Universitat Politècnica de València

Máster en Inteligencia Artificial, Reconocimiento de Formas e Imagen Digital

MACHINE TRANSLATION

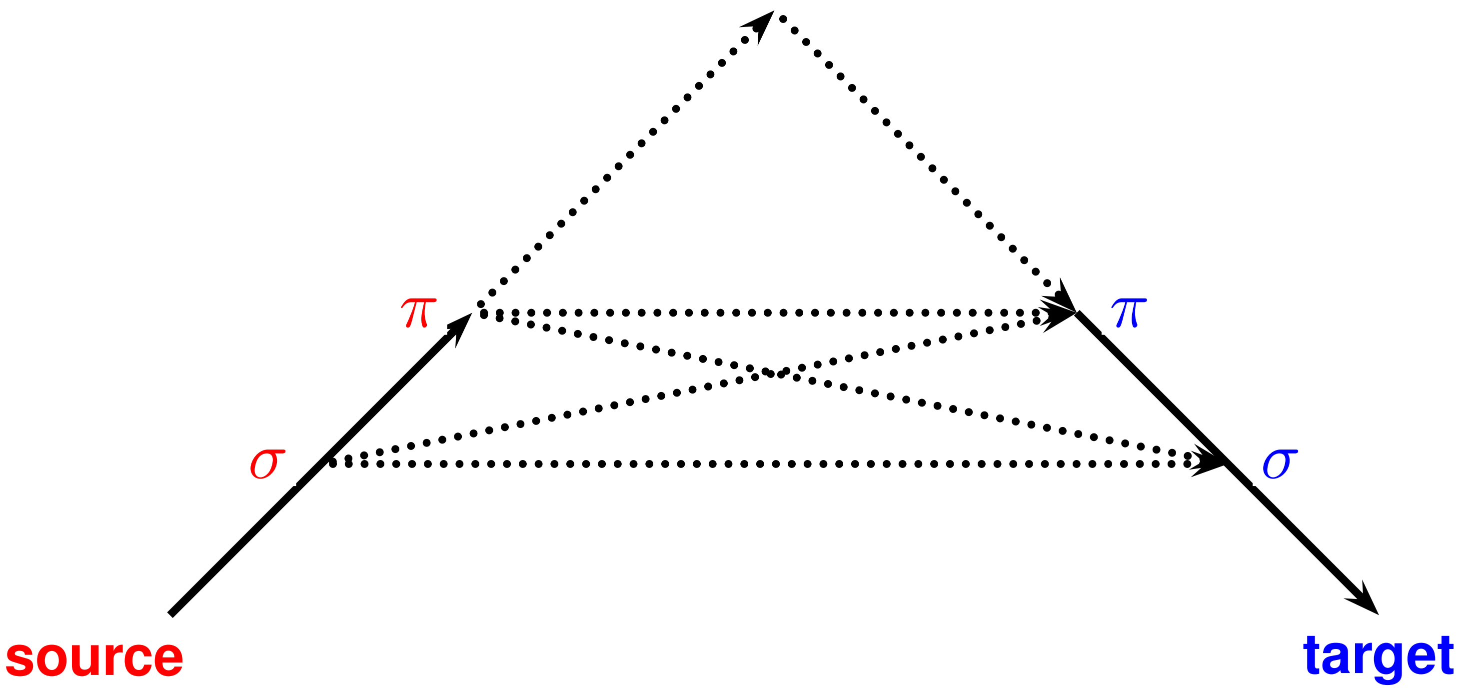
Syntax-based models

Introduction

Joan Andreu Sánchez

jandreu@prhlt.upv.es

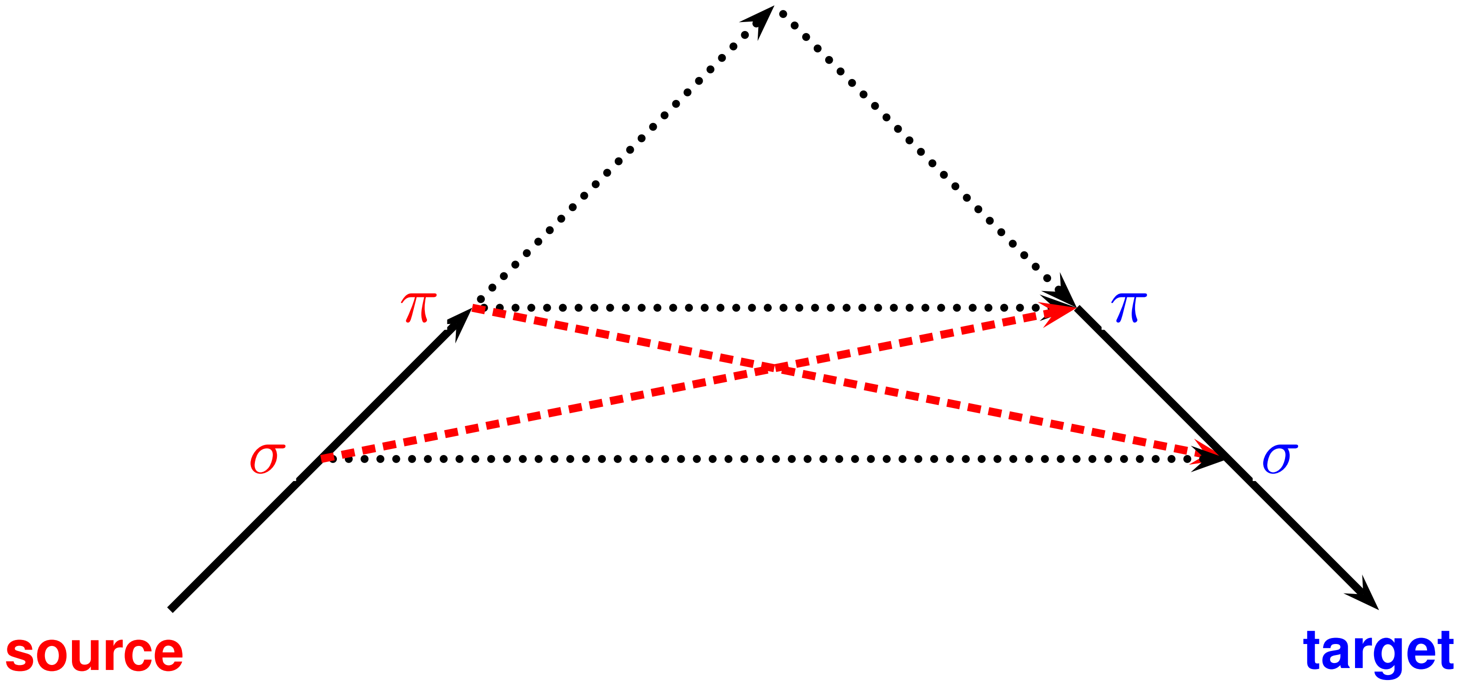
Levels of representation in Machine Translation



- $\pi \rightarrow \sigma$: tree-to-string
- $\sigma \rightarrow \sigma$: string-to-string
- $\sigma \rightarrow \pi$: string-to-tree

Which is the appropriate level of representation? Research field

Levels of representation in Machine Translation

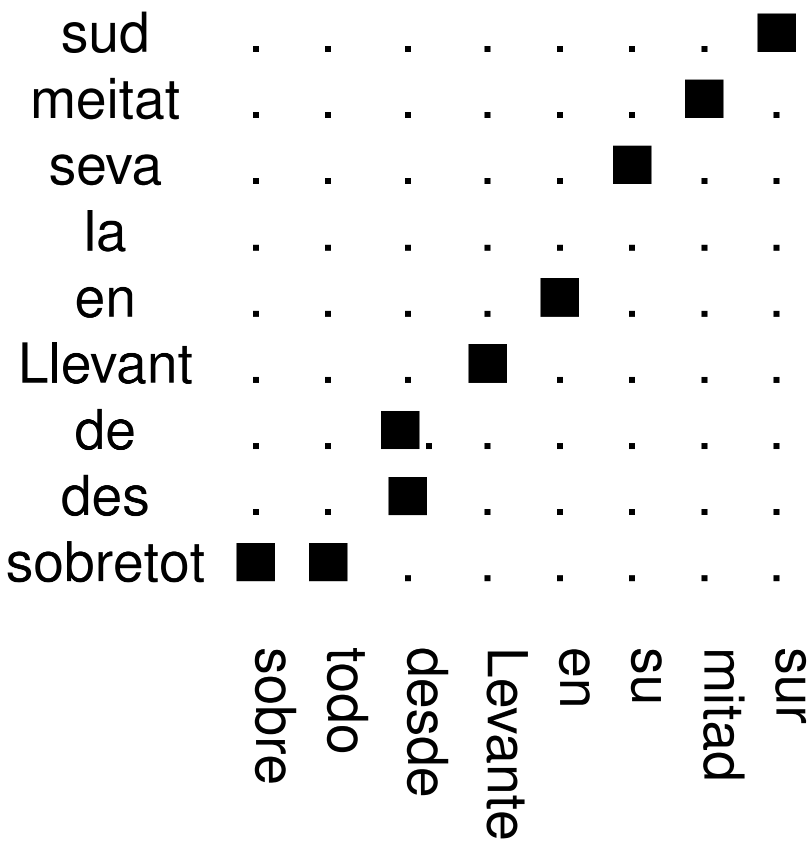


- $\pi \rightarrow \sigma$: **tree-to-string**
- $\sigma \rightarrow \sigma$: string-to-string
- $\sigma \rightarrow \pi$: **string-to-tree**

This part will focus on approaches in boldface

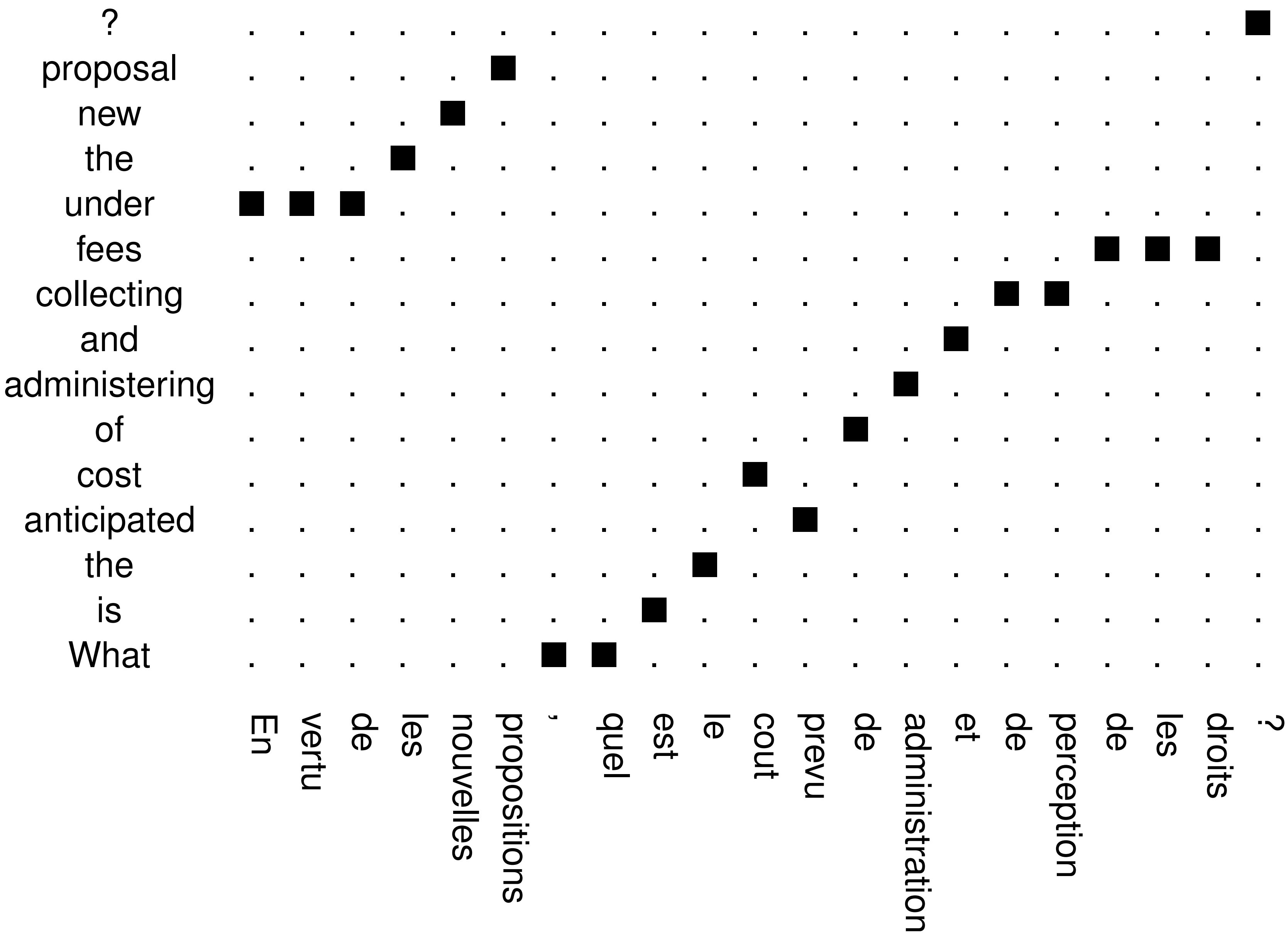
Exemple of word alignments: monotone translation

METEO corpus



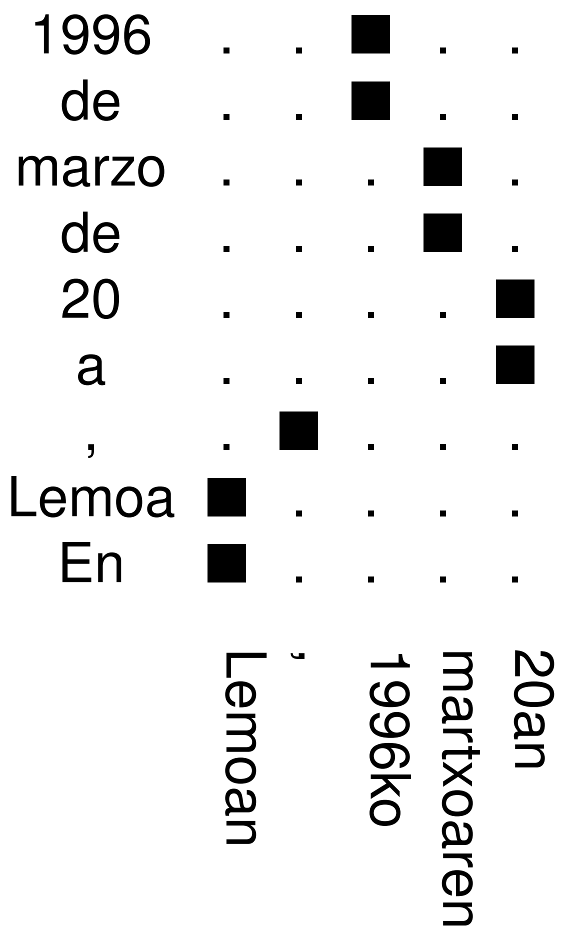
Exemple of word alignments: non-monotone translation parts

