Sequence to Sequence Modeling

CS 288 Spring 2026
UC Berkeley
cal-cs288.github.io/sp26

Berkeley EECS

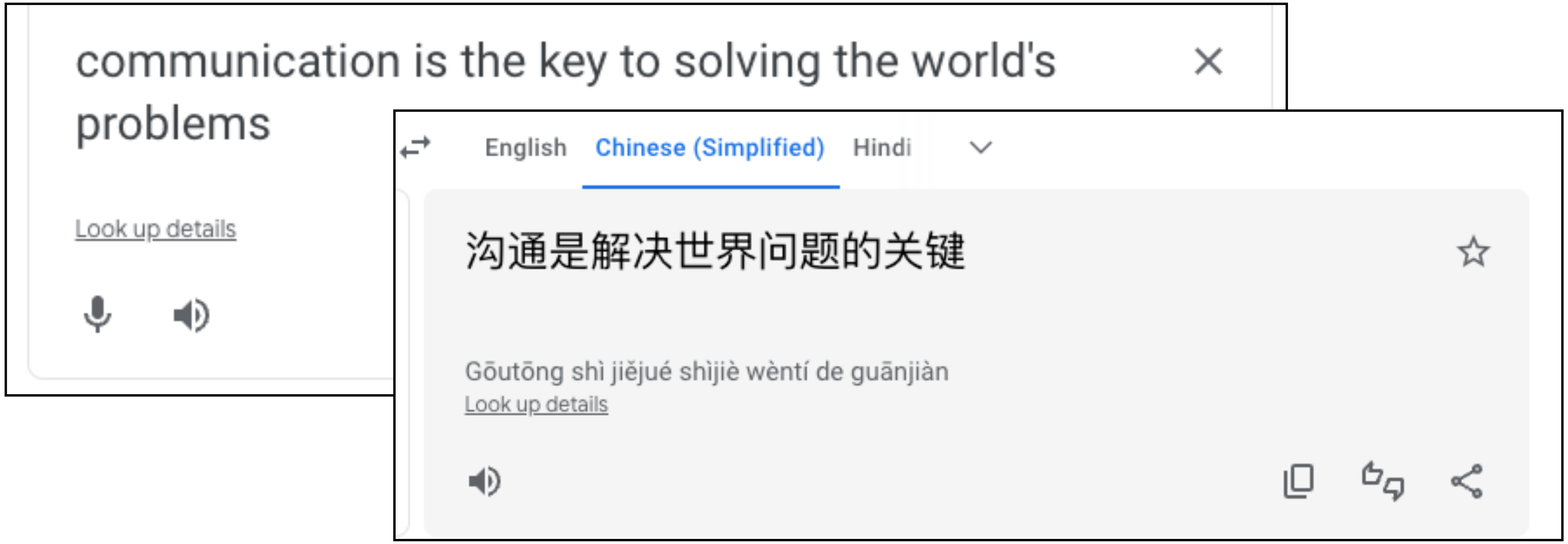
BAIR

Sequence-to-sequence modeling

- We’ll talk about sequence-to-sequence (seq2seq)!
- Encoder-decoder architectures
    - The bottleneck: compressing a sentence into one vector
    - The birth of “attention” as a weighting mechanism.
- We’ll talk about Machine Translation (MT) as a Case Study
- Lecture plans: A little about MT (15min) → Seq2seq (50min)

A bit about Machine Translation

Translation



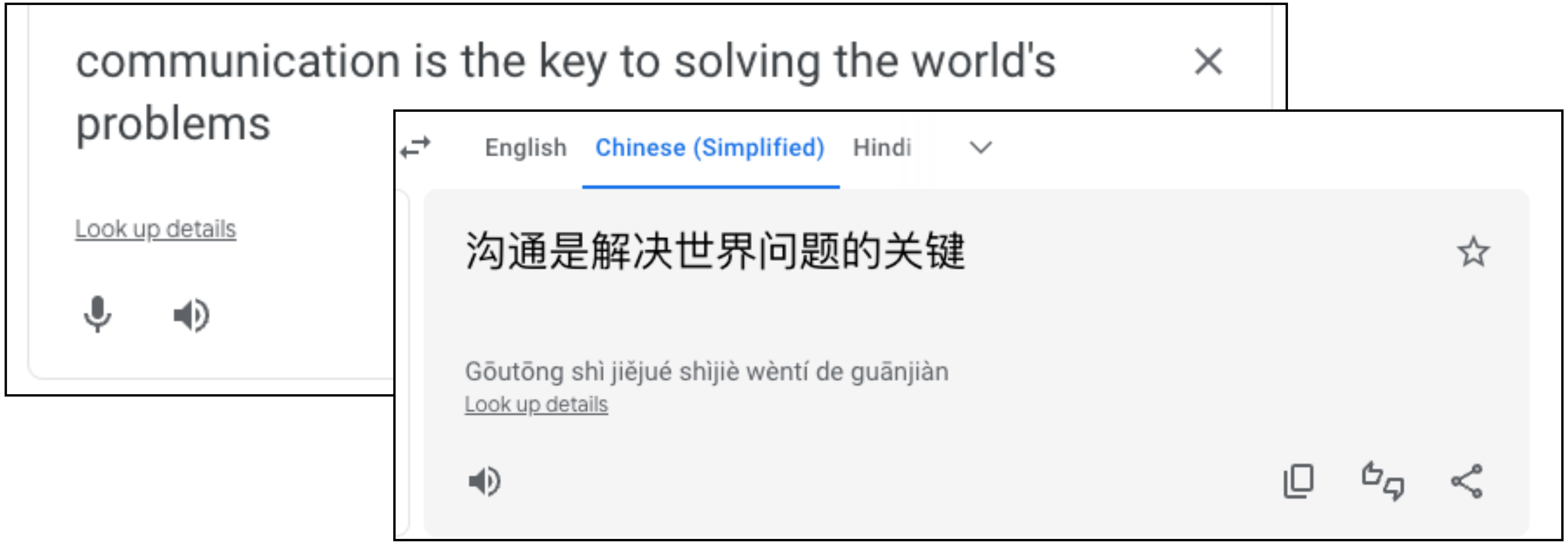
communication is the key to solving the world's problems

沟通是解决世界问题的关键

Gōutōng shì jiějué shìjiè wèntí de guānjiàn

- 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



communication is the key to solving the world's problems

沟通是解决世界问题的关键

Gōutōng shì jiějué shìjiè wèntí de guānjiàn

How many languages do you speak?

A) 1
B) 2
C) 3
D) 4+

Machine Translation (MT)

- Goal: Translate a sentence $\mathbf{w}^{(s)}$ in a source language (input) to a sentence $\mathbf{w}^{(t)}$ in the target language (output)

I like apples ↔ ich mag Äpfel (German)

- Why is MT challenging?
  - Single words may be replaced with multi-word phrases:

    I like apples ↔ J'aime les pommes (French)
  - Reordering of phrases:

    I like red apples ↔ J'aime les pommes rouges (French)
  - Context-dependent translations:

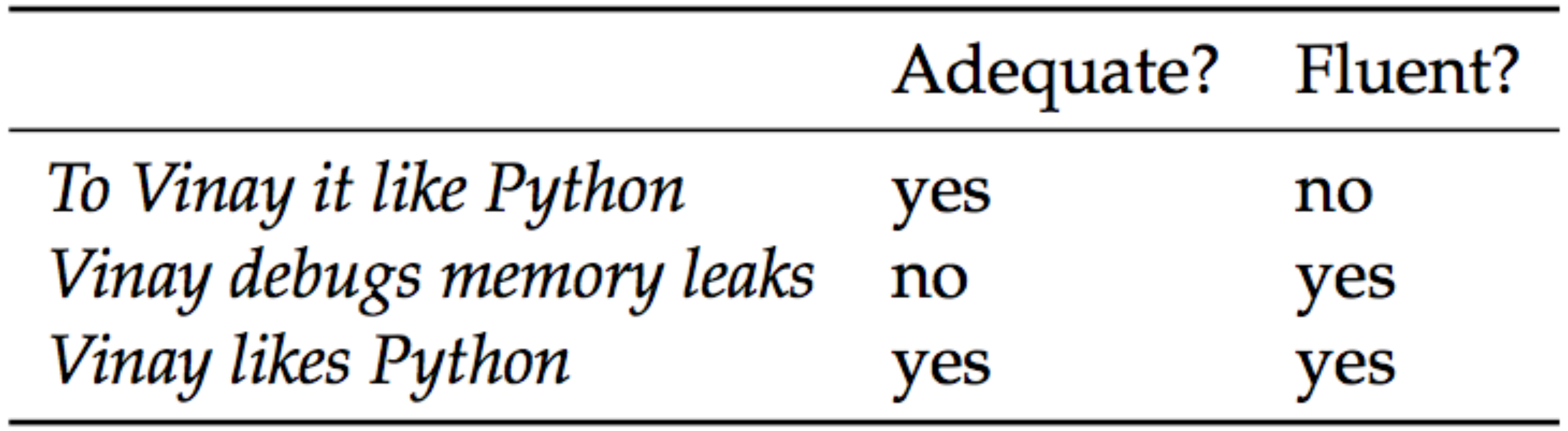
    les ↔ *the* but *les pommes* ↔ *apples*

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

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



	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” (Spanish)

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

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

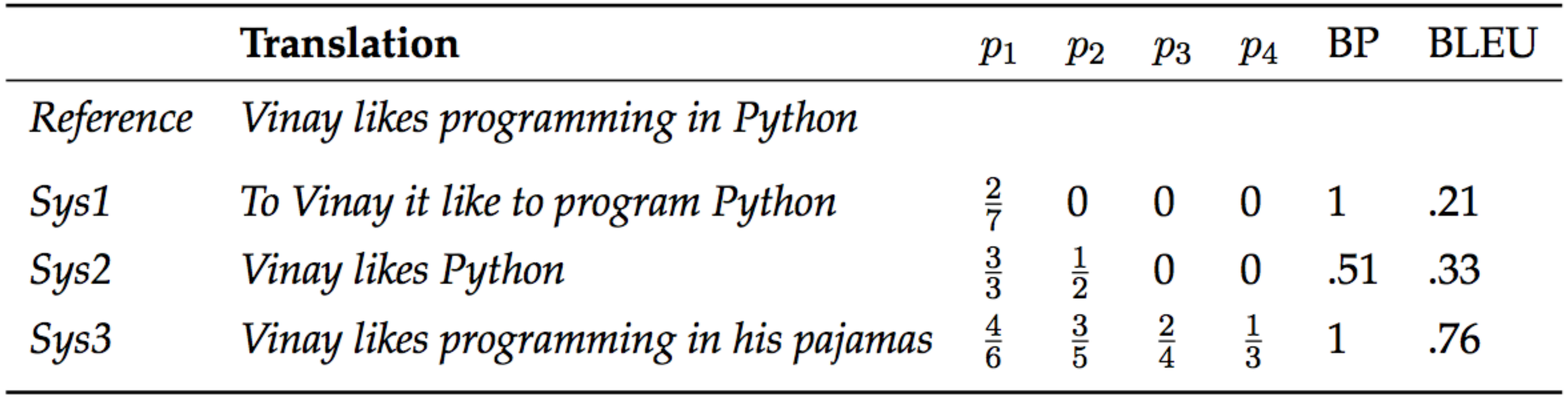
Reference translation

System predictions

Evaluation metric: BLEU

- Calculate modified n-gram precision p_n (usually for 1, 2, 3 and 4-grams)
- Plus a “brevity penalty” for too-short system translations
- The final BLEU score takes the geometric mean of p_n (with smoothing) × brevity penalty
- BLEU ranges between 0 and 1 and people usually express them in percentage

BP: brevity penalty



	Translation	p_1	p_2	p_3	p_4	BP	BLEU
Reference	*Vinay likes programming in Python*						
Sys1	*To Vinay it like to program Python*	$\frac{2}{7}$	0	0	0	1	.21
Sys2	*Vinay likes Python*	$\frac{3}{3}$	$\frac{1}{2}$	0	0	.51	.33
Sys3	*Vinay likes programming in his pajamas*	$\frac{4}{6}$	$\frac{3}{5}$	$\frac{2}{4}$	$\frac{1}{3}$	1	.76

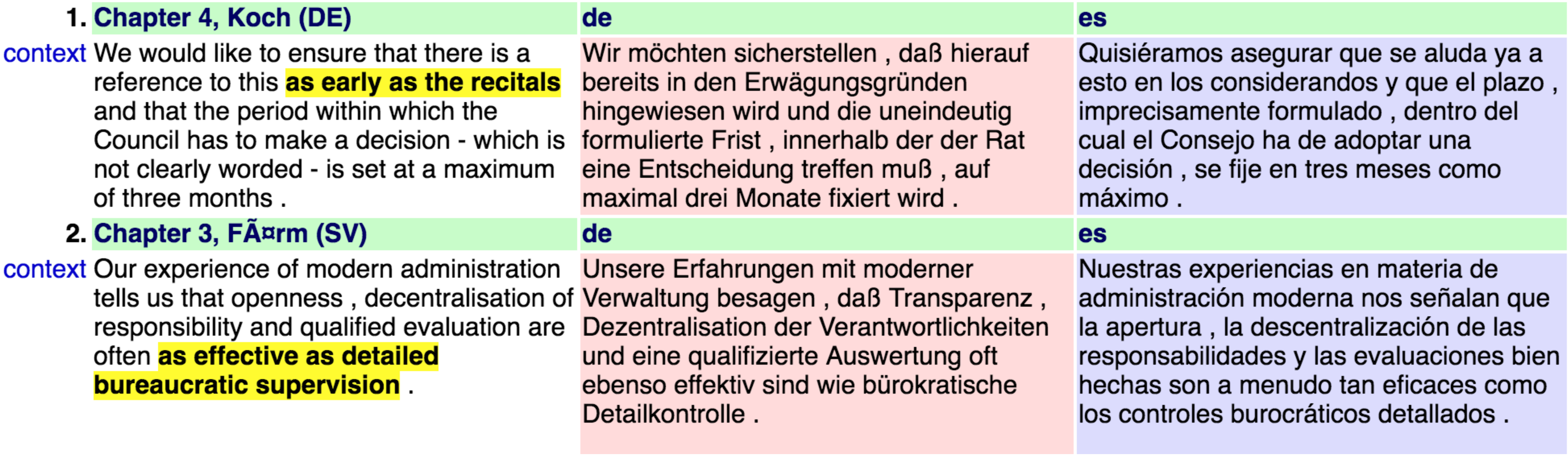
Sample BLEU scores for various system outputs

BLEU is **useful (and widely used)** but **far from perfect**

A **good** translation can get a **poor** BLEU score because it has low n-gram overlap with human translation

Machine translation: Data

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



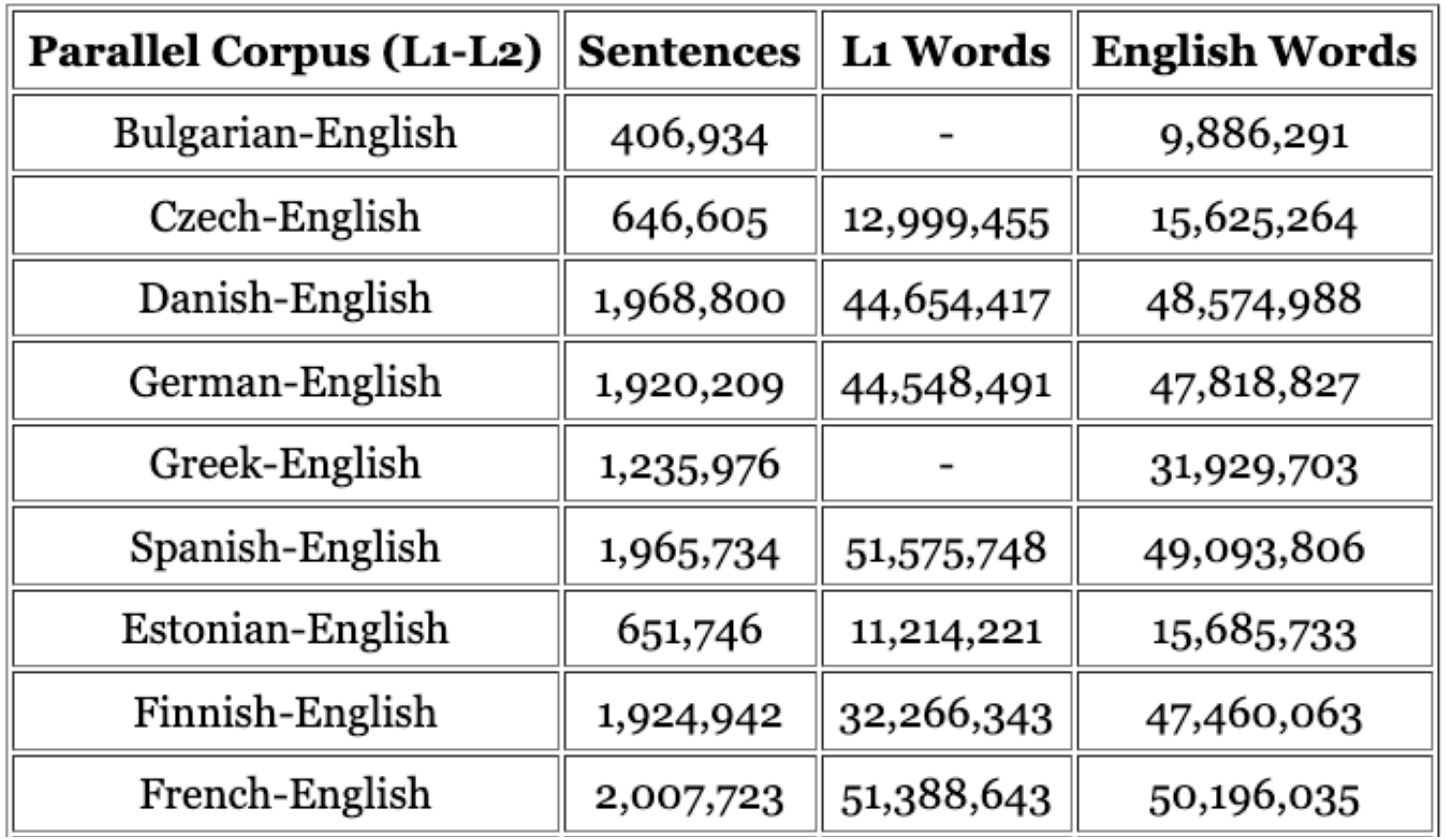
1.	**Chapter 4, Koch (DE)**	**de**	**es**
context	We would like to ensure that there is a reference to this **as early as the recitals** and that the period within which the Council has to make a decision - which is not clearly worded - is set at a maximum of three months .	Wir möchten sicherstellen , 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/

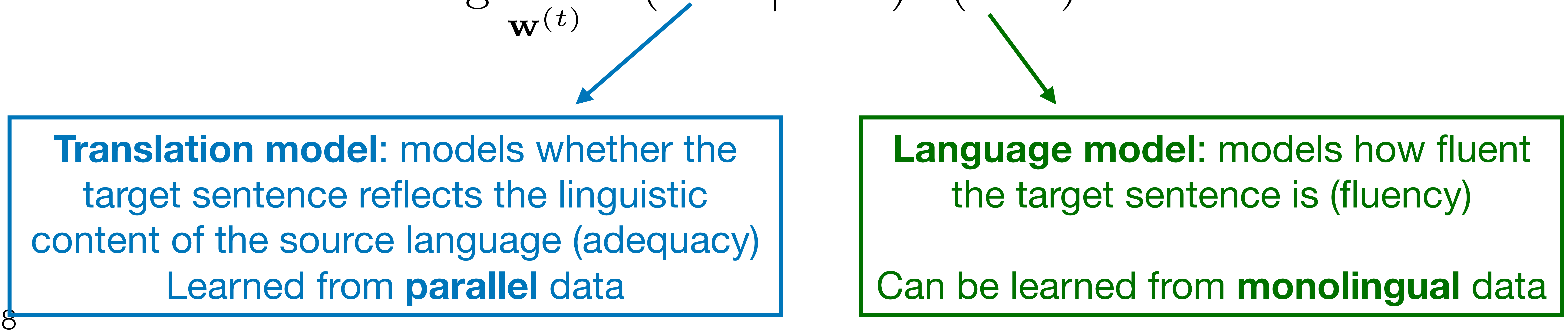
Statistical machine translation (SMT)

- 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

Statistical machine translation (SMT)

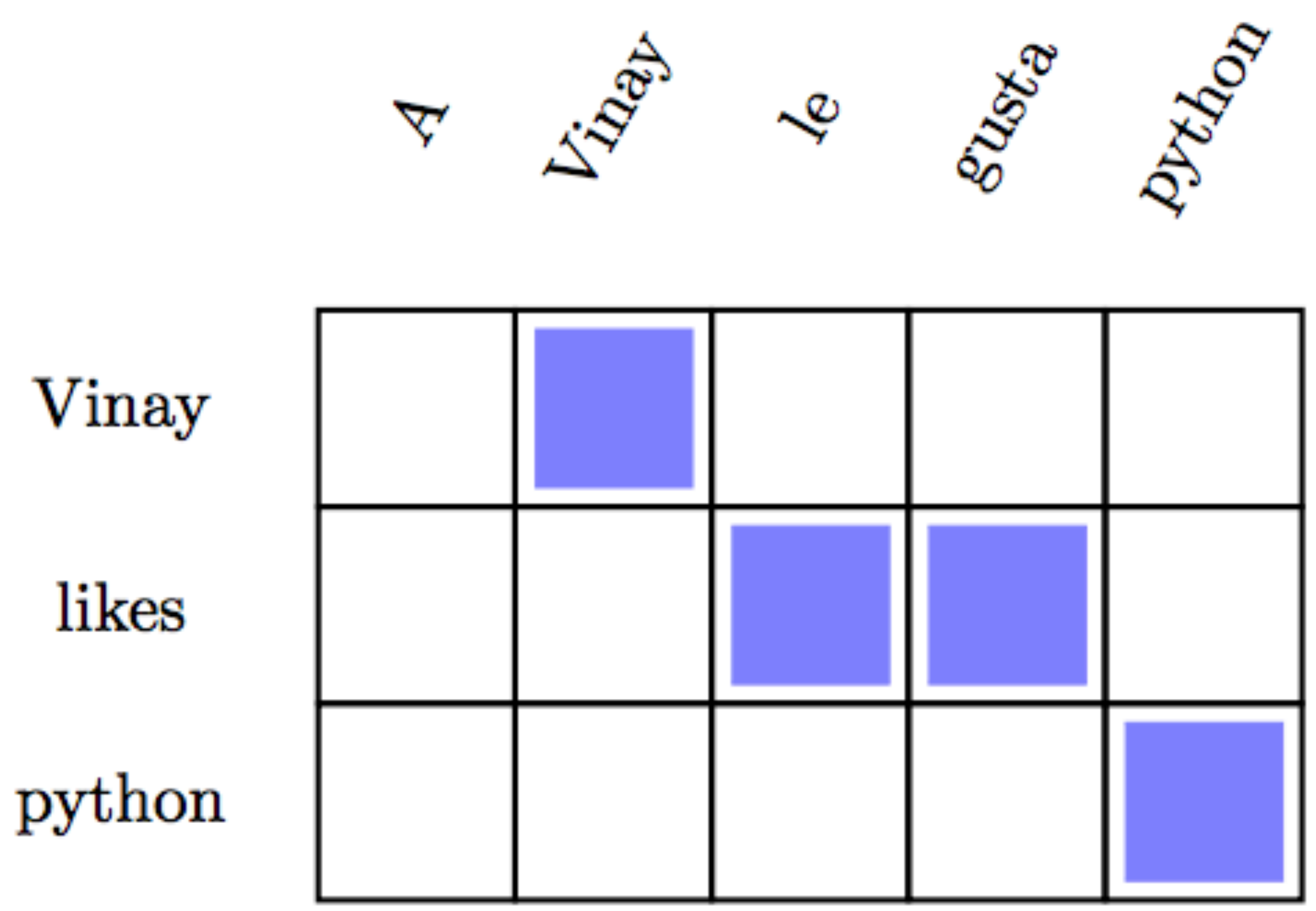
$$\arg\max_{\mathbf{w}^{(t)}} P(\mathbf{w}^{(\mathbf{s})} \mid \mathbf{w}^{(\mathbf{t})}) P(\mathbf{w}^{(t)})$$

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

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



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

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

Examples: IBM models 1, 2, 3, 4, 5

Statistical machine translation (SMT)

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

Q: But I don't understand. Isn't $P(\mathbf{w}^{(\mathbf{s})} \mid \mathbf{w}^{(\mathbf{t})})$ as hard as $P(\mathbf{w}^{(t)} \mid \mathbf{w}^{(s)})$?

A: Yes, but it allows you to not worry about fluency!

Example: Imagine you are translating from Spanish to English.

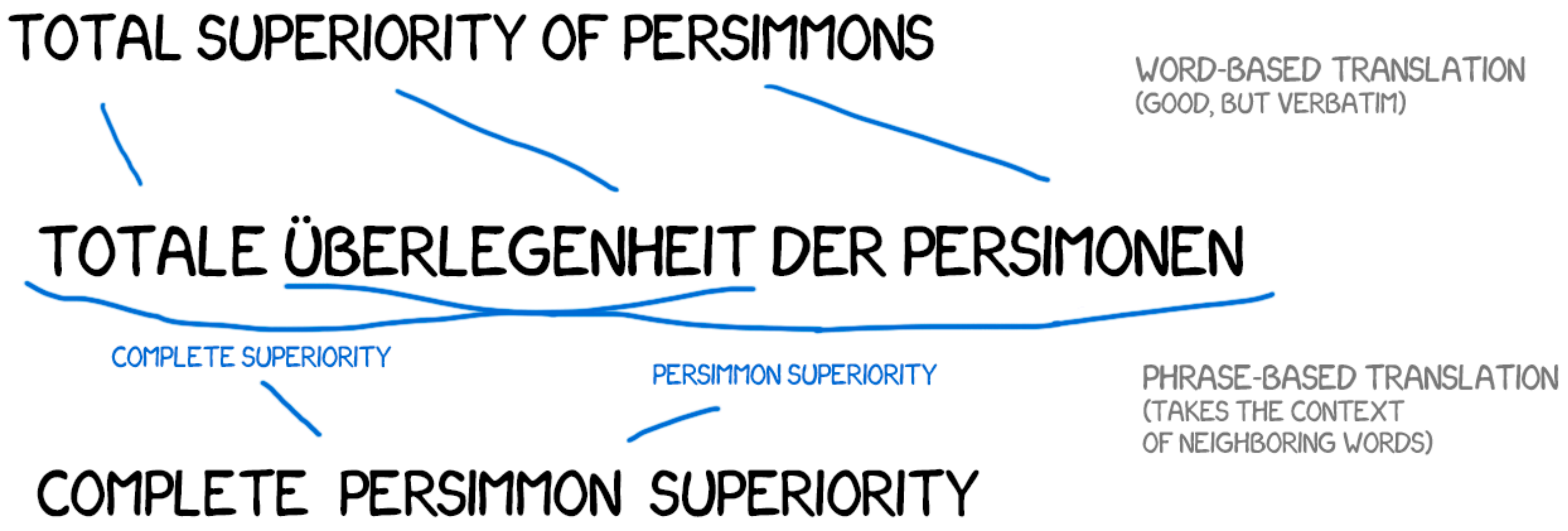
- Source (s): "La casa verde"
- Option 1 (t1): "The green house"
- Option 2 (t2): "The house green"

P(s|t) will actually give both t1 and t2 a high score. However, the language model P(t) will score t1 much higher.

