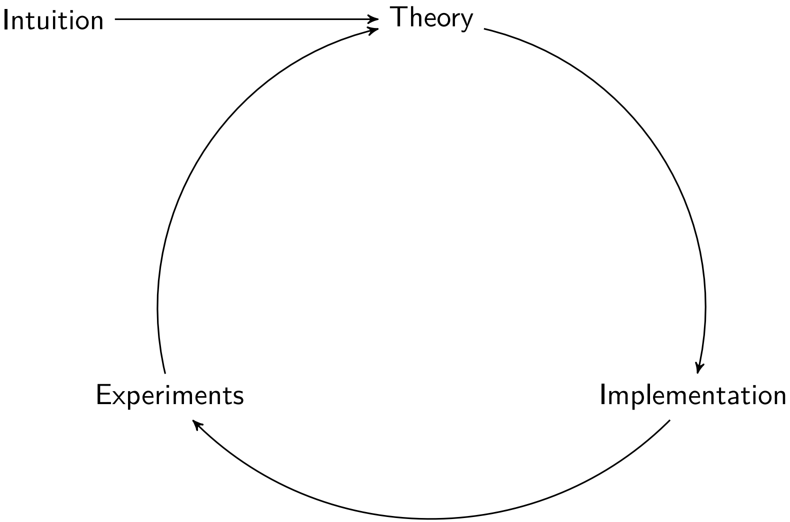
Bimorphism Machine Translation

Daniel Quernheim

April 10, 2017

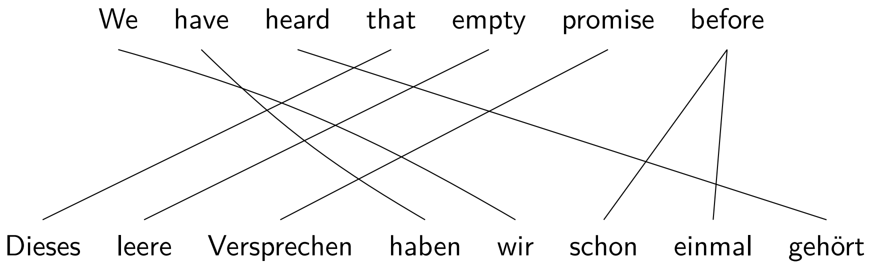
The Big Picture



Section 1

Intuition

Translation: Word by Word



Translation: Compositional

that $\widehat{=}$ Dieses | empty $\widehat{=}$ leere | promise $\widehat{=}$ Versprechen

---

heard $\widehat{=}$ gehört | that empty promise $\widehat{=}$ Dieses leere Versprechen

---

have $\widehat{=}$ haben | heard that empty promise $\widehat{=}$ Dieses leere Versprechen ... gehört | before $\widehat{=}$ schon einmal

---

We $\widehat{=}$ wir | have heard that empty promise before $\widehat{=}$ Dieses leere Versprechen haben ... schon einmal gehört

---

We have heard that empty promise before $\widehat{=}$ Dieses leere Versprechen haben wir schon einmal gehört

Instantiated inference rule

$$\frac{\text{that} \mathrel{\widehat{=}} \text{dieses} \quad \text{empty} \mathrel{\widehat{=}} \text{leere} \quad \text{promise} \mathrel{\widehat{=}} \text{Versprechen}}{\text{that empty promise} \mathrel{\widehat{=}} \text{Dieses leere Versprechen}}$$

Instantiated inference rule

$$\frac{\text{that} \mathrel{\widehat{=}} \text{dieses} \quad \text{empty} \mathrel{\widehat{=}} \text{leere} \quad \text{promise} \mathrel{\widehat{=}} \text{Versprechen}}{\text{that empty promise} \mathrel{\widehat{=}} \text{Dieses leere Versprechen}}$$

Abstract inference rule

$$\frac{x_1 \mathrel{\widehat{=}} y_1 \quad x_2 \mathrel{\widehat{=}} y_2 \quad x_3 \mathrel{\widehat{=}} y_3}{x_1 x_2 x_3 \mathrel{\widehat{=}} y_1 y_2 y_3}$$

Instantiated inference rule

$$\frac{\text{that} \mathrel{\widehat{=}} \text{dieses} \quad \text{empty} \mathrel{\widehat{=}} \text{leere} \quad \text{promise} \mathrel{\widehat{=}} \text{Versprechen}}{\text{that empty promise} \mathrel{\widehat{=}} \text{Dieses leere Versprechen}}$$

Abstract inference rule

$$\frac{x_1 \mathrel{\widehat{=}} y_1 \quad x_2 \mathrel{\widehat{=}} y_2 \quad x_3 \mathrel{\widehat{=}} y_3}{x_1 x_2 x_3 \mathrel{\widehat{=}} y_1 y_2 y_3}$$

Phrase labels

$$\frac{x_1[\text{DT}] \mathrel{\widehat{=}} y_1[\text{PDAT}] \quad x_2[\text{JJ}] \mathrel{\widehat{=}} y_2[\text{ADJA}] \quad x_3[\text{NN}] \mathrel{\widehat{=}} y_3[\text{NN}]}{x_1 x_2 x_3[\text{NP}] \mathrel{\widehat{=}} y_1 y_2 y_3[\text{NP-OA}]}$$

Inference Rules

Instantiated inference rule

$$\frac{\text{that} \mathrel{\widehat{=}} \text{dieses} \quad \text{empty} \mathrel{\widehat{=}} \text{leere} \quad \text{promise} \mathrel{\widehat{=}} \text{Versprechen}}{\text{that empty promise} \mathrel{\widehat{=}} \text{Dieses leere Versprechen}}$$

Abstract inference rule

$$\frac{x_1 \mathrel{\widehat{=}} y_1 \quad x_2 \mathrel{\widehat{=}} y_2 \quad x_3 \mathrel{\widehat{=}} y_3}{x_1 x_2 x_3 \mathrel{\widehat{=}} y_1 y_2 y_3}$$

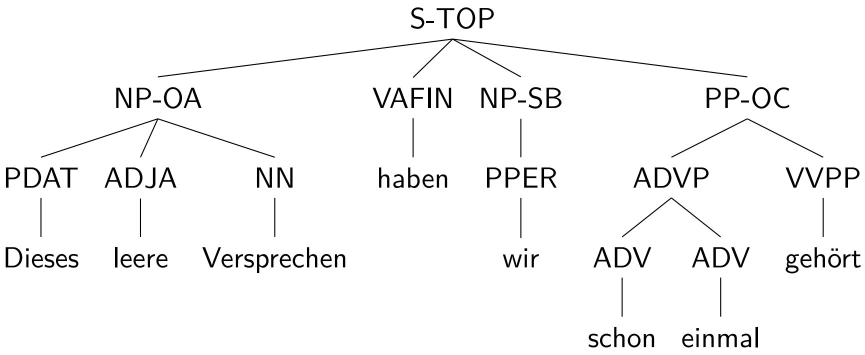
Phrase labels

$$\frac{\text{DT} \mathrel{\widehat{=}} \text{PDAT} \quad \text{JJ} \mathrel{\widehat{=}} \text{ADJA} \quad \text{NN} \mathrel{\widehat{=}} \text{NN}}{\text{NP} \mathrel{\widehat{=}} \text{NP-OA}}$$

$$\text{in}(x_1, x_2, x_3) = x_1 x_2 x_3$$

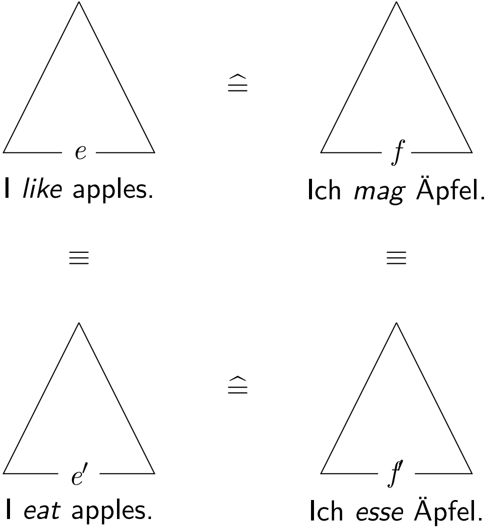
$$\text{out}(y_1, y_2, y_3) = y_1 y_2 y_3$$

Syntax Trees



Equivalence and Correspondence

$$\begin{array}{ccc} e = \text{'like'} & \widehat{=} & f = \text{'mag'} \\ \equiv & & \equiv \\ e' = \text{'eat'} & \widehat{=} & f' = \text{'esse'} \end{array} \quad \Longrightarrow$$



Assumptions

- inference rules have labels $\rightsquigarrow$ language-independent *ranked alphabet* Δ
- language-specific realization mappings $\rightsquigarrow$ 'in' and 'out' process trees over Δ

Derivation Trees

Assumptions

- inference rules have labels $\rightsquigarrow$ language-independent *ranked alphabet* Δ
- language-specific realization mappings $\rightsquigarrow$ 'in' and 'out' process trees over Δ

$$\dfrac{\dfrac{}{\mathsf{NP} \mathrel{\widehat{=}} \mathsf{NP}}\,(3) \qquad \dfrac{\dfrac{}{\mathsf{V} \mathrel{\widehat{=}} \mathsf{V}}\,(5) \qquad \dfrac{}{\mathsf{NP} \mathrel{\widehat{=}} \mathsf{NP}}\,(4)}{\mathsf{VP} \mathrel{\widehat{=}} \mathsf{VP}}\,(2)}{\mathsf{S} \mathrel{\widehat{=}} \mathsf{S}}\,(1)$$

in

out

I open the box. $\widehat{=}$ 私は箱を開けます。
(watashi wa hako wo akemasu)

Inference Rules on Strings

$$
\begin{aligned}
\mathsf{in}(1)(x_1, x_2) &= x_1 x_2 & \mathsf{out}(1)(x_1, x_2) &= x_1 x_2 \\
\mathsf{in}(2)(x_1, x_2) &= x_1 x_2 & \mathsf{out}(2)(x_1, x_2) &= x_2 x_1 \\
\mathsf{in}(3)() &= \mathsf{i} & \mathsf{out}(3)() &= \mathsf{watashi\ wa} \\
\mathsf{in}(4)() &= \mathsf{the\ box} & \mathsf{out}(4)() &= \mathsf{hako\ wo} \\
\mathsf{in}(5)() &= \mathsf{open} & \mathsf{out}(5)() &= \mathsf{akemasu}
\end{aligned}
$$

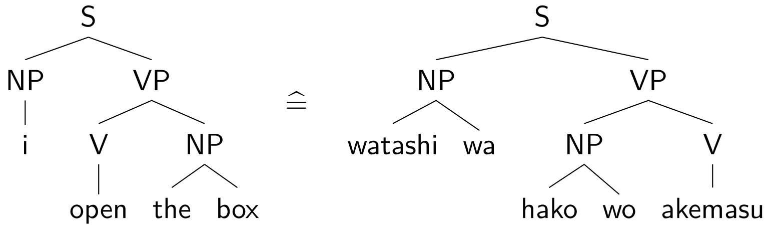
Derivation tree and string correspondence

$$
\dfrac{\dfrac{}{\mathsf{i} \mathrel{\widehat{=}} \mathsf{watashi\ wa}}\,(3) \qquad \dfrac{\dfrac{}{\mathsf{open} \mathrel{\widehat{=}} \mathsf{akemasu}}\,(5) \quad \dfrac{}{\mathsf{the\ box} \mathrel{\widehat{=}} \mathsf{hako\ wo}}\,(4)}{\mathsf{open\ the\ box} \mathrel{\widehat{=}} \mathsf{hako\ wo\ akemasu}}\,(2)}{\mathsf{i\ open\ the\ box} \mathrel{\widehat{=}} \mathsf{watashi\ wa\ hako\ wo\ akemasu}}\,(1)
$$

Inference Rules on Trees

$$\begin{aligned}
\mathsf{in}(1)(x_1, x_2) &= \mathsf{S}(x_1, x_2) & \mathsf{out}(1)(x_1, x_2) &= \mathsf{S}(x_1, x_2)\\
\mathsf{in}(2)(x_1, x_2) &= \mathsf{VP}(x_1, x_2) & \mathsf{out}(2)(x_1, x_2) &= \mathsf{VP}(x_2, x_1)\\
\mathsf{in}(3)() &= \mathsf{NP}(\mathsf{i}) & \mathsf{out}(3)() &= \mathsf{NP}(\mathsf{watashi},\ \mathsf{wa})\\
\mathsf{in}(4)() &= \mathsf{NP}(\mathsf{the},\ \mathsf{box}) & \mathsf{out}(4)() &= \mathsf{NP}(\mathsf{hako},\ \mathsf{wo})\\
\mathsf{in}(5)() &= \mathsf{V}(\mathsf{open}) & \mathsf{out}(5)() &= \mathsf{V}(\mathsf{akemasu})
\end{aligned}$$

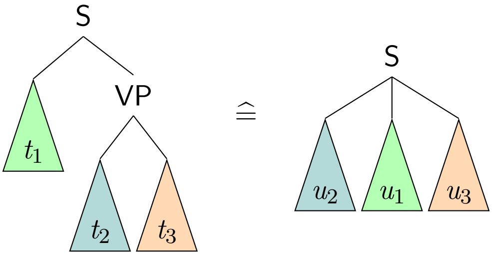
Tree correspondence



Linguistic Phenomena

Local rotations

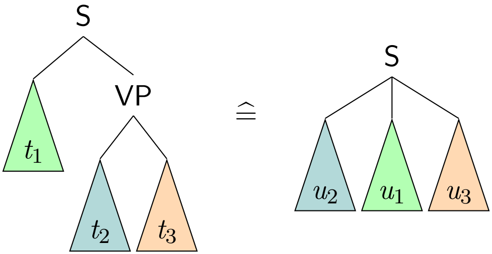
$$\frac{t_1 \mathrel{\widehat{=}} u_1 \qquad t_2 \mathrel{\widehat{=}} u_2 \qquad t_3 \mathrel{\widehat{=}} u_3}{}$$



