Faster Decoding for Phrases and Syntax

Kenneth Heafield

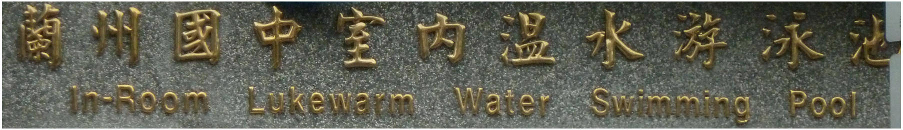
Translation is Expensive

"speed-up in tuning time but affects the performance"
"18 days using 12 cores"
[Williams et al WMT 2014]

"Time-sensitive BLEU score"
[Chung and Galley, 2012]

"Due to time constraints, this procedure was not used"
[Servan et al, WMT 2012]

$\Longrightarrow$ Routine Quality Compromises


蘭州國中室内温水游泳池
In-Room Lukewarm Water Swimming Pool

Blame the Language Model

"LM queries often account for more than 50% of the CPU"
[Green et al, WMT 2014]

Blame the Language Model

"LM queries often account for more than 50% of the CPU"
[Green et al, WMT 2014]

Faster queries (KenLM)
More effective queries

Cube pruning

- Widely used for phrase-based and syntax-based MT
- May be applied in conjunction with a bottom-up decoder, or as a second "rescoring" pass
  - Nodes may also be grouped together (for example, all nodes corresponding to a certain source span)
- Requirement for topological ordering means translation hypergraph may not have cycles

Cube pruning

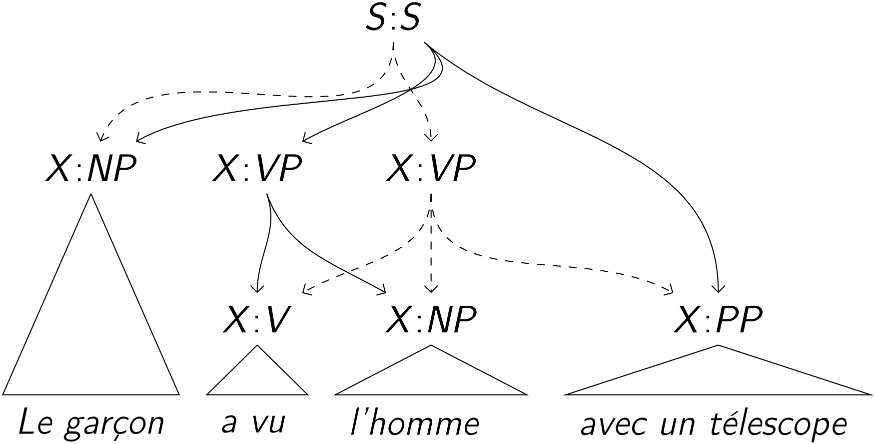
- Widely used for phrase-based and syntax-based MT
- May be applied in conjunction with a bottom-up decoder, or as a second "rescoring" pass
  - Nodes may also be grouped together (for example, all nodes corresponding to a certain source span)
- Requirement for topological ordering means translation hypergraph may not have cycles

