University of Ljubljana, Faculty of Computer and Information Science

Machine translation

Prof Dr Marko Robnik-Šikonja

Natural Language Processing, Edition 2022

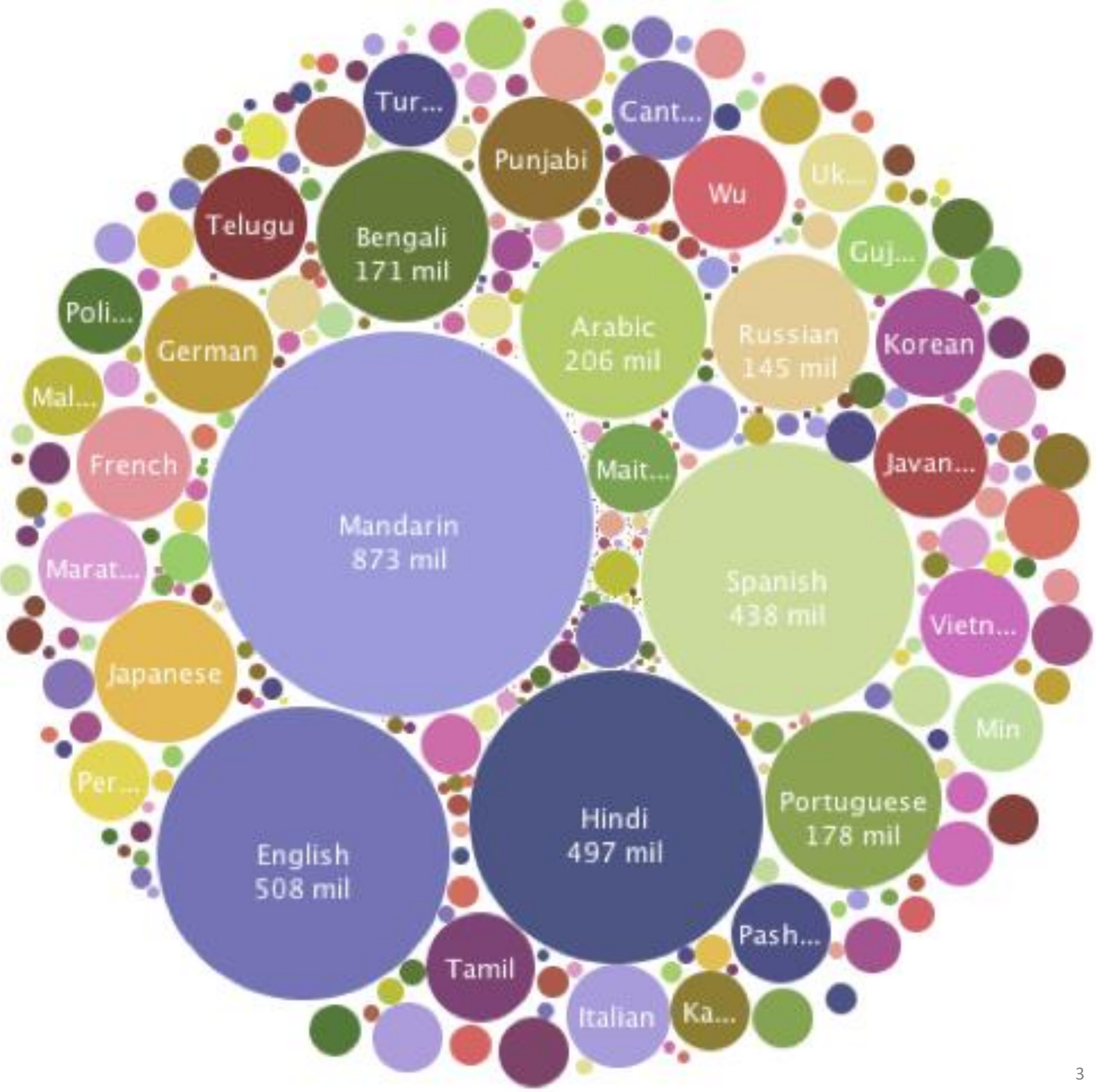
Contents

- statistical machine translation
- neural machine translation using sequence to sequence approach

