Chart-Based Decoding

Kenneth Heafield

University of Edinburgh

6 September, 2012

Most slides courtesy of Philipp Koehn
Overview of Syntactic Decoding

Input Sentence → Decoding → SCFG Parsing → Search → Output Sentence

- Decoding
- SCFG Parsing
- Search
- Output Sentence
Overview of Syntactic Decoding

- **Parallel Corpus**
- **Translation Model**
- **Input Sentence**
- **Decoding**
- **SCFG Parsing**
- **Search**
- **Monolingual Corpus**
- **Language Model**
- **Output Sentence**
Syntactic Decoding

Inspired by monolingual syntactic chart parsing:

During decoding of the source sentence, a chart with translations for the $O(n^2)$ spans has to be filled.
Syntax Decoding

German input sentence with tree
Purely lexical rule: filling a span with a translation (a constituent)
Purely lexical rule: filling a span with a translation (a constituent)
Syntax Decoding

Purely lexical rule: filling a span with a translation (a constituent)
Syntax Decoding

Complex rule: matching underlying constituent spans, and covering words
Complex rule with reordering
Sie will eine Tasse Kaffee trinken.

she wants a cup of coffee.
Bottom-Up Decoding

- For each span, a stack of (partial) translations is maintained.
- Bottom-up: a higher stack is filled, once underlying stacks are complete.
Chart consists of cells that cover contiguous spans over the input sentence
▶ Each cell contains a set of hypotheses
▶ Hypothesis = translation of span with target-side constituent
Dynamic Programming

Applying rule creates new hypothesis

apply rule:
NP → NP Kaffee ; NP → NP+P coffee
Dynamic Programming

Another hypothesis

apply rule:
NP → eine Tasse NP ; NP → a cup of NP

Both hypotheses are indistinguishable in future search
→ can be recombined
Recombinable States

Recombinable?

- NP: a cup of coffee
- NP: a cup of coffee
- NP: a mug of coffee

Yes, if max. 2-gram language model is used
Recombinable States

Recombinable?

- NP: a cup of coffee
- NP: a cup of coffee
- NP: a mug of coffee

Yes, if max. 2-gram language model is used
Recombinability

Hypotheses have to match in

- span of input words covered
- output constituent label
- first \( n-1 \) output words
  
  not properly scored, since they lack context
- last \( n-1 \) output words
  
  still affect scoring of subsequently added words,
  
  just like in phrase-based decoding

\( n \) is the order of the n-gram language model
When merging hypotheses, internal language model contexts are absorbed.

S
(minister of Germany met with Condoleezza Rice)
the foreign ... ... in Frankfurt

NP
(minister)
the foreign ... ... of Germany

VP
(Condoleezza Rice)
met with ... ... in Frankfurt

relevant history
un-scored words

$\text{p}_{\text{LM}}(\text{met I of Germany})$
$\text{p}_{\text{LM}}(\text{with I Germany met})$
Stack Pruning

- Number of hypotheses in each chart cell explodes
  ⇒ need to discard bad hypotheses
  e.g., keep 100 best only
- Different stacks for different output constituent labels?
- Cost estimates
  - translation model cost known
  - language model cost for internal words known
    → estimates for initial words
  - outside cost estimate?
    (how useful will be a NP covering input words 3–5 later on?)
Naive Algorithm: Blow-ups

- Many subspan sequences
  - for all sequences $s$ of hypotheses and words in span [start,end]
- Many rules
  - for all rules $r$
- Checking if a rule applies not trivial
  - rule $r$ applies to chart sequence $s$
⇒ Unworkable
Solution

- Prefix tree data structure for rules
- Dotted rules
- Cube pruning
Storing Rules

- First concern: do they apply to span?
  → have to match available hypotheses and input words
- Example rule
  \[ \text{NP} \rightarrow x_1 \text{des} x_2 \mid \text{NP}_1 \text{ of the NN}_2 \]
- Check for applicability
  - is there an initial sub-span that with a hypothesis with constituent label \text{NP}?
  - is it followed by a sub-span over the word \text{des}?
  - is it followed by a final sub-span with a hypothesis with label \text{NN}?
- Sequence of relevant information
  \[ \text{NP} \bullet \text{des} \bullet \text{NN} \bullet \text{NP}_1 \text{ of the NN}_2 \]
Rule Applicability Check

