Grouping Language Model Boundary Words to Speed K–Best Extraction from Hypergraphs

Kenneth Heafield, Philipp Koehn, and Alon Lavie
Machine Translation is Slow

5–25 CPU seconds/sentence with target syntax

“Since decoding is very time-intensive...”

[Jehl et al, 2012]
Decoding for Parsing-Based MT

- Input Sentence
- Parsing
- **Search** (New Search Algorithm)
- Output Sentence
Decoding Example: Input

Le garçon a vu l’homme avec un télescope
Decoding Example: Parse with SCFG

S

X:NP

X:VP

X:VP

X:PP

Le garçon

a vu

l’homme

avec un télescope
Decoding Example: Read Target Side

Le garçon: The boy
a vu: seen
l'homme: man
avec un télescope: with the telescope

X:NP

X:VP

X:VP

X:NP

X:PP

S:S

The boy saw the man with a telescope.
Decoding Example: One Constituent

Le garçon
The boy
A boy

a vu
seen
saw
view

l’homme
man
the man
some men

avec un télescope
with the telescope
to an telescope
with a telescope
Hyp
seen
saw
view

Hyp
man
the man
some men
Introduction

String Concatenation

Incremental Expansion

**Hypothesis**

- seen man
- seen the man
- seen some men
- saw man
- saw the man
- saw some men
- view man
- view the man
- view some men
Introduction

String Concatenation

Incremental Expansion

\[ X:VP \]

\[ X:V \]

\[ a\ vu\ ]

HypScore
seen \(-3.8\)
saw \(-4.0\)
view \(-4.0\)

Hyp
man
the man
some men

Score
\(-3.6\)
\(-4.3\)
\(-6.3\)

HypScore
seen man \(-8.8\)
seen the man \(-7.6\)
seen some men \(-9.5\)
saw man \(-8.3\)
saw the man \(-6.9\)
saw some men \(-8.5\)
view man \(-8.5\)
view the man \(-8.9\)
view some men \(-10.8\)

Hypothesis

Score

a vu l’homme

\(-8.8\)
\(-7.6\)
\(-9.5\)
\(-8.3\)
\(-6.9\)
\(-8.5\)
\(-8.5\)
\(-8.9\)
\(-10.8\)
**Introduction**

**String Concatenation**

**Incremental Expansion**

---

- **X**: VP
  - **X**: V
    - *a vu*
      - HypScore: seen = -3.8, saw = -4.0, view = -4.0
    - Hyp: man, the man, some men
    - Score: -3.6, -4.3, -6.3

- **X**: NP
  - *l’homme*
  - Hyp: man, some men
  - Score: -3.6, -6.3

---

**Hypothesis**

- *a vu l’homme*
  - *saw the man* = -6.9
  - *seen the man* = -7.6
  - *saw man* = -8.3
  - *saw some men* = -8.5
  - *view man* = -8.5
  - *seen man* = -8.8
  - *view the man* = -8.9
  - *seen some men* = -9.5
  - *view some men* = -10.8

---
### Introduction

String Concatenation

Incremental Expansion

---

$X: VP$

$X: V$

 menor

$X: NP$

*Hyp Score*

<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>

<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>
</tbody>
</table>

---

$X: VP$

$a$ vu *l’homme*

*Hypothesis* *Score*

<table>
<thead>
<tr>
<th>Hyp</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>saw the man</td>
<td>−6.9</td>
</tr>
<tr>
<td>seen the man</td>
<td>−7.6</td>
</tr>
<tr>
<td>saw man</td>
<td>−8.3</td>
</tr>
<tr>
<td>saw some men</td>
<td>−8.5</td>
</tr>
<tr>
<td>view man</td>
<td>−8.5</td>
</tr>
<tr>
<td>seen man</td>
<td>−8.8</td>
</tr>
<tr>
<td>view the man</td>
<td>−8.9</td>
</tr>
<tr>
<td>seen some men</td>
<td>−9.5</td>
</tr>
<tr>
<td>view some men</td>
<td>−10.8</td>
</tr>
</tbody>
</table>
Introduction

String Concatenation

Incremental Expansion

**X: VP**

- **X: V**
  - *a vu*

**HypScore**

<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>

**X: NP**

- *l'homme*

**Hyp**

- man: -3.6
- the man: -4.3
- some men: -6.3

**Hypothesis**

- *a vu l’homme*

**Score**

- saw the man: -6.9
- seen the man: -7.6
- saw man: -8.3
- saw some men: -8.6
- view man: -8.5
- seen man: -8.8
- view the man: -8.9
- seen some men: -9.5
- view some men: -10.8
Goal

Search for hypotheses faster and more accurately.

