Hierarchical Phrase-Based Translation with Weighted Finite State Transducers

Bill Byrne

Department of Engineering
University of Cambridge

IWSLT 2010, Paris, France,

2 December 2010

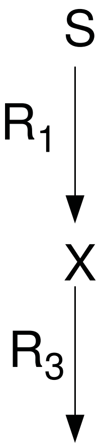
Overview

- WFST formulation of Hiero-style translation
  - Alternative implementation to Cube Pruning (CPH)
  - Works with lattices rather than individual translation hypotheses
  - Based on Google OpenFST toolkit
- Fast Hiero grammars
  - language-specific grammars to avoid pruning, search errors, ...
- Fast linearized lattice based minimum Bayes risk decoding (LMBR) with weighted finite state transducers

Joint work with Adrià de Gispert, Gonzalo Iglesias, Graeme Blackwood, Juan Pino

Hierarchical Phrase-based Translation

said



s_1 s_2 s_3 s_4 s_5 s_6

wqAl Alr}ys Alswry Ams Anh syEwd

(وقال الرئيس السوري امس انه سيعود)

R_1: $\mathbf{S}\longrightarrow\langle \mathbf{X}\ ,\ \mathbf{X}\rangle$

R_2: $S\longrightarrow\langle S\ X\ ,\ S\ X\rangle$

R_3: $\mathbf{X}\longrightarrow\langle \mathbf{s_1}\ ,\ \textbf{said}\rangle$

R_4: $X\longrightarrow\langle s_1\ s_2\ ,\ \text{the president said}\rangle$

R_5: $X\longrightarrow\langle s_1\ s_2\ s_3\ ,\ \text{Syrian president says}\rangle$

R_6: $X\longrightarrow\langle s_2\ ,\ \text{president}\rangle$

R_7: $X\longrightarrow\langle s_3\ ,\ \text{the Syrian}\rangle$

R_8: $X\longrightarrow\langle s_4\ ,\ \text{yesterday}\rangle$

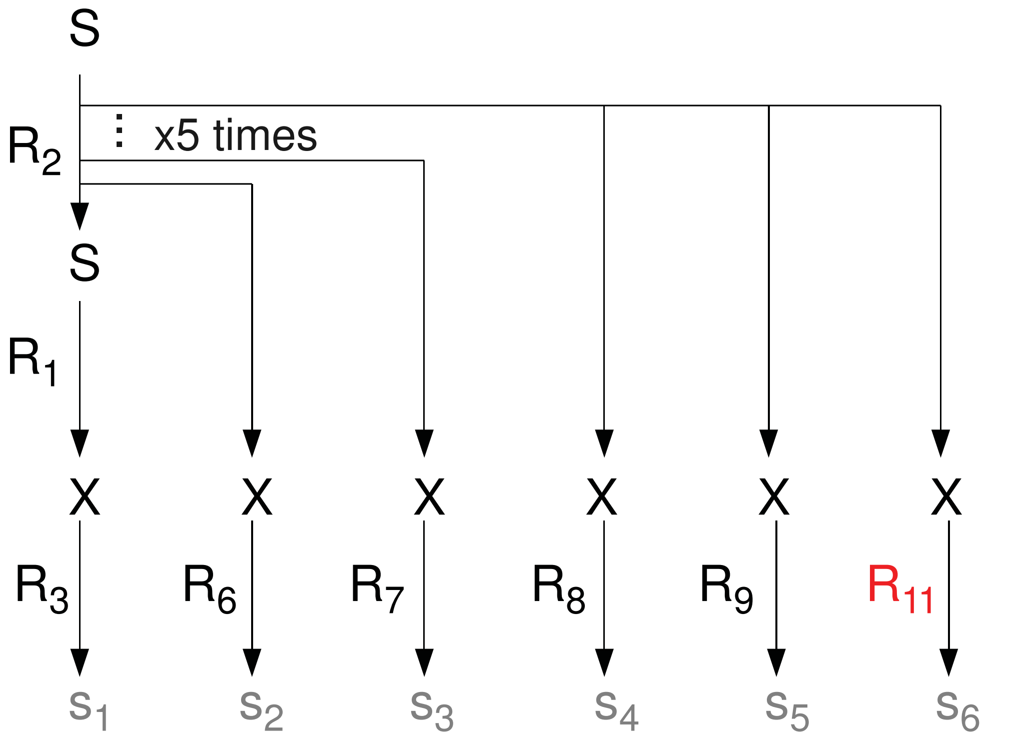
R_9: $X\longrightarrow\langle s_5\ ,\ \text{that}\rangle$

R_{10}: $X\longrightarrow\langle s_6\ ,\ \text{would return}\rangle$

R_{11}: $X\longrightarrow\langle s_6\ ,\ \text{he would return}\rangle$

Hierarchical Phrase-based Translation

said president the Syrian yesterday that he would return



wqAl Alr}ys Alswry Ams Anh syEwd

(وقال الرئيس السوري امس انه سيعود)

R_1: $S \rightarrow \langle X \ , \ X \rangle$

R_2: $\mathbf{S \rightarrow \langle S\ X \ , \ S\ X \rangle}$

R_3: $X \rightarrow \langle s_1$, said$\rangle$

R_4: $X \rightarrow \langle s_1\ s_2$, the president said$\rangle$

R_5: $X \rightarrow \langle s_1\ s_2\ s_3$, Syrian president says$\rangle$

R_6: $X \rightarrow \langle s_2$, president$\rangle$

R_7: $X \rightarrow \langle s_3$, the Syrian$\rangle$

R_8: $X \rightarrow \langle s_4$, yesterday$\rangle$

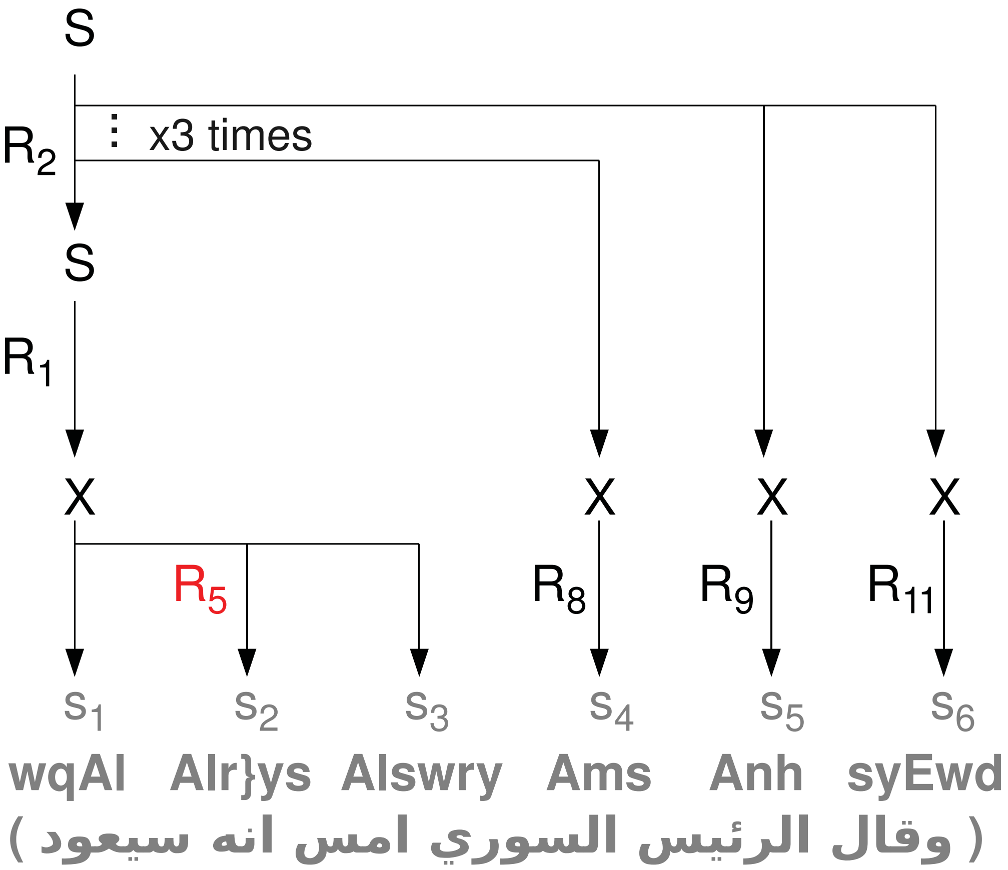
R_9: $X \rightarrow \langle s_5$, that$\rangle$

R_{10}: $\mathbf{X \rightarrow \langle s_6}$ **, would return**$\rangle$

R_{11}: $\mathbf{X \rightarrow \langle s_6}$ **, he would return**$\rangle$

Hierarchical Phrase-based Translation

Syrian president says yesterday that he would return


S
R2
⋮ x3 times
S
R1
X
X
X
X
R5
R8
R9
R11
s1
s2
s3
s4
s5
s6
wqAl Alr}ys Alswry Ams Anh syEwd
(وقال الرئيس السوري امس انه سيعود)

R_1: $S \rightarrow \langle X , X \rangle$

R_2: $S \rightarrow \langle S\ X , S\ X \rangle$

R_3: $X \rightarrow \langle s_1$, said$\rangle$

R_4: $X \rightarrow \langle s_1\ s_2$, the president said$\rangle$

R_5: $X \rightarrow \langle s_1\ s_2\ s_3$, **Syrian president says**$\rangle$

R_6: $X \rightarrow \langle s_2$, president$\rangle$

R_7: $X \rightarrow \langle s_3$, the Syrian$\rangle$

R_8: $X \rightarrow \langle s_4$, yesterday$\rangle$

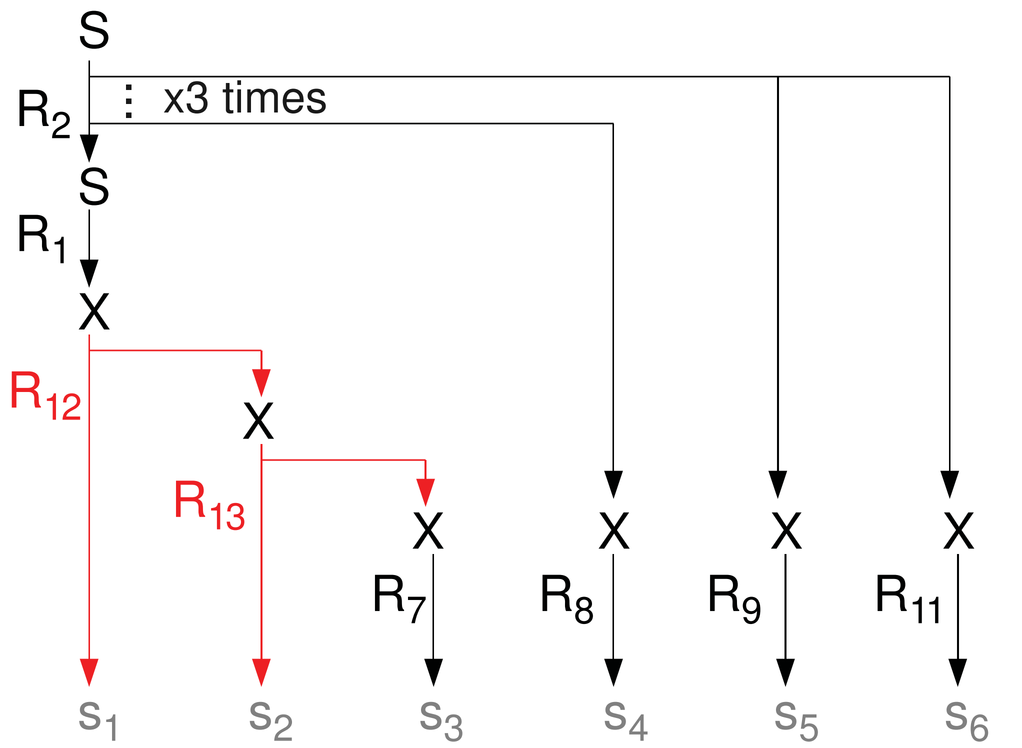
R_9: $X \rightarrow \langle s_5$, that$\rangle$

R_{10}: $X \rightarrow \langle s_6$, would return$\rangle$

R_{11}: $X \rightarrow \langle s_6$, he would return$\rangle$

Hierarchical Phrase-based Translation (2)

the Syrian president said yesterday that he would return



S

R$_2$ ⋮ x3 times

S

R$_1$

X

R$_{12}$

X

R$_{13}$

X X X X

R$_7$ R$_8$ R$_9$ R$_{11}$

s_1 s_2 s_3 s_4 s_5 s_6

wqAl Alr}ys Alswry Ams Anh syEwd

(وقال الرئيس السوري امس انه سيعود)

R_1: $S \rightarrow \langle X\ ,\ X \rangle$

R_2: $S \rightarrow \langle S\ X\ ,\ S\ X \rangle$

R_3: $X \rightarrow \langle s_1\ ,\ \text{said} \rangle$

...