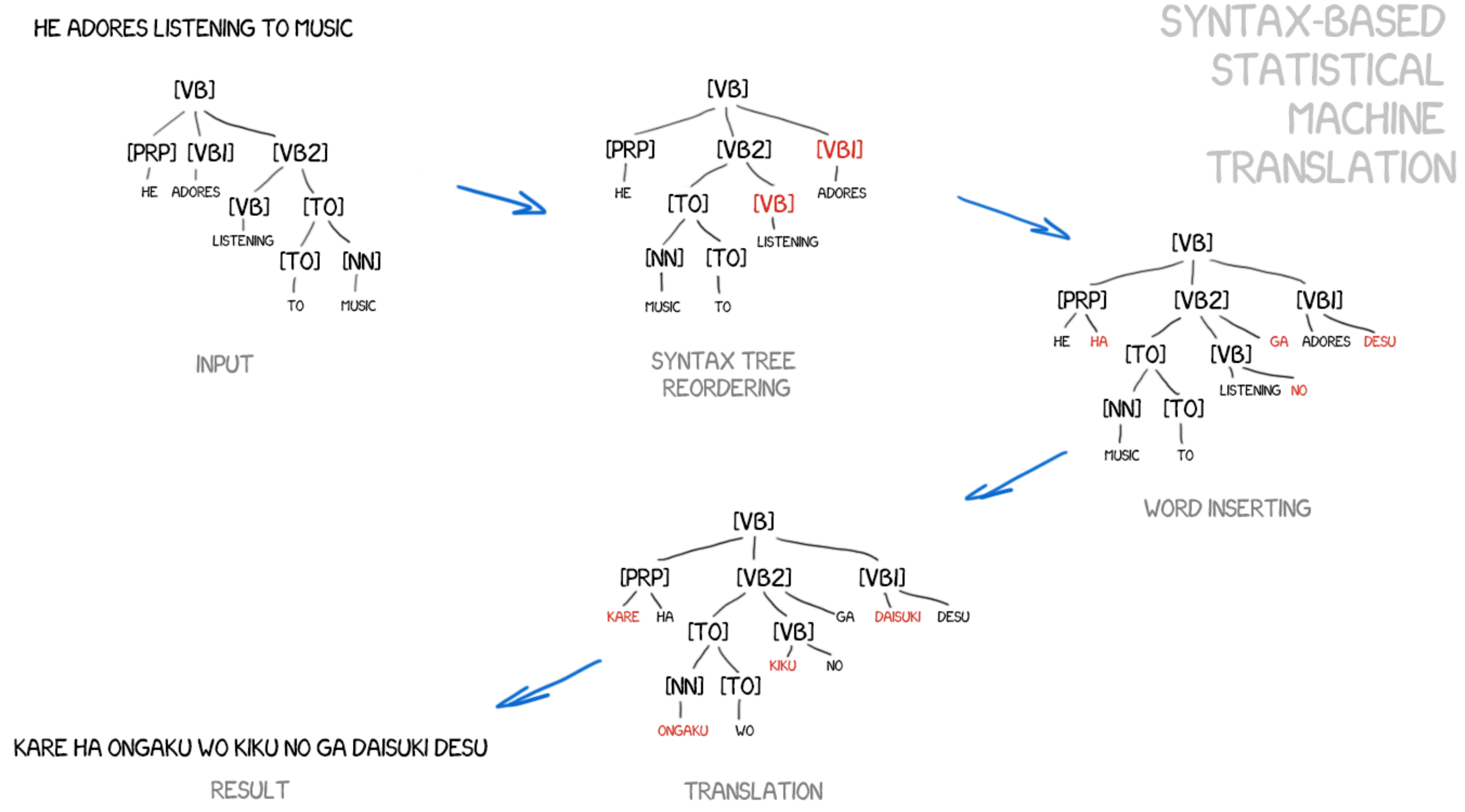
Statistical machine translation (SMT)

- 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

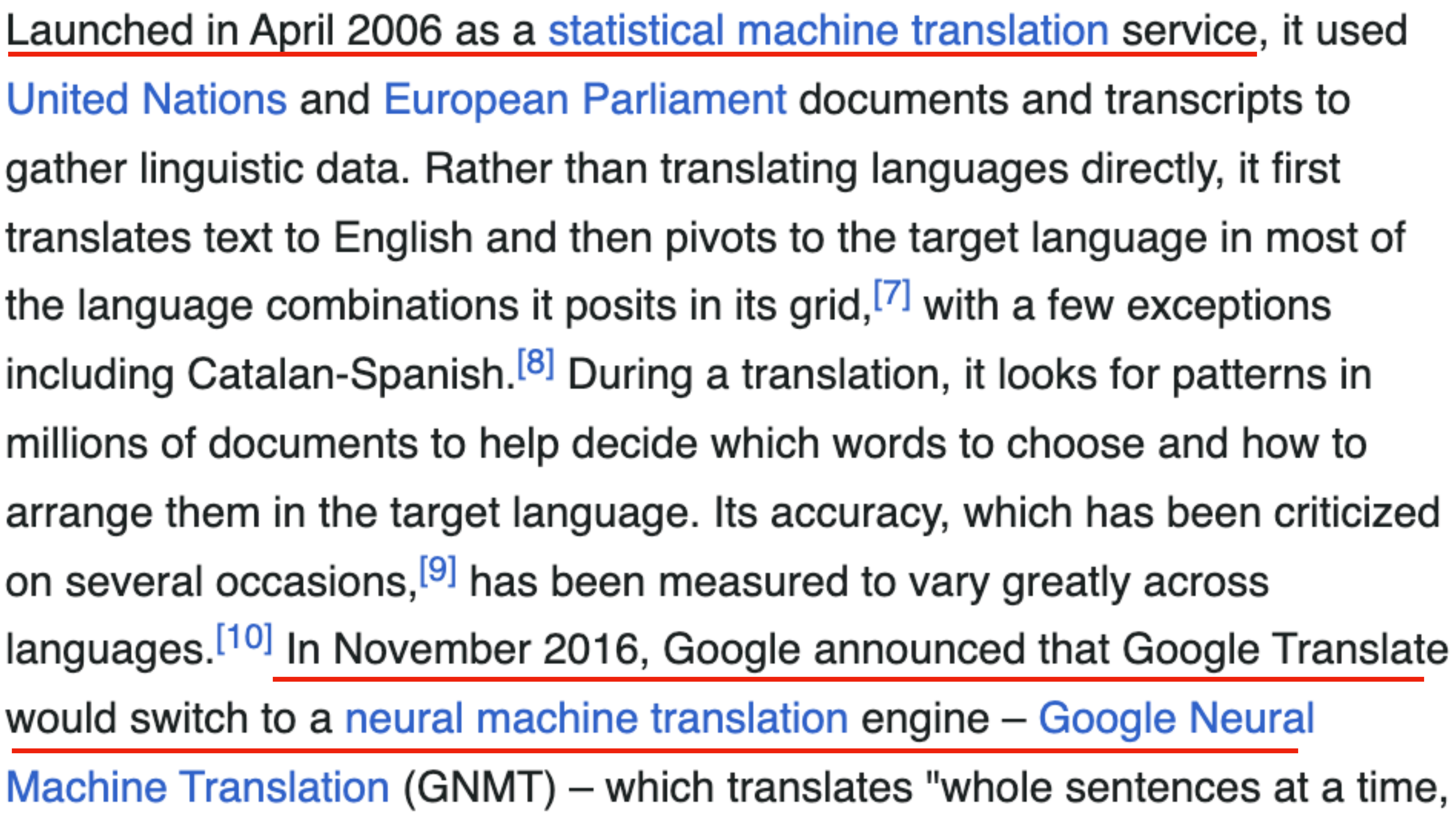


https://translartisan.wordpress.com/tag/statistical-machine-translation/

Neural Machine Translation (NMT)

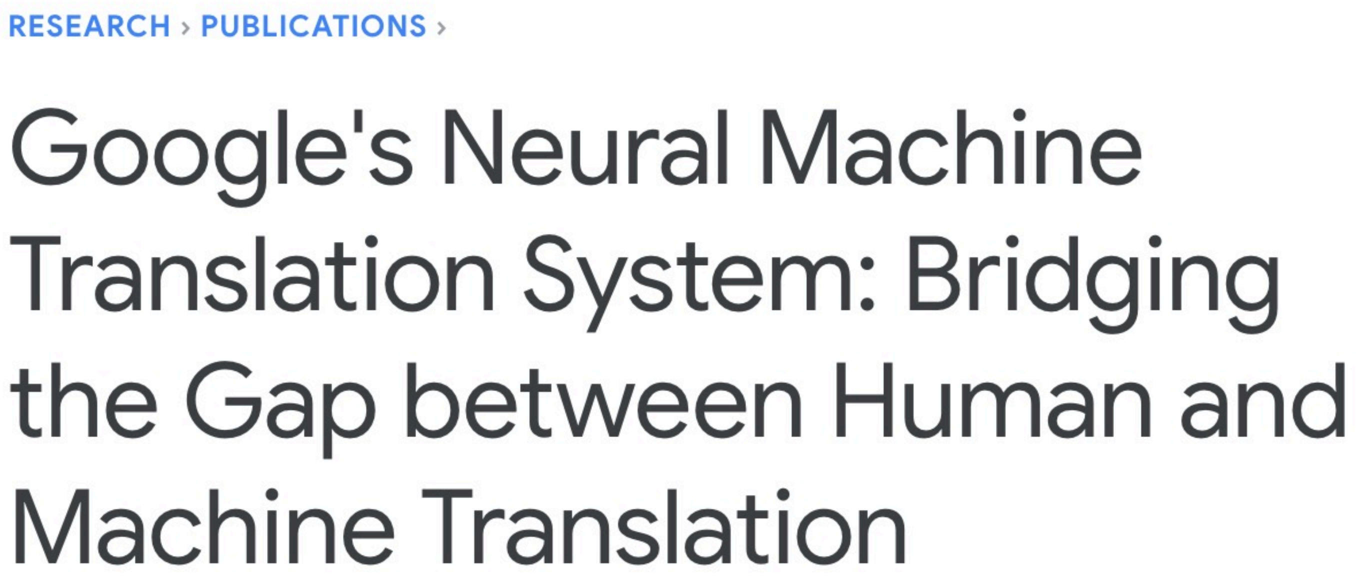
SMT ⟶ NMT

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

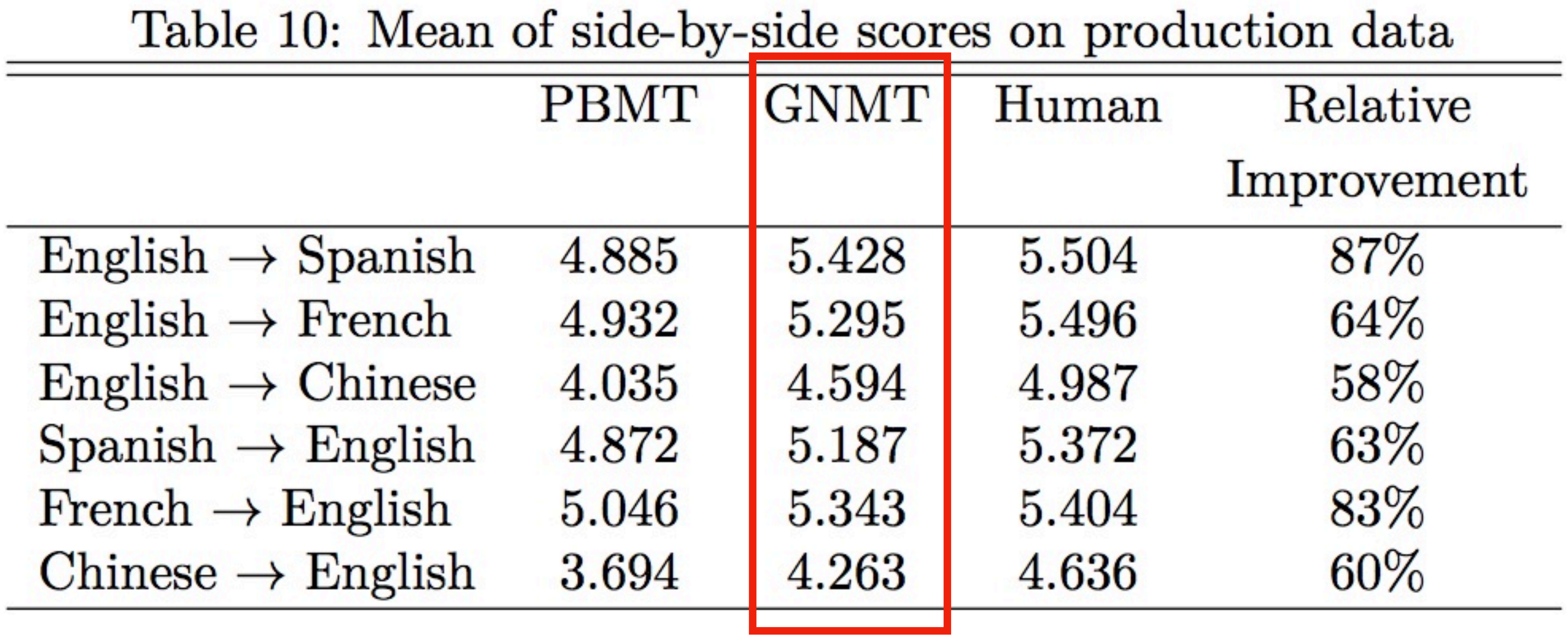


Table 10: Mean of side-by-side scores on production data

	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

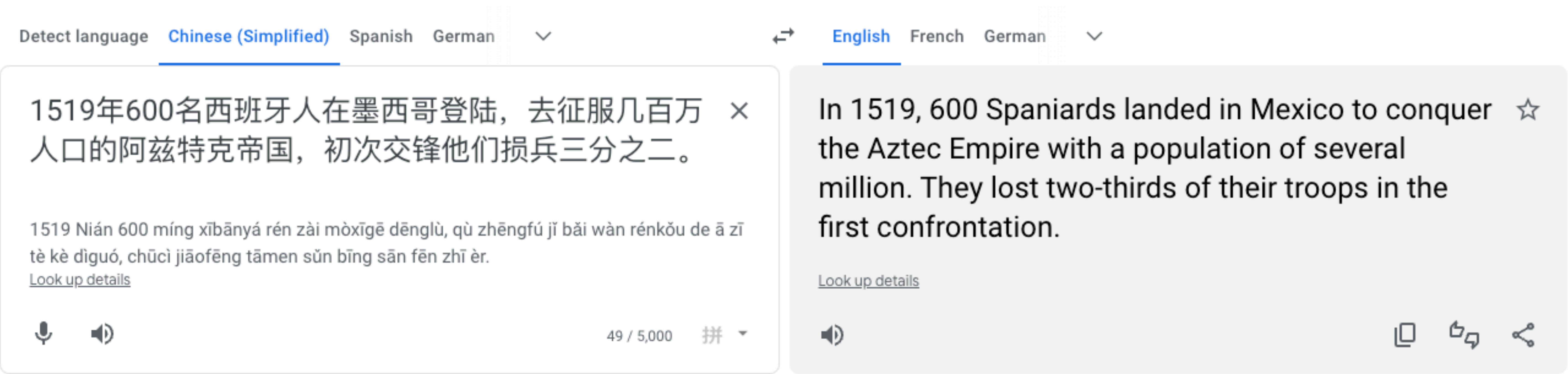
1519年600名西班牙人在墨西哥登陆，去征服几百万人口的阿兹特克帝国，初次交锋他们损兵三分之二。

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.



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。

1519 Nián 600 míng xībānyá rén zài mòxīgē dēnglù, qù zhēngfú jǐ bǎi wàn rénkǒu de ā zī tè kè dìguó, chūcì jiāofēng tāmen sǔn bīng sān fēn zhī èr.

In 1519, 600 Spaniards landed in Mexico to conquer the Aztec Empire with a population of several million. They lost two-thirds of their troops in the first confrontation.

Feb 5 lecture starts from here

CS 288 Advanced Natural Language Processing

Course website: cal-cs288.github.io/sp26

Ed: edstem.org/us/join/XvztdK

- Class starts at 15:40!
- Next Tuesday (Feb 10): A1 due, team registration for A3 & project due
- Lecture plan: seq2seq (50min) → Transformers (30min)

Recap: Machine Translation (MT)

- Goal: Translate a sentence $\mathbf{w}^{(s)}$ in a source language (input) to a sentence $\mathbf{w}^{(t)}$ in the target language (output)

I like apples ↔ ich mag Äpfel (German)

- Why is MT challenging?
  - Single words may be replaced with multi-word phrases:

    I like apples ↔ J'aime les pommes (French)
  - Reordering of phrases:

    I like red apples ↔ J'aime les pommes rouges (French)
  - Context-dependent translations:

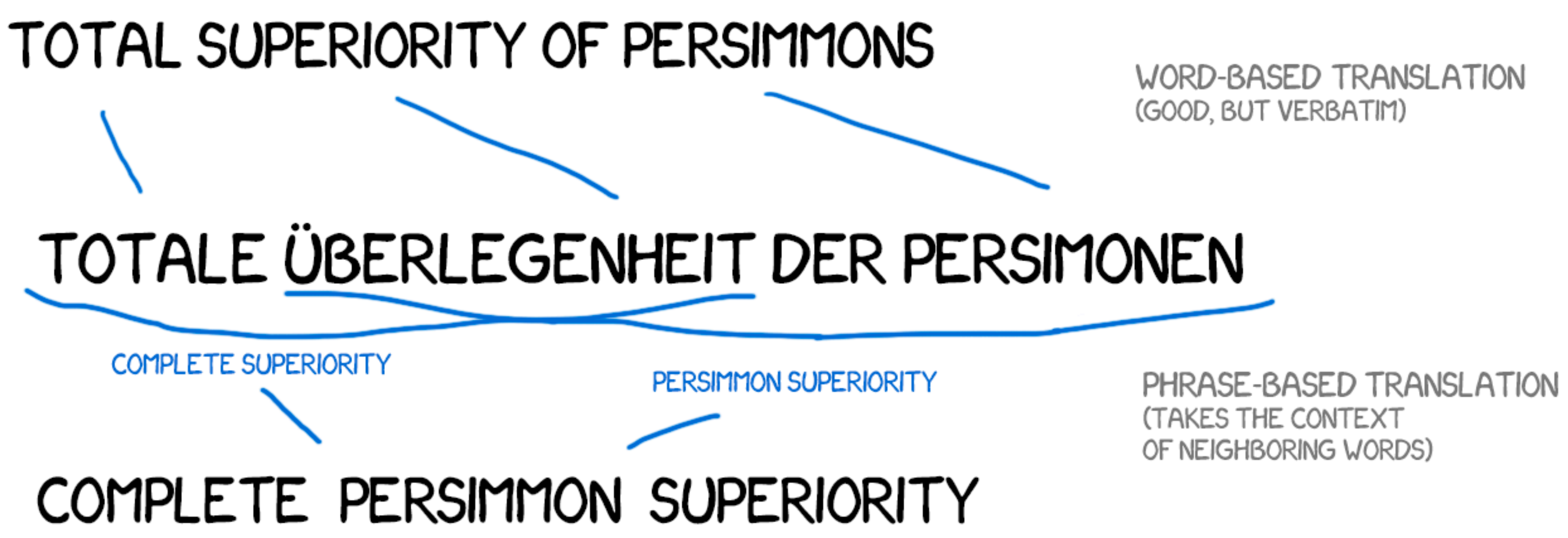
    les ↔ *the* but *les pommes* ↔ *apples*

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

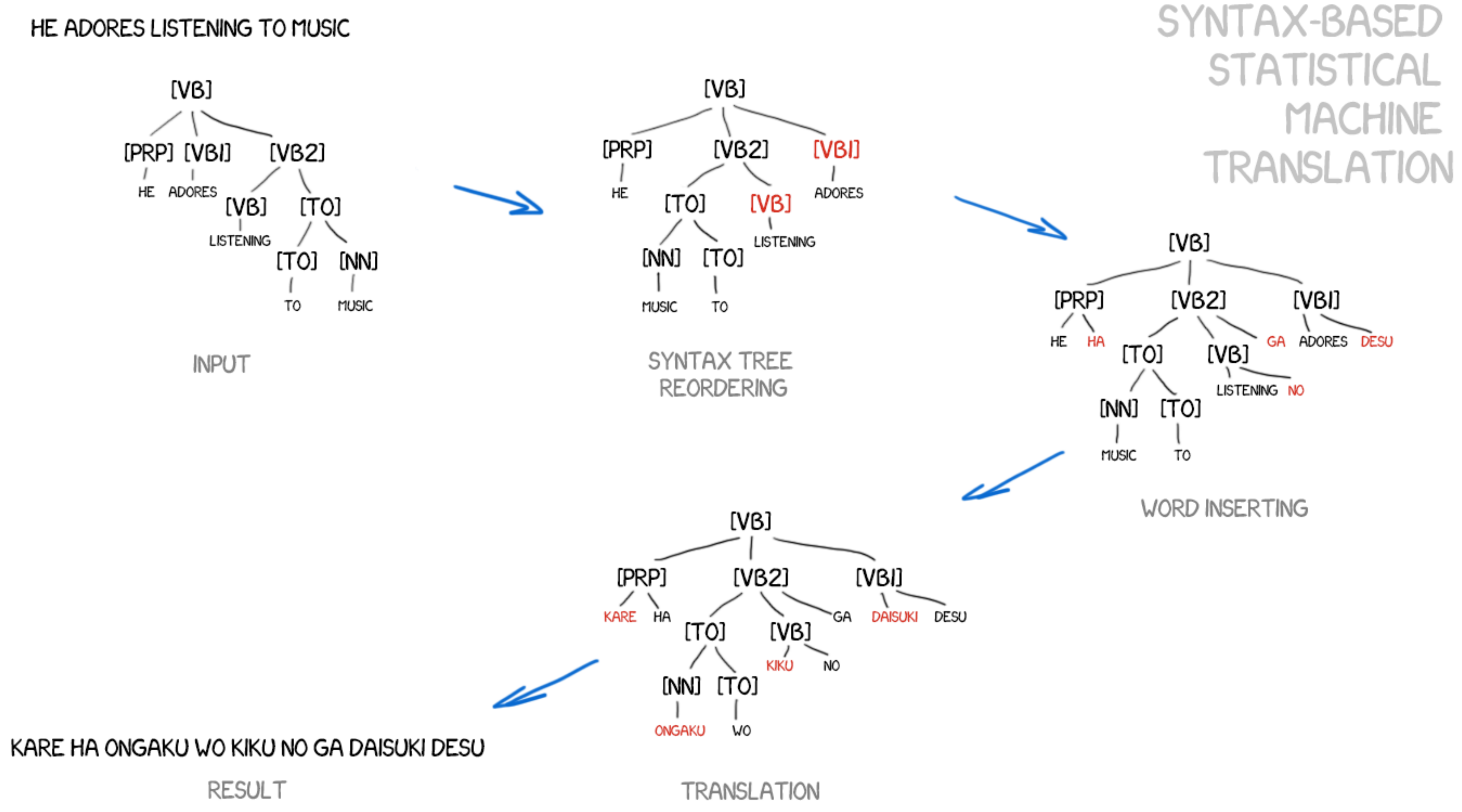
Recap: Statistical machine translation (SMT)