H. Ney, *Statistical Natural Language Processing*, 2003: Canadian Hansards

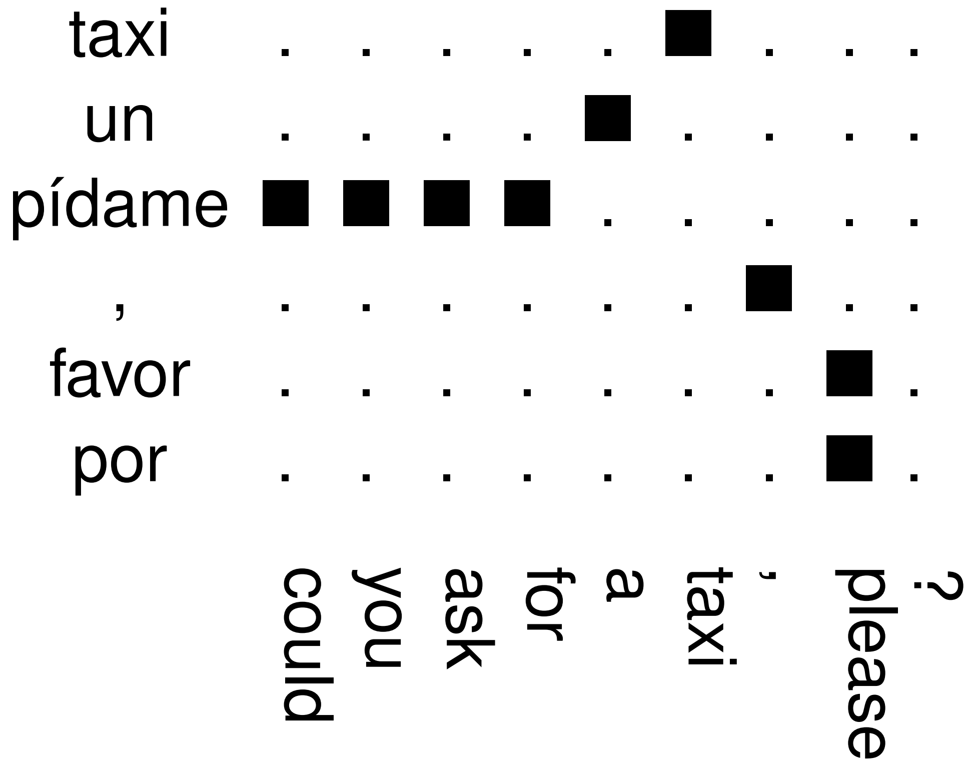


Exemple of word alignments: non-monotone translation parts

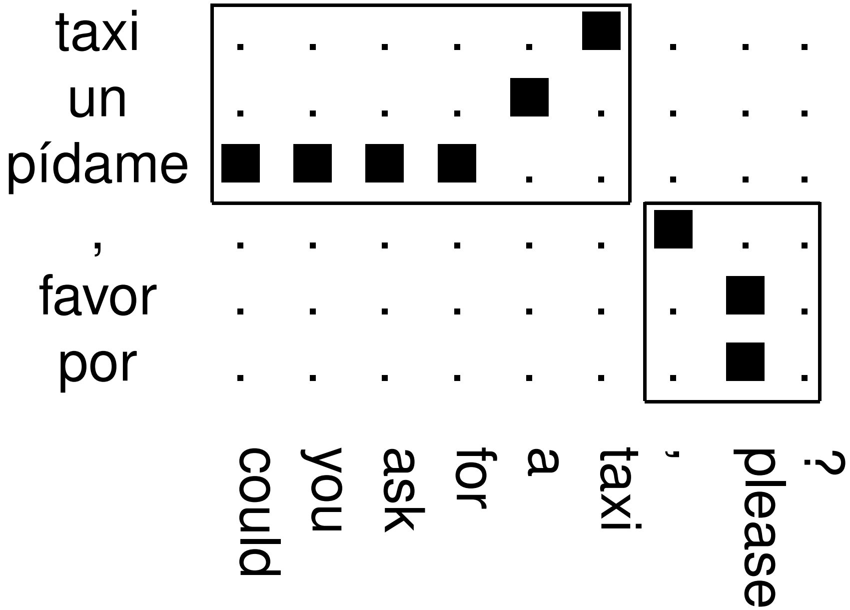
AMETRA corpus

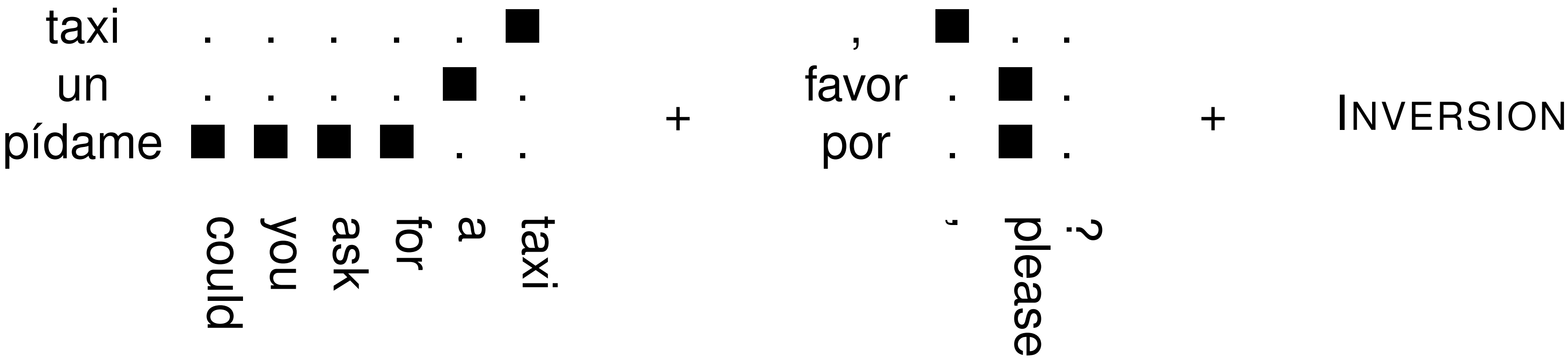


TURIST corpus

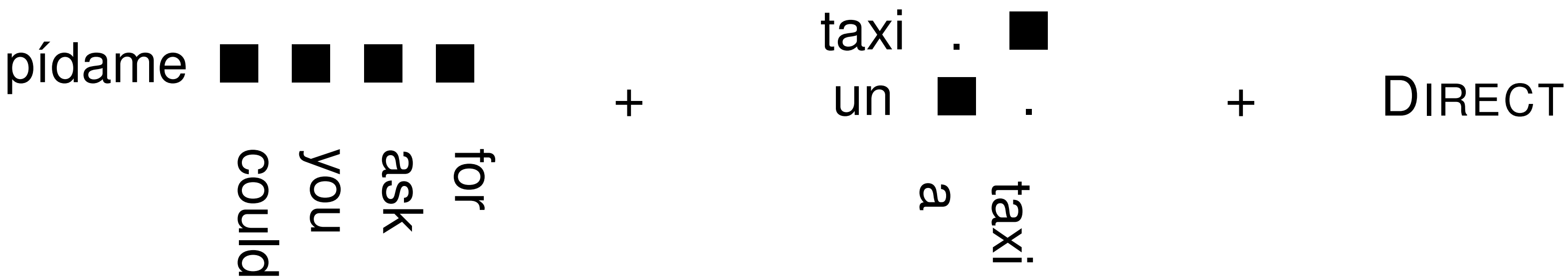


Example of word alignments

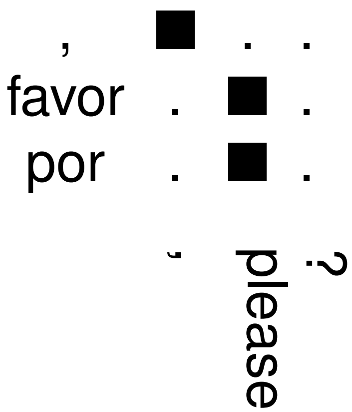


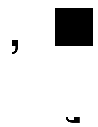


Example of word alignments

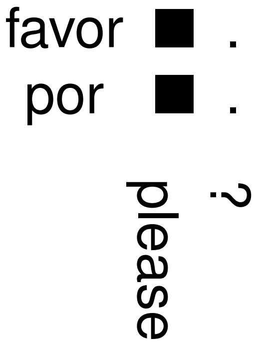


Example of word alignments



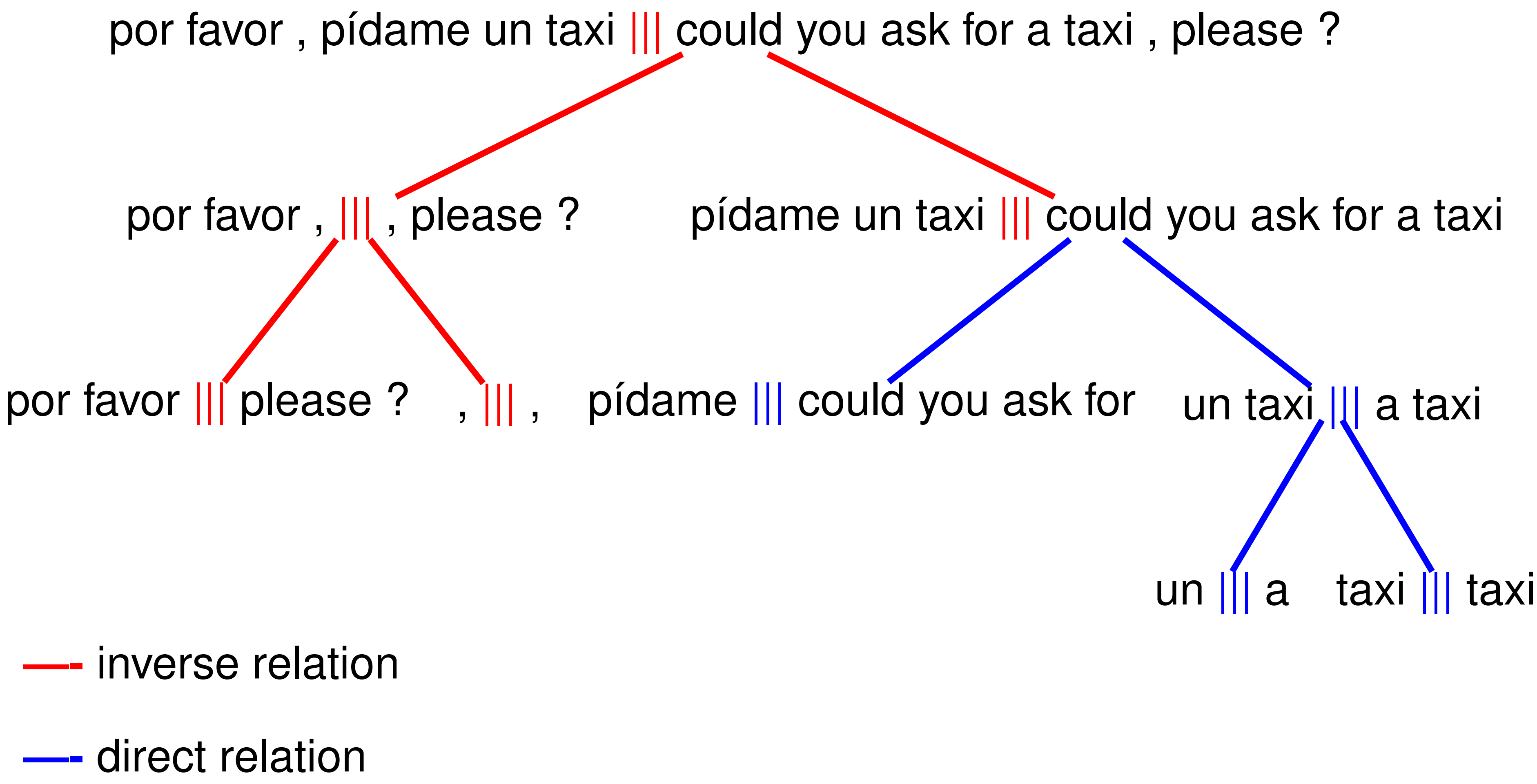


+



+ INVERSION

Example of word alignments



Syntax in MT

Example SCFG*

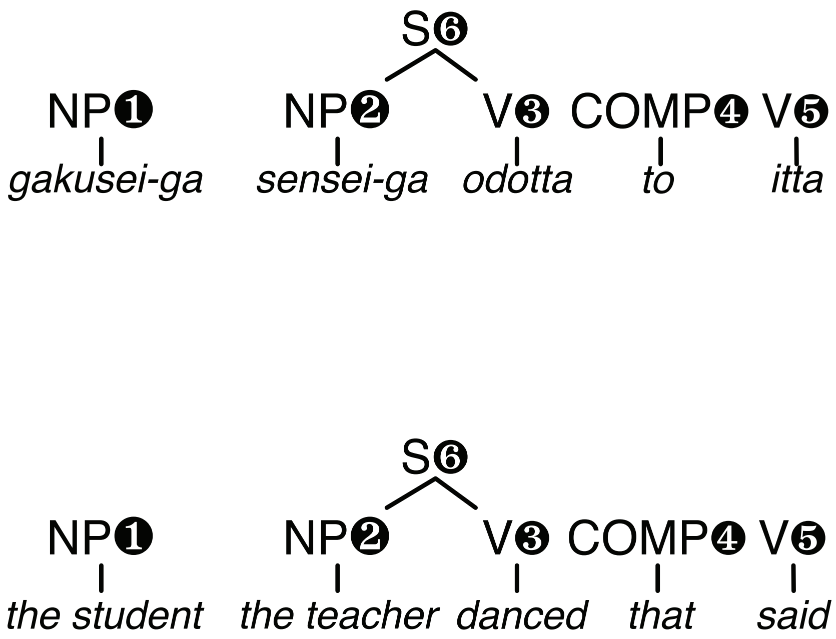
	Japanese	English
S →	NP① VP②	NP① VP②
S'→	S① COMP②	COMP② S①
VP →	NP① V②	V② NP①
NP →	*gakusei-ga*	*student*
NP →	*sensei-ga*	*teacher*
V →	*odotta*	*danced*
V →	*itta*	*said*
COMP →	*to*	*that*

* Source: `http://www.mt-archive.info/MTMarathon-2009-Li-ppt.pdf`

Syntax in MT

Example SCFG

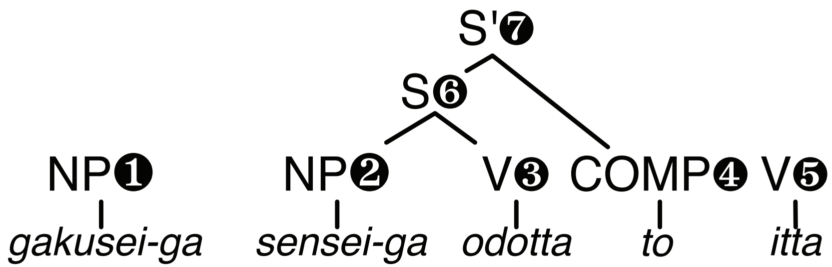
	Japanese	English
S →	NP① VP②	NP① VP②
S'→	S① COMP②	COMP② S①
VP →	NP① V②	V② NP①
NP →	*gakusei-ga*	*student*
NP →	*sensei-ga*	*teacher*
V →	*odotta*	*danced*
V →	*itta*	*said*
COMP →	*to*	*that*

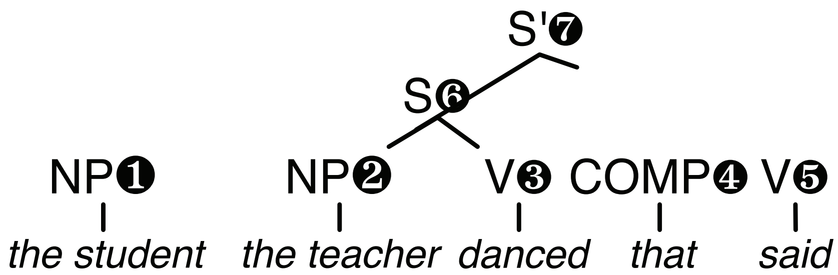


Syntax in MT

Example SCFG

	Japanese	English
S →	NP① VP②	NP① VP②
S'→	S① COMP②	COMP② S①
VP →	NP① V②	V② NP①
NP →	*gakusei-ga*	*student*
NP →	*sensei-ga*	*teacher*
V →	*odotta*	*danced*
V →	*itta*	*said*
COMP →	*to*	*that*

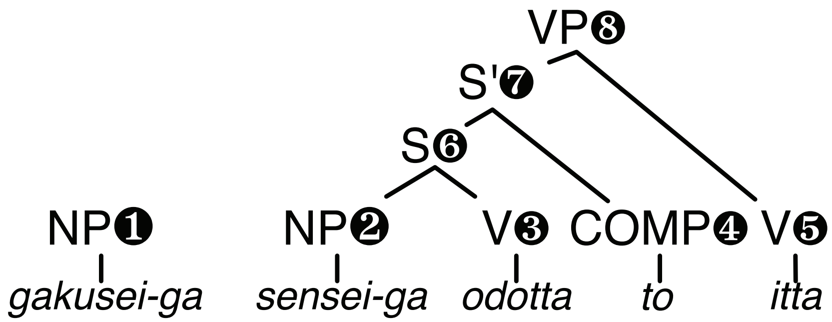


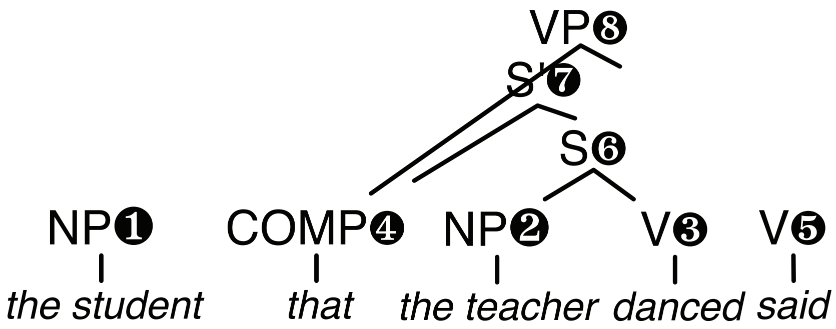


Syntax in MT

Example SCFG

	Japanese	English
S →	NP① VP②	NP① VP②
S'→	S① COMP②	COMP② S①
VP →	NP① V②	V② NP①
NP →	*gakusei-ga*	*student*
NP →	*sensei-ga*	*teacher*
V →	*odotta*	*danced*
V →	*itta*	*said*
COMP →	*to*	*that*

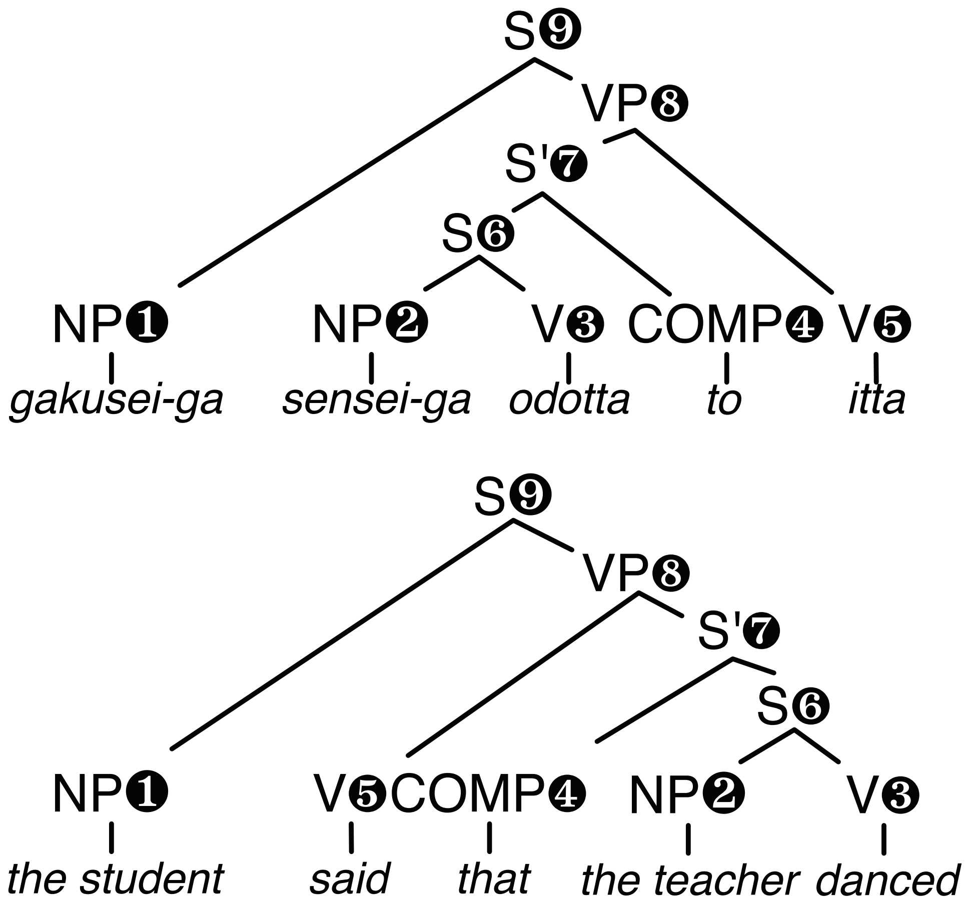




Syntax in MT

Example SCFG

	Japanese	English
S →	NP① VP②	NP① VP②
S'→	S① COMP②	COMP② S①
VP →	NP① V②	V② NP①
NP →	*gakusei-ga*	*student*
NP →	*sensei-ga*	*teacher*
V →	*odotta*	*danced*
V →	*itta*	*said*
COMP →	*to*	*that*



Syntax in MT

Some relevant concepts:

- rooted ordered trees
- internal nodels labeled with syntactic categories
- leaf nodes labeled with words
- linear and hierchical relations between tree nodes
- internal nodes can represent direct order or reverse order at different levels

Universitat Politècnica de València

Máster en Inteligencia Artificial, Reconocimiento de Formas e Imagen Digital

MACHINE TRANSLATION

Stochastic inversion transduction grammars

Joan Andreu Sánchez

jandreu@prhlt.upv.es

Index

- **SITG Definitions**
- Problems with SITG for MT
- Inside algorithm with SITG
- Stochastic learning of SITG
- Practical use of SITG
- Experiments with SITG
- Bibliography
- Exercises

Stochastic inversion transduction grammars [Wu 97]

A context-free based approach to bilingual segmentation

For a non-terminal symbol A, B and C and for any source word s and any target word t,

$$\begin{aligned} A &\rightarrow BC/BC \\ A &\rightarrow BC/CB \\ A &\rightarrow x/y \\ A &\rightarrow x/\epsilon \\ A &\rightarrow \epsilon/y \end{aligned} \qquad\qquad \begin{aligned} A &\rightarrow [BC] \\ A &\rightarrow \langle BC \rangle \\ A &\rightarrow x/y \\ A &\rightarrow x/\epsilon \\ A &\rightarrow \epsilon/y \end{aligned}$$

- Syntactical rules: to model long-term relations
- Lexical rules: to model word-level translations

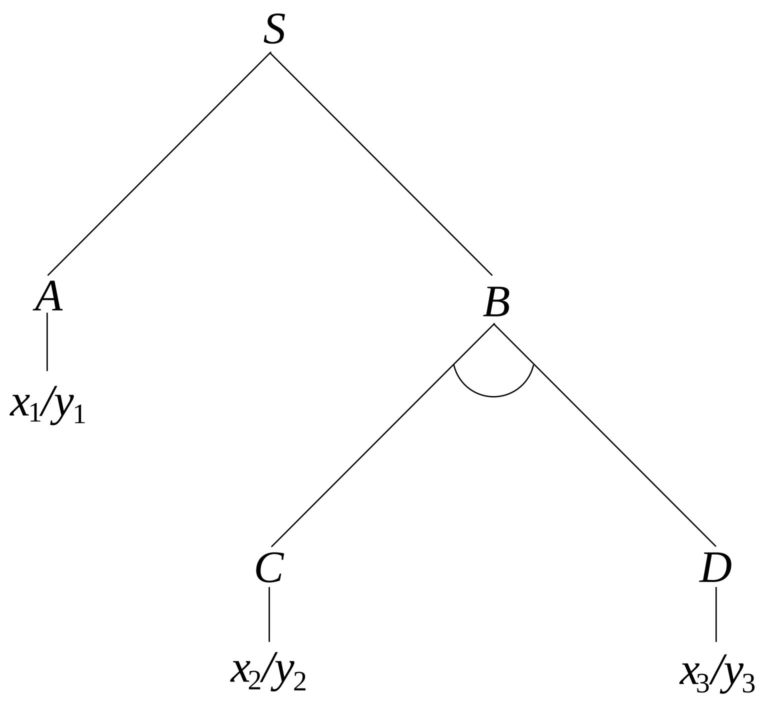
An example

Generative process:

1. Write down the source and the target start symbols
2. Choose a synchronous rule whose left-hand side is the left-most written down source non-terminal symbol
3. Simultaneously rewrite the source symbols and its corresponding target symbol with source and the target side of the rule, respectively
4. Repeat step 2 while there are written down source and target non-terminal symbols

An example

$$\begin{aligned} S &\rightarrow [AB] \\ A &\rightarrow x_1 \;/\; y_1 \\ B &\rightarrow \langle CD \rangle \\ C &\rightarrow x_2 \;/\; y_2 \\ D &\rightarrow x_3 \;/\; y_3 \end{aligned}$$



$$(S, S) \Rightarrow (AB, AB) \Rightarrow (x_1 B, y_1 B) \Rightarrow (x_1 CD, y_1 DC) \Rightarrow (x_1 x_2 D, y_1 D y_2) \Rightarrow (x_1 x_2 x_3, y_1 y_3 y_2)$$

For getting the source and target strings:

- Direct Rule $\Rightarrow$ Preorder
- Inverse Rule $\Rightarrow$ Postorder

Formal definition

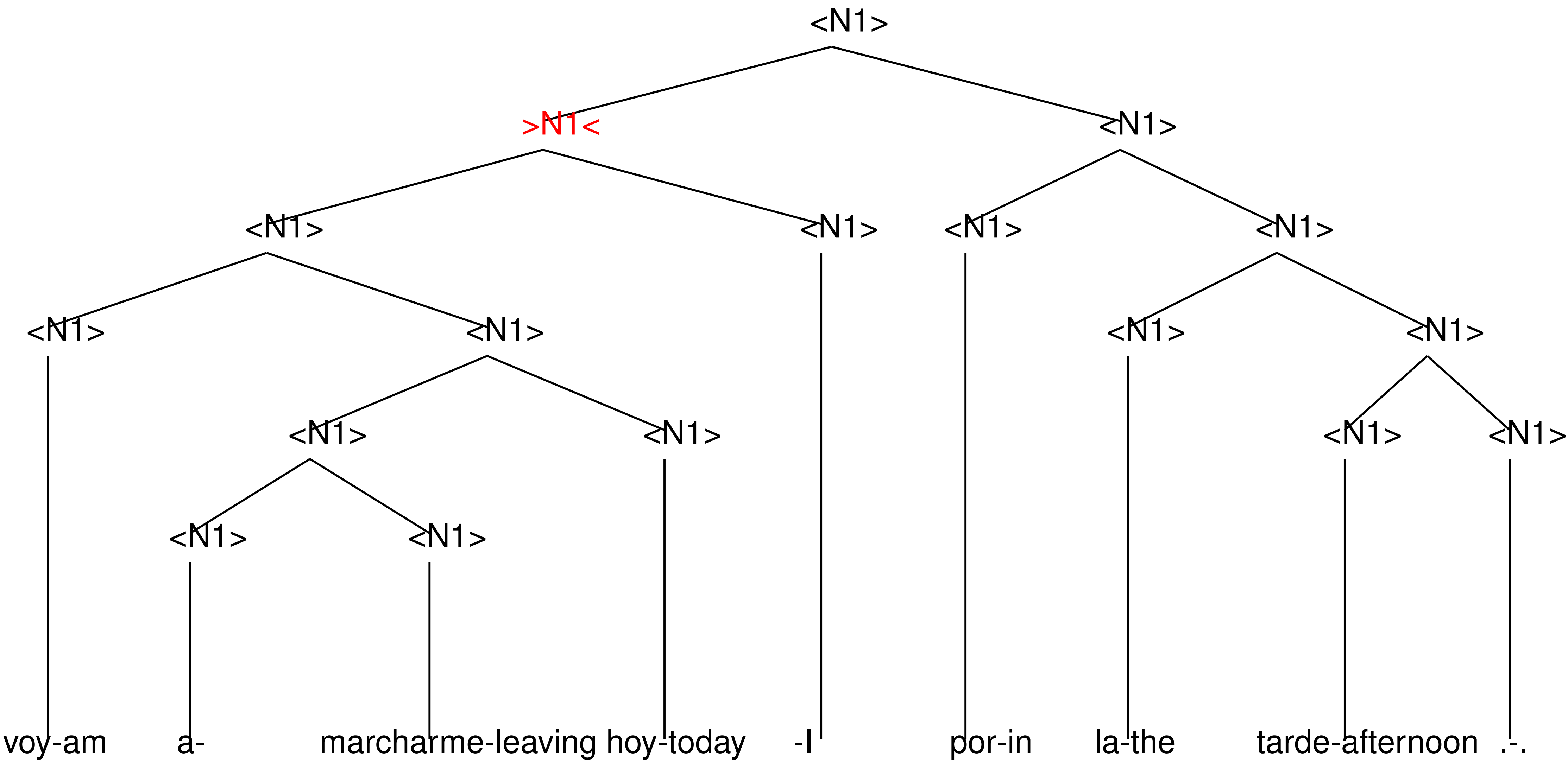
An ITG is denoted by $G = (N, W_1, W_2, R, S)$ where:

- N is a finite set of nonterminals
- W_1 is a finite set of words of language 1
- W_2 is a finite set of words of language 2
- $S \in N$ is the start symbol
- R is a finite set of straight orientation rules $A \to [a_1 a_2 \ldots a_r]$ and inverted orientation rules $A \to \langle a_1 a_2 \ldots a_r \rangle$, $a_i \in N \cup X$ and $X = (W_1 \cup \{\epsilon\}) \times (W_2 \cup \{\epsilon\})$

Theorem. For any ITG G, there exists an equivalent ITG G' in which every production takes one of the following forms:

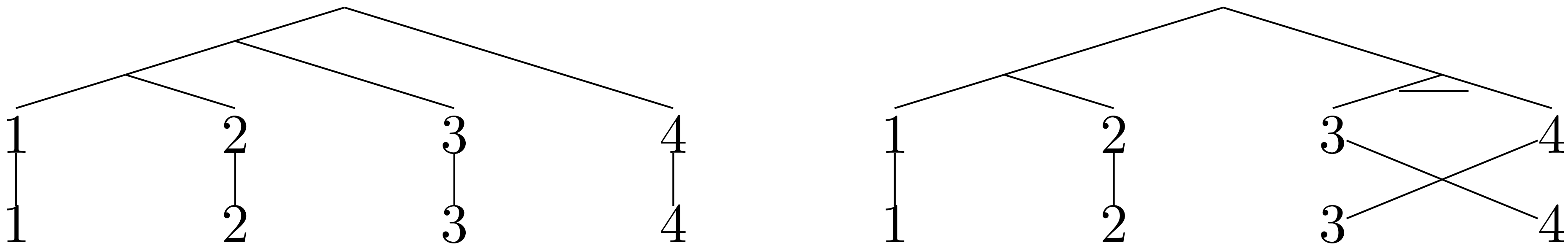
$$\begin{array}{lll} S \to \epsilon/\epsilon & A \to x/\epsilon & A \to [BC] \\ S \to x/y & A \to \epsilon/y & A \to \langle BC \rangle \end{array}$$

An example


<N1>
>N1<
<N1>
<N1>
<N1>
<N1>
<N1>
<N1>
<N1>
<N1>
<N1>
<N1>
<N1>
<N1>
<N1>
<N1>
<N1>
voy-am
a-
marcharme-leaving
hoy-today
-I
por-in
la-the
tarde-afternoon
.-.

Expressiveness of ITGs

YES

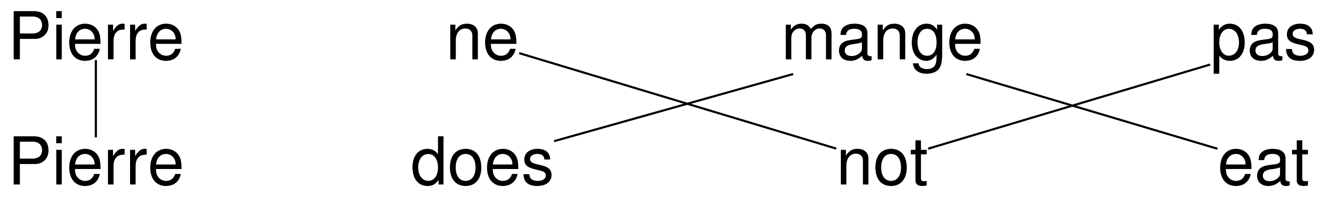


NO (inside-out alignment)

r	ITG	all matchings	ratio
1	1	1	1.000
2	2	2	1.000
3	6	6	1.000
4	22	24	0.917
5	90	120	0.750
6	394	720	0.547
7	1,806	5,040	0.358
8	8,558	40,320	0.212
9	41,586	362,880	0.115
10	206,098	3,628,800	0.057

Expressiveness of ITGs

- *Bonbon* alignment example from [Simard 05, Simard 11]:



  0.25-1% of translation units are part of bonbons in practice. Bonbon alignment can not be represented by ITG

- Other translation units exist that are not representable by ITG: 38% according to [Søgaard 11]

- Local reordering models are not able to represent 61% of reorderings

Stochastic inversion transduction grammars

A SITG is denoted by $G_s = (G, p)$ where:

- G is an ITG
- p is a function that attaches a probability to each rule:

$p : R \rightarrow]0, 1]$
$$\sum_{1 \leq j \leq n_i} p(A_i \rightarrow \alpha_j) = 1, \qquad \forall A_i \in N$$

Stochastic derivation

$$(S, S) = (\alpha_0, \beta_0) \stackrel{r_1}{\Rightarrow} (\alpha_1, \beta_1) \stackrel{r_2}{\Rightarrow} (\alpha_2, \beta_2) \cdots (\alpha_{m-1}, \beta_{m-1}) \stackrel{r_m}{\Rightarrow} (\alpha_m, \beta_m) = (x, y)$$

Probability of (x, y) being generated by $G_s = (G, p)$ from the rule sequence $d_x = (r_1, \ldots, r_m)$:

$$P_{G_s}((x, y), d_x) = \prod_{j=1 \cdots m} p(r_j)$$

Probability of a string pair

$$P_{G_s}(x,y) = \sum_{d_x \in D_x} P_{G_s}((x,y), d_x)$$

Probability of the best derivation

$$\widehat{P}_{G_s}(x,y) = \max_{d_x \in D_x} P_{G_s}((x,y), d_x)$$

Language generated by a SITG

$$L(G_s) = \{(x,y) \mid P_{G_s}(x,y) > 0\}$$

Index

- SITG Definitions
- **Problems with SITG for MT**
- Inside algorithm with SITG
- Stochastic learning of SITG
- Practical use of SITG
- Experiments with SITG
- Bibliography
- Exercises

Stochastic inversion transduction grammars

- Parsing:
  - Inside algorithm
  - Viterbi algorithm
- Learning:
  - Structure learning: rule learning
  - Probabilistic estimation: Inside-outside estimation
Viterbi-based estimation
- Translation:
  - Adapted Cooker-Younger-Kasami parser algorithm

Index

Stochastic parsing with a SITG

Let

$$\delta_{i,j,k,l}(A) = \widehat{P}(A \stackrel{*}{\Rightarrow} x_{i+1} \cdots x_j / y_{k+1} \cdots y_l)$$

Then

$$\delta_{0,|x|,0,|y|}(S) = \widehat{P}(x, y)$$

1. Initialization

$$\begin{aligned}
\delta_{i-1,i,k-1,k}(A) &= p(A \to x_i / y_k) & \qquad 1 \leq i \leq |x|, 1 \leq k \leq |y| \\
\delta_{i-1,i,k,k}(A) &= p(A \to x_i / \epsilon) & \qquad 1 \leq i \leq |x|, 0 \leq k \leq |y| \\
\delta_{i,i,k-1,k}(A) &= p(A \to \epsilon / y_k) & \qquad 0 \leq i \leq |x|, 1 \leq k \leq |y|
\end{aligned}$$

$2.$ Recursion. For all $A \in N$, and i, j, k, l such that $0 \leq i < j \leq |x|$, $0 \leq k < l \leq |y|$ and $j - i + l - k > 2$:

$$\delta_{ijkl}(A) = \max(\delta^{[]}_{ijkl}(A), \delta^{\langle\rangle}_{ijkl}(A))$$

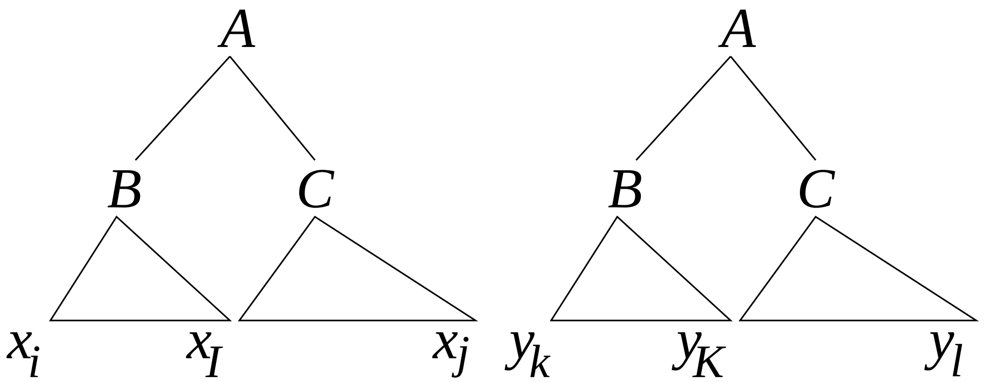
$$\delta^{[]}_{ijkl}(A) = \max_{\substack{B,C \in N \\ i \leq I \leq j, k \leq K \leq l \\ (I-i)(j-I)+(K-k)(l-K) \neq 0}} p(A \to [BC])\delta_{iIkK}(B)\delta_{IjKl}(C)$$

$$\delta^{\langle\rangle}_{ijkl}(A) = \max_{\substack{B,C \in N \\ i \leq I \leq j, k \leq K \leq l \\ (I-i)(j-I)+(K-k)(l-K) \neq 0}} p(A \to \langle BC\rangle)\delta_{iIKl}(B)\delta_{IjkK}(C)$$

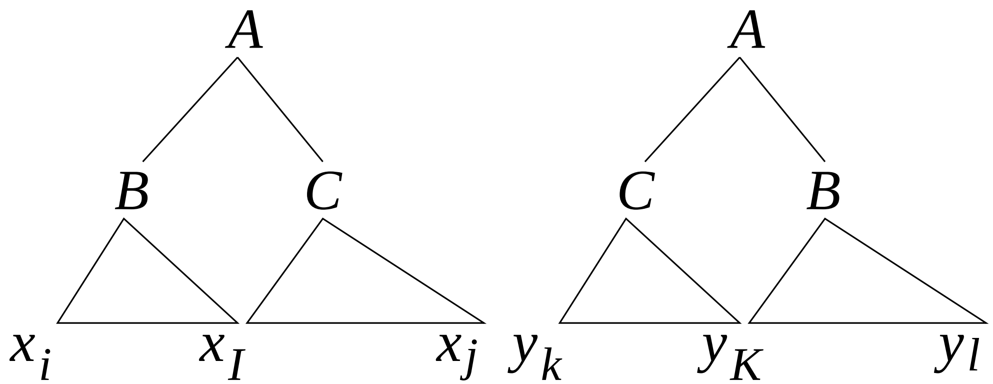
2. Recursion. For all $A \in N$, and i, j, k, l such that $0 \leq i < j \leq |x|$, $0 \leq k < l \leq |y|$ and $j - i + l - k > 2$:

$$\delta_{ijkl}(A) = \max(\delta^{[]}_{ijkl}(A), \delta^{\langle\rangle}_{ijkl}(A))$$

$$\delta^{[]}_{ijkl}(A) = \max_{\substack{B,C\in N \\ i\leq I\leq j, k\leq K\leq l \\ (I-i)(j-I)+(K-k)(l-K)\neq 0}} p(A \to [BC])\delta_{iIkK}(B)\delta_{IjKl}(C)$$



$$\delta^{\langle\rangle}_{ijkl}(A) = \max_{\substack{B,C\in N \\ i\leq I\leq j, k\leq K\leq l \\ (I-i)(j-I)+(K-k)(l-K)\neq 0}} p(A \to \langle BC\rangle)\delta_{iIKl}(B)\delta_{IjkK}(C)$$



Additional considerations

- Time complexity: $O(|x|^3|y|^3|R|)$
- Space complexity: $O(|x|^2|y|^2|N|)$
- Inside probability: replace maximization ($\max$) by addition ($\sum$)

Index

Stochastic learning of SITG

SITG are closely related to SCFG

- Open problem: initial rules of the SITG
- Merit functions to be used for stochastic learning of SITG
  - Likelihood function
    - ∗ Estimation similar to inside-outside algorithm
    - ∗ Viterbi-based estimation: useful when the amount of training data is large
  - Discriminative function: similar ideas to [Gopalakrishnan 91].
    To be researched!
    - ∗ Numerator based on exisiting parser
    - ∗ Denominator based on a relaxed SITG

    - ∗ **Problems:** difficult to formalize and to implement

Index

Use of SITG

- Translation-driven segmentation: useful for languages with hight ambiguity degree in segmentation tasks (Chinese).
- Bracketing: useful for constraining subsequent training processes

$$
\begin{aligned}
a : A &\rightarrow [AA] \\
a : A &\rightarrow \langle AA \rangle \\
b_{xy} : A &\rightarrow x/y \\
b_{x\epsilon} : A &\rightarrow x/\epsilon \\
b_{\epsilon y} : A &\rightarrow \epsilon/y
\end{aligned}
$$

- Alignment:
  - word segments
  - sentence splitting
- Bilingual constraint transfer

Use of SITG

- Translation-driven segmentation: useful for languages with hight ambiguity degree in segmentation tasks (Chinese).

- Bracketing: useful for constraining subsequent training processes

$$
\begin{aligned}
a : A &\rightarrow [AA] \\
a : A &\rightarrow \langle AA \rangle \\
b_{xy} : A &\rightarrow x/y \\
b_{x\epsilon} : A &\rightarrow x/\epsilon \\
b_{\epsilon y} : A &\rightarrow \epsilon/y
\end{aligned}
$$

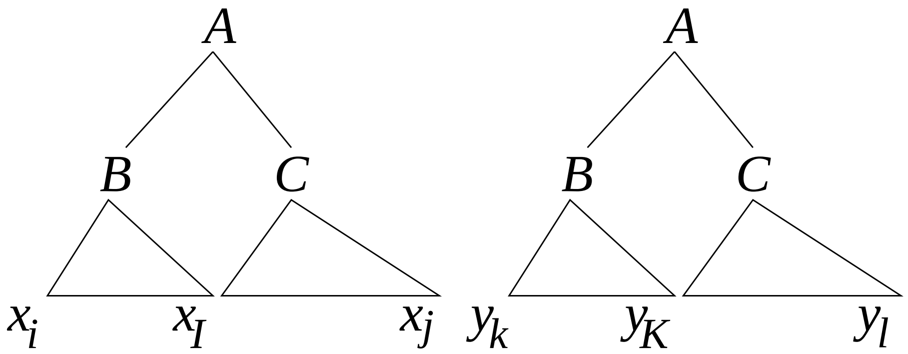
- Alignment:
  - **word segments**
  - sentence splitting

- Bilingual constraint transfer

Index

Use of SITG (I) [Sánchez 06]

Problem: Obtaining bilingual translations phrases for phrase-based translation
Solution: Learning bilingual word phrases by using Stochastic Inversion Transduction Grammars



$$\Downarrow$$

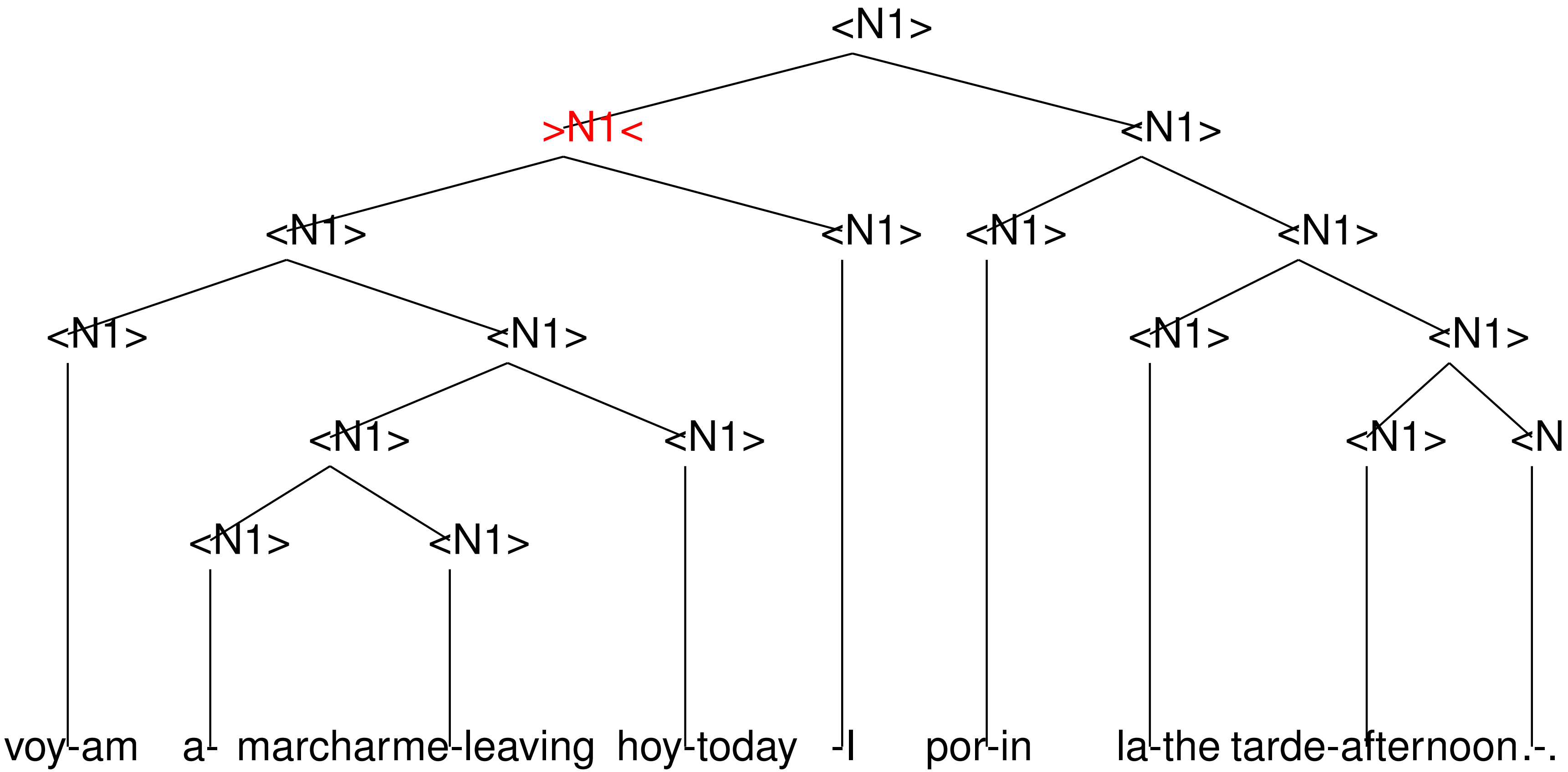
$$\begin{array}{cc} x_{i+1}\cdots x_I & y_{k+1}\cdots y_K \\ x_{I+1}\cdots x_j & y_{K+1}\cdots y_l \end{array}$$

$$\Downarrow$$

$$x_{i+1}\cdots x_I \;|||\; y_{k+1}\cdots y_K \;|||\; p$$
$$x_{I+1}\cdots x_j \;|||\; y_{K+1}\cdots y_l \;|||\; p$$

Each span defines a bilingual phrase

Example: bilingual phrases from bilingual trees



voy ||| am ||| . . .
marcharme ||| leaving ||| . . .
a marcharme ||| leaving ||| . . .
hoy ||| today ||| . . .
a marcharme hoy ||| leaving today ||| . . .

por ||| in ||| . . .
la ||| the |||. . .
tarde ||| afternoon ||| . . .
la tarde ||| the afternoon ||| . . .
por la tarde ||| in the afternoon ||| . . .

