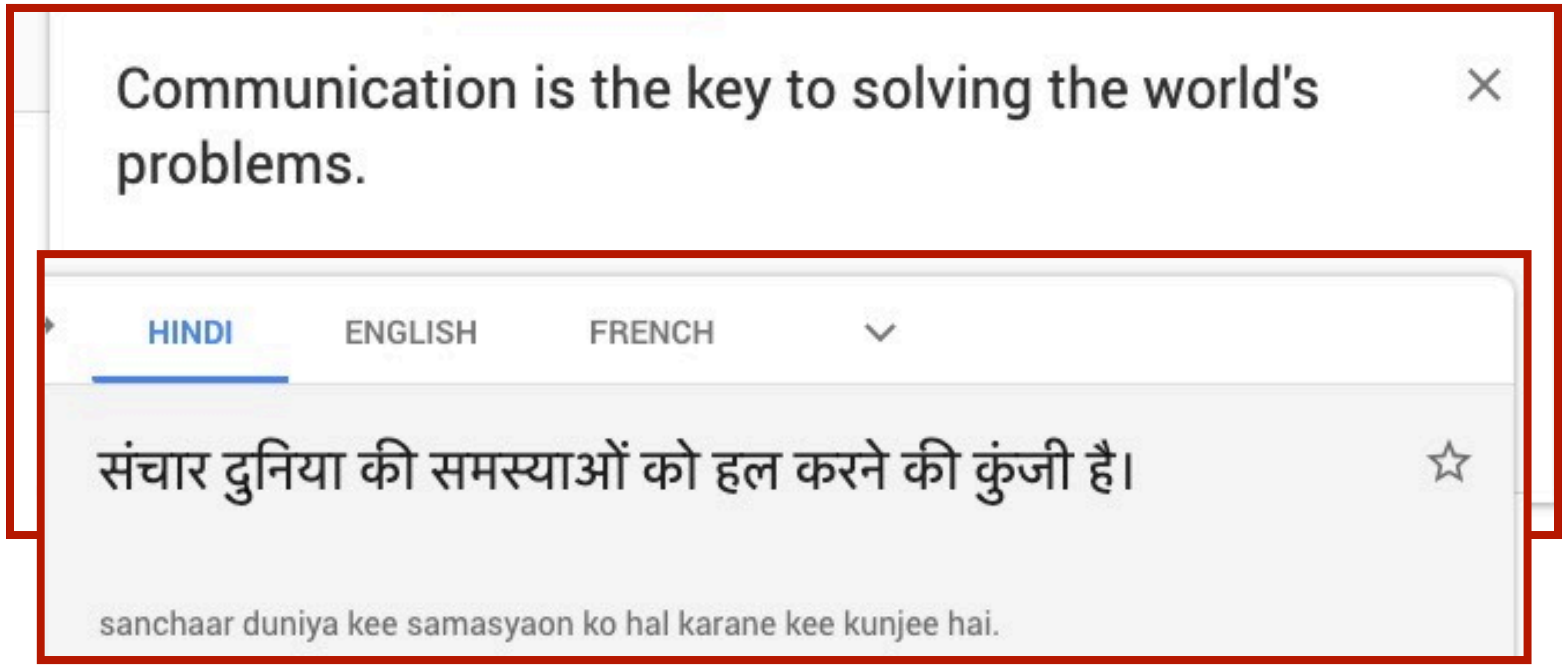
COS 484

L11: Machine Translation

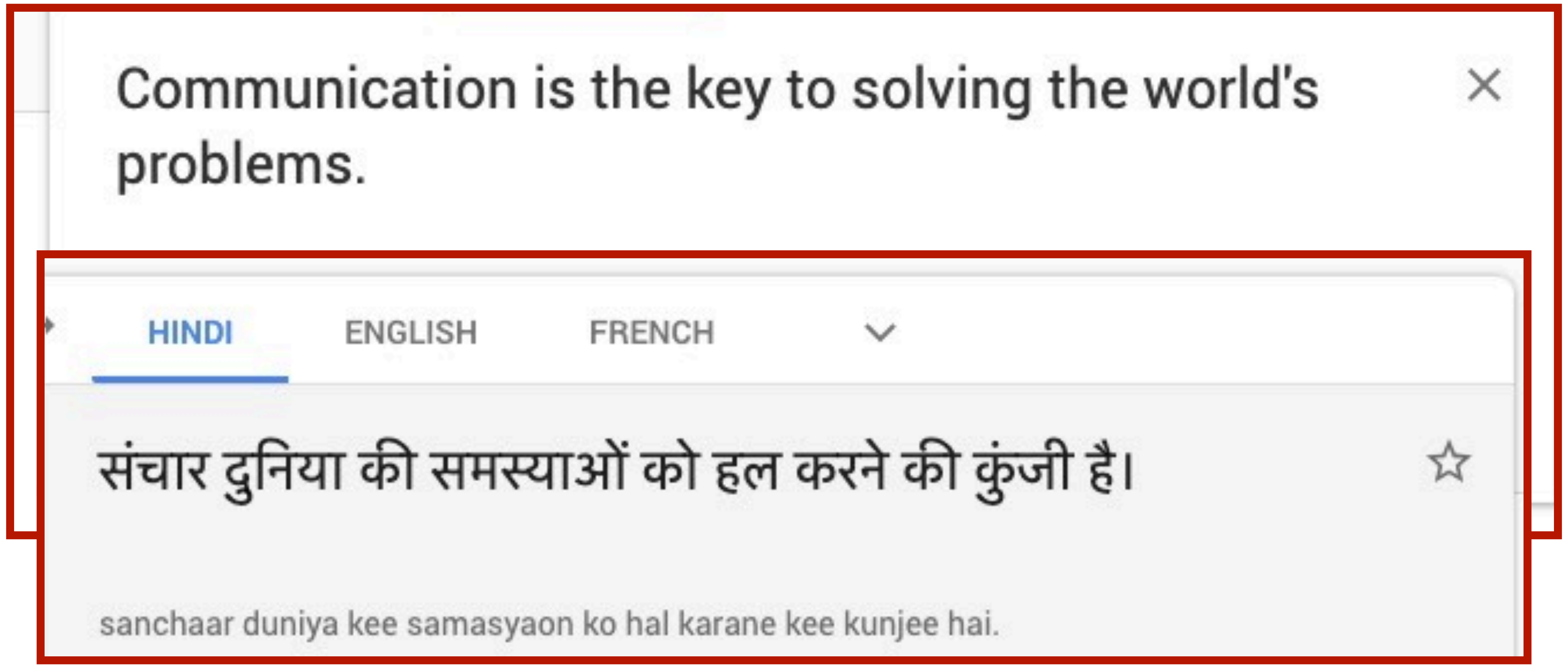
Spring 2024

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
D) 4+

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

Source

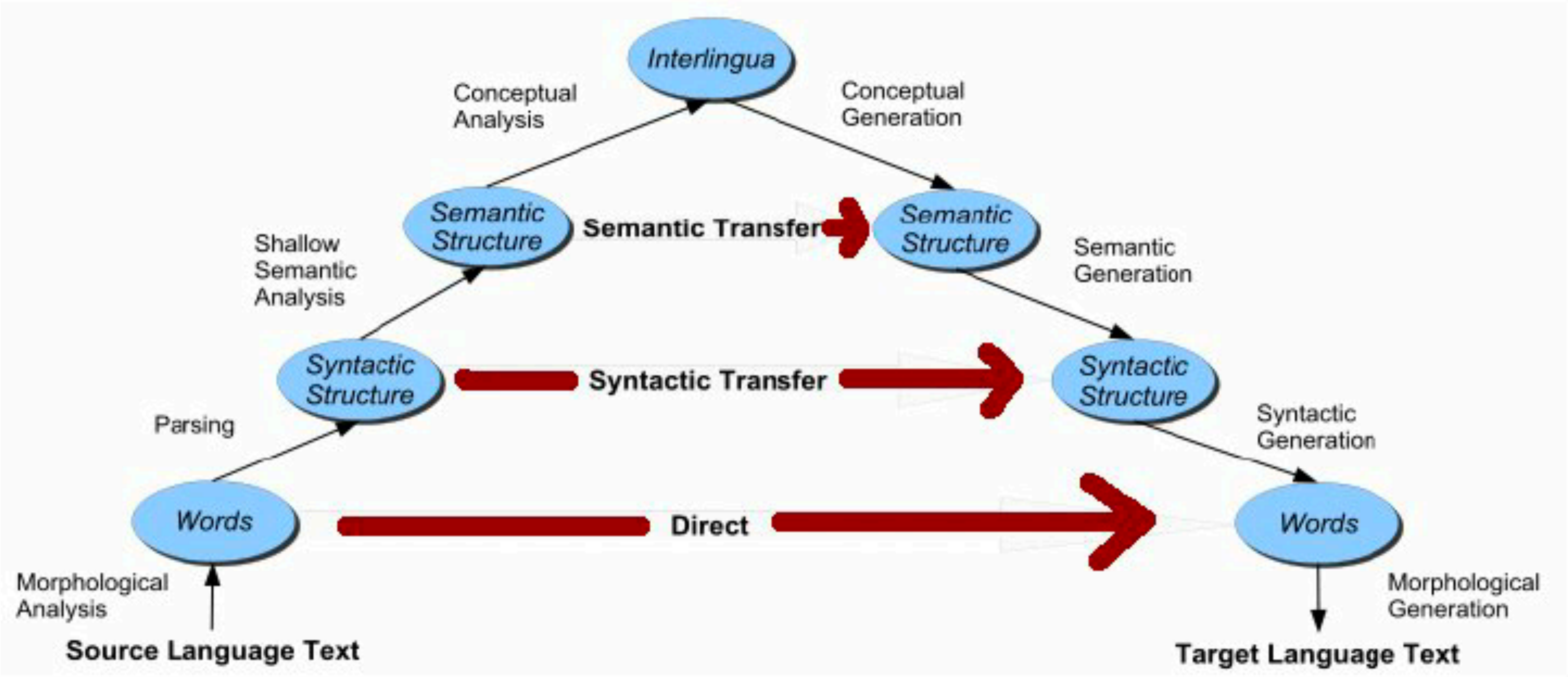
Target

Why is MT challenging?

- Single words may be replaced with multi-word phrases
    - I like apples $\leftrightarrow$ J'aime les pommes
- Reordering of phrases
    - I like red apples $\leftrightarrow$ J'aime les pommes rouges
- Contextual dependence
    - *les* $\leftrightarrow$ *the* but *les pommes* $\leftrightarrow$ *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 $\mathbf{w}^{(t)}$ should adequately reflect the linguistic content of $\mathbf{w}^{(s)}$
  - Fluency: Translation $\mathbf{w}^{(t)}$ 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