1 Decoding problem
2 Cube pruning
3 Incremental

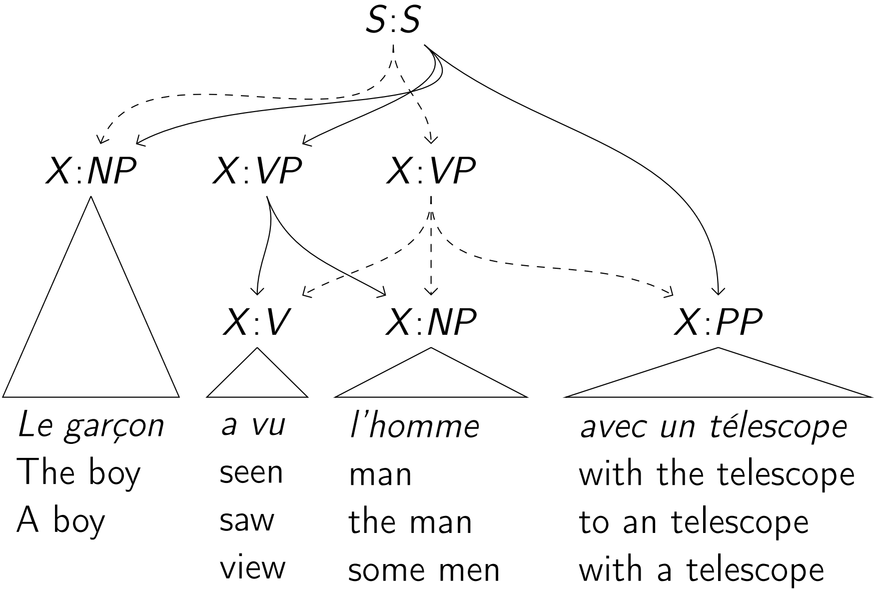
Decoding Example: Input

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

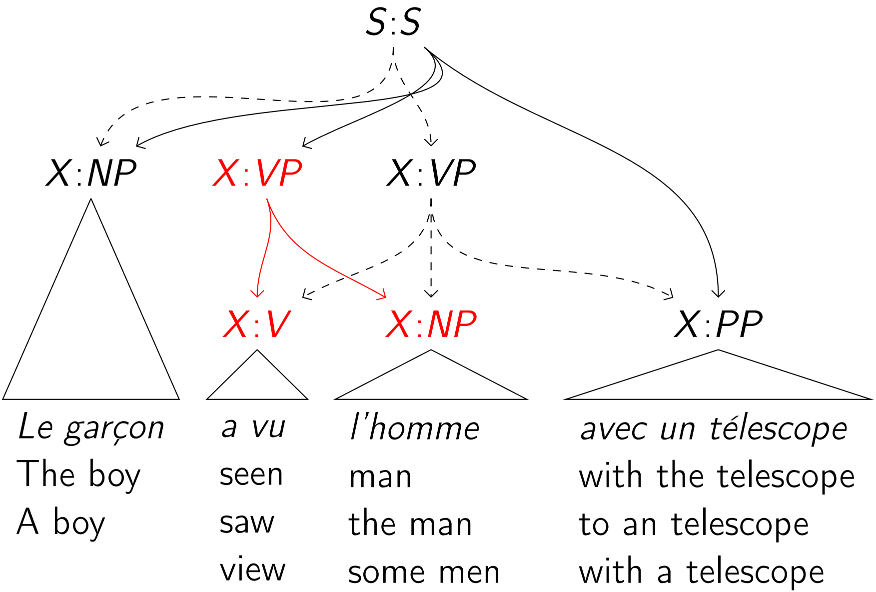
Decoding Example: Parse with SCFG

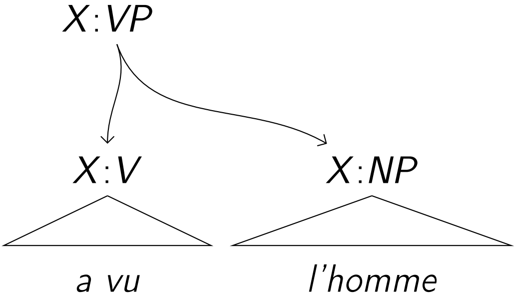


Decoding Example: Read Target Side



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

$X : VP$

$X : V$ — *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
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

Scores do not sum

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~~

Pruning is Approximate

Appending Strings

Hypotheses are built by string concatenation.
Language model probability changes when this is done:

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

Appending Strings

Hypotheses are built by string concatenation.
Language model probability changes when this is done:

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

Log probability is part of the score
$\implies$ Scores do not sum
$\implies$ Local decisions may not be globally optimal
$\implies$ Search is hard.

1 Decoding problem

2 **Cube pruning**

3 Incremental

Beam Search

		man	**−3.6**	**the man**	**−4.3**	**some men**	**−6.3**
seen	**−3.8**	seen man	−8.8	seen the man	−7.6	seen some men	−9.5
saw	**−4.0**	saw man	−8.3	saw the man	−6.9	saw some men	−8.5
view	**−4.0**	view man	−8.5	view the man	−8.9	view some men	−10.8

[Lowerre, 1976; Chiang, 2005]

Cube Pruning

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

Queue

Hypothesis	Sum
→seen man	$-3.8-3.6=-7.4$

[Chiang, 2007]

Cube Pruning

	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

	Hypothesis	Sum
→	saw man	$-4.0 - 3.6 = -7.6$
	seen the man	$-3.8 - 4.3 = -8.1$

[Chiang, 2007]

Cube Pruning

	man **−3.6**	**the man** **−4.3**	**some men** **−6.3**
seen **−3.8**	seen man −8.8	Queue	
saw **−4.0**	saw man −8.3	Queue	
view **−4.0**	Queue		

Queue

Hypothesis	**Sum**
→ view man	$-4.0-3.6=-7.6$
seen the man	$-3.8-4.3=-8.1$
saw the man	$-4.0-4.3=-8.3$

[Chiang, 2007]

Cube Pruning

	man $-\mathbf{3.6}$	**the man** $-\mathbf{4.3}$	**some men** $-\mathbf{6.3}$
seen $-\mathbf{3.8}$	seen man -8.8	Queue	
saw $-\mathbf{4.0}$	saw man -8.3	Queue	
view $-\mathbf{4.0}$	view man -8.5	Queue	

Queue

Hypothesis	**Sum**
→seen the man	$-3.8-4.3=-8.1$
saw the man	$-4.0-4.3=-8.3$
view the man	$-4.0-4.3=-8.3$

[Chiang, 2007]

Beam Search

Make every dish. Keep the best k, throw the rest out.