- 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

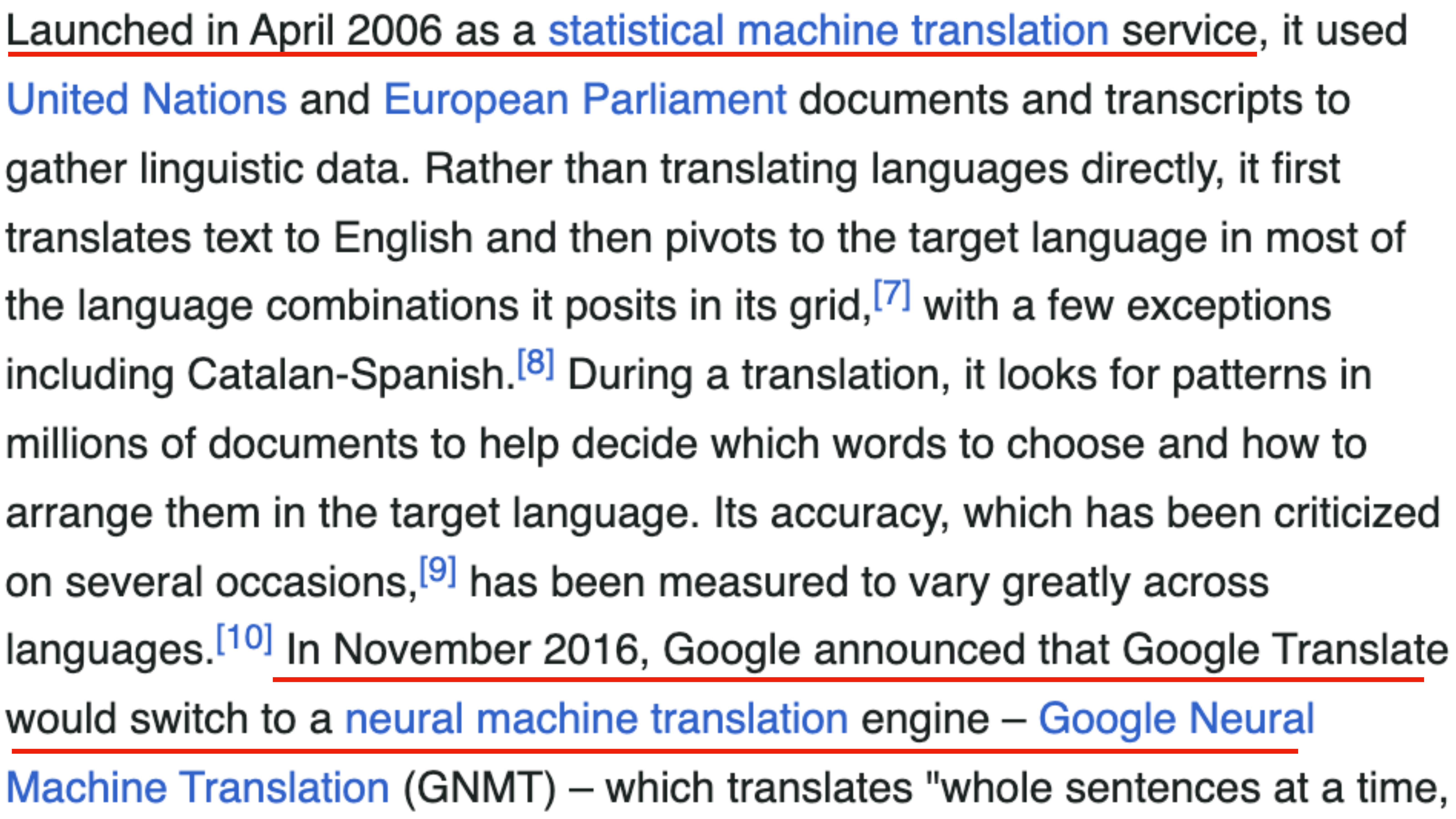


Syntax-based SMT



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

Recap: Transitioning from SMT to NMT

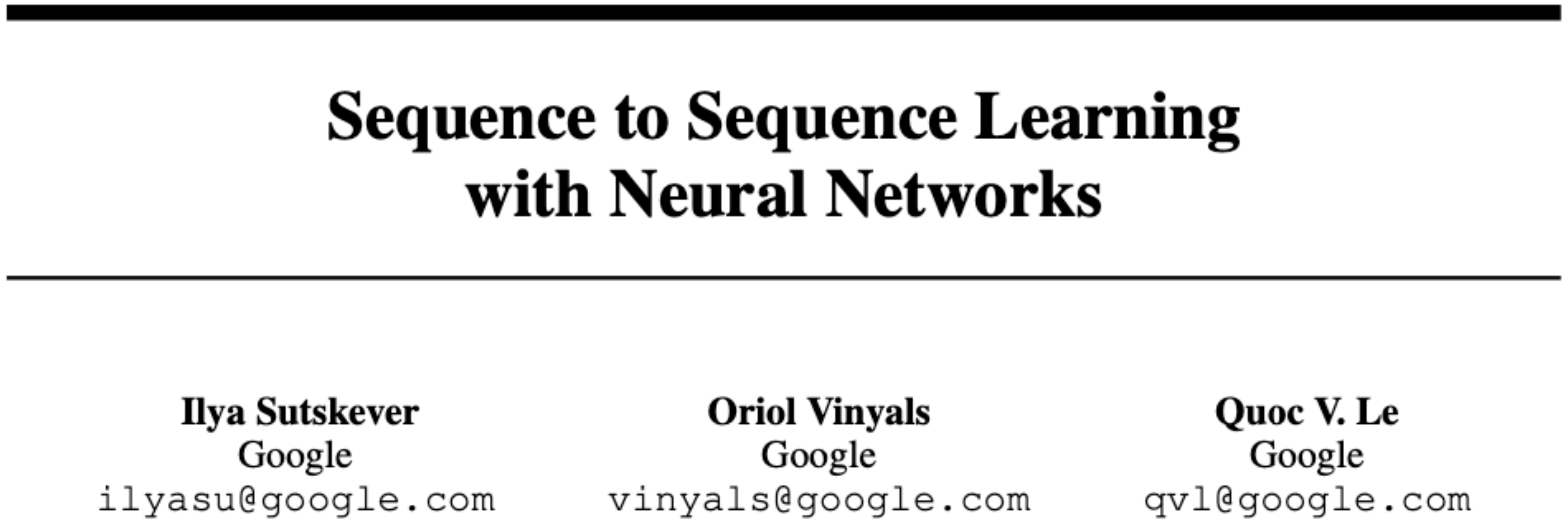


Launched in April 2006 as a statistical machine translation service, it used United Nations and European Parliament documents and transcripts to gather linguistic data. Rather than translating languages directly, it first translates text to English and then pivots to the target language in most of the 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,

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
Google
ilyasu@google.com

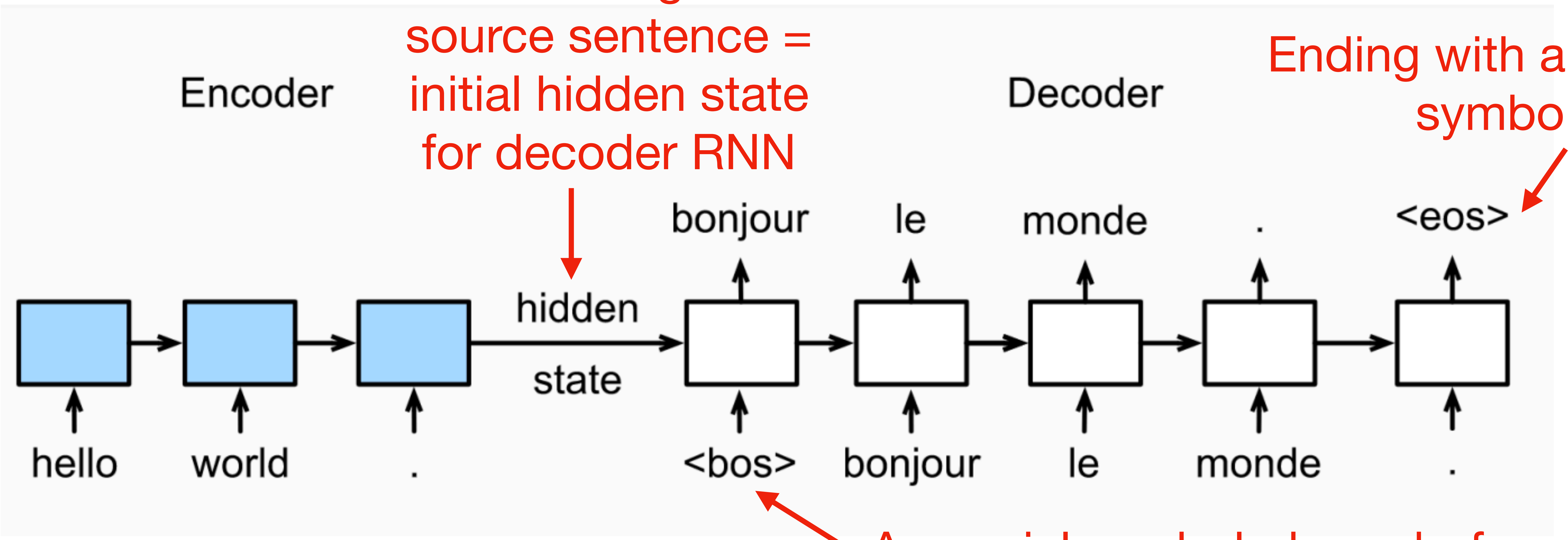
Oriol Vinyals
Google
vinyals@google.com

Quoc V. Le
Google
qvl@google.com

(Sutskever et al., 2014)

Ilya Sutskever

The sequence-to-sequence model (seq2seq)



Encoding of source sentence = initial hidden state for decoder RNN

Ending with another special symbol <eos>

A special symbol <bos> before generating the first word

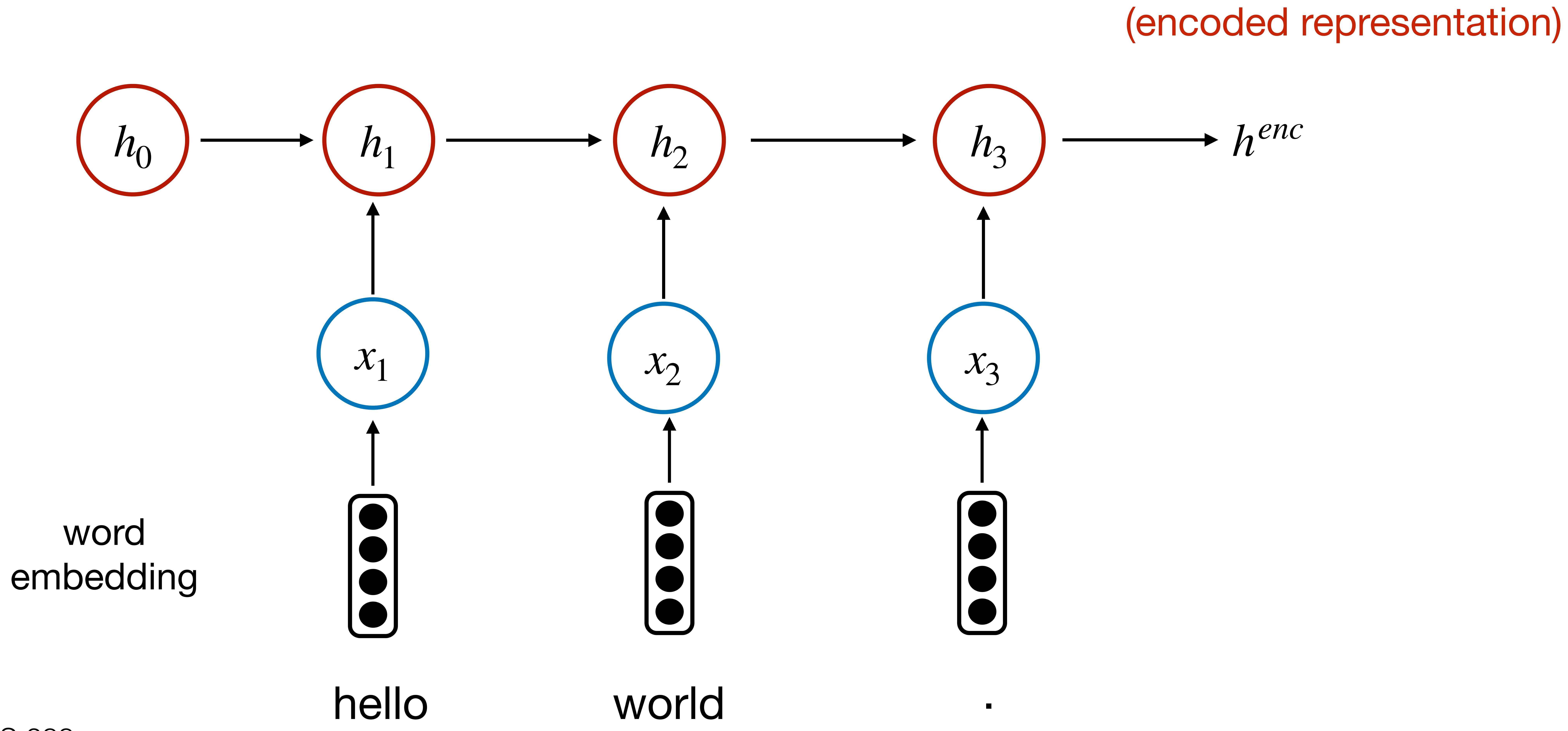
It is called an **encoder-decoder** architecture

- 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

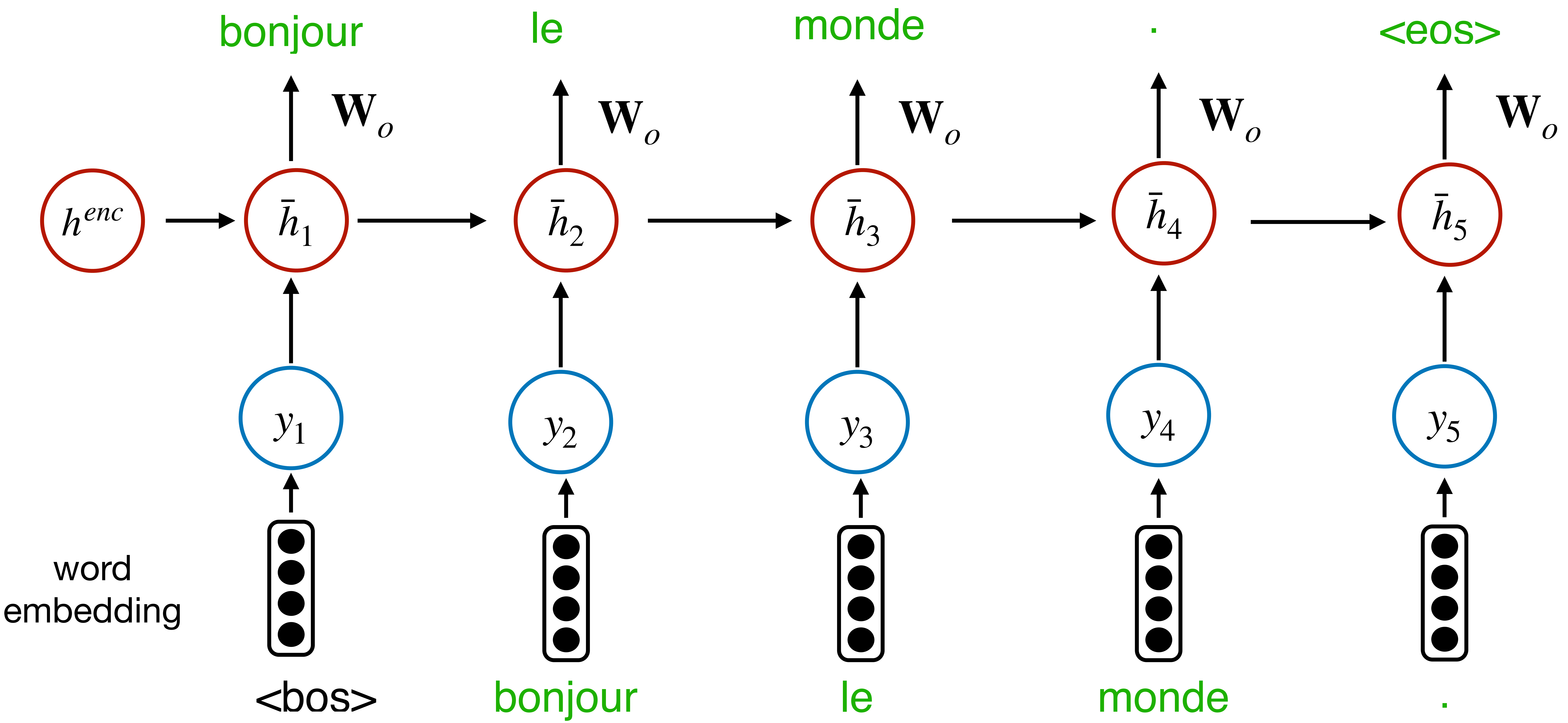
Seq2seq: Encoder

Sentence: hello world .

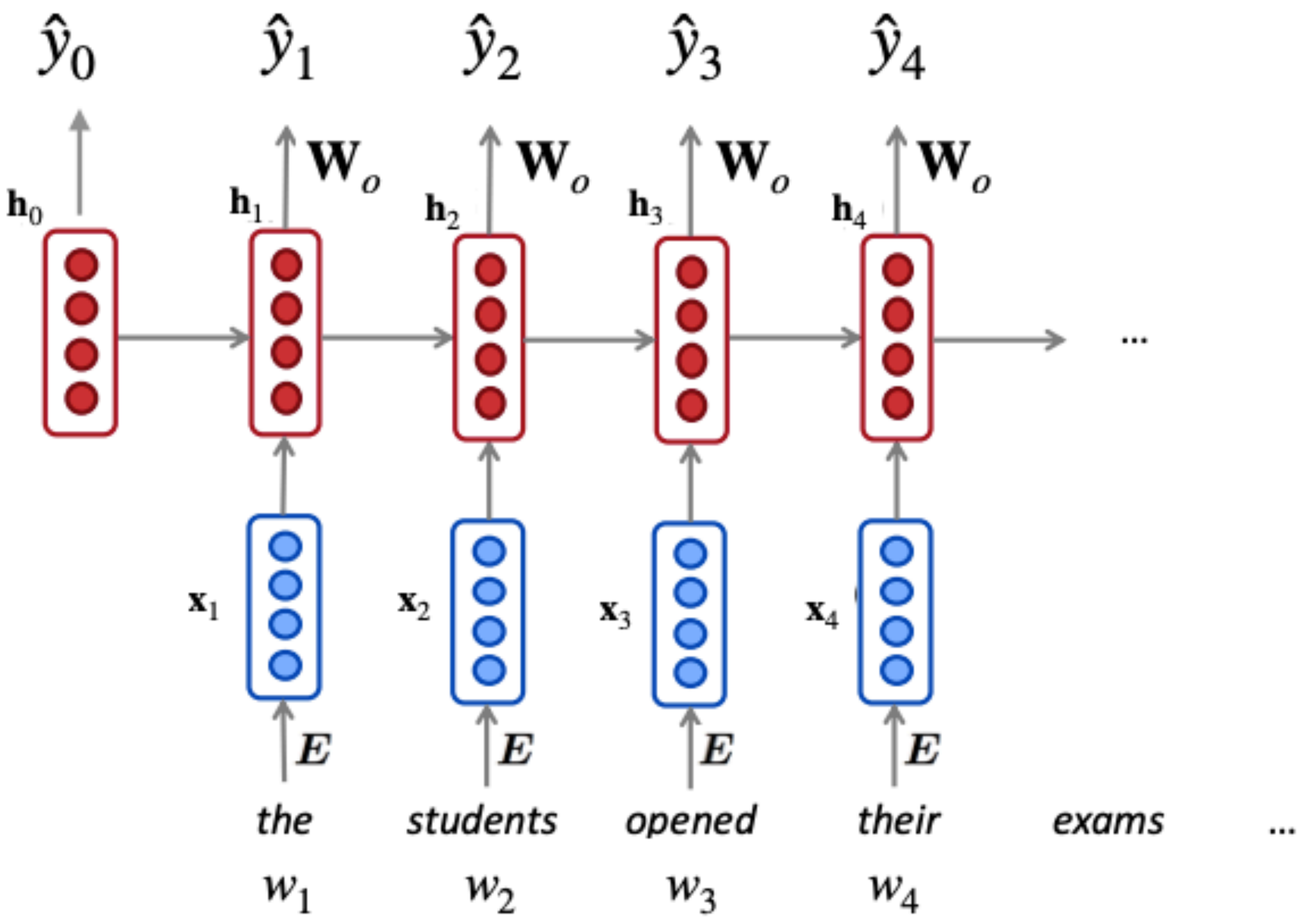


Seq2seq: Decoder

- A **conditional** language model



Recap: recurrent neural models
(The case of language modeling)



$$\mathbf{h}_t = g(\mathbf{W}\mathbf{h}_{t-1} + \mathbf{U}\mathbf{x}_t + \mathbf{b}) \in \mathbb{R}^h$$

$$\hat{\mathbf{y}}_t = softmax(\mathbf{W}_o\mathbf{h}_t)$$

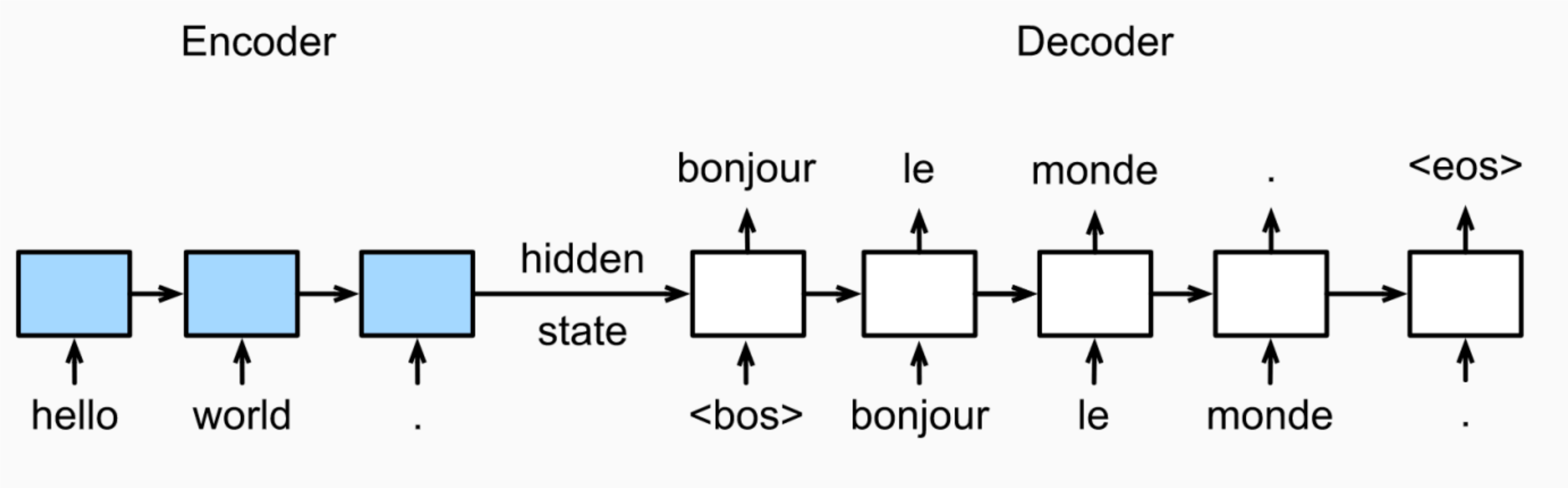
Training loss:

$$\boxed{L(\theta) = -\frac{1}{n}\sum_{t=1}^{n} \log \hat{\mathbf{y}}_{t-1}(w_t)}$$

Trainable parameters:

$$\theta = \{\mathbf{W}, \mathbf{U}, \mathbf{b}, \mathbf{W}_o, \mathbf{E}\}$$

Understanding seq2seq

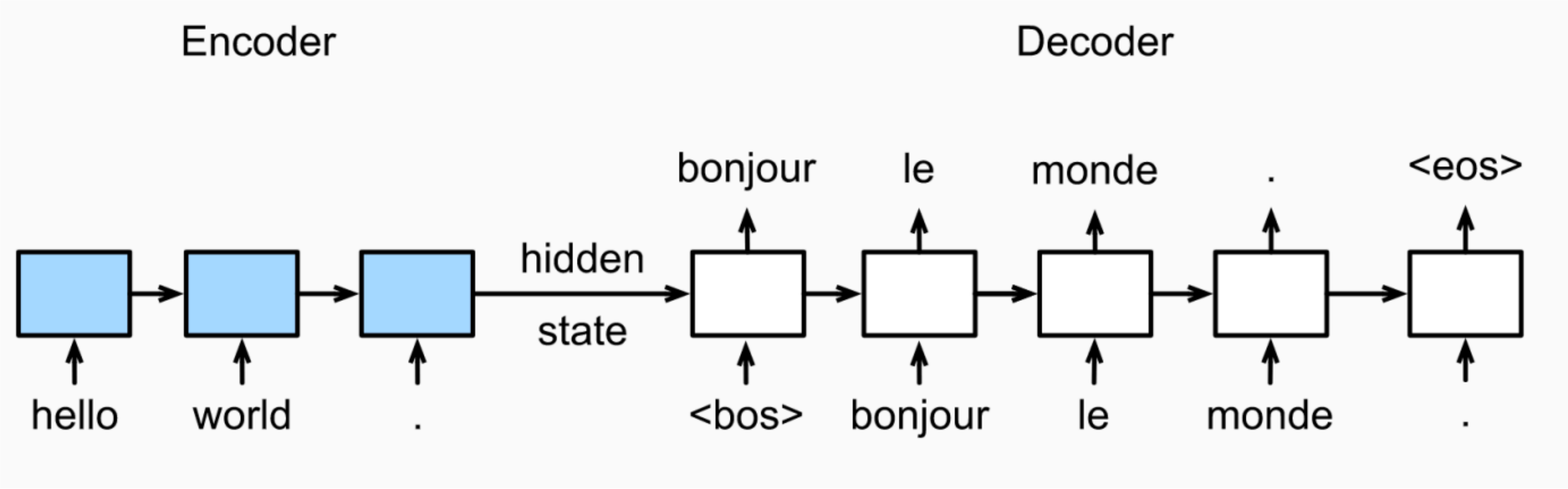


Which of the following is correct?

- (A) We can use bidirectional RNNs for both encoder and decoder
- (B) The decoder has more parameters because of the output matrix $\mathbf{W}_o$
- (C) The encoder and decoder have separate word embeddings
- (D) The encoder and decoder's parameters are optimized together

Both (C) and (D) are correct.

Understanding seq2seq



Encoder RNN:

- word embeddings $\mathbf{E}^{(s)}$ for source language
- RNN parameters, e.g., $\{\mathbf{W}^{(s)}, \mathbf{U}^{(s)}, \mathbf{b}^{(s)}\}$ for simple RNNs and 4x parameters for LSTMs
- Encoder RNN can be bidirectional!

Decoder RNN:

- word embeddings $\mathbf{E}^{(t)}$ for target language
- RNN parameters, e.g., $\{\mathbf{W}^{(t)}, \mathbf{U}^{(t)}, \mathbf{b}^{(t)}\}$ for simple RNNs and 4x parameters for LSTMs
- Output embedding matrix $\mathbf{W}_o$ = can be tied with $\mathbf{E}^{(t)}$
- Decoder RNN has to be unidirectional (left to right)!

