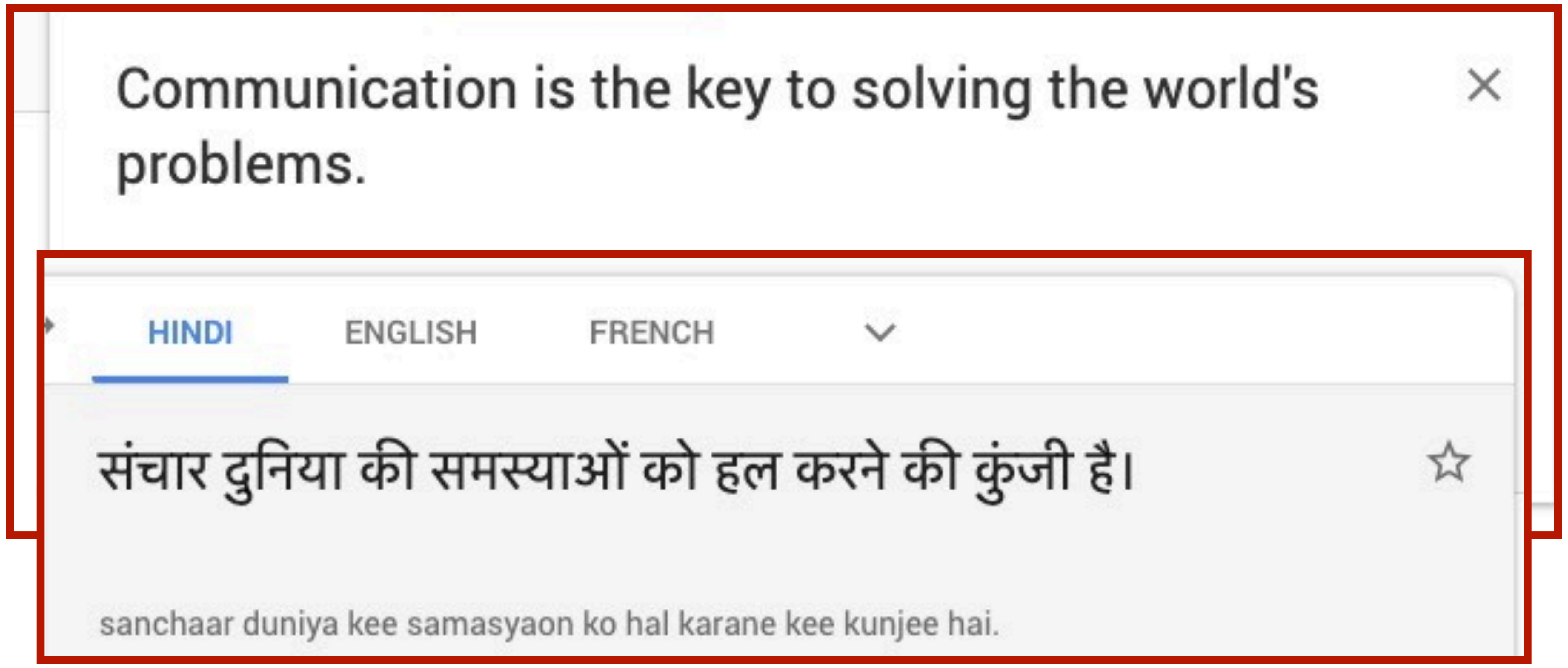
COS 484/584

L15: Machine Translation

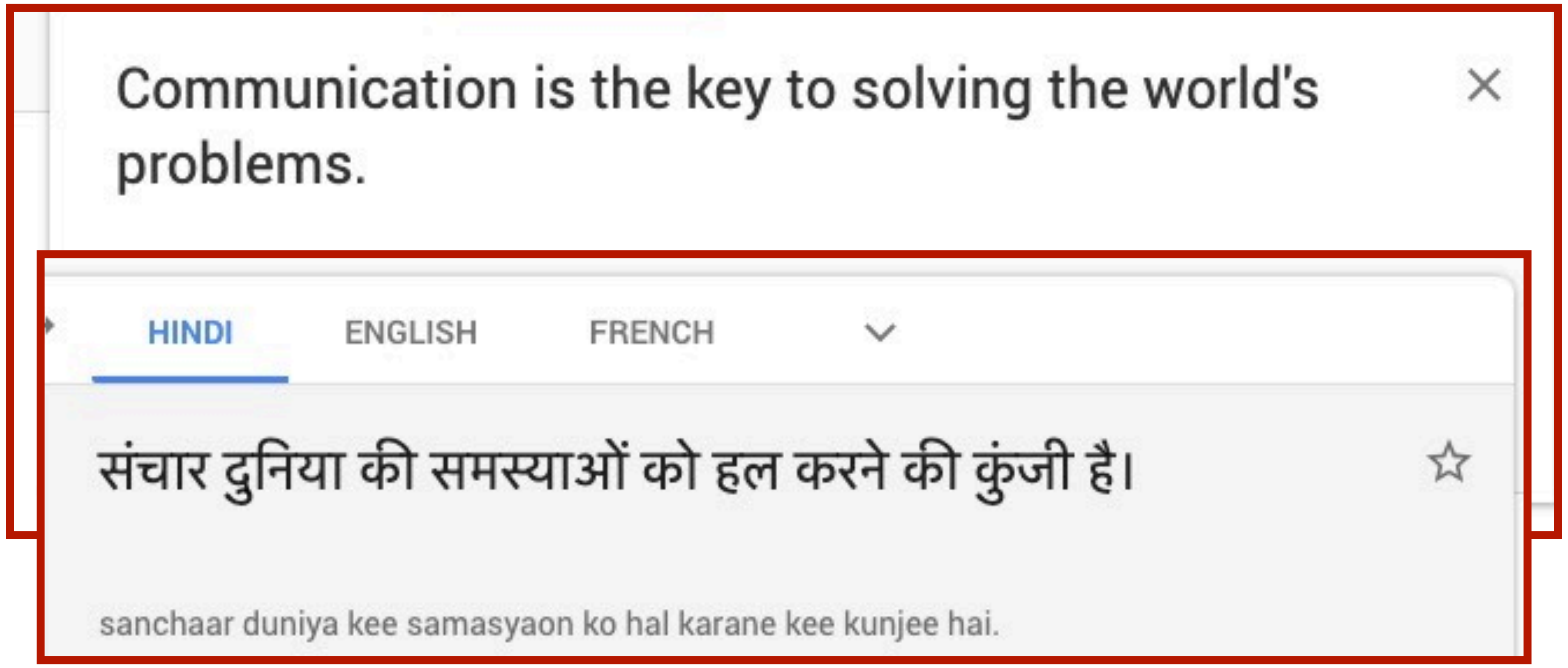
Spring 2021

Translation



- One of the "holy grail" problems in artificial intelligence
- Practical use case: Facilitate communication between people in the world
- Extremely challenging (especially for low-resource languages)

Translation



How many languages do you speak?

A) 1
B) 2
C) 3+

Some translations

- Easy:
  - I like apples ↔ ich mag Äpfel (German)
- Not so easy:
  - I like apples ↔ J'aime les pommes (French)
  - I like red apples ↔ J'aime les pommes rouges (French)
  - *les* ↔ *the* but *les pommes* ↔ *apples*

Basics of machine translation

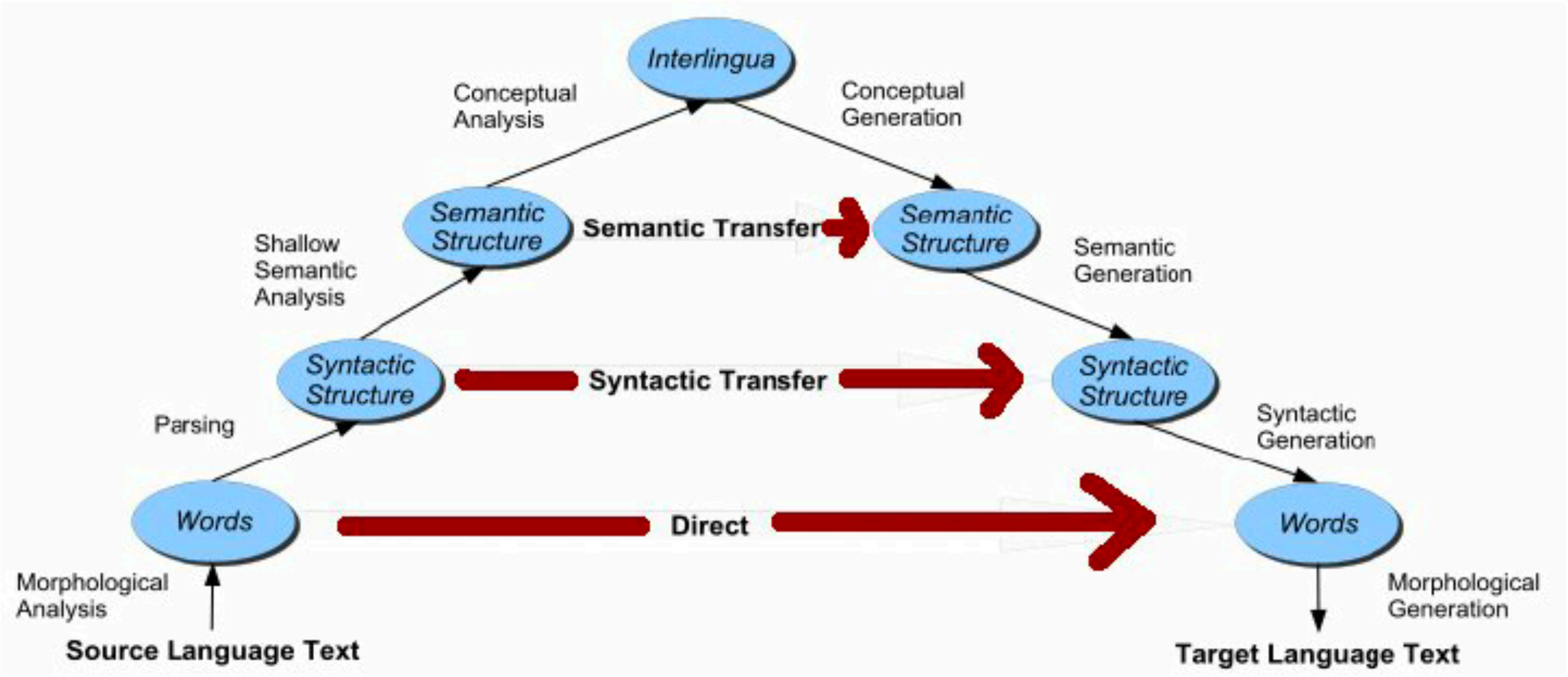
- Goal: Translate a sentence $w^{(s)}$ in a **source language (input)** to a sentence in the **target language (output)**
- Can be formulated as an optimization problem:
  - Most likely translation, $\hat{w}^{(t)} = \arg\max_{w^{(t)}} \psi\left(w^{(s)}, w^{(t)}\right)$
  - where ψ is a scoring function over source and target sentences
- Requires two components:
  - *Learning algorithm* to compute parameters of ψ
  - *Decoding algorithm* for computing the best translation $\hat{w}^{(t)}$