R_6: $X \rightarrow \langle s_2\ ,\ \text{president} \rangle$

R_7: $X \rightarrow \langle s_3\ ,\ \text{the Syrian} \rangle$

R_8: $X \rightarrow \langle s_4\ ,\ \text{yesterday} \rangle$

R_9: $X \rightarrow \langle s_5\ ,\ \text{that} \rangle$

R_{10}: $X \rightarrow \langle s_6\ ,\ \text{would return} \rangle$

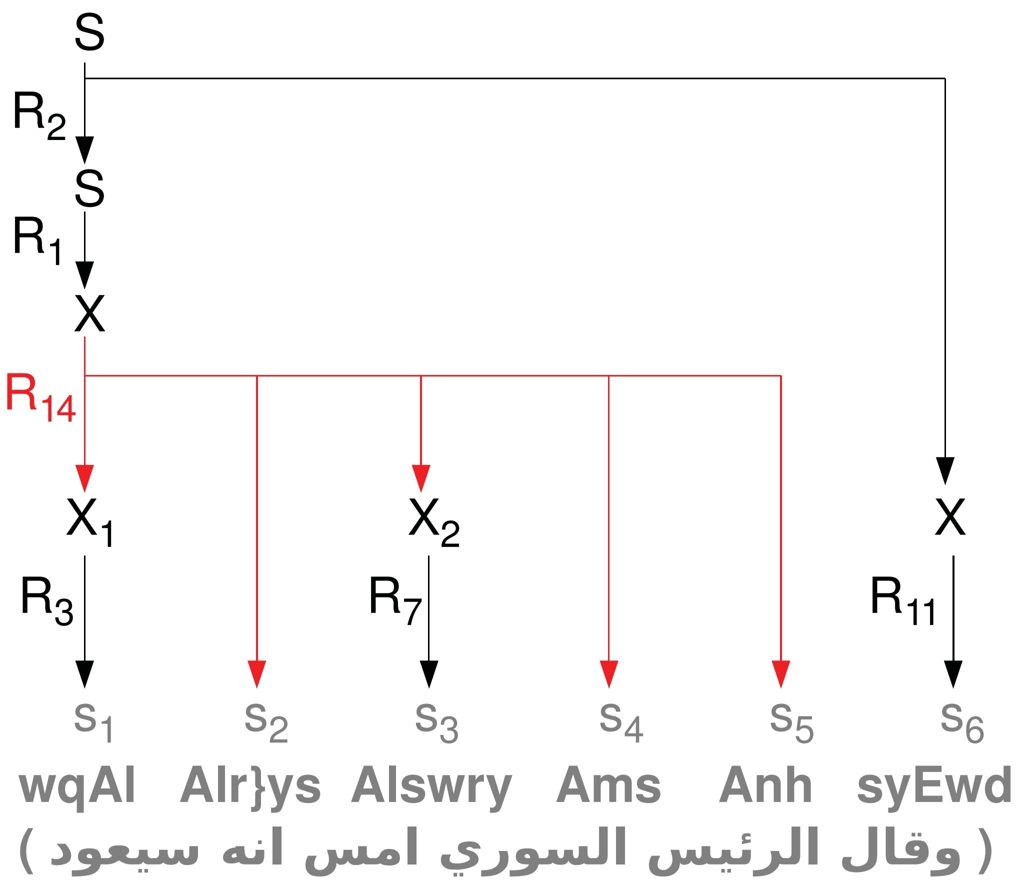
R_{11}: $X \rightarrow \langle s_6\ ,\ \text{he would return} \rangle$

R_{12}: $X \rightarrow \langle s_1\ X\ ,\ X\ \text{said} \rangle$

R_{13}: $X \rightarrow \langle s_2\ X\ ,\ X\ \text{president} \rangle$

Hierarchical Phrase-based Translation (2)

yesterday the Syrian president said that he would return



R_1: $S \longrightarrow \langle X\ ,\ X \rangle$

R_2: $S \longrightarrow \langle S\ X\ ,\ S\ X \rangle$

R_3: $X \longrightarrow \langle s_1\ ,\ \text{said} \rangle$

...

R_6: $X \longrightarrow \langle s_2\ ,\ \text{president} \rangle$

R_7: $X \longrightarrow \langle s_3\ ,\ \text{the Syrian} \rangle$

R_8: $X \longrightarrow \langle s_4\ ,\ \text{yesterday} \rangle$

R_9: $X \longrightarrow \langle s_5\ ,\ \text{that} \rangle$

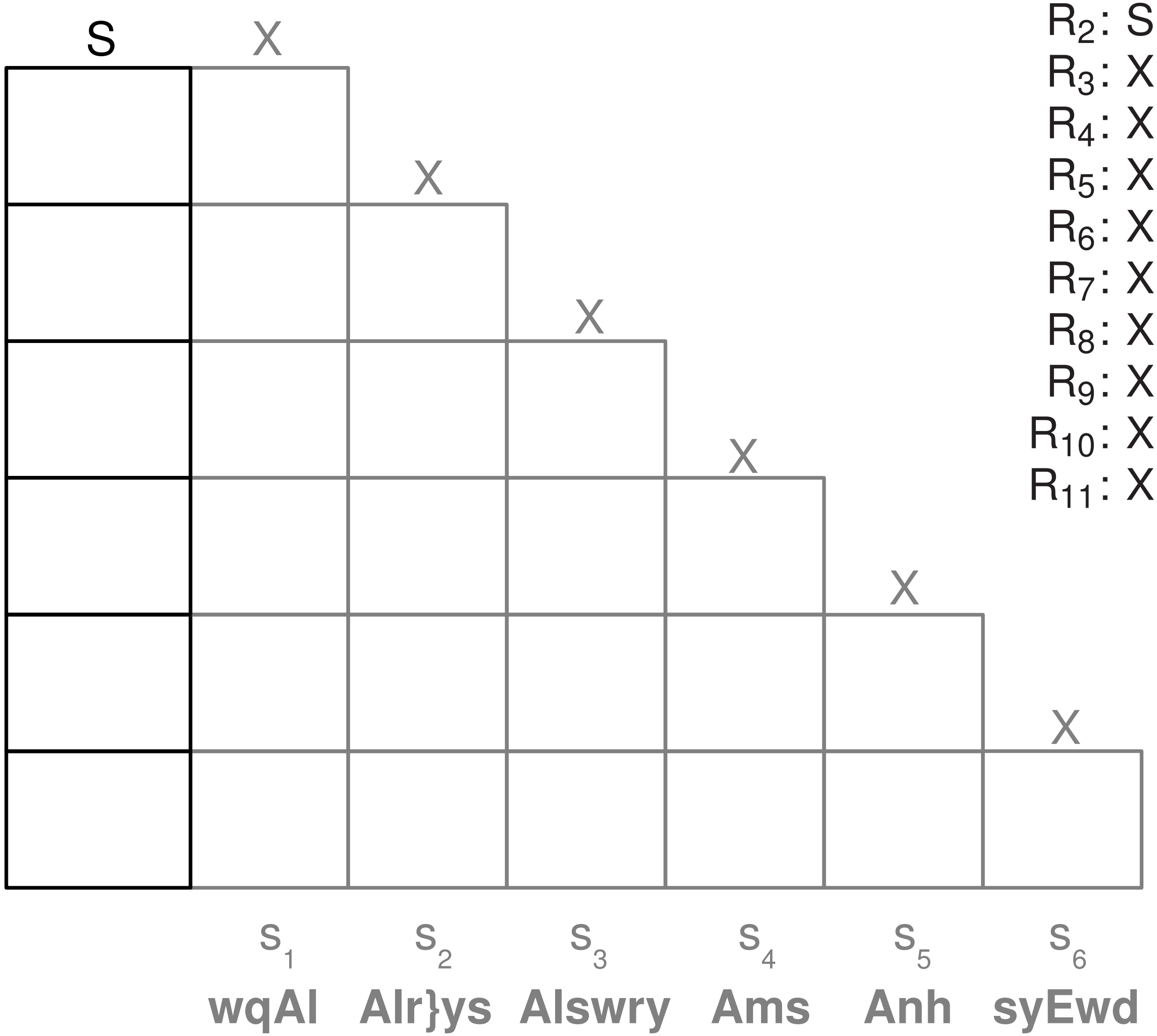
R_{10}: $X \longrightarrow \langle s_6\ ,\ \text{would return} \rangle$

R_{11}: $X \longrightarrow \langle s_6\ ,\ \text{he would return} \rangle$

R_{14}: $X \longrightarrow \langle X_1\ s_2\ X_2\ s_4\ s_5\ ,$
$\text{y'day}\ X_2\ \text{president}\ X_1\ \text{that} \rangle$

- Each rule has a probability assigned by the Translation Model

Keeping Track of All Derivations. CYK Grid



R_1: $S \rightarrow \langle X \text{ , } X \rangle$

R_2: $S \rightarrow \langle S\ X \text{ , } S\ X \rangle$

R_3: $X \rightarrow \langle s_1 \text{ , said} \rangle$

R_4: $X \rightarrow \langle s_1\ s_2 \text{ , the president said} \rangle$

R_5: $X \rightarrow \langle s_1\ s_2\ s_3 \text{ , Syrian president says} \rangle$

R_6: $X \rightarrow \langle s_2 \text{ , president} \rangle$

R_7: $X \rightarrow \langle s_3 \text{ , the Syrian} \rangle$

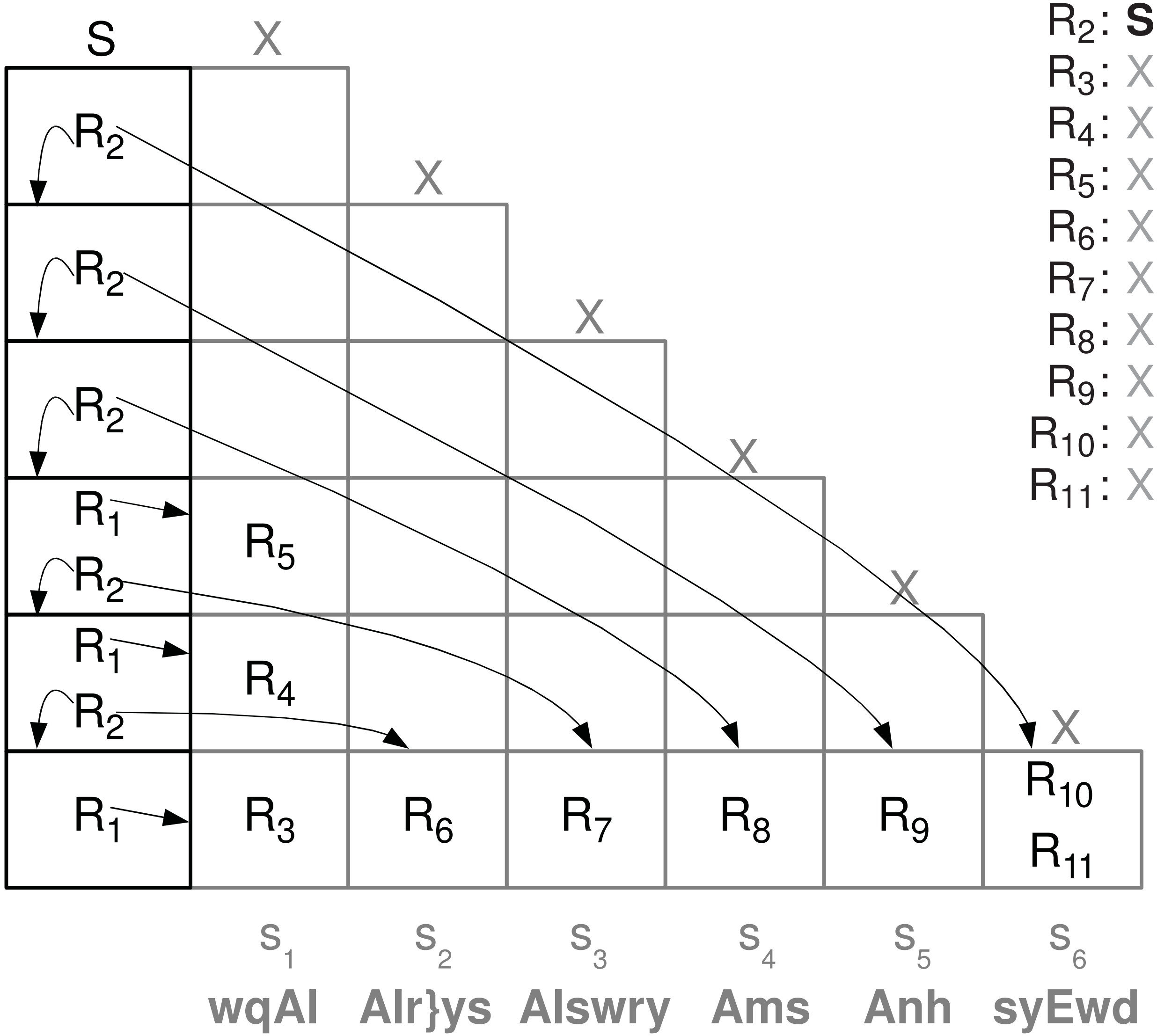
R_8: $X \rightarrow \langle s_4 \text{ , yesterday} \rangle$

R_9: $X \rightarrow \langle s_5 \text{ , that} \rangle$

R_{10}: $X \rightarrow \langle s_6 \text{ , would return} \rangle$

R_{11}: $X \rightarrow \langle s_6 \text{ , he would return} \rangle$

Keeping Track of All Derivations. CYK Grid



R_1: $\mathbf{S} \rightarrow \langle \mathbf{X} , \mathbf{X} \rangle$

R_2: $\mathbf{S} \rightarrow \langle \mathbf{S\ X} , \mathbf{S\ X} \rangle$

R_3: $X \rightarrow \langle s_1 , \text{said} \rangle$

R_4: $X \rightarrow \langle s_1\ s_2 , \text{the president said} \rangle$

R_5: $X \rightarrow \langle s_1\ s_2\ s_3 , \text{Syrian president says} \rangle$