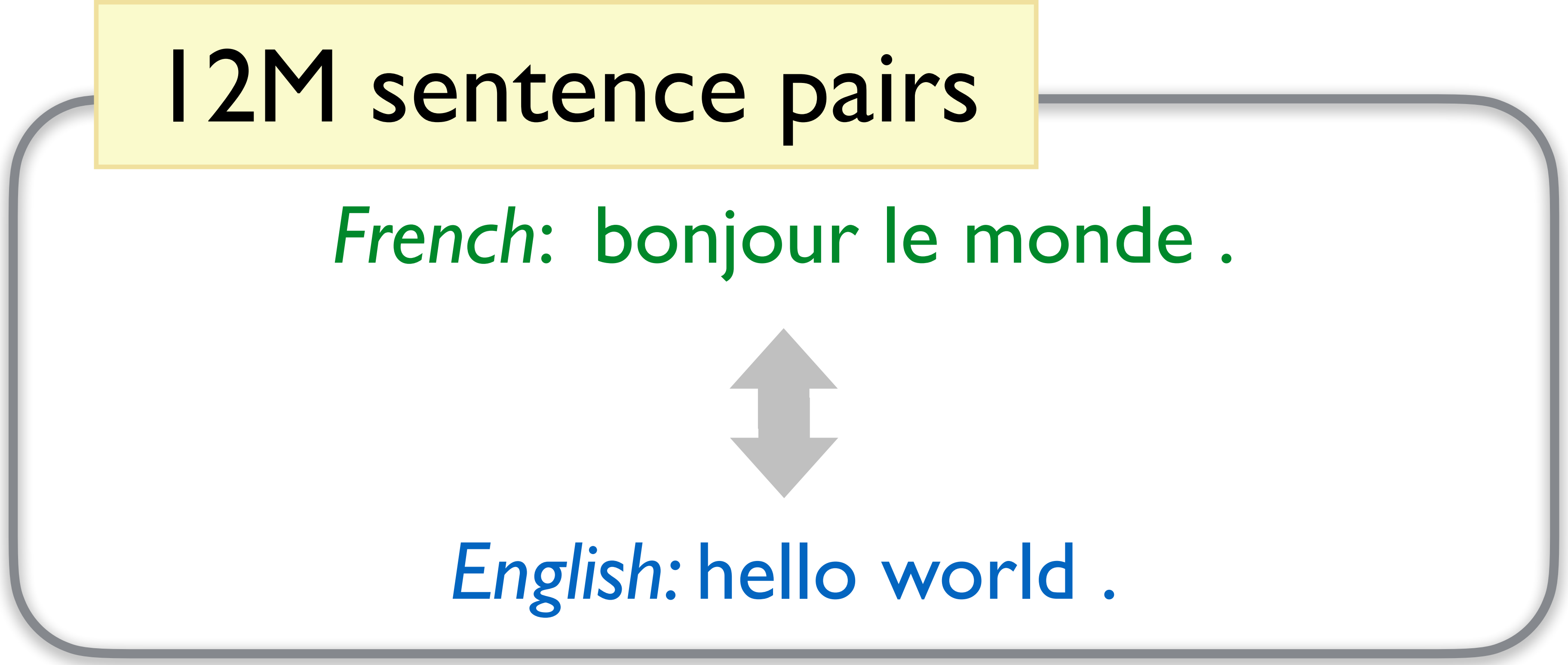
Training seq2seq models

- Training data: parallel corpus $\{(\mathbf{w}_i^{(s)}, \mathbf{w}_i^{(t)})\}$

- Minimize cross-entropy loss:

$$\sum_{t=1}^{T} -\log P(y_t \mid y_1, \ldots, y_{t-1}, \mathbf{w}^{(s)})$$

(denote $\mathbf{w}^{(t)} = y_1, \ldots, y_T$)



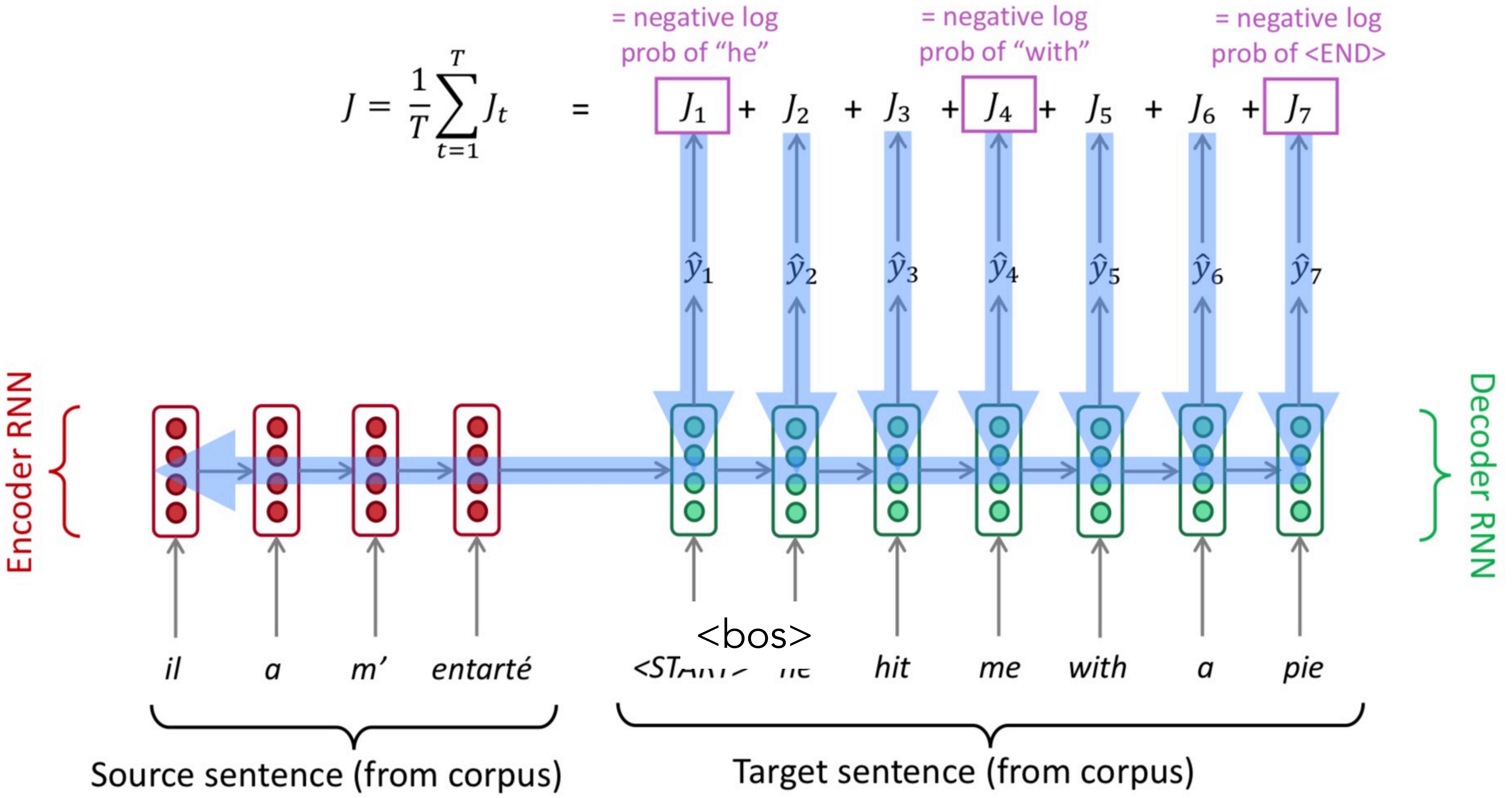
12M sentence pairs

French: bonjour le monde .

English: hello world .

- Back-propagate gradients through both encoder and decoder

Training seq2seq models



$J = \frac{1}{T}\sum_{t=1}^{T} J_t = J_1 + J_2 + J_3 + J_4 + J_5 + J_6 + J_7$
= negative log prob of "he"
= negative log prob of "with"
= negative log prob of <END>
$\hat{y}_1$
$\hat{y}_2$
$\hat{y}_3$
$\hat{y}_4$
$\hat{y}_5$
$\hat{y}_6$
$\hat{y}_7$
Encoder RNN
Decoder RNN
il
a
m'
entarté
<bos>
<START>
he
hit
me
with
a
pie
Source sentence (from corpus)
Target sentence (from corpus)

Seq2seq is optimized as a **single system.**
Backpropagation operates *"end-to-end"*.

Decoding seq2seq models

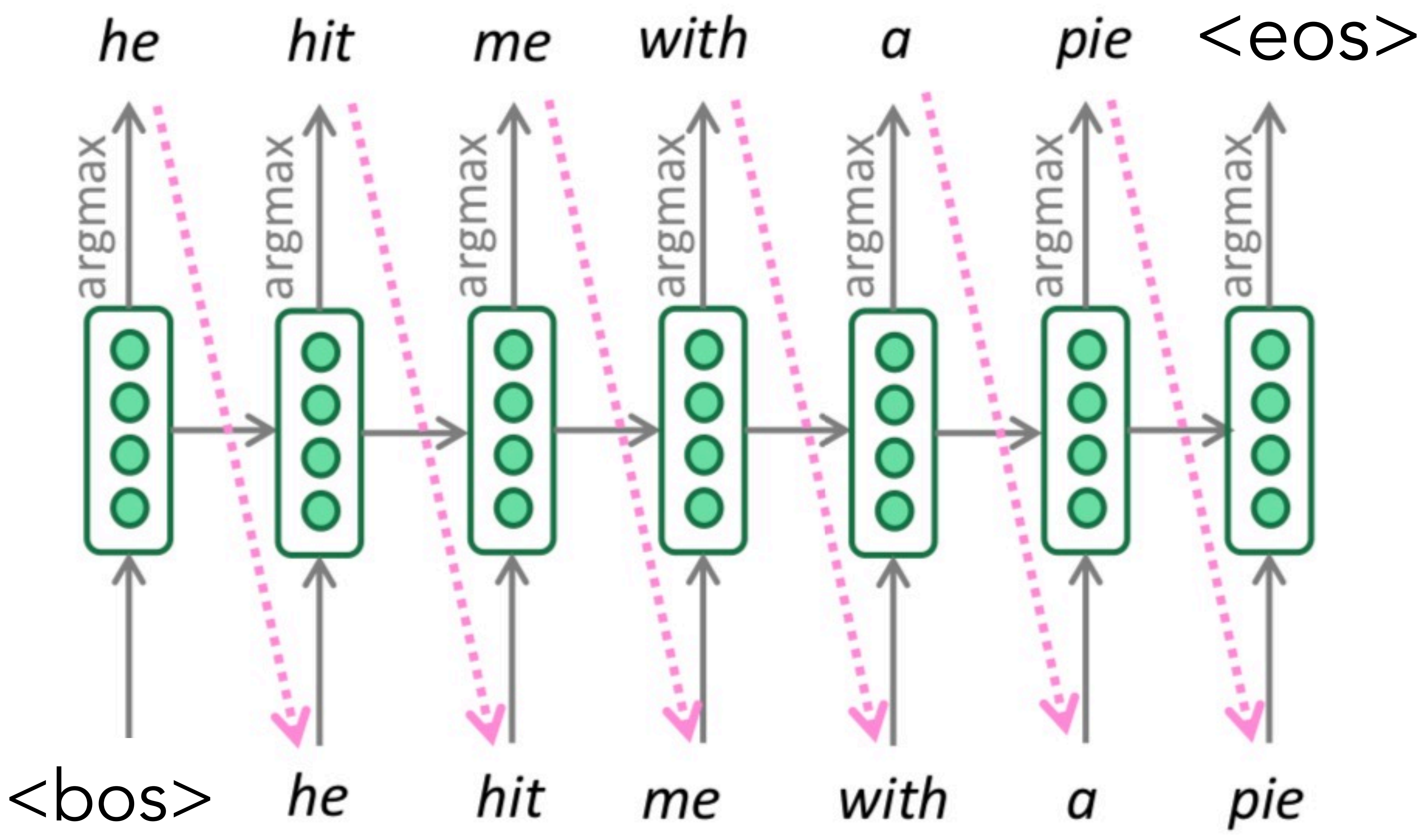
$$\arg\max_{y_1,\ldots,y_T} P(y_1,\ldots,y_T \,|\, \mathbf{w}^{(s)})$$

- Problem: Exhaustive search is very expensive — we even don’t know what T is
- Need some approximation!

Decoding seq2seq models

- Greedy decoding

  = Compute argmax at every step of decoder to generate word



A middle ground: Beam search

- At every step, keep track of the k most probable partial translations (hypotheses)

- Score of each hypothesis = log probability of sequence so far

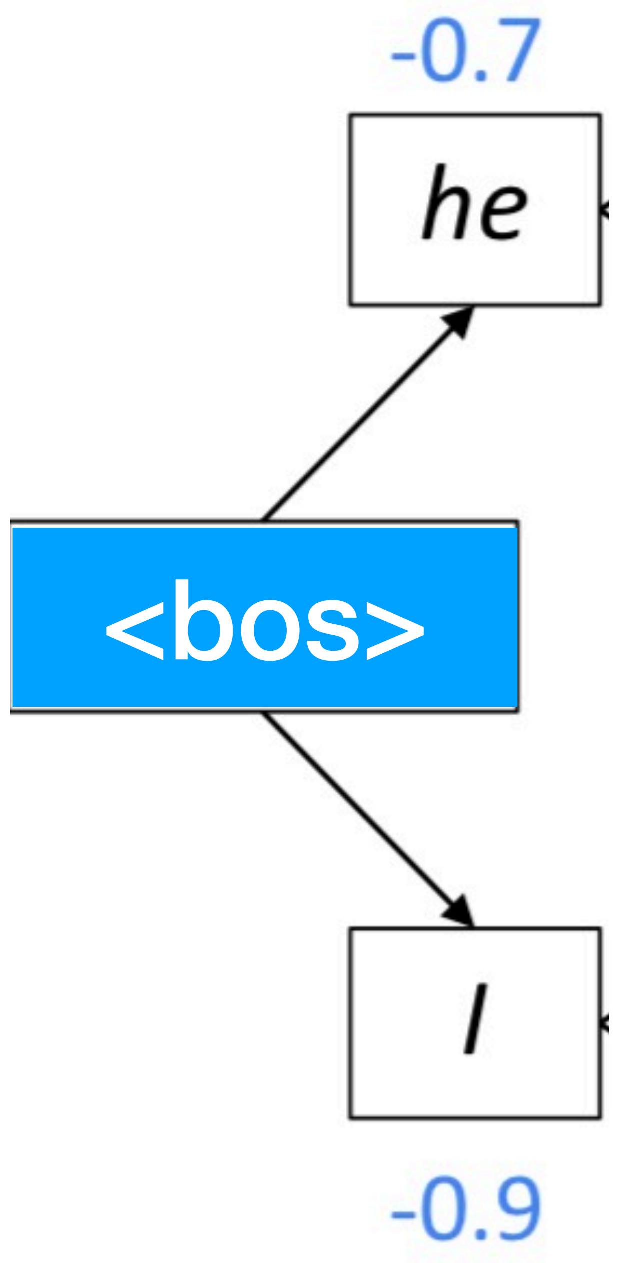
$$\sum_{i=1}^{t} \log P(y_i \,|\, y_1, \ldots, y_{i-1}, \mathbf{w}^{(s)})$$

- Not guaranteed to be optimal

- More efficient than exhaustive search

Beam search

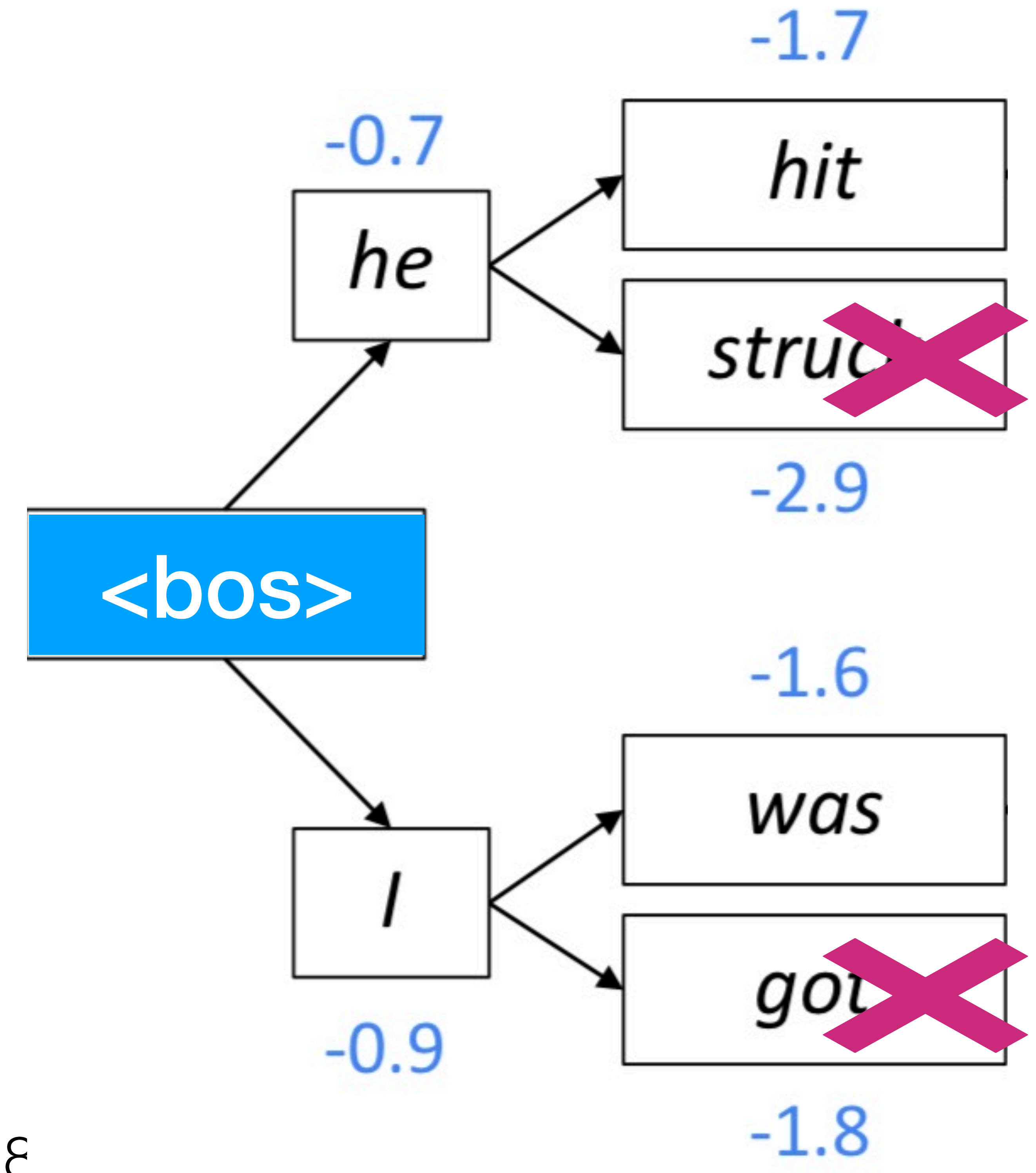
Beam size = k = 2. Blue numbers = $\text{score}(y_1, \ldots, y_t) = \sum_{i=1}^{t} \log P_{\text{LM}}(y_i | y_1, \ldots, y_{i-1}, x)$



(slide credit: Abigail See)

Beam search

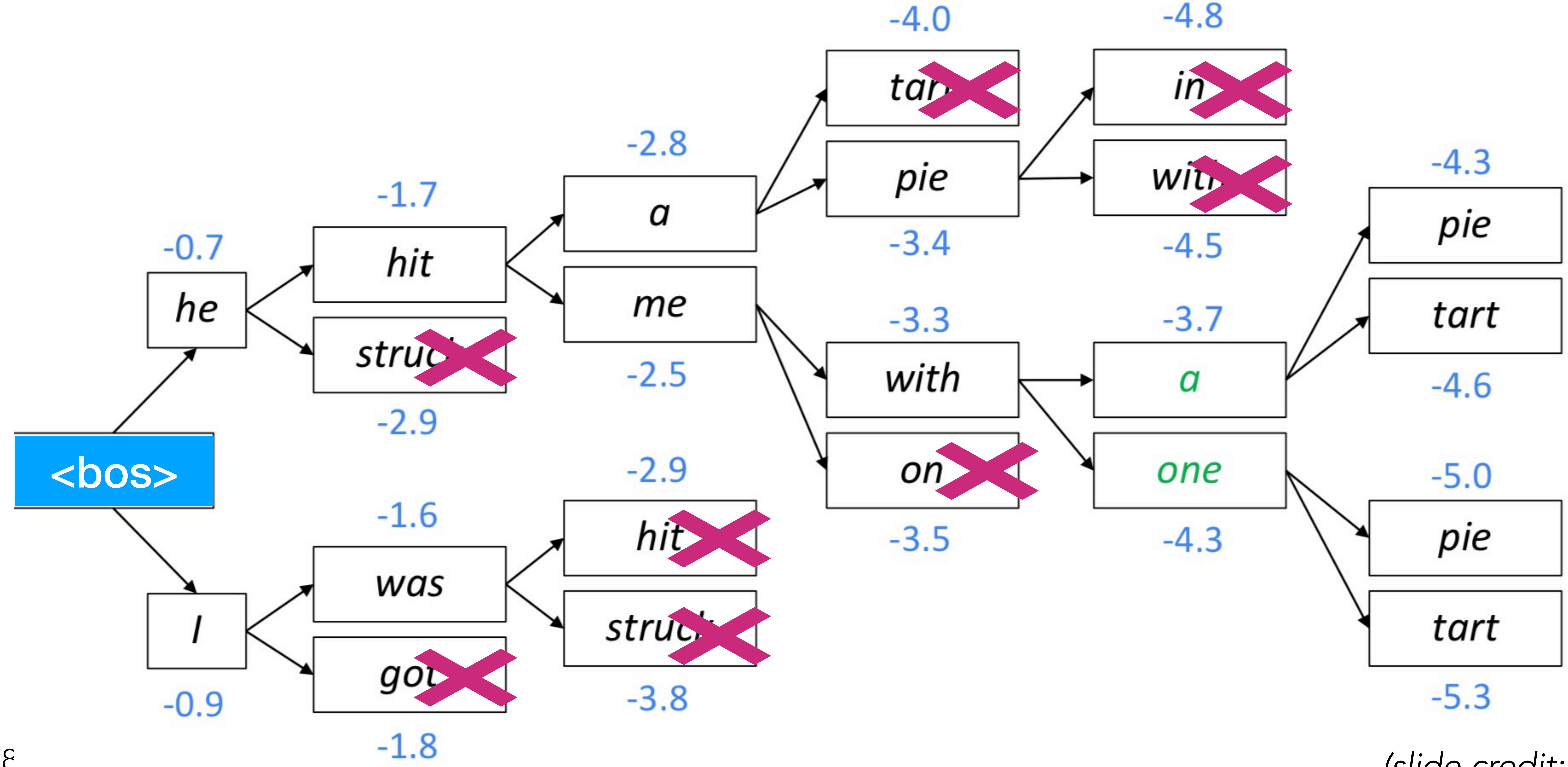
Beam size = k = 2. Blue numbers = $\text{score}(y_1, \ldots, y_t) = \sum_{i=1}^{t} \log P_{\text{LM}}(y_i | y_1, \ldots, y_{i-1}, x)$



(slide credit: Abigail See)

Beam search

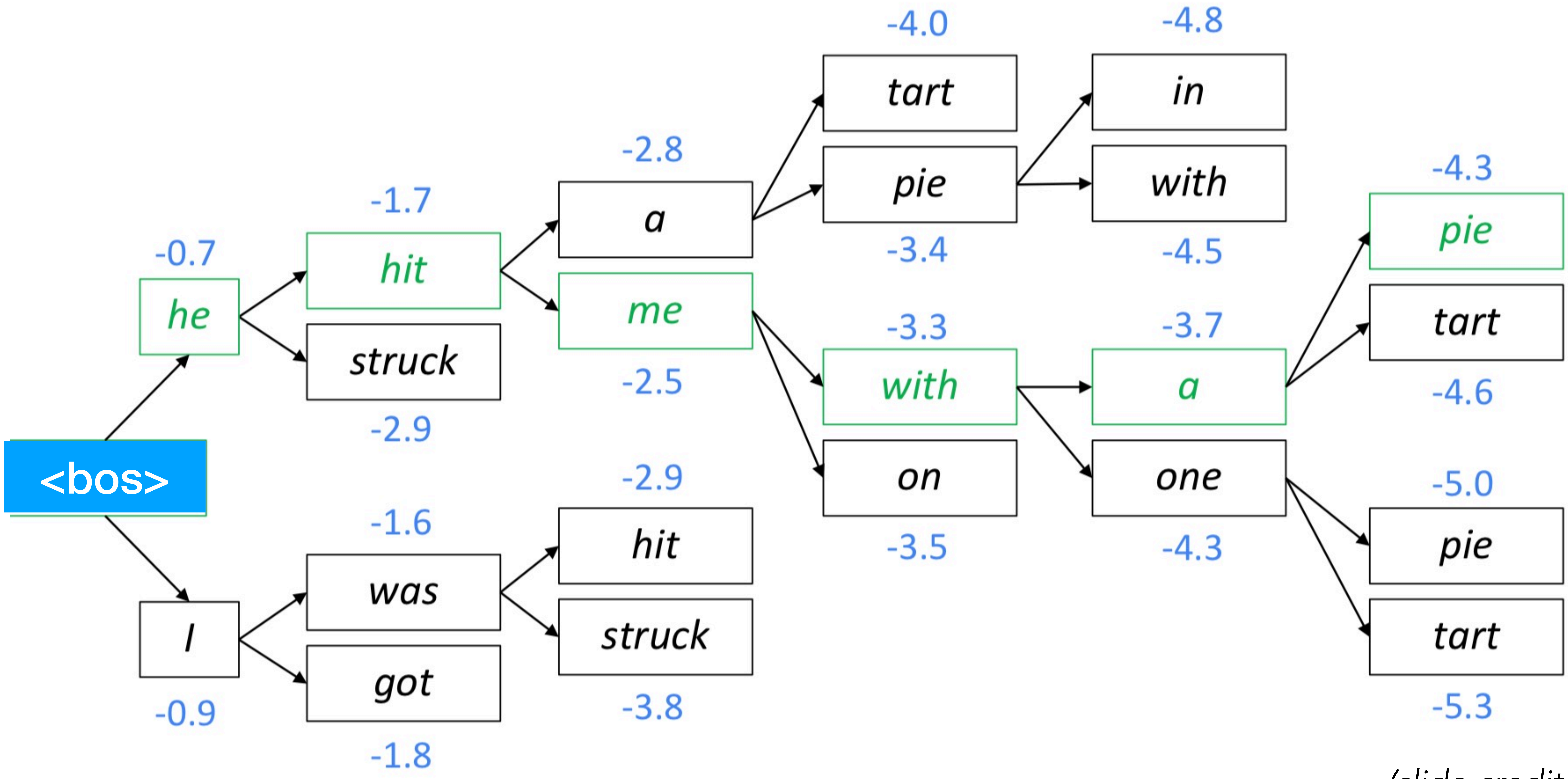
Beam size = k = 2. Blue numbers = $\text{score}(y_1, \ldots, y_t) = \sum_{i=1}^{t} \log P_{\text{LM}}(y_i | y_1, \ldots, y_{i-1}, x)$



(slide credit: Abigail See)

Backtrack

Beam size = k = 2. Blue numbers = $\text{score}(y_1, \ldots, y_t) = \sum_{i=1}^{t} \log P_{\text{LM}}(y_i | y_1, \ldots, y_{i-1}, x)$