D) none of them

Evaluating machine translation

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

	Adequate?	Fluent?
To Vinay it like Python	yes	no
Vinay debugs memory leaks	no	yes
Vinay likes Python	yes	yes

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

Which of these translations is both adequate and fluent?

A) first

B) second

C) third

D) none of them

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

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

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

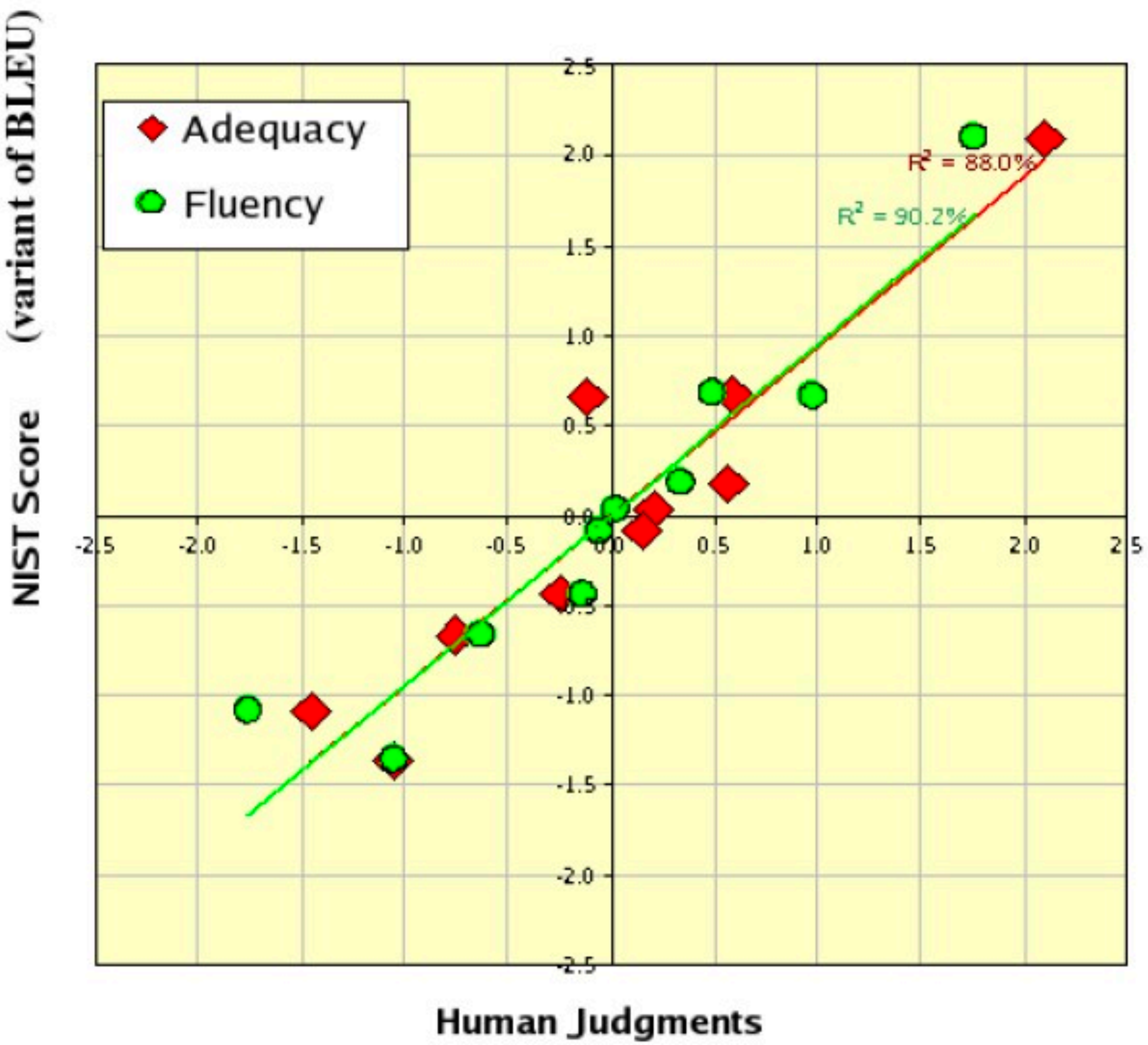
- To avoid $\log 0$, all precisions are smoothed
- Each n-gram in reference can be used at most once
  - Ex. **Hypothesis**: *to to to to to* vs **Reference**: *to be or not to be* should not get a unigram precision of 1
- BLEU-k: average of BLEU scores computed using 1-gram through k-gram.