R_6: $X \rightarrow \langle s_2 , \text{president} \rangle$

R_7: $X \rightarrow \langle s_3 , \text{the Syrian} \rangle$

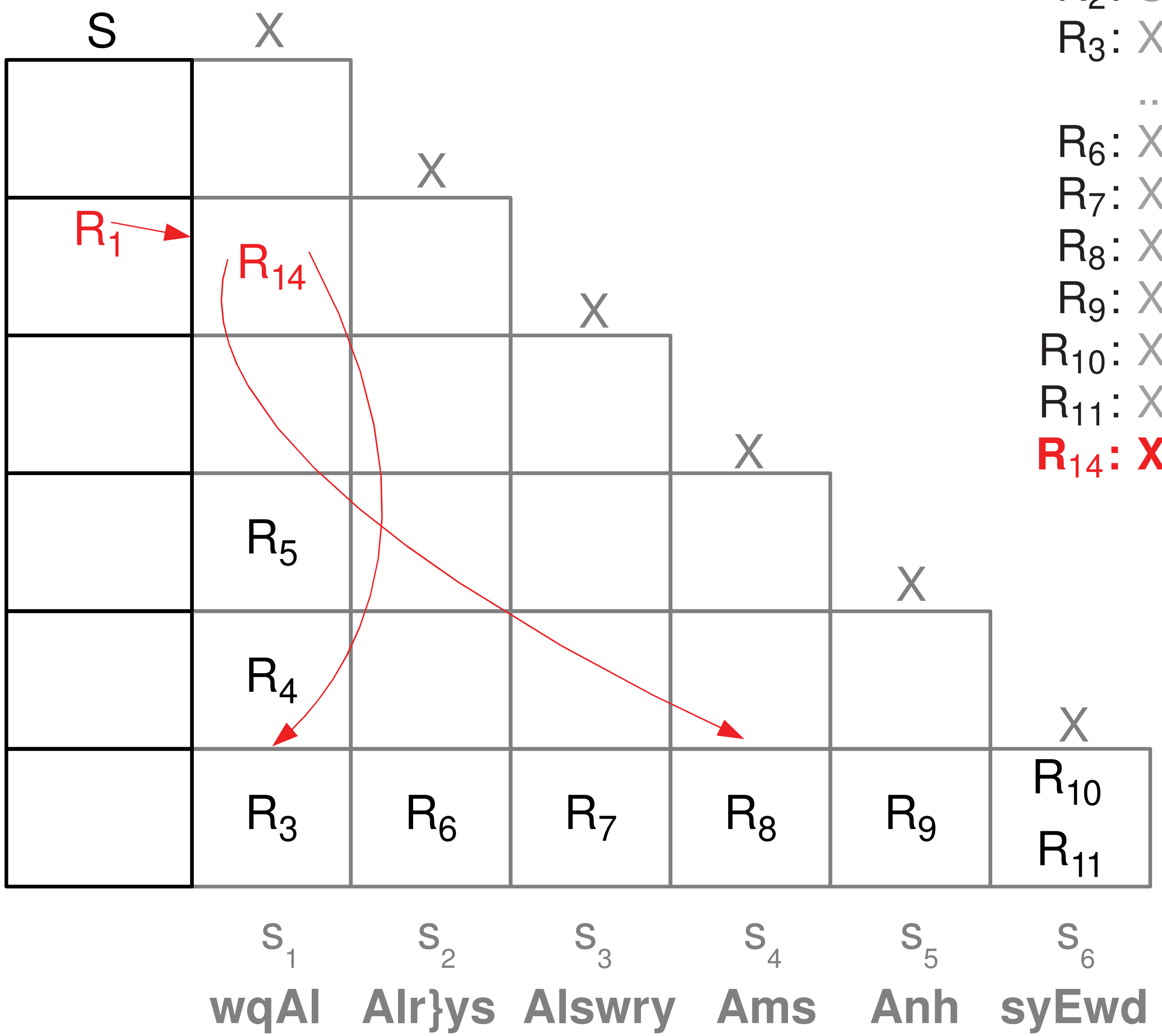
R_8: $X \rightarrow \langle s_4 , \text{yesterday} \rangle$

R_9: $X \rightarrow \langle s_5 , \text{that} \rangle$

R_{10}: $X \rightarrow \langle s_6 , \text{would return} \rangle$

R_{11}: $X \rightarrow \langle s_6 , \text{he would return} \rangle$

Keeping Track of All Derivations. CYK Grid (2)

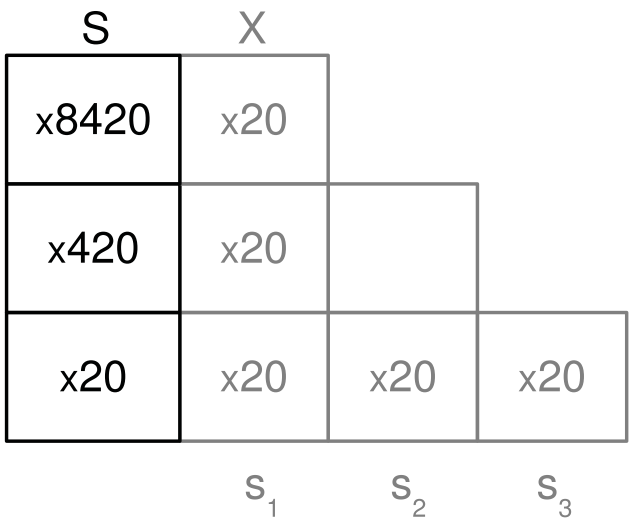


R_1: $S \rightarrow \langle X \text{ , } X \rangle$
R_2: $S \rightarrow \langle S\ X \text{ , } S\ X \rangle$
R_3: $X \rightarrow \langle s_1 \text{ , said} \rangle$
...
R_6: $X \rightarrow \langle s_2 \text{ , president} \rangle$
R_7: $X \rightarrow \langle s_3 \text{ , the Syrian} \rangle$
R_8: $X \rightarrow \langle s_4 \text{ , yesterday} \rangle$
R_9: $X \rightarrow \langle s_5 \text{ , that} \rangle$
R_{10}: $X \rightarrow \langle s_6 \text{ , would return} \rangle$
R_{11}: $X \rightarrow \langle s_6 \text{ , he would return} \rangle$
R_{14}: $X \rightarrow \langle X_1\ s_2\ X_2\ s_4\ s_5$,
y'day X_2 president X_1 that$\rangle$

Cube Pruning Algorithm [1]

- The number of derivations can be vast
- Each derivation will produce a translation candidate
- Each candidate has a score
- Find best candidate

$$\underset{t \in \mathcal{T}}{\text{argmax}}\ P(s|t)\ P(t)$$



- Cube-Pruning Algorithm
  - One-by-one processing of all derivations is not feasible
  - Lists of k-best hypotheses are kept in each cell (k=10^4)
  - Local decisions based on Translation and Language Model
  - ✓ Translation Model fits well in this grid representation
  - × Language Model does not: $P(t) = \prod_{n=1}^{I} p(t_n|t_{n-1})$

would return ← $p(return|would) \times p(would|?)$

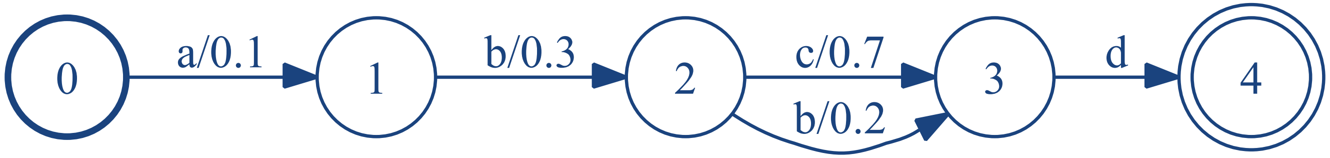
he would return ← $p(return|would) \times p(would|he) \times p(he|?)$

- In pruning, local decisions should be avoided!

[1] Chiang, D. 2005. A Hierarchical Phrase-Based Model for Statistical Machine Translation. Proc. ACL.

Weighted Finite-State Acceptors (WFSAs) [2]

- WFSAs are devices that compactly model a formal language
- A **Weighted Acceptor** of strings ‘a b c d’ and ‘a b b d’ :



is defined by a set of states Q and a set of arcs : $q \xrightarrow{x/w} q'$

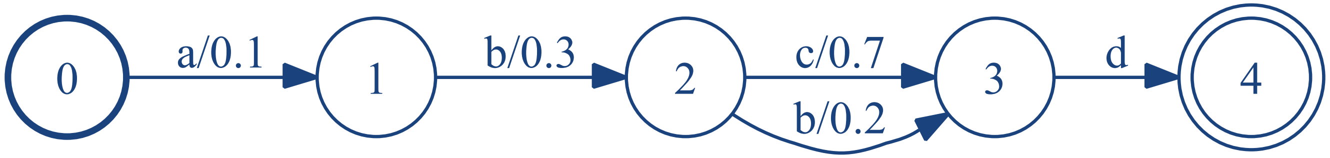
- Weighted Acceptors can assign costs to strings:
  - strings are associated with paths, which are sequences of arcs
  - weights are accumulated over paths by means of a **product operation** $\otimes$

$$w(p) = w(e_1) \otimes \cdots \otimes w(e_n)$$

[2] Mohri, Mehryar, Fernando Pereira, and Michael Riley. 2002. Weighted finite-state transducers in speech recognition. Computer Speech and Language, v16

Weighted Finite-State Acceptors (WFSAs)

- WFSAs are devices that compactly model a formal language
- A **Weighted Acceptor** of strings 'a b c d' and 'a b b d' :



is defined by a set of states Q and a set of arcs : $q \xrightarrow{x/w} q'$

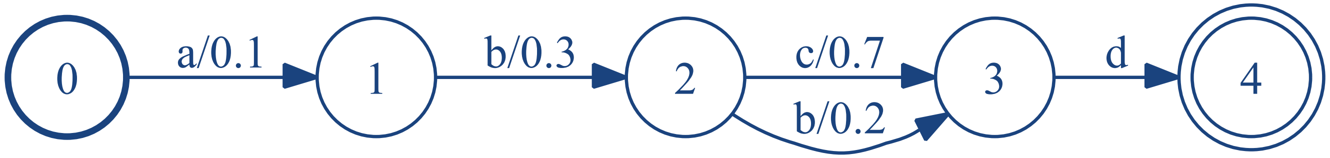
- Weighted Acceptors can assign costs to strings:
  - strings are associated with paths, which are sequences of arcs
  - weights are accumulated over paths by means of a **product operation** $\otimes$

$$w(p) = w(e_1) \otimes \cdots \otimes w(e_n)$$

Probability Semiring: $w(\text{'a b c d'}) = 0.1 \times 0.3 \times 0.7 \times 1.0 = 0.021$ ← BEST
$w(\text{'a b b d'}) = 0.1 \times 0.3 \times 0.2 \times 1.0 = 0.006$

Weighted Finite-State Acceptors (WFSAs)

- WFSAs are devices that compactly model a formal language
- A **Weighted Acceptor** of strings ‘a b c d’ and ‘a b b d’ :



is defined by a set of states Q and a set of arcs : $q \xrightarrow{x/w} q'$

- Weighted Acceptors can assign costs to strings:
  - strings are associated with paths, which are sequences of arcs
  - weights are accumulated over paths by means of a **product operation** $\otimes$

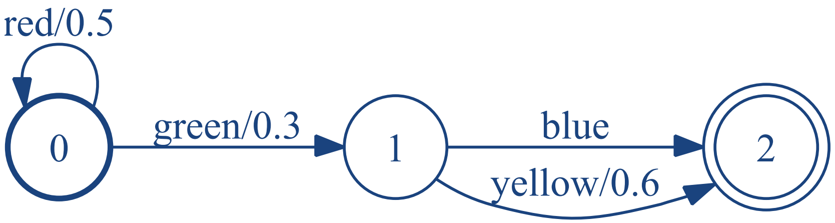
$$w(p) = w(e_1) \otimes \cdots \otimes w(e_n)$$

Tropical Semiring: $w(\text{‘a b c d’}) = 0.1 + 0.3 + 0.7 + 0.0 = 1.1$
$w(\text{‘a b b d’}) = 0.1 + 0.3 + 0.2 + 0.0 = 0.6 \leftarrow$ BEST