Problems

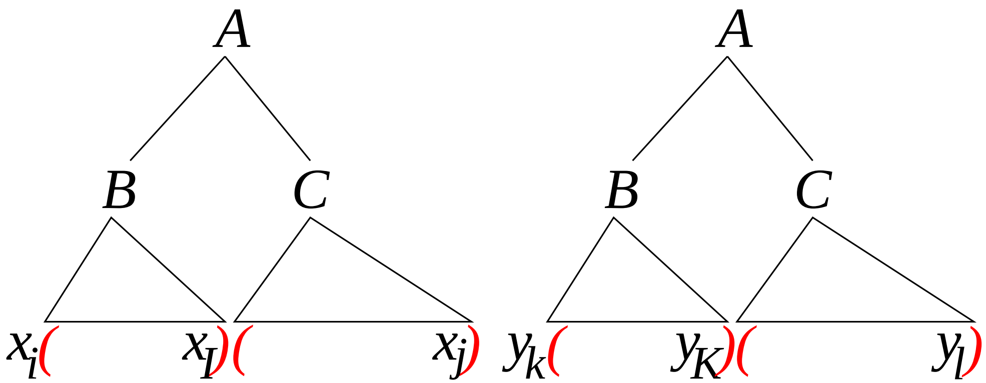
- Parsing time $O(n^6)$
- Obtaining initial SITG
- Probabilities associated to phrases

Parsing bracketed text with SITG

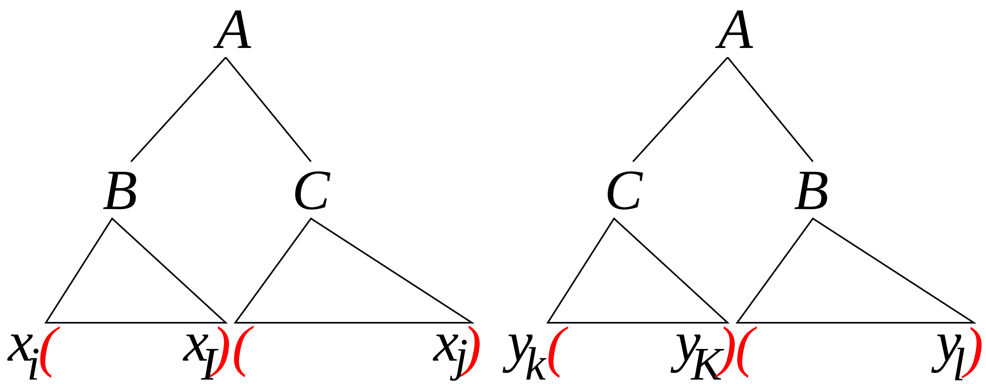
2. Recursion. For all $A \in N$, and i, j, k, l such that $0 \leq i < j \leq |x|$, $0 \leq k < l \leq |y|$ and $j - i + l - k > 2$:

$$\delta_{ijkl}(A) = \max(\delta^{[]}_{ijkl}(A), \delta^{\langle\rangle}_{ijkl}(A)) \; c(i+1, j, k+1, l)$$

$$\delta^{[]}_{ijkl}(A) = \max_{\substack{B,C \in N \\ i \leq I \leq j, k \leq K \leq l \\ (I-i)(j-I)+(K-k)(l-K) \neq 0}} p(A \rightarrow [BC])\delta_{iIkK}(B)\delta_{IjKl}(C)$$



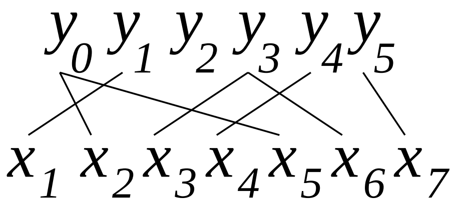
$$\delta^{\langle\rangle}_{ijkl}(A) = \max_{\substack{B,C \in N \\ i \leq I \leq j, k \leq K \leq l \\ (I-i)(j-I)+(K-k)(l-K) \neq 0}} p(A \rightarrow \langle BC\rangle)\delta_{iIKl}(B)\delta_{IjkK}(C)$$



Time complexity: linear if full bracketing !!

Experiments: Obtaining a SITG from an aligned corpus

1. Aligning words



2. Obtaining lexical rules

$$p : A \to x/y$$

$$p : A \to x/\epsilon$$

$$p : A \to \epsilon/y$$

3. Syntactic rules

$$p : A \to \langle AA \rangle$$

$$p : A \to [AA]$$

4. Additional rules

$$p : A \to x/\epsilon$$

$$p : A \to \epsilon/y$$

Experiments

Experiment setup:

- Europarl corpus
- 5-gram LM
- Moses system (2 models):
  $p(e|f)$, $p(f|e)$
- MERT training
- Baseline: 31.0 (5 mod.)
  29.6 (2 mod.)

Experiments

Experiment setup:

- Europarl corpus
- 5-gram LM
- Moses system (2 models): $p(e|f)$, $p(f|e)$
- MERT training
- Baseline: 31.0 (5 mod.)
  29.6 (2 mod.)

Results are shown in BLEU/WER. 0 iterations means the SITG was obtained by the heuristic technique.

$\|N\|$	It. 0	It. 1
1	26.8/62.5	26.9/62.6
2	27.0/62.6	27.5/62.1
3	26.9/62.7	27.0/62.7
4	26.6/63.2	27.9/61.5

Syntax-based probabilities

A syntax-based probability

- Let Ω be the multiset of spans
- Let $\Omega_{\mathbf{s},\mathbf{t}} \subseteq \Omega$ the multiset of $(\mathbf{s}, \mathbf{t})$ spans.

Expected value of $\widehat{p}(\mathbf{s}, \mathbf{t})$ according to the empirical distribution

$$E_\Omega(\widehat{p}(\mathbf{s}, \mathbf{t})) = \frac{\sum_{(\mathbf{a},\mathbf{b}) \in \Omega_{\mathbf{s},\mathbf{t}}} \widehat{p}(\mathbf{a}, \mathbf{b})}{|\Omega|}.$$

Marginalise for the input side and for the output side

$$E_\Omega(\widehat{p}(\mathbf{s})) = \sum_{\mathbf{t}} E_\Omega(\widehat{p}(\mathbf{s}, \mathbf{t})) \qquad E_\Omega(\widehat{p}(\mathbf{t})) = \sum_{s} E_\Omega(\widehat{p}(\mathbf{s}, \mathbf{t})).$$

Syntax-based probabilities

Syntax-based models

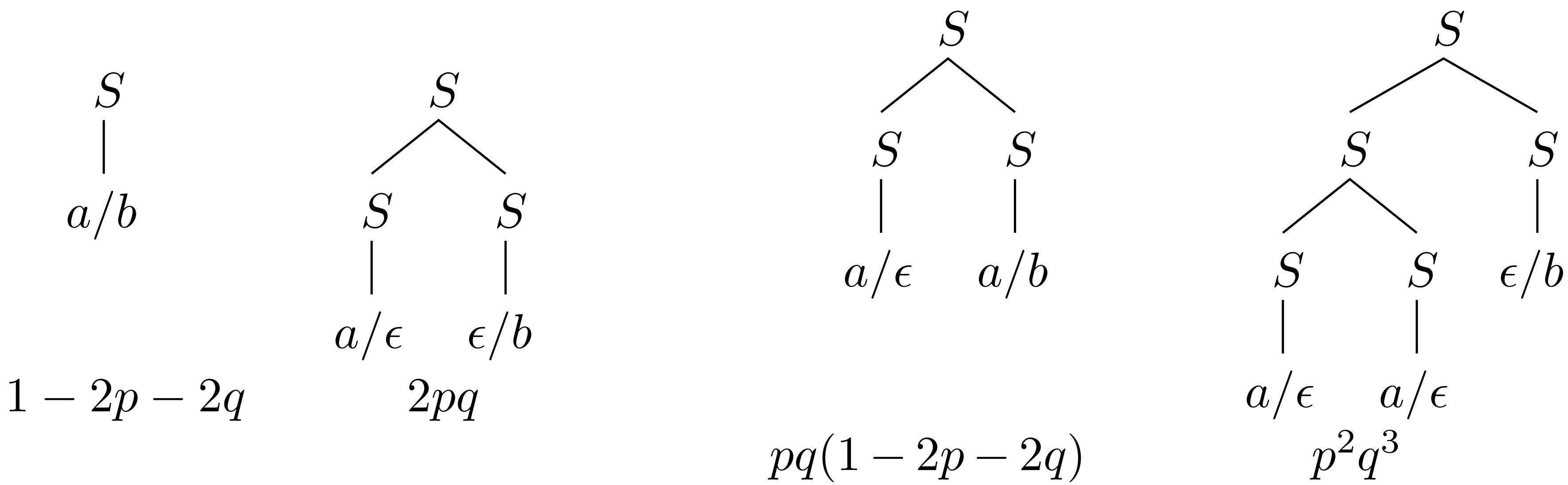
$$p(\mathbf{s}|\mathbf{t}) = \frac{E_\Omega(\widehat{p}(\mathbf{s},\mathbf{t}))}{E_\Omega(\widehat{p}(\mathbf{t}))} \qquad p(\mathbf{t}|\mathbf{s}) = \frac{E_\Omega(\widehat{p}(\mathbf{s},\mathbf{t}))}{E_\Omega(\widehat{p}(\mathbf{s}))}.$$

Results are shown in BLEU/WER.

|N|	It. 1	+ syntactic
1	26.9/62.6	27.7/61.6
2	27.5/62.1	28.3/61.1
3	27.0/62.7	28.2/61.3
4	27.9/61.5	28.9/60.0

Correction on stochastic parsing with a SITG [Gascó 10a]

$$
\begin{array}{rlll}
p & S & \to [SS] & \qquad p \quad S \quad \to \langle SS \rangle \\
q & S & \to \epsilon/b & \qquad q \quad S \quad \to a/\epsilon \\
1-2p-2q & S & \to a/b &
\end{array}
$$



Correction on stochastic parsing with a SITG

1. Initialization

$$\delta_{i-1,i,k-1,k}(A) = p(A \rightarrow x_i/y_k) \qquad 1 \leq i \leq |x|, 1 \leq k \leq |y|$$

$$\delta_{i-1,i,k,k}(A) = p(A \rightarrow x_i/\epsilon) \qquad 1 \leq i \leq |x|, 0 \leq k \leq |y|$$

$$\delta_{i,i,k-1,k}(A) = p(A \rightarrow \epsilon/y_k) \qquad 0 \leq i \leq |x|, 1 \leq k \leq |y|$$

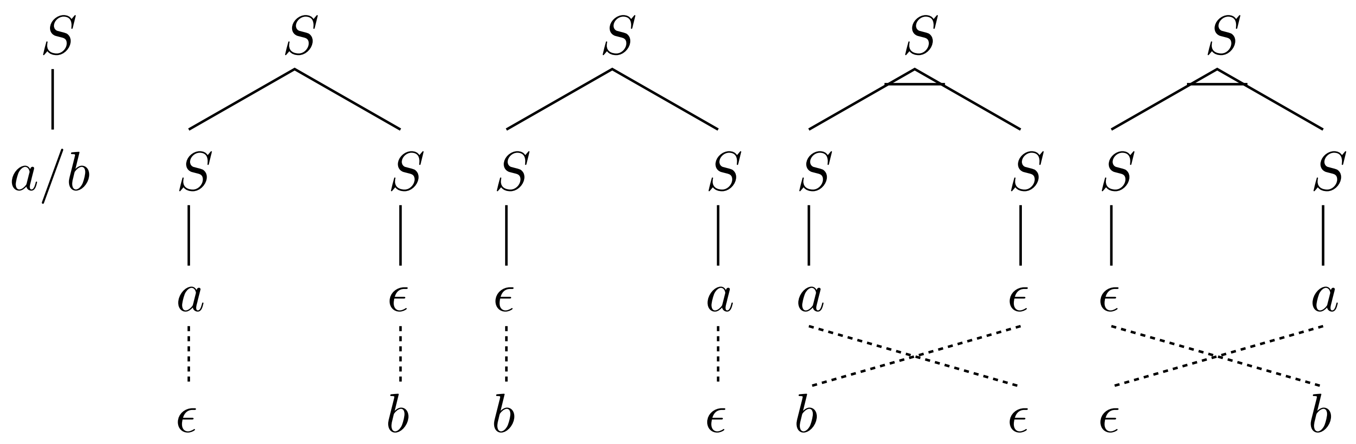
2. Recursion. For all $A \in N$, and i, j, k, l such that $0 \leq i < j \leq |x|$, $0 \leq k < l \leq |y|$ and $j - i + l - k \geq 2$:

$$\delta_{ijkl}(A) = \max(\delta^{[]}_{ijkl}(A), \delta^{\langle\rangle}_{ijkl}(A))$$

$$\delta^{[]}_{ijkl}(A) = \max_{\substack{B,C \in N \\ i \leq I \leq j, k \leq K \leq l \\ ((j-I)+(l-K)) \times ((I-i)+(K-k)) \neq 0}} p(A \rightarrow [BC])\delta_{iIkK}(B)\delta_{IjKl}(C)$$

$$\delta^{\langle\rangle}_{ijkl}(A) = \max_{\substack{B,C \in N \\ i \leq I \leq j, k \leq K \leq l \\ ((j-I)+(K-k)) \times ((I-i)+(I-K)) \neq 0}} p(A \rightarrow \langle BC\rangle)\delta_{iIKl}(B)\delta_{IjkK}(C)$$

Correction on stochastic parsing with a SITG



n	Wu's alg.	Modified alg.	ratio
1	1	5	0.200
2	34	290	0.117
3	1,928	34,088	0.057
4	131,880	5,152,040	0.026
5	10,071,264	890,510,432	0.011
6	827,969,856	167,399,588,160	0.005

* See exercice

Stochastic parsing with a SITG: Inside probability

Inside probability of $(x_{i+1}\ldots x_{i+j}, y_{k+1}\ldots y_{k+l})$ from non-terminal A:

$$\mathcal{E}_{i,i+j,k,k+l}[A] = p(A \stackrel{*}{\Rightarrow} x_{i+1}\cdots x_{i+j}/y_{k+1}\cdots y_{k+l})$$

Theorem. If the *inside* algorithm is applied to the substring pair $(x_{i+1}\ldots x_{i+j}, y_{k+1}\ldots y_{k+j})$ with a SITG $\mathcal{G}$, then the probabilistic parse matrix $\mathcal{E}$ collects correctly the probability of this substring pair.

Corollary. The probability of the pair string $(x_1\ldots x_{|x|}, y_1\ldots y_{|y|})$ can be computed by means of the probabilistic parse matrix $\mathcal{E}$ as:

$$p(S \stackrel{+}{\Rightarrow} x_1\ldots x_{|x|}/y_1\ldots y_{|y|}) = \mathcal{E}_{0,|x|,0,|y|}[S]$$

Experiments

⇒ To test the differences of both Viterbi parsing algorithms with two languages with a **very distinct** syntax structure.

Statistics for IWSLT 2009 Chinese-English BTEC corpus.

Corpus Set	Statistic	Chinese	English
	Sentences	42,655	
Training	Words	330,163	380,431
	Vocabulary Size	8,773	8,387
	Sentences	511	
Test	Words	3,352	3,821
	Vocabulary Size	888	813

Experiment	% of sentences with a different parse tree	% of sentences not parsed with the original algorithm
Ch - En	36.3%	0.2%
[Ch] - En	37.2%	1.4%
Ch - [En]	37.0%	1.0%
[Ch] - [En]	40.9%	3.9%

Experiments

⇒ To test the differences of both Viterbi parsing algorithms with two languages with a **similar** syntax structure.

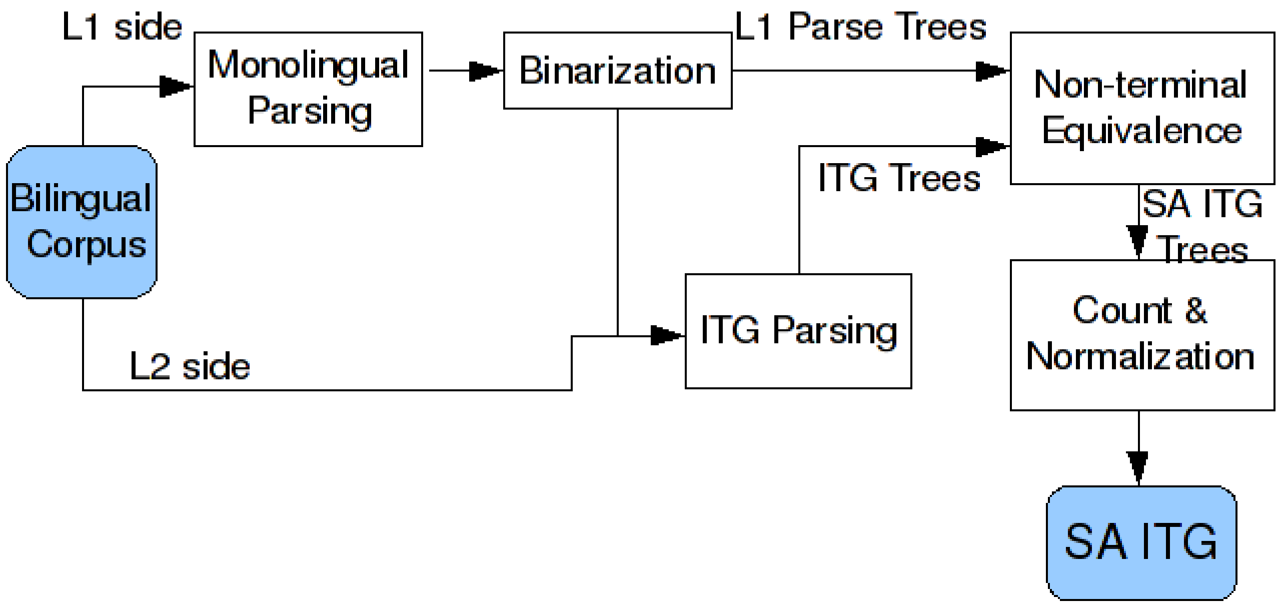
Statistics for Hansard French-English corpus (less than 40 words).