Trying to cover a span of six words with given rule

NP • des • NN → NP: NP of the NN

das     Haus     des     Architekten     Frank     Gehry
Rule Applicability Check

First: check for hypotheses with output constituent label \textit{NP}

\textit{NP \textbullet des \textbullet NN \rightarrow NP: NP of the NN}

das \hspace{0.5cm} Haus \hspace{1cm} des \hspace{1cm} Architekten \hspace{1cm} Frank \hspace{0.5cm} Gehry
Rule Applicability Check

Found \textbf{NP} hypothesis in cell, matched first symbol of rule

\textbf{NP} \bullet \textbf{des} \bullet \textbf{NN} \rightarrow \textbf{NP}: \textbf{NP} of the \textbf{NN}

\begin{itemize}
  \item das \quad \textbf{Haus}
  \item des \quad \textbf{Architekten}
  \item Frank \quad \textbf{Gehry}
\end{itemize}
Matched word des, matched second symbol of rule

NP • des • NN → NP: NP of the NN

das Haus des Architekten Frank Gehry
Rule Applicability Check

Found a **NN** hypothesis in cell, matched last symbol of rule

NP • des • NN → NP: NP of the NN

das Haus des Architekten Frank Gehry
Matched entire rule → apply to create a NP hypothesis

NP • des • NN → NP: NP of the NN
Rule Applicability Check

Look up output words to create new hypothesis
(note: there may be many matching underlying \textit{NP} and \textit{NN} hypotheses)

\[ \text{NP} \circ \text{des} \circ \text{NN} \rightarrow \text{NP}: \text{NP of the NN} \]

NP: the house of the architect Frank Gehry

NP: the house
NN: architect Frank Gehry
das Haus des Architekten Frank Gehry
Checking Rules vs. Finding Rules

What we showed:
- given a rule
- check if and how it can be applied

But there are too many rules (millions) to check them all

Instead:
- given the underlying chart cells and input words
- find which rules apply
Prefix Tree for Rules

Highlighted Rules

\[
\begin{align*}
    \text{NP} & \rightarrow \text{NP}_1 \ \text{DET}_2 \ \text{NN}_3 \mid \text{NP}_1 \ \text{IN}_2 \ \text{NN}_3 \\
    \text{NP} & \rightarrow \text{NP}_1 \mid \text{NP}_1 \\
    \text{NP} & \rightarrow \text{NP}_1 \ \text{des} \ \text{NN}_2 \mid \text{NP}_1 \ \text{of} \ \text{the} \ \text{NN}_2 \\
    \text{NP} & \rightarrow \text{NP}_1 \ \text{des} \ \text{NN}_2 \mid \text{NP}_2 \ \text{NP}_1 \\
    \text{NP} & \rightarrow \text{DET}_1 \ \text{NN}_2 \mid \text{DET}_1 \ \text{NN}_2 \\
    \text{NP} & \rightarrow \text{das} \ \text{Haus} \mid \text{the} \ \text{house}
\end{align*}
\]
Dotted Rules: Key Insight

▶ If we can apply a rule like

\[ p \rightarrow A \; B \; C \mid x \]

to a span

▶ Then we could have applied a rule like

\[ q \rightarrow A \; B \mid y \]

to a sub-span with the same starting word

⇒ We can re-use rule lookup by storing \( A \; B \bullet \) (dotted rule)
Finding Applicable Rules in Prefix Tree

das | Haus | des | Architekten | Frank | Gehry
Covering the First Cell

das  |  Haus  |  des  |  Architekten  |  Frank  |  Gehry

das Haus des Architekten Frank Gehry
Looking up Rules in the Prefix Tree
Taking Note of the Dotted Rule

das Haus des Architekten Frank Gehry
Checking if Dotted Rule has Translations

das ❶ DET: the

Haus des Architekten Frank Gehry
Applying the Translation Rules

das ❶

das Haus des Architekten Frank Gehry
Looking up Constituent Label in Prefix Tree

das ❶
DET ❷

das
DET : that
DET : the
das ❶
das Haus des Architekten Frank Gehry
DET : the
DET : that
Add to Span’s List of Dotted Rules