(slide credit: Abigail See)

Beam search: Details

- Different hypotheses may produce $\langle eos \rangle$ token at different time steps
  - When a hypothesis produces $\langle eos \rangle$, stop expanding it and place it aside
- Continue beam search until:
  - All k hypotheses produce $\langle eos \rangle$ OR
  - Hit max decoding limit T
- Select top hypotheses using the *normalized* likelihood score

$$\frac{1}{T}\sum_{t=1}^{T}\log P(y_t \,|\, y_1, \ldots, y_{t-1}, \mathbf{w}^{(s)})$$

  - Otherwise shorter hypotheses have higher scores

NMT vs. SMT

Pros:

- Better performance (more **fluent**, better use of **context**, better use of **phrase similarities**)
- A **single neural network** to be optimized end-to-end (no individual subcomponents)
- **Less human engineering effort** - same method for all language pairs

Cons:

- NMT is **less interpretable**
- NMT is **difficult to control**

NMT: the first big success of NLP deep learning

- 2014: First seq2seq paper published
- 2016: Google Translate switches from SMT to NMT - and by 2018 everyone has

Microsoft SYSTRAN beyond language Google facebook

Baidu 百度 網易 NetEase www.163.com Tencent 腾讯 搜狗搜索

- SMT systems, built by hundreds of engineers over many years, outperformed by NMT systems trained by a small group of engineers in a few months

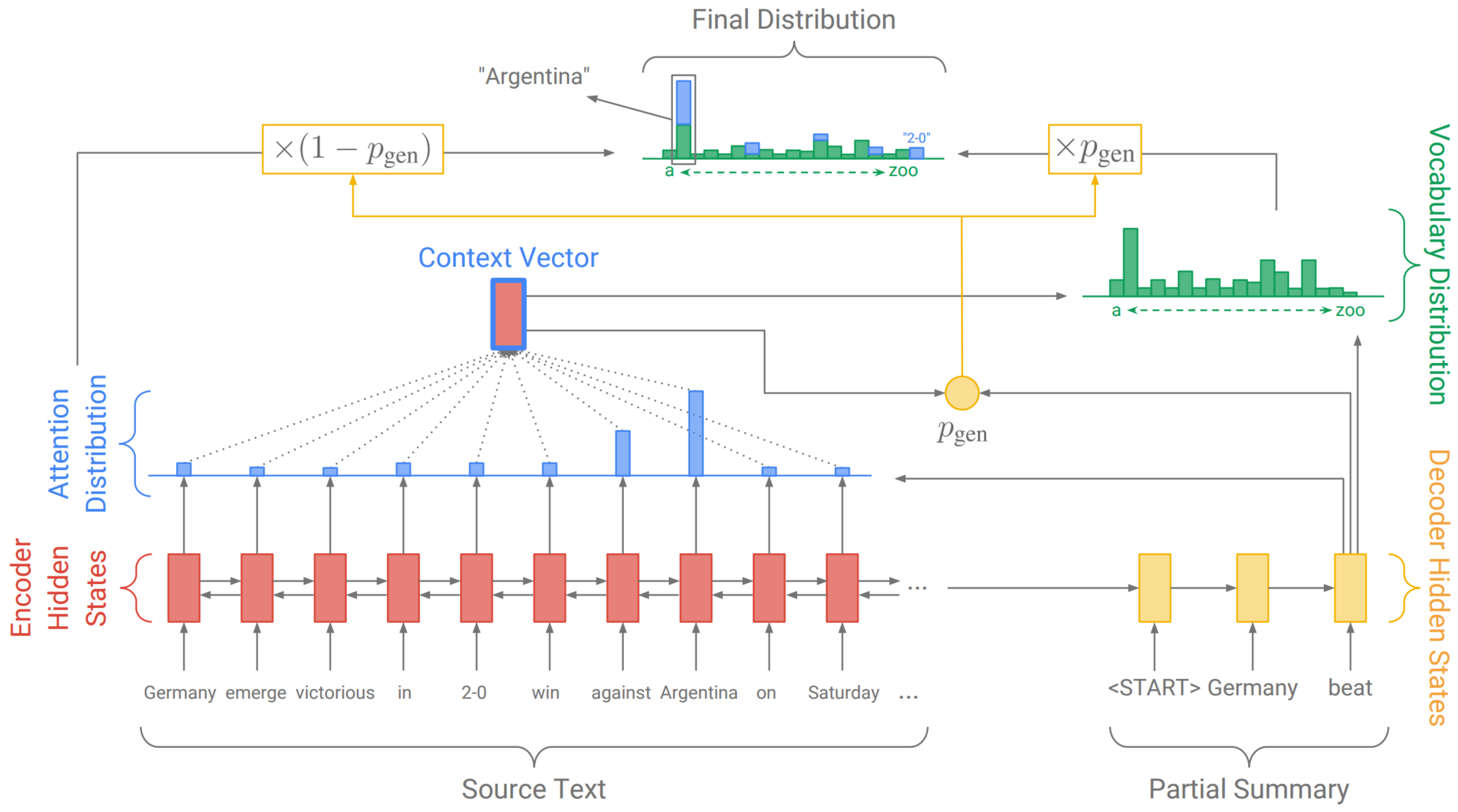
Other applications of seq2seq

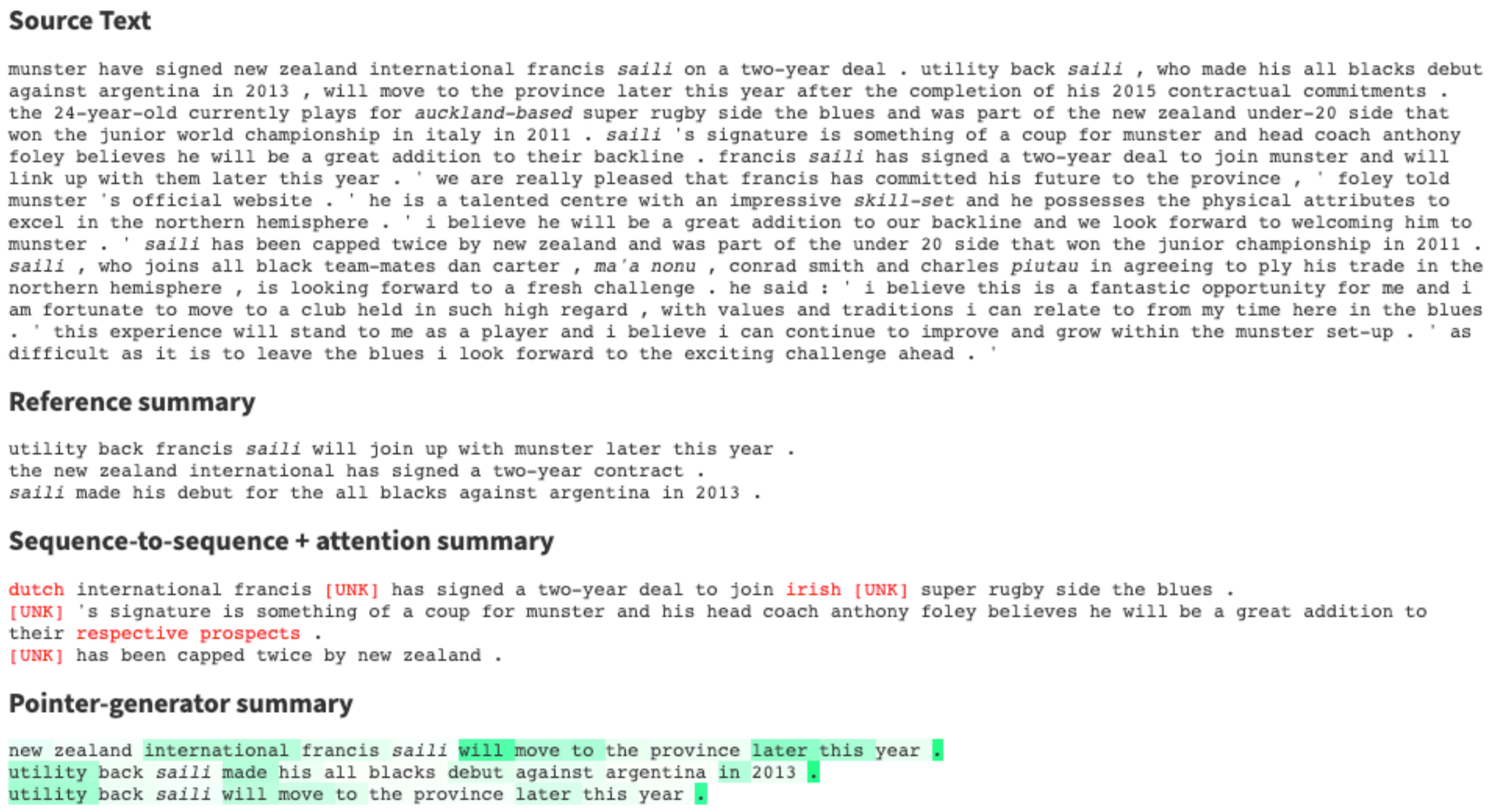
Seq2seq is versatile

- Sequence-to-sequence is useful for more than just MT
- Many NLP tasks can be framed as sequence-to-sequence problems
    - **Summarization** (long text → short text)
    - **Dialogue** (previous utterances → next utterance)
    - **Code generation** (natural language → Python code)
    - …

Seq2seq is versatile

- Summarization





Source Text

munster have signed new zealand international francis *saili* on a two-year deal . utility back *saili* , who made his all blacks debut against argentina in 2013 , will move to the province later this year after the completion of his 2015 contractual commitments . the 24-year-old currently plays for *auckland-based* super rugby side the blues and was part of the new zealand under-20 side that won the junior world championship in italy in 2011 . *saili* 's signature is something of a coup for munster and head coach anthony foley believes he will be a great addition to their backline . francis *saili* has signed a two-year deal to join munster and will link up with them later this year . ' we are really pleased that francis has committed his future to the province , ' foley told munster 's official website . ' he is a talented centre with an impressive *skill-set* and he possesses the physical attributes to excel in the northern hemisphere . ' i believe he will be a great addition to our backline and we look forward to welcoming him to munster . ' *saili* has been capped twice by new zealand and was part of the under 20 side that won the junior championship in 2011 . *saili* , who joins all black team-mates dan carter , *ma'a nonu* , conrad smith and charles *piutau* in agreeing to ply his trade in the northern hemisphere , is looking forward to a fresh challenge . he said : ' i believe this is a fantastic opportunity for me and i am fortunate to move to a club held in such high regard , with values and traditions i can relate to from my time here in the blues . ' this experience will stand to me as a player and i believe i can continue to improve and grow within the munster set-up . ' as difficult as it is to leave the blues i look forward to the exciting challenge ahead . '

Reference summary

utility back francis *saili* will join up with munster later this year .
the new zealand international has signed a two-year contract .
saili made his debut for the all blacks against argentina in 2013 .

Sequence-to-sequence + attention summary

dutch international francis [UNK] has signed a two-year deal to join irish [UNK] super rugby side the blues .
[UNK] 's signature is something of a coup for munster and his head coach anthony foley believes he will be a great addition to their respective prospects .
[UNK] has been capped twice by new zealand .

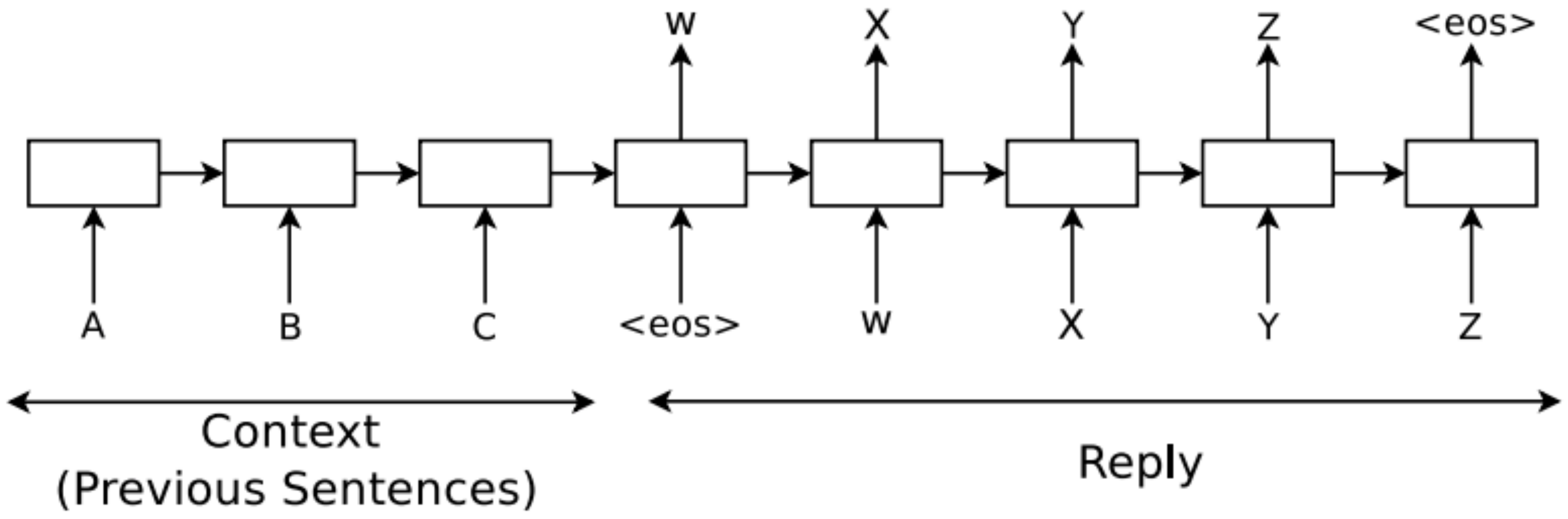
Pointer-generator summary

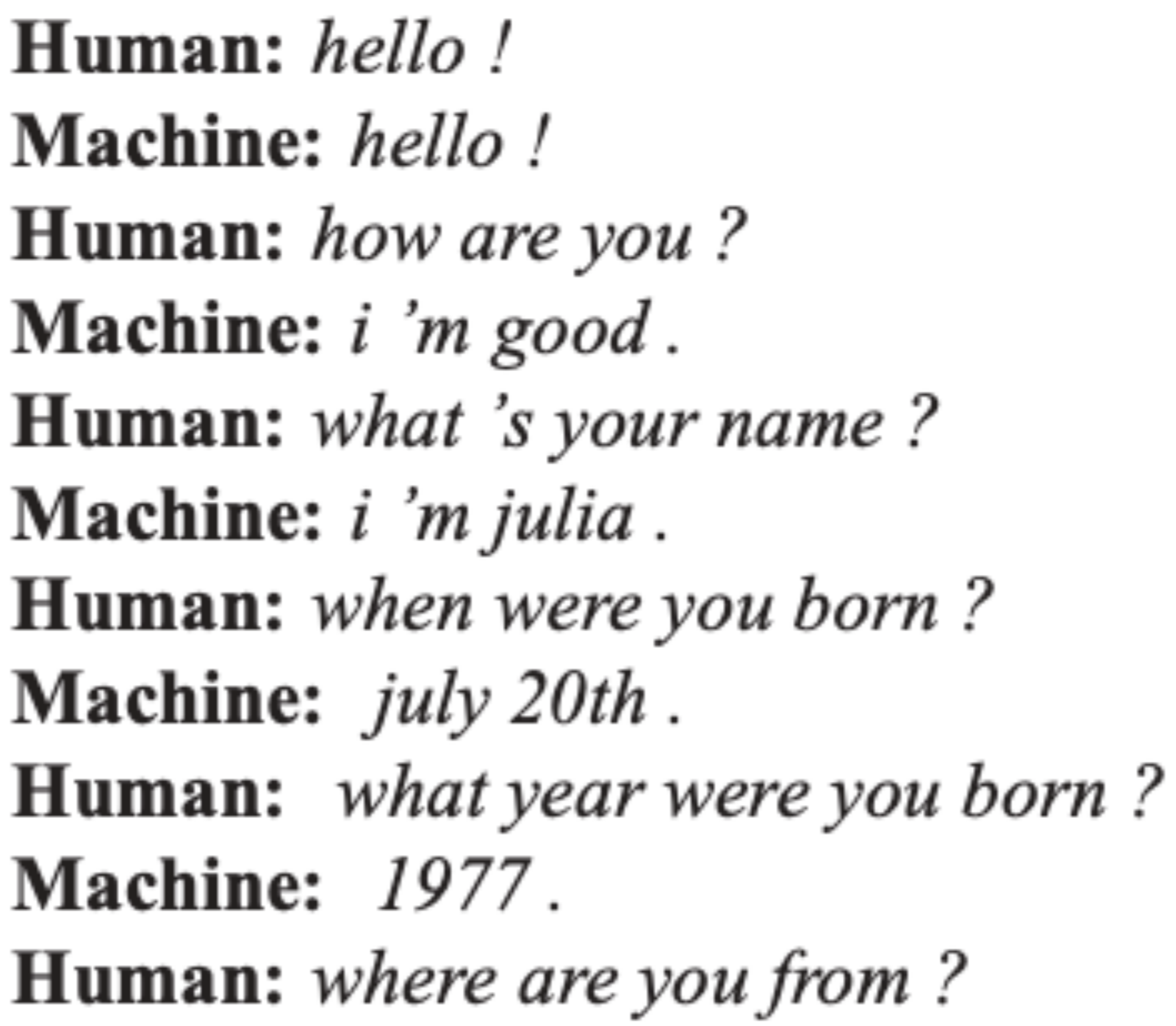
new zealand international francis *saili* will move to the province later this year .
utility back *saili* made his all blacks debut against argentina in 2013 .
utility back *saili* will move to the province later this year .

See et al., 2017: Get To The Point: Summarization with Pointer-Generator Networks

Seq2seq is versatile

- Dialogue



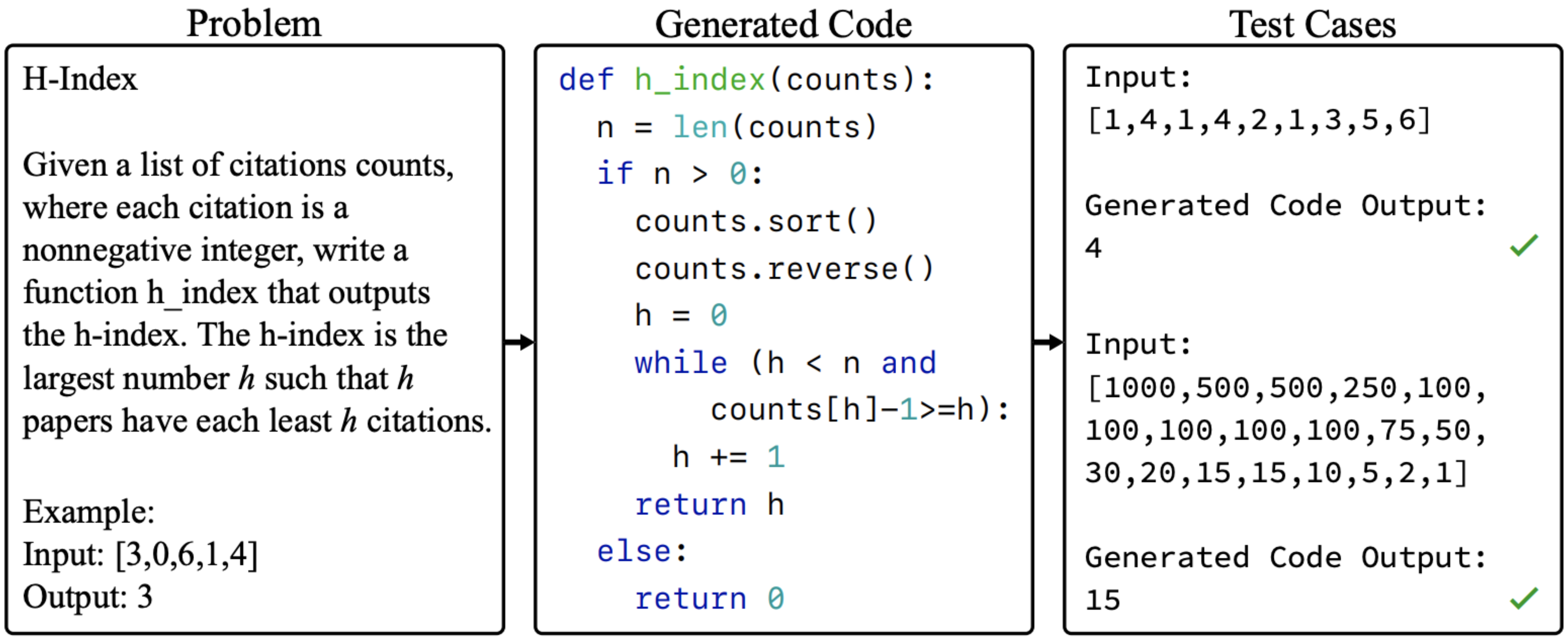


Human: *hello !*
Machine: *hello !*
Human: *how are you ?*
Machine: *i 'm good .*
Human: *what 's your name ?*
Machine: *i 'm julia .*
Human: *when were you born ?*
Machine: *july 20th .*
Human: *what year were you born ?*
Machine: *1977 .*
Human: *where are you from ?*

Vinyals and Le 2015: A Neuarl Conversational Model

Seq2seq is versatile

- Code generation



Problem

H-Index

Given a list of citations counts, where each citation is a nonnegative integer, write a function h_index that outputs the h-index. The h-index is the largest number *h* such that *h* papers have each least *h* citations.

Example:
Input: [3,0,6,1,4]
Output: 3

Generated Code

```
def h_index(counts):
 n = len(counts)
 if n > 0:
  counts.sort()
  counts.reverse()
  h = 0
  while (h < n and
    counts[h]-1>=h):
   h += 1
  return h
 else:
  return 0
```

Test Cases

```
Input:
[1,4,1,4,2,1,3,5,6]

Generated Code Output:
4
```

✓

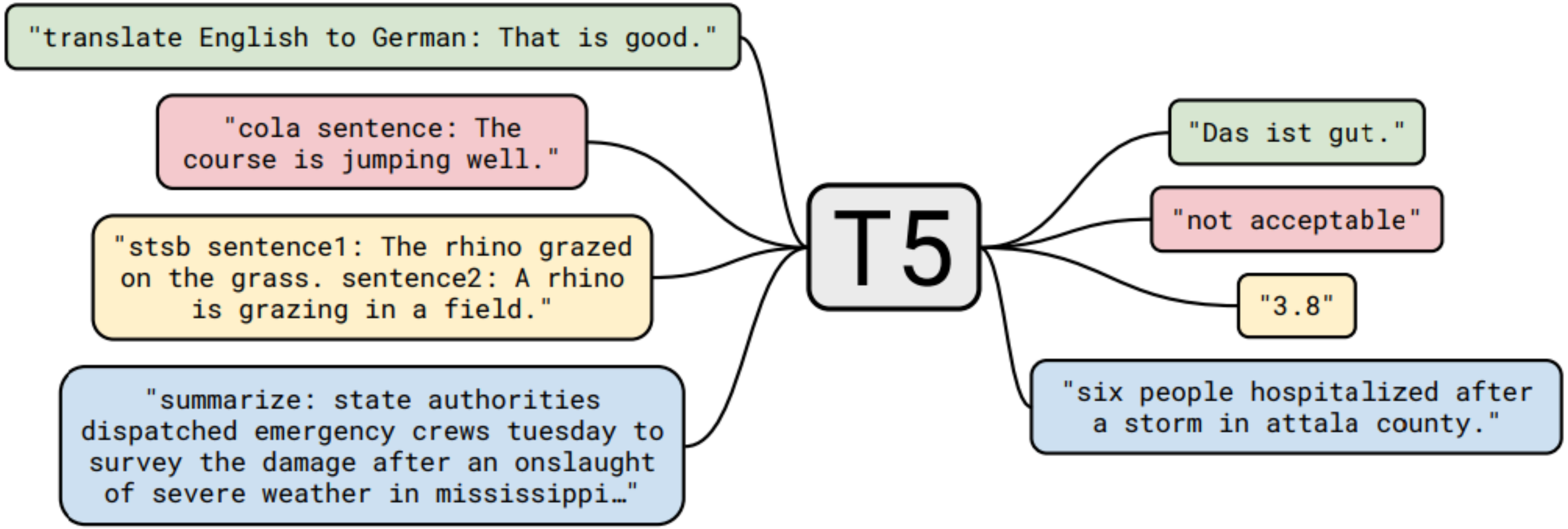
```
Input:
[1000,500,500,250,100,
100,100,100,100,75,50,
30,20,15,15,10,5,2,1]

Generated Code Output:
15
```

✓

Seq2seq is versatile

- All language tasks can be converted into a text-to-text problem!
    - T5 = **T**ext-**t**o-**t**ext **T**rasnfer **T**ransformer



"translate English to German: That is good."

"cola sentence: The course is jumping well."

"stsb sentence1: The rhino grazed on the grass. sentence2: A rhino is grazing in a field."

"summarize: state authorities dispatched emergency crews tuesday to survey the damage after an onslaught of severe weather in mississippi…"

"Das ist gut."

"not acceptable"

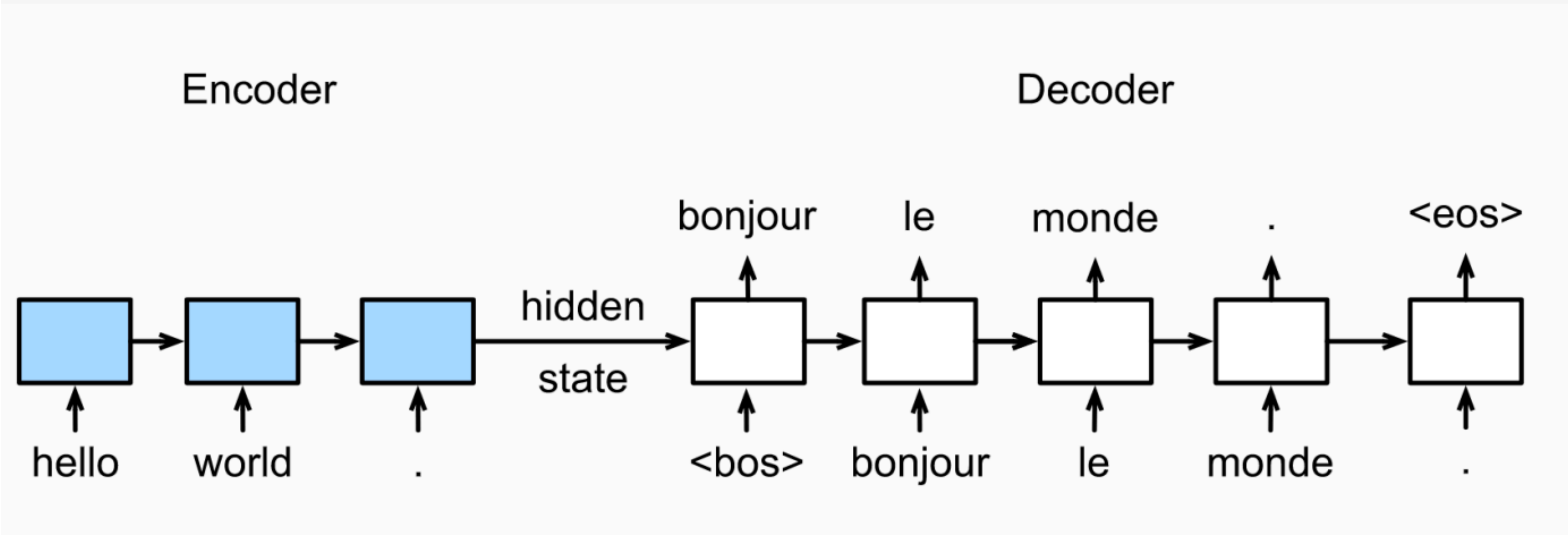
"3.8"

"six people hospitalized after a storm in attala county."