Problem: 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

Machine translation: Data

21 European languages: Romanic (French, Italian, Spanish, Portuguese, Romanian), Germanic (English, Dutch, German, Danish, Swedish), Slavik (Bulgarian, Czech, Polish, Slovak, Slovene), Finni-Ugric (Finnish, Hungarian, Estonian), Baltic (Latvian, Lithuanian), and Greek.

Parallel Corpus (L1-L2)	Sentences	L1 Words	English Words
Bulgarian-English	406,934	-	9,886,291
Czech-English	646,605	12,999,455	15,625,264
Danish-English	1,968,800	44,654,417	48,574,988
German-English	1,920,209	44,548,491	47,818,827
Greek-English	1,235,976	-	31,929,703
Spanish-English	1,965,734	51,575,748	49,093,806
Estonian-English	651,746	11,214,221	15,685,733
Finnish-English	1,924,942	32,266,343	47,460,063
French-English	2,007,723	51,388,643	50,196,035

https://www.statmt.org/europarl/

Statistical machine translation (SMT)

- Core idea: Learn a probabilistic model from data
- Suppose we are translating French → English
- We want to find **best target sentence** $\mathbf{w}^{(t)}$, given **source sentence** $\mathbf{w}^{(s)}$

$$\arg\max_{\mathbf{w}^{(t)}} P(\mathbf{w}^{(t)} \mid \mathbf{w}^{(s)})$$

- According to Bayes' rule, we can break this down into two components:

$$= \arg\max_{\mathbf{w}^{(t)}} P(\mathbf{w}^{(\mathbf{s})} \mid \mathbf{w}^{(\mathbf{t})}) P(\mathbf{w}^{(t)})$$

Translation model: models whether the target sentence reflects the linguistic content of the source language (adequacy)
Learned from **parallel** data

Language model: models how fluent the target sentence is (fluency)

Can be learned from **monolingual** data

Noisy channel model

$p_T \longrightarrow$ **Target sentence** $\xrightarrow{\quad p_{S|T} \quad}$ **Source sentence**

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

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

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

- 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

$p_T \longrightarrow$ Target sentence $\xrightarrow{\;p_{S|T}\;}$ Source sentence

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

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

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

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

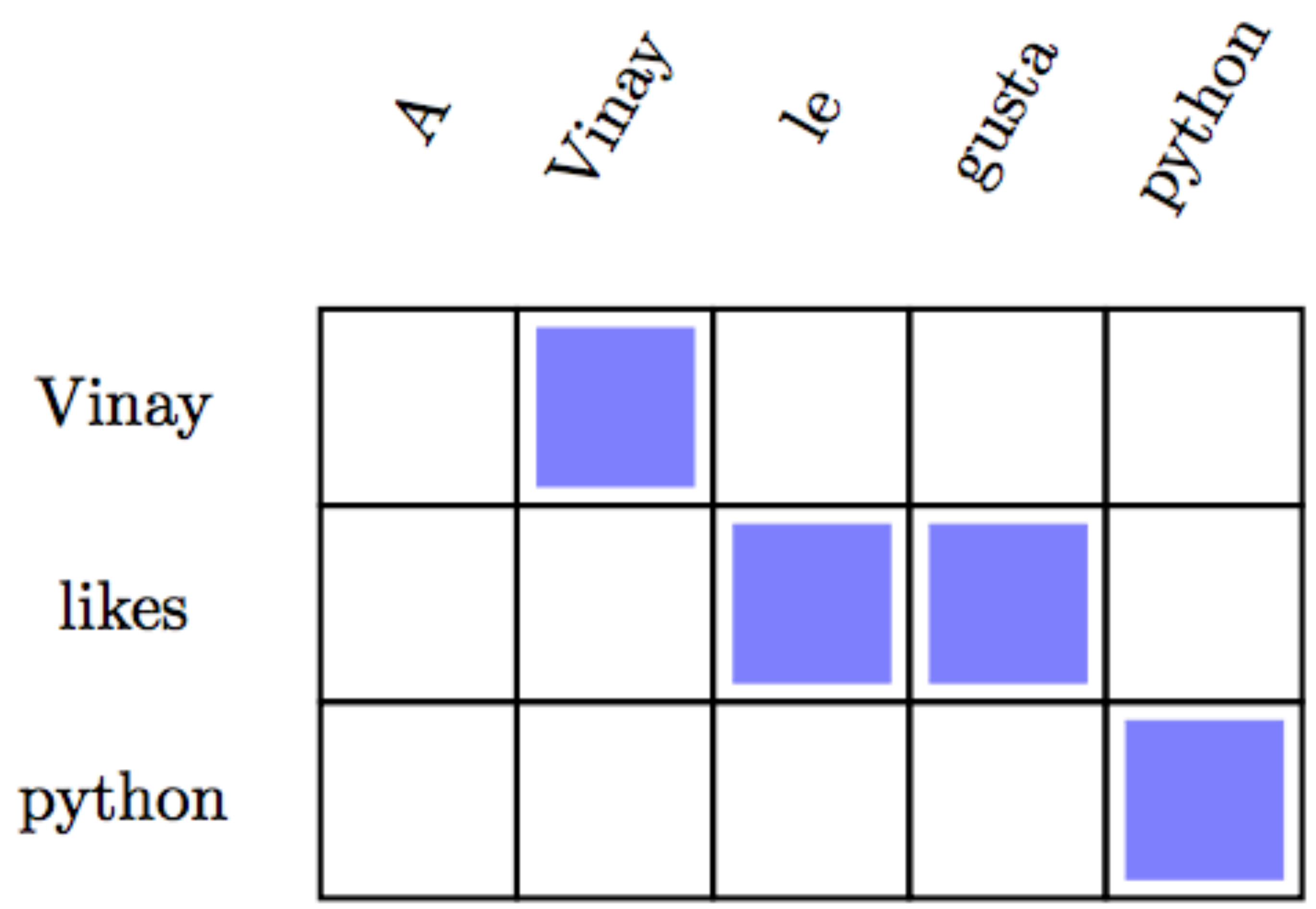
- Use Bayes rule to recover $w^{(t)}$ that is maximally likely under the conditional distribution $p_{T|S}$ (which is what we want)

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:

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

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

- $M^{(s)}, M^{(t)}$ 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

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

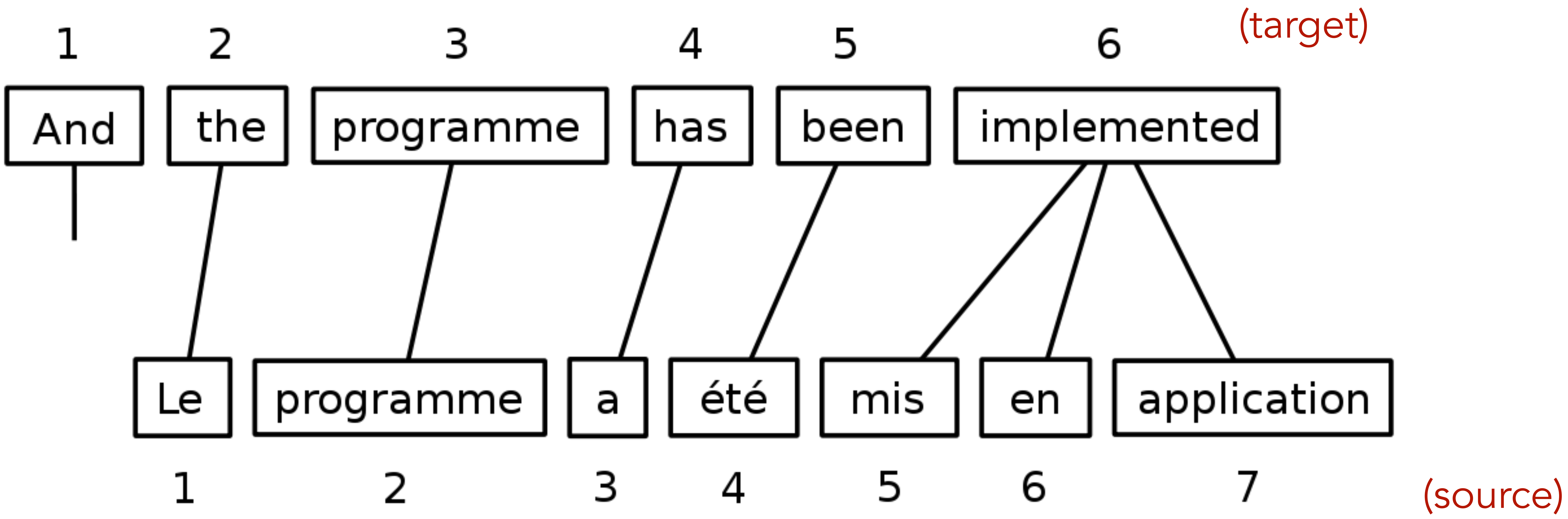
- Two independence assumptions:
  - Alignment probability factors across tokens:

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

  - Translation probability factors across tokens:

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

Limitations

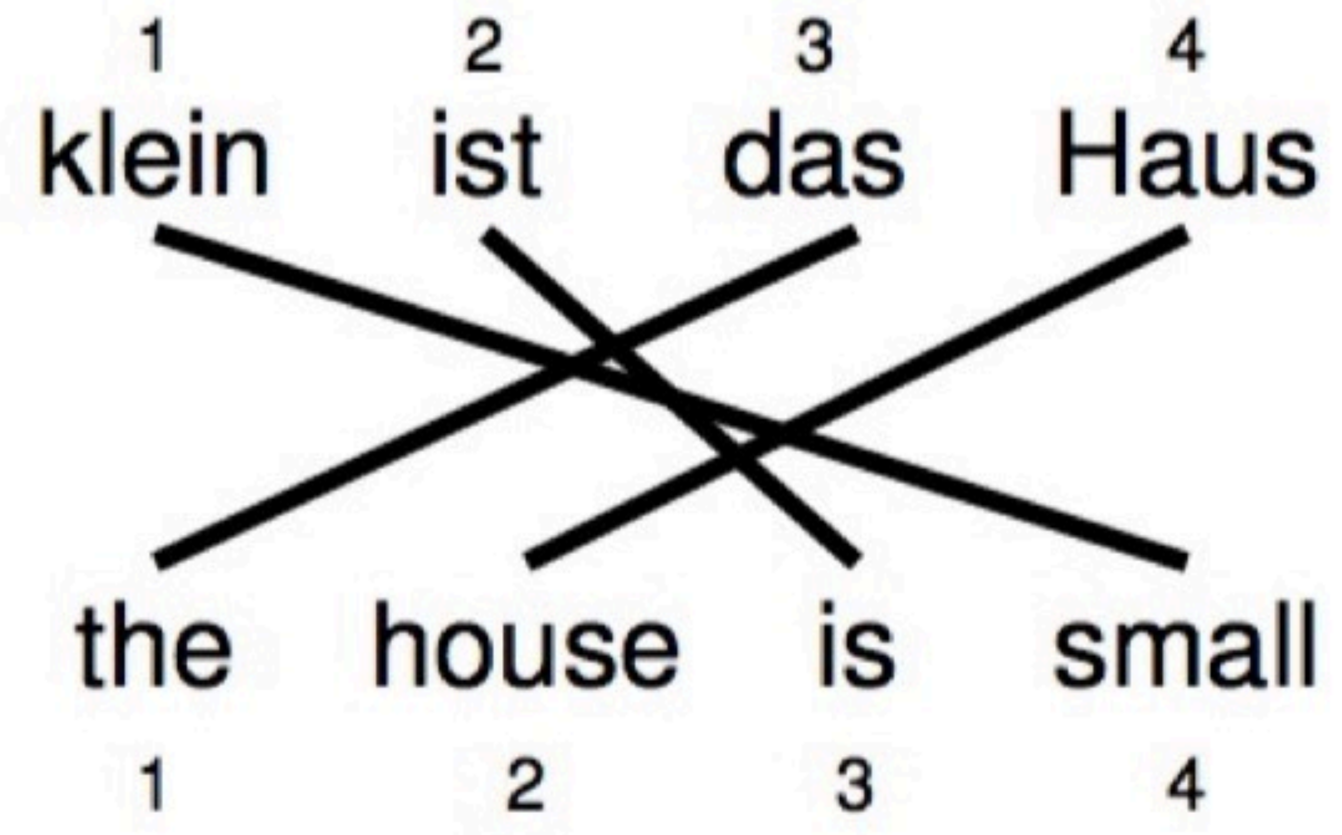


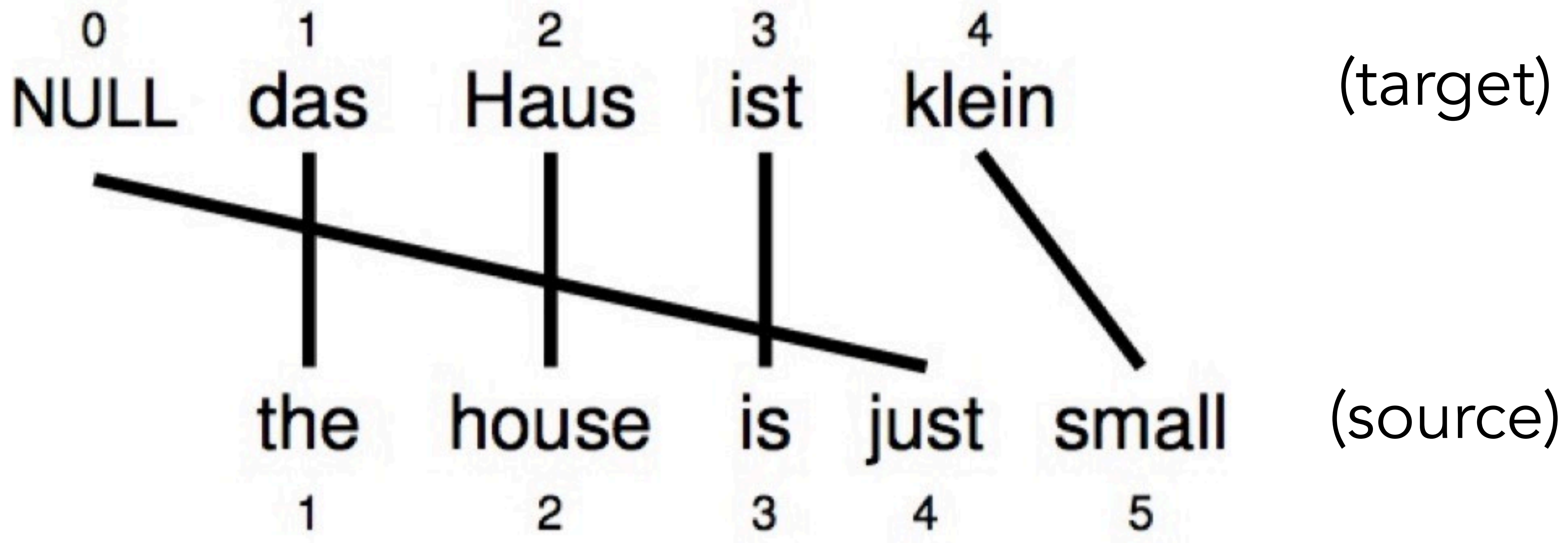
$$a_1 = 2,\ a_2 = 3,\ a_3 = 4, \ldots$$

Multiple source words may align to the same target word!

Or a source word may not have any corresponding target.

Reordering and word insertion





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

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

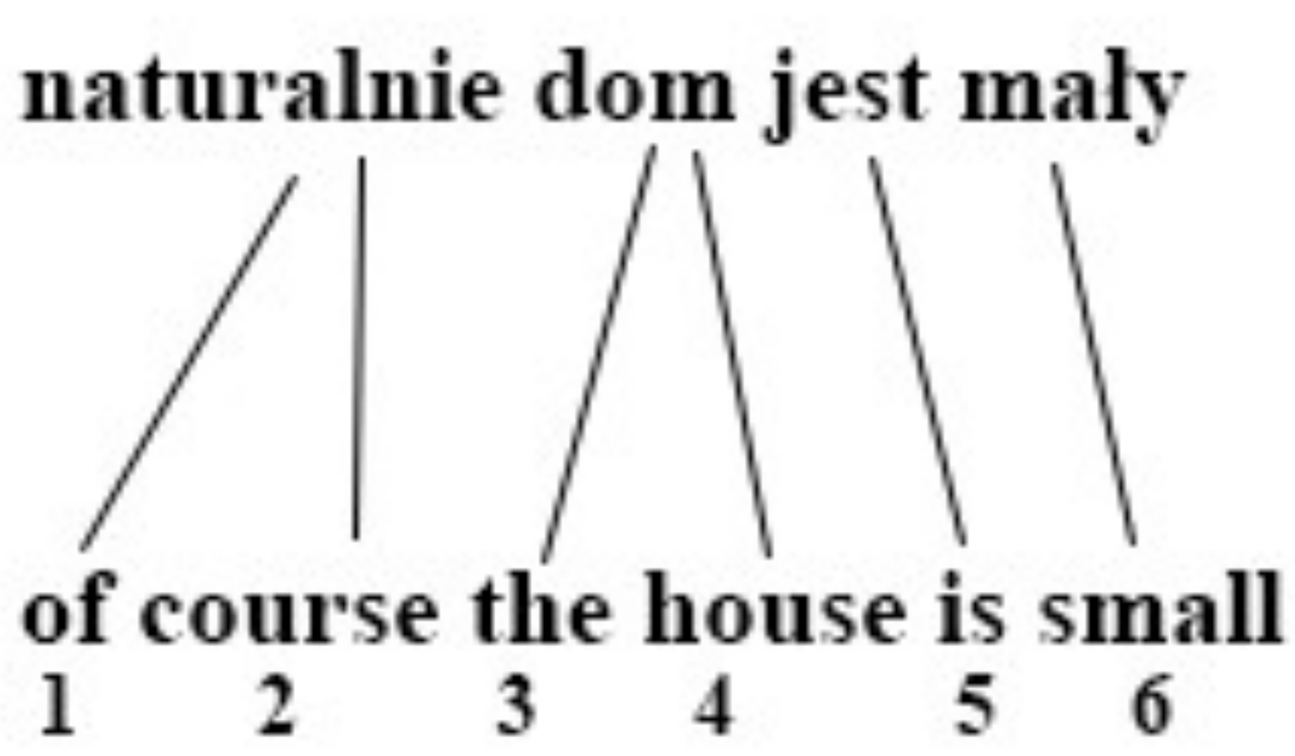
Assume extra NULL token

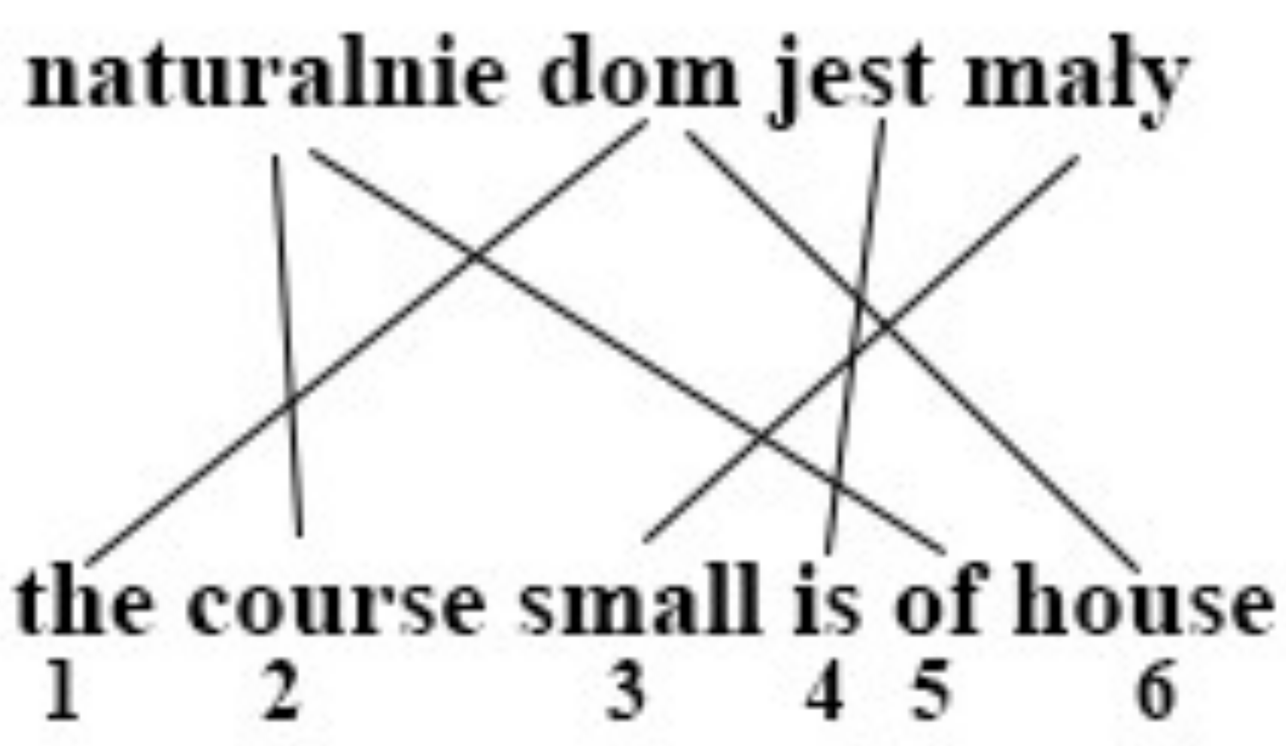
(Slide credit: Brendan O'Connor)

IBM Model 1

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

- We then have (for each pair of words in source and target):

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

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

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

Solution: Unsupervised learning

Expectation Maximization (advanced)

- **(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:

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

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

How do we translate?