Baseline: cube pruning [Chiang, 2007].
Cube Pruning

Overgenerate a fixed number of hypotheses. Prioritize by sum of scores.
Beam Size 5: Finds best option.

<table>
<thead>
<tr>
<th>Option</th>
<th>Sum</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>seen man</td>
<td>-7.4</td>
<td>-8.8</td>
</tr>
<tr>
<td>saw man</td>
<td>-7.6</td>
<td>-8.3</td>
</tr>
<tr>
<td>view man</td>
<td>-7.6</td>
<td>-8.5</td>
</tr>
<tr>
<td>seen the man</td>
<td>-8.1</td>
<td>-7.6</td>
</tr>
<tr>
<td>saw the man</td>
<td>-8.3</td>
<td>-6.9</td>
</tr>
<tr>
<td>view the man</td>
<td>-8.3</td>
<td>-8.9</td>
</tr>
<tr>
<td>seen some men</td>
<td>10.1</td>
<td>-9.5</td>
</tr>
<tr>
<td>saw some men</td>
<td>-10.3</td>
<td>-8.5</td>
</tr>
<tr>
<td>view some men</td>
<td>-10.3</td>
<td>-10.8</td>
</tr>
</tbody>
</table>
**Beam Size 4: Search error.**

<table>
<thead>
<tr>
<th>Option</th>
<th>Sum</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>seen man</td>
<td>-7.4</td>
<td>-8.8</td>
</tr>
<tr>
<td>saw man</td>
<td>-7.6</td>
<td>-8.3</td>
</tr>
<tr>
<td>view man</td>
<td>-7.6</td>
<td>-8.5</td>
</tr>
<tr>
<td>seen the man</td>
<td>-8.1</td>
<td>-7.6</td>
</tr>
<tr>
<td>saw the man</td>
<td>-8.3</td>
<td>-6.9</td>
</tr>
<tr>
<td>view the man</td>
<td>-8.3</td>
<td>-8.9</td>
</tr>
<tr>
<td>seen some men</td>
<td>10.1</td>
<td>-9.5</td>
</tr>
<tr>
<td>saw some men</td>
<td>-10.3</td>
<td>-8.5</td>
</tr>
<tr>
<td>view some men</td>
<td>-10.3</td>
<td>-10.8</td>
</tr>
</tbody>
</table>
Problem With Cube Pruning

Hyp is a countries that
are a countries which
country

Hypothesis

Option

is a countries that
are a countries that
are a countries which :

No notion that “a countries” is bad.
Outline

1. String Concatenation
2. Incremental Expansion
Hypotheses are built by string concatenation. The language model score changes when this is done:

$$\frac{p(\text{saw the man})}{p(\text{saw})p(\text{the man})} = \frac{p(\text{the} | \text{saw})p(\text{man} | \text{saw the})}{p(\text{the})p(\text{man} | \text{the})}$$
Hypotheses are built by string concatenation. The language model score changes when this is done:

\[ c(\text{saw } \bullet \text{ the man}) = \]

\[
\frac{p(\text{saw the man})}{p(\text{saw})p(\text{the man})} = \frac{p(\text{the } | \text{ saw})p(\text{man } | \text{ saw the})}{p(\text{the})p(\text{man } | \text{ the})}
\]
Hypotheses are built by string concatenation. The language model score changes when this is done:

\[
c(saw \bullet \text{the man}) = \]

\[
\frac{p(saw \ \text{the man})}{p(saw)p(\text{the man})} = \frac{p(\text{the} \mid saw)p(\text{man} \mid saw \ \text{the})}{p(\text{the})p(\text{man} \mid \text{the})}
\]

What words does correction \( c \) examine?
Markov Assumption

A 5-gram language model uses up to 4 words of context:
\[ p(\text{man} \mid <s> \text{ the boy saw the}) = p(\text{man} \mid \text{the boy saw the}) \]

\[ \implies \]

Correction \( c \) examines up to 4 words from each string:
\[ c(<s> \leftarrow \text{the boy saw the} \bullet \text{man with a telescope} \rightarrow .) \]

Right State \hspace{1cm} Left State
Markov Assumption

A 5-gram language model uses up to 4 words of context:
\[ p(\text{man} | <s> \text{ the boy saw the}) = p(\text{man} | \text{the boy saw the}) \]

\[ \implies \]

Correction \( c \) examines up to 4 words from each string:
\[ c(<s> \ |- \text{ the boy saw the} \bullet \text{ man with a telescope} \ |- \ ) \]

State may be shorter than 4 words [Li and Khudanpur, 2008]
Partial translations have state. . .