Cube pruning

Combine the best ingredients. Only make k dishes.

Cube Pruning Hypotheses are Atomic

String
is a
are a

String
countries that
countries which
country

String
is a countries that
are a countries that
are a countries which
⋮

No notion that "a countries" is bad.

Beam Search

Make every dish. Keep the best k, throw the rest out.

Cube pruning

Combine the best ingredients. Only make k dishes.

Coarse-to-Fine

Make small portions, taste, and order the best ones.

Coarse-to-Fine

Decode multiple times, adding detail each time:

Increased LM order, words instead of classes

Detect and prune "a countries" with a bigram LM.

[Zhang et al, 2008; Petrov et al, 2008]

Coarse-to-Fine

Decode multiple times, adding detail each time:

Increased LM order, words instead of classes

Detect and prune "a countries" with a bigram LM.

[Zhang et al, 2008; Petrov et al, 2008]

Requires tuning each pruning pass.

Operates in lock step.

Coarse-to-Fine

Decode multiple times, adding detail each time:

Increased LM order, words instead of classes

Detect and prune "a countries" with a bigram LM.

[Zhang et al, 2008; Petrov et al, 2008]

Requires tuning each pruning pass.

Operates in lock step.

Can coarse-to-fine be done on the fly?

1 Decoding problem

2 Cube pruning

3 **Incremental**

Observations

Competing translations have words in common:

is a, are a

Observations

Competing translations have words in common:
is a, are a

Words at the boundary matter most:
a + country, a + countries

Observations

Competing translations have words in common:

is a, are a

Words at the boundary matter most:

a + country, a + countries

Emphasize boundary words

Beam Search

Make every dish. Keep the best k, throw the rest out.

Cube pruning

Combine the best ingredients. Only make k dishes.

Coarse-to-Fine

Make small portions, taste, and order the best ones.

Incremental

Taste during cooking. Share ingredients.

Boundary Words

1 Left-to-right phrase-based: one side
2 Bottom-up syntax: both sides

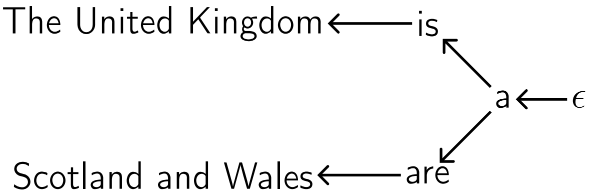
Partial Translations

Plain text

The United Kingdom is a + ...

Scotland and Wales are a + ...

