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

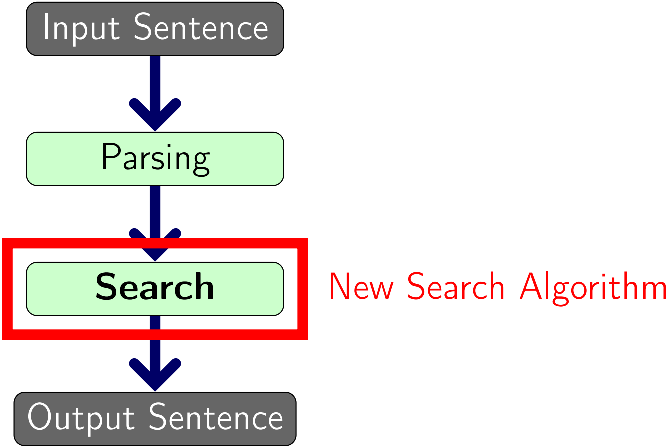
Kenneth Heafield, Philipp Koehn, and Alon Lavie

Machine Translation is Slow

5–25 CPU seconds/sentence with target syntax

"Since decoding is very time-intensive..."
[Jehl et al, 2012]

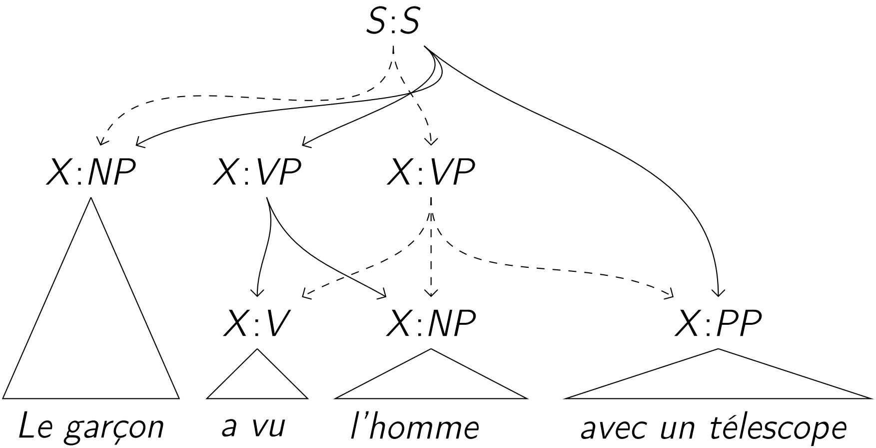
Decoding for Parsing-Based MT



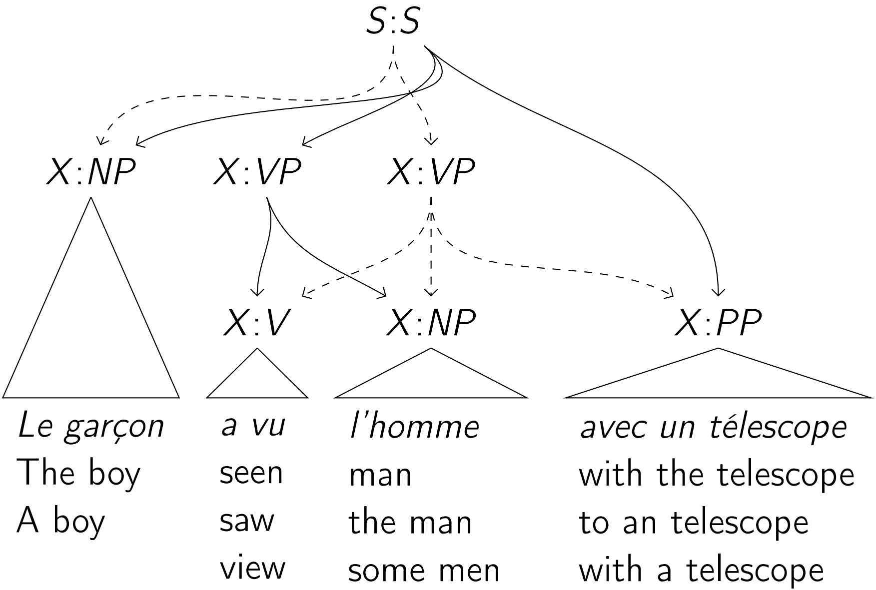
Decoding Example: Input

Le garçon *a vu* *l'homme* *avec un télescope*

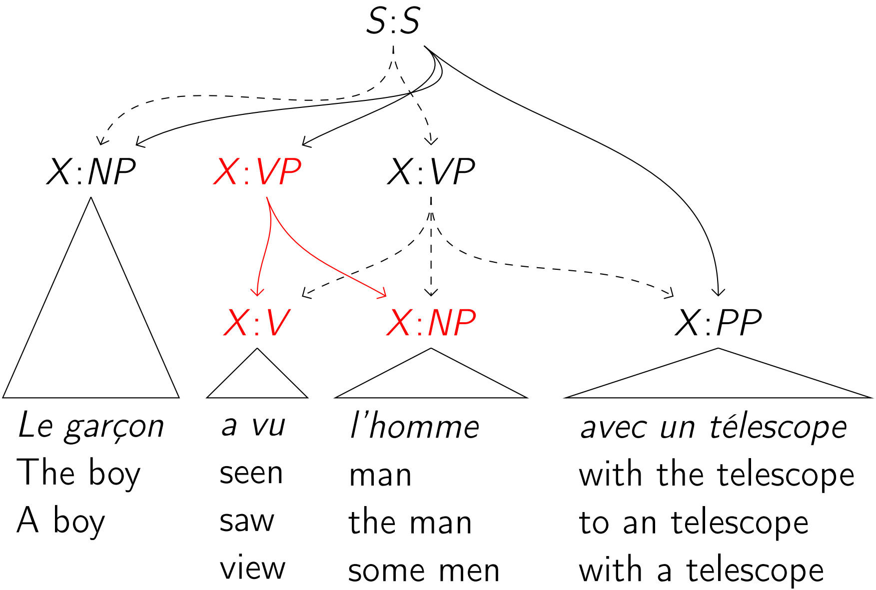
Decoding Example: Parse with SCFG

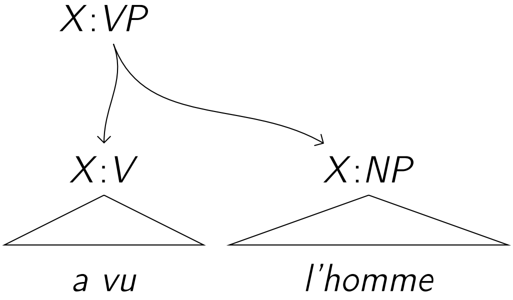


Decoding Example: Read Target Side



Decoding Example: One Constituent





Hyp	**Hyp**
seen	man
saw	the man
view	some men

X:*VP*

X:*V*	*X*:*NP*
a vu	*l'homme*
Hyp	**Hyp**
seen	man
saw	the man
view	some men

→

X:*VP*

a vu l'homme

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

$X\!:\!VP$

$X\!:\!V$ $X\!:\!NP$

a vu

Hyp	Score
seen	−3.8
saw	−4.0
view	−4.0

l'homme

Hyp	Score
man	−3.6
the man	−4.3
some men	−6.3

→

$X\!:\!VP$

a vu l'homme

Hypothesis	Score
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

X:*VP*

X:*V*

a vu

Hyp	Score
seen	−3.8
saw	−4.0
view	−4.0

X:*NP*

l'homme

Hyp	Score
man	−3.6
the man	−4.3
some men	−6.3

→

X:*VP*

a vu l'homme

Hypothesis	Score
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

$X\!:\!VP$

$X\!:\!V$

a vu

Hyp	Score
seen	-3.8
saw	-4.0
view	-4.0

$X\!:\!NP$

l'homme

Hyp	Score
man	-3.6
the man	-4.3
some men	-6.3

$X\!:\!VP$

a vu l'homme

Hypothesis	Score
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

$X:VP$

$X:V$ — *a vu*

Hyp	Score
seen	−3.8
saw	−4.0
view	−4.0

$X:NP$ — *l'homme*

Hyp	Score
man	−3.6
the man	−4.3
some men	−6.3

→

$X:VP$ — *a vu l'homme*

Hypothesis	Score
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~~

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.