<table>
<thead>
<tr>
<th>Left State</th>
<th>Right State</th>
</tr>
</thead>
<tbody>
<tr>
<td>countries that maintain diplomatic relations with North Korea.</td>
<td>countries that maintain diplomatic ties with North Korea.</td>
</tr>
</tbody>
</table>

. . . so they can concatenate on either side.
Partial translations have state... 

Left State countries that $\not\rightarrow$ maintain diplomatic relations with Right State $\rightarrow$ North Korea. 

...and recombine if states are equal. But what if the states are similar?
Outline

1. String Concatenation
2. Incremental Expansion
### Example Hypotheses

<table>
<thead>
<tr>
<th>Left State</th>
<th>Right State</th>
</tr>
</thead>
<tbody>
<tr>
<td>countries that (\neg) maintain diplomatic ties</td>
<td>(\neg) with North Korea.</td>
</tr>
<tr>
<td>countries that have (\neg) an embassy in</td>
<td>(\neg) DPR Korea.</td>
</tr>
<tr>
<td>country (\neg) that maintains some diplomatic ties</td>
<td>(\neg) in North Korea.</td>
</tr>
<tr>
<td>nations which has (\neg) some diplomatic ties</td>
<td>(\neg) with DPR Korea.</td>
</tr>
<tr>
<td>country (\neg) that maintains some diplomatic ties</td>
<td>(\neg) with DPR Korea.</td>
</tr>
<tr>
<td></td>
<td>⋄ denotes words omitted by state.</td>
</tr>
</tbody>
</table>
Example Hypotheses

<table>
<thead>
<tr>
<th>Left State</th>
<th>Right State</th>
</tr>
</thead>
<tbody>
<tr>
<td>(countries that)</td>
<td>with North Korea.</td>
</tr>
<tr>
<td>(nations which has)</td>
<td>with DPR Korea.</td>
</tr>
<tr>
<td>(countries that have)</td>
<td>DPR Korea.</td>
</tr>
<tr>
<td>(country)</td>
<td>in North Korea.</td>
</tr>
<tr>
<td>(country)</td>
<td>with DPR Korea.</td>
</tr>
</tbody>
</table>

◊ denotes words omitted by state.
High Level Idea of Incremental Expansion

Group hypotheses by common words.
Group by Leftmost Word

\[(\epsilon \diamond \epsilon) \rightarrow \text{countries} \rightarrow \text{nations} \rightarrow \text{country}\]
Reveal Common Words in Each Group

(countries that $\diamond$ Korea .)

$(\epsilon \diamond \epsilon) \rightarrow$ (nations which has $\vdash \diamond \dashv$ with DPR Korea .)

(country $\dashv \diamond$ Korea .)
Alternate Sides Until Tree is Full

\((\epsilon \diamond \epsilon) \rightarrow \) (nations which has \(\vdash \diamond \vdash\) with DPR Korea.)

\((\epsilon \diamond \epsilon) \rightarrow \) (countries that \(\vdash \diamond \vdash\) have DPR Korea.)

\((\epsilon \diamond \epsilon) \rightarrow \) (countries that \(\vdash \diamond \vdash\) with North Korea.)

\((\epsilon \diamond \epsilon) \rightarrow \) (countries that \(\vdash \diamond \vdash\) Korea.)

\((\epsilon \diamond \epsilon) \rightarrow \) (country \(\vdash \diamond \vdash\) in North Korea.)

\((\epsilon \diamond \epsilon) \rightarrow \) (country \(\vdash \diamond \vdash\) with DPR Korea.)
Using Rules

is a $X:NP1$ </s>

turns into

is a $(\epsilon \diamond \epsilon)$ </s>

$X:V1$ the $X:N2$

turns into

$(\epsilon \diamond \epsilon)$ the $(\epsilon \diamond \epsilon)$

$X:V1$ $X:N2$
Exploring and Backtracking

Does the LM like “is a (countries that ◇ Korea .) </s>”?  

Yes  Try more detail.

No  Consider alternatives.
Exploring and Backtracking

Does the LM like “is a (countries that ◊ Korea .) </s>”?

Yes  Try more detail.
No  Consider alternatives.

Formally: priority queue containing breadcrumbs.
Split and Leave Breadcrumbs

(countries that $\vdash \diamond \vdash$ with North Korea.)

(countries that $\diamond$ Korea.)

(nations which has $\vdash \diamond \vdash$ with DPR Korea.)

(countries that have $\vdash \diamond \vdash$ DPR Korea.)

(country $\vdash \diamond \vdash$ in North Korea.)