Das Haus des Architekten Frank Gehry
Moving on to the Next Cell

Das Haus des Architekten Frank Gehry
Looking up Rules in the Prefix Tree

das ❶
DET ❷
Haus ❸

Das Haus des Architekten Frank Gehry
Taking Note of the Dotted Rule

das Haus des Architekten Frank Gehry: the that house
Checking if Dotted Rule has Translations

- das
- DET
- Haus
- NN: house
- NP: house
- DET: that
- DET: the
- house
- des
- Architekten
- Frank
- Gehry
Applying the Translation Rules

das ❶
DET ❷
Haus ❸

DET ❷
das ❶
das Haus des Architekten Frank Gehry

DET: that
DET: the
DET ❷
das ❶

NP: house
NN: house

house ❸

does ❷
Architekten
Frank
Gehry
Looking up Constituent Label in Prefix Tree

- das ❶
- DET ❷
- Haus ❸
- NN ❹
- NP ❺

- DET ❷
- das ❶
- das Haus des Architekten Frank Gehry
- DET : the
- DET : that
- house ❸
- NP : house
- NP : house
Add to Span’s List of Dotted Rules

Das das Haus des Architekten Frank Gehry

- DET
- DET
- Haus
- NN
- NP
More of the Same

das ❶
DET ❷
Haus ❸
NN ❹
NP ❺

DET: that
DET: the

NP: house
NN: house

IN: of
DET: the

NP: architect
NN: architect

NNP: Frank

NNP: Gehry
Moving on to the Next Cell

das das Haus des Architekten Frank Gehry
Covering a Longer Span

Cannot consume multiple words at once
All rules are extensions of existing dotted rules
Here: only extensions of span over \textit{das} possible
Extensions of Span over das

Das ❶ NN, NP, Haus?

DET ❷ NN, NP, Haus?

Haus ❸ NN, NP, Haus?

NN ❹

NP ❺

Das ❶

Haus ❸

Architekten ❹

Frank ❺

Gehry ❺
Looking up Rules in the Prefix Tree

- das ❶
- Haus ❹
- Haus ❸
- DET ❷
- das ❶

- DET: that
- DET: the
- das ❶
- DET: the
- des•
- IN: of
- NN: house
- NN: architect
- NP: architect
- Architekten•

- NP: house
- house ❸
- NNP: Frank
- NNP: Gehry

- DET ❷
- Haus ❸
- Haus ❹
- NN ❷
- NN ❸
- Architekten•
- Frank•
- Gehry•
Taking Note of the Dotted Rule

das Haus des Architekten Frank Gehry

DET: that
DET: the
DET: the
det NN
Haus

DET: the
Haus

NP: house
NN: house

IN: of
des

NP: architect
Architekten

NN: architect
Frank

NP: Gehry

Frank

Gehry
Checking if Dotted Rules have Translations

- **das**
- **Haus**
- **des**
- **Architekten**
- **Frank**
- **Gehry**
Applying the Translation Rules
Looking up Constituent Label in Prefix Tree

- das ❶
- Haus ❶ NP: the house
- NN ❷ NP: the NN
- DET ❷
- Haus ❸ NP: DET house
- NN ❹ NP: DET NN
- NP ❺

- NP: that house
- NP: the house
- DET NN ❸
- DET Haus ❸
- das NN ❷
- das Haus ❶

- DET ❷
- das ❶
- Haus ❸
- des ❸
- Architekten ❸
- NNP: Frank ❸
- NNP: Gehry ❸

- DET: that
- DET: the
- DET: the
- DET ❷
- das ❶
- NN: house
- NN: house
- NN: architect
- NNP: Frank
- NNP: Gehry
Add to Span’s List of Dotted Rules

das ❶
Haus ❶
DET ❷
Haus ❺
NN ❹
NP: the house
NP: the NN
NP: DET house
NP: DET NN

NP ❺

DET NN ❶
DET Haus ❺
das NN ❹
das Haus ❺

NP: that house
NP: the house

NP: house
NN: house

IN: of
DET: the
des•
NN ❶
NP ❺

NP: architect
NN: architect

NNP: Frank

NNP: Gehry
Even Larger Spans

Extend lists of dotted rules with cell constituent labels

span’s dotted rule list (with same start)
plus neighboring
span’s constituent labels of hypotheses (with same end)
Reflections

