td>
<td></td>
<td>3rd pl</td>
<td>brechen</td>
<td></td>
<td>3rd sg</td>
<td>springt</td>
<td>brechen</td>
<td>2nd sg</td>
</tr>
<tr>
<td></td>
<td>W1</td>
<td></td>
<td></td>
<td>W2</td>
<td></td>
<td></td>
<td>W3</td>
<td>W4</td>
<td>W5</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>W5</th>
<th>treffen</th>
<th>triffst</th>
<th>treffen</th>
<th>triffst</th>
<th>treffen</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>treffe</th>
<th>treffen</th>
<th>W5</th>
<th>trifft</th>
<th>treffen</th>
<th>W5</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>brique?</th>
<th>breche?</th>
<th>brichen?</th>
<th>brechen?</th>
<th>W1</th>
<th>bricht</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>W1</th>
<th>bricht</th>
<th>W2</th>
<th>brechen</th>
<th>W3</th>
<th>springt</th>
</tr>
</thead>
</table>

|----------|---------|------------|------------|----|---------|

<table>
<thead>
<tr>
<th>W4</th>
<th>brechen</th>
<th>spreng?</th>
<th>sprengen?</th>
<th>W3</th>
<th>springt</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>W4</th>
<th>spreng?</th>
<th>sprengen?</th>
<th>W3</th>
<th>springt</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>W5</th>
<th>treffen</th>
<th>triffst</th>
<th>treffen</th>
<th>triffst</th>
<th>treffen</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>treffe</th>
<th>treffen</th>
<th>W5</th>
<th>trifft</th>
<th>treffen</th>
<th>W5</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>brique?</th>
<th>breche?</th>
<th>brichen?</th>
<th>brechen?</th>
<th>W1</th>
<th>bricht</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>W1</th>
<th>bricht</th>
<th>W2</th>
<th>brechen</th>
<th>W3</th>
<th>springt</th>
</tr>
</thead>
</table>

|----------|---------|------------|------------|----|---------|

<table>
<thead>
<tr>
<th>W4</th>
<th>spreng?</th>
<th>sprengen?</th>
<th>W3</th>
<th>springt</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>W5</th>
<th>treffen</th>
<th>triffst</th>
<th>treffen</th>
<th>triffst</th>
<th>treffen</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>treffe</th>
<th>treffen</th>
<th>W5</th>
<th>trifft</th>
<th>treffen</th>
<th>W5</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>brique?</th>
<th>breche?</th>
<th>brichen?</th>
<th>brechen?</th>
<th>W1</th>
<th>bricht</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>W1</th>
<th>bricht</th>
<th>W2</th>
<th>brechen</th>
<th>W3</th>
<th>springt</th>
</tr>
</thead>
</table>

|----------|---------|------------|------------|----|---------|

<table>
<thead>
<tr>
<th>W4</th>
<th>spreng?</th>
<th>sprengen?</th>
<th>W3</th>
<th>springt</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>W5</th>
<th>treffen</th>
<th>triffst</th>
<th>treffen</th>
<th>triffst</th>
<th>treffen</th>
</tr>
</thead>
</table>

| treffe | treffen | W5 | trifft | treffen | W5 |
Summary of the sampling process:

• Constantly update **frequency estimates** for lexemes and inflections

• Often update the “morphological grammar” $\theta$

• Keep **re-analyzing words** accordingly

• Run **finite-state BP** to fill in missing paradigm cells

• **Important:** Often, BP will produce a regular and some more irregular candidates, one of them is found in the corpus and placed in the cell, so we “learn” it!
Summary of the sampling process:

- Inflections and lexemes at the corpus positions are **sampled**.
- The missing paradigm cells are **marginalized** over.
- The “morphological grammar” \( \theta \) is **maximized**.
- We are using a collapsed **Gibbs** sampler, according to a hierarchical **Chinese Restaurant** Process, with interspersed finite-state **belief propagation** steps.
Sampling Speedup:

• Improve mixing and prevent “lock-in”

• Do not sample word by word

• Instead: Pick a whole lexeme, remove all its current words and perform Gibbs sampling just with those words (for one or more iterations)
Obtaining results for evaluation

• We add many paradigms, in which only the lemma form is given, but the other slots are empty.

