Recap

• Have already seen a few different techniques for deriving translation units

• **Heuristic method**: first perform word alignment, and then ‘read off’ translation rules for $p(f \mid e)$

• A bit dicey, so combine with a language model and other features
north wind howls

强北风呼啸

X → 北 / north
X → 北风 / north wind
X → 北 X / north X
X → X风 / X wind
X → 北风呼啸 / strong north wind
X → X风呼啸 / strong X wind

...
Recap

- **Direct method**: given a set of translation rules, learn their weights to maximise $p(e | f)$
  - powerful but complex
  - limited to re-weighting a given SCFG

\[
X \rightarrow \text{北} / \text{north} \\
X \rightarrow \text{北风} / \text{north wind} \\
X \rightarrow \text{北 X} / \text{north X} \\
X \rightarrow \text{X风} / \text{X wind} \\
X \rightarrow \text{北 X 呼啸} / \text{strong north wind} \\
X \rightarrow \text{X风 呼啸} / \text{strong X wind} \\
\ldots
\]

Weighted grammar
Every corner of Singapore is filled with fun.
Heuristic Grammar

Singapore

Every corner is filled with fun.

都 充满 着
Let’s add some Noise...

Every corner of Singapore is filled with fun.
Let’s add some Noise...

Every corner of Singapore is filled with fun.
The $1m Question

Can we learn the translation rules themselves?

... at the same time as learning their weights...

... could then learn translation phenomena beyond the word-level...

Could be tricky, but let’s try...
north wind howls

北 风 呼啸

strong north wind

X → 北 / north
X → 北 风 / north wind
X → 北 X / north X
X → X 风 / X wind
X → 北 风 呼啸 / strong north wind
X → X 风 呼啸 / strong X wind
...

X → 北 / wind
X → 北 风 / wind howls
X → 北 X / wind X
X → X 风 / X howls
X → 北 / howls
X → 北 / wind howls
...

weighted grammar
north wind howls

Observations:
1. the search space is enormous
Observations:

1. The search space is enormous
2. Very few rules are good ones
Containing the Search Space

We could use the word alignments...

... but then we’re back to square one, and won’t be able to model global phenomena.

Generally difficult to frame using dynamic programming, so will need to make some limitations and approximations

... could limit the rule size or use beam search
Maximum Likelihood Estimation

north wind howls

強烈的北風呼嘯

strong north wind

$X \xrightarrow{1/3} \text{北 / north}$

$X \xrightarrow{1/3} \text{风 / wind}$

$X \xrightarrow{1/3} X_1 X_2 \text{呼嘯 / strong } X_1 X_2$

$X \xrightarrow{1/2} \text{北风 / north wind}$

$X \xrightarrow{1/2} X \text{呼嘯 / strong } X$

$X \xrightarrow{1} \text{北风呼嘯 / strong north wind}$

Tuesday, 6 December 2011
Maximum Likelihood Estimation

north wind howls

strong north wind

Prob = $\frac{1}{3} \times \frac{1}{3} \times \frac{1}{3} = \frac{1}{27}$

$X \xrightarrow{1/3} \text{北风} / \text{north wind}$

$X \xrightarrow{1/3} X_1 X_2 \text{ 呼啸} / \text{strong } X_1 X_2$

$X \xrightarrow{1/2} \text{北风} / \text{north wind}$

$X \xrightarrow{1/2} X \text{ 呼啸} / \text{strong } X$

$X \xrightarrow{1} \text{北风 呼啸} / \text{strong north wind}$
Maximum Likelihood Estimation

north wind howls

strong north wind

\[ X \rightarrow \frac{1}{3} \text{ north} \]
\[ X \rightarrow \frac{1}{3} \text{ wind} \]
\[ X \rightarrow \frac{1}{3} \text{ strong X} \]

\[ \text{Prob} = \frac{1}{3} \times \frac{1}{3} \times \frac{1}{3} = \frac{1}{27} \]

\[ X \rightarrow \frac{1}{2} \text{ north} \]
\[ X \rightarrow \frac{1}{2} \text{ strong X} \]

\[ \text{Prob} = \frac{1}{2} \times \frac{1}{2} = \frac{1}{4} \]

\[ X \rightarrow \text{ strong north wind} \]
Maximum Likelihood Estimation

north wind howls

strong north wind

Prob = 1/3 * 1/3 * 1/3

Prob = 1/2 * 1/2

Prob = 1

Tuesday, 6 December 2011
Maximum Likelihood Estimation

north wind howls

北 风 呼啸

strong north wind

MLE chooses the largest possible rule to explain the data with as few a parameters as possible

Prob = 1/3 * 1/3 * 1/3

Prob = 1/2 * 1/2

Prob = 1

north wind howls

X → 北 / north

X → 风 / wind

X → X_1 X_2 呼啸 / strong X_1 X_2

X → 北 风 / = 1/4

X → 风 呼啸 / strong X

Prob = 1/27

Prob = 1/4
MLE Degeneracy

So the MLE’s no good... how do we fix it?