Tree



Phrase Continuations

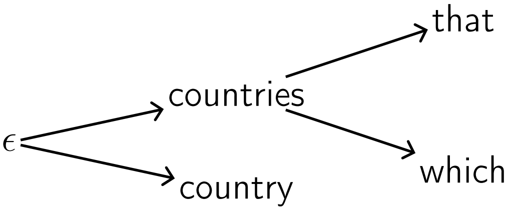
Plain text

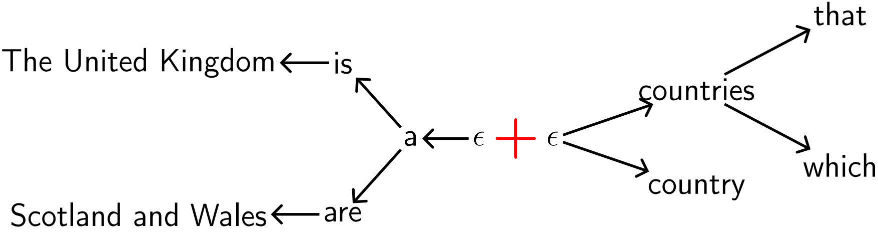
. . . + countries that

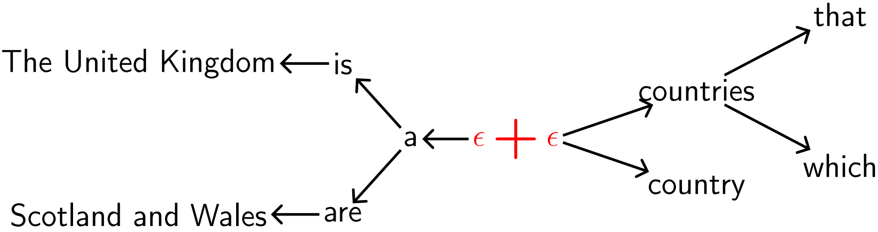
. . . + countries which

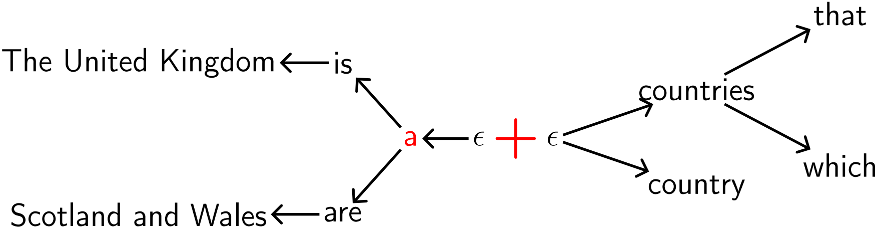
. . . + country

Tree

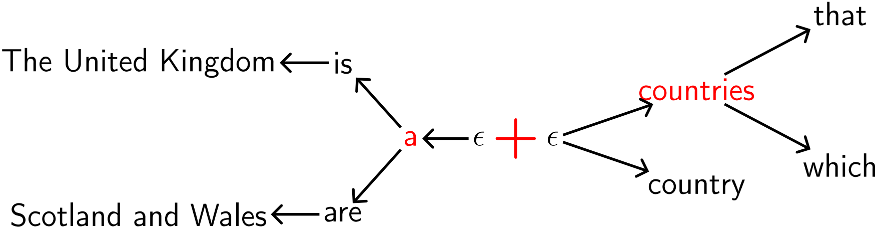



The United Kingdom
is
a
ϵ
ϵ
Scotland and Wales
are
countries
country
that
which


The United Kingdom
is
a
Scotland and Wales
are
ϵ + ϵ
countries
country
that
which



The United Kingdom
is
Scotland and Wales
are
a
ϵ
ϵ
countries
country
that
which



Does the model like "a + countries"?

Exploring and Backtracking

Does the model like "a + countries"?

Yes Try more detail.

No Consider alternatives.

Exploring and Backtracking

Does the model like "a + countries"?

Yes Try more detail.

No Consider alternatives.

Formally: best-first search with a priority queue.

The queue entry

"a + ϵ"

splits into

Best Child "a + countries"
Other Children "a + country"

Scores come from the best descendant:

$$\text{Score}(\text{a}) = \max\{\text{Score}(\text{is a}), \text{Score}(\text{are a})\}$$

Scores come from the best descendant:

$$\text{Score(a)} = \max\{\text{Score(is a)}, \text{Score(are a)}\}$$

The language model updates scores:

$$\text{Score(a + countries)} < \text{Score(a)} + \text{Score(countries)}$$

Scores come from the best descendant:

$$\text{Score(a)} = \max\{\text{Score(is a)}, \text{Score(are a)}\}$$

The language model updates scores:

$$\text{Score(a + countries)} < \text{Score(a)} + \text{Score(countries)}$$

Formally: $p(\text{countries} \mid \text{a})$ replaces $p(\text{countries})$

Best-First Algorithm Summary

Populate the queue with $\epsilon + \epsilon$

Loop until k complete options have been found:

Split the top-scoring option

Build a tree from the k complete options

Summary

Translations are assembled from left to right.

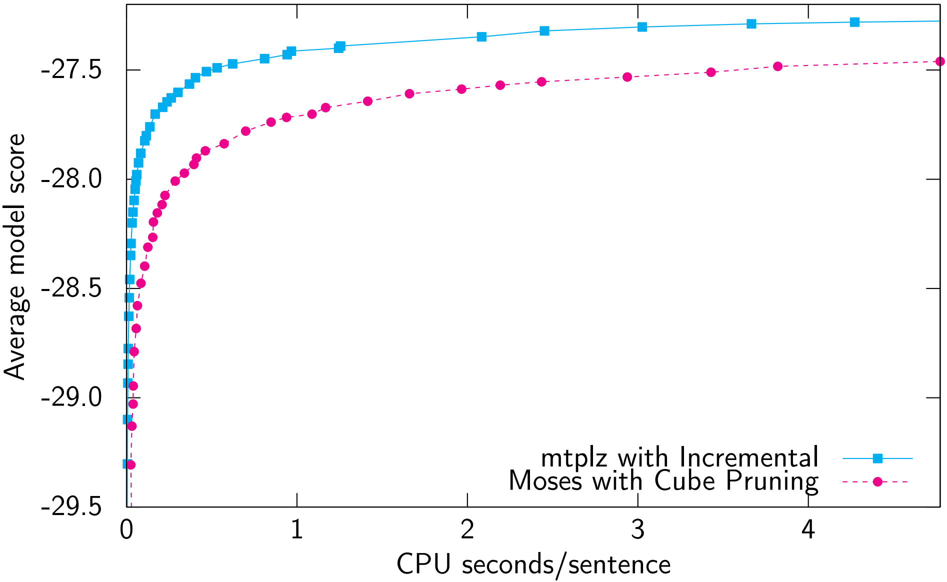
Partial translations often share suffixes.
Phrases often share prefixes.