Corpus Set	Statistic	French	English
Training	Sentences	997,823	
	Words	16,547,387	14,266,620
	Vocabulary Size	68,431	49,892
Test	Sentences	511	
	Words	3,352	3,821
	Vocabulary Size	888	813

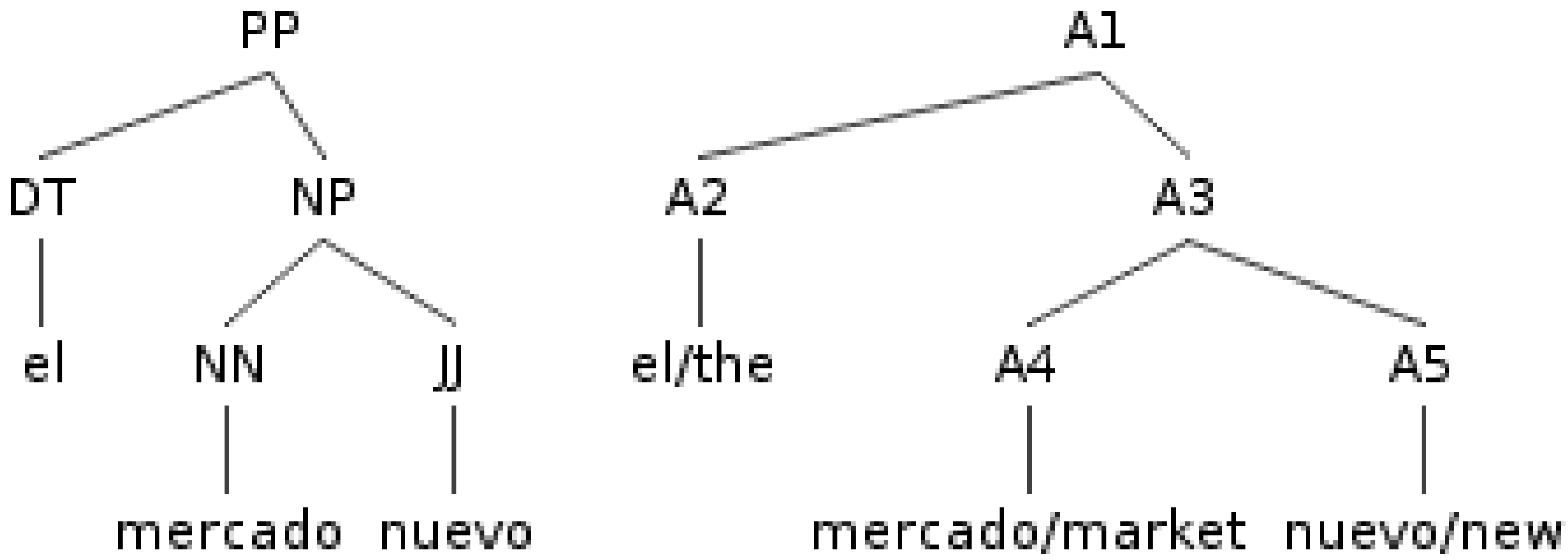
Experiment	% of sentences with a different parse tree
Fr - En	27.7%
[Fr] - En	28.0%
Fr - [En]	28.5%
[Fr] - [En]	30.6%

Syntax-Augmented SITGs [Gascó 10b]

1. To create an initial SITG
2. To estimate the probabilities
3. To attach linguistic information to the non-terminal symbols



Syntax-Augmented SITGs: Example



Syntax-Augmented SITGs: Experiments

- IWSLT 2008 (Chinese-English BTEC)
- Standard tools: GIZA++, ZMERT
- Stanford parser for Chinese
- Baseline: Moses, 5-gram

Corpus Set	Statistic	Chinese	English
Training	Sentences	42,655	
	Words	330,163	380,431
	Voc. Size	8,773	8,387
DevSet	Sentences	489	
	Words	3,169	3,861
	OOV Words	111	115
Test	Sentences	507	
	Words	3,357	-
	OOV Words	97	-

System	%BLEU
Baseline PBT	41.1
Initial ITG	41.2
Re-estimated ITG	41.8
Source SAITG	42.9
Target SAITG	43.0

Index

References

- P.S Gopalakrishnan, D. Kanevsky, A. Nadas, and D. Nahamoo: *An inequality for rational functions with applications to some statistical estimation problems*. IEEE Transactions on Information Theory, 37:107-113, 1991.
- [Sánchez 97] J.A. Sánchez, J.M. Benedí: *Consistency of stochastic context-free grammars from probabilistic estimation based on growth transformations* IEEE Transactions on Pattern Analysis and Machine Intelligence, 19 (9), 1052–1055.
- [Wu 97] D. Wu: *Stochastic Inversion Transduction Grammars and Bilingual Parsing of Parallel Corpora*. Comp. Ling. 1997.
- [Sánchez 06] J.A. Sánchez and J.M. Benedí: *Obtaining Word Phrases with Stochastic Inversion Transduction Grammars for Phrase-based Statistical Machine Translation*. EAMT 2006.
- [Sanchís 08] G. Sanchís and J.A. Sánchez. *Using parsed corpora for estimating stochastic inversion transduction grammars*. LREC, 2008.
- [Søgaard 09] A. Søgaard, D. Wu: *Empirical lower bounds on translation unit error rate for the full class of Inversion Transduction Grammars*. ICPT 2009.
- [Gascó 10a] G. Gascó, J.A. Sánchez, J.M. Benedí: *Enlarged Search Space for SITG Parsing.* NAACL-HLT 2010.
- [Gascó 10b] G. Gascó, J.A. Sánchez: *Syntax Augmented Inversion Transduction Grammars for Machine Translation.* CICLING 2010.
- [Søgaard 10] A. Søgaard: *Can Inversion Transduction Grammars Generate Hand Alignments.* EAMT 2010.
- [Søgaard 11] A. Søgaard: *A* $O(|G|n^6)$ *time extension of inversion transduction grammars.* Machine Translation, 25, 291–315, 2011.

Index

- SITG Definitions
- Use of SITG for MT
- Inside algorithm with SITG
- Practical use of SITG
- Experiments with SITG
- Bibliography
- **Exercises**

Exercises (I)

1. **(*)** Write a SITG and a pair of source and target strings that can not be parsed with the written SITG. Show why the SITG is not able to parse the pair of source and target strings.
2. **(*)** Write a SITG and a pair of source and target strings and apply the Wu parsing algorithm described in these slides.
3. **(*)** Write a SITG and and a pair of source and target strings and apply the Viterbi version $(\max)$ of Wu parsing algorithm mentioned in these slides.
4. **(**)** Write the *outside* algorithm for parsing with SITGs. Write a SITG and a pair of source and target strings and shows the analysis table.
5. **(**)** Look for references with the last results and advances obtained with SITGs and prepare a summary report.
6. **(***)** Write the *outside* algorithm for parsing with SITGs. Write a SITG and a pair of source and target strings and computes the analysis table. Compute the analysis table with the Wu's algoritms with the same source and target strings and the same SITG.
7. **(***)** Implement the *outside* algorithm for parsing with SITGs.
8. **(***)** Write a SITG and a two pairs of different source and target strings and use the idea described in [Sánchez 06] to obtain the set of possible word phrases.
9. **(***)** Perform experiments similar to the experiments described in [Søgaard 10]. The software will be provided by the professor.

Exercises (II)

10. **(***)** Perform experiments similar to the experiments described in [Søgaard 10]. The software will be provided by the professor.
11. **(****)** Implement an A* algorithm for parsing with SITG.
12. **(****)** Study the relation between results in references [Søgaard 09] and [Gascó 10a]. Validate your discussion with an experimental evaluation.
13. **(*****)** Implement a system for obtaining word phrases in which a weight based on confidence measures is used in phrase-based MT system.
14. **(*****)** Implement an Early vesion of the parsing algorithm for SITGs.
15. **(*****)** Read paper [Sánchez 97] and obtain similar results.

Universitat Politècnica de València

Máster en Inteligencia Artificial, Reconocimiento de Formas e Imagen Digital

MACHINE TRANSLATION

Tree to string translation

Joan Andreu Sánchez

jandreu@prhlt.upv.es

Index

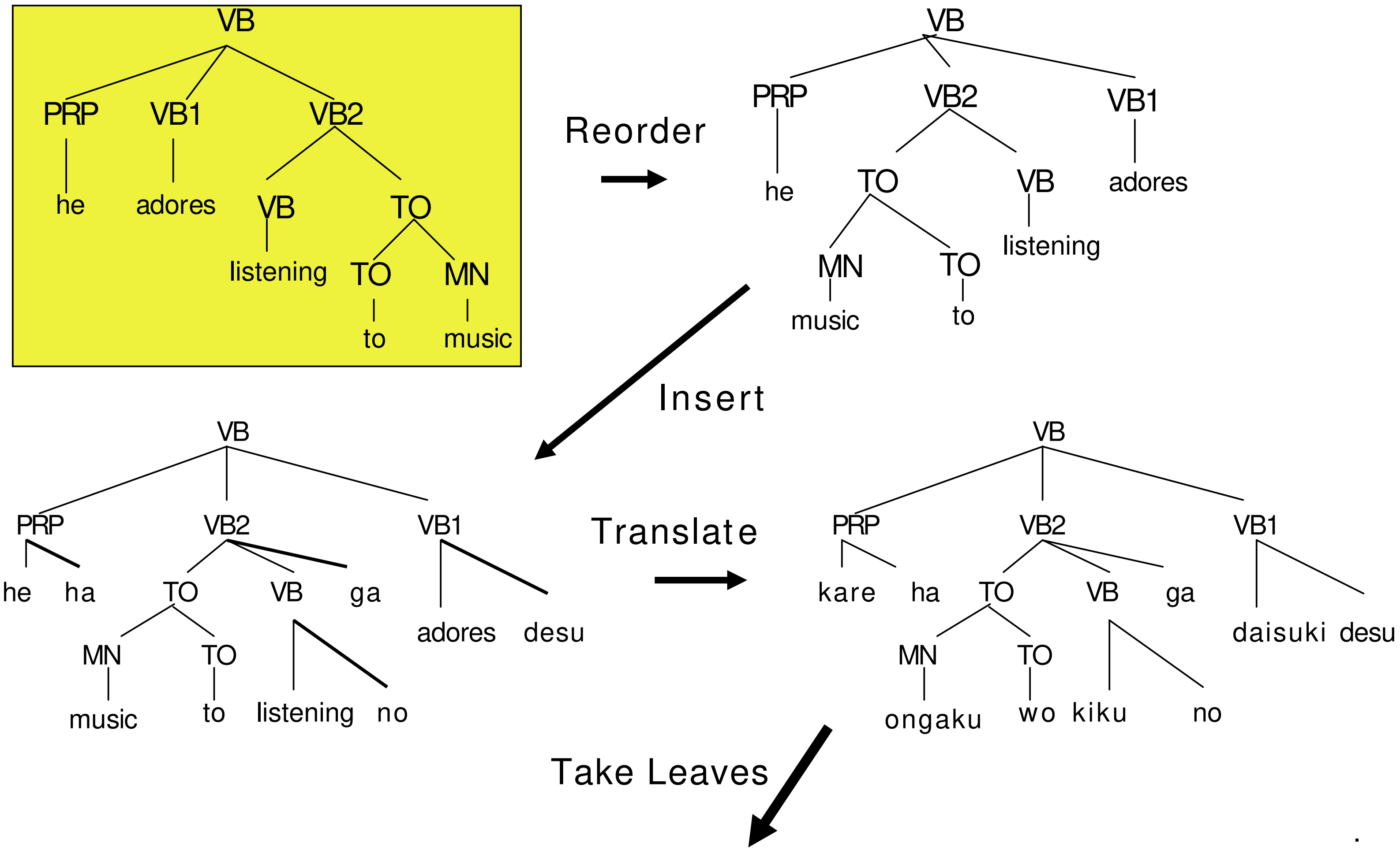
A Tree to String Transducer [Yamada 01a]

Main components:

- The input (source) sentence is pre-processed by a syntactic parser
- The input to the MT system is a parse tree
- A statistical channel performs operations on each node of the parse tree:
  - reordering child nodes
  - inserting extra words at each node
  - translating leaf words
- The translation process is performed as a parsing process
- The output of the model is a string associated to the leaf nodes

An Example*

Parse Tree(E)

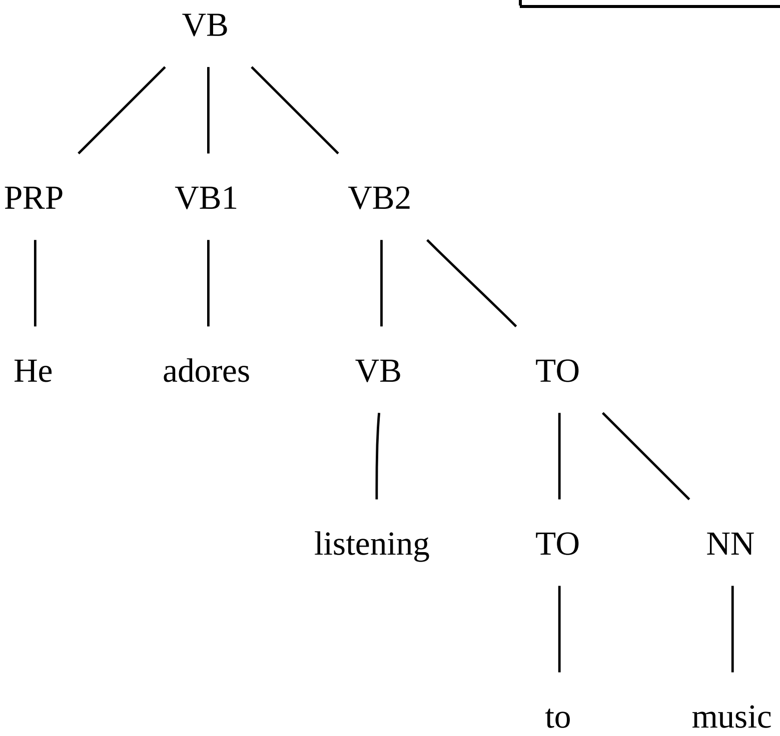


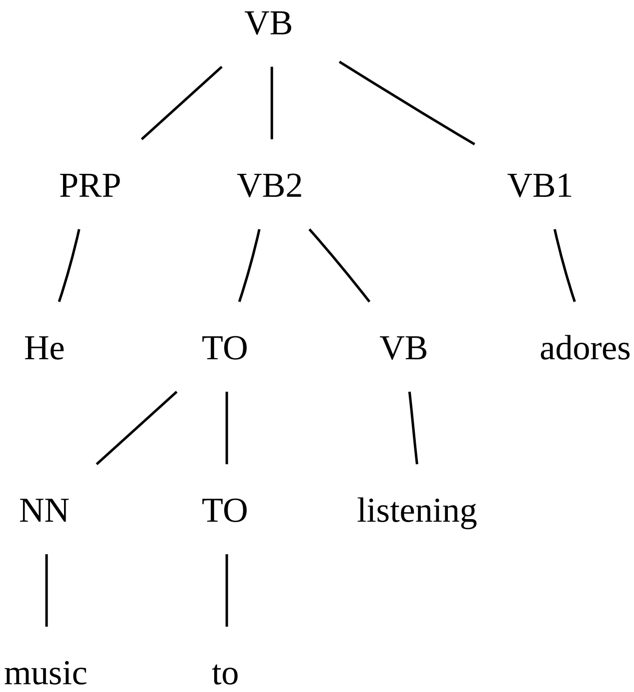
Sentence(J) *Kare ha ongaku wo kiku no ga daisuki desu*

*Source: `http://www.isi.edu/natural-language/people/cs562-8-22-06.pdf`

Re-ordering table: *r-table*

original order	reordering	P(reorder)
	PRP VB1 VB2	0.074
	PRP VB2 VB1	0.723
PRP VB1 VB2	VB1 PRP VB2	0.061
	...	...
VB TO	VB TO	0.252
	TO VB	0.749
TO NN	TO NN	0.107
	NN TO	0.893
	...	...



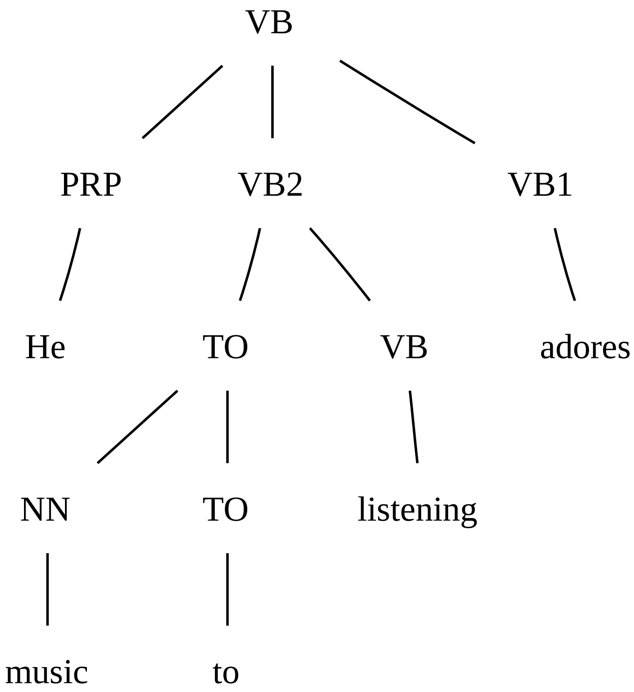


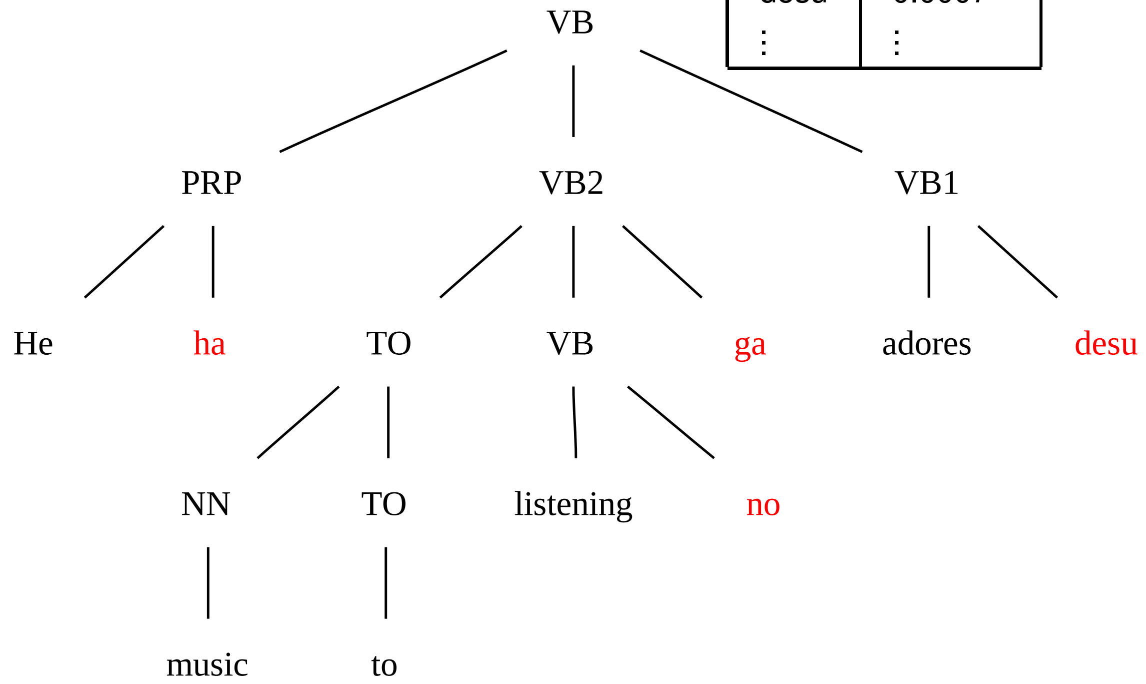
Reordering probability: $0.723 \cdot 0.749 \cdot 0.893 = 0.484$

Insertion table: *n-table*

w	P(ins-w)
ha	0.219
ta	0.131
wo	0.099
no	0.094
ni	0.080
te	0.078
ga	0.062
⋮	⋮
desu	0.0007
⋮	⋮

parent	TOP	VB	VB	VB	TO	TO	...
node	VB	VB	PRP	TO	TO	NN	...
P(None)	0.735	0.687	0.344	0.709	0.900	0.800	...
P(Left)	0.004	0.061	0.004	0.030	0.003	0.096	...
P(right)	0.260	0.252	0.652	0.261	0.007	0.104	...