	Option	Sum	Score
1	seen man	-7.4	-8.8
2	saw man	-7.6	-8.3
3	view man	-7.6	-8.5
4	seen the man	-8.1	-7.6
5	saw the man	-8.3	**-6.9**
✗	view the man	-8.3	-8.9
✗	seen some men	10.1	-9.5
✗	saw some men	-10.3	-8.5
✗	view some men	-10.3	-10.8

Beam Size 4: Search error.

	Option	Sum	Score
1	seen man	-7.4	-8.8
2	saw man	-7.6	-8.3
3	view man	-7.6	-8.5
4	seen the man	-8.1	-7.6
✗	saw the man	-8.3	**-6.9**
✗	view the man	-8.3	-8.9
✗	seen some men	10.1	-9.5
✗	saw some men	-10.3	-8.5
✗	view some men	-10.3	-10.8

Problem With Cube Pruning

Hyp
is a
are a

Hypothesis
countries that
countries which
country

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

String Concatenation

Hypotheses are built by string concatenation.
The language model score changes when this is done:

$$\frac{p(\text{saw the man})}{p(\text{saw})p(\text{the man})} = \frac{p(\text{the} \mid \text{saw})p(\text{man} \mid \text{saw the})}{p(\text{the}) \qquad p(\text{man} \mid \text{the})}$$

String Concatenation

Hypotheses are built by string concatenation.
The language model score changes when this is done:

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

$$\frac{p(\text{saw the man})}{p(\text{saw})p(\text{the man})} = \frac{p(\text{the} \mid \text{saw})p(\text{man} \mid \text{saw the})}{p(\text{the}) \qquad p(\text{man} \mid \text{the})}$$

String Concatenation

Hypotheses are built by string concatenation.
The language model score changes when this is done:

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

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

What words does correction c examine?

Markov Assumption

A 5-gram language model uses up to 4 words of context:

$$p(\text{man} \mid \text{<s> the boy saw the}) = p(\text{man} \mid \text{the boy saw the})$$

$$\Longrightarrow$$

Correction c examines up to 4 words from each string:

$$\text{c}(\text{<s> } \underbrace{\boxed{\vdash \text{ the boy saw the}}}_{\text{Right State}} \bullet \underbrace{\boxed{\text{man with a telescope} \dashv}}_{\text{Left State}} \text{ .})$$

Markov Assumption

A 5-gram language model uses up to 4 words of context:

$$p(\text{man} \mid \text{<s> the boy saw the}) = p(\text{man} \mid \text{the boy saw the})$$

$$\Longrightarrow$$

Correction c examines up to 4 words from each string:

$$\text{c}(\text{<s>} \underbrace{\boxed{\vdash \text{ the boy saw the}}}_{\text{Right State}} \bullet \underbrace{\boxed{\text{man with a telescope } \dashv}}_{\text{Left State}} \text{ .})$$

State may be shorter than 4 words [Li and Khudanpur, 2008]

Partial translations have state...

Left State				Right State
countries that ⊣	maintain	diplomatic	relations	⊢ with North Korea .
countries that ⊣	maintain	diplomatic	ties	⊢ with North Korea .

...so they can concatenate on either side.

Partial translations have state. . .

Left State				**Right State**
countries that ⊣	maintain	diplomatic	relations / ties	⊢ with North Korea .

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

Outline

1. String Concatenation
2. **Incremental Expansion**

Example Hypotheses

Left State		**Right State**
countries that ⊣	maintain diplomatic relations ties	⊢ with North Korea .
countries that have ⊣	an embassy in	⊢ DPR Korea .
country ⊣	that maintains some diplomatic ties	⊢ in North Korea .
nations which has ⊣	some diplomatic ties	⊢ with DPR Korea .
country ⊣	that maintains some diplomatic ties	⊢ with DPR Korea .

Example Hypotheses

Left State		Right State
(countries that	$\dashv \diamond \vdash$	with North Korea .)
(nations which has	$\dashv \diamond \vdash$	with DPR Korea .)
(countries that have	$\dashv \diamond \vdash$	DPR Korea .)
(country	$\dashv \diamond \vdash$	in North Korea .)
(country	$\dashv \diamond \vdash$	with DPR Korea .)

$\diamond$ denotes words omitted by state.

High Level Idea of Incremental Expansion

Group hypotheses by common words.