Test suffixes and prefixes before full combinations.

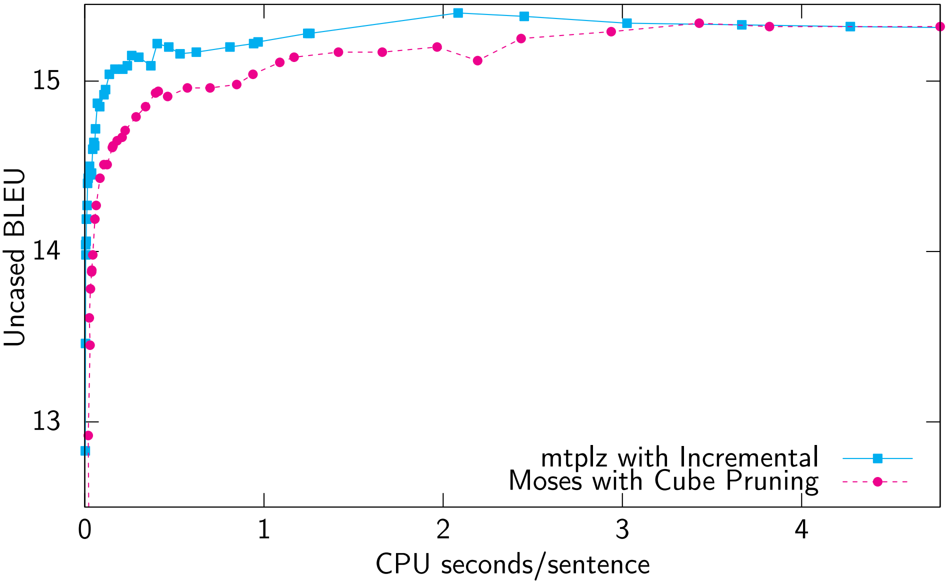
Experiment

Task	Chinese–English
Source	Stanford
Model	Phrase-based
Software	My own decoder, mtplz, versus Moses

Phrase-Based Results



Phrase-Based Results



Search

The language model cares most about adjacent words.

Test them first.

Share work for shared words.

Boundary Words

1 Left-to-right phrase-based: one side
2 **Bottom-up syntax: both sides**

Bottom-Up Syntax: Both Sides

is a X:$NP1$ </s>
is a X:$NP1$ that

How do we find the best value to substitute?
Manage words on both sides.

Example Hypotheses

Left State **Right State**

$\boxed{\text{countries that}}$ maintain diplomatic relations / ties $\boxed{\text{with North Korea .}}$

$\boxed{\text{countries that have}}$ an embassy in $\boxed{\text{DPR Korea .}}$

$\boxed{\text{country}}$ that maintains some diplomatic ties $\boxed{\text{in North Korea .}}$

$\boxed{\text{nations which has}}$ some diplomatic ties $\boxed{\text{with DPR Korea .}}$

$\boxed{\text{country}}$ that maintains some diplomatic ties $\boxed{\text{with DPR Korea .}}$

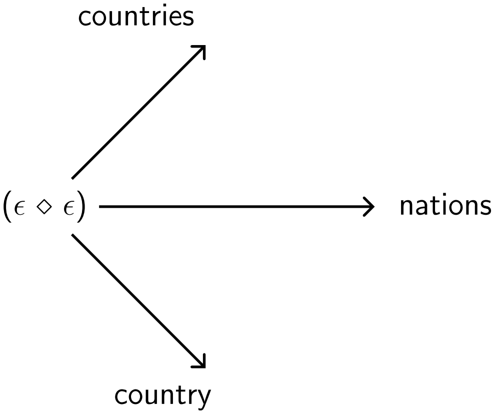
Example Hypotheses

Left State		**Right State**
(countries that	⋄	with North Korea .)
(nations which has	⋄	with DPR Korea .)
(countries that have	⋄	DPR Korea .)
(country	⋄	in North Korea .)
(country	⋄	with DPR Korea .)