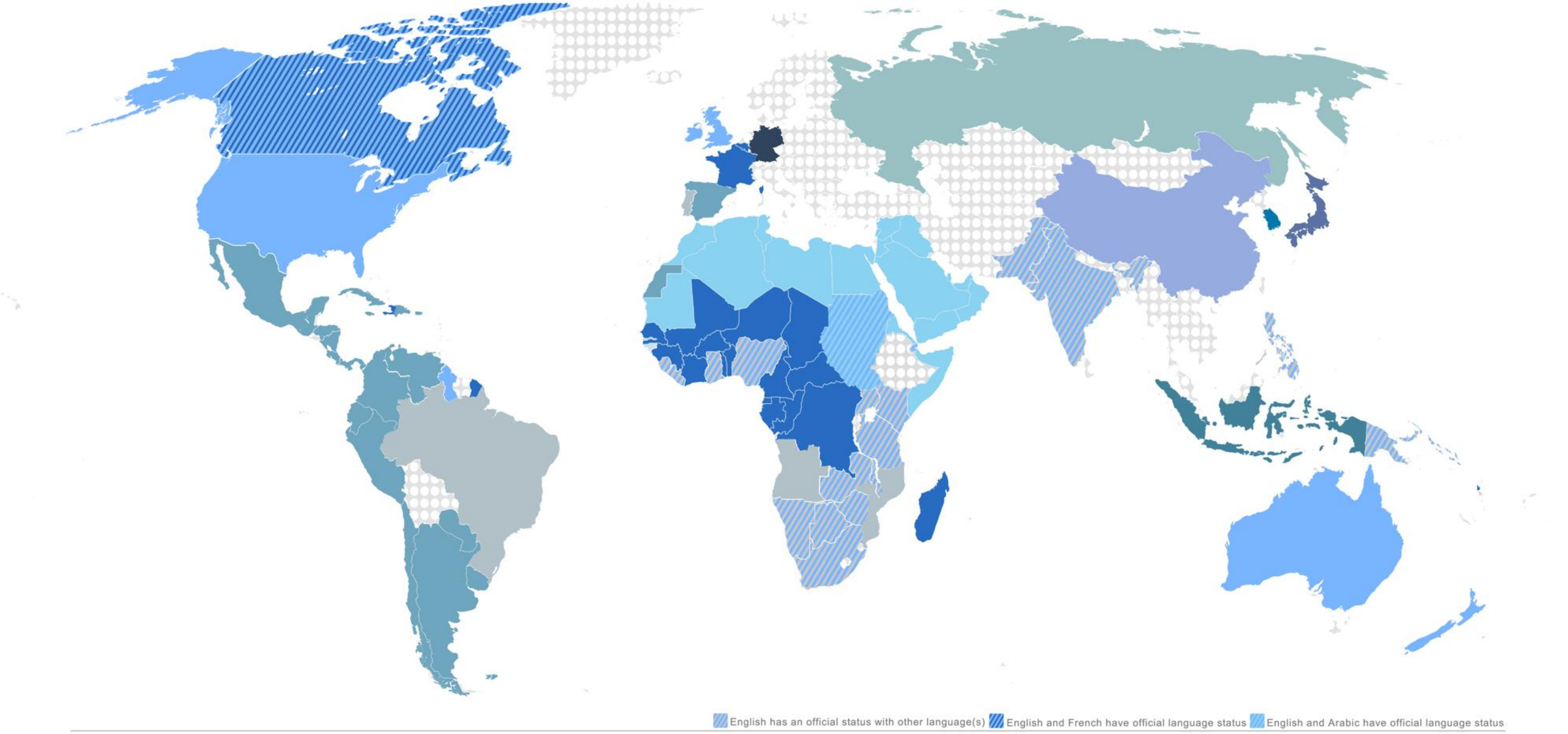
- Literature:
  Graham Neubig (2017). Neural Machine Translation and Sequence-to-sequence Models: A Tutorial. arXiv:1703.01619v1
- Stanford course CS224n: Natural Language Processing with Deep Learning https://web.stanford.edu/class/cs224n/

Word languages

Currently 6909 languages, 6% with more than one million speakers, together they cover 94% of world population.