Group by Leftmost Word

$$\begin{tikzcd} & \text{countries} \\ (\epsilon \diamond \epsilon) \arrow[ur] \arrow[r] \arrow[dr] & \text{nations} \\ & \text{country} \end{tikzcd}$$

Reveal Common Words in Each Group

$$\begin{tikzcd}
 & (\text{countries that} \diamond \text{Korea .}) \\
(\epsilon \diamond \epsilon) \arrow[ur] \arrow[r] \arrow[dr] & (\text{nations which has} \dashv \diamond \vdash \text{with DPR Korea .}) \\
 & (\text{country} \dashv \diamond \text{Korea .})
\end{tikzcd}$$

Alternate Sides Until Tree is Full

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

(countries that $\diamond$ Korea .)

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

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

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

(country $\dashv \diamond$ Korea .)

(country $\dashv \diamond \vdash$ 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

$\underbrace{(\epsilon \diamond \epsilon)}_{X:V1}$ the $\underbrace{(\epsilon \diamond \epsilon)}_{X:N2}$

Exploring and Backtracking

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

- Yes Try more detail.
- No Consider alternatives.

Exploring and Backtracking

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

Yes Try more detail.

No Consider alternatives.

Formally: priority queue containing breadcrumbs.

Split and Leave Breadcrumbs

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

(countries that $\diamond$ Korea .)

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

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

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

(country $\dashv \diamond$ Korea .)

(country $\dashv \diamond \vdash$ with DPR Korea .)

Split and Leave Breadcrumbs

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

(countries that $\diamond$ Korea .)

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

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

[1+]

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

(country $\dashv \diamond$ Korea .)

(country $\dashv \diamond \vdash$ with DPR Korea .)

Splitting

The queue entry

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

splits into

Zeroth Child "is a (countries that $\diamond$ Korea .) </s>"
Other Children "is a $(\epsilon \diamond \epsilon)$[1+] </s>"

Children except the zeroth.

Summary So Far

A priority queue contains competing entries:

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

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

Summary So Far

A priority queue contains competing entries:

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

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

Next: Scoring queue entries

Scores come from the best descendant:

Score($\epsilon \diamond \epsilon$)=
Score(countries that $\dashv \diamond \vdash$ with North Korea .)

$$\geq$$

Score($\epsilon \diamond \epsilon$)[1+]=
Score(nations which has $\dashv \diamond \vdash$ with DPR Korea .)

Estimates Update as Words are Revealed:

is a ($\epsilon \diamond \epsilon$) </s> → is a (countries that $\diamond$ Korea .) </s>

p(is)	p(is)
p(a | is)	p(a | is)
p(countries)	p(countries | is a)
p(that | countries)	p(that | is a countries)
p(</s>)	p(</s> | Korea .)

Tightly integrated coarse-to-fine [Petrov et al, 2008]

Summary

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).

Summary

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).