⋄ Words the language model does not care about

Idea: alternate between left and right side

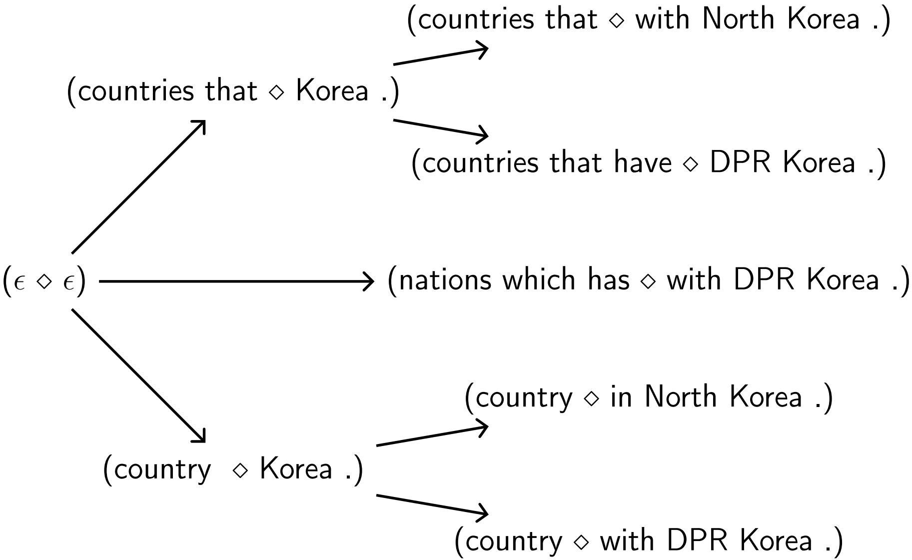
Group by Leftmost Word



Reveal Common Words in Each Group

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

Alternate Sides Until Tree is Full



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

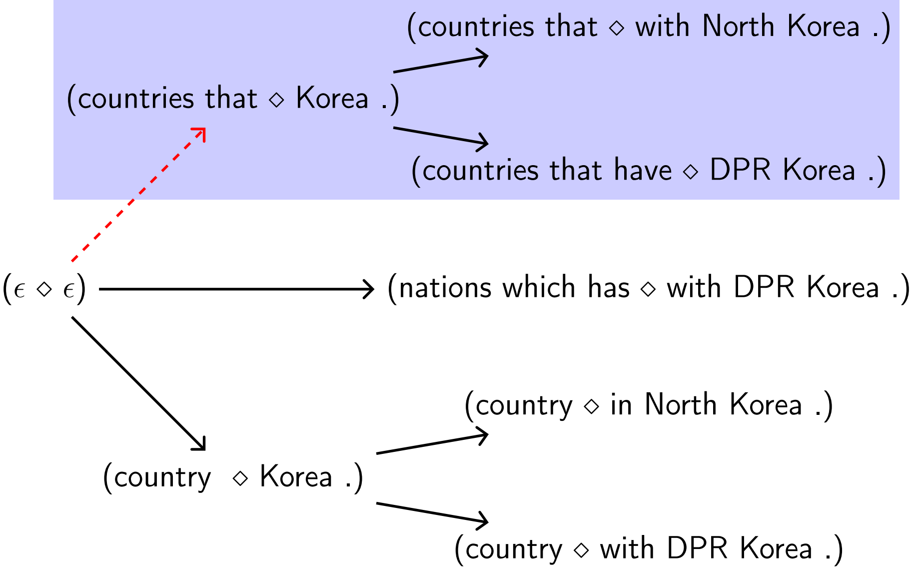
Does the LM like "is a (countries that $\diamond$ Korea .) </s>"?

Yes Try more detail.

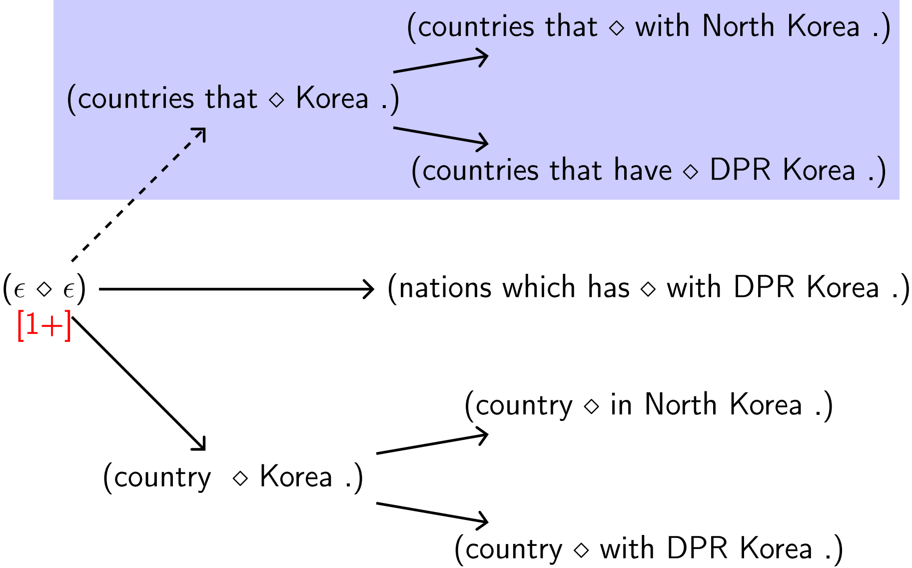
No Consider alternatives.