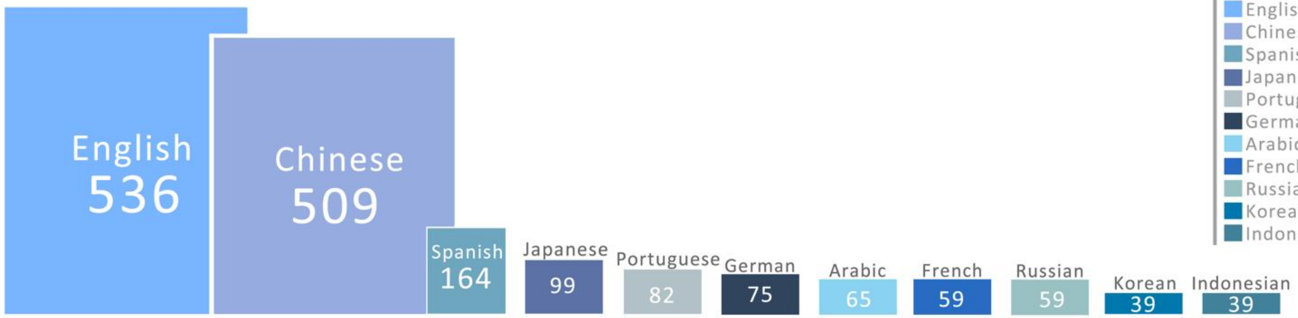


Top Languages on the Internet


▨ English has an official status with other language(s)
▨ English and French have official language status
▨ English and Arabic have official language status