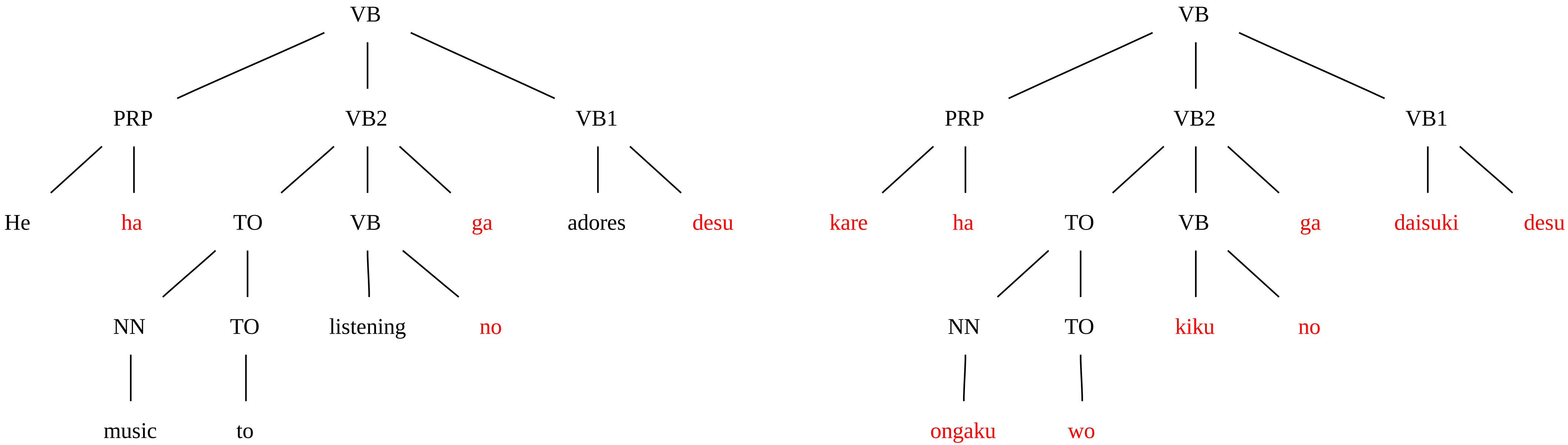




Insertion probability: $(0.652 \cdot 0.219) \cdot (0.252 \cdot 0.094) \cdot (0.252 \cdot 0.062) \cdot (0.252 \cdot 0.0007) \cdot 0.735 \cdot 0.709 \cdot 0.900 \cdot 0.800 = 3.498 \exp -9$

Insertion table: *t-table*

adores		he		listening		music		to		...
daisuki	1.000	kare	0.952	kiku	0.333	ongaku	0.900	ni	0.216	...
		NULL	0.016	kii	0.333	naru	0.100	NULL	0.204	
		nani	0.005	mi	0.333			to	0.133	
		⋮	⋮	⋮	⋮			⋮	⋮	



Translation probability: $0.952 \cdot 0.900 \cdot 0.038 \cdot 1.000 = 0.0108$

Similar ideas

- Tree-to-String Alignment Template for Statistical Machine Translation [Liu 06]
- A Forest-to-String Machine Translation Engine based on Tree Transducers [Neubig 13]*

*http://saffron.insight-centre.org/acl/topic/tree_to_string_translation/publications/
http://www.phontron.com/travatar/

Index

Formal definitions

- **Goal:** Transform an English parse tree $\mathcal{E}$ into a French sentence $\mathbf{f}$
- **Definitions**
  - $\mathcal{E}$ consists of nodes $\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_n$ (parent nodes, and sons or leafs)
  - $\mathbf{f}$ consists of words $f_1, f_2, \ldots, f_m$
  - $\theta_i = (\nu_i, \rho_i, \tau_i)$ is a set of values of random variables associated to ε_i
    - ν_i: variable representing an insertion (in the parent node)
    - ρ_i: variable representing a re-ordering (of the sons/siblings)
    - τ_i: variable representing a translation (in the leaf nodes)
  - $\boldsymbol{\theta} = \theta_1, \theta_2, \ldots, \theta_n$ is the set of all random variables associated with a parse tree $\mathcal{E} = \varepsilon_1, \varepsilon_2, \ldots, \varepsilon_n$
- **Model**

$$P(\mathbf{f}, \mathbf{e}) = P(\mathbf{e})P(\mathbf{f}|\mathbf{e}) = \sum_{\mathcal{E}:L(\mathcal{E})=\mathbf{e}} P(\mathbf{e}, \mathcal{E})P(\mathbf{f}|\mathbf{e}) \approx \max_{\mathcal{E}:L(\mathcal{E})=\mathbf{e}} P(\mathbf{e}, \mathcal{E})P(\mathbf{f}|\mathbf{e}) \triangleq P(\mathbf{f}|\mathcal{E})P(\mathbf{e})$$

Formal definitions

- **Decoding problem:**

$$\widehat{\mathbf{f}} = \arg\max P(\mathbf{f}|\mathcal{E})P(\mathbf{e})$$

where $\mathbf{e}, \mathcal{E}$ is a sentence/tree pair in English

- **Translation model**

$$P(\mathbf{f}|\mathcal{E}) = \sum_{\boldsymbol{\theta}:L(\boldsymbol{\theta}(\mathcal{E}))=\mathbf{f}} P(\boldsymbol{\theta}|\mathcal{E})$$

where

$$\begin{aligned} P(\boldsymbol{\theta}|\mathcal{E}) &= P(\theta_1, \theta_2, \ldots, \theta_n | \varepsilon_1, \varepsilon_2, \ldots, \varepsilon_n) \\ &= \prod_{i=1}^{n} P(\theta_i | \theta_1, \theta_2, \ldots, \theta_{i-1}, \varepsilon_1, \varepsilon_2, \ldots, \varepsilon_n) \\ &\approx \prod_{i=1}^{n} P(\theta_i | \varepsilon_i) \end{aligned}$$

Formal description

$$
\begin{aligned}
P(\theta_i|\varepsilon_i) = P(\nu_i, \rho_i, \tau_i|\varepsilon_i) &\approx P(\nu_i|\varepsilon_i)P(\rho_i|\varepsilon_i)P(\tau_i|\varepsilon_i) \\
&= P(\nu_i|\mathcal{N}(\varepsilon_i))P(\rho_i|\mathcal{R}(\varepsilon_i))P(\tau_i|\mathcal{T}(\varepsilon_i)) \\
&= n(\nu_i|\mathcal{N}(\varepsilon_i))r(\rho_i|\mathcal{R}(\varepsilon_i))t(\tau_i|\mathcal{T}(\varepsilon_i))
\end{aligned}
$$

where

$$
n(\nu|\mathcal{N}(\varepsilon)) \equiv n(\nu|_{\mathcal{N}}), \quad r(\rho|\mathcal{R}(\varepsilon)) \equiv r(\rho|_{\mathcal{R}}), \quad t(\tau|\mathcal{T}(\varepsilon)) \equiv t(\tau|_{\mathcal{T}})
$$

are the parameters of the model

For example:

- $n(\nu|_{\mathcal{N}}) = P(\mathrm{right}, \textit{ha}|\mathrm{VB} - \mathrm{PRP})$
- $r(\rho|_{\mathcal{R}}) = P(\mathrm{PRP} - \mathrm{VB2} - \mathrm{VB1}|\mathrm{PRP} - \mathrm{VB1} - \mathrm{VB2})$

$$
\boxed{P(\mathbf{f}|\mathcal{E}) = \sum_{\boldsymbol{\theta}:L(\boldsymbol{\theta}(\mathcal{E}))=\mathbf{f}} \prod_{i=1}^{n} n(\nu_i|\mathcal{N}(\varepsilon_i))r(\rho_i|\mathcal{R}(\varepsilon_i))t(\tau_i|\mathcal{T}(\varepsilon_i))}
$$

Intuition: $\mathbf{f}$ can be computed from all possible trees $\mathcal{E}$ that are able to generate $\mathbf{f}$. Each possible tree can be obtained by applying the transformation operations previously defined

Index

- Introduction
- Formal definitions
- **Stochastic estimation of the model**
- Decoding
- References
- Exercises

Stochastic estimation of the model

Theorem [Baum 72] Let $P(\Theta)$ be a homogeneous polynomial with non-negative coefficients. Let $\theta = \{\theta_{ij}\}$ be a point in the domain $D = \{\theta_{ij} \mid \theta_{ij} \geq 0; \sum_{j=1}^{q_i} \theta_{ij} = 1,\ i = 1, \ldots, p; \quad j = 1, \ldots, q_i\}$, and let $Q(\theta)$ be a close transformation in D, that is defined as:

$$Q(\theta)_{ij} = \frac{\theta_{ij}(\partial P / \partial \Theta_{ij})_\theta}{\sum_{k=1}^{q_i} \theta_{ik}(\partial P / \partial \Theta_{ik})_\theta}$$

with the denominator different from zero. Then, $P(Q(\theta)) > P(\theta)$ except if $Q(\theta) = \theta$.

```
input P(Θ)
θ = initial values
repeat
    compute Q(θ) using P(Θ)
    θ = Q(θ)
until convergence
output θ
```

Stochastic estimation of the model

In the previous theorem, the homogeneous polynomial is defined according to $P(\mathbf{f}, \mathcal{E})$ and the set of parameters of the model is a point according to the theorem. The log-likelihood for a training sample is defined as:

$$\ln \prod_{\mathbf{f},\mathcal{E}:L(\mathcal{E})=\mathbf{f}} P(\mathbf{f}, \mathcal{E})$$

Applying previous theorem to one of the parameters, e.g., $\nu_i = n_i = n(\nu|\mathcal{N})$, such that $\boldsymbol{\theta} = \theta_1, \theta_2, \ldots, \theta_n$, and $\theta_i = (\nu_i, \rho_i, \tau_i)$:

$$\begin{aligned}
\overline{n}_i &= \frac{n_i \left(\frac{\partial \ln \prod_{\mathbf{f},\mathcal{E}:L(\mathcal{E})=\mathbf{f}} P(\mathbf{f},\mathcal{E})}{\partial n_i}\right)_\theta}{\sum_i n_i \left(\frac{\partial \ln \prod_{\mathbf{f},\mathcal{E}:L(\mathcal{E})=\mathbf{f}} P(\mathbf{f},\mathcal{E})}{\partial n_i}\right)_\theta} = \frac{\sum_{\mathbf{f},\mathcal{E}:L(\mathcal{E})=\mathbf{f}} \frac{1}{P(\mathbf{f},\mathcal{E})}\, n_i \left(\frac{\partial P(\mathbf{f},\mathcal{E})}{\partial n_i}\right)_\theta}{\sum_{\mathbf{f},\mathcal{E}:L(\mathcal{E})=\mathbf{f}} \frac{1}{P(\mathbf{f},\mathcal{E})} \sum_i n_i \left(\frac{\partial P(\mathbf{f},\mathcal{E})}{\partial n_i}\right)_\theta} \\
&= \frac{\sum_{\mathbf{f},\mathcal{E}:L(\mathcal{E})=\mathbf{f}} \frac{1}{P(\mathbf{f},\mathcal{E})} \sum_{\theta:L(\theta(\mathcal{E})=\mathbf{f})} \mathrm{N}(n_i, \theta) \prod_{i=1}^{n} n_i\, r_i\, t_i}{\sum_{\mathbf{f},\mathcal{E}:L(\mathcal{E})=\mathbf{f}} \frac{1}{P(\mathbf{f},\mathcal{E})} \sum_{\theta:L(\theta(\mathcal{E})=\mathbf{f})} \sum_i \mathrm{N}(n_i, \theta) \prod_{j=1}^{n} n_i\, r_j\, t_j}
\end{aligned}$$

Stochastic estimation of the model

1. Initialize all probability tables: $n(\nu|\mathcal{N})$, $r(\rho|\mathcal{R})$ and $t(\tau|\mathcal{T})$
2. Reset all counters: $c(\nu, \mathcal{N})$, $c(\rho, \mathcal{R})$ and $c(\tau, \mathcal{T})$
3. For each pair $\langle \mathcal{E}, \mathbf{f} \rangle$ in the training corpus

   For all $\boldsymbol{\theta}$, such that $\mathbf{f} = L(\boldsymbol{\theta}(\mathcal{E}))$

   - Let cnt $= P(\boldsymbol{\theta}|\mathcal{E}) / \sum_{\boldsymbol{\theta}:L(\boldsymbol{\theta}(\mathcal{E}))=\mathbf{f}} P(\boldsymbol{\theta}|\mathcal{E})$
   - For $i = 1 \ldots n$,

     $c(\nu_i, \mathcal{N}(\varepsilon_i)) += \text{cnt}$

     $c(\rho_i, \mathcal{R}(\varepsilon_i)) += \text{cnt}$

     $c(\tau_i, \mathcal{T}(\varepsilon_i)) += \text{cnt}$
4. For each $\langle \nu, \mathcal{N} \rangle$, $\langle \rho, \mathcal{R} \rangle$, and $\langle \tau, \mathcal{T} \rangle$

   $n(\nu|\mathcal{N}) = c(\nu, \mathcal{N}) / \sum_{\nu} c(\nu, \mathcal{N})$

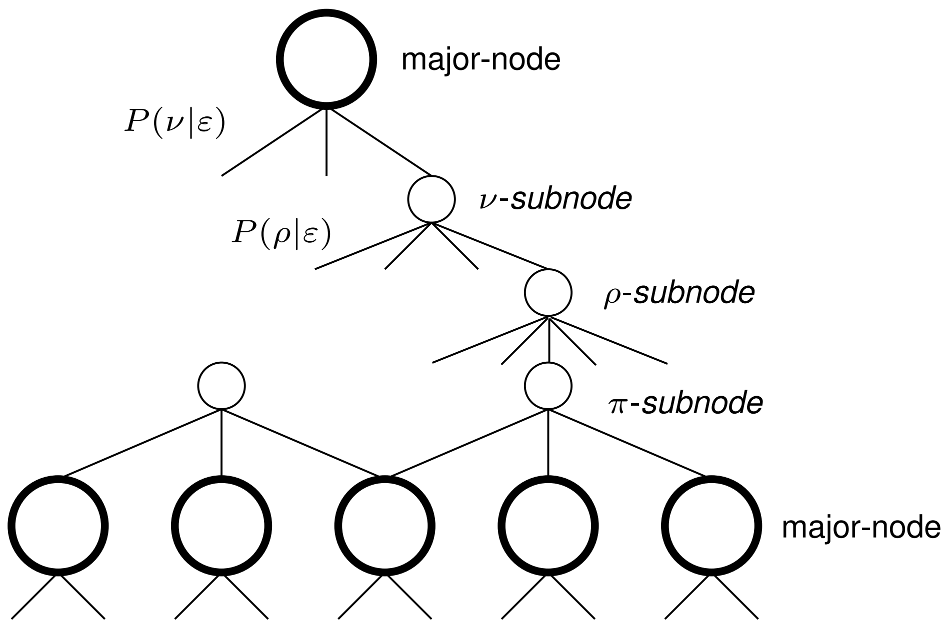
   $r(\rho|\mathcal{R}) = c(\rho, \mathcal{R}) / \sum_{\rho} c(\rho, \mathcal{R})$

   $t(\tau|\mathcal{T}) = c(\tau, \mathcal{T}) / \sum_{\tau} c(\tau, \mathcal{T})$
5. Repeat steps 2-4 for several iterations

Efficient EM training

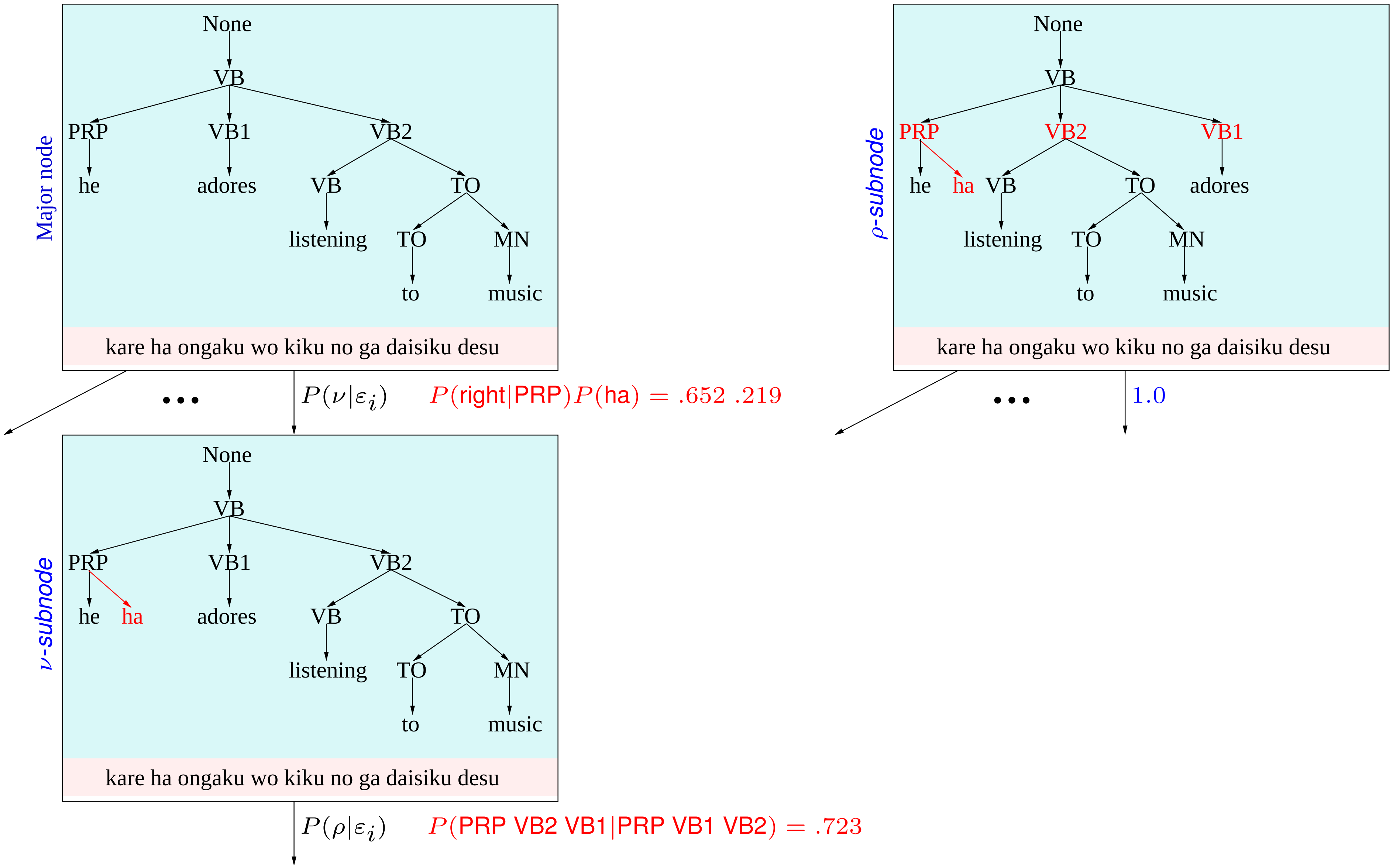
The EM algorithm uses a graph structure for a pair $\langle \mathcal{E}, \mathbf{f} \rangle$

- A *major-node* $v(\varepsilon_i, \mathbf{f}_k^l)$ shows a pairing of a subtree of $\mathcal{E}$ and a substring of $\mathbf{f}$
- Each major node connects to several ν-*subnode* $v(\nu; \varepsilon_i, \mathbf{f}_k^l)$, showing which value of ν is selected. The arc has weight $P(\nu|\varepsilon_i)$
- A ν-*subnode* $v(\nu; \varepsilon_i, \mathbf{f}_k^l)$ connects to a *final-node* with weight $P(\tau|\varepsilon_i)$ if ε_i is a terminal node
- A ν-*subnode* connects to several ρ-*subnodes* $v(\rho; \nu, \varepsilon_i, \mathbf{f}_k^l)$ with weight $P(\rho|\varepsilon_i)$
- A ρ-subnode is connected to π-subnodes $v(\pi; \rho, \nu, \varepsilon_i, \mathbf{f}_k^l)$ with weight 1.0. The variable π shows a particular way of partitioning $\mathbf{f}_k^l$