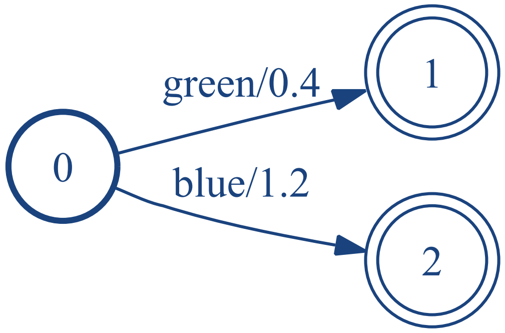
WFSA Operations - Union

A string x is accepted by $A = A \cup B$ if x is accepted by A or by B

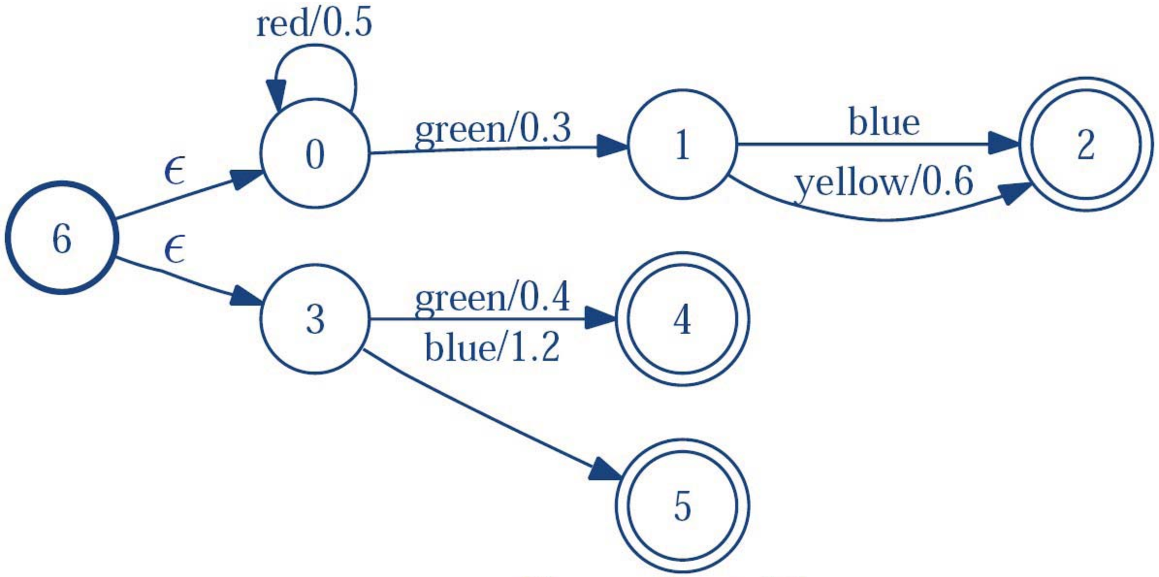
$$[\![C]\!](x) = [\![A]\!](x) \bigoplus [\![B]\!](x)$$



A



B

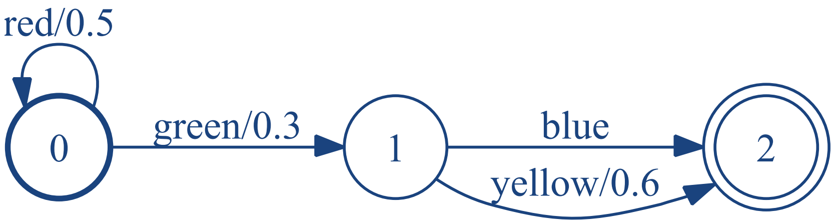


$C = A \cup B$

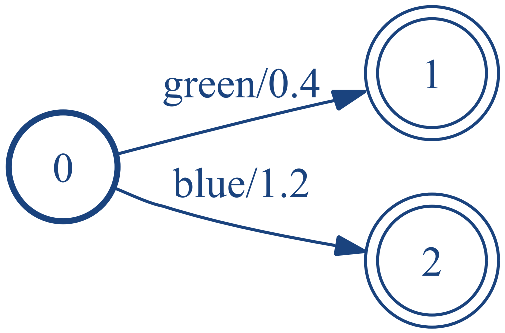
WFSA Operations - Concatenation (or Product)

A string x is accepted by $C = A \otimes B$ if x can be split into $x = x_1 x_2$ such that x_1 is accepted by A and x_2 is accepted by B

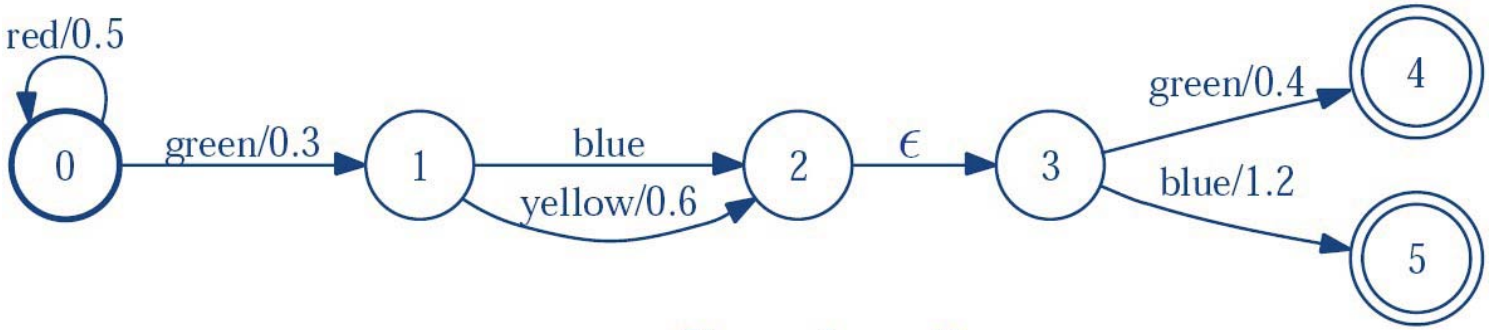
$$[\![C]\!](x) = \bigoplus_{x_1, x_2 :\, x = x_1 x_2} [\![A]\!](x_1) \otimes [\![B]\!](x_2)$$



A



B

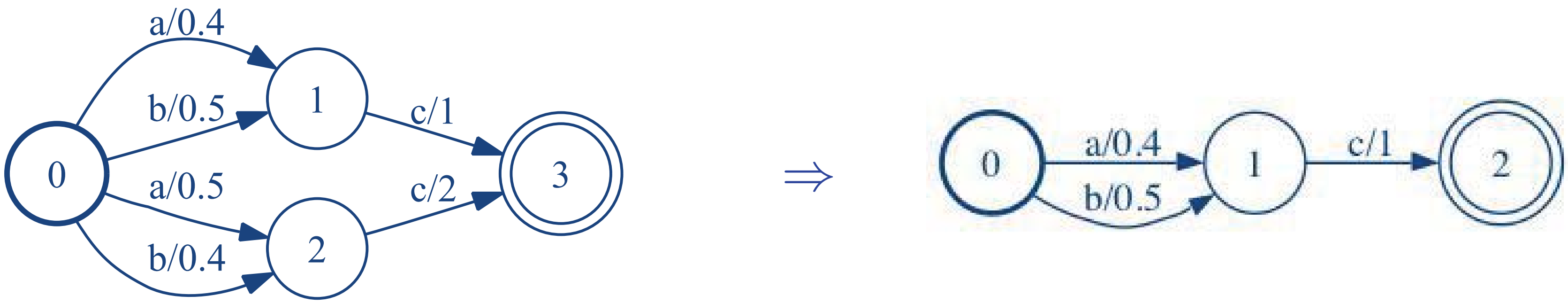


$C = A \otimes B$

WFSA Operations for Compactness

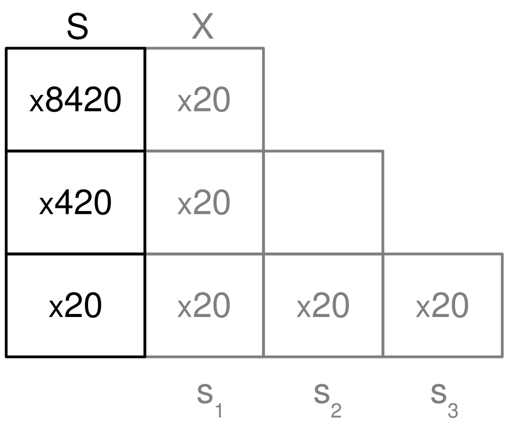
WFSAs can be **made compact** with operations that:

- reduce their size in number of states/arcs
- accept the same distinct strings
- *the cost of each string is respected* according to the semiring



- WFSAs can represent compactly very, very large numbers of paths
- Processing a WFSA is much faster than processing all of the paths individually

HiFST – Hierarchical Translation with WFSTs [3] [4]



S	X		
x8420	x20		
x420	x20		
x20	x20	x20	x20
	s_1	s_2	s_3

- Keep all possible derivations in each cell

  Efficiently explore largest $\mathcal{T}$ in

$$\underset{t \in \mathcal{T}}{\text{argmax}}\ P(s|t)\ P(t)$$

- **Build a WFSA in each cell**
  - They compactly store millions of paths with Translation Model costs
  - We can operate with them easily and faster than distinct hypotheses
  - Applying a Language Model to a WFSA is a well-established task
  - All parse information can be discarded

```
In each cell, do:

  For each rule in the cell:
    Build Rule WFSA by Concatenating target elements ( ⊗ )

  Build Cell WFSA by Unioning Rule WFSAs ( ⊕ )
```

[3] Iglesias, G. et al. 2009. Hierarchical Phrase-Based Translation with Weighted Finite State Transducers. Proc. of NAACL-HLT.

[4] de Gispert, et al. Hierarchical phrase-based translation with weighted finite state transducers and shallow-N grammars. Computational Linguistics, 36(3), 2010.

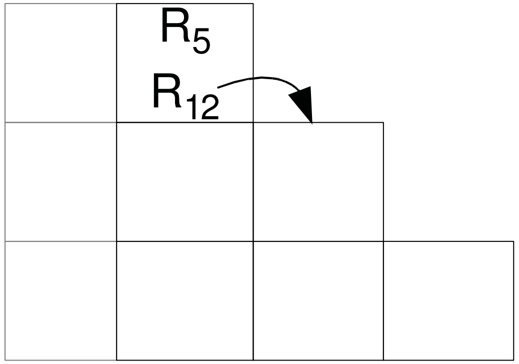
Building Rule WFSAs by Concatenation

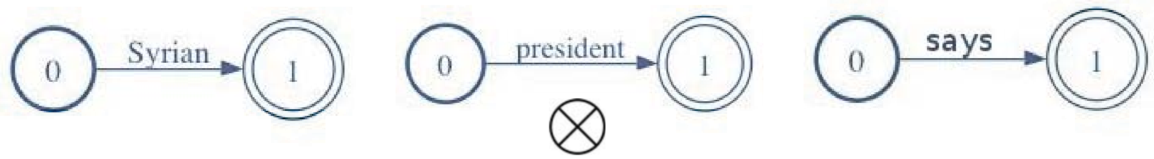
...

R_5: $\mathbf{X} \rightarrow \langle \mathbf{s_1\ s_2\ s_3}$ **, Syrian president says**$\rangle$

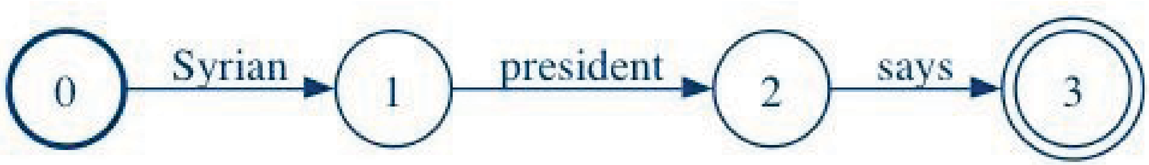
R_{12}: $\mathbf{X} \rightarrow \langle \mathbf{s_1\ X}$ **, X said**$\rangle$

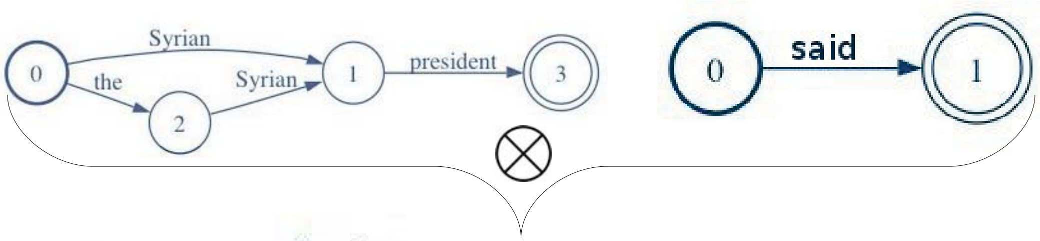
...