Raffel et al., 2019. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer

Attention

Seq2seq: the bottleneck

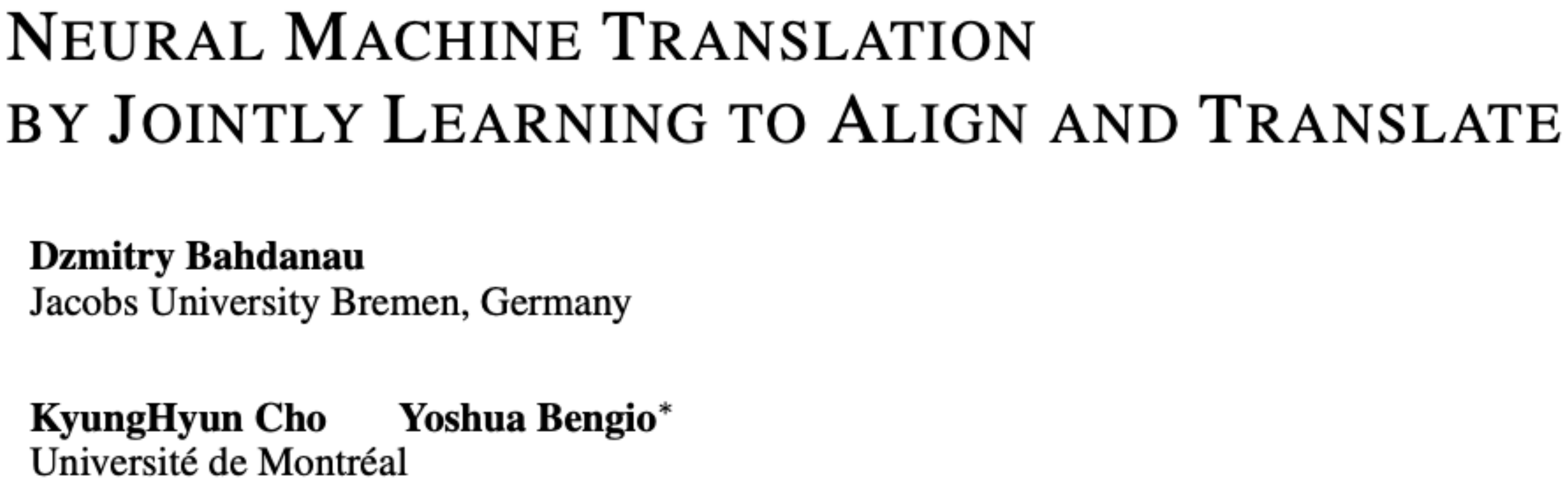


- A single encoding vector, h^{enc}, needs to capture **all the information** about source sentence
- Longer sequences can lead to vanishing gradients

Attention

- Attention provides a solution to the bottleneck problem

ICLR 2015

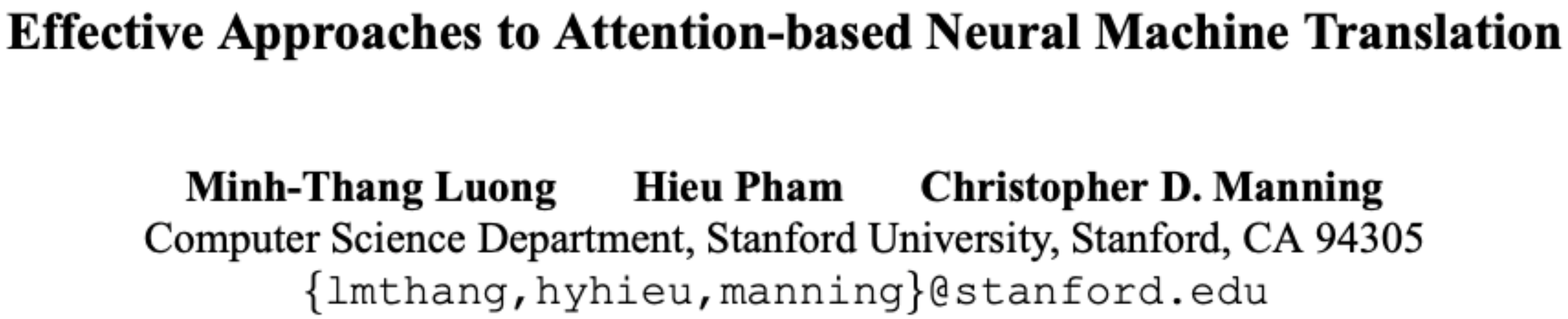


NEURAL MACHINE TRANSLATION
BY JOINTLY LEARNING TO ALIGN AND TRANSLATE

Dzmitry Bahdanau
Jacobs University Bremen, Germany

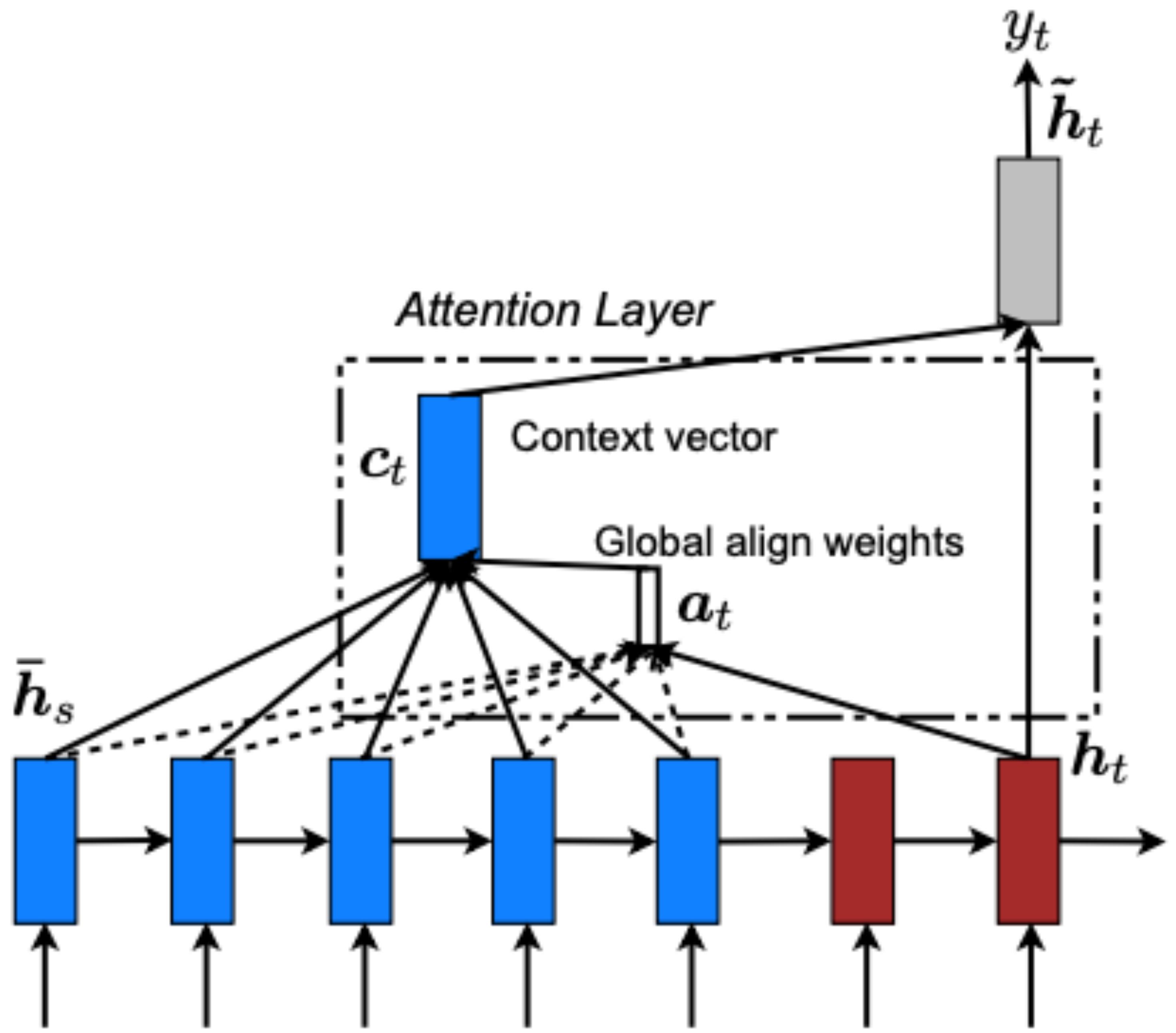
KyungHyun Cho **Yoshua Bengio***
Université de Montréal

EMNLP 2015



Effective Approaches to Attention-based Neural Machine Translation

Minh-Thang Luong **Hieu Pham** **Christopher D. Manning**
Computer Science Department, Stanford University, Stanford, CA 94305
{lmthang,hyhieu,manning}@stanford.edu



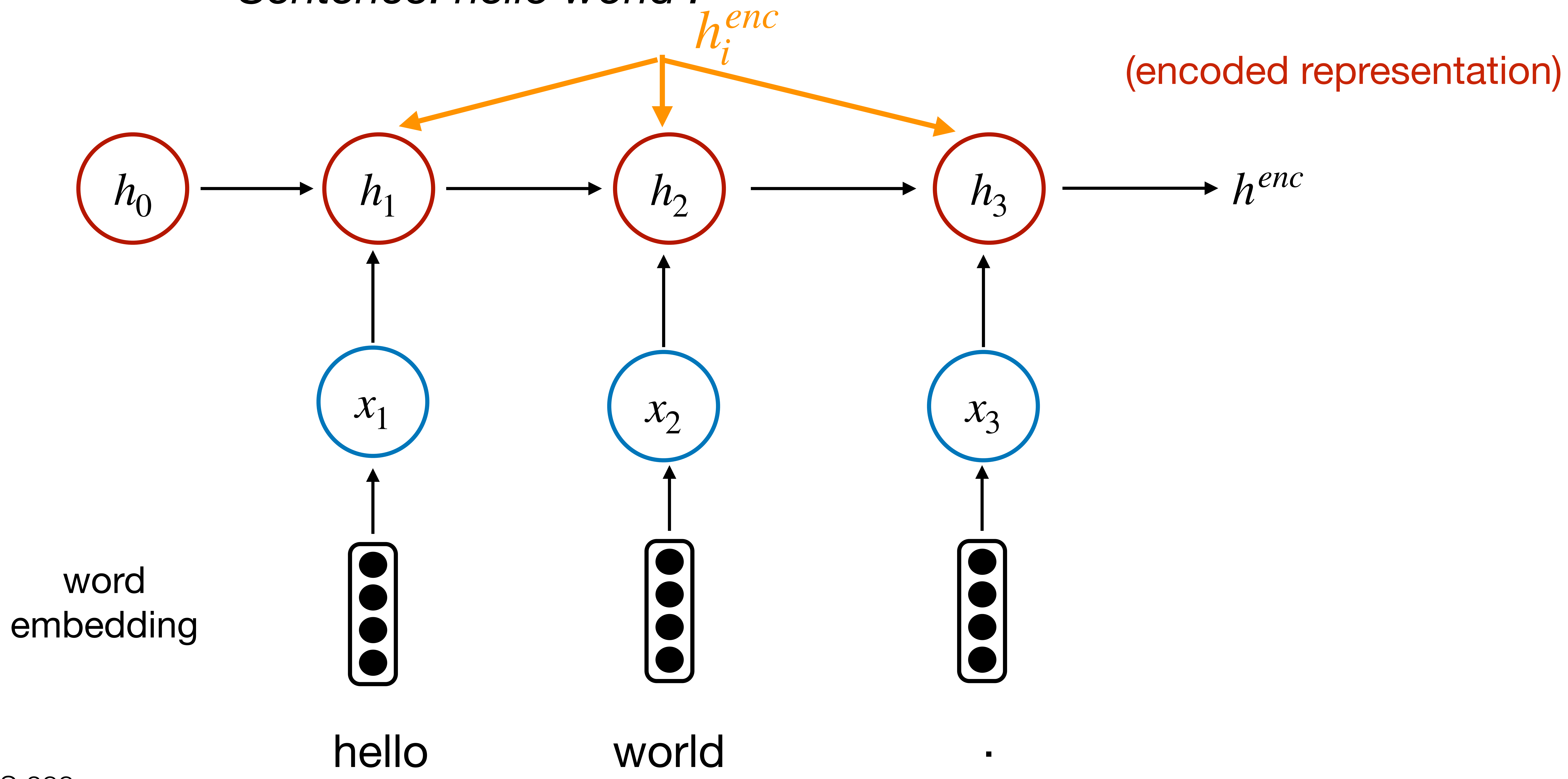
Attention

- Attention provides a solution to the bottleneck problem
- **Key idea:** At each time step during decoding, **focus on a particular part** of source sentence
  - This depends on the **decoder's** current hidden state h_t^{dec} (i.e. an idea of what you are trying to decode)
  - Usually implemented as a probability distribution over the hidden states of the **encoder** (h_i^{enc})

(Next lecture) Transformers = attention is all you need!

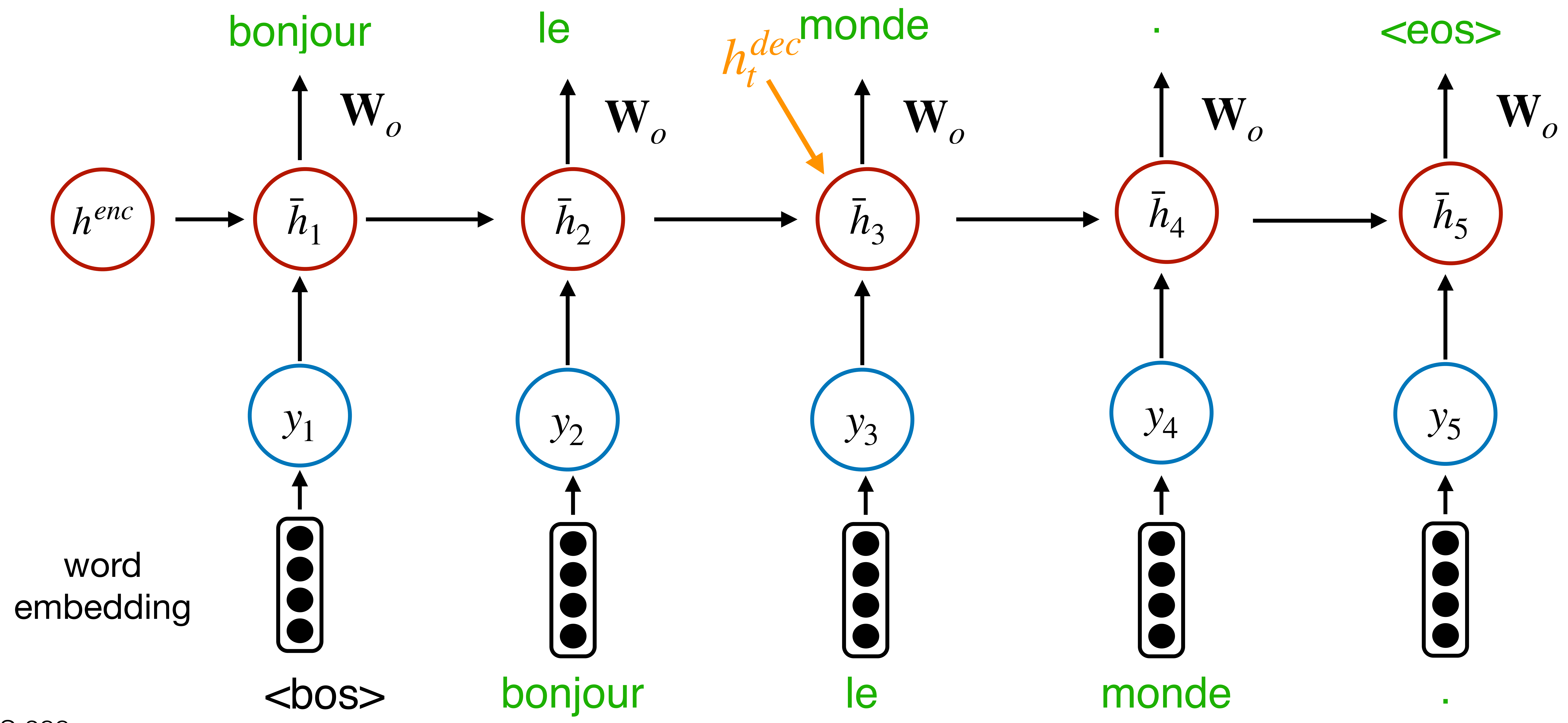
Recap: Seq2seq Encoder

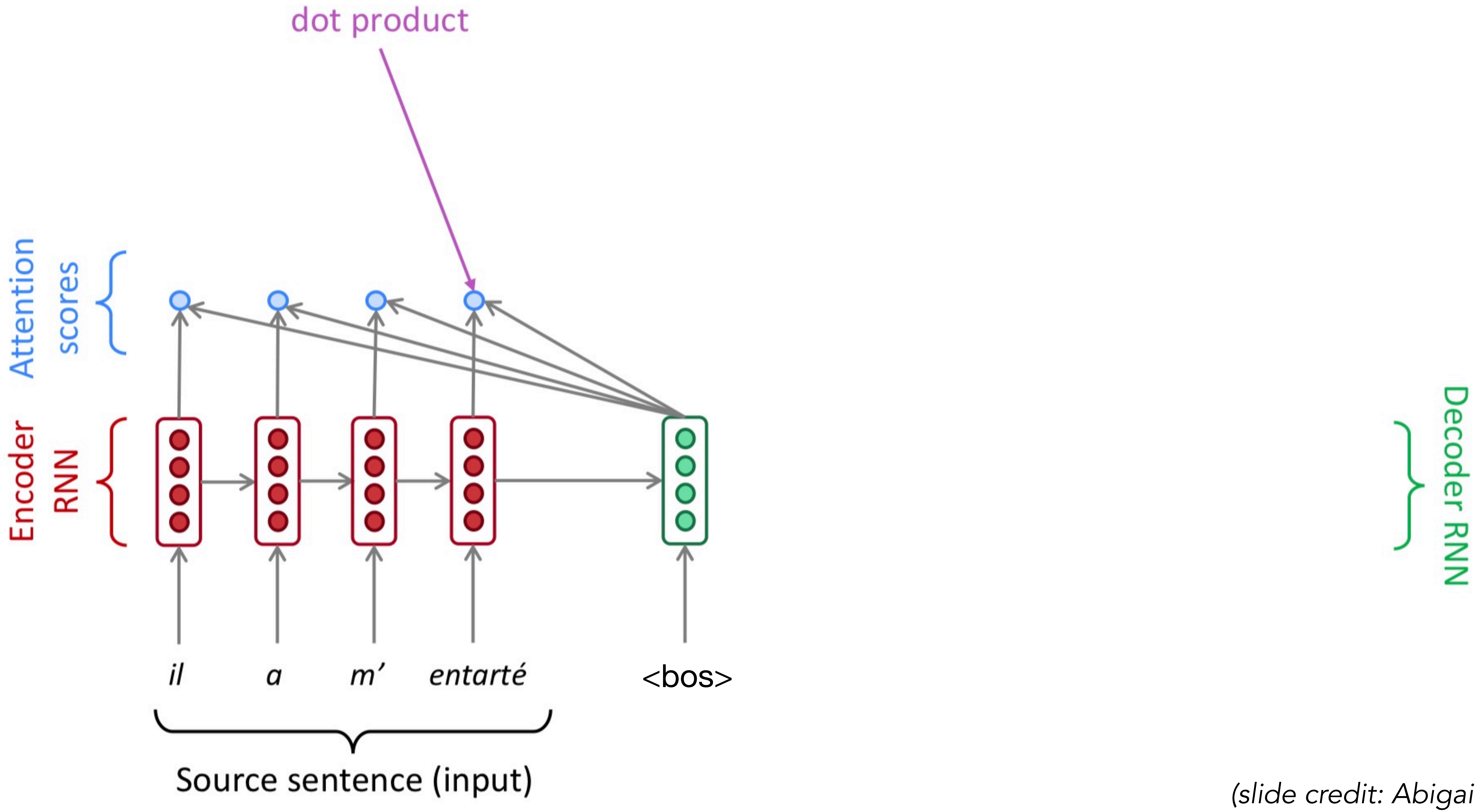
Sentence: hello world .



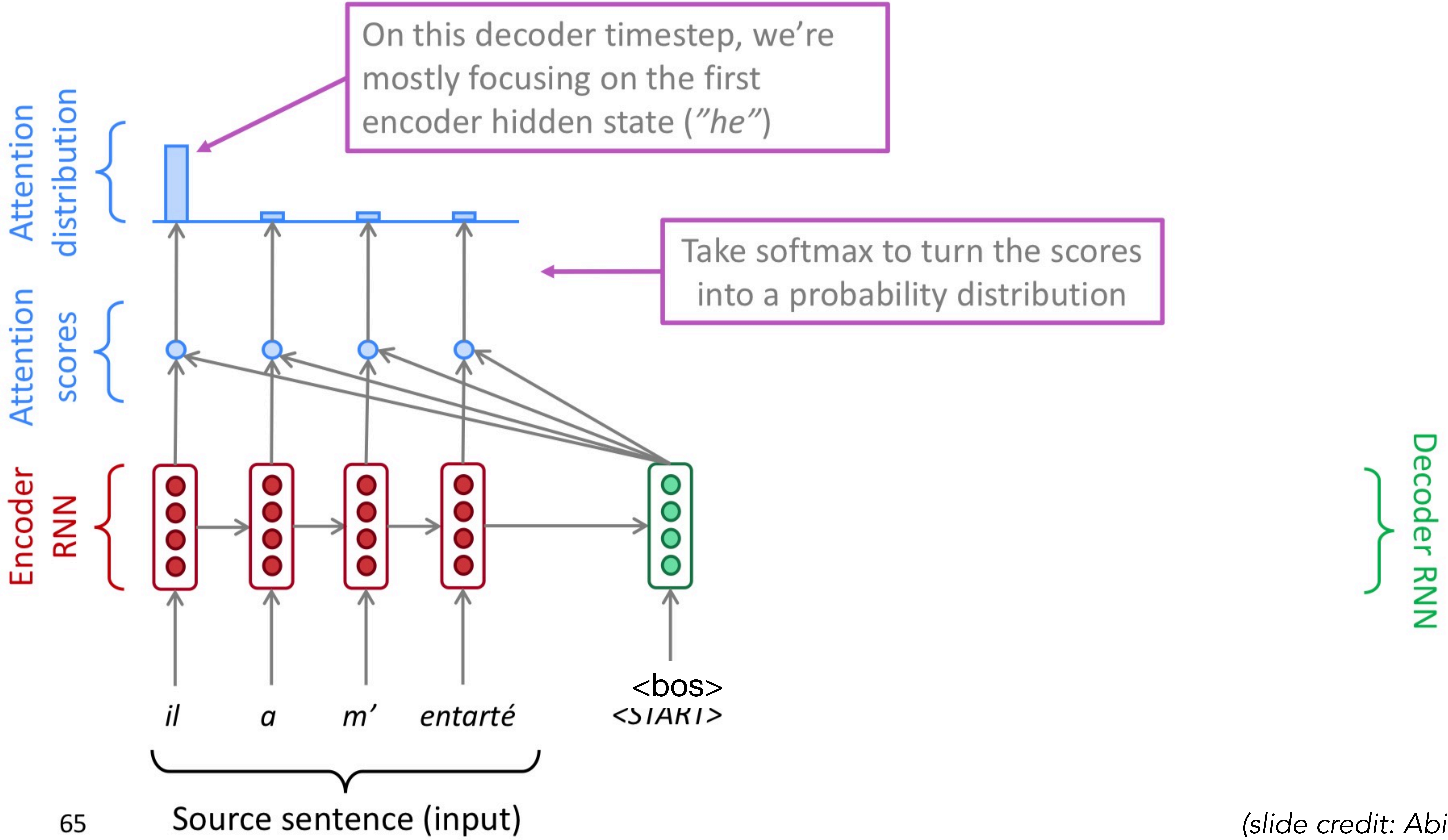
Recap: Seq2seq Decoder

- A **conditional** language model



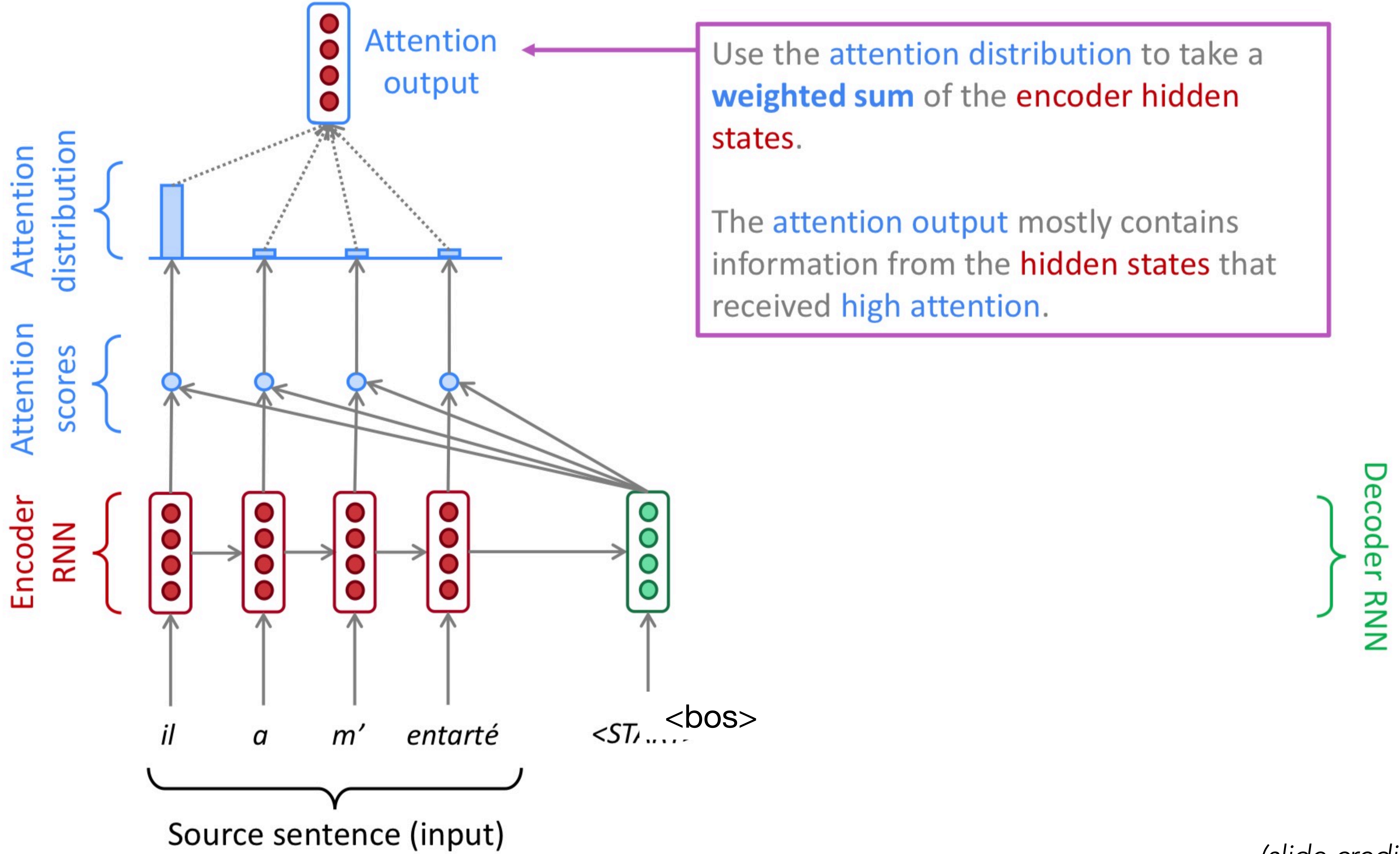


(slide credit: Abigail See)