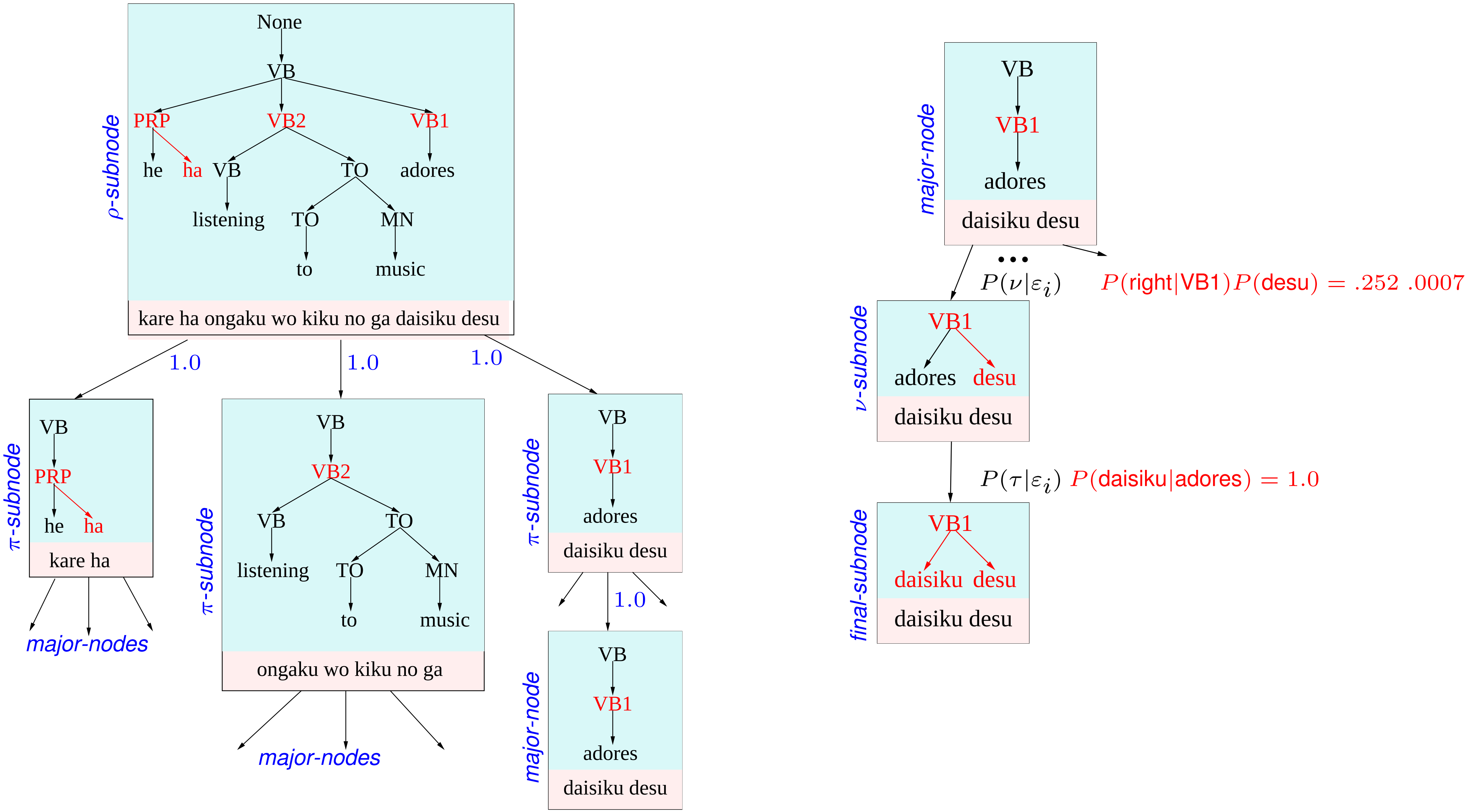


- A π-subnode is connected to major-nodes corresponding to children of ε_i with weight 1.0. A major-node can be connected from different π-subnodes

Efficient EM training: example (I)



Efficient EM training: example (II)



Efficient EM training

- The structure generated in the previous algorithm resembles a trellis generated when analizing a string with a final-state machine

- A trace starting from the graph root, selecting one of the arcs from major-nodes, ν-*subnodes* and ρ-subnodes and *all* the arcs from π-subnodes corresponds to a particular to a particular $\boldsymbol{\theta}$

- The addition of the weight of all possible $\boldsymbol{\theta}$ correspond to $P(\boldsymbol{\theta}|\mathcal{E})$

- An estimation algorithm similar to the forward-backward algorithm can be defined

- The time complexity is $O(n^3|\nu||\rho||\pi|)$

Index

Decoder description [Yamada 01b]

Modifications to the original MT for phrasal translations:

- Fertility μ is used to allow 1-to-N mapping:

$$t(\tau|\tau) = t(f_1 f_2 \dots f_l|e) = \mu(l|e) \prod_{i=1}^{l} t(f_i|e)$$

- Direct translation ϕ of an English phrase $e_1 e_2 \dots e_m$ to a foreign phrase $f_1 f_2 \dots f_l$ at non-terminal tree nodes:

$$ph(\phi|\Phi) = t(f_1 f_2 \dots f_l|e_1 e_2 \dots e_m) = \mu(l|e_1 e_2 \dots e_m) \prod_{i=1}^{l} t(f_i|e_1 e_2 \dots e_m)$$

- Linear combination (if ε_i is non-terminal):

$$P(\theta_i|\varepsilon_i) = \lambda_{\Phi_i} ph(\phi_i|\Phi_i) + (1 - \lambda_{\Phi_i}) r(\rho_i|\mathcal{R}_i) n(\nu_i|\mathcal{N}_i)$$

Decoder description

- Given a French sentence, the decoder will find the most plausible English parse tree
- Idea: a mechanism similar to normal parsing is used
- Steps:

1. Start from an English context-free grammar and incorporate to it the channel operations
2. For each non-lexical rule (such as “VP $\rightarrow$ VB NP PP”), supplement the grammar with reordered rules and probabilities are taken from the r-table
3. Rules such as “VP $\rightarrow$ VP X” and “X $\rightarrow$ *word”* are added and probabilities are taken from the n-table
4. For each lexical rule in the English grammar, we add rules such as “englishWord $\rightarrow$ foreingWord”
5. Parse a string of foreign words
6. Undo reordering operations and remove leaf nodes with foreign words
7. Among all possible tree, choose pick the best in which the product of the LM and the TM probability is the highest

Index

References

- [Baum 72] L.E. Baum: *An inequality and associated maximization technique in statistical estimation for probabilistic functions of markov processes.* Inequalities, 3:1-8, 1972.
- [Yamada 01a] K. Yamada, K. Knight: *A Syntax-Based Statistical Translation Model.* ACL 2001.
- [Yamada 01b] K. Yamada, K. Knight: *A decoder for Syntax-based Statistical MT.* ACL 2001.
- Y. Liu, Q. Liu, S. Lin: *Tree-to-String Alignment Template for Statistical Machine Translation*. ACL 2006.
- J. Graehl, K. Knight, J May: *Training Tree Transducers*. Computational Linguistics 34 (3), 391-427, 2008.
- [Neubig 13] G. Neubig: *Travatar: A Forest-to-String Machine Translation Engine based on Tree Transducers*. ACL 2013.

Index

Exercises

1. **(*)** Write an example similar to the slides in the pages 4–7 with the pair of languages that your prefer. Define the probabilities as you like.
2. **(*)** Compute the derivative of expression that you can see at the end of slide 11 with respect to any variable that you choose. Use the example that you can see in slides 14–16. Explain the obtained expression.
3. **(***)** Define a small model: a parsing tree, a r-table, an n-table and a t-table. Then, apply the estimation algorithm described in slide 15. Keep in mind that the models should be small enough to make the computations easy. Therefore the number of posible transformation should keep very small.
4. **(***)** Repeat the previus exercise but applying the algorithm described in slide 17, taking into acount the example shown in class.
5. **(****)** Reproduce the experiment with travatar system and aply the toolkit to another dataset different from the dataset provided with the toolkit.
6. **(*****)** Implement the estimation of the model described in slides 16–17.
7. **(*****)** Implement and test the translation algorithm described in slides 20-21.

Universitat Politècnica de València

Máster en Inteligencia Artificial, Reconocimiento de Formas e Imagen Digital

MACHINE TRANSLATION

Hierarchical machine translation

Joan Andreu Sánchez

jandreu@prhlt.upv.es

Hierarchical MT

Main ideas [Chiang 07]

- It allows to capture difficult reordering
- Hierarchical phrases: phrases that can contain other phrases
- Related to Synchronous CFG: useful for specifying relations between languages
- Rules are as follows:

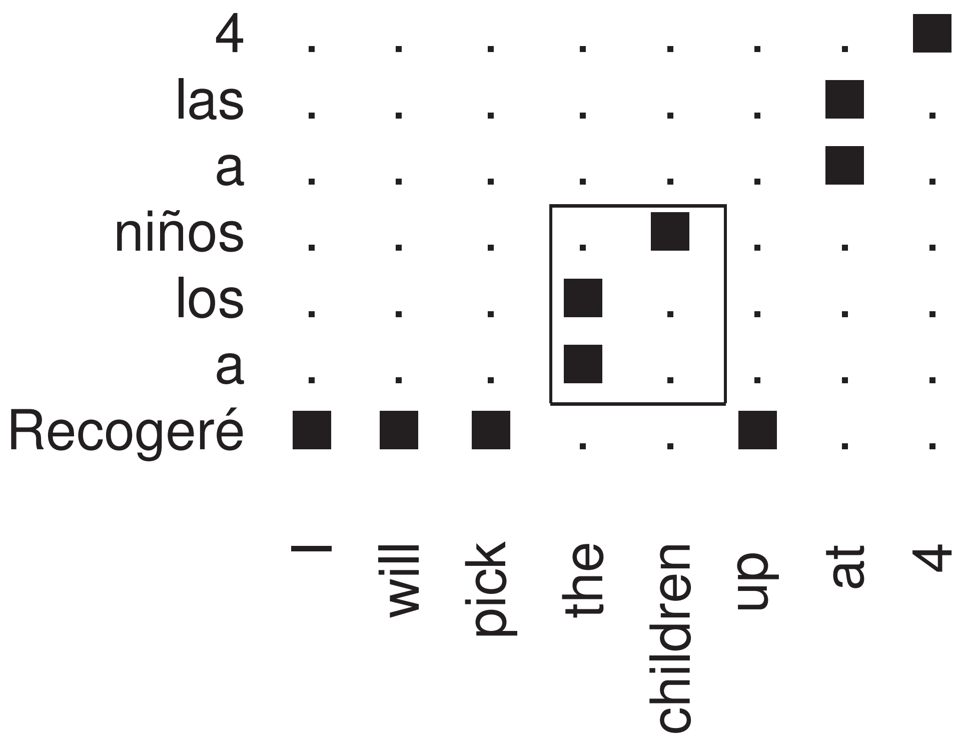
$$X \rightarrow \langle \gamma, \alpha, \sim \rangle$$

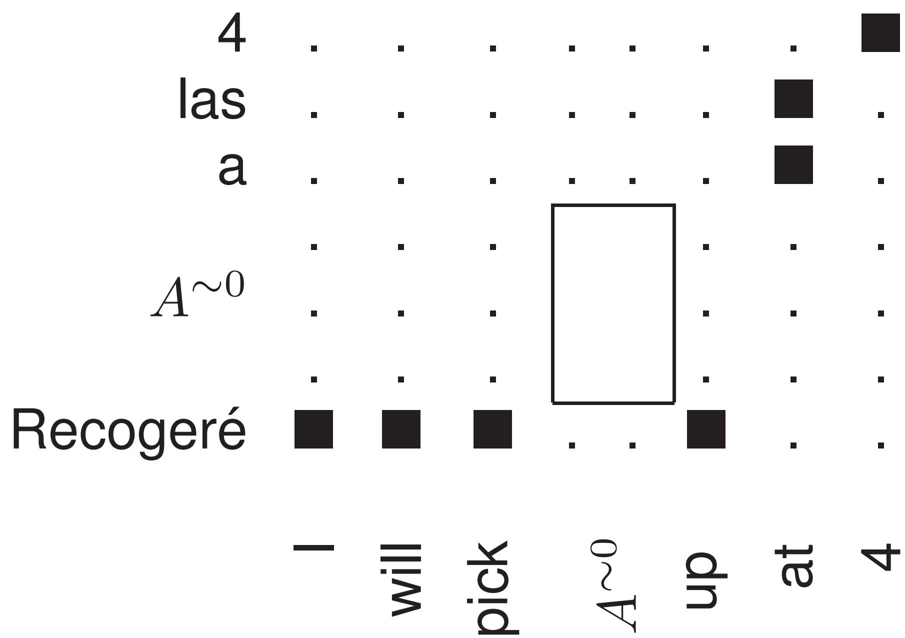
where

- X is a non-terminal symbol
- γ, α are strings of terminal and non-terminal symbols
- $\sim$ is one-to-one correspondence between non-terminal ocurrences in γ and α

Hierarchical MT

Motivation





Hierarchical MT

Rule extraction

- Rules are extracted from word-alignments sentences
  - Extract a rule for each phrase pair
  - Replace phra se pairs in each rule by a non-terminal symbol if another rule produces that phrase pair.

- The set of rules of two word-aligned sentences $\langle f, e, \sim \rangle$ is the smallest set satisfying the following:
  - If $\langle f_i^j, e_{i'}^{j'} \rangle$ is an initial phrase pair, then add the following rule:

$$X \rightarrow \langle f_i^j, e_{i'}^{j'} \rangle$$

  - If $(X \rightarrow \langle \gamma, \alpha \rangle)$ is a rule and $\langle f_i^j, e_{i'}^{j'} \rangle$ is an initial phrase pair such that $\gamma = \gamma_1 f_i^j \gamma_2$ and $\alpha = \alpha_1 e_{i'}^{j'} \alpha_2$, then add the following rule:

$$X \rightarrow \langle \gamma_1 X_k \gamma_2, \alpha_1 X_k \alpha_2 \rangle$$

- Glue rules:

$$S \rightarrow \langle S_1 X_2, S_1 X_2 \rangle$$
$$S \rightarrow \langle X_1, X_1 \rangle$$

Hierarchical MT

Some restrictions to alleviate computational complexity:

- at most two non-terminal symbols
- at least one but at most five words per language
- span at most 15 words (counting gaps)

Hierarchical MT

Translation model

- Log-linear model over derivations:

$$P(D) \propto \prod_i \Phi_i(D)^{\lambda_i}$$

  where Φ_i are features defined on derivations and λ_i are feature weights

- Features: functions on the rules and an additonal LM function:

$$P(D) \propto P_{LM}(e)^{\lambda_{LM}} \prod_{i \neq LM} \prod_{(X \rightarrow \langle \gamma, \alpha \rangle) \in D} \Phi_i(X \rightarrow \langle \gamma, \alpha \rangle)^{\lambda_i}$$

- Features on rules:
  - $P(\gamma \mid \alpha)$ and $P(\alpha \mid \gamma)$
  - Lexical weights: $P_w(\gamma \mid \alpha)$ and $P_w(\alpha \mid \gamma)$
  - A penalty $\exp(-1)$ to learn a preference for longer or shorter derivations
  - Word penalty: $\exp(-\#T(\alpha))$

Hierarchical MT

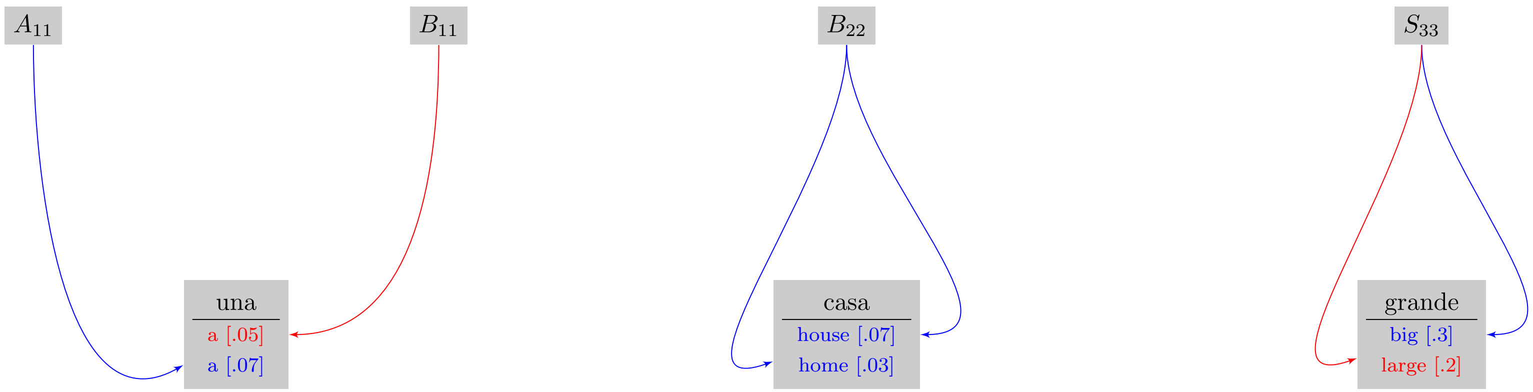
Training

- Rules probabilities obtained from frequencies
- λ_i: minimum-error-rate training [Och 02]
- CKY-based algorithm

Hierchical MT

The search problem: example

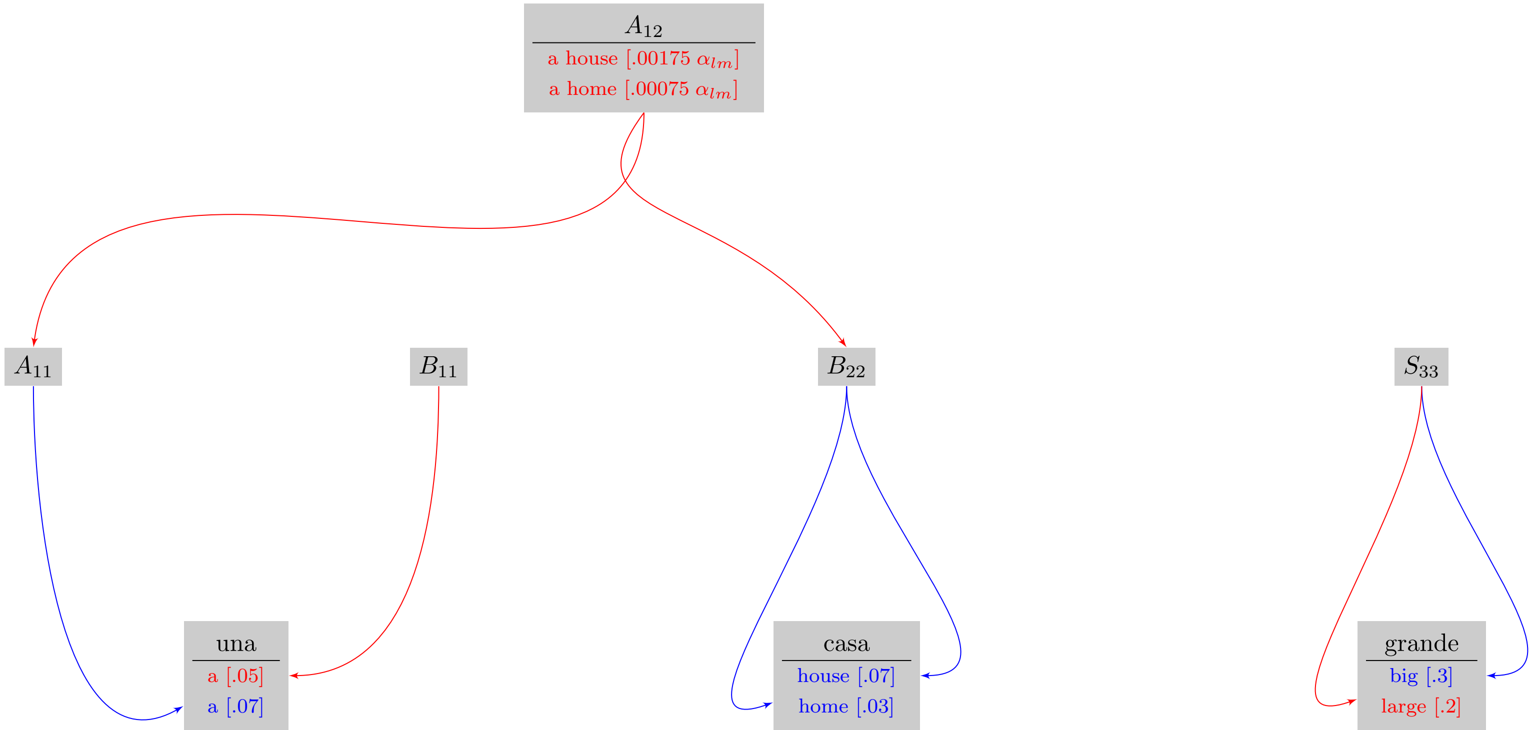
.3	$A \rightarrow AB$
.1	$S \rightarrow [AS]$
.4	$S \rightarrow [AB]$
.3	$S \rightarrow$ grande|big
.2	$S \rightarrow$ grande|large
.5	$A \rightarrow [AB]$
.4	$A \rightarrow < BB >$
.05	$A \rightarrow$ una|a
.03	$A \rightarrow$ el|the
.02	$A \rightarrow$ la|the
.7	$B \rightarrow < BS >$
.07	$B \rightarrow$ casa|house
.03	$B \rightarrow$ casa|home
.1	$B \rightarrow$ coche|car
.03	$B \rightarrow$ un|a
.07	$B \rightarrow$ una|a