Why is MT challenging?

- Single words may be replaced with multi-word phrases
  - I like apples ↔ J'aime les pommes
- Reordering of phrases
  - I like red apples ↔ J'aime les pommes rouges
- Contextual dependence
  - *les* ↔ *the* but *les pommes* ↔ *apples*

Extremely large output space $\Longrightarrow$ Decoding is NP-hard

Vauquois Pyramid



- Hierarchy of concepts and distances between them in different languages
- Lowest level: individual words/ characters
- Higher levels: syntax, semantics
- Interlingua: Generic language-agnostic representation of meaning

Evaluating machine translation

- Two main criteria:
  - Adequacy: Translation $w^{(t)}$ should adequately reflect the linguistic content of $w^{(s)}$
  - Fluency: Translation $w^{(t)}$ should be fluent text in the target language

To Vinay it like Python
Vinay debugs memory leaks
Vinay likes Python

Different translations of *"A Vinay le gusta Python"*

Which of these translations is both adequate and fluent?

A) first
B) second
C) third

Evaluation metrics

- Manual evaluation: ask a native speaker to verify the translation
  - Most accurate, but expensive
- Automated evaluation metrics:
  - Compare system hypothesis with reference translations
  - BiLingual Evaluation Understudy (BLEU) (Papineni et al., 2002):
    - Modified n-gram precision

$$p_n = \frac{\text{number of } n\text{-grams appearing in both } \boxed{\text{reference}} \text{ and } \boxed{\text{hypothesis}} \text{ translations}}{\text{number of } n\text{-grams appearing in the } \boxed{\text{hypothesis}} \text{ translation}}$$

Reference translation

System predictions

BLEU

$$\text{BLEU} = \exp \frac{1}{N} \sum_{n=1}^{N} \log p_n$$

$$p_n = \frac{\text{number of } n\text{-grams appearing in both reference and hypothesis translations}}{\text{number of } n\text{-grams appearing in the hypothesis translation}}$$

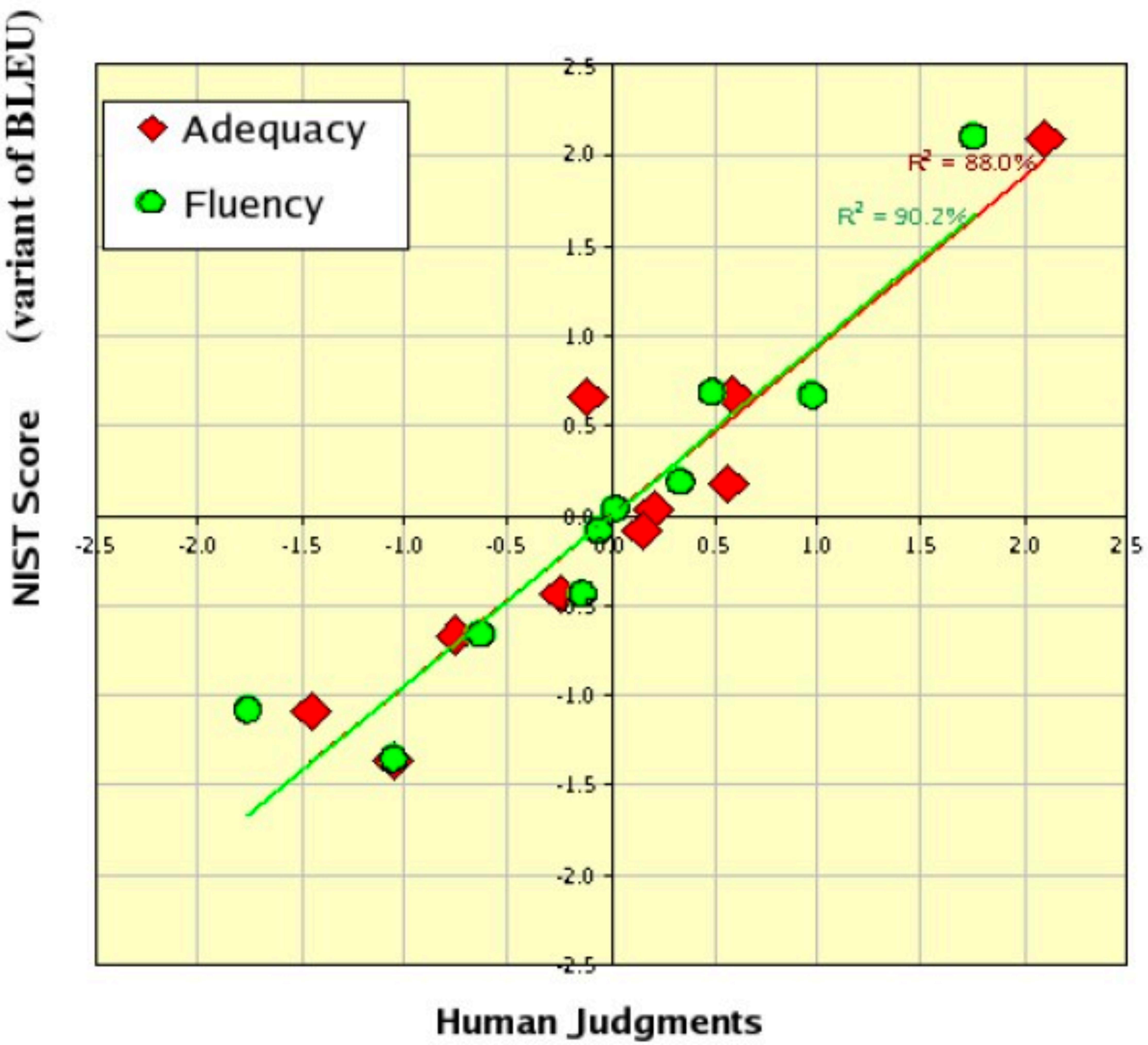
- To avoid $\log 0$, all precisions are smoothed
- Each n-gram in reference can be used at most once
  - Ex. **Hypothesis**: *to to to to to* vs **Reference**: *to be or not to be* should not get a unigram precision of 1
- BLEU-k: average of BLEU scores computed using 1-gram through k-gram.

Precision-based metrics favor short translations

- Solution: Multiply score with a brevity penalty for translations shorter than reference, $e^{1-r/h}$

BLEU

- Correlates with human judgements



(G. Doddington, NIST)

BLEU scores

BP: brevity penalty

	Translation	p_1	p_2	p_3	p_4	BP
Reference	*Vinay likes programming in Python*					
Sys1	*To Vinay it like to program Python*	$\frac{2}{7}$	0	0	0	1
Sys2	*Vinay likes Python*	$\frac{3}{3}$	$\frac{1}{2}$	0	0	.51
Sys3	*Vinay likes programming in his pajamas*	$\frac{4}{6}$	$\frac{3}{5}$	$\frac{2}{4}$	$\frac{1}{3}$	1

Sample BLEU scores for various system outputs

- Alternatives have been proposed:
  - METEOR: weighted F-measure
  - Translation Error Rate (TER): Edit distance between hypothesis and reference

Which of these translations do you think will have the highest BLEU-4 score?