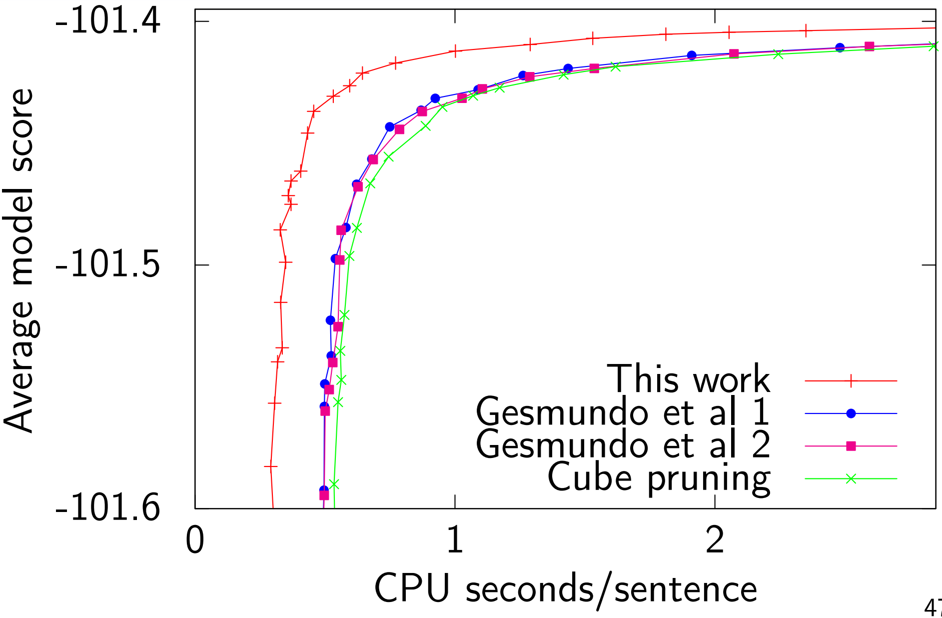
Experimental Setup

Task WMT 2011 German-English

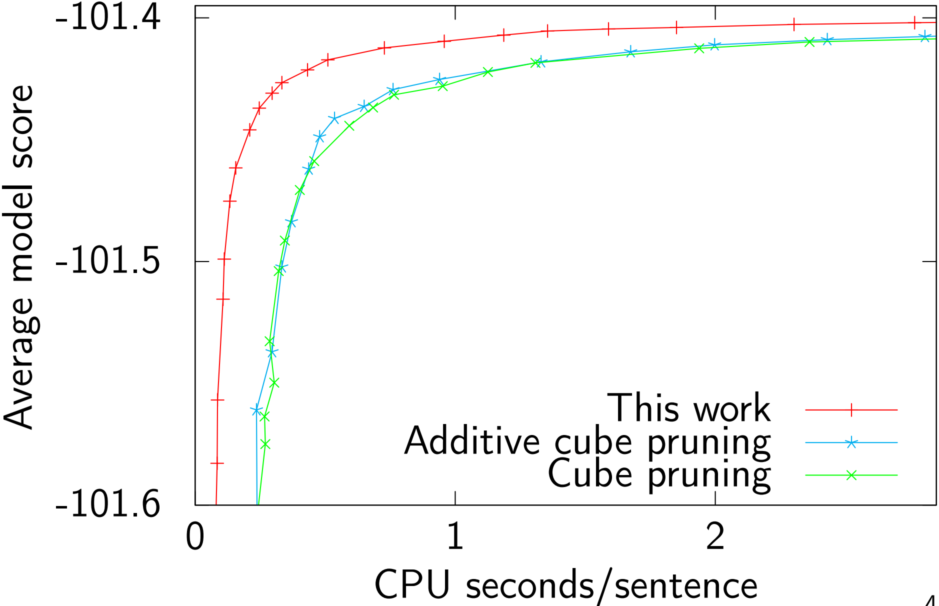
Builder [Koehn et al, 2011]