Formally: priority queue containing breadcrumbs.

Split and Leave Breadcrumbs



Split and Leave Breadcrumbs



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.

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.

Best-First Algorithm

Populate the queue with rules like "is a $(\epsilon \diamond \epsilon)$ </s>"

Loop until k complete options have been found:

- **Split** the top-scoring option, leave a breadcrumb

Build a tree from the k complete options

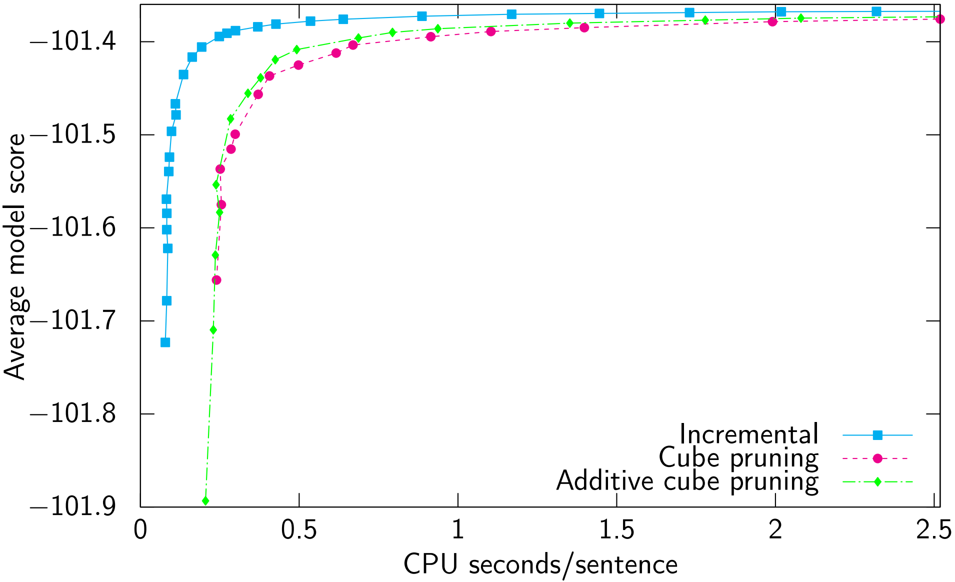
Syntax

Same as phrase-based, just concatenate on left and right.