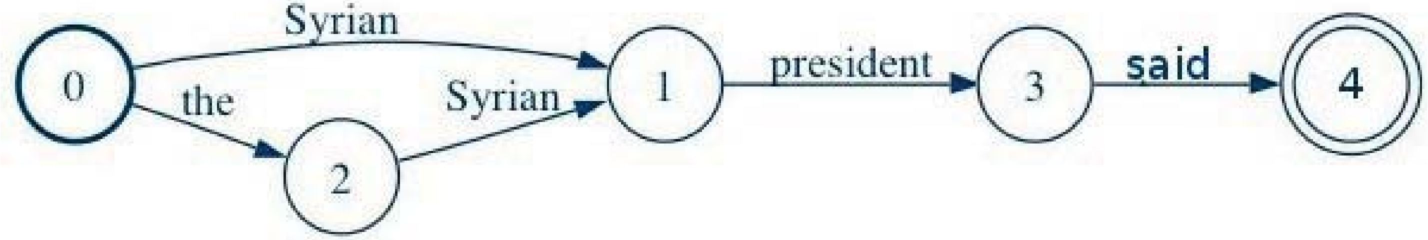


R_5 :



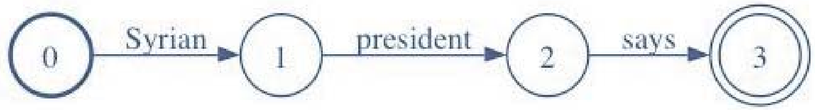


R_{12} :

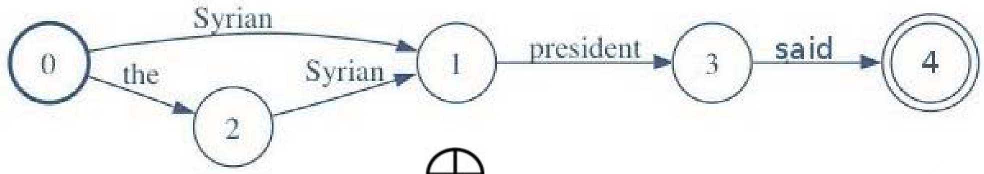


Building Cell WFSA by Union

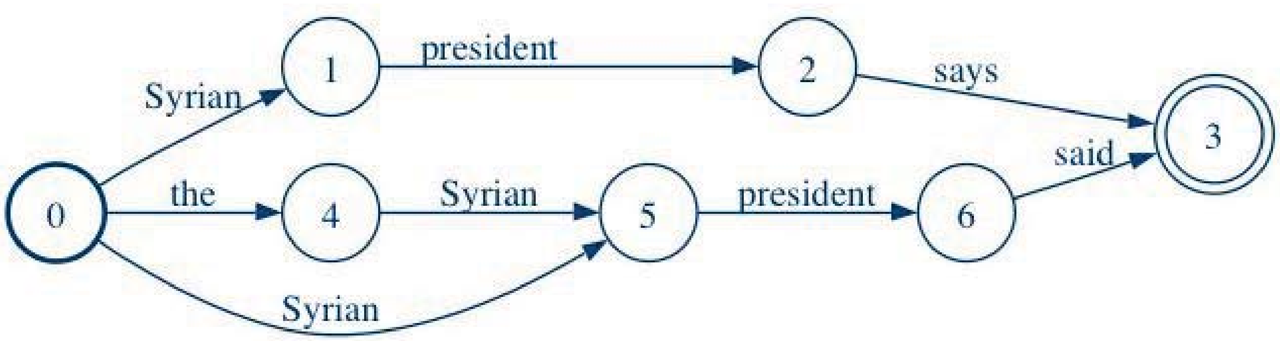
R_5 :



R_{12} :

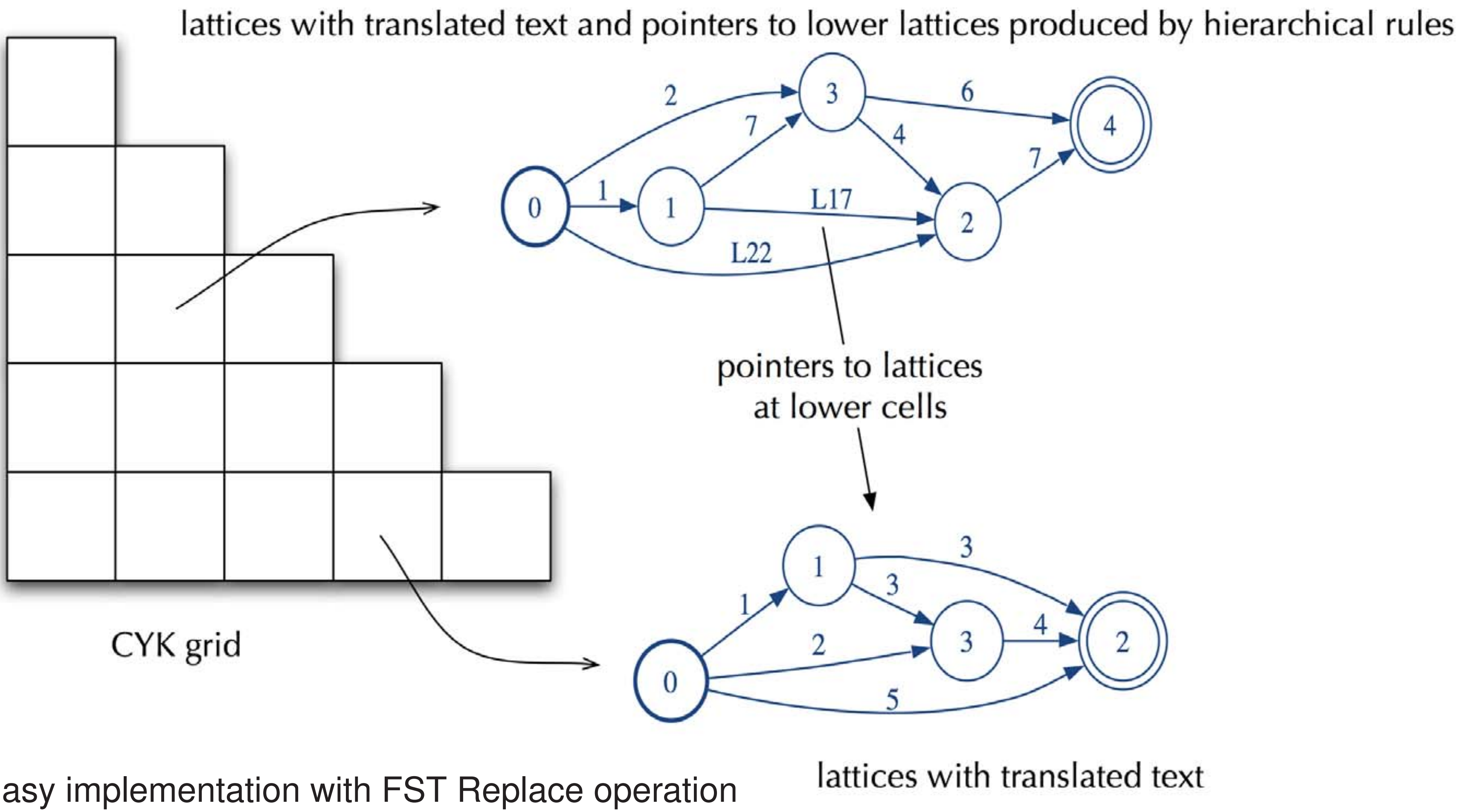


$\oplus$



- Can be made compact
- Target language model can be applied
- Search can be carried out efficiently

Delayed Translation


lattices with translated text and pointers to lower lattices produced by hierarchical rules
0
1
2
3
4
1
2
7
6
4
7
L17
L22
pointers to lattices
at lower cells
CYK grid
0
1
3
2
1
3
2
3
4
5
lattices with translated text

- ✓ Easy implementation with FST Replace operation
- ✓ Usual FST operations can be applied to skeleton → lattice size reduction

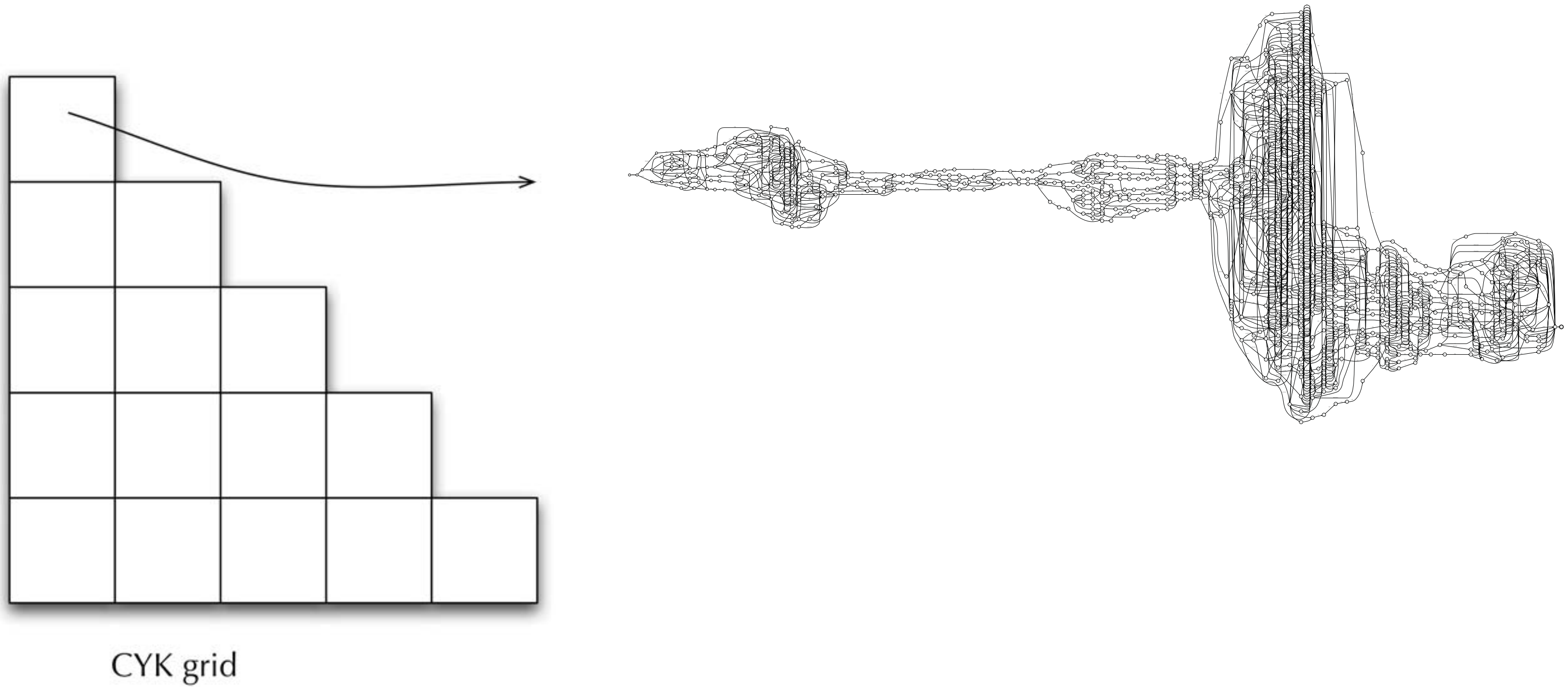
Output has the form of a Recursive Transition Network (RTN) [5] [6]

[5] Woods, W. A. 1970. Transition network grammars for natural language analysis. Commun. ACM, 13(10):591–606.

[6] Mohri, Mehryar. 1997. Finite-state transducers in language and speech processing. Computational Linguistics, v 23.

After expansion in the top cell, the result is a translation lattice

- Direct generation of translation lattices uses WFST operations
- Lattice contains **all permissible hypotheses** [7] under the grammar, with translation scores
  - Translation lattice is denoted $L(S, 1, J)$



Now need to apply the language model scores

[7] If no pruning is applied

Language Model Composition with $L(S, 1, J)$

LMs are usually large: e.g. zero cut-off, stupid back-off LM estimated over all available English

- Implement a backoff LM as an acceptor and apply it via composition

The following reviews a bigram implementation

$$\hat{P}(w_j|w_i) = \begin{cases} p(w_i, w_j) & f(w_i, w_j) > C \\ \alpha(w_i)\hat{P}(w_j) & \text{otherwise} \end{cases}$$

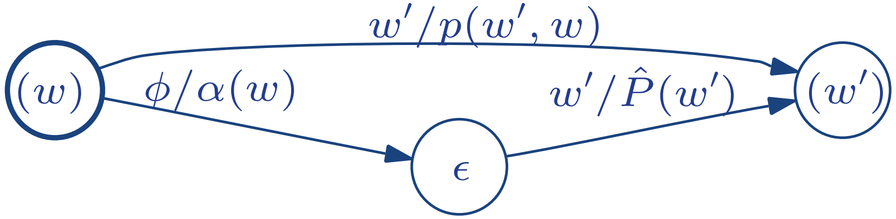
where $p(w_i, w_j) = d(f(w_i, w_j))\frac{f(w_i,w_j)}{f(w_i)}$.