• Just keep track of what corpus tokens the sampler places in those empty cells, or what candidates will be suggested from belief propagation.

• To get an answer for particular cell, get its marginal probability distribution at end of each iteration. At the end, get average prob. per spelling and report highest-scoring one
The complete probability model (simplified):

- **Base distribution over inflections (maxent)**
- **Dirichlet process**
- **Base distribution over Lexemes**
- **Lexeme**
- **Word**
- **Paradigm**
- **Distribution over paradigms**
- **“Morphological grammar”**
- **Finite-state transducer**
Experiment:

**Learn German inflectional morphology**

- **Given:**
  - 50 seed paradigms (from CELEX)
  - German corpus of 10 million words (from “WaCKy” corpus)

- **Test:**
  For 5,415 German verbs, predict paradigms with 21 inflections each
Adding a large text corpus significantly improves prediction accuracy.
Many forms are easy (~100% acc.)
Large gains on some forms (irregularities)

In rare cases, the corpus hurts.
• Simplifications:

• The finite-state MRFs use a simple factor graph that just connects the lemma to all other forms, but not the forms among each other

• Information flows from one form to the other through the lemma slot

• Better factor graphs give orthogonal improvements, see Ch. 3
Possible model extensions

• **Adding context:** Take neighboring words into account, so that *1st pl* can be more likely after “we” than after “she”, etc.

• **Adding topic variables:** Useful for deciding that a particular spelling belongs into one lexeme rather than another (singed does not fit into “sing” paradigm because it’s a different topic).
Remaining morphological issues

- Reduplication
- Metathesis
- Consonant doubling
- Circumfixes
- Templatic morphology
- Interaction with phonology
Conclusions / Contributions

- Presented novel, principled approach to **learning inflectional morphology** of a language
- Developed joint **probability model** over **text corpus** and **inflectional paradigms**, which are hidden
- Presented **type-based sampling procedure** that discovers inflectional paradigms from plain text
• Presented novel **generative story** for inflectional morphology, ...

• ... based on **ordinary linguistic notions** (lexemes, inflections, paradigms)

• Clusters corpus words into lexemes and inflections using hierarchical **Dirichlet process**

• Allows **unbounded number of lexemes**

• Handles **nonconcatenative**, irregular morphology
Related Work

String pairs

- Sherif & Kondrak (2007), Hong et al (2009), and others, get 1-best alignment, segment into chunks, and score chunks individually.
- Others get 1-best alignment and train conventional n-gram model (Bisani & Ney (2008), and others).
- In contrast, we sum over all alignments, use features, add latent variables, generate arbitrary-length output, use global normalization.
Related Work

2 Multiple Strings

- Joint models over multiple strings have not been tackled much before

- Exception: *Bouchard-Cote et al (2007)*, who defines a directed graphical model, does not run BP inference and does not use FSTs
No one has modeled structured inflectional paradigms before.

Typically, simple concatenative morphology is assumed (Harris (1955), Chan (2008)), but see Yarowsky and Wicentowski (2002).

Goldsmith (2001) and others extract “suffix paradigms” (lists of verb endings).

In contrast, we extract structured paradigms that seemlessly handle non-concatenative phenomena.
Conclusions / Contributions

• Presented several novel probability models step by step, each smaller one being a factor component in the next bigger one

• Developed a coherent, unified statistical approach to inflectional morphology, which advances the state-of-the-art in computational morphology

• Extracted detailed and structured morphological knowledge from plain text;

• Presented the most ambitious morphological knowledge discovery task and method to date
• All presented models have many further uses in NLP:

  • string-pair models for transliteration, pronunciation modeling, spelling correction, etc.

  • multiple-string models for bioinformatics, historical linguistics, phonology, transliteration, etc.

  • text & paradigms model for text generation, machine translation, etc.
Conclusions / Contributions

• This thesis naturally brings together many different concepts from machine learning, NLP, and linguistics, in various novel ways:
  
• In part 1, we use linguistically inspired features, latent variables, FSTs and dynamic programming.

• In part 2, we combine FSTs with graphical models and belief propagation.

• In part 3, we bring together all of the above with statistical tools like Dirichlet process and collapsed Gibbs sampling to tackle a novel task that people have not been able to tackle before.