(country $\vdash \diamond \vdash$ Korea.)

(country $\vdash \diamond \vdash$ with DPR Korea.)
Split and **Leave Breadcrumbs**

\[(\epsilon \diamond \epsilon)^{[1+]} \rightarrow (\text{countries that } \vdash \diamond \dashv \text{ with DPR Korea .})\]

\[(\text{countries that } \vdash \diamond \dashv \text{ Korea .}) \rightarrow (\text{countries that have } \vdash \diamond \dashv \text{ DPR Korea .})\]

\[(\text{countries which has } \vdash \diamond \dashv \text{ with DPR Korea .}) \rightarrow (\text{nations which has } \vdash \diamond \dashv \text{ with DPR Korea .})\]

\[(\text{country } \vdash \diamond \dashv \text{ in North Korea .}) \rightarrow (\text{country } \vdash \diamond \dashv \text{ with DPR Korea .})\]
The queue entry

is a \((\varepsilon \diamond \varepsilon) \) </s>

splits into

Zeroth Child “is a (countries that \diamond \ Korea .) </s>”

Other Children “is a \((\varepsilon \diamond \varepsilon)[1+] \) </s>”

Children except the zeroth.
A priority queue contains competing entries:

is a (countries that ◇ Korea .) </s>
(ε ◇ ε) the (ε ◇ ε)
is a (ε ◇ ε)[1+] </s>

The algorithm pops the top entry, splits a non-terminal, and pushes.
A priority queue contains competing entries:

is a (countries that  Korea .) </s>
(  the (  )
is a (  )[1+] </s>

The algorithm pops the top entry, splits a non-terminal, and pushes.

**Next: Scoring queue entries**
Scores come from the best descendant:

\[
\text{Score}(\epsilon \diamond \epsilon) = \\
\text{Score(} \text{countries that } \vdash \diamond \vdash \text{ with North Korea .)} \\
\geq \\
\text{Score}(\epsilon \diamond \epsilon)[1+] = \\
\text{Score(} \text{national which has } \vdash \diamond \vdash \text{ with DPR Korea .)}
\]
Estimates Update as Words are Revealed:

\[
\text{is a } (\epsilon \diamond \epsilon) \langle /s \rangle \quad \rightarrow \quad \text{is a (countries that } \diamond \text{ Korea .)} \langle /s \rangle
\]

\[
p(\text{is}) \quad \rightarrow \quad p(\text{is})
\]

\[
p(\text{a} \mid \text{is}) \quad \rightarrow \quad p(\text{a} \mid \text{is})
\]

\[
p(\text{countries}) \quad \rightarrow \quad p(\text{countries} \mid \text{is a})
\]

\[
p(\text{that} \mid \text{countries}) \quad \rightarrow \quad p(\text{that} \mid \text{is a countries})
\]

\[
p(\langle /s \rangle) \quad \rightarrow \quad p(\langle /s \rangle \mid \text{Korea .})
\]

Tightly integrated coarse-to-fine [Petrov et al, 2008]
Finding Hypotheses for a Constituent

1. **Initialize:** Push rules onto a priority queue.

2. **Best-First Loop:**
   1. Pop the top entry.
   2. If it’s complete, add to the beam. Otherwise, split and push.

3. **Finalize:** Convert the beam to a tree (lazily).
Finding Hypotheses for a Constituent

1. **Initialize:** Push rules onto a priority queue.

2. **Best-First Loop:**
   1. Pop the top entry.
   2. If it’s complete, add to the beam. Otherwise, split and push.

3. **Finalize:** Convert the beam to a tree (lazily).

Process constituents in bottom-up order (like cube pruning).
Experimental Setup

Task  WMT 2011 German-English
Builder [Koehn et al, 2011]
Model  Hierarchical
Introduction

String Concatenation

Incremental Expansion

cdec Hierarchical

Average model score vs. CPU seconds/sentence

This work
Gesmundo et al 1
Gesmundo et al 2
Cube pruning
Moses Hierarchical

Average model score vs. CPU seconds/sentence

-101.4 to -101.6

- This work
- Additive cube pruning
- Cube pruning
Moses Hierarchical

Uncased BLEU (%)

CPU seconds/sentence

This work
Additive cube pruning
Cube pruning
Now With Target Syntax

Task  WMT 2011 German-English
Builder  [Koehn et al, 2011]
Model  Target Syntax
This work
Cube pruning
1.50–3.50x As Fast
at attaining the same model score (except beam size 5).

http://kheafield.com/code/
- Moses
- cdec
- Library
- Standalone

ACL 2013: fast and scalable modified Kneser-Ney estimation.