- Complexity $O(rn^3)$ with sentence length $n$ and size of dotted rule list $r$
  - may introduce maximum size for spans that do not start at beginning
  - may limit size of dotted rule list (very arbitrary)
- Does the list of dotted rules explode?
- Yes, if there are many rules with neighboring target-side non-terminals
  - such rules apply in many places
  - rules with words are much more restricted
Difficult Rules

- Some rules may apply in too many ways
- Neighboring input non-terminals
  \[ VP \rightarrow \text{gibt } x_1 \ x_2 \ | \ \text{gives } \text{NP}_2 \ \text{to } \text{NP}_1 \]
  - non-terminals may match many different pairs of spans
  - especially a problem for hierarchical models (no constituent label restrictions)
  - may be okay for syntax-models
- Three neighboring input non-terminals
  \[ VP \rightarrow \text{trifft } x_1 \ x_2 \ x_3 \ \text{heute} \ | \ \text{meets } \text{NP}_1 \ \text{today} \ \text{PP}_2 \ \text{PP}_3 \]
  - will get out of hand even for syntax models
Where are we now?

- We know which rules apply
- We know where they apply (each non-terminal tied to a span)
- But there are still many choices
  - many possible translations
  - each non-terminal may match multiple hypotheses
    - number choices exponential with number of non-terminals
Rules with One Non-Terminal

Found applicable rules $PP \rightarrow \text{des } X \mid \ldots \text{NP} \ldots$

- Non-terminal will be filled any of $h$ underlying matching hypotheses
- Choice of $t$ lexical translations

$\Rightarrow$ Complexity $O(ht)$

(note: we may not group rules by target constituent label, so a rule $\text{NP} \rightarrow \text{des } X \mid \text{the NP}$ would also be considered here as well)
Rules with Two Non-Terminals

Found applicable rule $\text{NP} \rightarrow X_1 \text{ des } X_2 | \text{NP}_1 \ldots \text{NP}_2$

- Two non-terminal will be filled any of $h$ underlying matching hypotheses each
- Choice of $t$ lexical translations
  ⇒ Complexity $O(h^2t)$ — a three-dimensional ”cube” of choices

(note: rules may also reorder differently)
Filling a Constituent

\[ X:VP \]
\[ X:V \]
\[ X:NP \]

\[ a \ vu \]

<table>
<thead>
<tr>
<th>Hyp</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>seen</td>
<td>-3.8</td>
</tr>
<tr>
<td>saw</td>
<td>-4.0</td>
</tr>
<tr>
<td>view</td>
<td>-4.0</td>
</tr>
</tbody>
</table>

\[ l\'homme \]

<table>
<thead>
<tr>
<th>Hyp</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>man</td>
<td>-3.6</td>
</tr>
<tr>
<td>the man</td>
<td>-4.3</td>
</tr>
<tr>
<td>some men</td>
<td>-6.3</td>
</tr>
<tr>
<td></td>
<td>man</td>
</tr>
<tr>
<td>-----</td>
<td>--------</td>
</tr>
<tr>
<td>seen</td>
<td>-3.6</td>
</tr>
<tr>
<td>saw</td>
<td>-3.8</td>
</tr>
<tr>
<td>view</td>
<td>-4.0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>man</th>
<th>the man</th>
<th>some men</th>
</tr>
</thead>
<tbody>
<tr>
<td>seen</td>
<td>-8.8</td>
<td>-8.9</td>
<td>-10.8</td>
</tr>
<tr>
<td>saw</td>
<td>-8.3</td>
<td>-8.9</td>
<td>-10.8</td>
</tr>
<tr>
<td>view</td>
<td>-8.5</td>
<td>-8.9</td>
<td>-10.8</td>
</tr>
</tbody>
</table>
Cube Pruning [Chiang, 2007]

man -3.6  the man -4.3  some men -6.3
seen -3.8
saw -4.0
view -4.0

Queue

Hypothesis  Sum
seen man -3.8-3.6=-7.4
Cube Pruning [Chiang, 2007]