Number of Internet users by Language - mln people

The bars' heights correspond with the figure

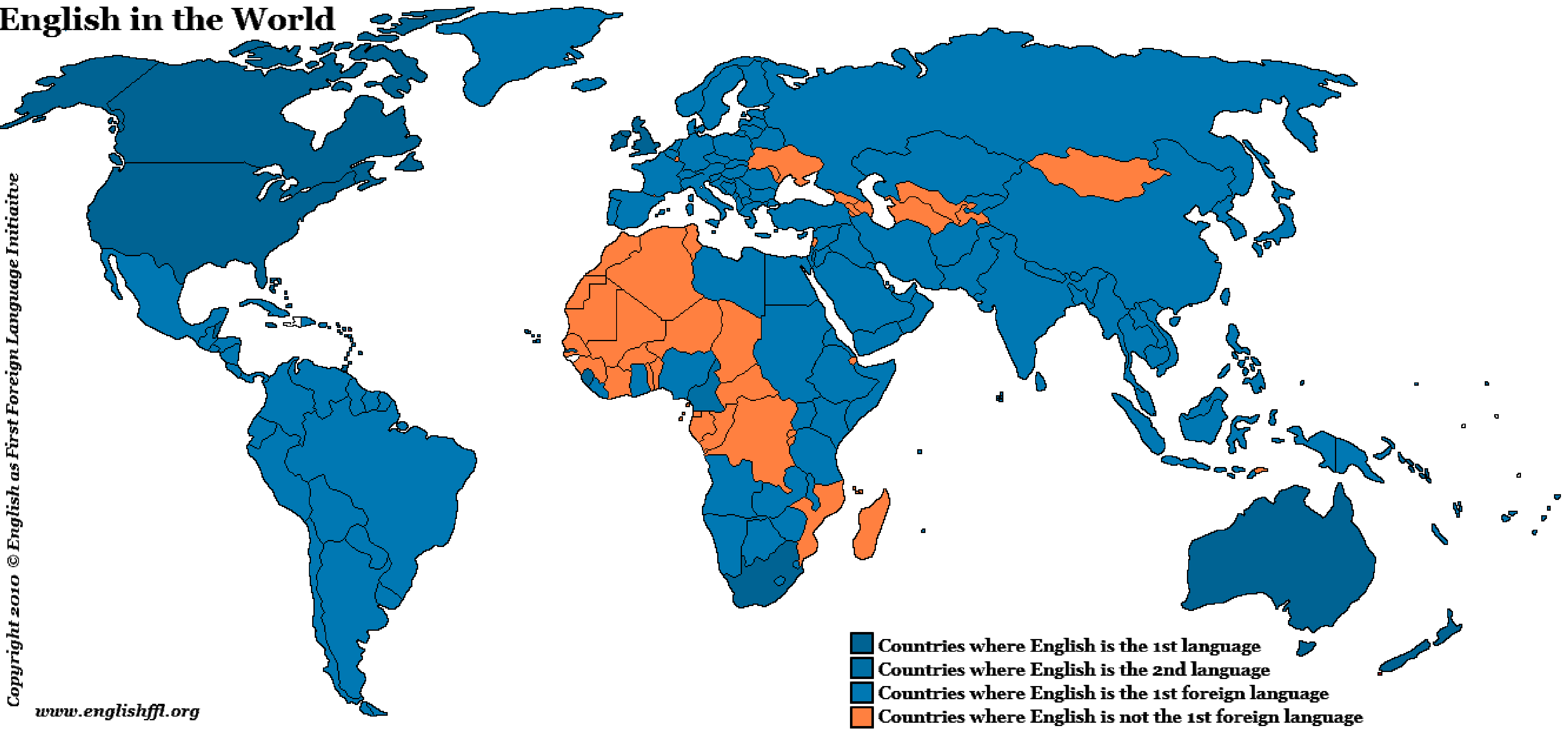


Internet Penetration by Language
English - 43%
Chinese - 37%
Spanish - 39%
Japanese - 78%
Portuguese - 32%
German - 79%
Arabic - 18%
French - 17%
Russian - 42%
Korean - 55%
Indonesian - 16%

World population by Language (mln)
English - 1302
Chinese - 1372
Spanish - 423
Japanese - 126
Portuguese - 253
German - 94
Arabic - 347
French - 347
Russian - 139
Korean - 71
Indonesian - 245

Source: Internet World Stats

English as lingua franca?


English in the World
Copyright 2010 © English as First Foreign Language Initiaitve
www.englishffl.org
Countries where English is the 1st language
Countries where English is the 2nd language
Countries where English is the 1st foreign language
Countries where English is not the 1st foreign language

Statistical machine translation (SMT)

- The intuition for Statistical MT comes from the **impossibility** of perfect translation
- Why perfect translation is impossible
  - Goal: Translating Hebrew `adonai roi` ("the lord is my shepherd") for a culture without sheep or shepherds
- Two options:
  - Something **fluent** and understandable, but not faithful:

    ```
    The Lord will look after me
    ```

  - Something **faithful**, but not fluent or natural

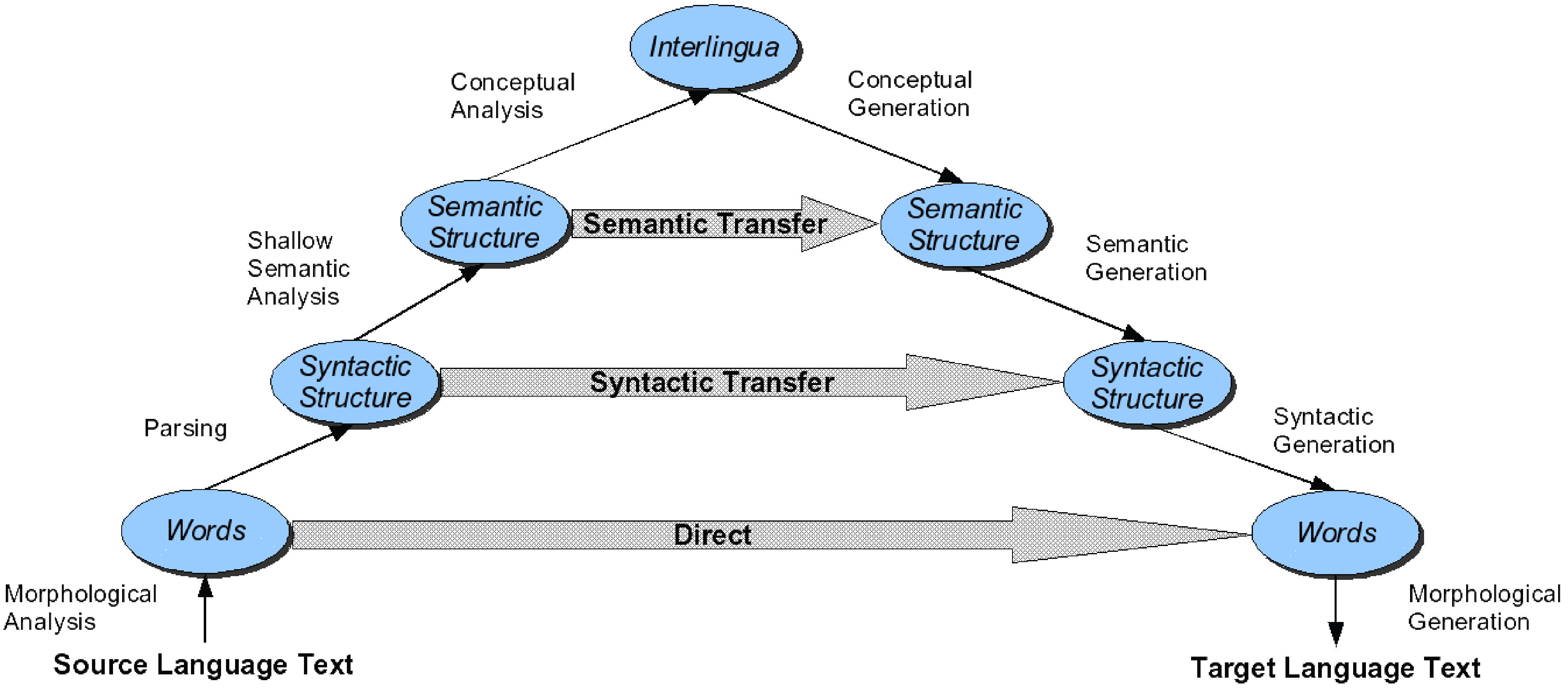
    ```
    The Lord is for me like somebody who
    looks after animals with cotton-like hair
    ```