Linguistic Phenomena

Local rotations

$$t_1 \mathrel{\widehat{=}} u_1 \quad t_2 \mathrel{\widehat{=}} u_2 \quad t_3 \mathrel{\widehat{=}} u_3$$

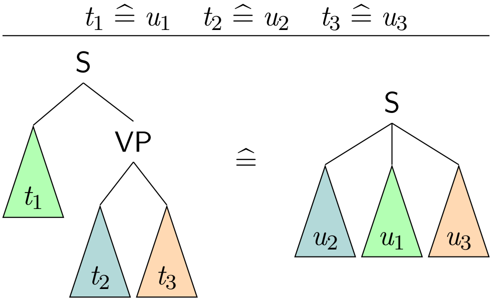


Discontiguities

$$\mathsf{VBD(went)} \mathrel{\widehat{=}} (\mathsf{VAFIN(ist)}, \mathsf{VVPP(gegangen)})$$

Linguistic Phenomena

Local rotations



Discontiguities

$$\mathsf{VBD}(\mathsf{went}) \mathrel{\widehat{=}} (\mathsf{VAFIN}(\mathsf{ist}), \qquad \mathsf{VVPP}(\mathsf{gegangen}))$$

Section 2

Theory

Notation

Δ	ranked alphabet of language-independent symbols
T_Δ	derivation trees over Δ

Notation

Δ	ranked alphabet of language-independent symbols
T_Δ	derivation trees over Δ
F	"foreign" or input (source) language
E	"English" or output (target) language

Notation

Δ ranked alphabet of language-independent symbols

T_Δ derivation trees over Δ

F "foreign" or input (source) language

E "English" or output (target) language

$\mathbf{F}$ surface strings of language F

$\mathbf{f}$ a string in $\mathbf{F}$

Notation

Δ	ranked alphabet of language-independent symbols
T_Δ	derivation trees over Δ
F	"foreign" or input (source) language
E	"English" or output (target) language
$\mathbf{F}$	surface strings of language F
$\mathbf{f}$	a string in $\mathbf{F}$
T_F	syntactic trees of language F
f	a tree in T_F

Notation

Δ	ranked alphabet of language-independent symbols
T_Δ	derivation trees over Δ
F	“foreign” or input (source) language
E	“English” or output (target) language
$\mathbf{F}$	surface strings of language F
$\mathbf{f}$	a string in $\mathbf{F}$
T_F	syntactic trees of language F
f	a tree in T_F
$\mathsf{yield}(f)$	string yield (frontier) of tree f

Weighted Regular Tree Languages

$$\frac{}{\mathsf{V} \mathrel{\widehat{=}} \mathsf{V}}\ (\alpha) \qquad \frac{}{\mathsf{NP} \mathrel{\widehat{=}} \mathsf{NP}}\ (\beta)$$

$$\frac{\mathsf{V} \mathrel{\widehat{=}} \mathsf{V} \quad \mathsf{NP} \mathrel{\widehat{=}} \mathsf{NP}}{\mathsf{VP} \mathrel{\widehat{=}} \mathsf{VP}}\ (\gamma)$$

- inference rules have
  - ***labels*** in $\Delta = \{\alpha, \beta, \gamma, \delta, \dots\}$

Weighted Regular Tree Languages

$$\frac{}{\mathsf{V} \mathrel{\widehat{=}} \mathsf{V}}\;(\alpha) \qquad \frac{}{\mathsf{NP} \mathrel{\widehat{=}} \mathsf{NP}}\;(\beta)$$

$$\frac{\mathsf{V} \mathrel{\widehat{=}} \mathsf{V} \quad \mathsf{NP} \mathrel{\widehat{=}} \mathsf{NP}}{\mathsf{VP} \mathrel{\widehat{=}} \mathsf{VP}}\;(\gamma)$$

$$\frac{\dfrac{}{\mathsf{V} \mathrel{\widehat{=}} \mathsf{V}} \quad \dfrac{}{\mathsf{NP} \mathrel{\widehat{=}} \mathsf{NP}}}{\mathsf{VP} \mathrel{\widehat{=}} \mathsf{VP}}$$

- inference rules have
  - ***labels*** in $\Delta = \{\alpha, \beta, \gamma, \delta, \dots\}$
  - typed premises and conclusions

Weighted Regular Tree Languages

$$\frac{}{\mathsf{V} \mathrel{\widehat{=}} \mathsf{V}}\ 0.6 \qquad \frac{}{\mathsf{NP} \mathrel{\widehat{=}} \mathsf{NP}}\ 0.4$$

$$\frac{\mathsf{V} \mathrel{\widehat{=}} \mathsf{V} \quad \mathsf{NP} \mathrel{\widehat{=}} \mathsf{NP}}{\mathsf{VP} \mathrel{\widehat{=}} \mathsf{VP}}\ 0.5$$

- inference rules have
  - ***labels*** in $\Delta = \{\alpha, \beta, \gamma, \delta, \dots\}$
  - typed premises and conclusions
  - ***weights*** in $\mathbb{R}$

Weighted Regular Tree Languages

$$\frac{}{\mathsf{V} \mathrel{\widehat{=}} \mathsf{V}}\ 0.6 \qquad \frac{}{\mathsf{NP} \mathrel{\widehat{=}} \mathsf{NP}}\ 0.4$$

$$\frac{\mathsf{V} \mathrel{\widehat{=}} \mathsf{V} \quad \mathsf{NP} \mathrel{\widehat{=}} \mathsf{NP}}{\mathsf{VP} \mathrel{\widehat{=}} \mathsf{VP}}\ 0.5$$

$$\frac{\dfrac{}{\mathsf{V} \mathrel{\widehat{=}} \mathsf{V}}\ 0.6 \quad \dfrac{}{\mathsf{NP} \mathrel{\widehat{=}} \mathsf{NP}}\ 0.4}{\mathsf{VP} \mathrel{\widehat{=}} \mathsf{VP}}\ 0.6 \cdot 0.4 \cdot 0.5 = 0.12$$

- inference rules have
  - ***labels*** in $\Delta = \{\alpha, \beta, \gamma, \delta, \dots\}$
  - typed premises and conclusions
  - ***weights*** in $\mathbb{R}$
- weight of proof tree
  $=$ product of rule weights
- ***final weight mapping*** for conclusions

Weighted Regular Tree Languages

$$\frac{}{\mathsf{V} \mathrel{\widehat{=}} \mathsf{V}}\ 0.6 \qquad \frac{}{\mathsf{NP} \mathrel{\widehat{=}} \mathsf{NP}}\ 0.4$$

$$\frac{\mathsf{V} \mathrel{\widehat{=}} \mathsf{V} \quad \mathsf{NP} \mathrel{\widehat{=}} \mathsf{NP}}{\mathsf{VP} \mathrel{\widehat{=}} \mathsf{VP}}\ 0.5$$

$$\frac{\dfrac{}{\mathsf{V} \mathrel{\widehat{=}} \mathsf{V}}\ 0.6 \quad \dfrac{}{\mathsf{NP} \mathrel{\widehat{=}} \mathsf{NP}}\ 0.4}{\mathsf{VP} \mathrel{\widehat{=}} \mathsf{VP}}\ 0.6 \cdot 0.4 \cdot 0.5 = 0.12$$

- inference rules have
  - ***labels*** in $\Delta = \{\alpha, \beta, \gamma, \delta, \dots\}$
  - typed premises and conclusions
  - ***weights*** in $\mathbb{R}$
- weight of proof tree
  $=$ product of rule weights
- ***final weight mapping*** for conclusions
- weight of derivation tree $t \in T_\Delta$
  $=$ sum of all proof trees for t

Weighted Regular Tree Languages

$$\frac{}{\mathsf{V} \mathrel{\widehat{=}} \mathsf{V}}\ 0.6 \qquad \frac{}{\mathsf{NP} \mathrel{\widehat{=}} \mathsf{NP}}\ 0.4$$

$$\frac{\mathsf{V} \mathrel{\widehat{=}} \mathsf{V} \quad \mathsf{NP} \mathrel{\widehat{=}} \mathsf{NP}}{\mathsf{VP} \mathrel{\widehat{=}} \mathsf{VP}}\ 0.5$$

$$\frac{\dfrac{}{\mathsf{V} \mathrel{\widehat{=}} \mathsf{V}}\ 0.6 \quad \dfrac{}{\mathsf{NP} \mathrel{\widehat{=}} \mathsf{NP}}\ 0.4}{\mathsf{VP} \mathrel{\widehat{=}} \mathsf{VP}}\ 0.6 \cdot 0.4 \cdot 0.5 = 0.12$$

- inference rules have
  - ***labels*** in $\Delta = \{\alpha, \beta, \gamma, \delta, \dots\}$
  - typed premises and conclusions
  - ***weights*** in $\mathbb{R}$
- weight of proof tree
  $=$ product of rule weights
- ***final weight mapping*** for conclusions
- weight of derivation tree $t \in T_\Delta$
  $=$ sum of all proof trees for t

$L \colon T_\Delta \to \mathbb{R}$ **is a *weighted regular tree language* (wRTL)**

Weighted Regular Tree Languages

$$\frac{}{\mathsf{V} \mathrel{\widehat{=}} \mathsf{V}}\ 0.6 \qquad \frac{}{\mathsf{NP} \mathrel{\widehat{=}} \mathsf{NP}}\ 0.4$$

$$\frac{\mathsf{V} \mathrel{\widehat{=}} \mathsf{V} \quad \mathsf{NP} \mathrel{\widehat{=}} \mathsf{NP}}{\mathsf{VP} \mathrel{\widehat{=}} \mathsf{VP}}\ 0.5$$

$$\frac{\dfrac{}{\mathsf{V} \mathrel{\widehat{=}} \mathsf{V}}\ 0.6 \quad \dfrac{}{\mathsf{NP} \mathrel{\widehat{=}} \mathsf{NP}}\ 0.4}{\mathsf{VP} \mathrel{\widehat{=}} \mathsf{VP}}\ 0.6 \cdot 0.4 \cdot 0.5 = 0.12$$

- inference rules have
  - ***labels*** in $\Delta = \{\alpha, \beta, \gamma, \delta, \dots\}$
  - typed premises and conclusions $\rightsquigarrow$ *states*
  - ***weights*** in $\mathbb{R}$
- weight of proof tree $\rightsquigarrow$ *run*
  $=$ product of rule weights
- ***final weight mapping*** for conclusions
- weight of derivation tree $t \in T_\Delta$
  $=$ sum of all proof trees for t

$L : T_\Delta \to \mathbb{R}$ **is a *weighted regular tree language* (wRTL)**

Homomorphism

$$\big(h_k : \Delta^{(k)} \to A^{(A^k)} \mid k \in \mathbb{N}\big) \qquad \text{extends to} \qquad \begin{aligned} & h : T_\Delta \to A \\ & h(t) = h_k(\delta)(h(t_1), \ldots, h(t_k)) \\ & \quad \text{for} \quad t = \delta(t_1, \ldots, t_k) \end{aligned}$$

Useful homomorphisms

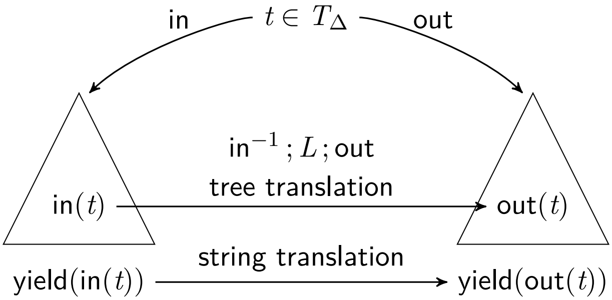
String concatenation $A = \mathbf{F}$

Tree homomorphisms $A = T_F$

Tree m-morphisms $A = (T_F)^m$

Bimorphism Machine Translation

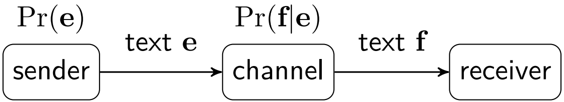
(tree) homomorphism	wRTL	(tree) homomorphism
in : $T_\Delta \to T_F$	$L : T_\Delta \to \mathbb{R}$	out : $T_\Delta \to T_E$



$(\text{in}, L, \text{out})$ **is a *weighted bimorphism***

History

Noisy Channel Model

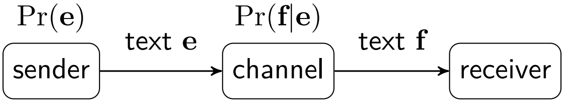


Probability space

$$(\Omega, 2^{\Omega}, \Pr) \text{ with } \Omega = \mathbf{F} \times \mathbf{E}$$

Random variables

$$\begin{array}{lll} S_{\mathbf{F}} : \Omega \to \mathbf{F} & S_{\mathbf{F}}(\mathbf{f}, \mathbf{e}) = \mathbf{f} & \Pr(\mathbf{f}) \text{ for } \Pr(S_{\mathbf{F}} = \mathbf{f}) \\ S_{\mathbf{E}} : \Omega \to \mathbf{E} & S_{\mathbf{E}}(\mathbf{f}, \mathbf{e}) = \mathbf{e} & \Pr(\mathbf{e}) \text{ for } \Pr(S_{\mathbf{E}} = \mathbf{e}) \end{array}$$