A) sys1

B) sys2

C) sys3

Data

- Statistical MT relies requires **parallel corpora (bilingual)**

		de	es
1.	**Chapter 4, Koch (DE)**	**de**	**es**
context	We would like to ensure that there is a reference to this **as early as the recitals** and that the period within which the Council has to make a decision - which is not clearly worded - is set at a maximum of three months .	Wir möchten sicherstellen , daß hierauf bereits in den Erwägungsgründen hingewiesen wird und die uneindeutig formulierte Frist , innerhalb der der Rat eine Entscheidung treffen muß , auf maximal drei Monate fixiert wird .	Quisiéramos asegurar que se aluda ya a esto en los considerandos y que el plazo , imprecisamente formulado , dentro del cual el Consejo ha de adoptar una decisión , se fije en tres meses como máximo .
2.	**Chapter 3, FÃ¤rm (SV)**	**de**	**es**
context	Our experience of modern administration tells us that openness , decentralisation of responsibility and qualified evaluation are often **as effective as detailed bureaucratic supervision** .	Unsere Erfahrungen mit moderner Verwaltung besagen , daß Transparenz , Dezentralisation der Verantwortlichkeiten und eine qualifizierte Auswertung oft ebenso effektiv sind wie bürokratische Detailkontrolle .	Nuestras experiencias en materia de administración moderna nos señalan que la apertura , la descentralización de las responsabilidades y las evaluaciones bien hechas son a menudo tan eficaces como los controles burocráticos detallados .

(Europarl, Koehn, 2005)

- And lots of it!

- Not easily available for many low-resource languages in the world

Statistical MT

$$\hat{w}^{(t)} = \arg\max_{w^{(t)}} \psi\left(w^{(s)}, w^{(t)}\right)$$

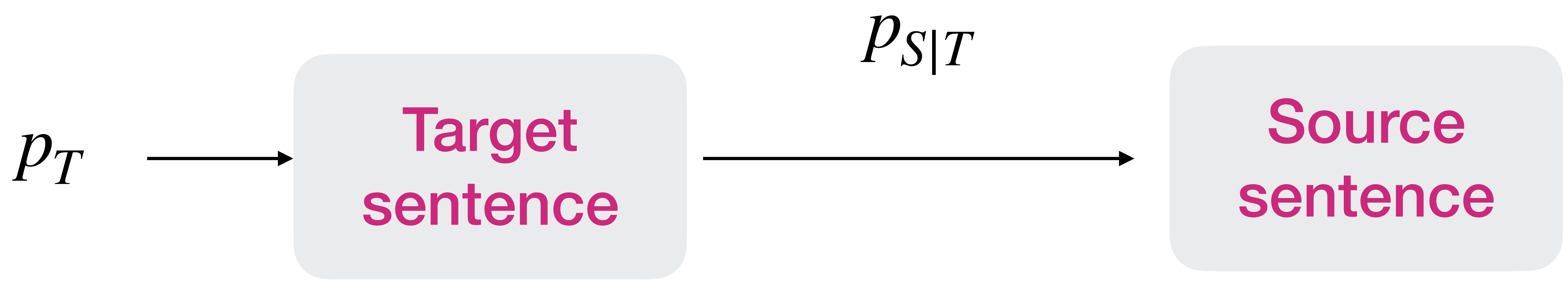
- We can break down the scoring function ψ as:

$$\psi\left(w^{(s)}, w^{(t)}\right) = \psi_A\left(w^{(s)}, w^{(t)}\right) + \psi_F\left(w^{(t)}\right)$$

(adequacy) *(fluency)*

- Allows us to estimate parameters of ψ on separate data
  - ψ_A from aligned bilingual corpora
  - ψ_F from monolingual corpora

Noisy channel model



$$\Psi_A(\boldsymbol{w}^{(s)}, \boldsymbol{w}^{(t)}) \triangleq \log \mathrm{p}_{S|T}(\boldsymbol{w}^{(s)} \mid \boldsymbol{w}^{(t)}) \qquad \textit{(adequacy)}$$

$$\Psi_F(\boldsymbol{w}^{(t)}) \triangleq \log \mathrm{p}_T(\boldsymbol{w}^{(t)}) \qquad \textit{(fluency)}$$

$$\Psi(\boldsymbol{w}^{(s)}, \boldsymbol{w}^{(t)}) = \log \mathrm{p}_{S|T}(\boldsymbol{w}^{(s)} \mid \boldsymbol{w}^{(t)}) + \log \mathrm{p}_T(\boldsymbol{w}^{(t)}) = \log \mathrm{p}_{S,T}(\boldsymbol{w}^{(s)}, \boldsymbol{w}^{(t)}). \qquad \textit{(overall)}$$

- Generative process for source sentence
- Use Bayes rule to recover $w^{(t)}$ that is maximally likely under the conditional distribution $p_{T|S}$ (which is what we want)

$$\arg\max_T p_{T|S} = \arg\max_T \frac{p_T \; p_{S|T}}{p_S}$$

Noisy channel model

$$\Psi_A(\boldsymbol{w}^{(s)}, \boldsymbol{w}^{(t)}) \triangleq \log \mathrm{p}_{S|T}(\boldsymbol{w}^{(s)} \mid \boldsymbol{w}^{(t)})$$

$$\Psi_F(\boldsymbol{w}^{(t)}) \triangleq \log \mathrm{p}_T(\boldsymbol{w}^{(t)})$$

$$\Psi(\boldsymbol{w}^{(s)}, \boldsymbol{w}^{(t)}) = \log \mathrm{p}_{S|T}(\boldsymbol{w}^{(s)} \mid \boldsymbol{w}^{(t)}) + \log \mathrm{p}_T(\boldsymbol{w}^{(t)}) = \log \mathrm{p}_{S,T}(\boldsymbol{w}^{(s)}, \boldsymbol{w}^{(t)}).$$

Allows us to use a standalone language model p_T to improve fluency

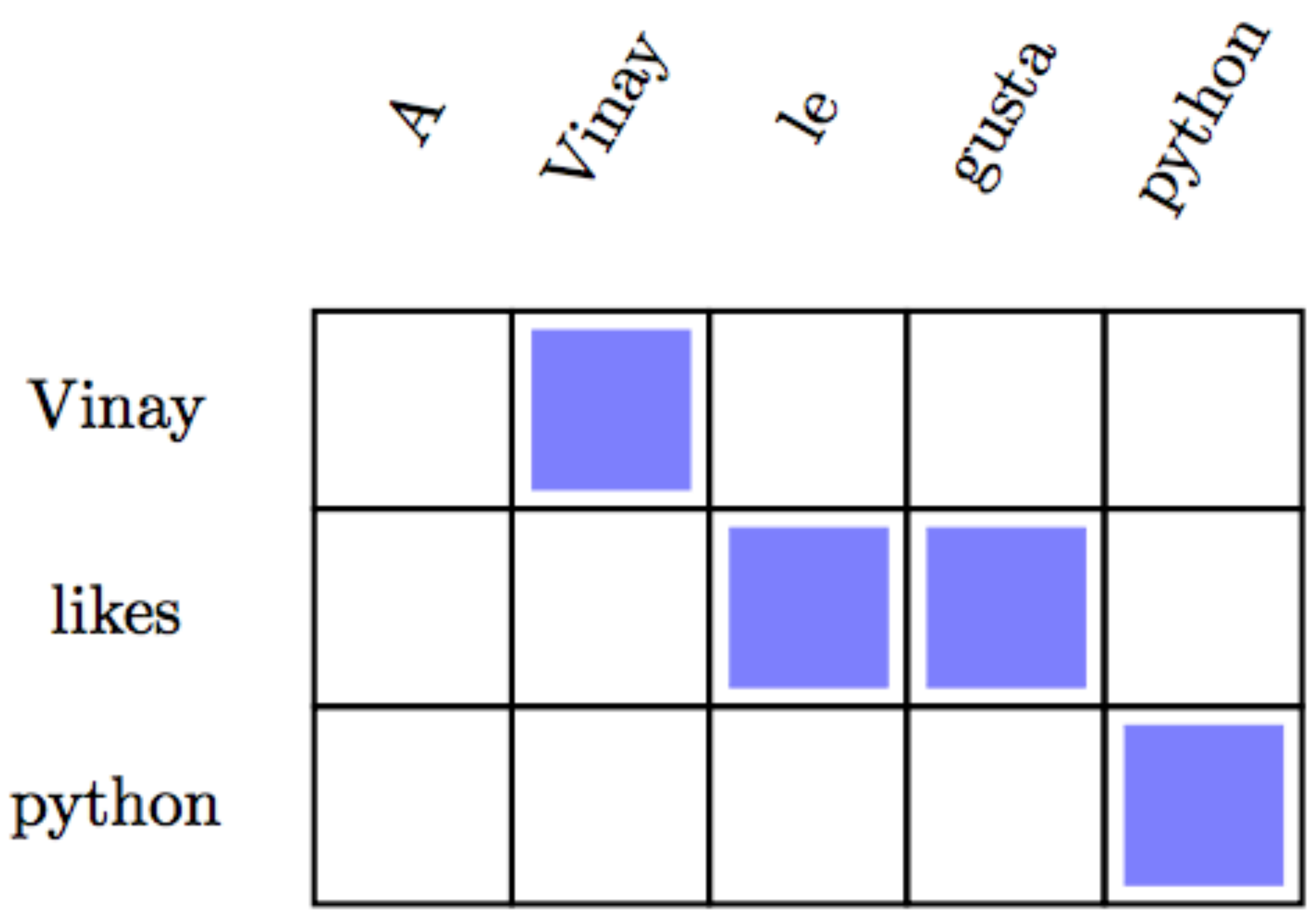
- Use Bayes rule to recover $w^{(t)}$ that is maximally likely under the conditional distribution $p_{T|S}$ (which is what we want)