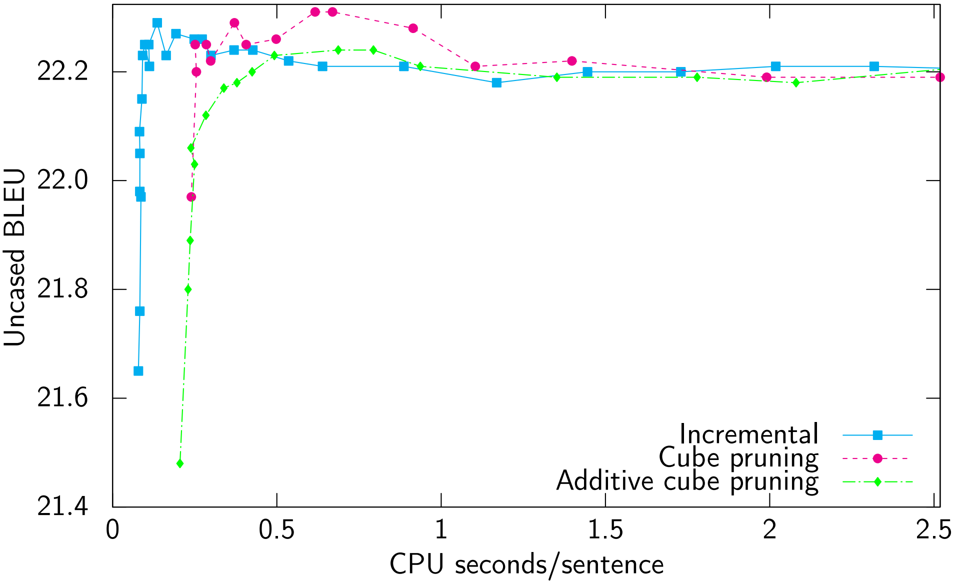
Experiment

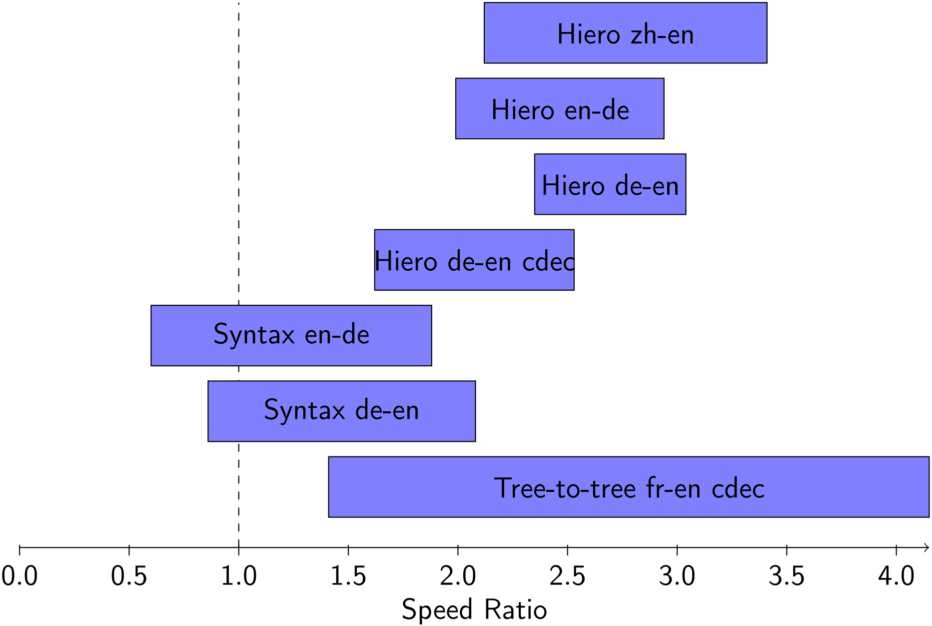
Task	WMT 2011 German-English
Model	Hierarchical
Decoder	Moses

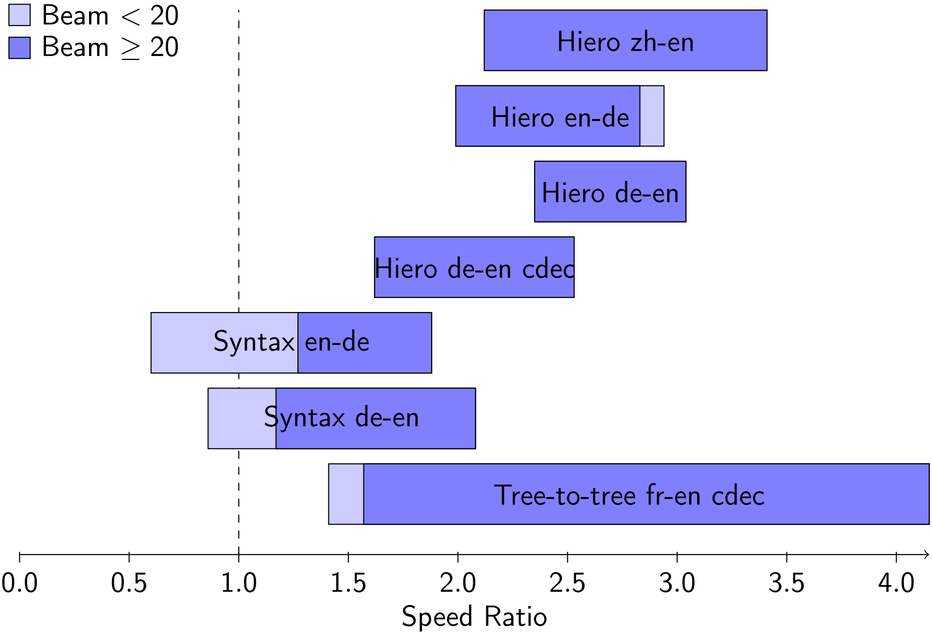
Moses Hierarchical



Moses Hierarchical




Hiero zh-en
Hiero en-de
Hiero de-en
Hiero de-en cdec
Syntax en-de
Syntax de-en
Tree-to-tree fr-en cdec
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
Speed Ratio


Beam < 20
Beam ≥ 20
Hiero zh-en
Hiero en-de
Hiero de-en
Hiero de-en cdec
Syntax en-de
Syntax de-en
Tree-to-tree fr-en cdec
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
Speed Ratio

Incremental

A series of coarse-to-fine estimates.

Continually taste the dish and adjust.

Takeaway

Search limits what translation can do.

Long-distance models like gender and number are harder.

Open the black box.

Language models can produce intermediate scores.