- An arc for each pair of words w and w' for which $f(w, w') > C$: $(w) \xrightarrow{w'\,/\,p(w'|w)} (w')$

- A back-off arc from every word state (w) to the backoff state ϵ : $(w) \xrightarrow{\phi/\,\alpha(w)} \epsilon$

  - ‘failure transitions’ are crucial

- A unigram arc from the back-off state ϵ to every word state (w') : $\epsilon \xrightarrow{w'\,/\hat{P}(w')} (w')$



Approach developed for ASR ; very suitable for left-to-right application

- Generalized Algorithms for Constructing Statistical Language Models. C. Allauzen, et al. ACL’03

Language Model Composition with $L(S, 1, J)$ - LM Servers

A different approach:

1. Extract 3/4/5-grams from $L(S, 1, J)$ into a list
   - Can be done very efficiently using transducers
2. Query LM server(s) for probabilities $P(w_5 | w_1^4)$ for the n-grams in the list
   - We use Hadoop to serve raw n-gram counts
3. Build a *deterministic* transducer with arcs

$$(w_1^4) \xrightarrow{w_5 \,/\, p(w_5 | w_1^4)} (w_2^5)$$

   - only add states and arcs for the n-grams in the list
   - lower order histories starting with SENT_START are handled separately
   - deterministic WFST implementations of stupid backoff LMs are built on the fly
4. Apply the LM transducer at $L(S, 1, J)$ and prune based on complete scores

Makes use of the compact representation of all translation hyps in $L(S, 1, J)$

- Unlike ASR and similar problems, the hypothesis space can be created before the language model is applied
- Simpler topology: many states and arcs, but no need for failure transitions
- Separates the computation of back-off n-gram probabilities from the WFST implementation

Top Level Pruning and Pruning in Search

Final translation lattice $L(S, 1, J)$ typically requires pruning

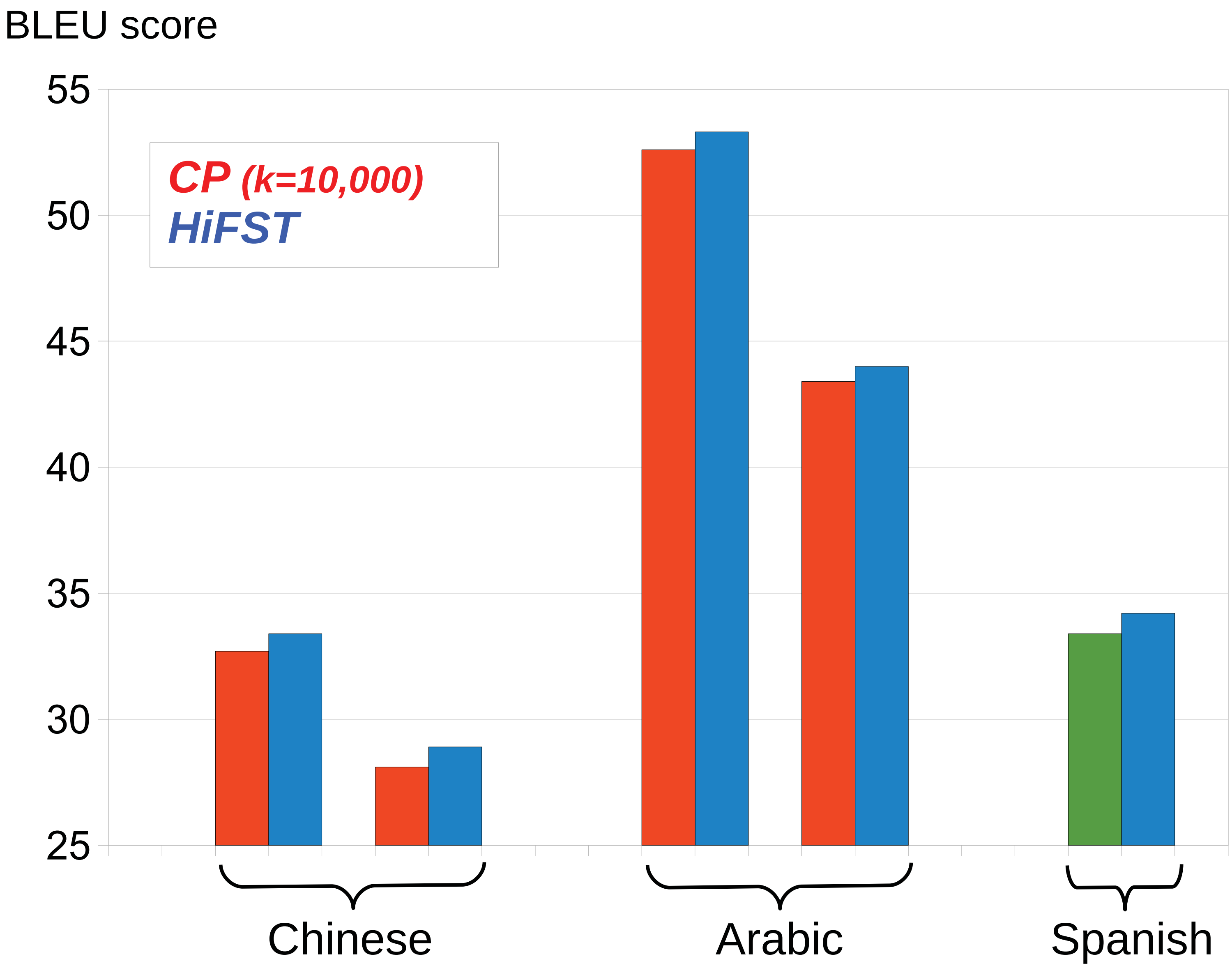
- Compose with target Language Model
- Perform likelihood-based pruning

Pruning in Search

- For a CYK cell, if number of states, non-terminal category and source span meet certain conditions, then:
  - ▷ Expand pointers in the translation lattice
  - ▷ Compose with the language model
  - ▷ Perform likelihood-based pruning of the resulting lattice
  - ▷ Remove language model
- Only required for certain language pairs, e.g. Chinese→English
- Goal is to prune in search only when sublattices get large
  - pruning triggers can vary based on cell height
- Use a ‘small’ LM too weak for translation but which is fine for pruning
  - Or use an appropriated sized grammar which doesn’t need pruning in search - more on this later

Much opportunity for improvement – need faster/better implementations

Translation Results into English. Contrast CP vs HiFST



Translation Results into English. Change in Semiring

✓ Changing the Semiring is easy

- Can have significant impact [8]

$$\text{Derivations } D : E, F \leftarrow D$$

- **Tropical Semiring**: Viterbi likelihood

  - single most likely derivation :

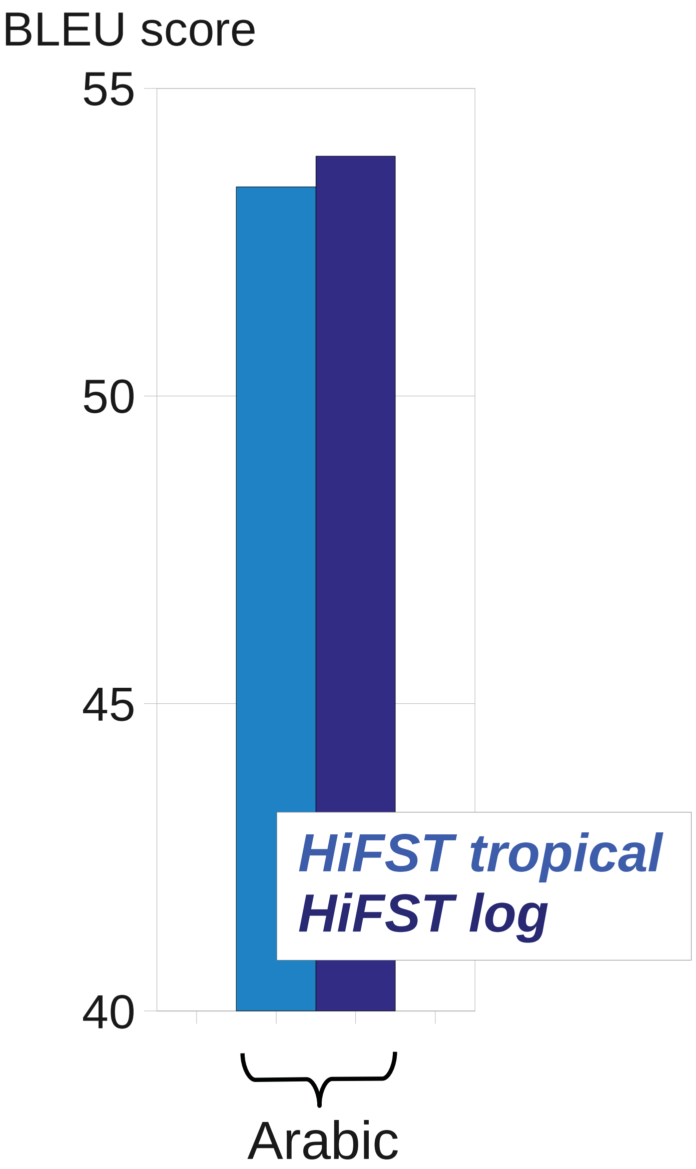
$$\underset{D:\, E,F \leftarrow D}{\text{argmax}} \; P(D|F)$$

- **Log Semiring**: Marginal probability

  - sum over all derivations :

$$\underset{E}{\text{argmax}} \sum_{D:\, E,F \leftarrow D} P(D|F)$$

✓ Additional gains with no extra programming effort



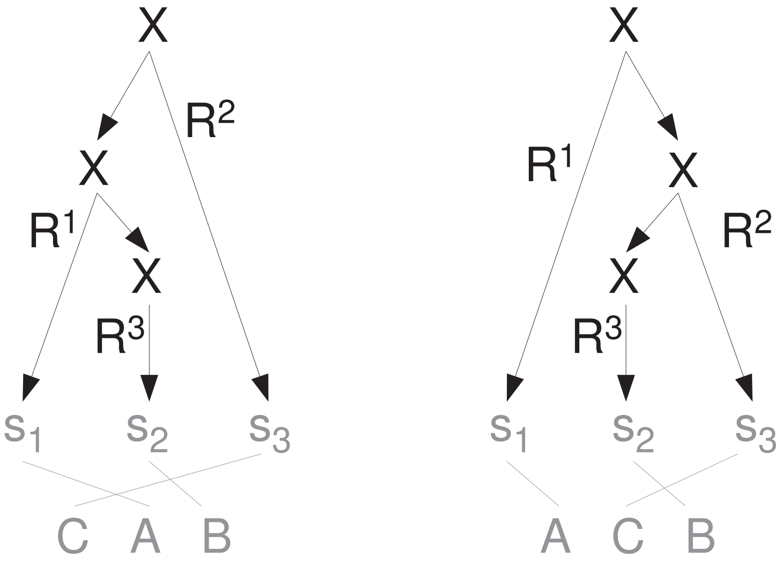
[8] P. Blunsom et al. A discriminative latent variable model for statistical machine translation. ACL-HLT, 2008

Translation Grammar Issues – Overgeneration

Overgeneration: different translations arising from the same set of rules

Translations of the source sequence $s_1\ s_2\ s_3$

R^1: $X \longrightarrow \langle s_1\ X, A\ X \rangle$
R^2: $X \longrightarrow \langle X\ s_3, X\ C \rangle$
R^3: $X \longrightarrow \langle s_2, B \rangle$



- not necessarily a bad thing in that new translations can be synthesized from rules extracted from training data
- a strong target language model, such as a high order n-gram, is typically relied upon to discard unsuitable hypotheses

Overgeneration complicates translation in that many hypotheses are introduced only to be subsequently discarded. These must be kept until the LM can be applied to discard them.

Translation Grammar Issues – Spurious Ambiguity

Spurious ambiguity (Chiang 2005):

> *... a situation where the decoder produces many derivations that are distinct yet have the same model feature vectors and give the same translation. This can result in n-best lists with very few different translations which is problematic for the minimum-error-rate training algorithm ...*

- This is due in part to the cube pruning procedure (Chiang 2007) which enumerates all distinct hypotheses to a fixed depth by means of k-best hypothesis lists.
- If enumeration was not necessary, or if the lists could be arbitrarily deep, there might still be many duplicate derivations but at least the hypothesis space would not be impoverished.

Translation Grammar Issues – Search Errors

- Any search procedure which relies on pruning during search is at risk of search errors
  - the risk is made worse if the grammars tend to introduce many similar scoring hypotheses, many similar hypotheses, or repeated instances of hypotheses
- We have found that cube pruning is very prone to search errors, i.e. the hypotheses produced by cube pruning are not the top scoring hypotheses which should be found under the Hiero grammar

Translation Grammar Issues – Grammars and Language Pairs

Not all language pairs require the full power of the full Hiero grammar

- it may be that no language pairs require the full power of the full Hiero grammar ...

Goal: For a given language pair, identify the (classes of) rules which matter

- interacts strongly with all previously mentioned issues:
  spurious ambiguity, overgeneration, search errors, ...

Current work not discussed: Automatic Induction of Hiero Grammars [9]

[9] A de Gispert, J Pino, and W Byrne. Hierarchical phrase-based translation grammars extracted from alignment posterior probabilities. EMNLP 2010.

Grammar configurations: Shallow grammars [10]

Goal: avoid nested hierarchical rules of non-terminals

full hierarchical grammar	
$S \rightarrow \langle X,X \rangle$	glue rule
$S \rightarrow \langle S\ X, S\ X \rangle$	glue rule
$X \rightarrow \langle \gamma, \alpha, \sim \rangle\ ,\ \gamma, \alpha \in \{X \cup \mathbf{T}\}^+$	hiero rules of any level

- For Arabic-to-English, shallow-1 grammar performs as full hiero - **but ~20× faster**
- Constrained search space, but can be built exactly and quickly - **no pruning required**

Shallow-N grammars include special sets of non-terminals to control the degree of nesting

shallow-1 grammar	
$\mathrm{S} \rightarrow \langle X^1, X^1 \rangle$	glue rule
$\mathrm{S} \rightarrow \langle \mathrm{S}\ X^1, \mathrm{S}\ X^1 \rangle$	glue rule
$X^1 \rightarrow \langle \gamma^0, \alpha^0, \sim \rangle\ ,\ \gamma^0, \alpha^0 \in \{\{X^0\} \cup \mathbf{T}\}^+$	hiero rules level 1
$X^0 \rightarrow \langle \gamma^p, \alpha^p \rangle\ ,\ \gamma^p, \alpha^p \in \mathbf{T}^+$	regular phrases

- Fast and effective for European language translation

[10] Iglesias, G. et al. 2009. Hierarchical Phrase-Based Translation with Weighted Finite State Transducers. Proc. of NAACL-HLT.