IBM Models

- Early approaches to statistical MT
- *Key questions:*
  - How do we define the translation model $p_{S|T}$?
  - How can we estimate the parameters of the translation model from parallel training examples?
- Make use of the idea of **alignments**

Alignments

How should we align words in source to words in target?



good $\mathcal{A}(\boldsymbol{w}^{(s)}, \boldsymbol{w}^{(t)}) = \{(A, \varnothing), (\textit{Vinay, Vinay}), (\textit{le, likes}), (\textit{gusta, likes}), (\textit{Python,Python})\}.$

bad $\mathcal{A}(\boldsymbol{w}^{(s)}, \boldsymbol{w}^{(t)}) = \{(\textit{A, Vinay}), (\textit{Vinay, likes}), (\textit{le, Python}), (\textit{gusta}, \varnothing), (\textit{Python}, \varnothing)\}.$

Incorporating alignments

- Let us define the joint probability of alignment and translation as:

$$\begin{aligned} p(\boldsymbol{w}^{(s)}, \mathcal{A} \mid \boldsymbol{w}^{(t)}) &= \prod_{m=1}^{M^{(s)}} p(w_m^{(s)}, a_m \mid w_{a_m}^{(t)}, m, M^{(s)}, M^{(t)}) \\ &= \prod_{m=1}^{M^{(s)}} p(a_m \mid m, M^{(s)}, M^{(t)}) \times p(w_m^{(s)} \mid w_{a_m}^{(t)}). \end{aligned}$$

- $M^{(s)}, M^{(t)}$ are the number of words in source and target sentences
- a_m is the alignment of the m^{th} word in the source sentence
  - i.e. it specifies that the m^{th} word in source is aligned to the $a_m{}^{th}$ word in target
- Translation probability for word in source to be a translation of its alignment word

Independence assumptions

$$\begin{aligned}
p(\boldsymbol{w}^{(s)}, \mathcal{A} \mid \boldsymbol{w}^{(t)}) &= \prod_{m=1}^{M^{(s)}} p(w_m^{(s)}, a_m \mid w_{a_m}^{(t)}, m, M^{(s)}, M^{(t)}) \\
&= \prod_{m=1}^{M^{(s)}} p(a_m \mid m, M^{(s)}, M^{(t)}) \times p(w_m^{(s)} \mid w_{a_m}^{(t)}).
\end{aligned}$$

- Two independence assumptions:
  - Alignment probability factors across tokens:

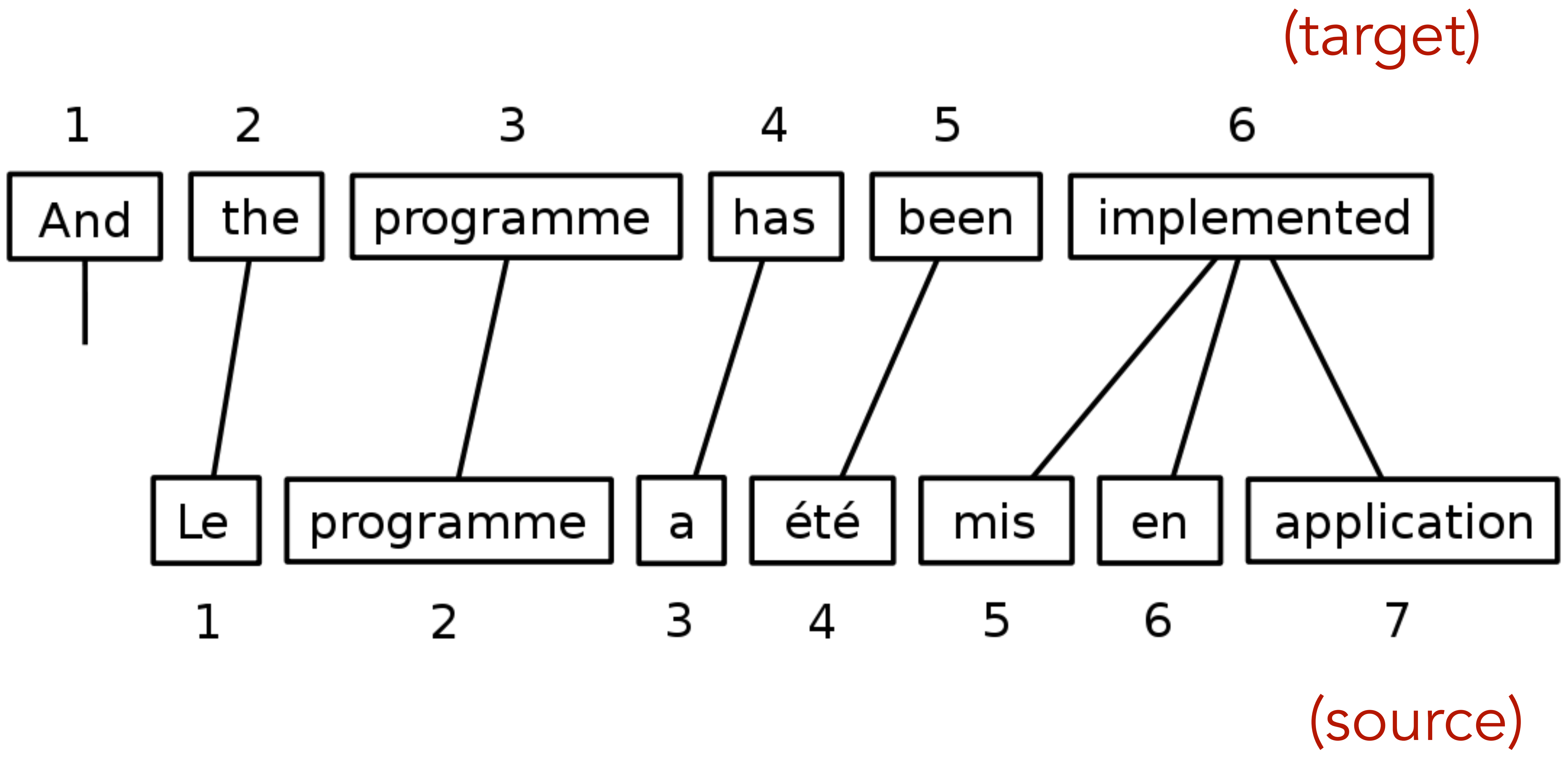
$$p(\mathcal{A} \mid \boldsymbol{w}^{(s)}, \boldsymbol{w}^{(t)}) = \prod_{m=1}^{M^{(s)}} p(a_m \mid m, M^{(s)}, M^{(t)}).$$

  - Translation probability factors across tokens:

$$p(\boldsymbol{w}^{(s)} \mid \boldsymbol{w}^{(t)}, \mathcal{A}) = \prod_{m=1}^{M^{(s)}} p(w_m^{(s)} \mid w_{a_m}^{(t)}),$$

$$p(\boldsymbol{w}^{(s)}, \mathcal{A} \mid \boldsymbol{w}^{(t)}) = \prod_{m=1}^{M^{(s)}} p(w_m^{(s)}, a_m \mid w_{a_m}^{(t)}, m, M^{(s)}, M^{(t)})$$

$$= \prod_{m=1}^{M^{(s)}} p(a_m \mid m, M^{(s)}, M^{(t)}) \times p(w_m^{(s)} \mid w_{a_m}^{(t)}).$$



Can our translation model work well in this case?