Search for the best translation

$$
\begin{aligned}
\hat{\mathbf{e}} &= \arg\max_{\mathbf{e}\in\mathbf{E}} \Pr(\mathbf{e}|\mathbf{f}) \\
&= \arg\max_{\mathbf{e}\in\mathbf{E}} \frac{\Pr(\mathbf{f}|\mathbf{e}) \cdot \Pr(\mathbf{e})}{\Pr(\mathbf{f})} \\
&= \arg\max_{\mathbf{e}\in\mathbf{E}} \left(\underbrace{\Pr(\mathbf{f}|\mathbf{e})}_{\text{translation model}} \cdot \underbrace{\Pr(\mathbf{e})}_{\text{language model}} \right)
\end{aligned}
$$

Feature functions

$$(\phi_i : \mathbf{F} \times \mathbf{E} \to [0,1] \mid i \in [m])$$

Linear combination

$$\begin{aligned}\Pr(\mathbf{e}|\mathbf{f}) &\propto \prod_{i=1}^{m} \phi_i(\mathbf{f},\mathbf{e})^{\lambda_i} \\ &= \exp\left(\sum_{i=1}^{m} \lambda_i \cdot \log \phi_i(\mathbf{f},\mathbf{e})\right)\end{aligned}$$

("log-linear model")

A Different Story (1)

Probability space

$$(\Omega, 2^{\Omega}, \Pr) \text{ with } \Omega = T_{\Delta}$$

Random variables

$$\begin{array}{lll}
T_{\mathbf{F}} : \Omega \to T_F & T_{\mathbf{F}}(t) = \mathsf{in}(t) & \Pr(f) \text{ for } \Pr(T_{\mathbf{F}} = f) \\
T_{\mathbf{E}} : \Omega \to T_E & T_{\mathbf{E}}(t) = \mathsf{out}(t) & \Pr(e) \text{ for } \Pr(T_{\mathbf{E}} = e) \\
S_{\mathbf{F}} : \Omega \to \mathbf{F} & S_{\mathbf{F}}(t) = \mathsf{yield}(\mathsf{in}(t)) & \Pr(\mathbf{f}) \text{ for } \Pr(S_{\mathbf{F}} = \mathbf{f}) \\
S_{\mathbf{E}} : \Omega \to \mathbf{E} & S_{\mathbf{E}}(t) = \mathsf{yield}(\mathsf{out}(t)) & \Pr(\mathbf{e}) \text{ for } \Pr(S_{\mathbf{E}} = \mathbf{e})
\end{array}$$

A Different Story (2)

$$\hat{\mathbf{e}} = \arg\max_{\mathbf{e} \in \mathbf{E}} \Pr(\mathbf{e}|\mathbf{f})$$

A Different Story (2)

$$\begin{aligned}\hat{\mathbf{e}} &= \arg\max_{\mathbf{e}\in\mathbf{E}} \Pr(\mathbf{e}|\mathbf{f}) \\ &= \arg\max_{\mathbf{e}\in\mathbf{E}} \frac{\Pr(\mathbf{f},\mathbf{e})}{\Pr(\mathbf{f})}\end{aligned}$$

A Different Story (2)

$$\begin{aligned}
\hat{\mathbf{e}} &= \arg\max_{\mathbf{e}\in\mathbf{E}} \Pr(\mathbf{e}|\mathbf{f}) \\
&= \arg\max_{\mathbf{e}\in\mathbf{E}} \frac{\Pr(\mathbf{f},\mathbf{e})}{\Pr(\mathbf{f})} \\
&= \arg\max_{\mathbf{e}\in\mathbf{E}} \Pr(\mathbf{f},\mathbf{e})
\end{aligned}$$

A Different Story (2)

$$
\begin{aligned}
\hat{\mathbf{e}} &= \arg\max_{\mathbf{e} \in \mathbf{E}} \Pr(\mathbf{e}|\mathbf{f}) \\
&= \arg\max_{\mathbf{e} \in \mathbf{E}} \frac{\Pr(\mathbf{f}, \mathbf{e})}{\Pr(\mathbf{f})} \\
&= \arg\max_{\mathbf{e} \in \mathbf{E}} \Pr(\mathbf{f}, \mathbf{e}) \\
&= \arg\max_{\mathbf{e} \in \mathbf{E}} \sum_{t \in T_{\Delta}} \Pr(\mathbf{f}, t, \mathbf{e})
\end{aligned}
$$

A Different Story (2)

$$
\begin{aligned}
\hat{\mathbf{e}} &= \arg\max_{\mathbf{e}\in\mathbf{E}} \Pr(\mathbf{e}|\mathbf{f}) \\
&= \arg\max_{\mathbf{e}\in\mathbf{E}} \frac{\Pr(\mathbf{f},\mathbf{e})}{\Pr(\mathbf{f})} \\
&= \arg\max_{\mathbf{e}\in\mathbf{E}} \Pr(\mathbf{f},\mathbf{e}) \\
&= \arg\max_{\mathbf{e}\in\mathbf{E}} \sum_{t\in T_\Delta} \Pr(\mathbf{f},t,\mathbf{e}) \\
&= \arg\max_{\mathbf{e}\in\mathbf{E}} \sum_{\substack{\text{yield}(\text{in}(t))=\mathbf{f}\\ \text{yield}(\text{out}(t))=\mathbf{e}}} \Pr(t)
\end{aligned}
$$

because $\Pr(\mathbf{f},t,\mathbf{e}) = 0$
if $\text{yield}(\text{in}(t)) \neq \mathbf{f}$
or $\text{yield}(\text{out}(t)) \neq \mathbf{e}$

A Different Story (3)

$$\hat{\mathbf{e}} = \arg\max_{\mathbf{e} \in \mathbf{E}} \sum_{\substack{\mathsf{yield}(\mathsf{in}(t)) = \mathbf{f} \\ \mathsf{yield}(\mathsf{out}(t)) = \mathbf{e}}} \Pr(t)$$

A Different Story (3)

$$\hat{\mathbf{e}} = \arg\max_{\mathbf{e} \in \mathbf{E}} \sum_{\substack{\mathsf{yield}(\mathsf{in}(t))=\mathbf{f} \\ \mathsf{yield}(\mathsf{out}(t))=\mathbf{e}}} \Pr(t)$$

$$= \arg\max_{\mathbf{e} \in \mathbf{E}} \sum_{\substack{\mathsf{yield}(\mathsf{in}(t))=\mathbf{f} \\ \mathsf{yield}(\mathsf{out}(t))=\mathbf{e}}} \Pr(t)^{\lambda_1} \cdot \Pr(t)^{\lambda_2} \cdot \Pr(t)^{\lambda_3}$$

with $\lambda_1, \lambda_2, \lambda_3 \in [0, 1]$
such that $\lambda_1 + \lambda_2 + \lambda_3 = 1$

Factored Model (1)

Decompositions

$$\begin{aligned}
\Pr(t) &= \Pr(t, f) \\
&= \Pr(f) \cdot \Pr(t|f) \\
&= \Pr(f, \mathbf{f}) \cdot \Pr(t|f)
\end{aligned}$$

Factored Model (1)

Decompositions

$$\begin{aligned}\Pr(t) &= \Pr(t, f) \\ &= \Pr(f) \cdot \Pr(t|f) \\ &= \Pr(f, \mathbf{f}) \cdot \Pr(t|f)\end{aligned}$$

$$\begin{aligned}\Pr(t) &= \Pr(t, e) \\ &= \Pr(t|e) \cdot \Pr(e) \\ &= \Pr(t|e) \cdot \Pr(e, \mathbf{e}) \\ &= \Pr(t|e) \cdot \frac{\Pr(e, \mathbf{e})}{\Pr(\mathbf{e})} \cdot \Pr(\mathbf{e})\end{aligned}$$

Factored Model (1)

Decompositions

$$\begin{aligned}\Pr(t) &= \Pr(t, f) \\ &= \Pr(f) \cdot \Pr(t|f) \\ &= \Pr(f, \mathbf{f}) \cdot \Pr(t|f)\end{aligned}$$

$$\begin{aligned}\Pr(t) &= \Pr(t, e) \\ &= \Pr(t|e) \cdot \Pr(e) \\ &= \Pr(t|e) \cdot \Pr(e, \mathbf{e}) \\ &= \Pr(t|e) \cdot \frac{\Pr(e, \mathbf{e})}{\Pr(\mathbf{e})} \cdot \Pr(\mathbf{e})\end{aligned}$$

Factored Model

$$\begin{aligned}\Pr(t) &= \Pr(t)^{\lambda_1} \cdot \Pr(t)^{\lambda_2} \cdot \Pr(t)^{\lambda_3} \\ &= \big(\Pr(f, \mathbf{f}) \cdot \Pr(t|f)\big)^{\lambda_1} \cdot \Pr(t)^{\lambda_2} \cdot \left(\Pr(t|e) \cdot \frac{\Pr(e, \mathbf{e})}{\Pr(\mathbf{e})} \cdot \Pr(\mathbf{e})\right)^{\lambda_3}\end{aligned}$$

Factored Model (2)

$$\Pr(t) = \Big(\underbrace{\Pr(f, \mathbf{f})}_{\text{parser}} \cdot \underbrace{\Pr(t|f)}_{\text{forward}}\Big)^{\lambda_1} \cdot \underbrace{\Pr(t)}_{\text{symm.}}{}^{\lambda_2} \cdot \left(\underbrace{\Pr(t|e)}_{\text{backward}} \cdot \underbrace{\frac{\Pr(e, \mathbf{e})}{\Pr(\mathbf{e})}}_{\text{synLM}} \cdot \underbrace{\Pr(\mathbf{e})}_{\text{LM}}\right)^{\lambda_3}$$

Factored Model (2)

$$\Pr(t) = \Big(\underbrace{\Pr(f, \mathbf{f})}_{\text{parser}} \cdot \underbrace{\Pr(t|f)}_{\text{forward}}\Big)^{\lambda_1} \cdot \underbrace{\Pr(t)}_{\text{symm.}}{}^{\lambda_2} \cdot \left(\underbrace{\Pr(t|e)}_{\text{backward}} \cdot \underbrace{\frac{\Pr(e, \mathbf{e})}{\Pr(\mathbf{e})}}_{\text{synLM}} \cdot \underbrace{\Pr(\mathbf{e})}_{\text{LM}}\right)^{\lambda_3}$$

Parser $\Pr(f, \mathbf{f})$

Factored Model (2)

$$\Pr(t) = \Big(\underbrace{\Pr(f, \mathbf{f})}_{\text{parser}} \cdot \underbrace{\Pr(t|f)}_{\text{forward}}\Big)^{\lambda_1} \cdot \underbrace{\Pr(t)}_{\text{symm.}}{}^{\lambda_2} \cdot \left(\underbrace{\Pr(t|e)}_{\text{backward}} \cdot \underbrace{\frac{\Pr(e, \mathbf{e})}{\Pr(\mathbf{e})}}_{\text{synLM}} \cdot \underbrace{\Pr(\mathbf{e})}_{\text{LM}}\right)^{\lambda_3}$$

Parser $\Pr(f, \mathbf{f})$

Forward translation probability $\Pr(t|f)$

Symmetric translation probability $\Pr(t)$

Backward translation probability $\Pr(t|e)$

Factored Model (2)

$$\Pr(t) = \Big(\underbrace{\Pr(f, \mathbf{f})}_{\text{parser}} \cdot \underbrace{\Pr(t|f)}_{\text{forward}}\Big)^{\lambda_1} \cdot \underbrace{\Pr(t)}_{\text{symm.}}^{\lambda_2} \cdot \left(\underbrace{\Pr(t|e)}_{\text{backward}} \cdot \underbrace{\frac{\Pr(e, \mathbf{e})}{\Pr(\mathbf{e})}}_{\text{synLM}} \cdot \underbrace{\Pr(\mathbf{e})}_{\text{LM}}\right)^{\lambda_3}$$

Parser $\Pr(f, \mathbf{f})$

Forward translation probability $\Pr(t|f)$

Symmetric translation probability $\Pr(t)$

Backward translation probability $\Pr(t|e)$

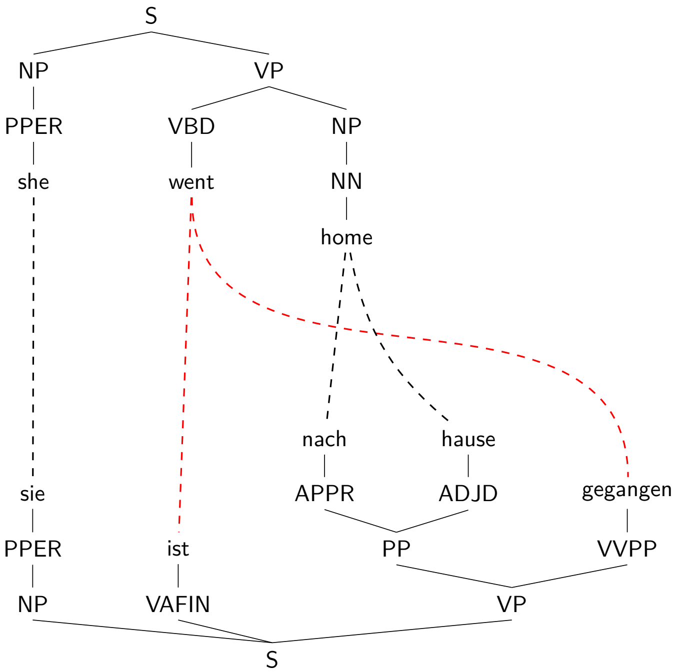
Syntactic language model $\Pr(e, \mathbf{e}) \cdot \Pr(\mathbf{e})^{-1} = \Pr(e|\mathbf{e})$