Need to counter its bias

need to encourage small rules

and reuse of rules

X → 風 / wind
X → 北 / north
X → 北風 / north wind
But how?

Time for a brief detour...

Recall maximum likelihood estimation formulation

“Find the parameters that maximise the likelihood of the data”

\[
\arg\max_{\theta} p(X | \theta)
\]
But how?

- But not all parameters should be treated equally
- Some are much more benign than others (e.g., small vs large rules)
- Let’s modify the objective to include this knowledge...

\[
\text{argmax}_\theta p(X|\theta)p(\theta)
\]

- likelihood
- prior over the parameters
What prior?

- For 2 parameters... Beta

bias towards **high** values

bias towards **sparse** values
What prior?

- For 3 or more parameters... Dirichlet

Figures taken from Bishop, Pattern Recognition and Machine Learning, 2006
What prior?

- For 3 or more parameters... Dirichlet

No expectations

Figures taken from Bishop, Pattern Recognition and Machine Learning, 2006
What prior?

- For 3 or more parameters... Dirichlet

Expect uniform values

No expectations

Figures taken from Bishop, Pattern Recognition and Machine Learning, 2006
What prior?

- For 3 or more parameters... Dirichlet

Expect uniform values

No expectations

Expect sparse values

Figures taken from Bishop, Pattern Recognition and Machine Learning, 2006
What prior?

- We use a *Dirichlet Process*, a generalisation of a Dirichlet prior
- Encodes biases for small rules and sparse distributions
What prior?

- We use a **Dirichlet Process**, a generalisation of a Dirichlet prior

- Encodes biases for **small rules and sparse distributions**

```
north wind howls
北风呼啸

strong north wind
```

```
X \rightarrow 北 / north
X \rightarrow 风 / wind
X \rightarrow X_1 X_2 呼啸 / strong X_1 X_2
X \rightarrow 北风 / north wind
X \rightarrow X 呼啸 / strong X
X \rightarrow 北风呼啸 / strong north wind
```
What prior?

- We use a Dirichlet Process, a generalisation of a Dirichlet prior
- Encodes biases for small rules and sparse distributions

*north wind howls*

北 风 呼啸

*strong north wind*

X → 北 / north
X → 风 / wind
X → X₁ X₂ 呼啸 / strong X₁ X₂
X → 北 风 / north wind
X → X 呼啸 / strong X
X → 北 风 呼啸 / strong north wind

low penalty
What prior?

• We use a Dirichlet Process, a generalisation of a Dirichlet prior

• Encodes biases for small rules and sparse distributions

north wind howls
北风呼啸

strong north wind

medium penalty

Tuesday, 6 December 2011
What prior?

- We use a Dirichlet Process, a generalisation of a Dirichlet prior
- Encodes biases for small rules and sparse distributions

\[
\begin{align*}
X \rightarrow & \text{北 / north} \quad \text{low penalty} \\
X \rightarrow & \text{风 / wind} \\
X \rightarrow & X_1 X_2 \text{ 呼啸 / strong } X_1 X_2 \\
X \rightarrow & \text{北风 / north wind} \\
X \rightarrow & X \text{ 呼啸 / strong } X \\
X \rightarrow & \text{北风呼啸 / strong north wind}
\end{align*}
\]
What prior?

- We use a Dirichlet Process, a generalisation of a Dirichlet prior.
- Encodes biases for small rules and sparse distributions.