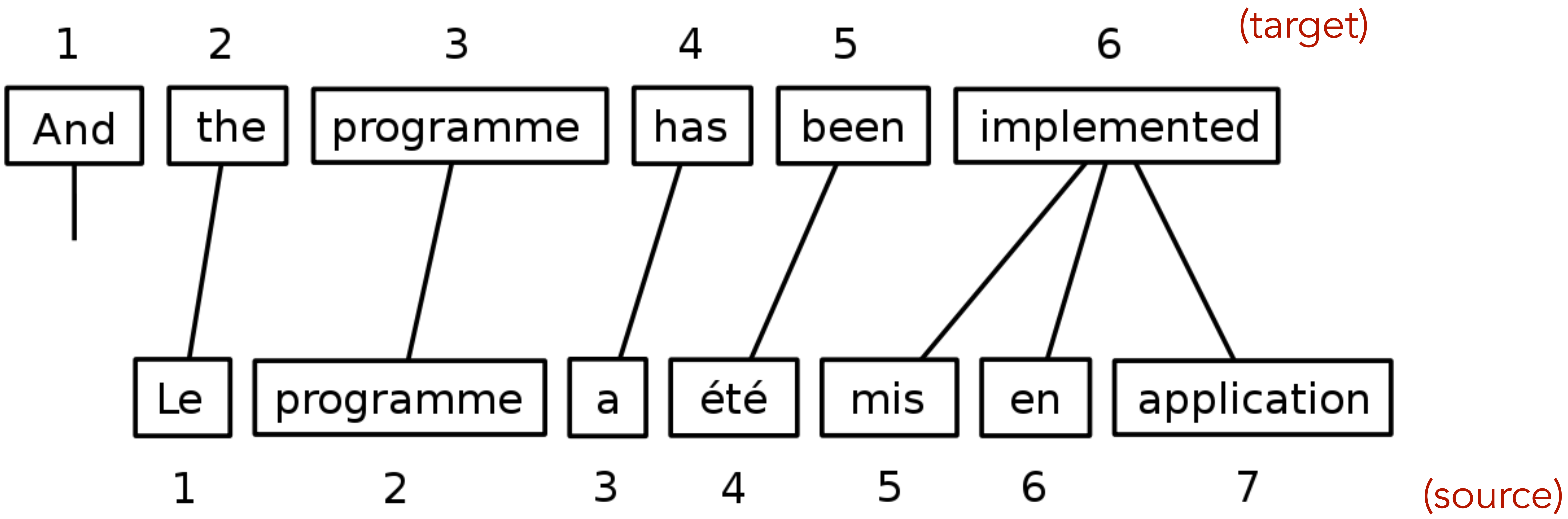
A) Yes

B) No

C) Sometimes

$a_1 = 2,\ a_2 = 3,\ a_3 = 4, \ldots$

Limitations

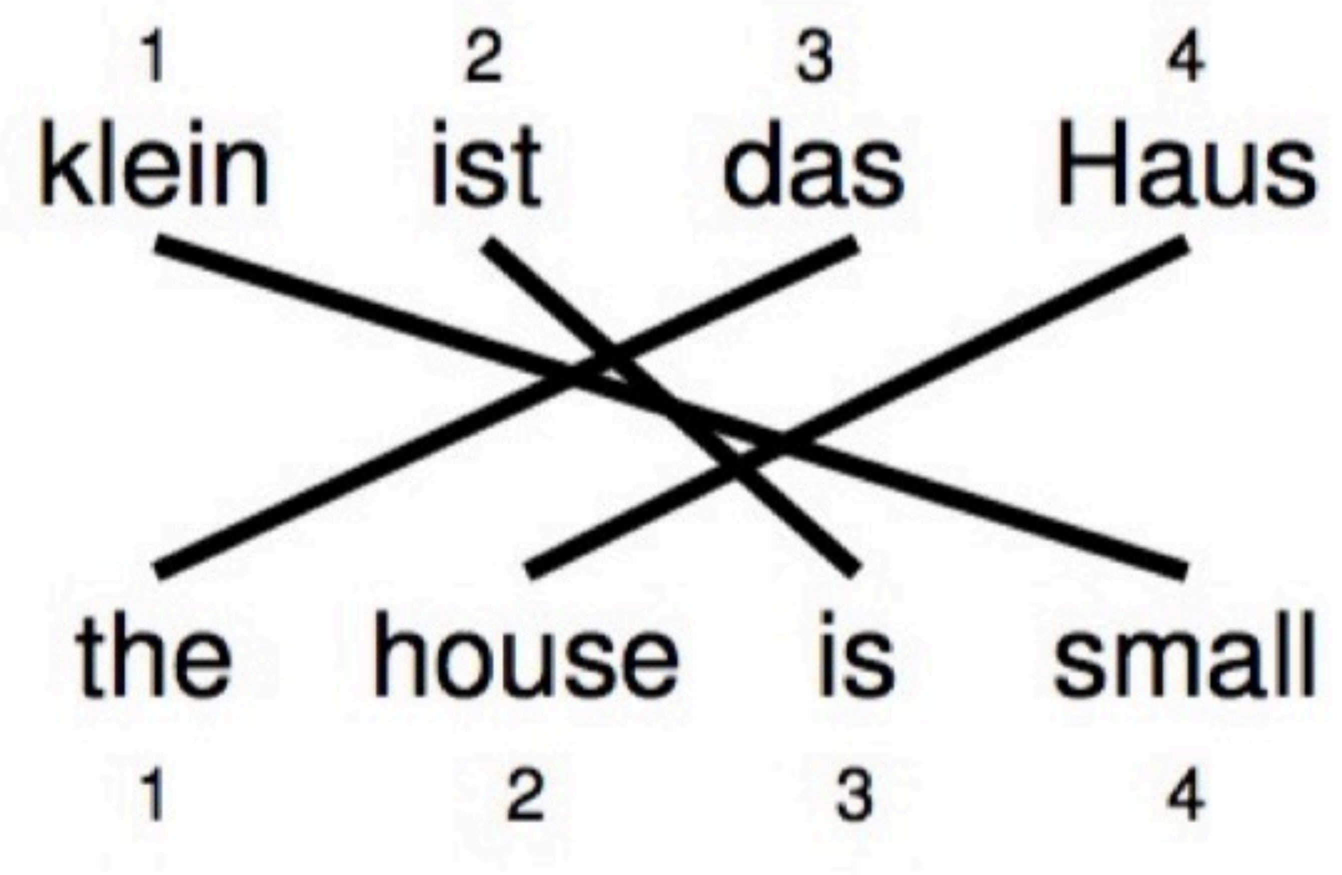


$$a_1 = 2,\ a_2 = 3,\ a_3 = 4, \ldots$$

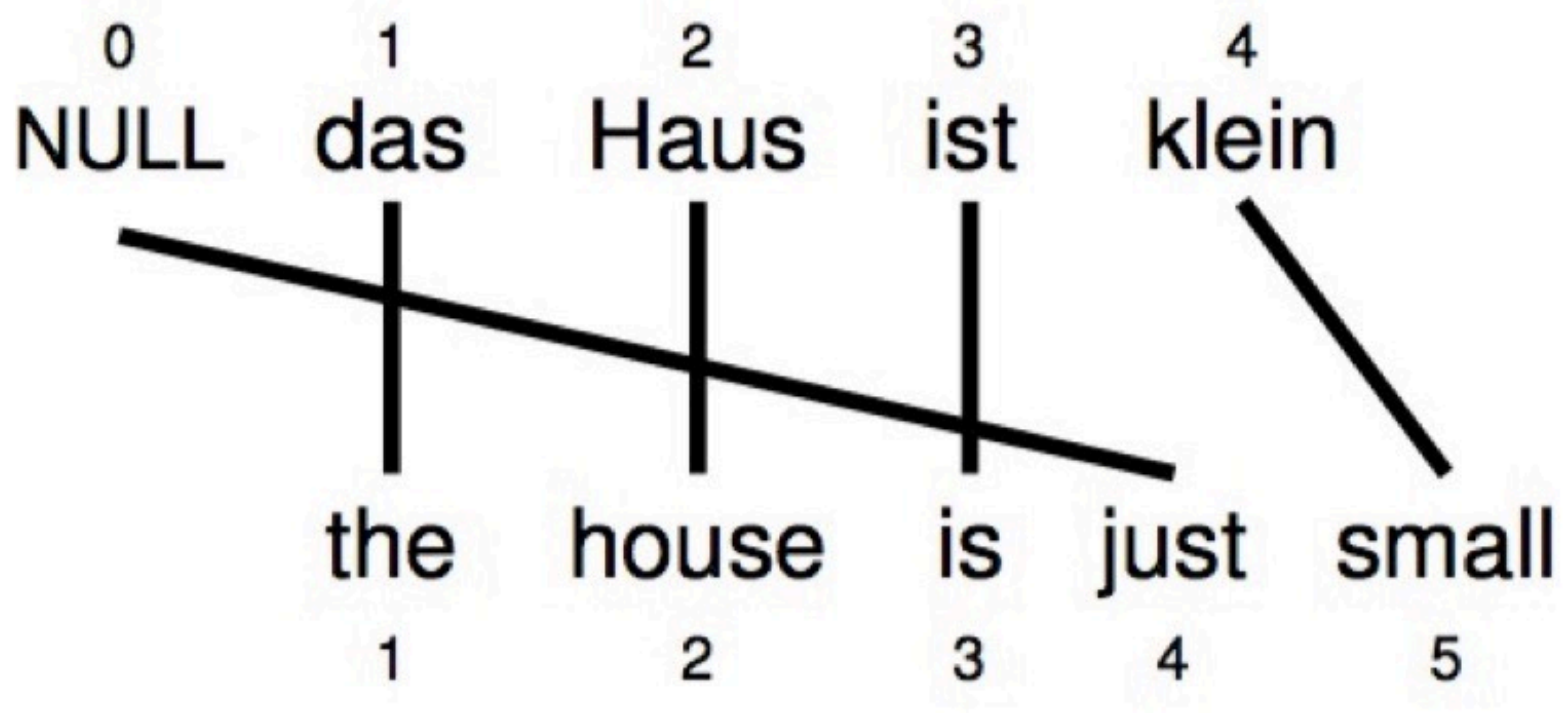
Multiple source words may align to the same target word!

Or a source word may not have any corresponding target.