Hierchical MT

The search problem: example

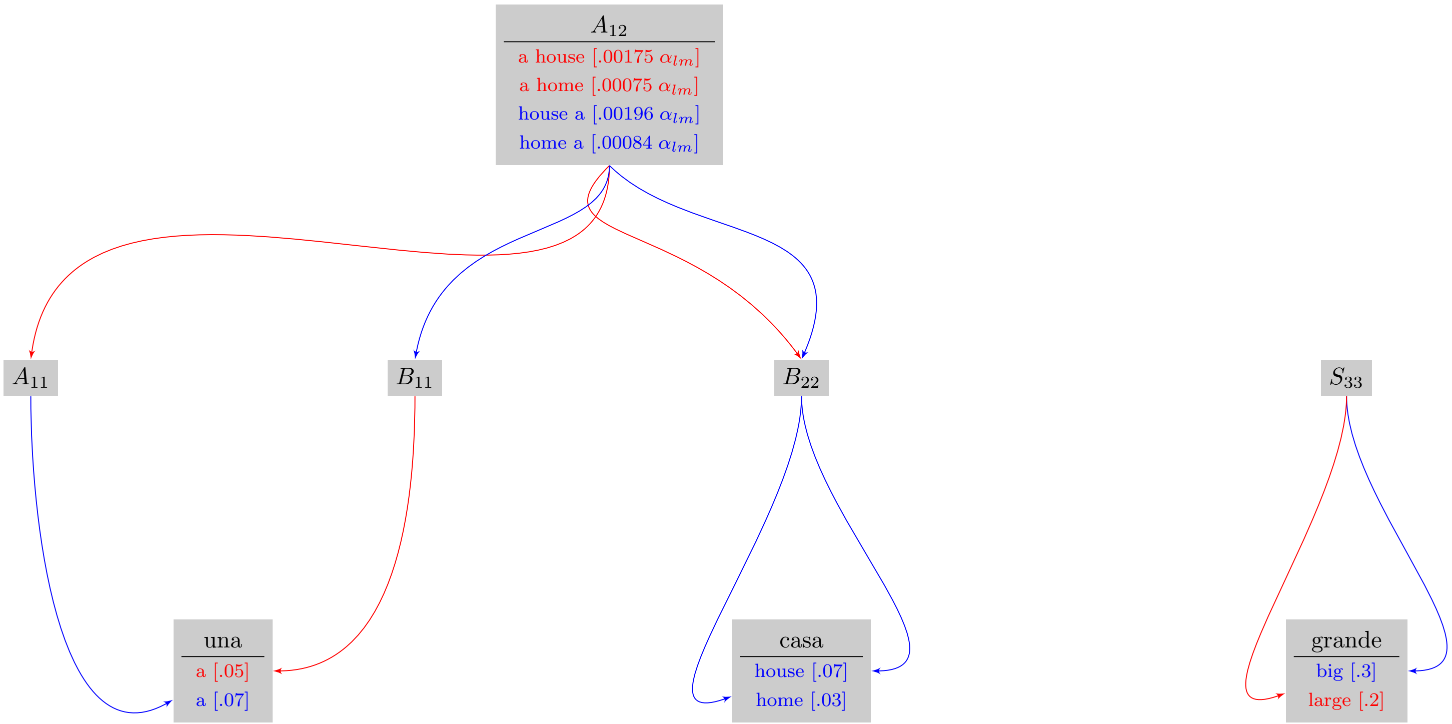
.3 $A \to AB$
.1 $S \to [AS]$
.4 $S \to [AB]$
.3 $S \to$ grande|big
.2 $S \to$ grande|large
.5 $A \to [AB]$
.4 $A \to < BB >$
.05 $A \to$ una|a
.03 $A \to$ el|the
.02 $A \to$ la|the
.7 $B \to < BS >$
.07 $B \to$ casa|house
.03 $B \to$ casa|home
.1 $B \to$ coche|car
.03 $B \to$ un|a
.07 $B \to$ una|a



Hierchical MT

The search problem: example

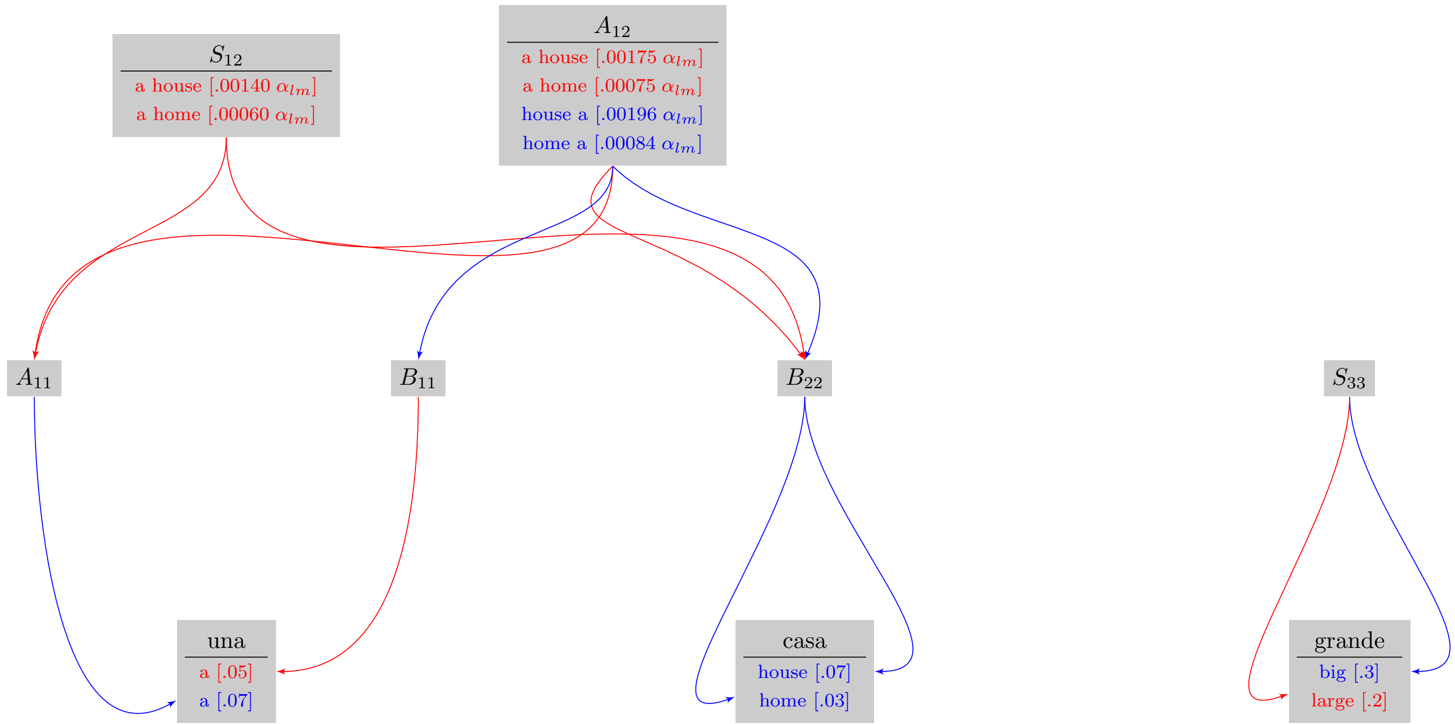
.3 $A \rightarrow AB$
.1 $S \rightarrow [AS]$
.4 $S \rightarrow [AB]$
.3 $S \rightarrow$ grande|big
.2 $S \rightarrow$ grande|large
.5 $A \rightarrow [AB]$
.4 $A \rightarrow < BB >$
.05 $A \rightarrow$ una|a
.03 $A \rightarrow$ el|the
.02 $A \rightarrow$ la|the
.7 $B \rightarrow < BS >$
.07 $B \rightarrow$ casa|house
.03 $B \rightarrow$ casa|home
.1 $B \rightarrow$ coche|car
.03 $B \rightarrow$ un|a
.07 $B \rightarrow$ una|a



Hierchical MT

The search problem: example

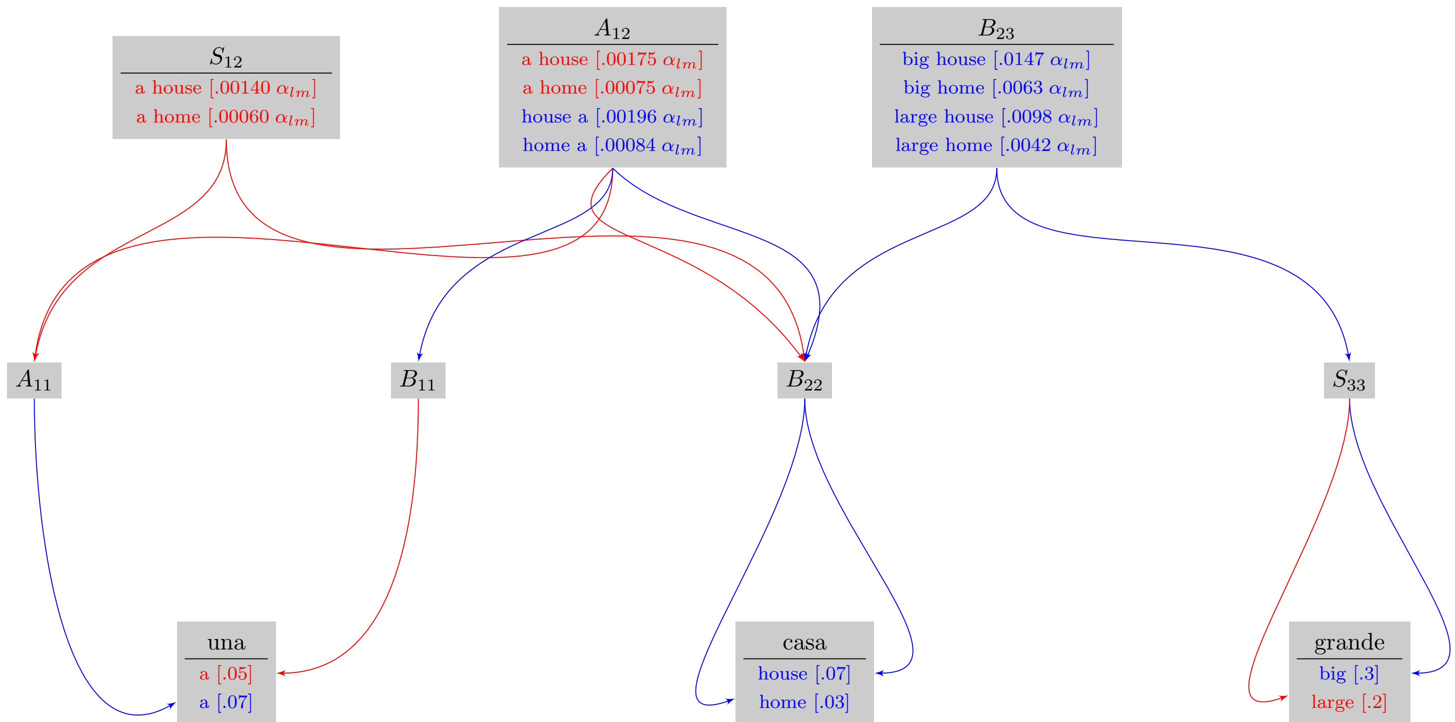
.3	$A \rightarrow AB$
.1	$S \rightarrow [AS]$
.4	$S \rightarrow [AB]$
.3	$S \rightarrow$ grande|big
.2	$S \rightarrow$ grande|large
.5	$A \rightarrow [AB]$
.4	$A \rightarrow < BB >$
.05	$A \rightarrow$ una|a
.03	$A \rightarrow$ el|the
.02	$A \rightarrow$ la|the
.7	$B \rightarrow < BS >$
.07	$B \rightarrow$ casa|house
.03	$B \rightarrow$ casa|home
.1	$B \rightarrow$ coche|car
.03	$B \rightarrow$ un|a
.07	$B \rightarrow$ una|a



Hierchical MT

The search problem: example

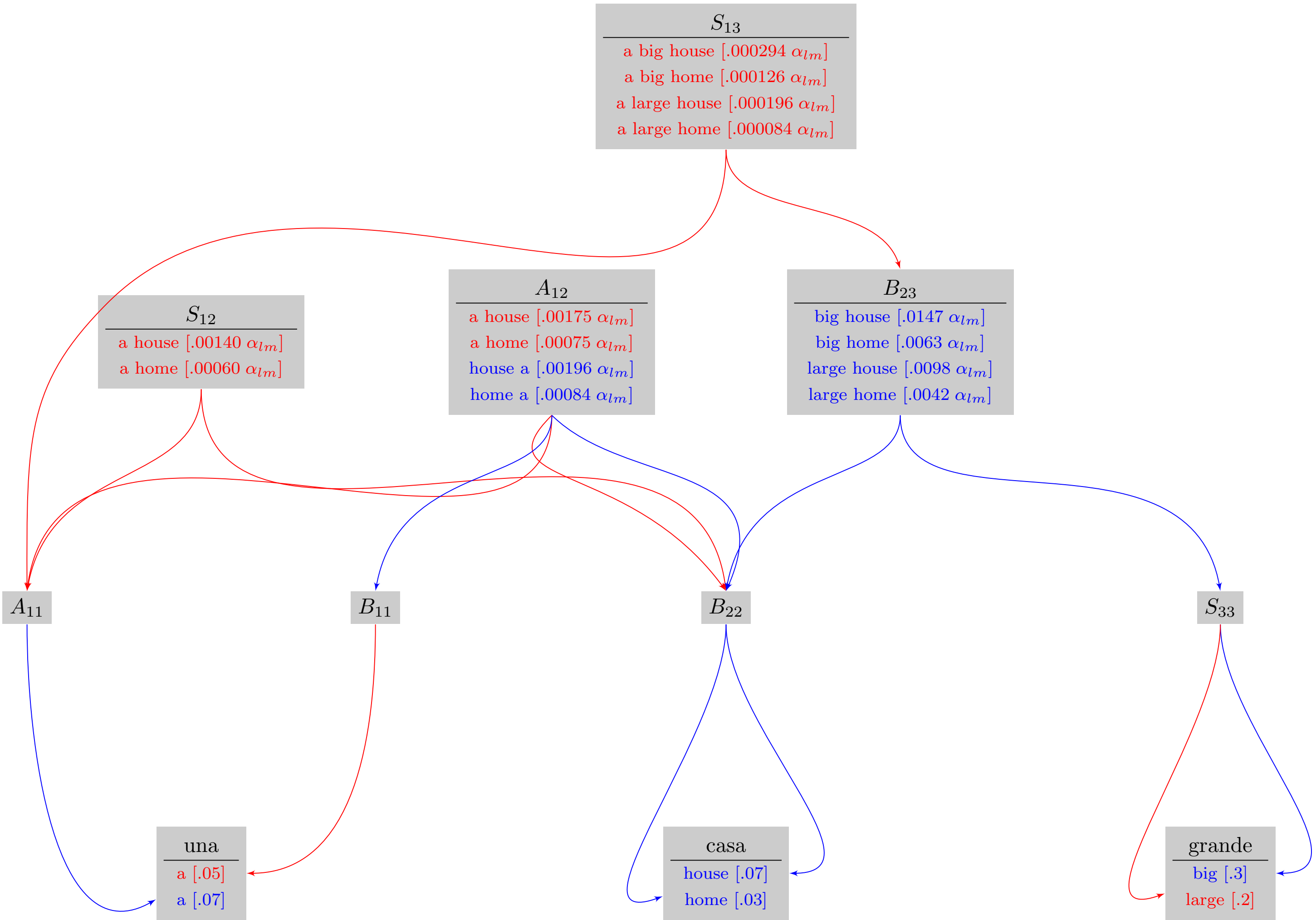
.3	$A \rightarrow AB$
.1	$S \rightarrow [AS]$
.4	$S \rightarrow [AB]$
.3	$S \rightarrow$ grande|big
.2	$S \rightarrow$ grande|large
.5	$A \rightarrow [AB]$
.4	$A \rightarrow < BB >$
.05	$A \rightarrow$ una|a
.03	$A \rightarrow$ el|the
.02	$A \rightarrow$ la|the
.7	$B \rightarrow < BS >$
.07	$B \rightarrow$ casa|house
.03	$B \rightarrow$ casa|home
.1	$B \rightarrow$ coche|car
.03	$B \rightarrow$ un|a
.07	$B \rightarrow$ una|a



Hierchical MT

The search problem: example

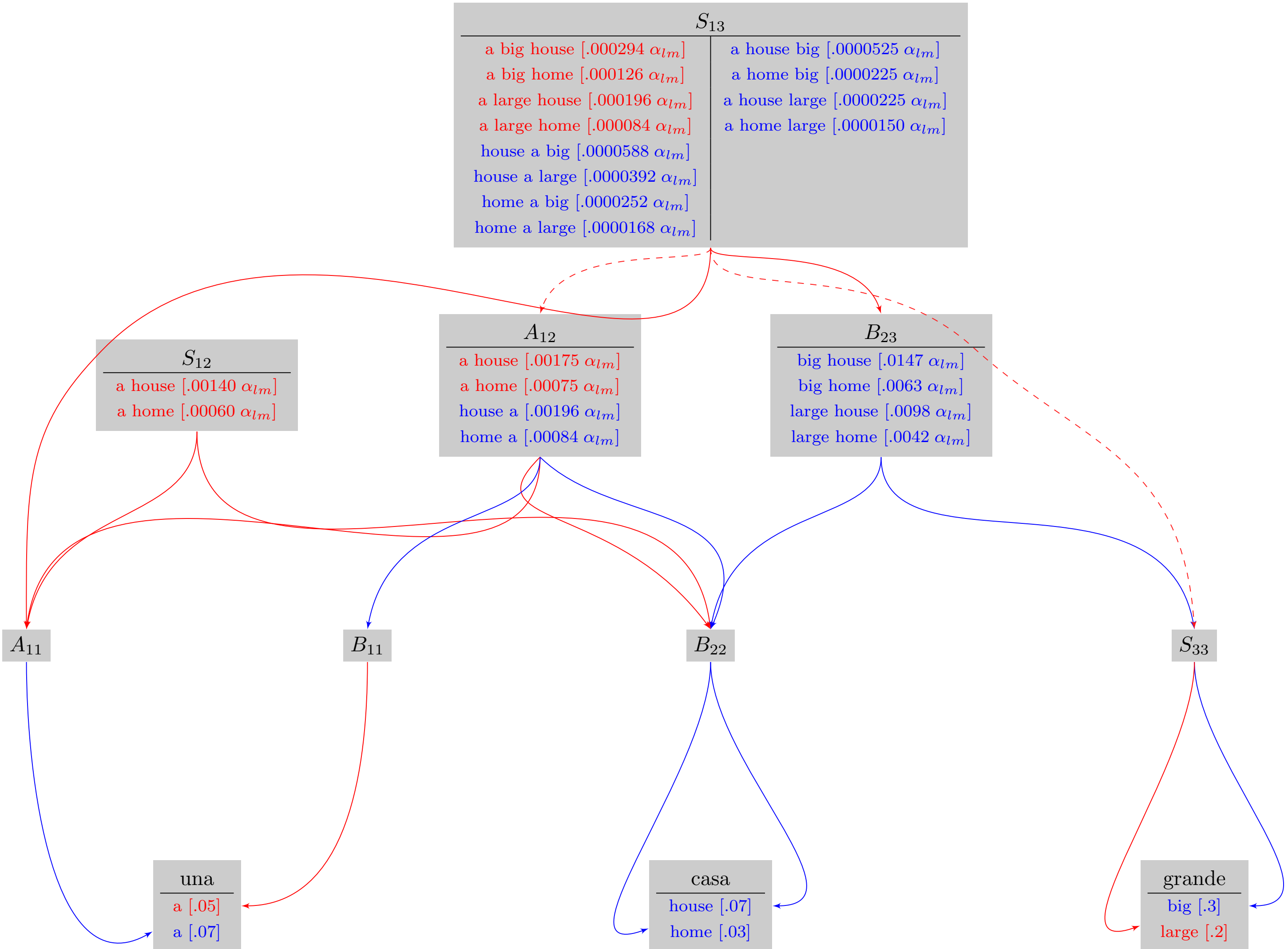
.3	$A \to AB$
.1	$S \to [AS]$
.4	$S \to [AB]$
.3	$S \to$ grande|big
.2	$S \to$ grande|large
.5	$A \to [AB]$
.4	$A \to < BB >$
.05	$A \to$ una|a
.03	$A \to$ el|the
.02	$A \to$ la|the
.7	$B \to < BS >$
.07	$B \to$ casa|house
.03	$B \to$ casa|home
.1	$B \to$ coche|car
.03	$B \to$ un|a
.07	$B \to$ una|a



Hierchical MT

The search problem: example

.3	$A \to AB$
.1	$S \to [AS]$
.4	$S \to [AB]$
.3	$S \to$ grande|big
.2	$S \to$ grande|large
.5	$A \to [AB]$
.4	$A \to < BB >$
.05	$A \to$ una|a
.03	$A \to$ el|the
.02	$A \to$ la|the
.7	$B \to < BS >$
.07	$B \to$ casa|house
.03	$B \to$ casa|home
.1	$B \to$ coche|car
.03	$B \to$ un|a
.07	$B \to$ una|a



Exercises

1. **(*)** Write an example of an alignment between two sentences and the rules that can be obtained with that alignment.
2. **(*)** Look for a reference related to hierarchical models published in that last two years a write a summary with the main contributions of that paper.

References

- [Chiang 07] D. Chiang: *Hierarchical phrase-based translation.* Computational Linguistics, 33(2), 2007.
- [Koehn 10] P. Koehn: *Statistical Machine Translation* Cambridge, University Press, 2010.