On this decoder timestep, we're mostly focusing on the first encoder hidden state (*"he"*)
Take softmax to turn the scores into a probability distribution
Attention distribution
Attention scores
Encoder RNN
Decoder RNN
il
a
m'
entarté
<bos>
<START>
Source sentence (input)
65

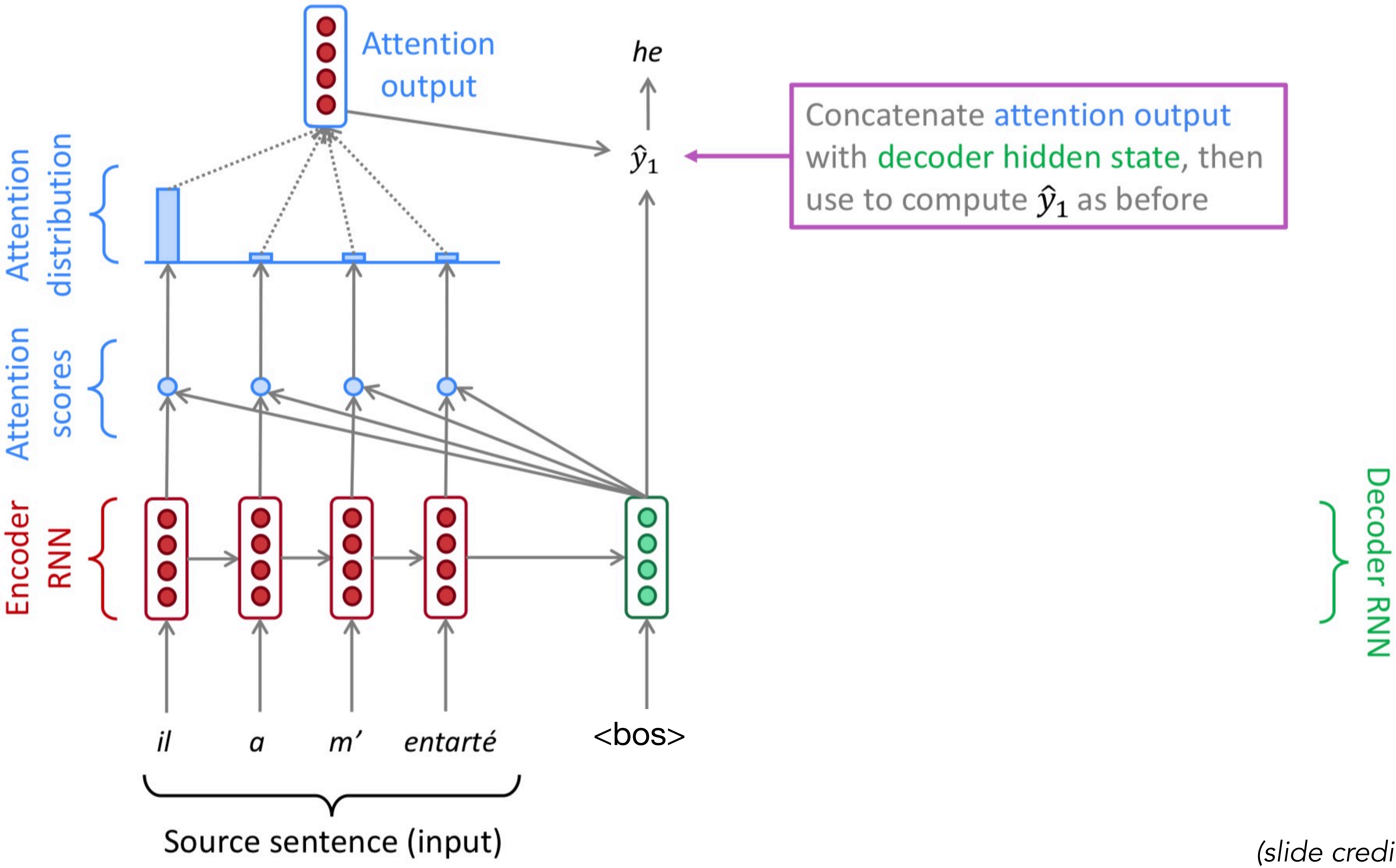
(slide credit: Abigail See)



Use the attention distribution to take a **weighted sum** of the encoder hidden states.

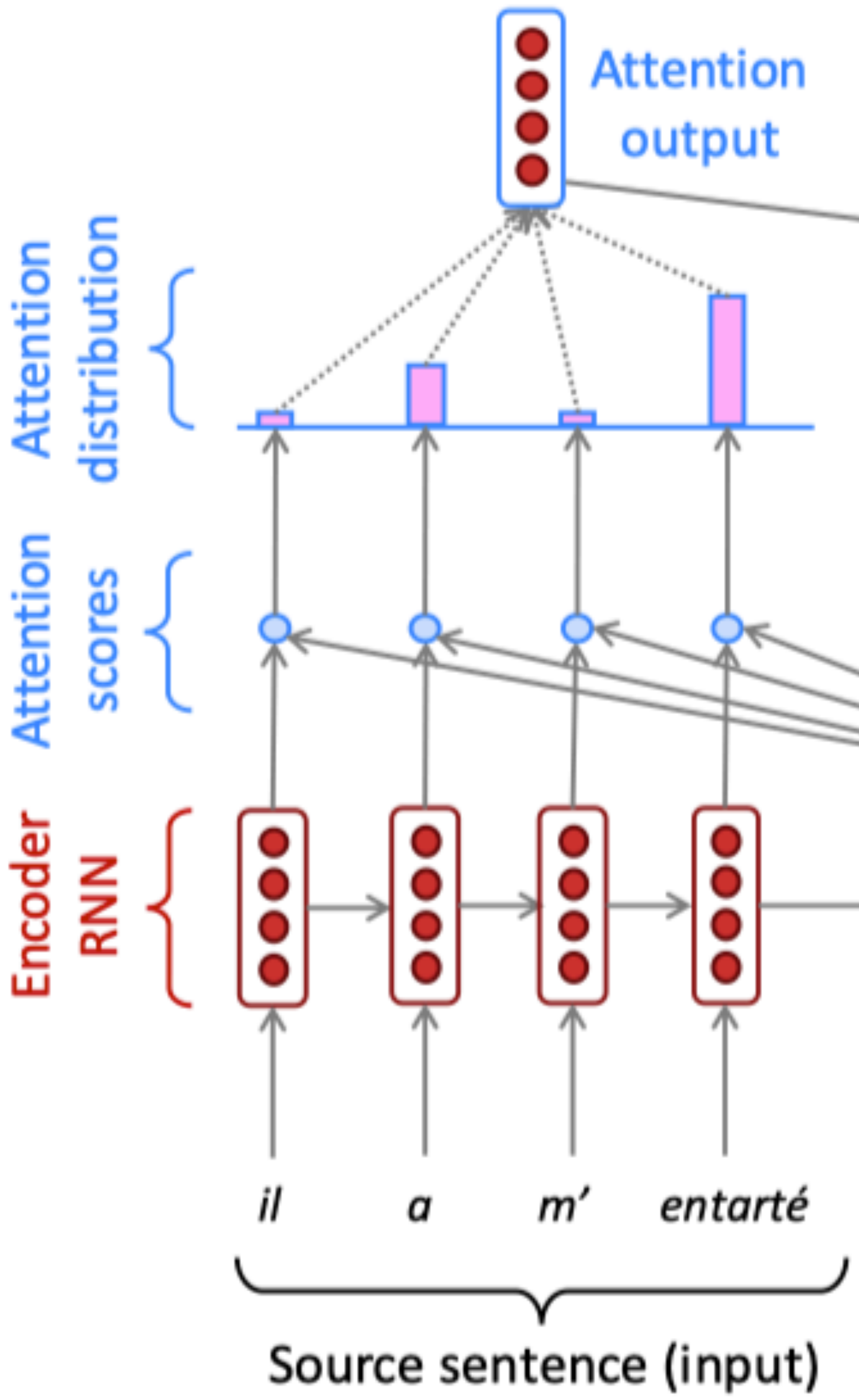
The attention output mostly contains information from the hidden states that received high attention.

(slide credit: Abigail See)

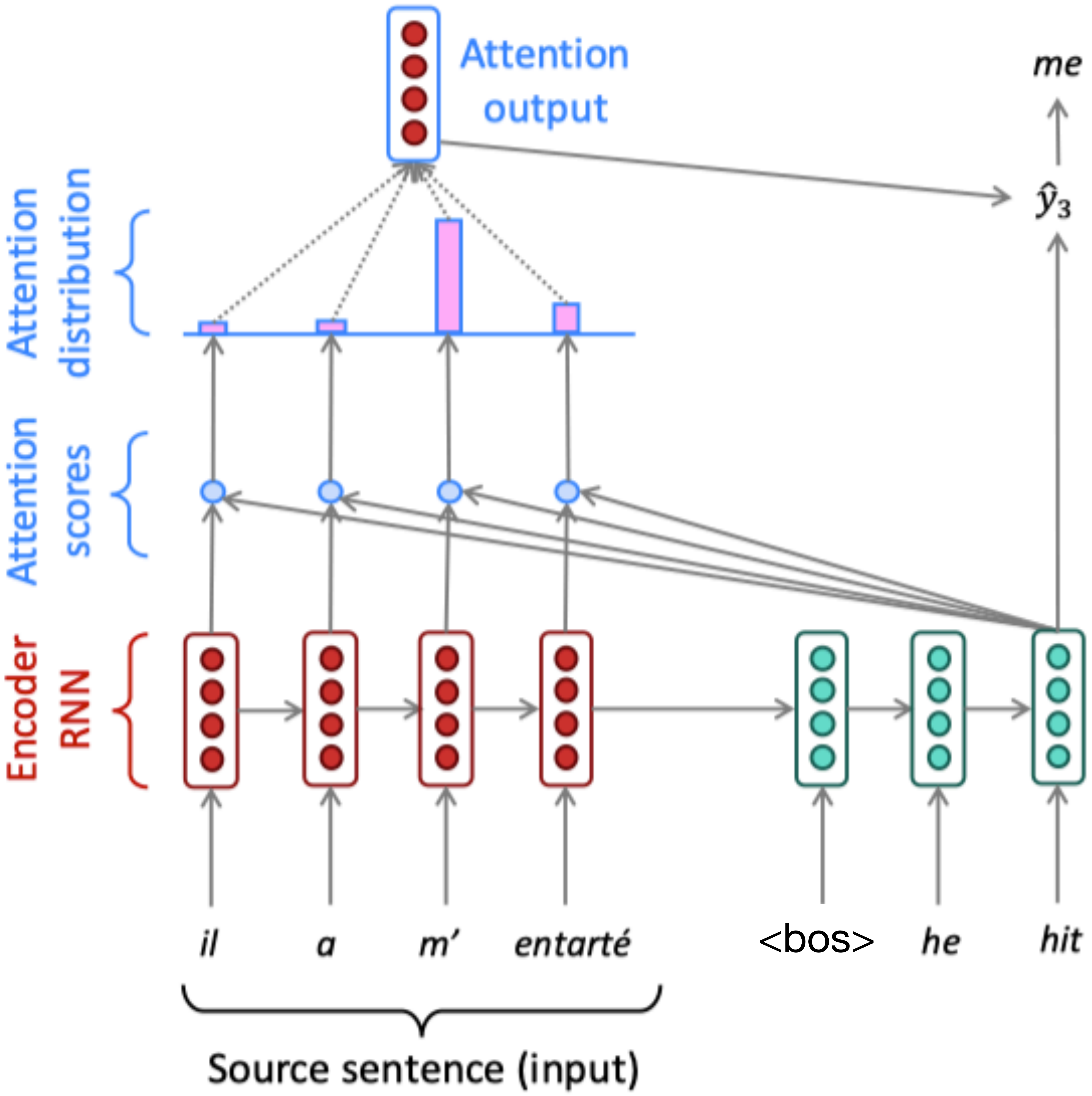


Concatenate attention output with decoder hidden state, then use to compute $\hat{y}_1$ as before

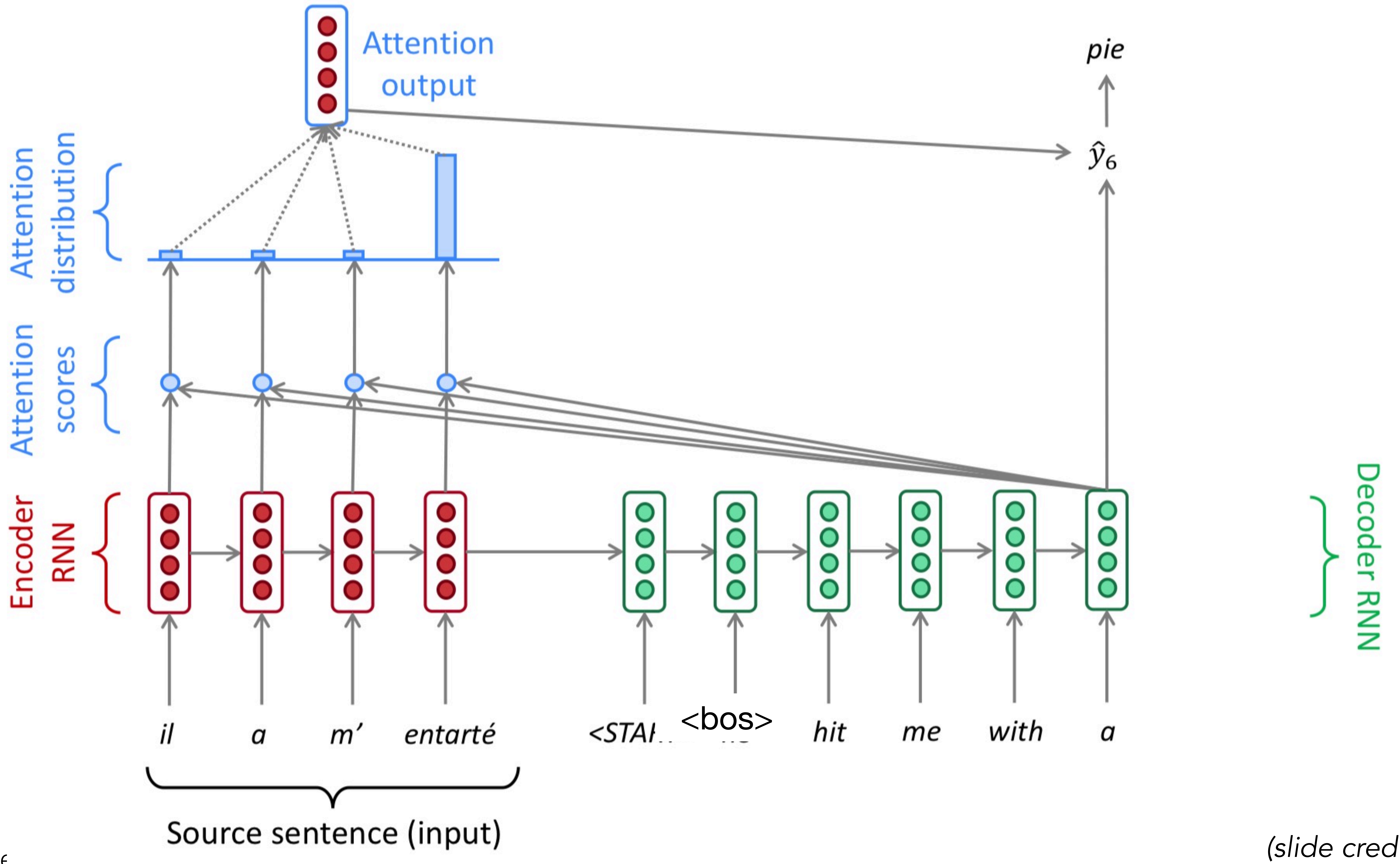
(slide credit: Abigail See)



(slide credit: Abigail See)

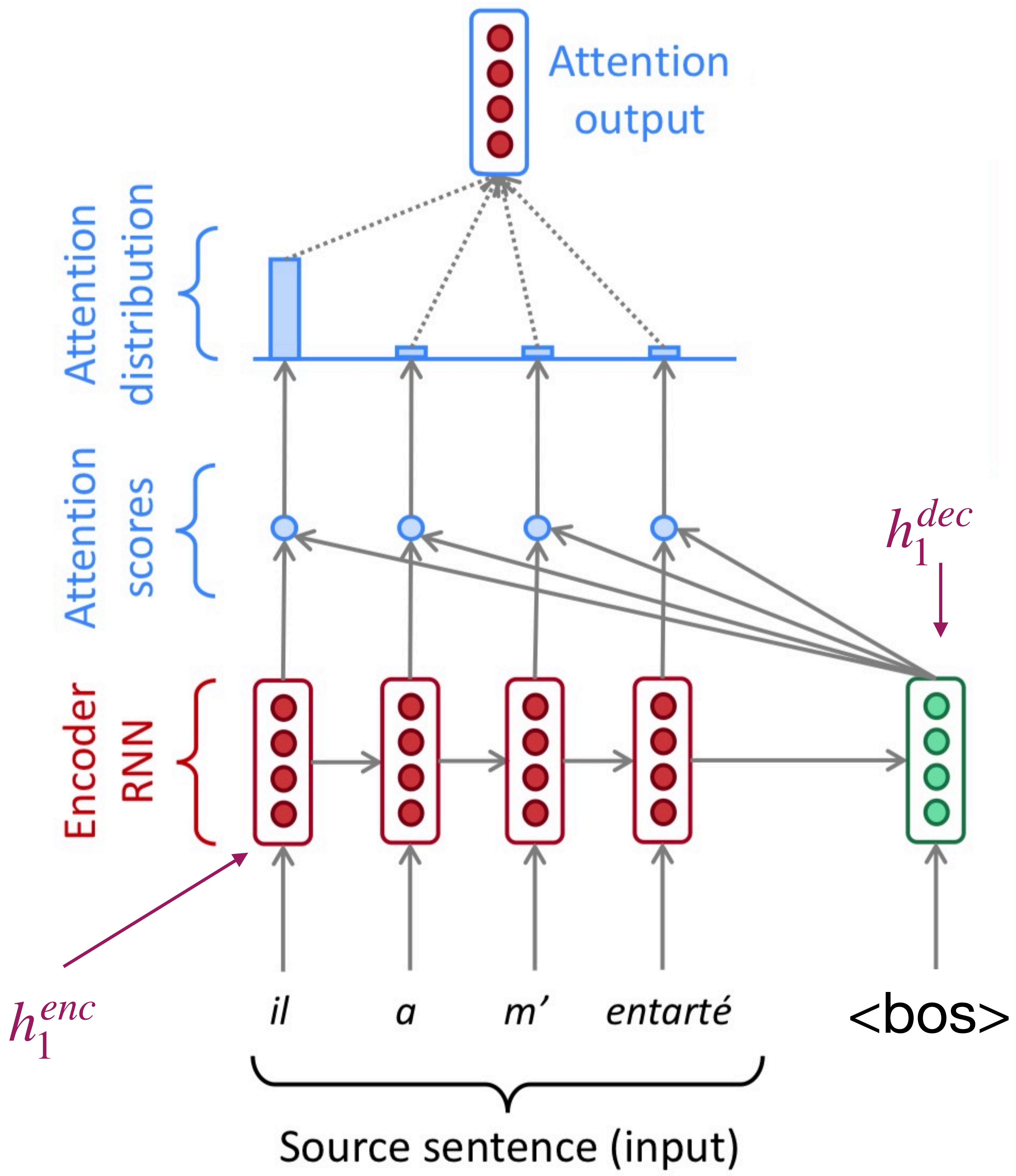


(slide credit: Abigail See)



(slide credit: Abigail See)

Computing attention



- Encoder hidden states: $h_1^{enc}, \ldots, h_n^{enc}$ (n: # of words in source sentence)
- Decoder hidden state at time t: h_t^{dec}
- First, get attention scores for this time step of decoder:

$$e^t = [g(h_1^{enc}, h_t^{dec}), \ldots, g(h_n^{enc}, h_t^{dec})]$$

- Obtain the attention distribution using softmax:

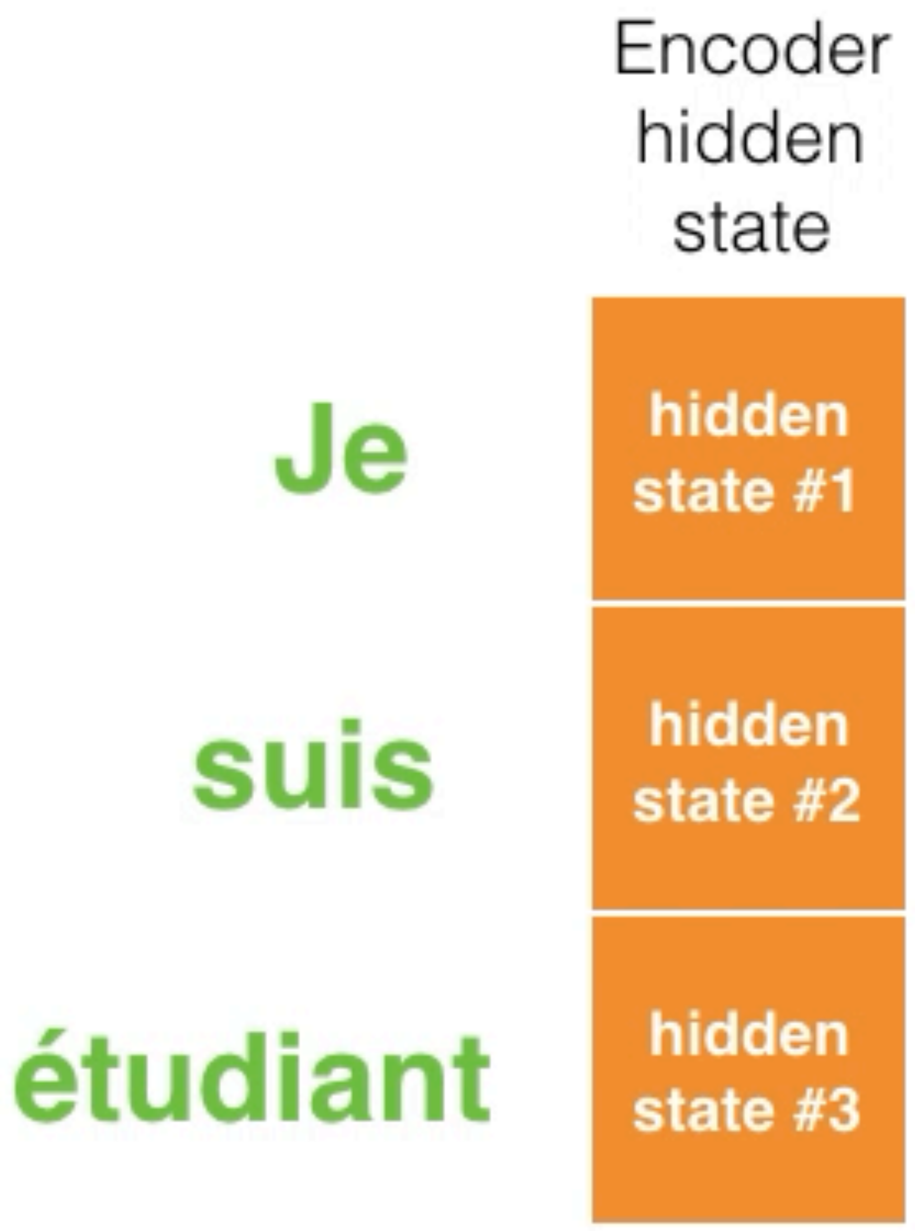
$$\alpha^t = \text{softmax}\,(e^t) \in \mathbb{R}^n$$