Reordering and word insertion



$$\mathbf{a} = (3, 4, 2, 1)^\top$$



$$\mathbf{a} = (1, 2, 3, 0, 4)^\top$$

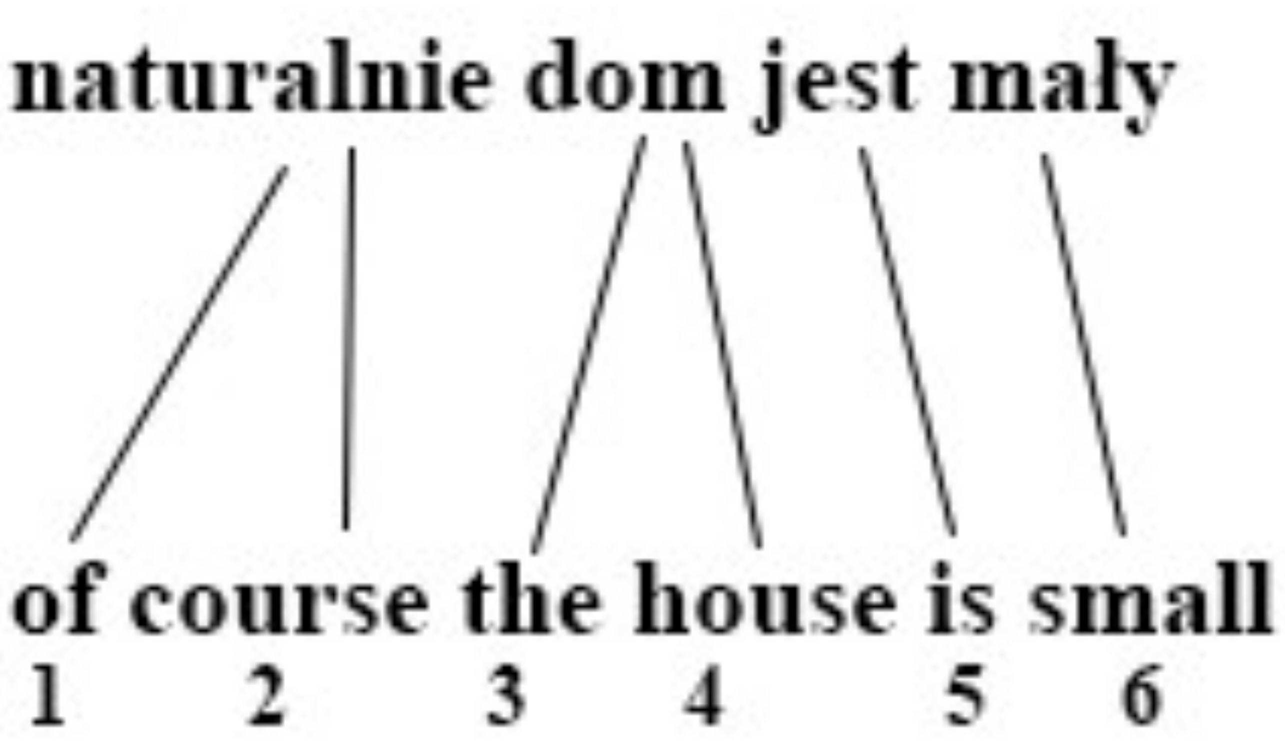
Assume extra NULL token

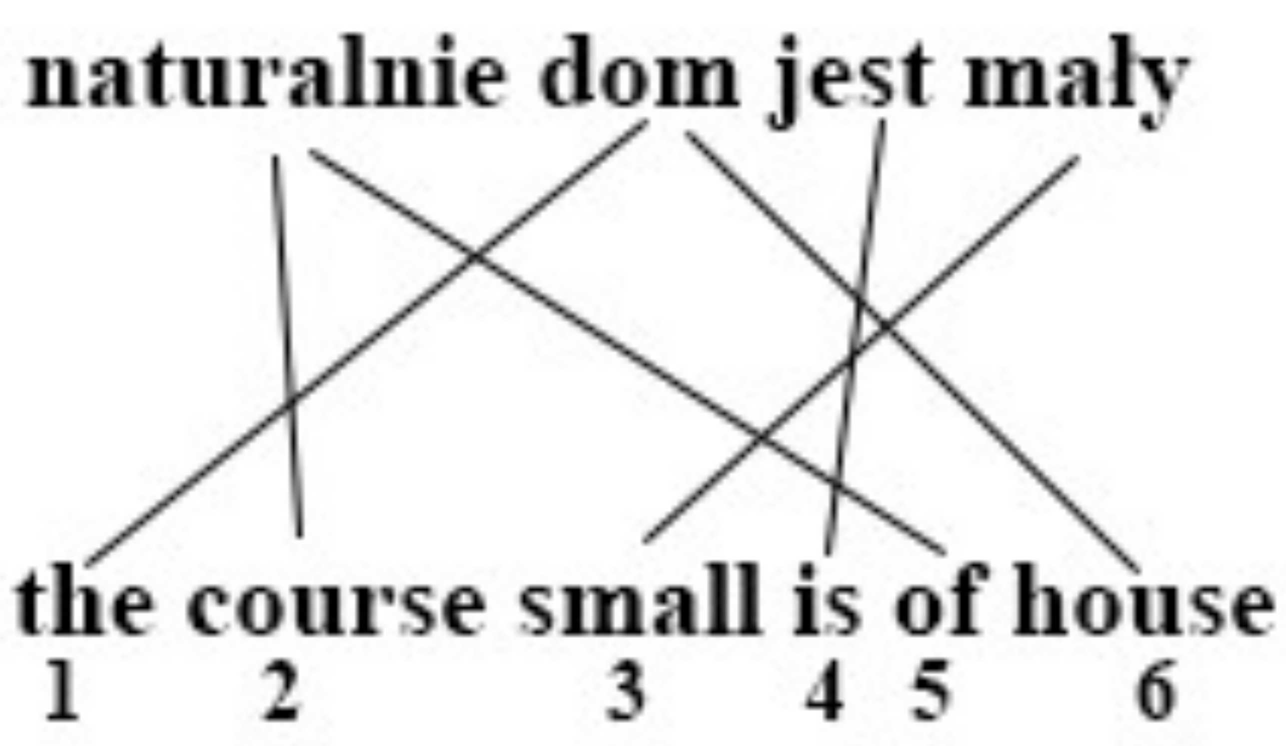
(Slide credit: Brendan O'Connor)

IBM Model 1

- Assume $p(a_m \mid m, M^{(s)}, M^{(t)}) = \frac{1}{M^{(t)}}$
- Is this a good assumption?

$$\mathrm{p}(\mathcal{A} \mid \boldsymbol{w}^{(s)}, \boldsymbol{w}^{(t)}) = \prod_{m=1}^{M^{(s)}} \mathrm{p}(a_m \mid m, M^{(s)}, M^{(t)}).$$





Every alignment is equally likely!

IBM Model 1

- Assume $p(a_m | m, M^{(s)}, M^{(t)}) = \frac{1}{M^{(t)}}$

- We then have:

$$p(w^{(s)}, w^{(t)}) = p(w^{(t)}) \sum_{A} (\frac{1}{M^{(t)}})^{M^{(s)}} \; p(w^{(s)} | w^{(t)})$$

- How do we estimate $p(w^{(s)} = v | w^{(t)} = u)$?

IBM Model 1

- If we have word-to-word alignments, we can compute the probabilities using the MLE:
  - $p(v \mid u) = \dfrac{count(u, v)}{count(u)}$
  - where $count(u, v)$ = #instances where target word u was aligned to source word v in the training set
- However, word-to-word alignments are often hard to come by

What can we do?

EM for Model 1

- **(E-Step)** If we had an accurate translation model, we can estimate likelihood of each alignment as:

$$q_m(a_m \mid \boldsymbol{w}^{(s)}, \boldsymbol{w}^{(t)}) \propto \mathrm{p}(a_m \mid m, M^{(s)}, M^{(t)}) \times \mathrm{p}(w_m^{(s)} \mid w_{a_m}^{(t)}),$$

Remember these are fixed

- **(M Step)** Use expected count to re-estimate translation parameters:

$$p(v \mid u) = \frac{E_q[count(u, v)]}{count(u)}$$

$$E_q\left[\mathrm{count}(u, v)\right] = \sum_m q_m(a_m \mid \boldsymbol{w}^{(s)}, \boldsymbol{w}^{(t)}) \times \delta(w_m^{(s)} = v) \times \delta(w_{a_m}^{(t)} = u).$$

How do we translate?

- We want: $\arg\max_{w^{(t)}} p(w^{(t)} \mid w^{(s)}) = \arg\max_{w^{(t)}} \frac{p(w^{(s)}, w^{(t)})}{p(w^{(s)})}$