Language model $\Pr(\mathbf{e})$

Section 3

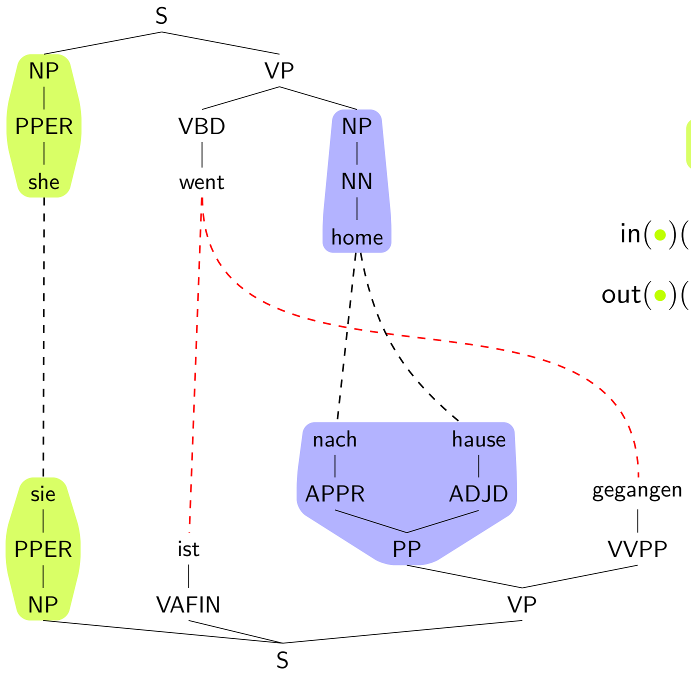
Intermission: Rule Extraction

STSG Rules



S
NP
VP
PPER
VBD
NP
she
went
NN
home
nach
hause
sie
APPR
ADJD
gegangen
PPER
ist
PP
VVPP
NP
VAFIN
VP
S

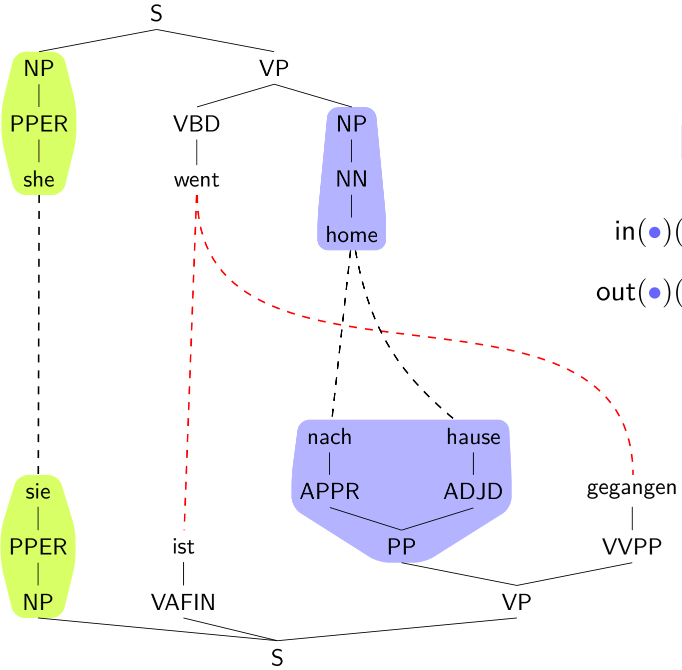
STSG Rules



$$\frac{}{\text{NP} \mathrel{\widehat{=}} \text{NP}}$$

$$\text{in}(\bullet)() = \text{NP}(\text{PPER}(\text{she}))$$

$$\text{out}(\bullet)() = \text{NP}(\text{PPER}(\text{sie}))$$

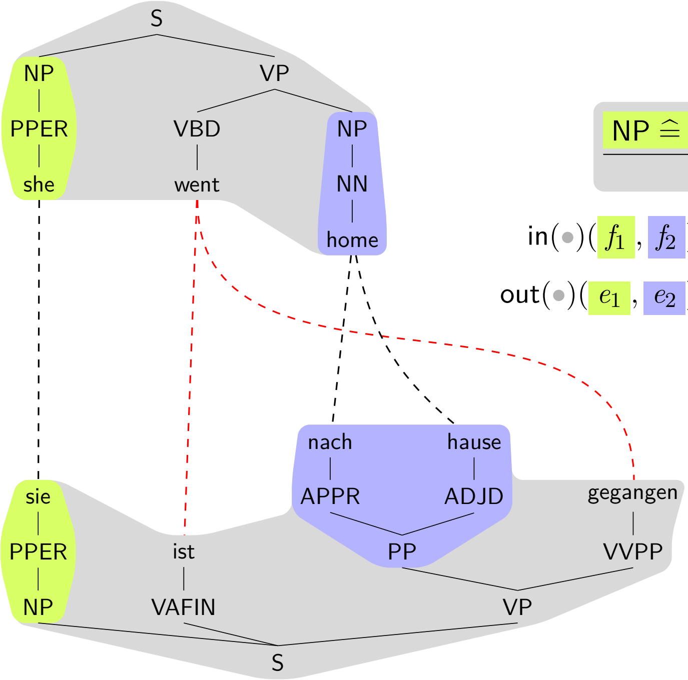


STSG Rules

$$\overline{\mathrm{NP} \mathrel{\widehat{=}} \mathrm{PP}}$$

$$\mathrm{in}(\bullet)() = \mathrm{NP}(\mathrm{NN}(\mathrm{home}))$$

$$\mathrm{out}(\bullet)() = \mathrm{PP}(\mathrm{APPR}(\mathrm{nach}), \mathrm{ADJD}(\mathrm{hause}))$$



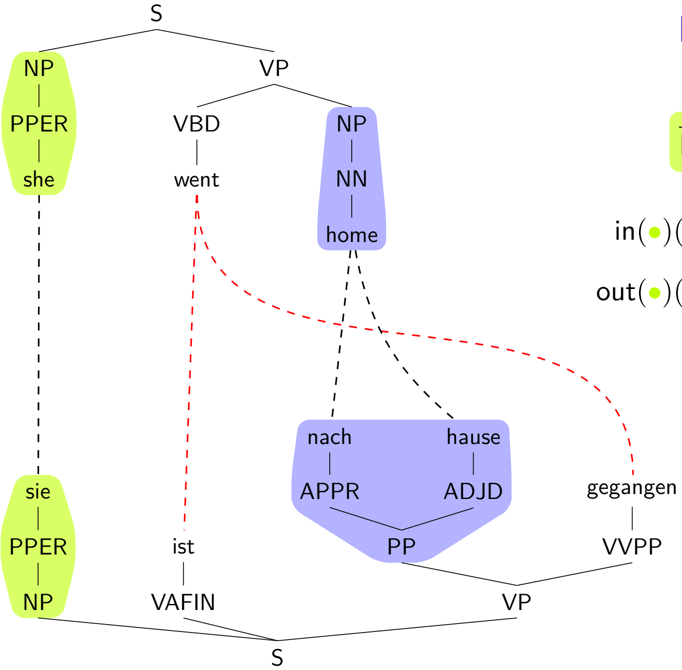
STSG Rules

$$\frac{\mathrm{NP} \mathrel{\hat{=}} \mathrm{NP} \qquad \mathrm{NP} \mathrel{\hat{=}} \mathrm{PP}}{\mathrm{S} \mathrel{\hat{=}} \mathrm{S}}$$

$$\mathrm{in}(\bullet)(f_1, f_2) = \mathrm{S}(f_1, \mathrm{VP}(\mathrm{VBD}(\mathrm{went}), f_2))$$

$$\mathrm{out}(\bullet)(e_1, e_2) = \mathrm{S}(e_1, \mathrm{VAFIN}(\mathrm{ist}), \mathrm{VP}(e_2, \mathrm{VVPP}(\mathrm{gegangen})))$$

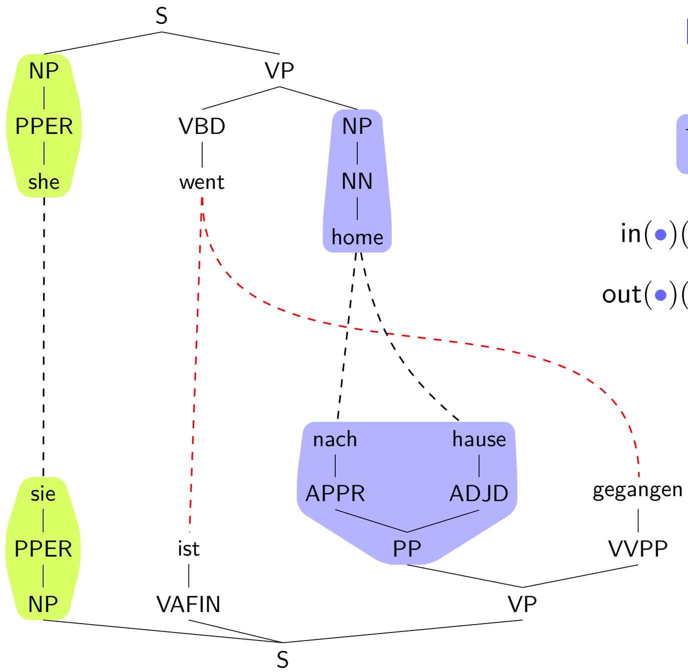
In-XMBOT Rules



$$\overline{}\;\; NP \mathrel{\widehat{=}} (NP)$$

$$\mathrm{in}(\bullet)() = NP(PPER(she))$$

$$\mathrm{out}(\bullet)() = (NP(PPER(sie)))$$

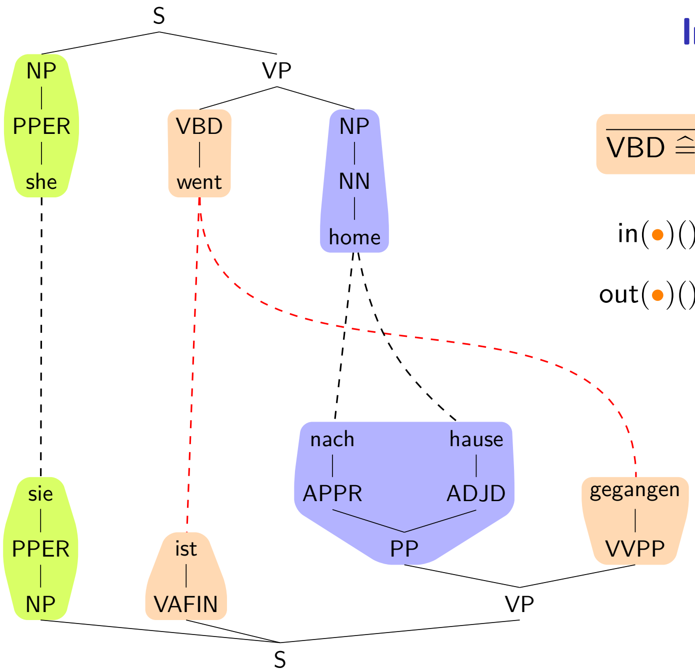


In-XMBOT Rules

$$\overline{\mathrm{NP} \mathrel{\widehat{=}} (\mathrm{PP})}$$

$$\mathrm{in}(\bullet)() = \mathrm{NP}(\mathrm{NN}(\mathrm{home}))$$

$$\mathrm{out}(\bullet)() = (\mathrm{PP}(\mathrm{APPR}(\mathrm{nach}), \mathrm{ADJD}(\mathrm{hause})))$$



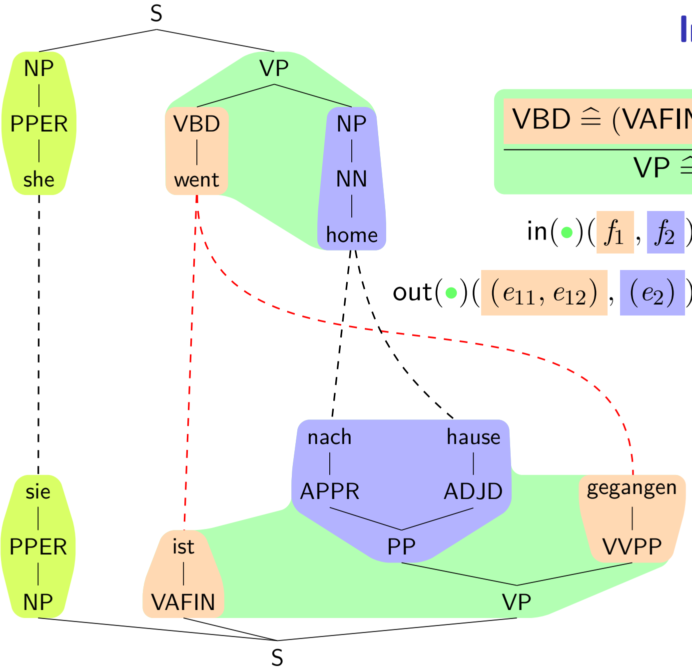
In-XMBOT Rules

$$\overline{\text{VBD} \mathrel{\widehat{=}} (\text{VAFIN}, \text{VVPP})}$$

$$\text{in}(\bullet)() = \text{VBD}(\text{went})$$

$$\text{out}(\bullet)() = (\text{VAFIN}(\text{ist}), \text{VVPP}(\text{gegangen}))$$