Encourage sparsity: penalty over the number of rules rewriting X.
Good and bad news

- Easy to encode our *prior knowledge*
- Makes inference *intractable*
- Every decision is dependent on all other decisions

When we use rule in deriving one sentence

\[ X \rightarrow \text{北} / \text{north} \]
Good and bad news

• Easy to encode our *prior knowledge*

• Makes inference *intractable*

• Every decision is dependent on all other decisions

When we use rule in deriving one sentence

\[ X \rightarrow \text{北} / \text{north} \]

\[ X \rightarrow \text{北} / \text{north} \]

... increases the probability of using it elsewhere
Tractability

- Ok, so we now have a problem
  - with intractably large number of translation rules
  - and an intractable model
- Solution
  - but can use approximate inference
  - specifically, we use Gibbs sampling
Gibbs Sampling

- Start with some initial state (derivations)
- Try making a minor local change
- Work out the probability of the old and new states, $p(\text{old})$, $p(\text{new})$
- Update to the new state with probability $p(\text{new}) / \{p(\text{new}) + p(\text{old})\}$
- Repeat
- Do this for long enough and provably converges to sample from the model
Gibbs Sampling for SCFGs

Step 1: Initialise with a derivation

Where is the currency exchange office?
Gibbs Sampling for SCFGS

Step 2: Sample updates!

2a: Split or merge preterminal expansions
Gibbs Sampling for SCFGS

Step 2: Sample updates!

2b: Modify branching of tree structure
Gibbs Sampling for SCFGS

Step 2: Sample updates!

2b: Modify branching of tree structure

Reachability requirement:
Must be able to move between any pair of valid derivations through sampling steps.
Gibbs sampling for STSGs

Step 1: Initialise derivation

Every corner of Singapore is filled with fun.
Gibbs sampling for STSGs

Step 2: Sample updates
Gibbs sampling for STSGs

Sample local updates to the alignment

And local updates to reordering
So, does it work?

- IWSLT 05
  - Heuristic: 47
  - Bayes SCFG: 49

- NIST ('03,'04,'05)
  - Heuristic: 27.5
  - Bayes SCFG: 28.0

Tuesday, 6 December 2011
Why does it work?

Learns smaller grammars, with better implied word alignments

(a) Heuristic

(b) Gibbs
How about the STSG?

- Mild improvement (Zh-En, MT’03 < 20)

<table>
<thead>
<tr>
<th>Model</th>
<th>BLEU score</th>
</tr>
</thead>
<tbody>
<tr>
<td>GHKM</td>
<td>26.0</td>
</tr>
<tr>
<td>Our model</td>
<td>26.6</td>
</tr>
</tbody>
</table>

- Grammar much sparser
Some Example Rules

**GHKM Grammar**

<table>
<thead>
<tr>
<th>Rule</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>( (NP \text{JJ}_1 \text{NNS}_2), \text{1 的 2} )</td>
<td></td>
</tr>
<tr>
<td>( (NP \text{JJ}_1 \text{NN}_2), \text{1 的 2} )</td>
<td></td>
</tr>
<tr>
<td>( (NP \text{DT}_1 \text{JJ}_2 \text{NN}_3), \text{1 2 的 3} )</td>
<td></td>
</tr>
<tr>
<td>( (NP \text{PRP}$}_1 \text{NN}_2), \text{1 的 2} )</td>
<td></td>
</tr>
<tr>
<td>( (NP \text{NP}_1 \text{PP}_2), \text{2 的 1} )</td>
<td></td>
</tr>
</tbody>
</table>

**Gibbs Grammar**

<table>
<thead>
<tr>
<th>Rule</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>( (NP \text{DT the}) \text{NN}_1), \text{的 1} )</td>
<td></td>
</tr>
<tr>
<td>( (NP (NP (DT the) \text{NN}_1) (PP (IN of) \text{NP}_2)), \text{2 的 1} )</td>
<td></td>
</tr>
<tr>
<td>( (NP \text{DT the}) \text{NN}_1), \text{的} )</td>
<td></td>
</tr>
<tr>
<td>( (NP (NP (DT the) \text{JJ}_1 \text{NN}_2) (PP (IN of) \text{NP}_3)), \text{3 的 1 2} )</td>
<td></td>
</tr>
<tr>
<td>( (PP (IN of) \text{NP}_1), \text{1 的} )</td>
<td></td>
</tr>
</tbody>
</table>
Summary

- Heuristic method of grammar extraction ubiquitous
  - good performance
  - but produces massive grammars / FSTs
  - ... with many big (useless) rules
  - ... and can’t model non-word based phenomena

- Unsupervised grammar induction techniques
  - cleaner way of modelling the data
  - addresses many issues using Bayesian prior
  - just need to solve inference problems!