Fast Hiero Grammars for Chinese-English SMT

Baseline: full hierarchical grammar for the ZHEN task

- requires pruning in search (slow, search errors)

Idea: faster grammars with fewer rule patterns

- these faster grammars can be (almost) fully explored in decoding
- better balance between time and performance
- same decoder – grammar design determines speed and translation performance

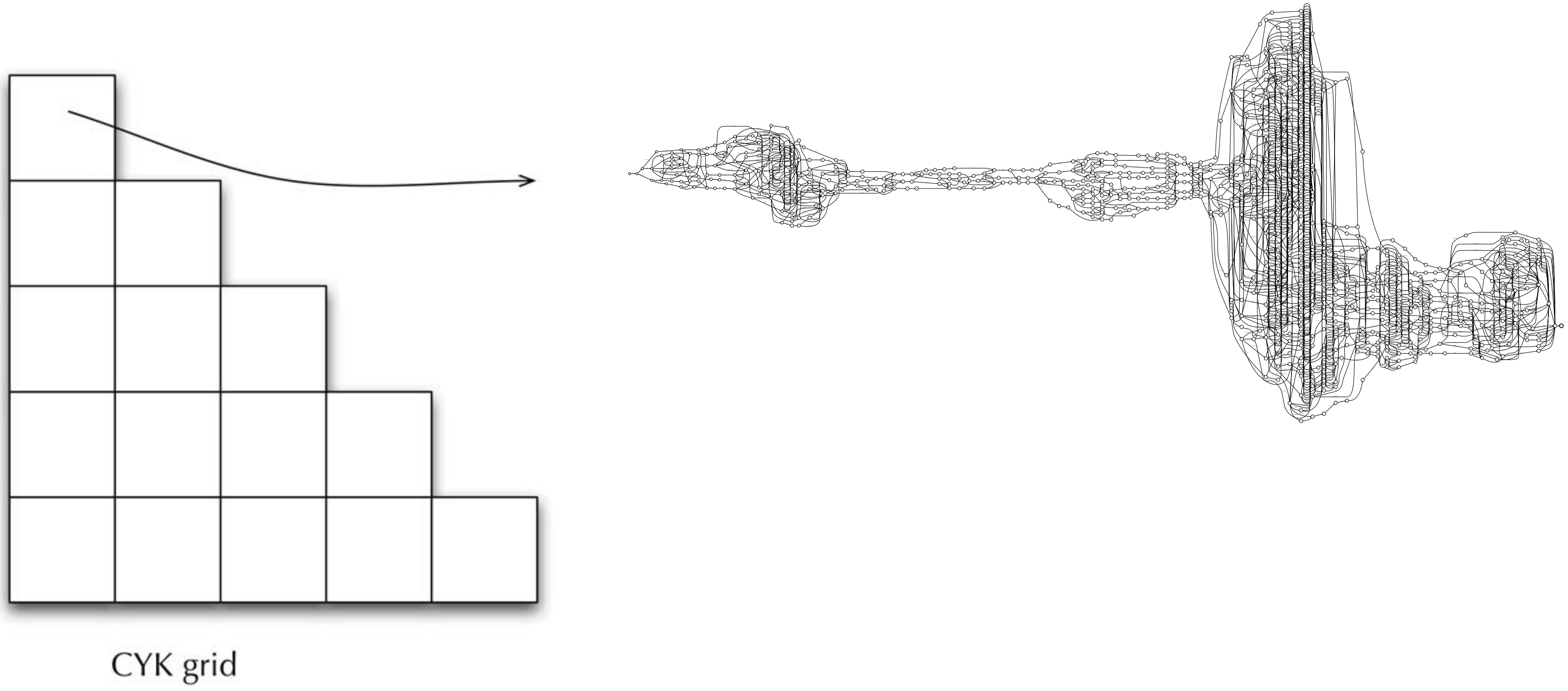
Preliminary results with Fast Hiero Grammars
(training size 2M sent. pairs)

	BLEU	time (seconds per word)	pruning (instances per word)
baseline	34.2	12.0	2.7
fast-G1	33.8	0.6	0.0
fast-G2	34.0	0.9	0.0
fast-G3	34.0	2.0	0.1
fast-G4	34.6	7.2	0.9

- Syntax-based SMT can be useful, even if language pairs are very similar
- Syntactic systems can be fast enough for many research problems

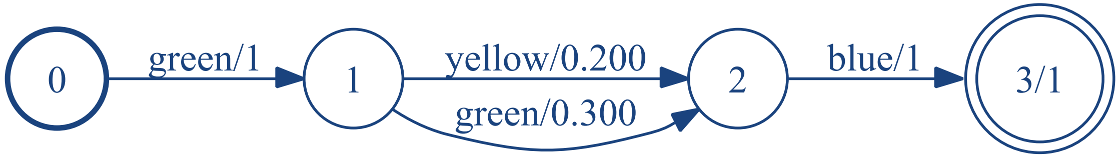
Translation lattice with translation and language model scores

- Lattice contains **all permissible hypotheses** [11] under the grammar
  - paths weights contain translation scores and language model scores



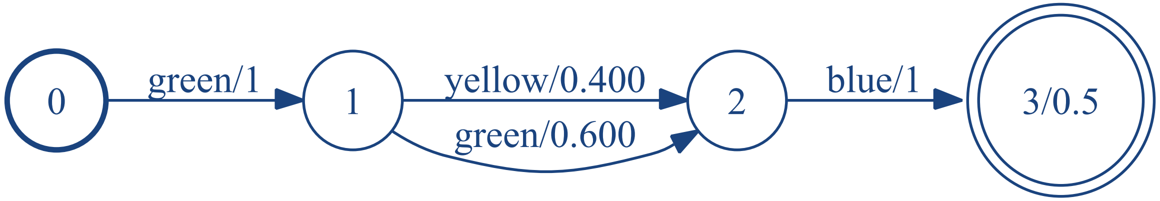
[11] If no pruning is applied

Weight Pushing – standard WFST operations



Weights are probabilities

Arc weights and labels can be moved so long as path weights are preserved :



Weight Pushing -- Real Semiring

- Final state has sum of all path weights
- Arc weights have posterior probabilities

Q: What is the posterior probability that any path contains ’yellow’ ?

A: 0.4

Translation Confidence Measures Derived from Lattice Posteriors [12] [13]

Source

Les actions de la bourse de sydney perdaient plus de cinq pour cent , mais finalement on réussit à réduire les pertes à à 4,3 pour cent .

Reference translation

Stocks on the market in sydney lost more than five percent , but ultimately lowered their losses to 4.3 percent .

ML 1-best

<s> the actions **of the sydney stock exchange lost** more than five percent , but finally we managed to reduce the losses to **4.3 percent . </s>**

Posterior probabilities assigned to 4-grams in the ML 1-best, computed by weight pushing:

0.33 <s> the actions of
0.33 the actions of the
0.32 actions of the sydney
0.69 of the sydney stock
0.73 the sydney stock exchange
0.63 sydney stock exchange lost
0.52 stock exchange lost more
0.53 exchange lost more than

0.58 lost more than five
0.43 more than five percent
0.41 than five percent ,
0.51 five percent , but
0.31 percent , but finally
0.23 , but finally we
0.19 but finally we managed
0.19 finally we managed to

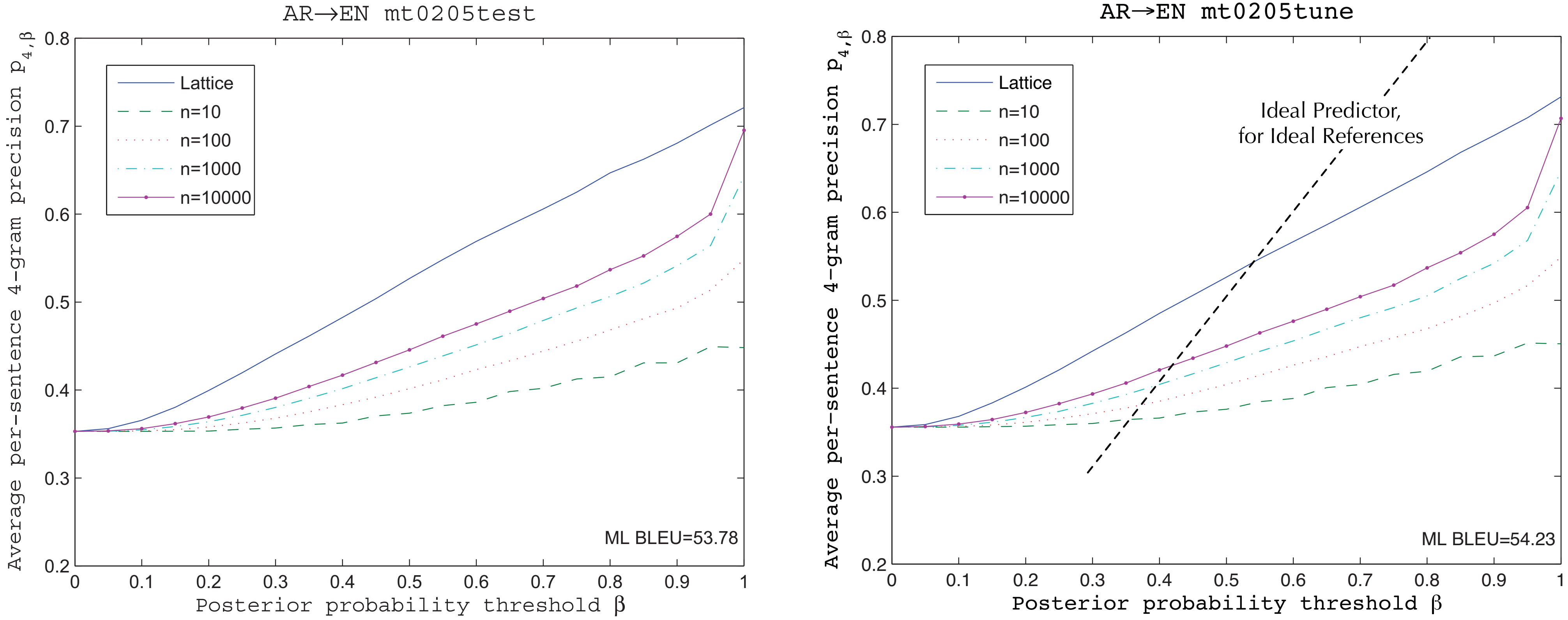
0.29 we managed to reduce
0.27 managed to reduce the
0.26 to reduce the losses
0.15 reduce the losses to
0.15 the losses to 4.3
0.25 losses to 4.3 percent
0.35 to 4.3 percent .
0.77 4.3 percent . </s>

[12] G. Blackwood et al. Fluency constraints for minimum Bayes-risk decoding of statistical machine translation lattices. COLING, 2010.

[13] de Gispert, et al. Hierarchical phrase-based translation with weighted finite state transducers and shallow-N grammars. Computational Linguistics, 36(3), 2010.

Reliability of N-gram Posterior Distributions – lattices vs k-best lists

Precision of 4-grams in the translation hypotheses as a function of their posterior



- High posterior n-grams are more likely to be found in the references
- Using the full evidence space of the lattice is much better than even very large k-best lists for computing posterior probabilities
- Can be useful for translation confidence measures

Minimum Bayes Risk Decoding

- Alternative to maximum likelihood decoding
  - Related to ASR techniques such as ROVER, consensus decoding, ...
- MBR search over the space of hypotheses $\mathcal{H}$ [14]

$$\widehat{E}_{MBR} = \underset{E' \in \mathcal{H}}{\text{argmin}} \sum_{E \in \mathcal{E}} L(E, E')P(E|F)$$

- Goal: find the minimum risk hypothesis under the translation grammar $P(E|F)$

Requires the posterior distribution $P(E|F)$ provided by the SMT grammar to be reliable

- Direct generation of SMT lattices gives improvements relative to N-Best lists

$L(E, E')$ is the (negative of) the BLEU score. We use a series approximation to it, for tractability.

[14] S. Kumar and W. Byrne. *Minimum Bayes-risk decoding for statistical machine translaton.* NAACL 2004.

LMBR Decoding Procedures

Linear approximation allows search over lattices, rather than n-best lists [15]

$$\widehat{E} = \operatorname*{argmax}_{E' \in \mathcal{E}} \left\{ \theta_0 |E'| + \sum_{k=1}^{4} \sum_{u \in \mathcal{N}_k} \theta_u \#_u(E') p(u|\mathcal{E}) \right\} \tag{1}$$

- $\mathcal{E}$ is a translation lattice and $\mathcal{N}_k$ are its n-grams of order k
- total probability of all paths in $\mathcal{E}$ which contain u : $p(u|\mathcal{E}) = \sum_{E:\#_u(E)>0} P(E|F)$

An approximation – $c(u|\mathcal{E}) = \sum_E \#_u(E)\, P(E|F)$ [16] :

- $p(u|\mathcal{E})$ is difficult to compute
- $c(u|\mathcal{E})$ is relatively easy to compute
- Reasonable assumption: $p(u|\mathcal{E}) \approx c(u|\mathcal{E})$ for u of order 2 and higher

Goal: compute Equation 1 exactly and efficiently using WFSTs [17]

[15] R. Tromble, S. Kumar, F. Och, and W. Macherey. 2008. Lattice Minimum Bayes-Risk Decoding for Statistical Machine Translation. Proc EMNLP

[16] C. Allauzen, S. Kumar, W. Macherey, M. Mohri, and M. Riley. 2010. Expected sequence similarity maximization. Proc. HLT.