In-XMBOT Rules

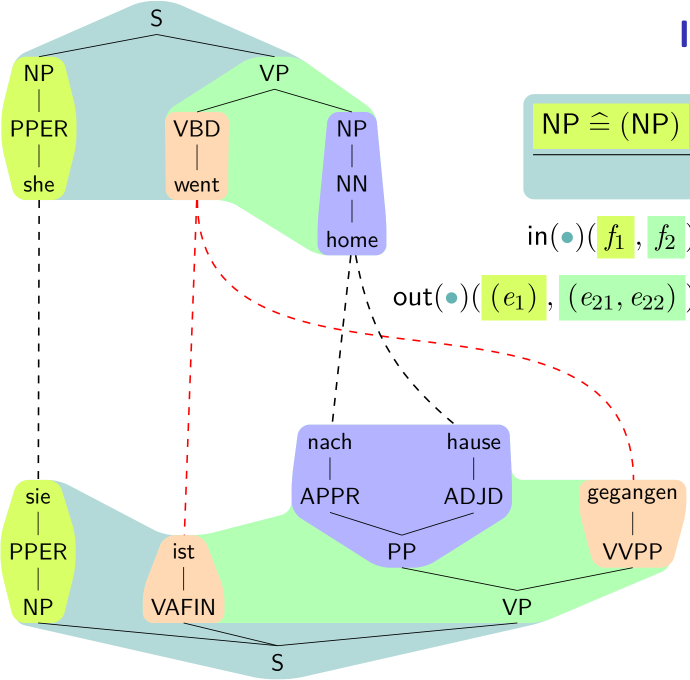


$$\frac{\mathrm{VBD} \mathrel{\widehat{=}} (\mathrm{VAFIN}, \mathrm{VVPP}) \qquad \mathrm{NP} \mathrel{\widehat{=}} (\mathrm{PP})}{\mathrm{VP} \mathrel{\widehat{=}} (\mathrm{VAFIN}, \mathrm{VP})}$$

$$\mathrm{in}(\bullet)(f_1, f_2) = \mathrm{VP}(f_1, f_2)$$

$$\mathrm{out}(\bullet)((e_{11}, e_{12}), (e_2)) = (e_{11}, \mathrm{VP\text{-}OC/pp}(e_2, e_{12}))$$

In-XMBOT Rules



$$\frac{\mathrm{NP} \mathrel{\widehat{=}} (\mathrm{NP}) \qquad \mathrm{VP} \mathrel{\widehat{=}} (\mathrm{VAFIN}, \mathrm{VP})}{\mathrm{S} \mathrel{\widehat{=}} (\mathrm{S})}$$

$$\mathrm{in}(\bullet)(f_1, f_2) = \mathrm{S}(f_1, f_2)$$

$$\mathrm{out}(\bullet)\big((e_1), (e_{21}, e_{22})\big) = \big(\mathrm{S}(e_1, e_{21}, e_{22})\big)$$

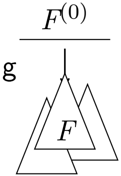
Section 4

Implementation

Decoding pipeline

$$\frac{F^{(0)}}{} \quad \text{string} \in \mathbf{F}$$

Decoding pipeline

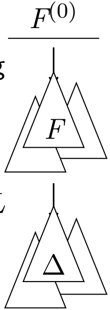


string parsing

string $\in \mathbf{F}$

parse forest : $T_F \to \mathbb{R}$

Decoding pipeline



string $\in \mathbf{F}$

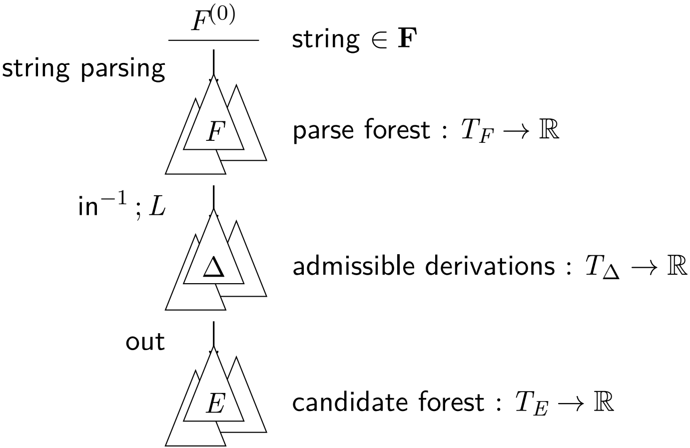
string parsing

parse forest : $T_F \to \mathbb{R}$

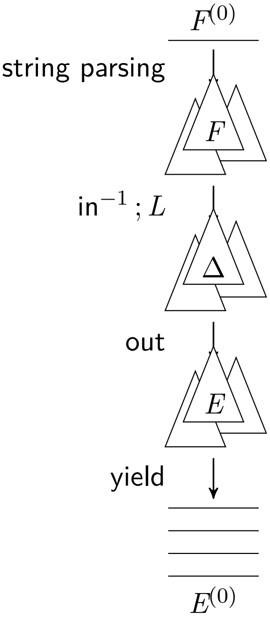
in^{-1} ; L

admissible derivations : $T_\Delta \to \mathbb{R}$

Decoding pipeline



Decoding pipeline



string $\in \mathbf{F}$

parse forest : $T_F \to \mathbb{R}$

admissible derivations : $T_\Delta \to \mathbb{R}$

candidate forest : $T_E \to \mathbb{R}$

candidate translations : $\mathbf{E} \to \mathbb{R}$

Approximation

of best translation $\hat{\mathbf{e}}$ by $\text{yield}(\text{out}(\hat{t}))$:

Viterbi Derivation

Approximation

of best translation $\hat{\mathbf{e}}$ by $\mathsf{yield}(\mathsf{out}(\hat{t}))$:

$$\hat{t} = \operatorname*{arg\,max}_{t \in \mathsf{in}^{-1}(\mathsf{yield}^{-1}(\mathbf{f}))} L(t)$$

Approximation

of best translation $\hat{\mathbf{e}}$ by $\mathsf{yield}(\mathsf{out}(\hat{t}))$:

$$\begin{aligned}\hat{t} &= \operatorname*{arg\,max}_{t \in \mathsf{in}^{-1}(\mathsf{yield}^{-1}(\mathbf{f}))} L(t) \\ &= \operatorname*{arg\,max}_{t \in \mathsf{in}^{-1}(\mathsf{yield}^{-1}(\mathbf{f}))} (\mathcal{F} \cdot \mathcal{T} \cdot \mathcal{E})(t)\end{aligned}$$

Approximation

of best translation $\hat{\mathbf{e}}$ by $\mathsf{yield}(\mathsf{out}(\hat{t}))$:

$$\begin{aligned}\hat{t} &= \operatorname*{arg\,max}_{t \in \mathsf{in}^{-1}(\mathsf{yield}^{-1}(\mathbf{f}))} L(t) \\ &= \operatorname*{arg\,max}_{t \in \mathsf{in}^{-1}(\mathsf{yield}^{-1}(\mathbf{f}))} (\mathcal{F} \cdot \mathcal{T} \cdot \mathcal{E})(t) \\ &= \operatorname*{arg\,max} \left(\underbrace{\left(\underbrace{\left(\underbrace{\mathsf{in}^{-1}(\mathsf{yield}^{-1}(\mathbf{f})) \cdot \mathcal{F}}_{\text{input}} \right) \cdot \mathcal{T}}_{\text{input + translation}} \right) \cdot \mathcal{E}}_{\text{input + translation + output}} \right)\end{aligned}$$

Probabilistic context-free parser

$P_F : T_F \to \mathbb{R}$ weighted regular tree language

Probabilistic context-free parser

$$P_F : T_F \to \mathbb{R} \qquad \text{weighted regular tree language}$$

Parser as input model

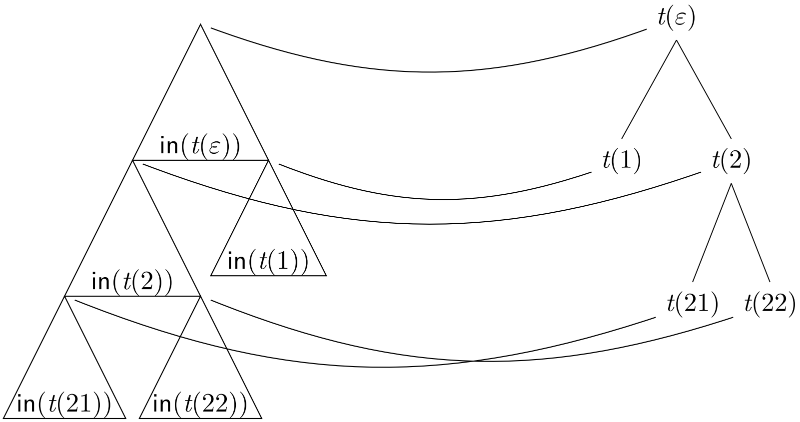
$$\begin{aligned}
\mathsf{in}^{-1}(\underbrace{\mathsf{yield}^{-1}(\mathbf{f}) \cdot P_F}_{\text{parse forest}}) &= \mathsf{in}^{-1}(\mathsf{yield}^{-1}(\mathbf{f})) \cdot \mathsf{in}^{-1}(P_F) \\
&= \mathsf{in}^{-1}(\mathsf{yield}^{-1}(\mathbf{f})) \cdot (\mathsf{in}\,;P_F) \\
&= \mathsf{in}^{-1}(\mathsf{yield}^{-1}(\mathbf{f})) \cdot \mathcal{F}
\end{aligned}$$

Product construction

$$\left(\mathsf{in}^{-1}(\mathsf{yield}^{-1}(\mathbf{f}) \cdot P_F) \cdot \mathfrak{T}\right) : T_\Delta \to \mathbb{R}$$

Product construction

$$\big(\mathsf{in}^{-1}(\mathsf{yield}^{-1}(\mathbf{f}) \cdot P_F) \cdot \mathcal{T}\big) : T_\Delta \to \mathbb{R}$$



Product Construction

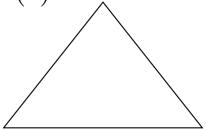
$\mathcal{T} : T_\Delta \to \mathbb{R}$

$$\frac{p_1 \quad \ldots \quad p_k}{p}$$

inference rule
with label δ
and weight a

$\mathsf{in} : T_\Delta \to T_F$

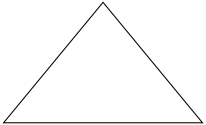
$\mathsf{in}(\delta) =$



$x_{\pi(1)}, \ldots, x_{\pi(k)}$

$\big(\mathsf{yield}^{-1}(\mathbf{f}) \cdot P_F\big) : T_F \to \mathbb{R}$

q



$q_{\pi(1)}, \ldots, q_{\pi(k)}$

proof tree for $\mathsf{in}(\delta)$
with weight w

Product Construction

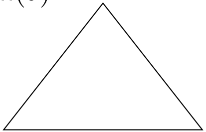
$\mathcal{T} : T_\Delta \to \mathbb{R}$

$$\frac{p_1 \quad \cdots \quad p_k}{p}$$

inference rule
with label δ
and weight a

$\mathsf{in} : T_\Delta \to T_F$

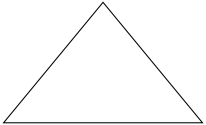
$\mathsf{in}(\delta) =$



$x_{\pi(1)}, \ldots, x_{\pi(k)}$

$\big(\mathsf{yield}^{-1}(\mathbf{f}) \cdot P_F\big) : T_F \to \mathbb{R}$

q



$q_{\pi(1)}, \ldots, q_{\pi(k)}$

proof tree for $\mathsf{in}(\delta)$
with weight w

New inference rule

$$\frac{\big(p_1, q_{\pi(1)}\big) \quad \cdots \quad \big(p_k, q_{\pi(k)}\big)}{(p, q)}$$ with label δ and weight $w \cdot a$

Lazy Feature Scoring

Local scoring (symbol scoring)

$$w_i : \Delta \to \mathbb{R} \qquad \text{extends to} \qquad \phi_i : T_\Delta \to \mathbb{R}$$

$$\phi_i(t) = \prod_{p \in \mathsf{pos}(t)} w_i(t(p))$$

Local scoring (symbol scoring)

$$w_i : \Delta \to \mathbb{R} \qquad \text{extends to} \qquad \phi_i : T_\Delta \to \mathbb{R}$$

$$\phi_i(t) = \prod_{p \in \mathsf{pos}(t)} w_i(t(p))$$

$$\mathcal{T} = \mathcal{T}' \cdot (\phi_1 \cdots \phi_j)^{\lambda_1} \cdot (\phi_{j+1} \cdots \phi_k)^{\lambda_2} \cdot (\phi_{k+1} \cdots \phi_n)^{\lambda_3}$$

with $\mathcal{T}'$ unweighted

Lazy Feature Scoring

Local scoring (symbol scoring)

$$w_i : \Delta \to \mathbb{R} \qquad \text{extends to} \qquad \phi_i : T_\Delta \to \mathbb{R}$$

$$\phi_i(t) = \prod_{p \in \mathrm{pos}(t)} w_i(t(p))$$

$$\mathcal{T} = \mathcal{T}' \cdot (\phi_1 \cdots \phi_j)^{\lambda_1} \cdot (\phi_{j+1} \cdots \phi_k)^{\lambda_2} \cdot (\phi_{k+1} \cdots \phi_n)^{\lambda_3}$$

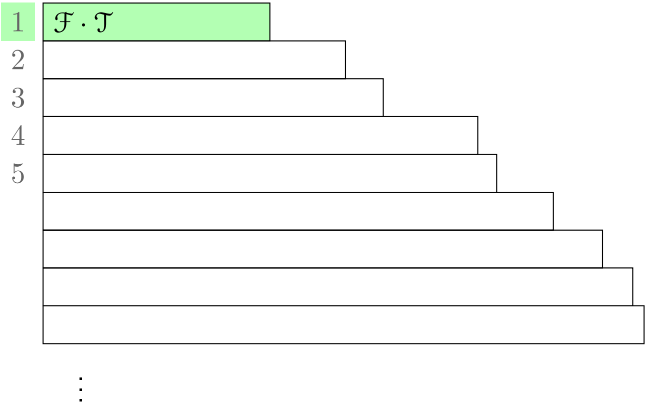
with $\mathcal{T}'$ unweighted

Lazy scoring

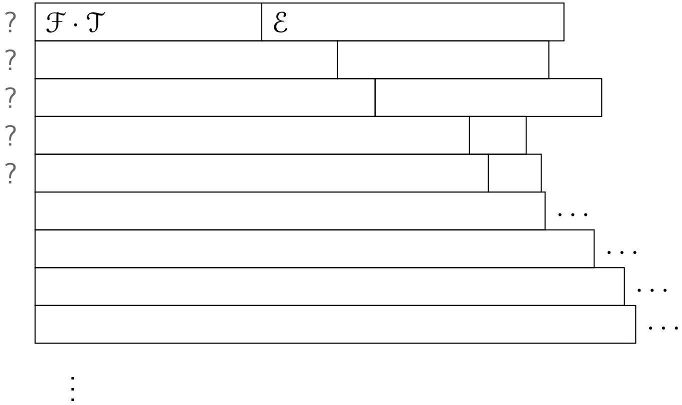
$$\Big(\mathrm{in}^{-1}(\mathrm{yield}^{-1}(\mathbf{f}) \cdot P_F) \cdot \mathcal{T}'\Big) \cdot (\phi_1 \cdots \phi_j)^{\lambda_1} \cdot (\phi_{j+1} \cdots \phi_k)^{\lambda_2} \cdot (\phi_{k+1} \cdots \phi_n)^{\lambda_3}$$