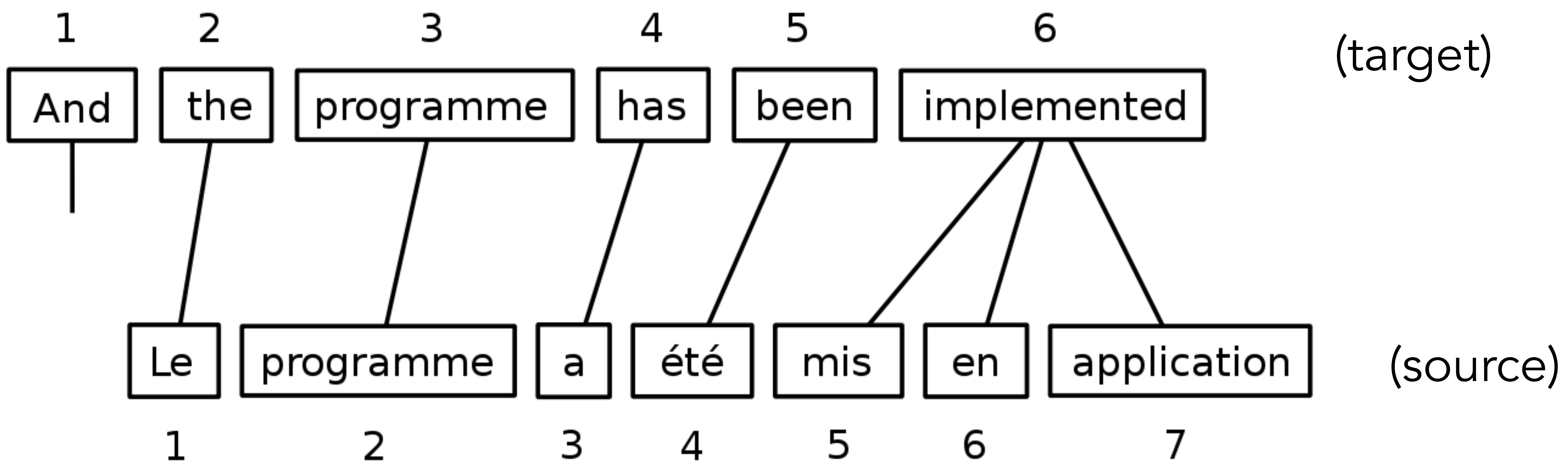
- Sum over all possible alignments:

$$\begin{aligned} \mathrm{p}(\boldsymbol{w}^{(s)}, \boldsymbol{w}^{(t)}) = & \sum_{\mathcal{A}} \mathrm{p}(\boldsymbol{w}^{(s)}, \boldsymbol{w}^{(t)}, \mathcal{A}) \\ = & \mathrm{p}(\boldsymbol{w}^{(t)}) \sum_{\mathcal{A}} \mathrm{p}(\mathcal{A}) \times \mathrm{p}(\boldsymbol{w}^{(s)} \mid \boldsymbol{w}^{(t)}, \mathcal{A}) \end{aligned}$$

- Alternatively, take the max over alignments

- Decoding: Greedy/beam search

Model 1: Decoding



At every step m, pick target word $w_m^{(t)}$ to maximize product of:

1. Language model: $p_{LM}(w_m^{(t)} \mid w_{<m}^{(t)})$
2. Translation model: $p(w_{b_m}^{(s)} \mid w_m^{(t)})$

where b_m is the inverse alignment from target to source

IBM Model 1

- Assume $p(a_m \mid m, M^{(s)}, M^{(t)}) = \frac{1}{M^{(t)}}$
- Each source word is aligned to at most one target word
- We then have:

$$p(w^{(s)}, w^{(t)}) = p(w^{(t)}) \sum_{A} (\frac{1}{M^{(t)}})^{M^{(s)}} \; p(w^{(s)} \mid w^{(t)})$$

Restrictive assumptions

IBM Model 2

- Slightly relaxed assumption:
  - $p(a_m | m, M^{(s)}, M^{(t)})$ is also estimated/learned, not set to constant
- Some independence assumptions from Model 1 still required:
  - Alignment probability factors across tokens:

$$\mathrm{p}(\mathcal{A} \mid \boldsymbol{w}^{(s)}, \boldsymbol{w}^{(t)}) = \prod_{m=1}^{M^{(s)}} \mathrm{p}(a_m \mid m, M^{(s)}, M^{(t)}).$$

  - Translation probability factors across tokens:

$$\mathrm{p}(\boldsymbol{w}^{(s)} \mid \boldsymbol{w}^{(t)}, \mathcal{A}) = \prod_{m=1}^{M^{(s)}} \mathrm{p}(w_m^{(s)} \mid w_{a_m}^{(t)}),$$

Other IBM models

Model 1: lexical translation

Model 2: additional absolute alignment model

Model 3: extra fertility model

Model 4: added relative alignment model

Model 5: fixed deficiency problem.

Model 6: Model 4 combined with a HMM alignment model in a log linear way

- Models 3 - 6 make successively weaker assumptions
  - But get progressively harder to optimize
- Simpler models are often used to ‘initialize’ complex ones
  - e.g train Model 1 and use it to initialize Model 2 translation parameters

Phrase-based MT

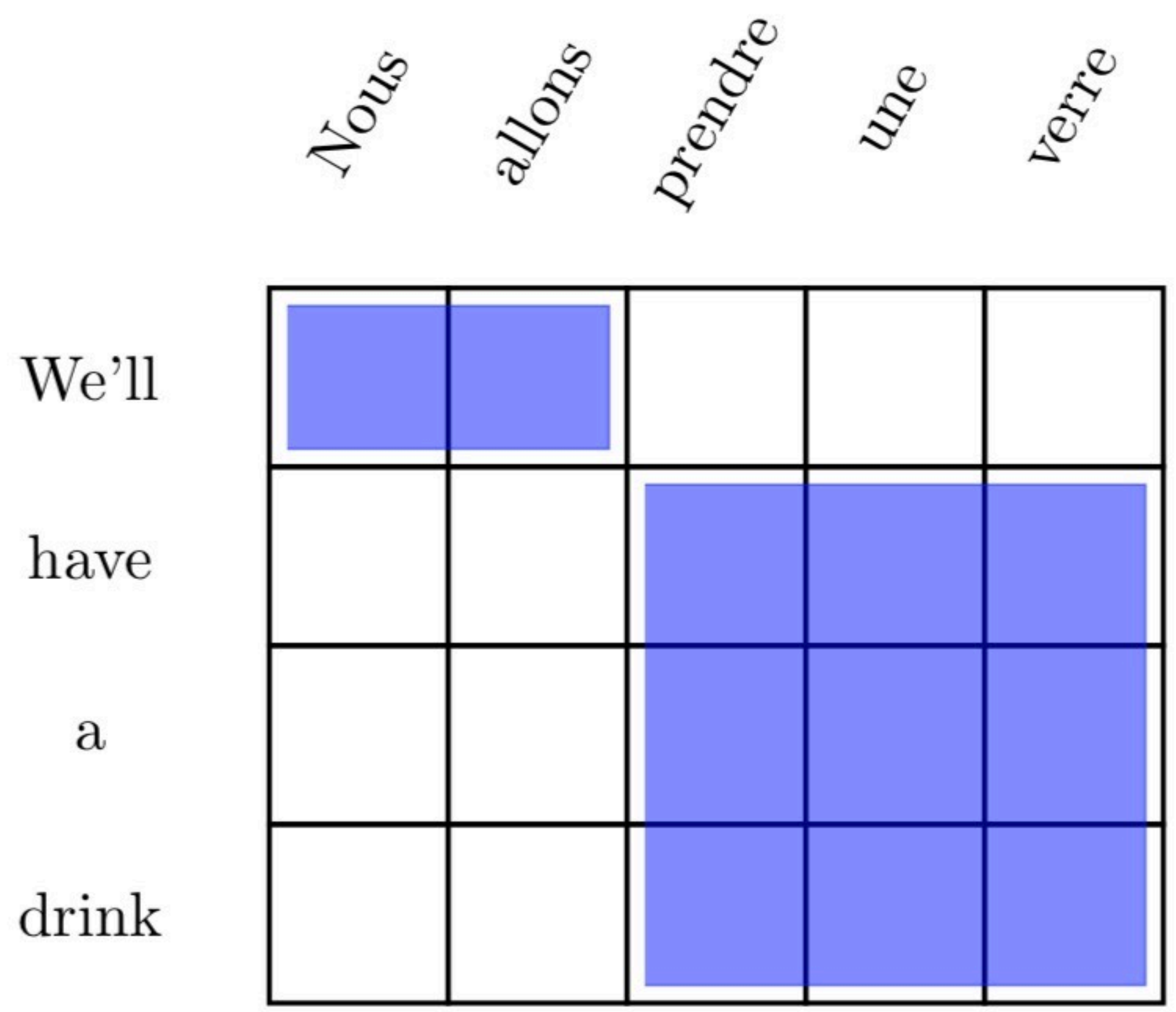
- Word-by-word translation is not sufficient in many cases

Nous allons prendre un verre

(literal) We will take a glass

(actual) We'll have a drink

- Solution: build alignments and translation tables between multiword spans or "phrases"