[17] G. Blackwood A. de Gispert, W. Byrne. 2010. Efficient path counting transducers for minimum Bayes-risk decoding of statistical machine translation lattices. ACL, 2010.

Hiero CubePruning (HCP) vs HiFST – ArEn Shallow-1

Contrastive translation results (lower-cased IBM BLEU) after first-pass decoding and subsequent rescoring steps.

	decoder	*mt02-05-tune*	*mt02-05-test*	*mt08*
a	HCP	52.5	51.9	42.8
	+5g	53.4	52.9	43.5
	+5g+MBR	53.6	53.0	43.6
b	HiFST	52.5	51.9	42.8
	+5g	53.6	53.2	43.9
	+5g+LMBR	54.3	53.7	44.8
Decoding time in secs/word: 1.1 for HCP; 0.5 for HiFST.				

- Decoding time reported for *mt02-05-tune*
- Both systems are optimized using MERT over the k-best lists generated by HCP
- *Search errors:* HCP fails to find the optimum translation in 18.5% of the hyps

Resources:

- NIST MT09 Arabic Constrained Data track ($\sim$150M words per language)
- First-pass 4-gram LM estimated over 0.9B words of English
- Zero-cutoff stupid-backoff 5-gram LM estimated over 6.6B words of English
- Exact computation of LMBR

LMBR– N-Gram Counting Trick

Computing $p(u|\mathcal{E})$ for $u \in \mathcal{N}_1$ (unigrams) is relatively easy (next slide)

- operationally more difficult for the higher order n-grams - machines get a bit complex

Trick: [18] [19]

1. Build a transducer that maps word sequences to n-gram sequences (for a fixed n)
   For $u = w_1^k$ and $u \in \mathcal{N}_k$:
   $$(w_1^{k-1}) \xrightarrow{w_k:u} (w_2^k)$$
2. Transform the word lattices to n-gram lattices ; path probabilities are preserved
   $$\ldots w_1\ w_2\ w_3\ w_4\ w_5 \ldots \longrightarrow \ldots \underbrace{w_1\_w_2\_w_3}_{u_1}\ \underbrace{w_2\_w_3\_w_4}_{u_2}\ \underbrace{w_3\_w_4\_w_5}_{u_3} \ldots$$
3. Count ‘unigrams’ in the n-gram lattices

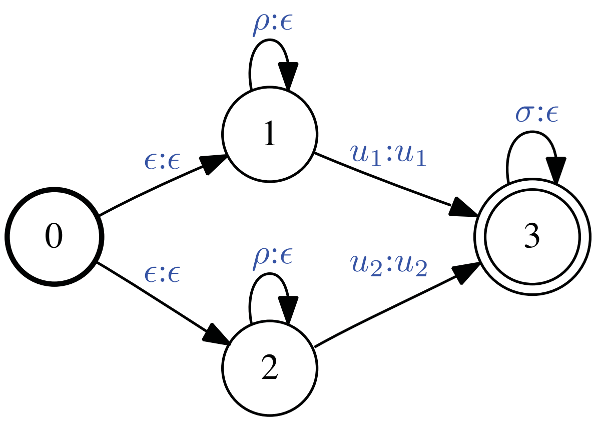
Easier to count unigrams in n-gram lattices than n-grams in unigram lattices

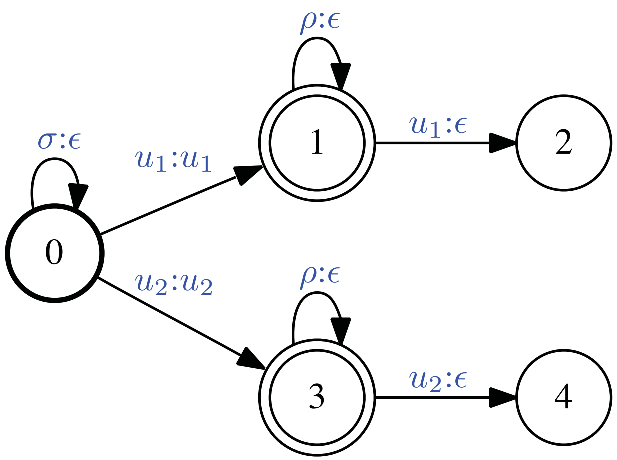
Can use the simple architecture of the unigram counting machine for all order n-grams

[18] G. Blackwood A. de Gispert, W. Byrne. 2010. Efficient path counting transducers for minimum Bayes-risk decoding of statistical machine translation lattices. ACL, 2010.

[19] G. Blackwood. *Lattice Rescoring Methods for Statistical Machine Translation*. CUED PhD Thesis. 2010

LMBR– Efficient Path Counting Transducers

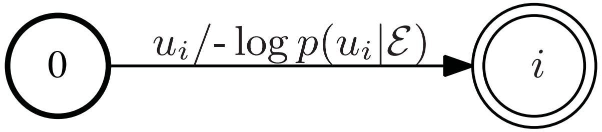




Right: Ψ_n^R deletes all but the ***last*** occurrence of u_1 or u_2 on every path[19]

Left: Ψ_n^L deletes all but the ***first*** occurrence of u_1 or u_2 on every path[20]

After composition with either Ψ_n in the log semiring, output projection, ϵ-removal, determinizing, minimizing, and weight pushing, the output machine has the following form:



- BUT these operations can be done *much* faster with Ψ_n^R than with Ψ_n^L

[19] G. Blackwood. *Lattice Rescoring Methods for Statistical Machine Translation*. CUED PhD Thesis. 2010

[20] C. Allauzen et al. HLT 2010.

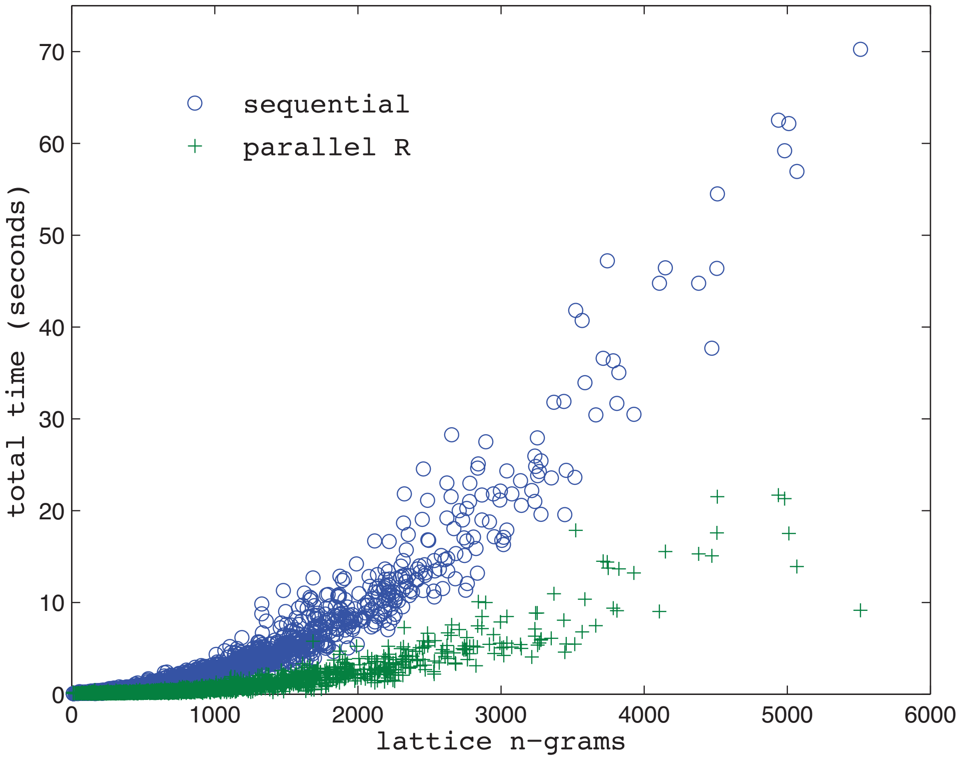
Exact Implementation of the Linear LMBR Approximation

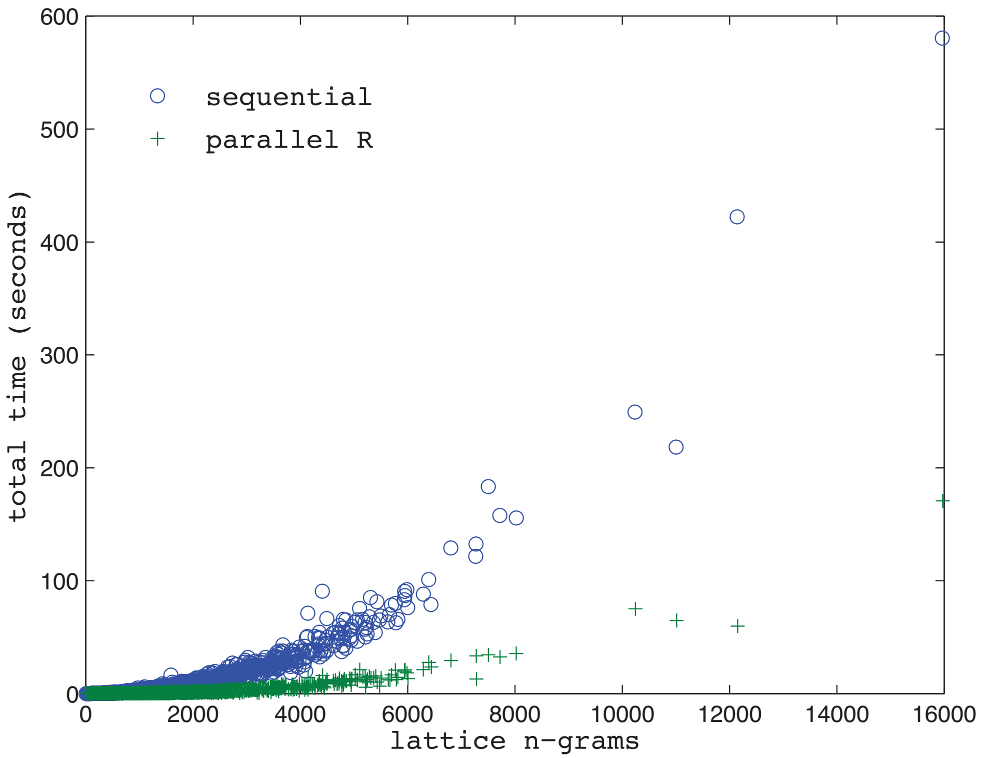
Time in seconds for LMBR – all with exact realisations of Equation 1

		Ar→En		Zh→En	
		mt08nw	mt08ng	mt08nw	mt08ng
Total	sequential	4437	8085	28506	5922
	Ψ_n^L	4495	9199	20341	4554
	Ψ_n^R	1468	3149	5157	1047

- Allows for a (relatively) speedy and exact implementation of LMBR with WFSTs
  - easy generation of lattices, etc.
- Avoids an $\sim$0.2 BLEU degradation in using $c(u|\mathcal{E})$ rather than $p(u|\mathcal{E})$ (in ArEn)
  - $p(u|\mathcal{E}) \approx c(u|\mathcal{E})$ holds for u of order 3 and higher but should use $p(u|\mathcal{E})$ for order 1 and 2

Total decoding time (secs) vs. number of lattice n-grams: ArEn & ZhEn





Summary of Recent Work

Good performance in recent MT evaluations

- NIST MT09 Arabic-English, common data track [21]
  – A de Gispert, G Iglesias, G Blackwood, J Brunning, and B Byrne
  - Top system in the Single System Track
  - Top system in the System Combination Track
- WMT 2010 [22] (BLEU scores)
  - Top Spanish→English single system – Gonzalo Iglesias
  - Top French→English single system – Juan Pino
  - 2^{nd} place single system in English→French – Juan Pino
  - 3^{rd} place single system in English→Spanish – Gonzalo Iglesias

Current focus

- Improving speed and throughput without loss of quality
- Chinese→English
- Translation out of English into European languages

[21] http://www.nist.gov/speech/tests/mt/2008/doc/mt08_official_results_v0.html

[22] matrix.statmt.org/matrix

Conclusions

An efficient decoder makes some modelling issues less of a problem

- spurious ambiguity, overgeneration, search errors cause less trouble

Exact implementation of PSCFGs (Hiero)

- RTNs are used to compactly store hypotheses in Delayed Translation

Hiero translation lattices are rich search spaces for subsequent rescoring procedures

- Hiero lattices are very well suited for LMBR and large n-gram rescoring

Some current research topics not discussed

- Grammar induction, discriminative training, verb movement...
- improved parallel text alignment procedures
- incorporation of natural language generation into SMT

Work supported in part by

- GALE program of the Defense Advanced Research Projects Agency, C.No. HR0011-06-C-0022
- European Community's Seventh Framework Programme (FP7/2007-2013), grant agreement 213845 (EMIME) and grant agreement 247762 (FAUST).
- PhD research of G. Iglesias supported by Spanish Government research grant BES-2007-15956 (project TEC2006-13694-C03-03).

Thanks to Michael Riley and Cyril Allauzen of Google, Inc. for help with OpenFST.