- man -3.6
- the man -4.3
- some men -6.3
- seen -3.8
- seen man -8.8
- Queue
- saw -4.0 Queue
- view -4.0

Queue

<table>
<thead>
<tr>
<th>Hypothesis</th>
<th>Sum</th>
</tr>
</thead>
<tbody>
<tr>
<td>saw man</td>
<td>-4.0-3.6=-7.6</td>
</tr>
<tr>
<td>seen the man</td>
<td>-3.8-4.3=-8.1</td>
</tr>
</tbody>
</table>
Cube Pruning [Chiang, 2007]

<table>
<thead>
<tr>
<th></th>
<th>man</th>
<th>the man</th>
<th>some men</th>
<th>seen</th>
<th>seen man</th>
<th>Queue</th>
<th>saw</th>
<th>saw man</th>
<th>Queue</th>
<th>view</th>
<th>Queue</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>-3.6</td>
<td>-4.3</td>
<td>-6.3</td>
<td>-3.8</td>
<td>-8.8</td>
<td></td>
<td>-4.0</td>
<td>-8.3</td>
<td></td>
<td>-4.0</td>
<td></td>
</tr>
</tbody>
</table>

Queue

<table>
<thead>
<tr>
<th>Hypothesis</th>
<th>Sum</th>
</tr>
</thead>
<tbody>
<tr>
<td>view man</td>
<td>-4.0-3.6=-7.6</td>
</tr>
<tr>
<td>seen the man</td>
<td>-3.8-4.3=-8.1</td>
</tr>
<tr>
<td>saw the man</td>
<td>-4.0-4.3=-8.3</td>
</tr>
</tbody>
</table>
Cube Pruning versus Beam Search

Same  Bottom-up with fixed-size beams
Different  Beam filling algorithm
Queue of Cubes

- Several groups of rules will apply to a given span
- Each of them will have a cube
- We can create a queue of cubes
  - Always pop off the most promising hypothesis, regardless of cube

- May have separate queues for different target constituent labels
Bottom-Up Chart Decoding Algorithm

1: for all spans (bottom up) do
2: extend dotted rules
3: for all dotted rules do
4: find group of applicable rules
5: create a cube for it
6: create first hypothesis in cube
7: place cube in queue
8: end for
9: for specified number of pops do
10: pop off best hypothesis of any cube in queue
11: add it to the chart cell
12: create its neighbors
13: end for
14: extend dotted rules over constituent labels
15: end for
Two-Stage Decoding

- First stage: decoding without a language model (-LM decoding)
  - may be done exhaustively
  - eliminate dead ends
  - optionably prune out low scoring hypotheses

- Second stage: add language model
  - limited to packed chart obtained in first stage

- Note: essentially, we do two-stage decoding for each span at a time
Coarse-to-Fine

- Decode with increasingly complex model

- Examples
  - reduced language model [Zhang and Gildea, 2008]
  - reduced set of non-terminals [DeNero et al., 2009]
  - language model on clustered word classes [Petrov et al., 2008]
Outside Cost Estimation

- Which spans should be more emphasized in search?
- Initial decoding stage can provide outside cost estimates

- Use min/max language model costs to obtain admissible heuristic
  (or at least something that will guide search better)
Open Questions

- Where does the best translation fall out the beam?
- Are particular types of rules too quickly discarded?
- Are there systemic problems with cube pruning?
Summary

- Synchronous context free grammars
- Extracting rules from a syntactically parsed parallel corpus
- Bottom-up decoding
- Chart organization: dynamic programming, stacks, pruning
- Prefix tree for rules
- Dotted rules
- Cube pruning