A good translation is:

- Faithful
  - Has the same meaning as the source
  - (Causes the reader to draw the same inferences as the source would have)
- Fluent
  - Is natural, fluent, grammatical in the target

- Real translations trade off these two factors

Three MT Approaches: Direct, Transfer, Interlingual



Machine translation as decoding

- Norbert Wiener (1947, in a letter): … When I look at an article in Russian, I say, “This is really written in English, but it has been coded in some strange symbols. I will now proceed to decode.” …

Classical statistical machine translation

- word-based models
- phrase-based models
- tree based models
- factored models

Statistical MT:
Faithfulness and Fluency formalized

Peter Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, Robert L. Mercer. 1993. The Mathematics of Statistical Machine Translation: Parameter Estimation. Computational Linguistics 19:2, 263-311. **"The IBM Models"**

Given a French (foreign) sentence F, find an English sentence

$$\hat{E} = \underset{E\hat{I}\ English}{\operatorname{argmax}} P(E \mid F)$$

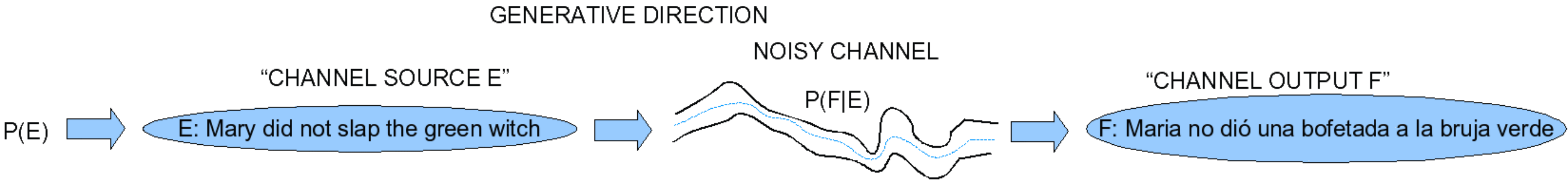
$$= \underset{E\hat{I}\ English}{\operatorname{argmax}} \frac{P(F \mid E)P(E)}{P(F)}$$

$$= \underset{E\hat{I}\ English}{\operatorname{argmax}} \underbrace{P(F \mid E)}_{\text{Translation Model}} \underbrace{P(E)}_{\text{Language Model}}$$