Exact Rescoring

Incorporation of an external language model



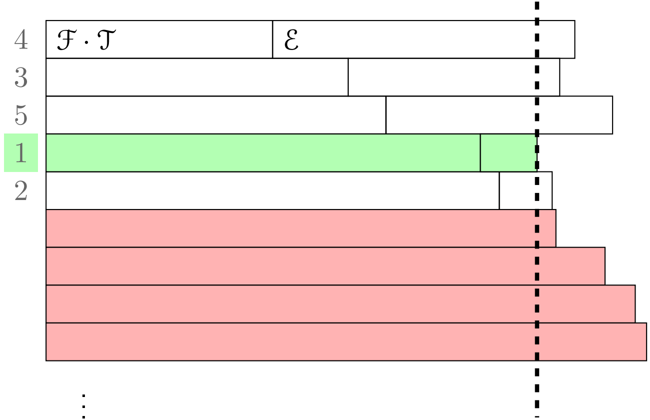
step 1 incremental construction of k-best list



step 2 rescoring of k candidates

Exact Rescoring

Incorporation of an external language model



step 3 monotonic feature functions allow safe pruning for large enough k

Section 5

Experiments

Theory vs. Engineering

Factored model

$$\Pr(t) = \Big(\underbrace{\Pr(f, \mathbf{f})}_{\text{parser}} \cdot \underbrace{\Pr(t|f)}_{\text{forward}}\Big)^{\lambda_1} \cdot \underbrace{\Pr(t)}_{\text{symm.}}{}^{\lambda_2} \cdot \left(\underbrace{\Pr(t|e)}_{\text{backward}} \cdot \underbrace{\frac{\Pr(e, \mathbf{e})}{\Pr(\mathbf{e})}}_{\text{synLM}} \cdot \underbrace{\Pr(\mathbf{e})}_{\text{LM}}\right)^{\lambda_3}$$

Maximum entropy model

$$\Pr(t) \approx \Pr(f, \mathbf{f})^{\mu_1} \cdot \Pr(t|f)^{\mu_2} \cdot \Pr(t)^{\mu_3} \cdot \Pr(t|e)^{\mu_4} \cdot \left(\frac{\Pr(e, \mathbf{e})}{\Pr(\mathbf{e})}\right)^{\mu_5} \cdot \Pr(\mathbf{e})^{\mu_6}$$

Theory vs. Engineering

Factored model

$$\Pr(t) = \Big(\underbrace{\Pr(f,\mathbf{f})}_{\text{parser}} \cdot \underbrace{\Pr(t|f)}_{\text{forward}}\Big)^{\lambda_1} \cdot \underbrace{\Pr(t)}_{\text{symm.}}{}^{\lambda_2} \cdot \left(\underbrace{\Pr(t|e)}_{\text{backward}} \cdot \underbrace{\frac{\Pr(e,\mathbf{e})}{\Pr(\mathbf{e})}}_{\text{synLM}} \cdot \underbrace{\Pr(\mathbf{e})}_{\text{LM}}\right)^{\lambda_3}$$

Maximum entropy model

$$\Pr(t) \approx \Pr(f,\mathbf{f})^{\mu_1} \cdot \Pr(t|f)^{\mu_2} \cdot \Pr(t)^{\mu_3} \cdot \Pr(t|e)^{\mu_4} \cdot \left(\frac{\Pr(e,\mathbf{e})}{\Pr(\mathbf{e})}\right)^{\mu_5} \cdot \Pr(\mathbf{e})^{\mu_6}$$

Conclusion

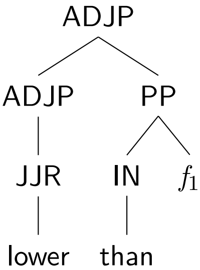
Maximum Entropy performs significantly better by ~ 1 Bleu point

Search Errors

in a state-of-the-art MT system

Travatar tree-to-string system

$\frac{\text{PP}}{\text{ADJP}}\ (\delta)$ $\quad$ $\text{in}(\delta)(f_1) =$



$\text{out}(\delta)(e_1) = \text{niedriger als } e_1$

Diagnostics

Pruning branches of search space are cut off

Diagnostics

Pruning branches of search space are cut off

Search error model-optimal translation lost in pruning

Diagnostics

Pruning branches of search space are cut off

Search error model-optimal translation lost in pruning

Model error model-optimal translation not a good translation

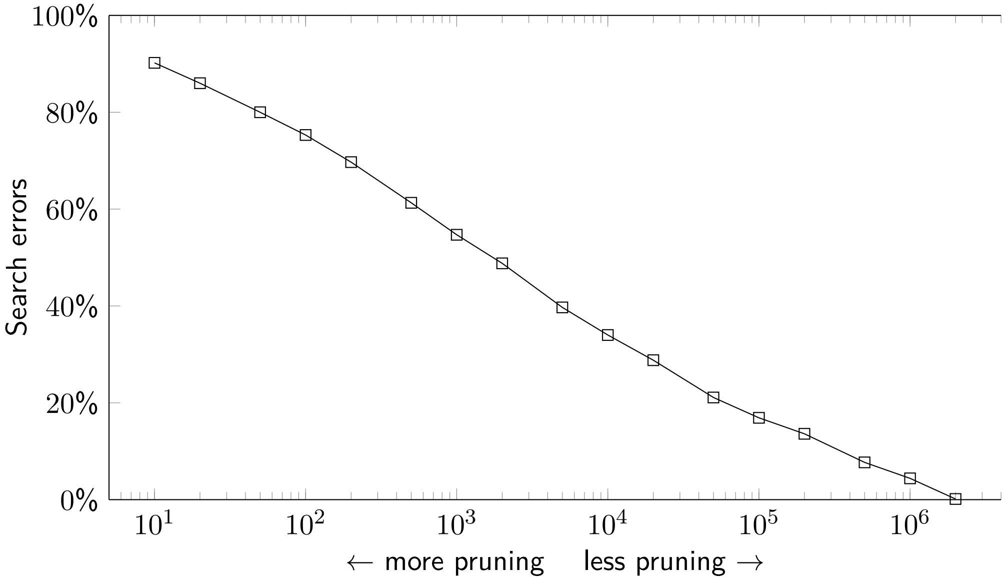
Pruning branches of search space are cut off

Search error model-optimal translation lost in pruning

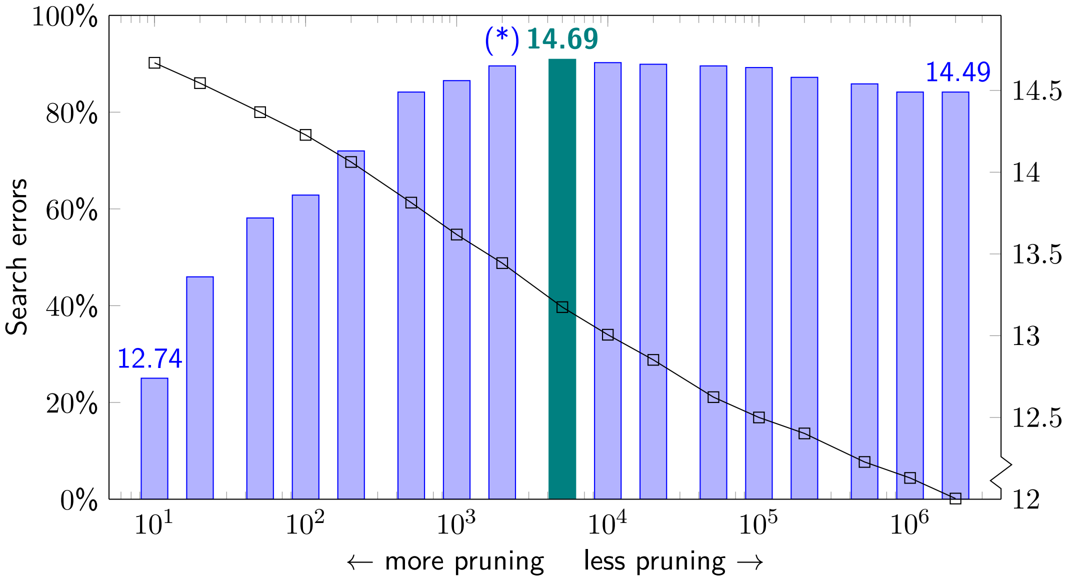
Model error model-optimal translation not a good translation

Does pruning affect translation quality?

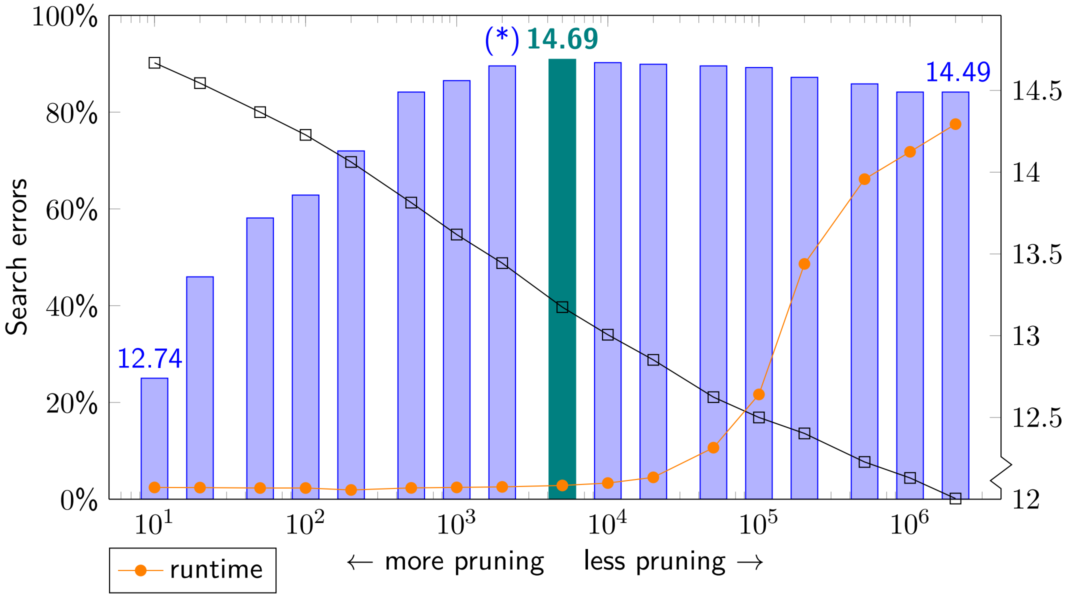
Pruning vs. Translation Quality



Pruning vs. Translation Quality

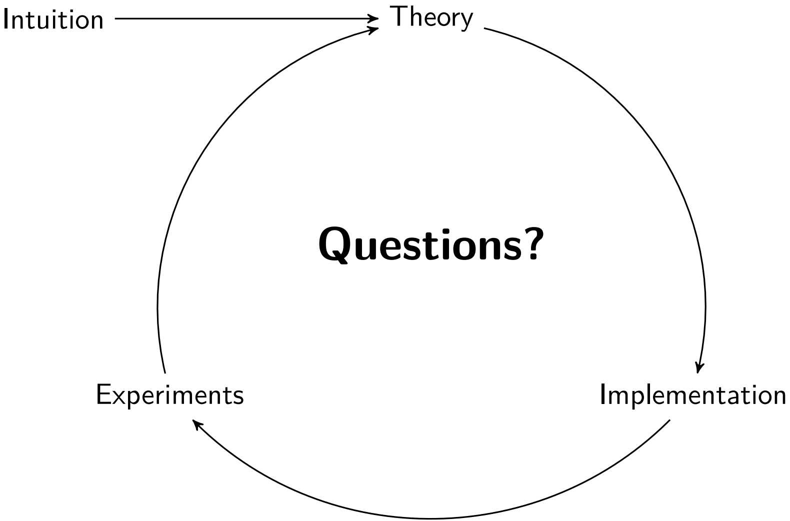


Pruning vs. Translation Quality



The End

...or back to the beginning?