- Compute weighted sum of encoder hidden states:

$$a_t = \sum_{i=1}^{n} \alpha_i^t h_i^{enc} \in \mathbb{R}^h$$

- Finally, concatenate with decoder state and pass on to output layer: $\tilde{h}_t = \tanh(\mathbf{W}_c[a_t; h_t^{dec}]) \in \mathbb{R}^h \quad \mathbf{W}_c \in \mathbb{R}^{2h \times h}$

$$\hat{\mathbf{y}}_t = \text{softmax}(\mathbf{W}_o \tilde{h}_t)$$



https://jalammar.github.io/visualizing-neural-machine-translation-mechanics-of-seq2seq-models-with-attention/

(credits: Jay Alammar)

Types of attention

- Assume encoder hidden states $h_1^{enc}, h_2^{enc}, \ldots, h_n^{enc}$ and a decoder hidden state h_t^{dec}

1. **Dot-product attention** (assumes equal dimensions for h^{enc} and h_t^{dec}):

$$g(h_i^{enc}, h_t^{dec}) = (h_t^{dec})^T \; h_i^{enc} \in \mathbb{R}$$

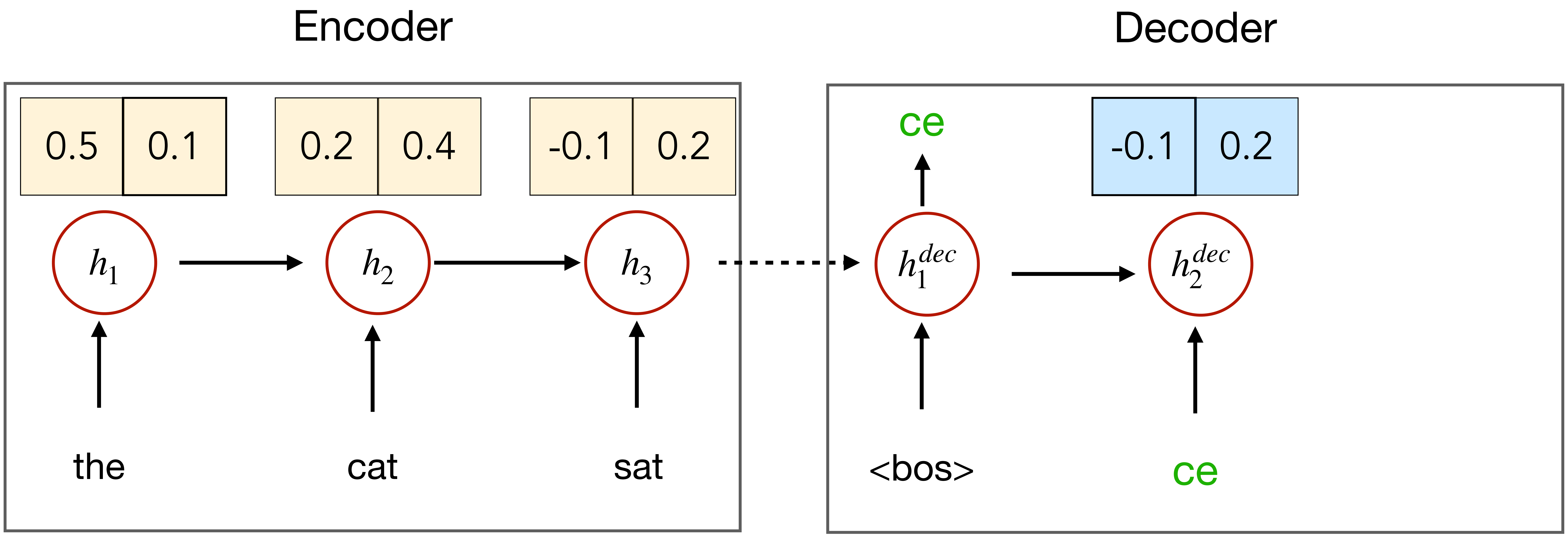
2. **Multiplicative attention:**

$g(h_i^{enc}, h_t^{dec}) = (h_t^{dec})^T \; W \; h_i^{enc} \in \mathbb{R}$, where W is a weight matrix (learned)

3. **Additive attention:**

$$g(h_i^{enc}, h_t^{dec}) = v^T \, \text{tanh} \, (W_1 h_i^{enc} + W_2 h_t^{dec}) \in \mathbb{R}$$

where W_1, W_2 are weight matrices (learned) and v is a weight vector (learned)



Assuming we use dot product attention, which input word will have the highest attention value at current time step?

Dot-product attention:

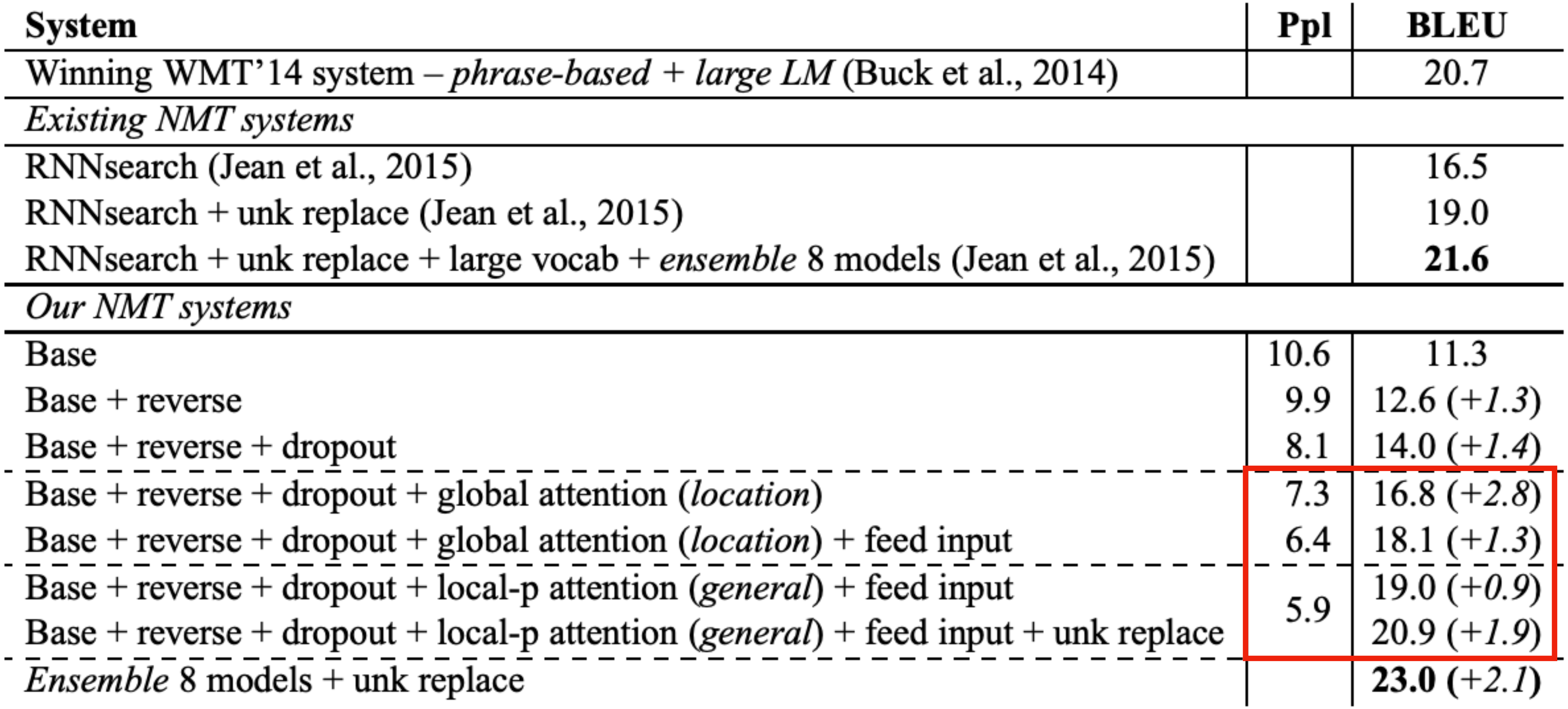
$$g(h_i^{enc}, h_t^{dec}) = h_t^{dec} \cdot h_i^{enc}$$

A) the
B) cat
C) sat

The answer is (B)

the: -0.05 + 0.02
cat: -0.02 + 0.08
sat: 0.01 + 0.04

Attention improves translation

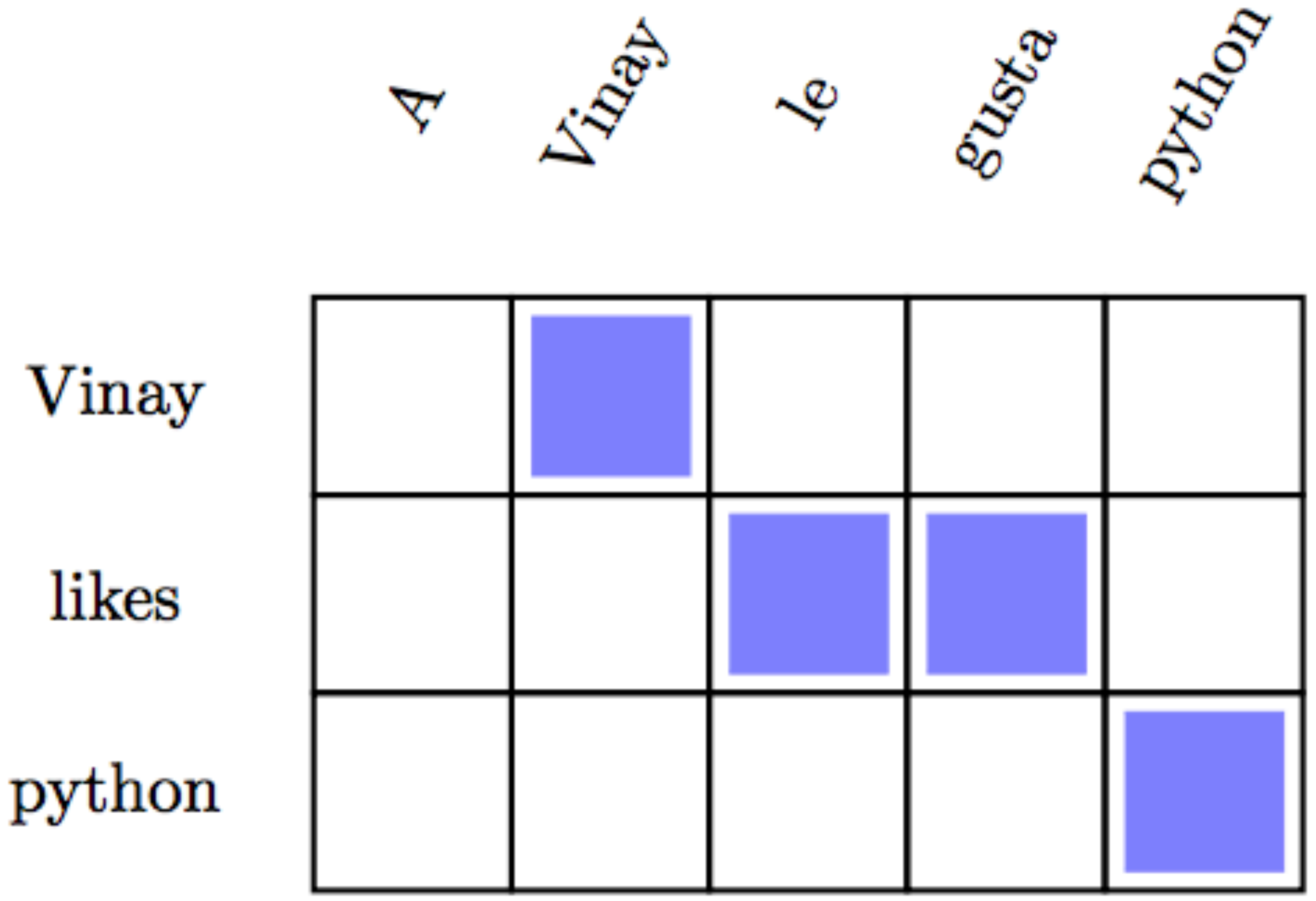


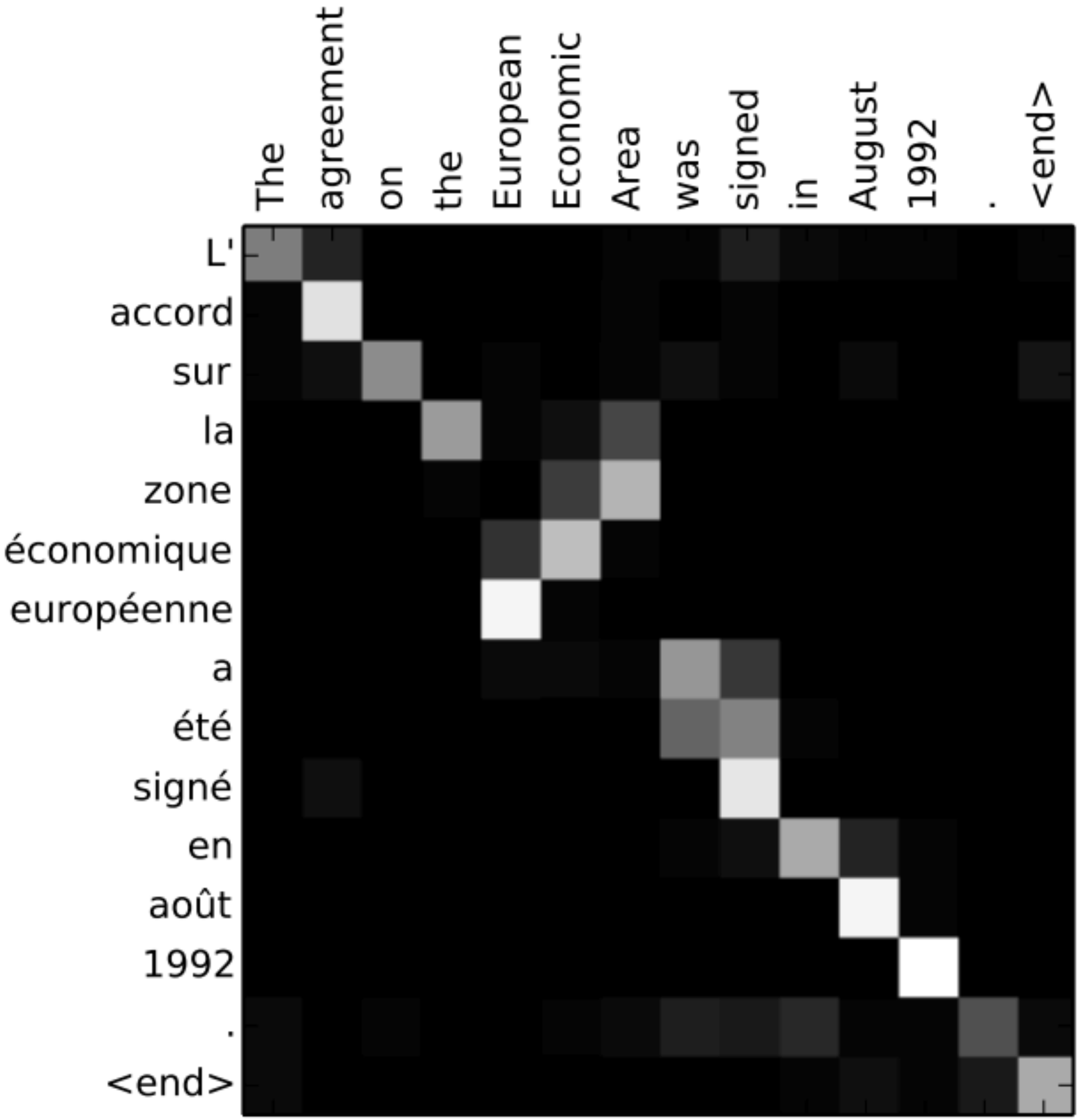
System	**Ppl**	**BLEU**
Winning WMT'14 system – *phrase-based* + *large LM* (Buck et al., 2014)		20.7
Existing NMT systems		
RNNsearch (Jean et al., 2015)		16.5
RNNsearch + unk replace (Jean et al., 2015)		19.0
RNNsearch + unk replace + large vocab + *ensemble* 8 models (Jean et al., 2015)		**21.6**
Our NMT systems		
Base	10.6	11.3
Base + reverse	9.9	12.6 (+*1.3*)
Base + reverse + dropout	8.1	14.0 (+*1.4*)
Base + reverse + dropout + global attention (*location*)	7.3	16.8 (+*2.8*)
Base + reverse + dropout + global attention (*location*) + feed input	6.4	18.1 (+*1.3*)
Base + reverse + dropout + local-p attention (*general*) + feed input	5.9	19.0 (+*0.9*)
Base + reverse + dropout + local-p attention (*general*) + feed input + unk replace		20.9 (+*1.9*)
Ensemble 8 models + unk replace		**23.0** (+*2.1*)

(Luong et al., 2015)

Visualizing attention

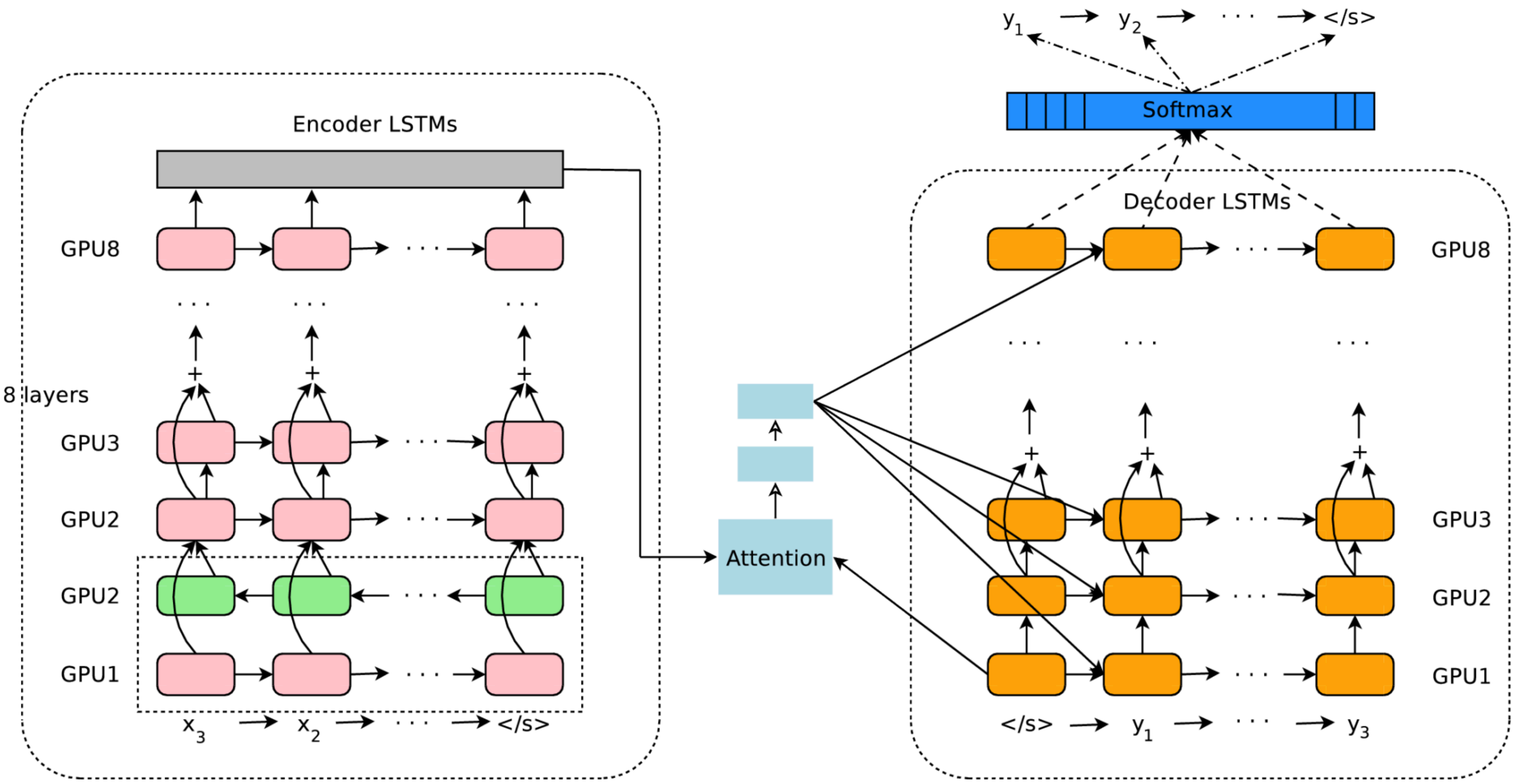
Recall: alignment





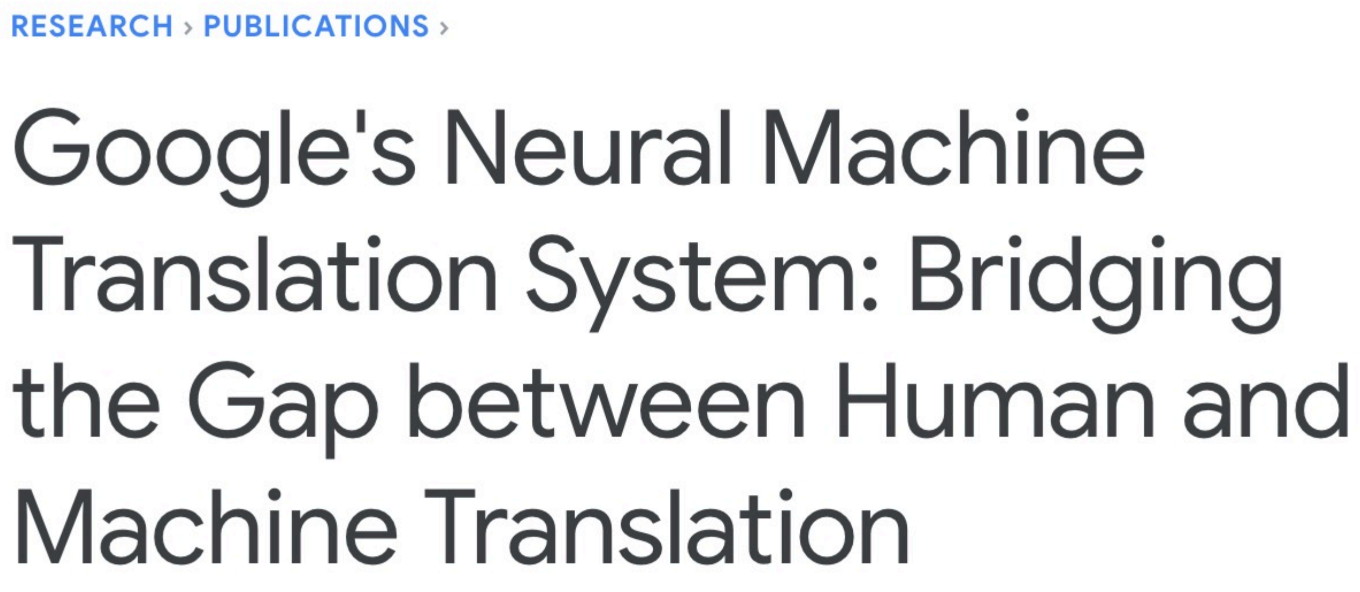
(credits: Jay Alammar)

Putting everything together: Google's NMT System



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

Putting everything together: Google's NMT System



RESEARCH › PUBLICATIONS ›

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

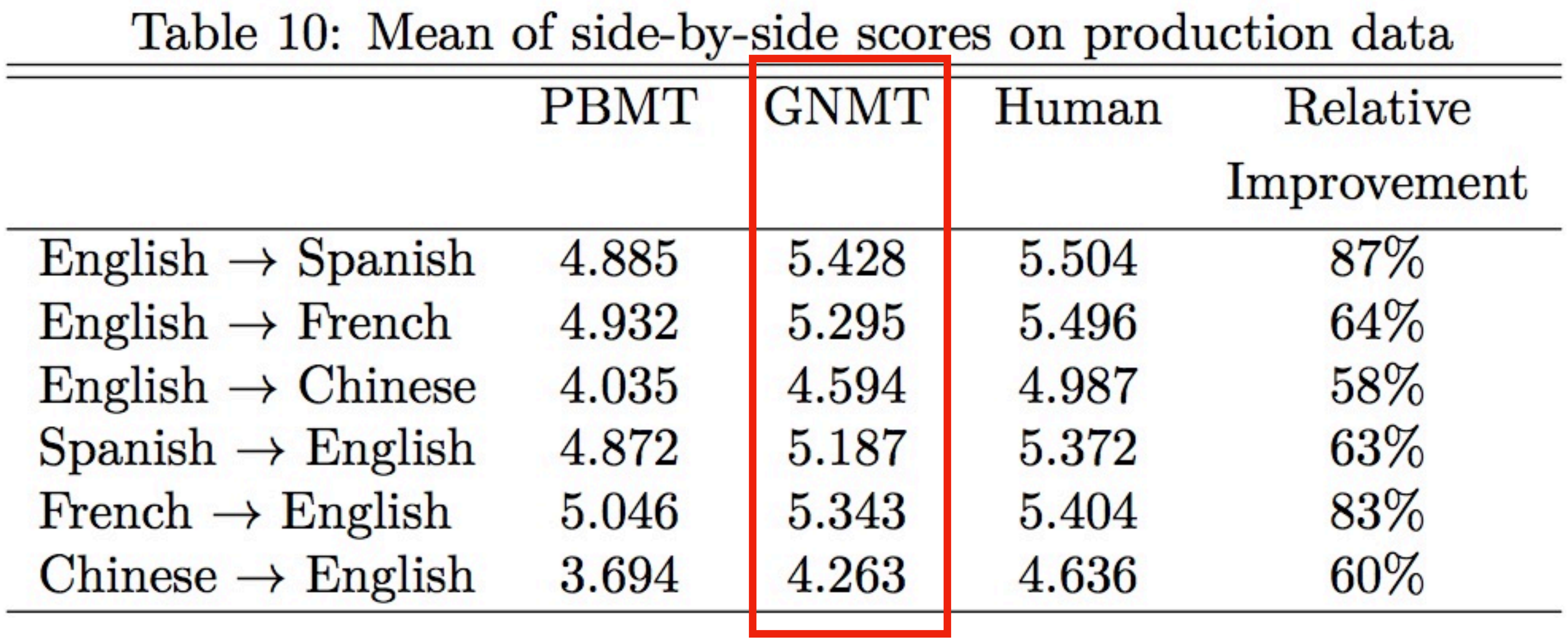


Table 10: Mean of side-by-side scores on production data

	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

Questions?

Acknowledgement

Princeton COS 484 by Danqi Chen, Tri Dao, Vikram Ramaswamy
CMU CS11-711 Advanced NLP by Graham Neubig & Sean Welleck