Convention in Statistical MT

- We always refer to translating
  - from input F, the foreign language (originally F = French)
  - to output E, English.
- Obviously statistical MT can translate from English into another language or between any pair of languages
- The convention helps avoid confusion about which way the probabilities are conditioned for a given example

The noisy channel model for MT


GENERATIVE DIRECTION
NOISY CHANNEL
"CHANNEL SOURCE E"
P(F|E)
"CHANNEL OUTPUT F"
P(E)
E: Mary did not slap the green witch
F: Maria no dió una bofetada a la bruja verde

Fluency: P(E)

- We need a metric that ranks this sentence

  `That car almost crash to me`

as less fluent than this one:

  `That car almost hit me.`

- Answer: language models (e.g., N-grams)

  P(me|hit) > P(to|crash)

  – And we can use any other more sophisticated model of grammar
- Advantage: this is monolingual knowledge!

Faithfulness: P(F|E)

- Spanish:
  - Maria no dió una bofetada a la bruja verde
- English candidate translations:
  - Mary didn’t slap the green witch
  - Mary not give a slap to the witch green
  - The green witch didn’t slap Mary
  - Mary slapped the green witch

- More faithful translations will be composed of phrases that are high probability translations
  - How often was “slapped” translated as “dió una bofetada” in a large **bitext** (parallel English-Spanish corpus)
  - in classical MT, we’ll need to align phrases and words to each other in bitext

We treat Faithfulness and Fluency as independent factors

- P(F|E)'s job is to model "bag of words"; which words come from English to Spanish.
  - P(F|E) doesn't have to worry about internal facts about English word order.
- P(E)'s job is to do bag generation: put the following words in order:
  - a ground there in the hobbit hole lived a in

Three Problems for Statistical MT

- Language Model: given E, compute P(E)

  good English string → high P(E)

  random word sequence → low P(E)

- Translation Model: given (F,E) compute P(F | E)

  (F,E) look like translations → high P(F | E)

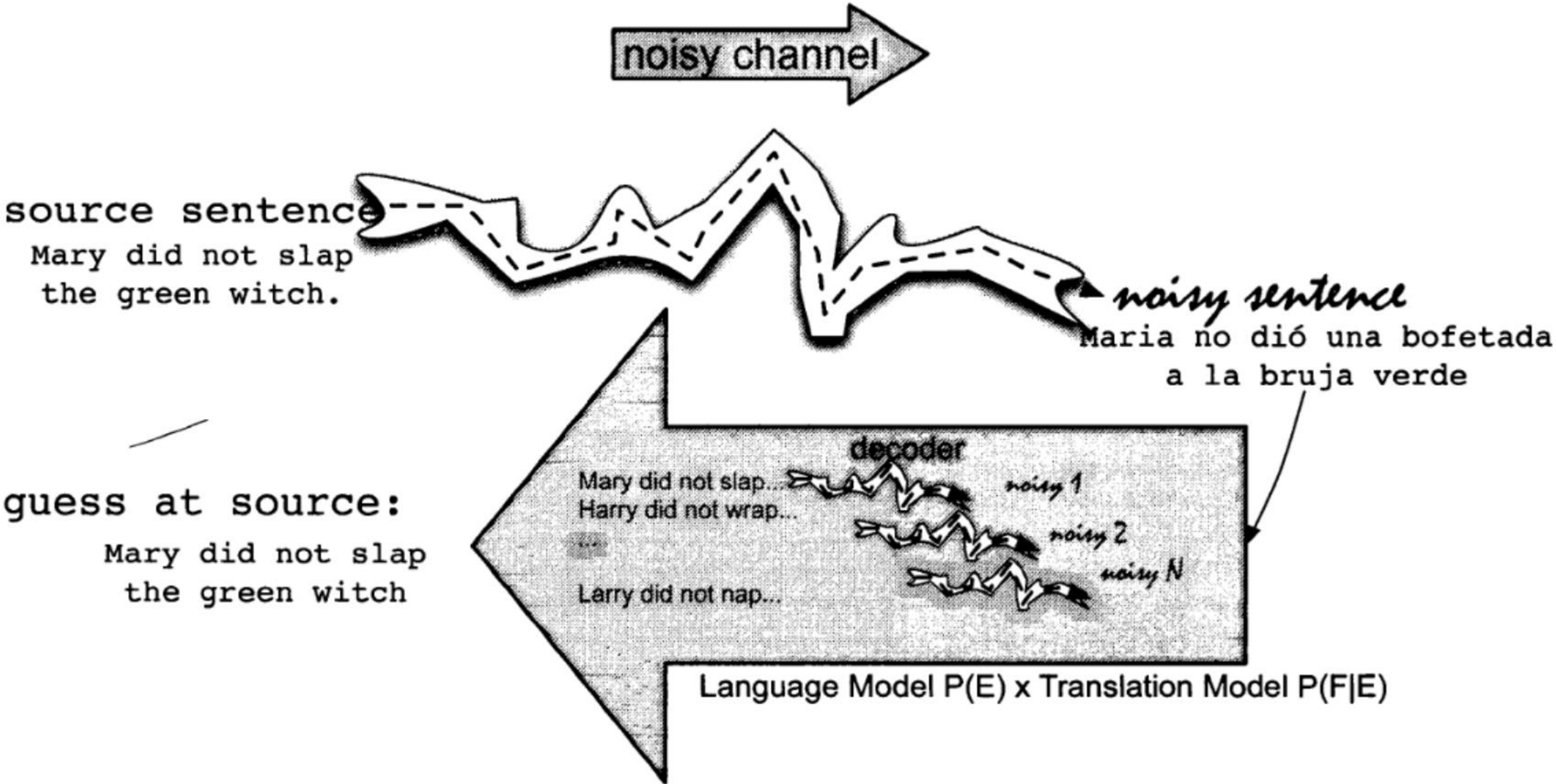
  (F,E) don't look like translations → low P(F | E)