Section 6

Appendix

Ranked alphabet

- (Σ, rk) where Σ is an alphabet and $\mathsf{rk} : \Sigma \to \mathbb{N}$

Trees

- $\Sigma(V) = \bigcup_{k\in\mathbb{N}}\{\sigma(v_1, \dots, v_k) \mid \sigma \in \mathsf{rk}^{-1}(k), v_1, \dots, v_k \in V\}$
- $T_\Sigma(V)$ is the smallest set T such that $V \subseteq T$ and $\Sigma(T) \subseteq T$
- $T_\Sigma = T_\Sigma(\emptyset)$

Positions of a tree

$$\mathsf{pos}(t) = \begin{cases} \{\varepsilon\} & \text{if } t \in V \\ \{\varepsilon\} \cup \bigcup_{1\leq i\leq k}(\{i\}.\,\mathsf{pos}(t_i)) & \text{if } t = \sigma(t_1, \dots, t_k) \end{cases}$$

Trees (2)

Label at position

$$t(w) = \begin{cases} t & \text{if } w = \varepsilon \text{ and } t \in V \\ \sigma & \text{if } w = \varepsilon \text{ and } t = \sigma(t_1, \ldots, t_k) \\ t_i(w') & \text{if } w = iw', i \in [k], w' \in \mathbb{N}^* \text{ and } t = \sigma(t_1, \ldots, t_k) \end{cases}$$

Subtree at position

$$t|_w = \begin{cases} t & \text{if } w = \varepsilon \\ t_i|_{w'} & \text{if } w = iw' \text{ and } t = \sigma(t_1, \ldots, t_k), i \in [k] \end{cases}$$

Yield of a tree

$$\text{yield}(t) = \begin{cases} t & \text{if } t \in \Sigma^{(0)} \cup V \\ \text{yield}(t_1) \cdots \text{yield}(t_k) & \text{if } t = \sigma(t_1, \ldots, t_k) \end{cases}$$

Replacement and Substitutions

Replacement

$$t[u]_w = \begin{cases} u & \text{if } w = \varepsilon \\ \sigma(t_1, \ldots, t_i[u]_{w'}, \ldots, t_k) & \text{if } w = iw' \text{ and } t = \sigma(t_1, \ldots, t_k), i \in [k] \end{cases}$$

Substitutions

$\vartheta : V \to T_\Sigma(V)$ extended to trees:

$$\vartheta(t) = \begin{cases} \vartheta(t) & \text{if } t \in V \\ \sigma(\vartheta(t_1), \ldots, \vartheta(t_k)) & \text{if } t = \sigma(t_1, \ldots, t_k) \end{cases}$$

Ground substitution $\vartheta : V \to T_\Sigma$

Weighted Tree Automata

$$\boxed{\mathcal{G} = (Q, I, P)}$$

Q	finite set of states
$I : Q \to \mathbb{R}$	initial state mapping
$P : Q \times \Delta(Q) \to \mathbb{R}$	transition weight mapping

Weight of a run $r : \mathbf{pos}(t) \to Q$

$$\mathsf{wt}(r) = \prod_{p \in \mathsf{pos}(t)} P(r(p), r'(p))$$

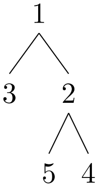
Language of a tree automaton $\mathcal{G}$

$$L_{\mathcal{G}}(t) = \sum_{r \in \mathsf{runs}_{\mathcal{G}}(t)} I(r(\varepsilon)) \cdot \mathsf{wt}(r)$$

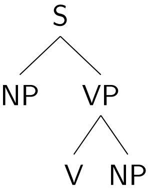
Weighted Tree Automata (2)

$$\mathsf{S} \xrightarrow{1.0} 1(\mathsf{NP}, \mathsf{VP})$$
$$\mathsf{VP} \xrightarrow{1.0} 2(\mathsf{V}, \mathsf{NP})$$
$$\mathsf{NP} \xrightarrow{0.6} 3()$$
$$\mathsf{NP} \xrightarrow{0.4} 4()$$
$$\mathsf{V} \xrightarrow{1.0} 5()$$

weighted transitions



tree t



run on t
weight: 0.24

Regular tree languages

$$\mathsf{REC}(\Delta) = \{L_{\mathcal{G}} \mid \mathcal{G} \text{ is a weighted tree automaton over } \Delta\}$$

Tree Homomorphisms

Let Σ and Δ be ranked alphabets, and let $X_n = \{x_i \mid i \in [n]\}$ be a set of variables for every $n \in \mathbb{N}$. A family f of mappings $\left(f_k : \Sigma^{(k)} \to T_\Delta(X_k)\right)$ determines a homomorphism $h_f : T_\Sigma \to T_\Delta$, called *tree homomorphism*, given by

$$h_f(\sigma(t_1, \ldots, t_k)) = \xi(f_k(\sigma))$$

for every $\sigma \in \Sigma^{(k)}$, where $\xi : X_k \to T_\Delta$ is the ground substitution defined by $\xi(x_i) = h_f(t_i)$.

- if $|\operatorname{pos}_{\{x\}}(f_k(\sigma))| \leq 1$ for every $x \in X_k$ and $\sigma^{(k)} \in \Sigma$, then h is linear
- if $|\operatorname{pos}_{\{x\}}(f_k(\sigma))| \geq 1$ for every $x \in X_k$ and $\sigma^{(k)} \in \Sigma$, then h is nondeleting

If L is a weighted regular tree language over Δ, then $h^{-1}(L)$ is a weighted regular tree language over Σ.

Tree m-morphisms

Let Σ and Δ be ranked alphabets, and $m \in \mathbb{N}^+$. Let $X_n^m = \{x_{(i,j)} \mid i \in [n], j \in [m]\}$ be a set of variables for every $n \in \mathbb{N}$. A family f of mappings $\left(f_k : \Sigma^{(k)} \to T_\Delta(X_k^m)^m\right)$ determines a homomorphism $h_f : T_\Sigma \to T_\Delta^m$, called *tree m-morphism*, given by

$$h_f(\sigma(t_1, \ldots, t_k)) = \xi(f_k(\sigma))$$

for every $\sigma \in \Sigma^{(k)}$, where $\xi : X_k^m \to T_\Delta$ is the ground substitution defined by $\xi(x_{(i,j)}) = \left(h_f(t_i)\right)_j$.

Bimorphism Decompositions

Formalism	Bimorphism	Source
In-BOT =	$\mathcal{B}(\mathsf{REL}, \mathsf{In\text{-}HOM})$	Fülöp et al. (2011)
BOT =	$\mathcal{B}(\mathsf{REL}, \mathsf{HOM})$	Fülöp et al. (2011)
SDTS =	$\mathcal{B}(\mathsf{qA}, \mathsf{qA})$	Steinby and Tîrnăucă (2009)
In-XTOP =	$\mathcal{B}(\mathsf{In\text{-}HOM}, \mathsf{In\text{-}HOM})$	Engelfriet et al. (2009)
In-XMBOT =	$\mathcal{B}(\mathsf{In\text{-}HOM}, \mathsf{MM} ; \pi_1)$	Engelfriet et al. (2009)
SFSG =	$\mathcal{B}(\mathsf{MM} ; \pi_1, \mathsf{MM} ; \pi_1)$	Raoult (1997)
STAG =	$\mathcal{B}(\mathsf{In\text{-}E}, \mathsf{In\text{-}E})$	Shieber (2006)
SCFTG =	$\mathcal{B}(\mathsf{MAC}, \mathsf{MAC})$	Nederhof and Vogler (2012)

Empirical Adequacy

	ROT	DIS	TCH	REG	REG^{-1}	CMP	SYM
In-TOP	no	no	?	yes	yes	yes	no
TOP	yes	no	yes	no	yes	no	no
In-XTOP	yes	no	yes	yes	yes	no	yes
In-XMBOT	yes	yes	yes	no	yes	yes	no
SFSG	yes	yes	yes	no	no	no	yes
STAG	yes	yes	yes	no	no	no	yes

ROT = handles rotations

DIS = handles discontiguity

TCH = efficiently trainable

REG = preservation of regularity

REG^{-1} = preservation of regularity of the inverse

CMP = closure under composition

SYM = symmetry

BLEU Score

n-grams

$$\mathsf{grams}_n(a) = \{v \in A^n \mid \exists u, w \in A^* \text{ such that } uvw = a\}$$

Clipped precision

$$\mathsf{prec}_n(C, R) = \sum_{i=1}^{m} \frac{\sum_{w \in \mathsf{grams}_n(C_i) \cap \mathsf{grams}_n(R_i)} \min\{\#_{C_i}(w), \#_{R_i}(w)\}}{\sum_{w \in \mathsf{grams}_n(C_i)} \#_{C_i}(w)}$$

BLEU-N score

$$\mathsf{bp}(C, R) = \min\left\{1,\ \exp\left(1 - \frac{|R|}{|C|}\right)\right\}$$

$$\mathsf{bleu}_N(C, R) = \mathsf{bp}(C, R) \cdot \prod_{n \in [N]} \mathsf{prec}_n(C, R)^{1/N}$$

Rule Scoring

Relative frequency

$$\phi_{\sim}(\delta) = \frac{\#(\delta)}{\sum_{\delta' \in [\delta]_{\sim}} \#(\delta')}$$

Lexical scoring

$$\mathsf{lex}(f|\,e, a) = \prod_{(i,p)\in \mathsf{dom}(a)} \frac{1}{|a(i,p)|} \sum_{(j,p')\in a(i,p)} w(f_i(p)\,|\,e_j(p'))$$

$$\phi_{\mathsf{lex}}(\delta) = \mathsf{lex}(e'|f', a) \qquad \text{(direct)}$$

$$\phi_{\mathsf{lex}^{-1}}(\delta) = \mathsf{lex}(f'|\,e', a^{-1}) \qquad \text{(inverse)}$$

Inside weight

$$\beta_{\mathcal{G}}(q) = \sum_{(q,\delta(q_1,\dots,q_k))\in \mathsf{dom}(P)} \left(P(q,\delta(q_1,\dots,q_k)) \cdot \prod_{i=1}^{k} \beta_{\mathcal{G}}(q_i) \right)$$

Outside weight

$$\alpha_{\mathcal{G}}(q) = I(q) + \sum_{(p,\delta(p_1,\dots,p_k),w)\in \mathsf{dom}(P)} w \cdot \alpha_{\mathcal{G}}(p) \cdot \sum_{\substack{i\in[k] \\ q=p_j}} \prod_{\substack{i\in[k] \\ i\neq j}} \beta_{\mathcal{G}}(p_i)$$

Importance of a production

$$\gamma_{\mathcal{G}}(\rho) = \alpha_{\mathcal{G}}(p) \cdot w \cdot \prod_{i=1}^{k} \beta_{\mathcal{G}}(p_i)$$

EM Training (2)

Corpus likelihood

$$\sum_{i=1}^{m} \sum_{t \in T_\Delta} L_{D_i}(t)$$

Weighted count

$$\mathsf{ex}(\mathcal{D})(\delta) = \sum_{i=1}^{m} \frac{\sum_{\rho=\delta} \gamma_{D_i}(\rho)}{\sum_{q \in Q} I(q) \cdot \beta_{D_i}(q)}$$

Normalization

$$\mathsf{norm}(P)(\delta) = \frac{P(\delta)}{\sum_{\delta' \sim \delta} P(\delta')}$$

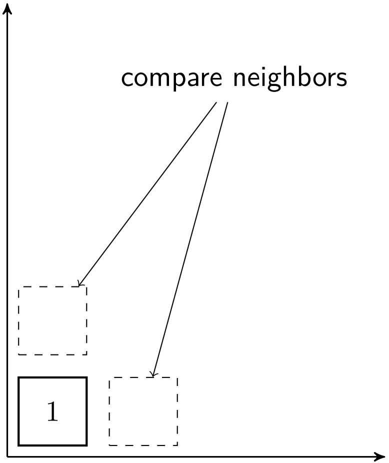
Expectation-maximization

$\mathcal{D}(P) = (D_i(P) \mid i \in [m])$ such that
$P_i(P)(\rho) = P(\rho_2)$ for all $\rho \in \operatorname{dom} P_i(P)$ and all $i \in [m]$

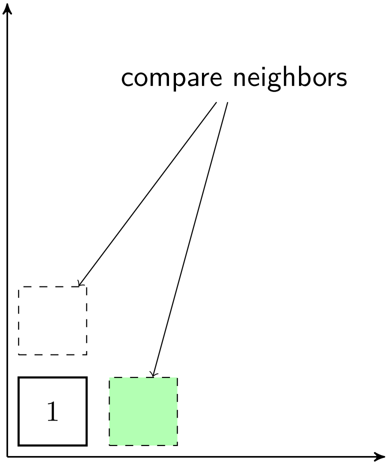
$$P^{(0)} = P_L$$
$$\mathcal{D}^{(n)} = \mathcal{D}(P_n)$$
$$P^{(n+1)} = (\mathrm{ex}\,; \mathrm{norm})\left(\mathcal{D}^{(n)}\right)$$

$$j \leq j' \implies \sum_{i=1}^{m} \sum_{t \in T_\Delta} L_{\mathcal{D}_i^{(j)}}(t) \leq \sum_{i=1}^{m} \sum_{t \in T_\Delta} L_{\mathcal{D}_i^{(j')}}(t)$$

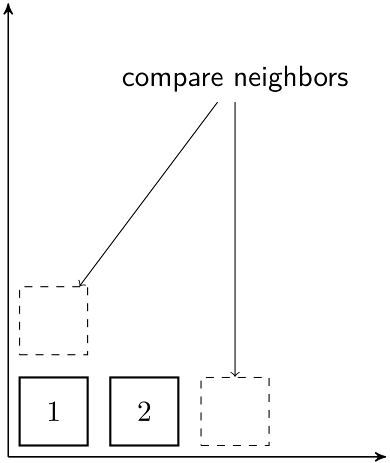
Huang and Chiang (2005)



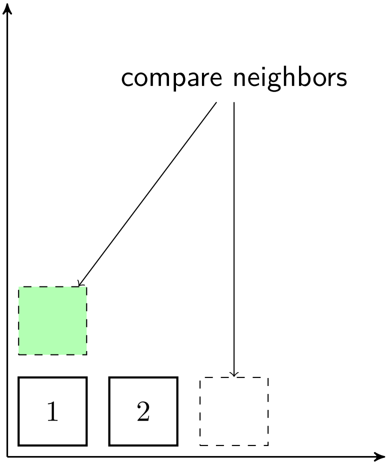
Huang and Chiang (2005)