Model Hierarchical

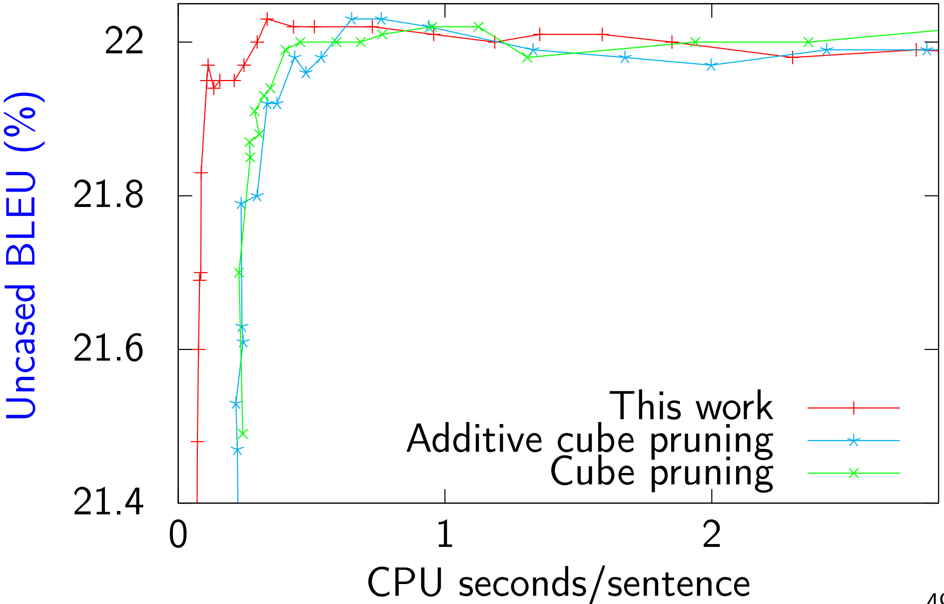
cdec Hierarchical



Moses Hierarchical



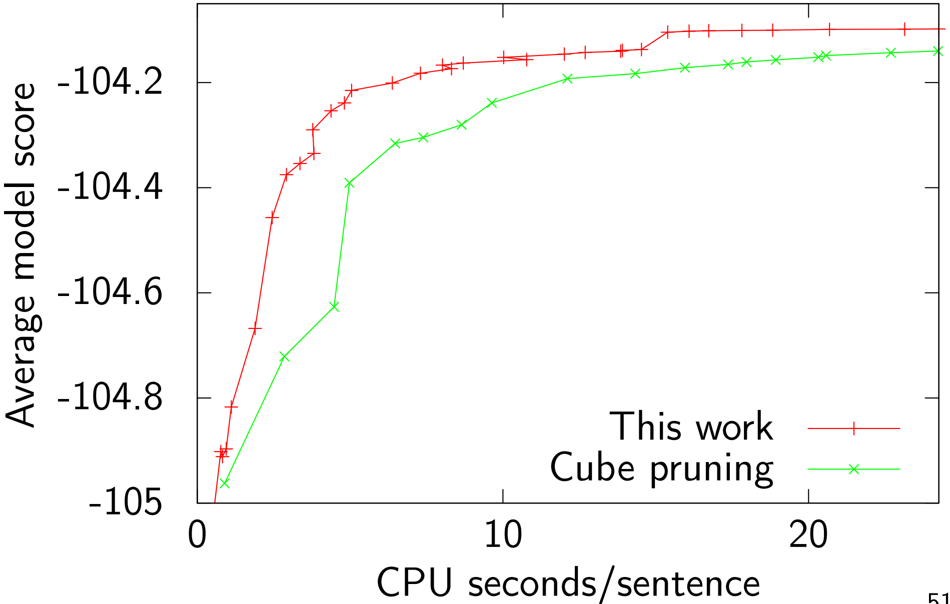
Moses Hierarchical



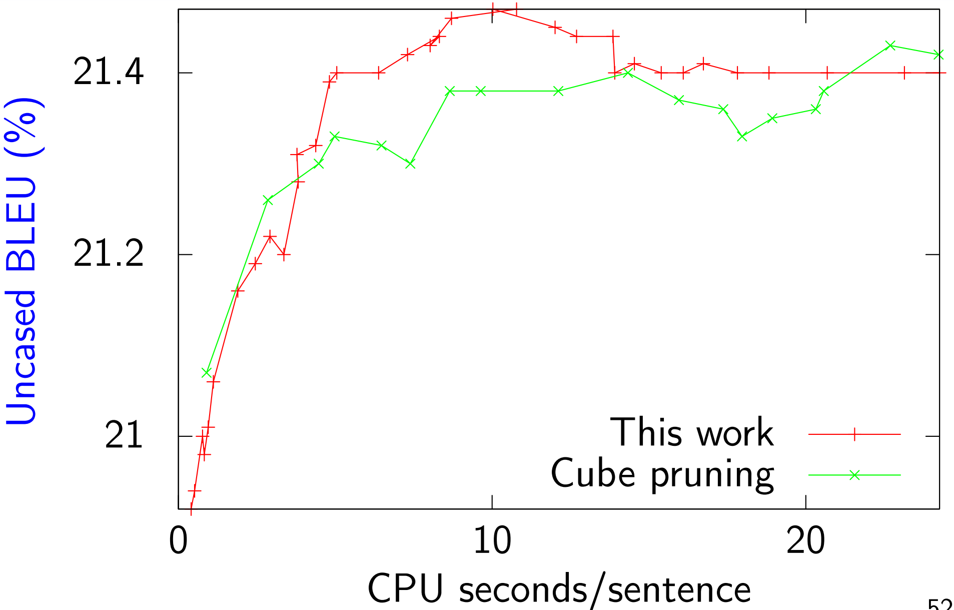
Now With Target Syntax

Task	WMT 2011 German-English
Builder	[Koehn et al, 2011]
Model	Target Syntax

Moses Target Syntax



Moses Target Syntax



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.