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

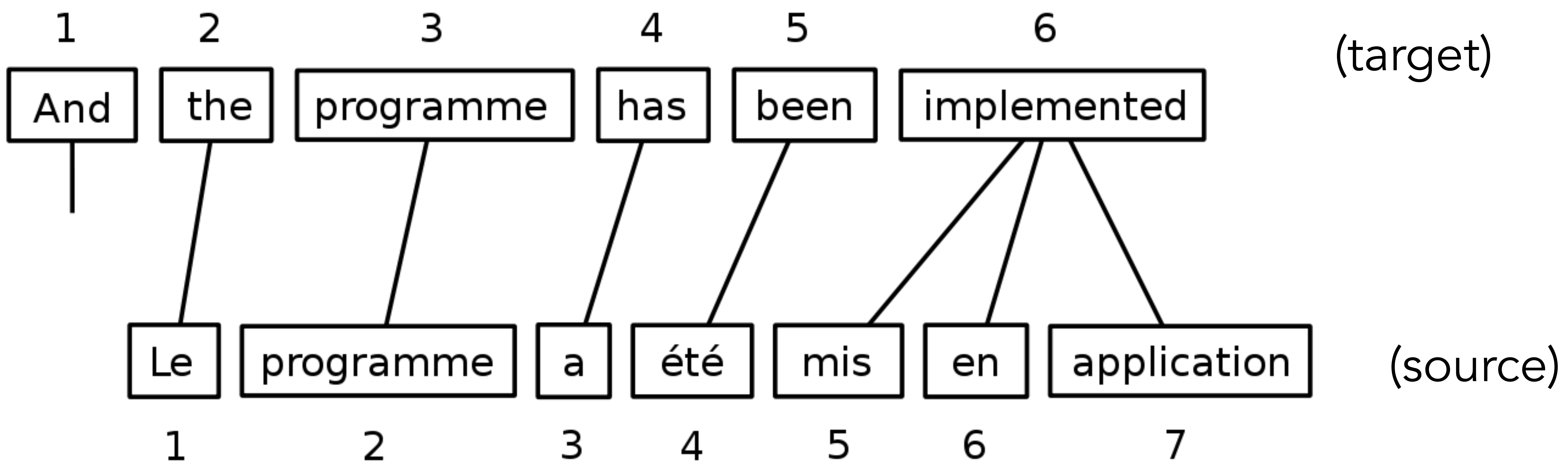
- Sum over all possible alignments:

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

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

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

Restrictive assumptions

Other IBM models

Model 1: lexical translation

Model 2: additional absolute alignment model

Model 3: extra fertility model

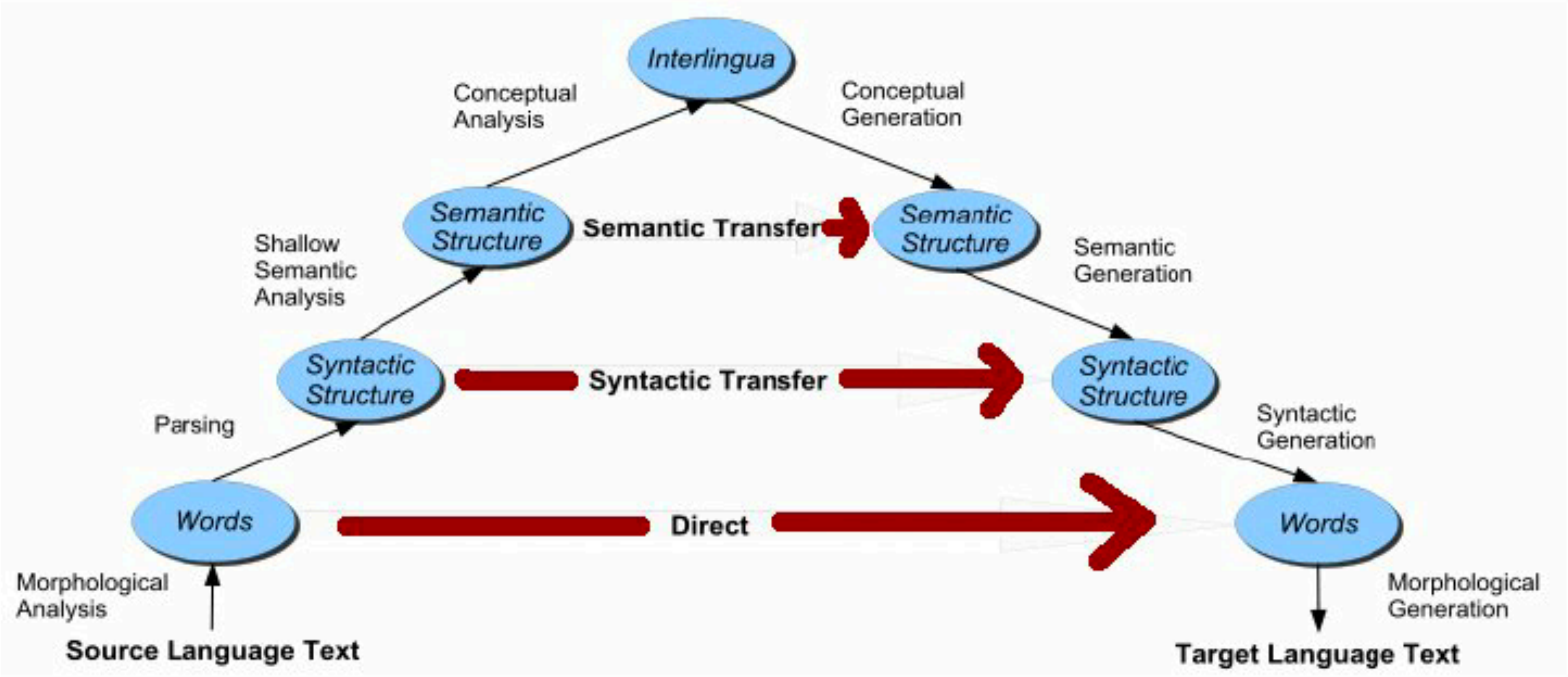
Model 4: added relative alignment model

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

Vauquois Pyramid

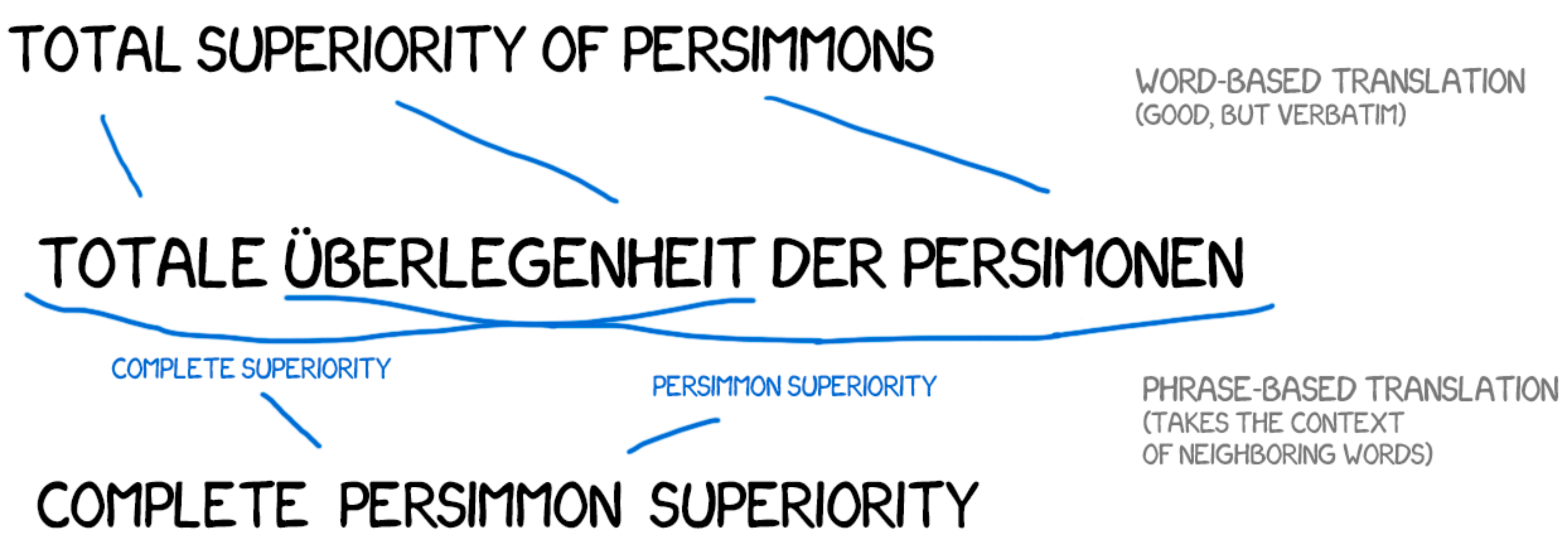


- Hierarchy of concepts and distances between them in different languages
- Lowest level: individual words/ characters
- Higher levels: syntax, semantics
- Interlingua: Generic language-agnostic representation of meaning

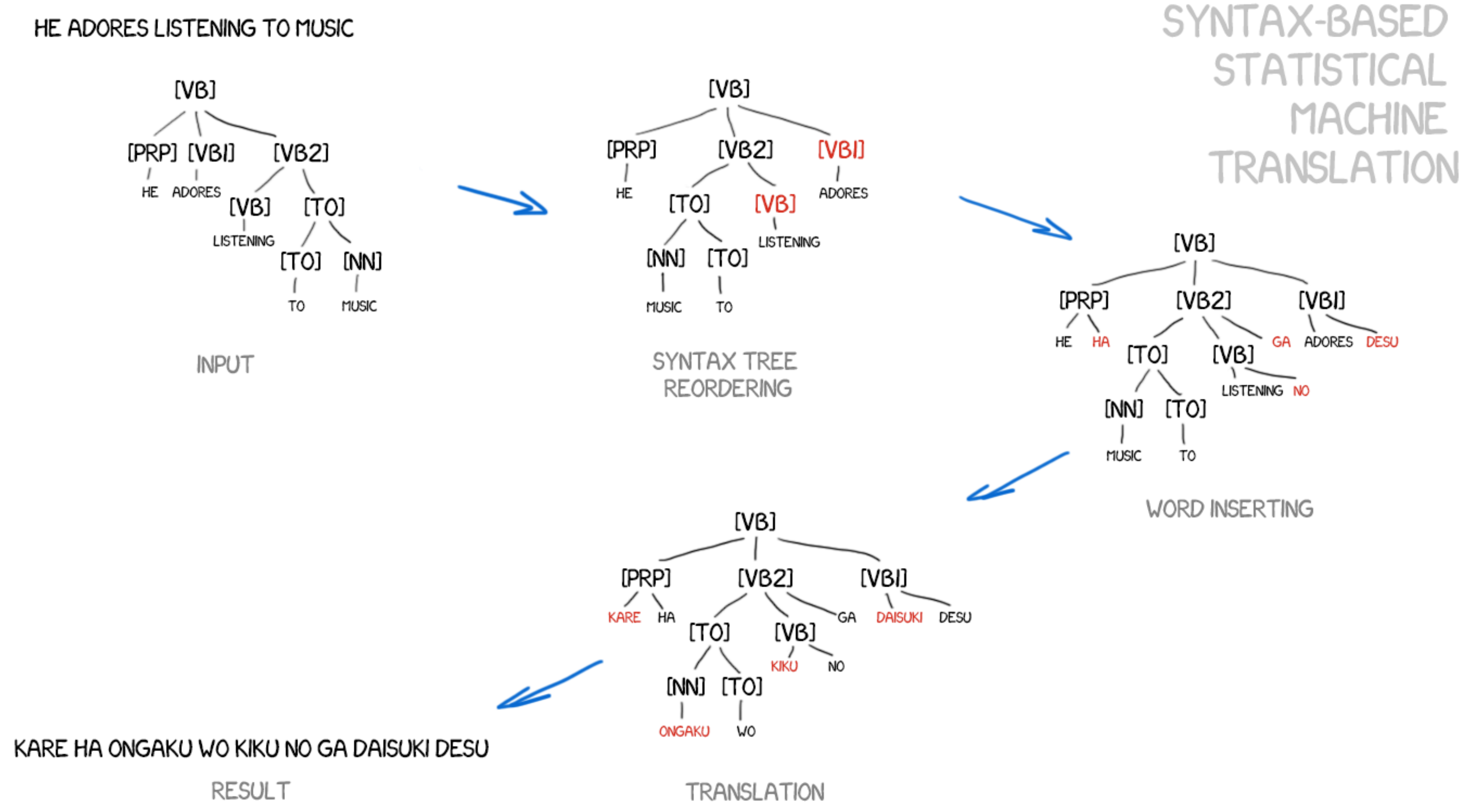
Statistical machine translation (SMT)

- SMT was a huge field (1990s-2010s) - The best systems were **extremely complex**
- Systems had many separately-designed subcomponents
  - Need to **design features** to capture particular language phenomena
  - Required compiling and maintaining **extra resources**
  - Lots of **human effort** to maintain - repeated effort for each language pair!

Phrase-based SMT



Syntax-based SMT



https://translartisan.wordpress.com/tag/statistical-machine-translation/

SMT $\longrightarrow$ NMT

Q. Do you know when Google Translate was first launched?

Launched in April 2006 as a statistical machine translation service, it used United Nations and European Parliament documents and transcripts to gather linguistic data. Rather than translating languages directly, it first translates text to English and then pivots to the target language in most of the language combinations it posits in its grid,[7] with a few exceptions including Catalan-Spanish.[8] During a translation, it looks for patterns in millions of documents to help decide which words to choose and how to arrange them in the target language. Its accuracy, which has been criticized on several occasions,[9] has been measured to vary greatly across languages.[10] In November 2016, Google announced that Google Translate would switch to a neural machine translation engine – Google Neural Machine Translation (GNMT) – which translates "whole sentences at a time,

Google’s NMT system in 2016

RESEARCH › PUBLICATIONS ›

Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation

Table 10: Mean of side-by-side scores on production data

	PBMT	GNMT	Human	Relative Improvement
English → Spanish	4.885	5.428	5.504	87%
English → French	4.932	5.295	5.496	64%
English → Chinese	4.035	4.594	4.987	58%
Spanish → English	4.872	5.187	5.372	63%
French → English	5.046	5.343	5.404	83%
Chinese → English	3.694	4.263	4.636	60%