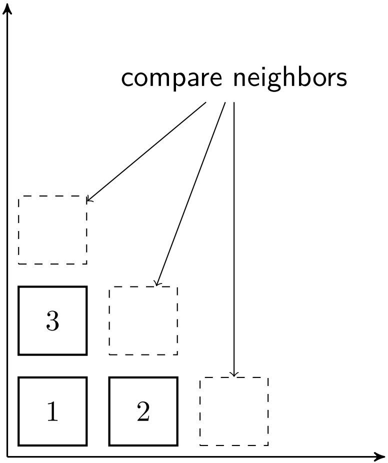
Huang and Chiang (2005)



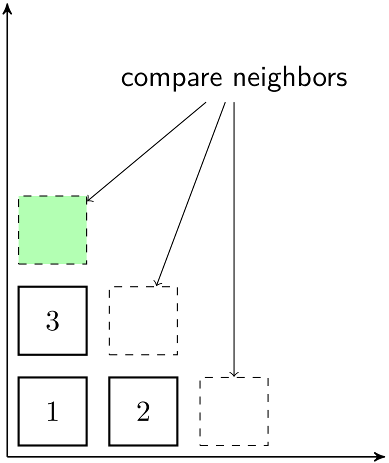
Huang and Chiang (2005)



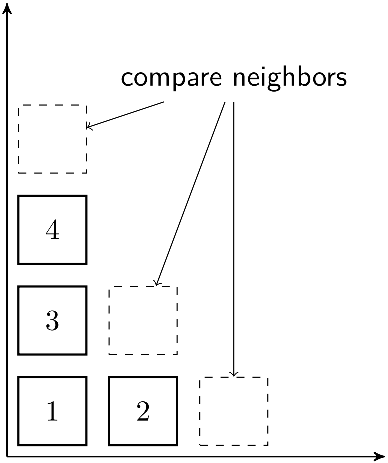
Huang and Chiang (2005)



Huang and Chiang (2005)



Huang and Chiang (2005)



Proof for Exact Rescoring

Assuming that $\mathcal{E}_2(t) = \text{LM(yield(out(t)))}$ for every $t \in \mathcal{F}([k])$, the best candidate in the reordered k-best list is

$$\begin{aligned}\hat{t} &= \underset{t \in \mathcal{F}([k])}{\arg\max}\ F(t) \cdot \text{LM}(\text{yield}(\text{out}(t)))^{\lambda_3} \\ &= \underset{t \in \mathcal{F}([k])}{\arg\max}\, (\mathcal{F} \cdot \mathcal{T} \cdot \mathcal{E}_1^{\lambda_3})(t) \cdot \mathcal{E}_2^{\lambda_3}(t) \\ &= \underset{t \in \mathcal{F}([k])}{\arg\max}\ L(t)\end{aligned}$$

However, $\mathcal{F}([k]) \subseteq \mathcal{D}(\mathbf{f})$, and therefore

$$\hat{t} = \underset{t \in \mathcal{F}([k])}{\arg\max}\, L(t) \text{ does not imply } \hat{t} = \underset{t \in \mathcal{D}(\mathbf{f})}{\arg\max}\, L(t)$$

i.e., we do not have any guarantee that $\hat{t}$ is optimal.

Proof for Exact Rescoring (2)

Lemma

There is $n' \in \mathbb{N}$ such that for every $n \geq n'$,

$$\arg\max_{t \in \mathcal{F}([n])} L(t) = \arg\max_{t \in \mathcal{D}(\mathbf{f})} L(t) \,.$$

Proof.

Let $\hat{t} = \arg\max_{t \in \mathcal{D}(\mathbf{f})} L(t)$. Since both $\mathrm{ran}(\mathrm{LM}) \subseteq [0,1]$ and $\mathrm{ran}(F) \subseteq [0,1]$, we know that

$$L(t) = F(t) \cdot \mathcal{E}_2^{\lambda_3}(t) \leq F(t)$$

for every tree $t \in T_\Delta$ (because $ab \leq a$ for every $a, b \in [0,1]$). Therefore, $F(t) < L(\hat{t})$ implies $L(t) < L(\hat{t})$, and because F is cycle-free, there exists n such that $F(t') < L(\hat{t})$ for all $t' \notin \mathcal{F}([n])$. $\square$

A discrete probability space is a triple $(\Omega, \mathcal{F}, \Pr)$ consisting of

- a non-empty countable set Ω, the set of elementary events
- the set $\mathcal{F} = 2^{\Omega}$, called events
- a mapping $\Pr : \mathcal{F} \to [0, 1]$ such that:
  - $\Pr(\Omega) = 1$; and
  - for every mapping $I : \mathbb{N} \to \mathcal{F}$ such that $I(i) \cap I(j) = \emptyset$ for every $i \neq j$, we have:

$$\Pr\left(\bigcup_{n\in\mathbb{N}} I(n)\right) = \sum_{n\in\mathbb{N}} \Pr(I(n))$$

We call $\Pr(\omega)$ the probability of the event $\omega \in \mathcal{F}$ occurring.

Probability Theory (2)

Let A be a non-empty countable set. A (discrete) random variable X over A is a mapping $X : \Omega \to A$. For $a \in A$, we set

$$\Pr(X = a) = \Pr(X^{-1}(a))$$

When the random variable is understood from context, we can write a instead of $X = a$, and $\Pr(a)$ instead of $\Pr(X = a)$.

Probability Theory (2)

Let X, Y be random variables over A. The joint probability $\Pr(x, y)$ is defined by

$$\Pr(x, y) = \Pr(X^{-1}(x) \cap Y^{-1}(y))$$

The conditional probability $\Pr(y|x)$ of $Y = y$ given $X = x$ is defined by:

$$\Pr(y|x) = \begin{cases} \frac{\Pr(x,y)}{\Pr(x)} & \text{if } \Pr(x) \neq 0 \\ 0 & \text{otherwise.} \end{cases}$$

The joint probability can be expressed using conditional probabilities by the chain rule:

$$\Pr(x, y) = \Pr(x) \cdot \Pr(y|x)$$

Another useful equation is Bayes' rule:

$$\Pr(y|x) = \frac{\Pr(x|y) \cdot \Pr(y)}{\Pr(x)}$$

$U \subseteq \mathbb{N}^*$ finite prefix-closed set (think pos(t) for any tree t), and $p \in U, P, B_1, B_2 \subseteq U$

$$\underline{\bigwedge}_{U} p = \{p' \in \max U \mid p \sqsubseteq p'\} \qquad \text{(“span”)}$$

$$\underline{\bigwedge}_{U} P = \bigcup_{p \in P} \underline{\bigwedge}_{U} p$$

$$\underline{\Delta}_{U} P = \left\{p \in U \mid \underline{\bigwedge}_{U} p \subseteq P\right\} \qquad \text{(“closure”)}$$

$$\min P = \{p \in P \mid \forall p' \in P : p' \sqsubseteq p \implies p' = p\}$$

$$B_1 \sqsubseteq B_2 \iff \forall b_2 \in B_2 : \exists b_1 \in B_1 : b_1 \sqsubseteq b_2$$

Alignment

given by (f, e, a) where $f \in T_F$, $e \in T_E$ and $a \subseteq \text{pos}(f) \times \text{pos}(e)$

$$U_f = \{p \in \text{pos}(f) \mid \exists p' \in \text{dom}(a) : p \sqsubseteq p'\}$$
$$U_e = \{p \in \text{pos}(e) \mid \exists p' \in \text{ran}(a) : p \sqsubseteq p'\}$$

Consistent alignment

$B \bowtie B'$ if there exist $S \subseteq \text{dom}(a)$ and $S' \subseteq \text{ran}(a)$ such that

$$a(S) = S' \text{ and } a^{-1}(S') = S \qquad \text{and}$$
$$B = \min \bigwedge_{U_f} S \text{ and } B' = \min \bigwedge_{U_e} S'$$

Intuitively: $\{f|_b \mid b \in B\}$ and $\{e|_{b'} \mid b' \in B'\}$ can be synchronously generated

Rule Extraction (3)

Unit of translation

Let $B \in \mathrm{dom}(\bowtie)$, and $C \subseteq \mathrm{dom}(\bowtie)$ such that $B \bowtie B'$ and $C \bowtie C'$.
Then $(B, C) \smile (B', C')$ if

- $B \sqsubset C_i$ for all $C_i \in C$
- for all different $C_i, C_j \in C$, all $b_i \in C_i$ and $b_j \in C_j$ are incomparable w.r.t. $\sqsubseteq$

Different models

Let $M \subseteq \bowtie$. Then $(B, C) \smile (B', C')$ is

- an M-unit of translation if $(B, B') \in M$ and $(C_i, C_i') \in M$ for every $C_i \in C$ such that $C_i \bowtie C_i'$
- M-minimal if there are no M-units $(B, C_1) \smile (B', C_1')$ and $(B_1, C_2) \smile (B_1', C_2')$ of translation such that $B_1 \in C_1$ (and $B_1' \in C_1'$) and $C = C_1 - \{B_1\} \cup C_2$

Rule Extraction (4)

Inference rule

$(B, C) \smile (B', C')$ can be represented by an inference rule

$$\frac{f(C_1) \mathrel{\widehat{=}} e(C'_1) \quad \cdots \quad f(C_k) \mathrel{\widehat{=}} e(C'_k)}{f(B) \mathrel{\widehat{=}} e(B')}$$

and partial mappings $\text{in} : (T^*_F)^k \to T^*_F$ and $\text{out} : (T^*_E)^k \to T^*_E$ such that

$$\text{in}(f|_{C_1}, \dots, f|_{C_k}) = f|_B \qquad \text{and}$$
$$\text{out}(e|_{C'_1}, \dots, e|_{C'_k}) = e|_{B'} \,.$$

(connection to tree m-morphisms!)

Rule templates

- generalize “in” and “out” to the $\equiv$ relation previously discussed
- use a rule identifier as an alphabet symbol

Rule Extraction (5)

Let $X = \{x_{(i,j)} \mid i, j \in \mathbb{N}\}$ be a set of variables, and let us define two mappings $v_f : \bigcup_{i=1}^k C_i \to X$ and $v_e : \bigcup_{i=1}^k C'_i \to X$ such that:

$$\text{index}(C_i)(p) = j \implies v_f(p) = x_{(i,j)} \quad \text{for all } p \in \bigcup_{i=1}^{k} C_i \text{ and}$$

$$\text{index}(C'_i)(p') = j \implies v_e(p') = x_{(i,j)} \quad \text{for all } p' \in \bigcup_{i=1}^{k} C'_i\,.$$

These mappings are unambiguously defined because $C_i \cap C_j = \emptyset$ and $C'_i \cap C'_j = \emptyset$ for all $i \neq j$. Now we can define in and out as follows:

$$\text{in}(f_1, \ldots, f_k) = \xi(f'|_B) \qquad \text{and}$$
$$\text{out}(e_1, \ldots, e_k) = \xi'(e'|_{B'})\,,$$

where $\xi(x_{(i,j)}) = (f_i)_j$ and $\xi'(x_{(i,j)}) = (e_i)_j$ are ground substitutions, and f' and e' are obtained by chained replacement of subtrees by variables as follows:

$$f' = (\ldots f[v_f(C_1^{\leq})]_{C_1^{\leq}} \ldots)[v_f(C_k^{\leq})]_{C_k^{\leq}} \qquad \text{and}$$
$$e' = (\ldots e[v_e(C_1'^{\leq})]_{C_1'^{\leq}} \ldots)[v_e(C_k'^{\leq})]_{C_k'^{\leq}}\,.$$

Note that in and out are determined by $f'|_B$ and $e'|_{B'}$, respectively.

Experiment Setups

Generic experiment setup

- Rule extraction from word-aligned bilingual data *(training set)*
- Estimation of rule weights by relative frequency
- Tuning of model parameters on reference translations *(development set)*
- Automatic evaluation on more reference translations *(test set)*

Training set	European parliament proceedings EN/DE
Development set	from WMT shared task
Test set	from WMT shared task
Automatic evaluation	BLEU metric computes n-gram precision

Experiment B: Search Errors

Search error
Pruning removes the model-optimal translation from the search space

Model error
Model-optimal translation is not a good translation

Experiment B: Search Errors

Search error

Pruning removes the model-optimal translation from the search space

Model error

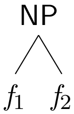
Model-optimal translation is not a good translation

Questions

- How many search errors does a standard MT system make?
- How does the number of search errors affect quality?
- Do some search errors hide underlying model errors?

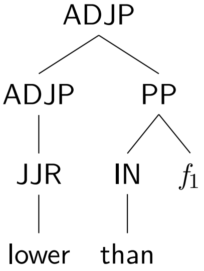
Travatar Tree-to-String System

$$\frac{\text{DT} \quad \text{NN}}{\text{NP}}$$

$\text{in}(f_1, f_2) =$ 

$\text{out}(e_1, e_2) = e_1\, e_2$

$$\frac{\text{PP}}{\text{ADJP}}$$

$\text{in}(f_1) =$ 

$\text{out}(e_1) = \text{niedriger als } e_1$

Related Work

Interpreted Regular Tree Grammar (iRTG)

- similar to bimorphisms
- regular tree grammar + homomorphisms into algebras + evaluation functions
- can be simulated by appropriately chosen homomorphism, and vice versa

Büchse (2015)

- algebraic specification of a decoder
- presentation limited to SCFG and STSG

Related Work (2)

Grammatical Framework[1]

- abstract syntax and a set of concrete syntaxes
- abstract syntax defines a system of syntax trees
- each concrete syntax defines a mapping from those syntax trees to nested tuples of strings and integers
- this mapping is compositional, i.e. homomorphic, and moreover reversible
- forward application of concrete syntax: linearization
- reverse application of concrete syntax: parsing

[1] A. Ranta. Grammatical Framework: A Type-Theoretical Grammar Formalism. Journal of Functional Programming, 14(2), pp. 145–189, 2004.