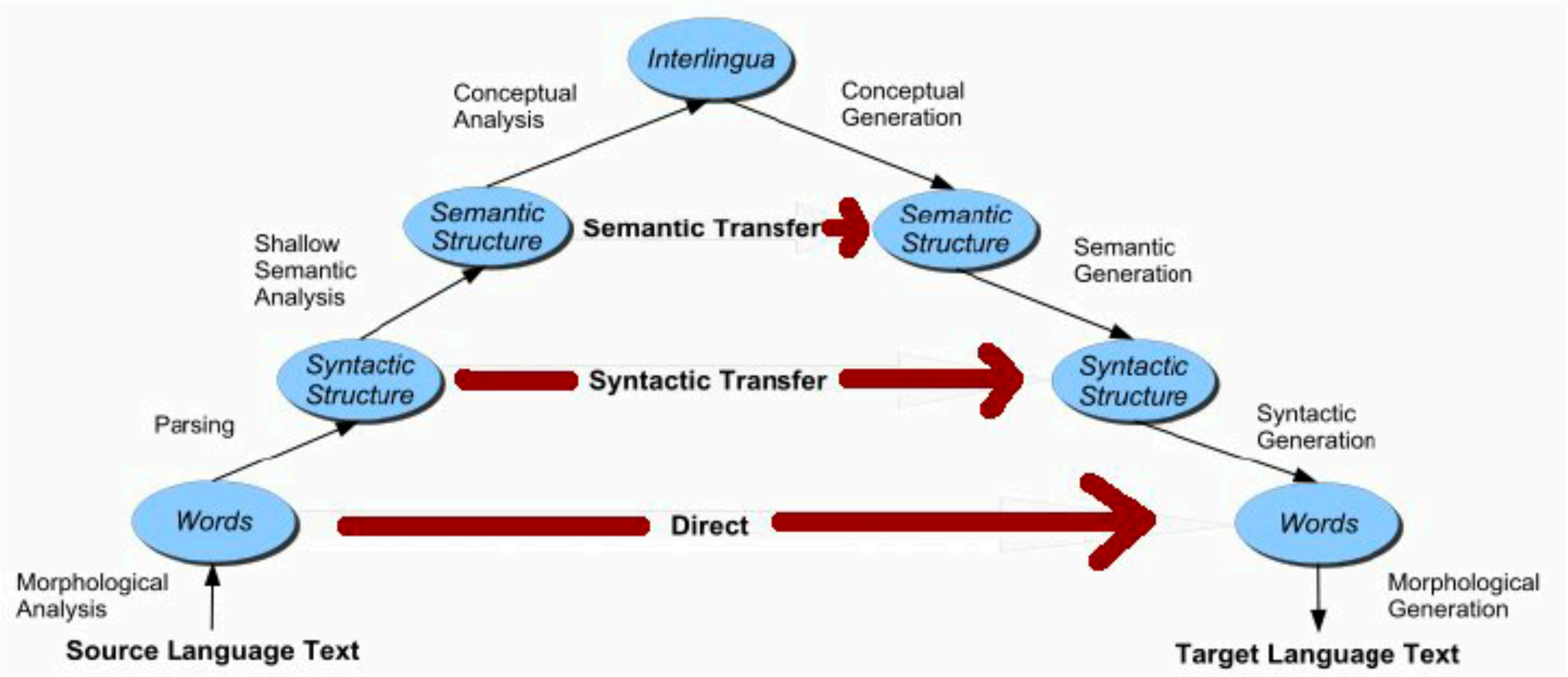


Phrase-based MT

- Solution: build alignments and translation tables between multiword spans or "phrases"
- Translations condition on multi-word units and assign probabilities to multi-word units
- Alignments map from spans to spans

$$\mathrm{p}(\boldsymbol{w}^{(s)} \mid \boldsymbol{w}^{(t)}, \mathcal{A}) = \prod_{((i,j),(k,\ell)) \in \mathcal{A}} \mathrm{p}_{w^{(s)}|w^{(t)}}(\{w_{i+1}^{(s)}, w_{i+2}^{(s)}, \ldots, w_{j}^{(s)}\} \mid \{w_{k+1}^{(t)}, w_{k+2}^{(t)}, \ldots, w_{\ell}^{(t)}\})$$

Vauquois Pyramid



- Hierarchy of concepts and distances between them in different languages
- Lowest level: individual words/ characters
- Higher levels: syntax, semantics
- Interlingua: Generic language-agnostic representation of meaning

Syntactic MT

- Rather than use phrases, use a *synchronous context-free grammar*: constructs "parallel" trees in two languages simultaneously

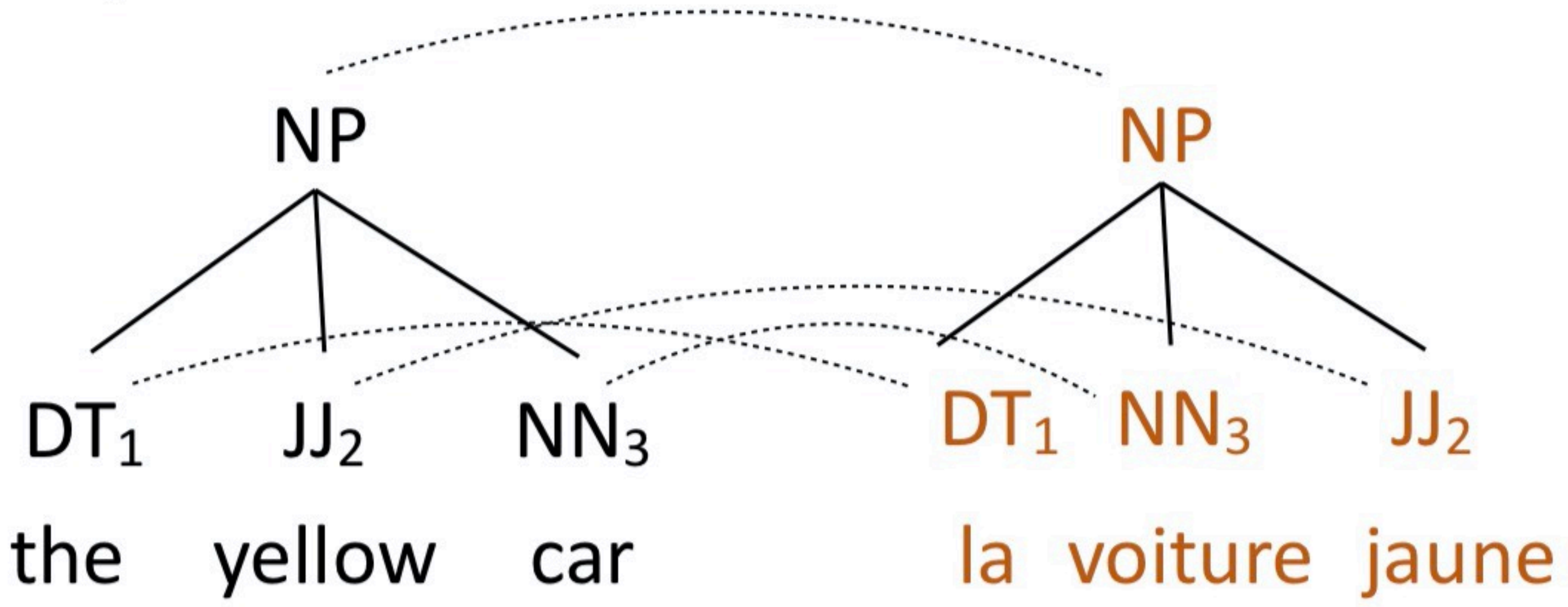
NP → [DT_1 JJ_2 NN_3; DT_1 NN_3 JJ_2]

DT → [the, la]

DT → [the, le]

NN → [car, voiture]

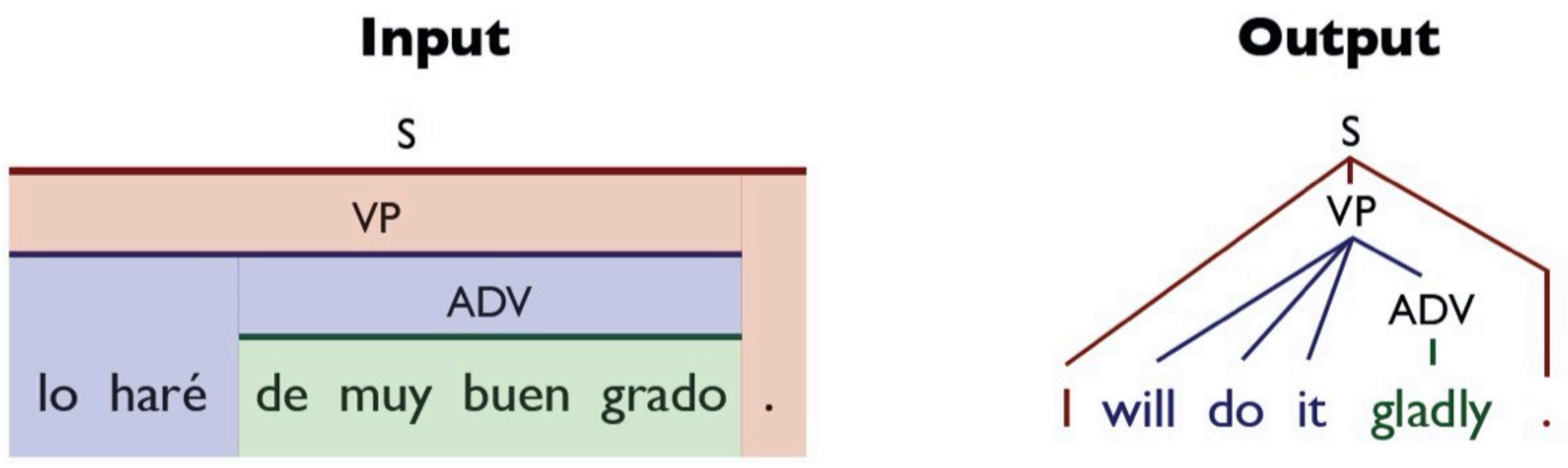
JJ → [yellow, jaune]



- Assumes parallel syntax up to reordering
- Translation = parse the input with "half" the grammar, read off other half

(*Slide credit: Greg Durrett*)

Syntactic MT



- Relax this by using lexicalized rules, like "syntactic phrases"
- Leads to HUGE grammars, parsing is slow

Grammar

S → ⟨ VP . ; I VP . ⟩ **OR** S → ⟨ VP . ; you VP . ⟩

VP → ⟨ lo haré ADV ; will do it ADV ⟩

S → ⟨ lo haré ADV . ; I will do it ADV . ⟩

ADV → ⟨ de muy buen grado ; gladly ⟩

Slide credit: Dan Klein

Next time: Neural machine translation