(Wu et al., 2016): Google’s Neural Machine Translation System: Bridging the Gap between Human and Machine Translation

SMT ⟶ NMT

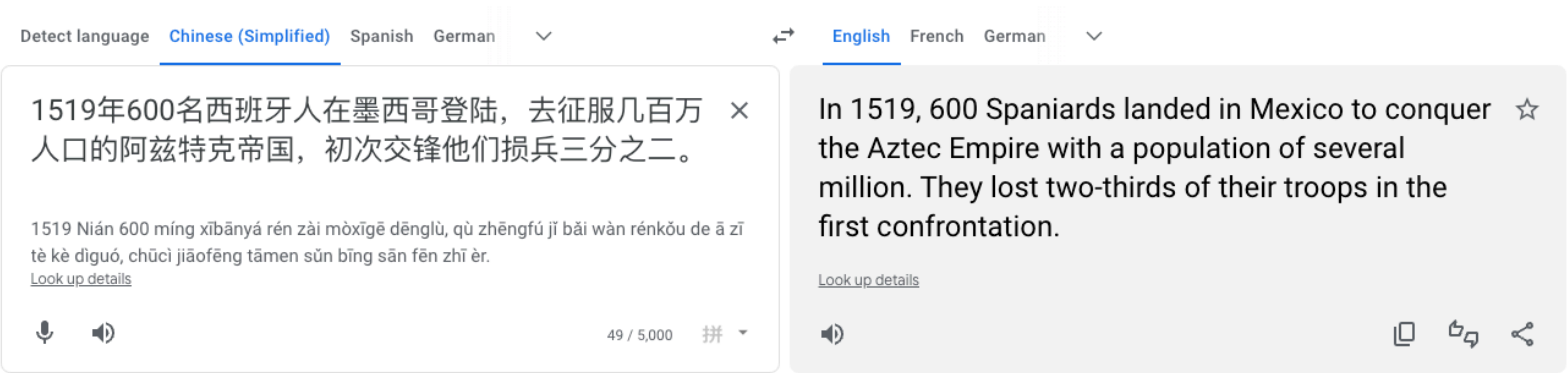
1519年600名西班牙人在墨西哥登陆，去征服几百万人口的阿兹特克帝国，初次交锋他们损兵三分之二。

In 1519, six hundred Spaniards landed in Mexico to conquer the Aztec Empire with a population of a few million. They lost two thirds of their soldiers in the first clash.

translate.google.com (2009): 1519 600 Spaniards landed in Mexico, millions of people to conquer the Aztec empire, the first two-thirds of soldiers against their loss.

translate.google.com (2013): 1519 600 Spaniards landed in Mexico to conquer the Aztec empire, hundreds of millions of people, the initial confrontation loss of soldiers two-thirds.

translate.google.com (2015): 1519 600 Spaniards landed in Mexico, millions of people to conquer the Aztec empire, the first two-thirds of the loss of soldiers they clash.



Neural machine translation (NMT)

- Neural Machine Translation (NMT) is a way to do machine translation with a **single end-to-end neural network**

- The neural network architecture is called a **sequence-to-sequence model** (aka **seq2seq**) and it involves two RNNs

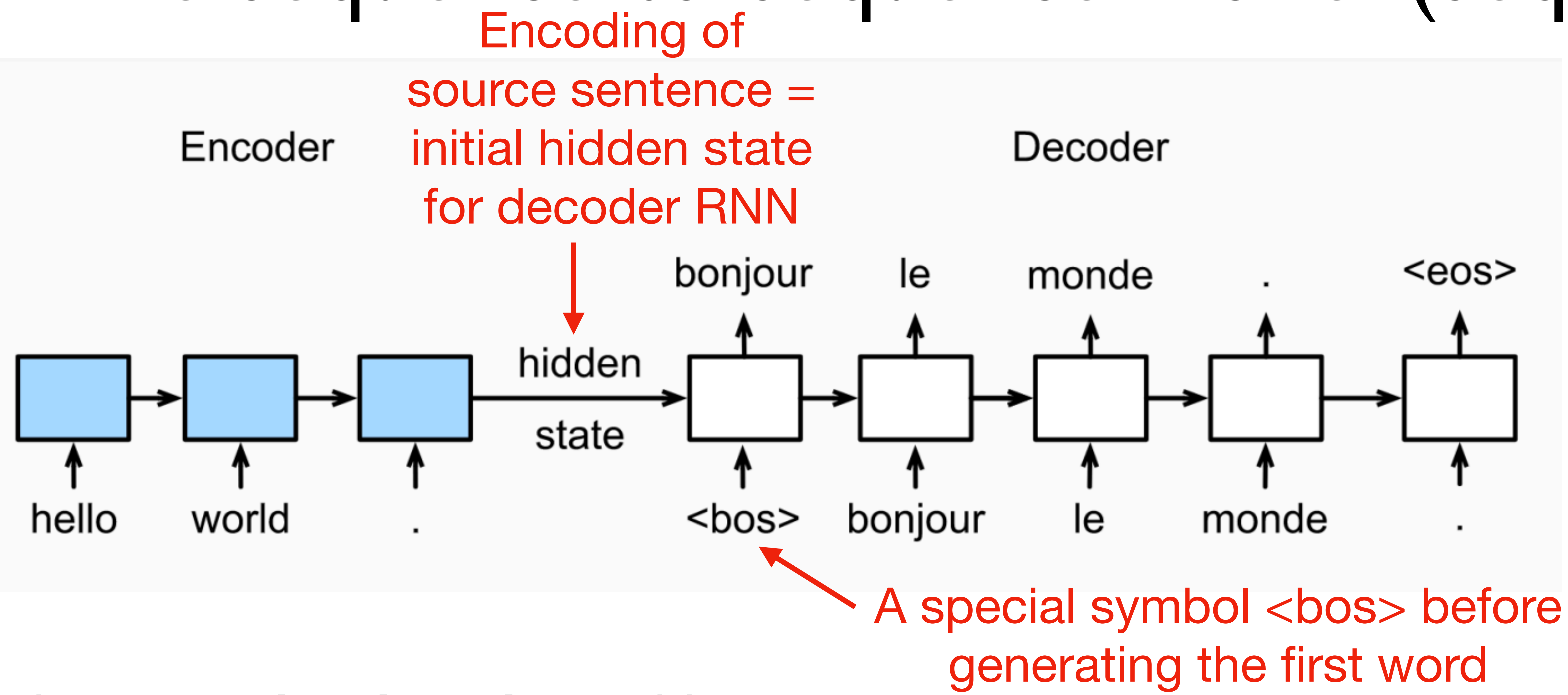
Sequence to Sequence Learning with Neural Networks

Ilya Sutskever	**Oriol Vinyals**	**Quoc V. Le**
Google	Google	Google
ilyasu@google.com	vinyals@google.com	qvl@google.com

(Sutskever et al., 2014)

Ilya Sutskever

The sequence-to-sequence model (seq2seq)



It is called an **encoder-decoder** architecture

- The encoder is an RNN to read the input sequence (source language)
- The decoder is another RNN to generate output word by word (target language)

Image: https://d2l.ai/chapter_recurrent-modern/seq2seq.html