- Decoding algorithm: given LM, TM, F, find Ê

  Find translation E that maximizes P(E) * P(F | E)

Noisy channel model

- inference goes backwards



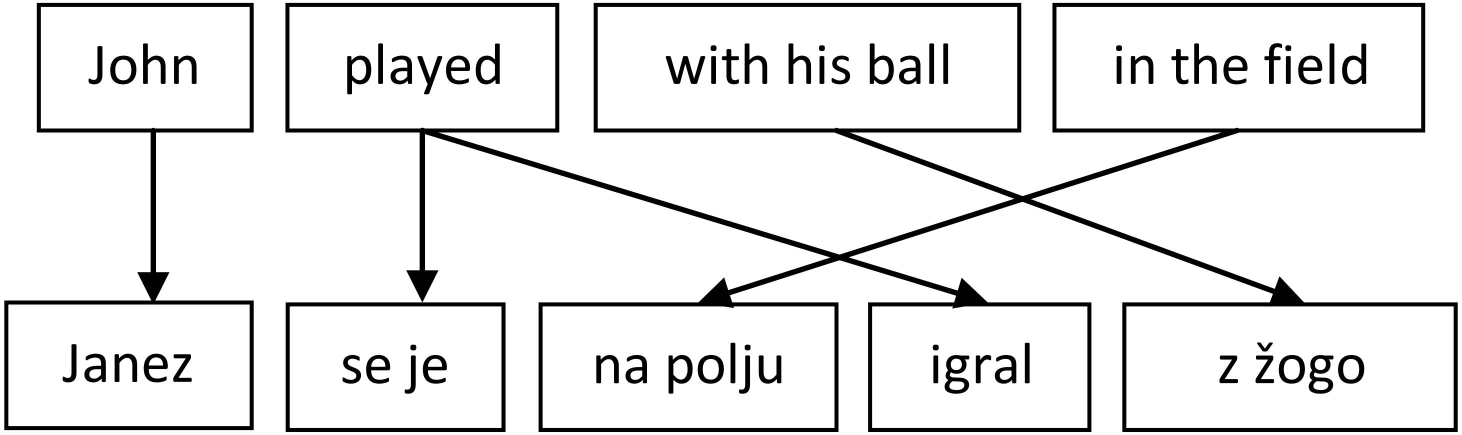
Language Model

- In SMT: use a standard *n*-gram language model for P(*E*).
- Can be trained on a large mono-lingual corpus
    - 5-gram grammar of English from terabytes of web data
    - More sophisticated parser-based language models can also help

– Neural LMs

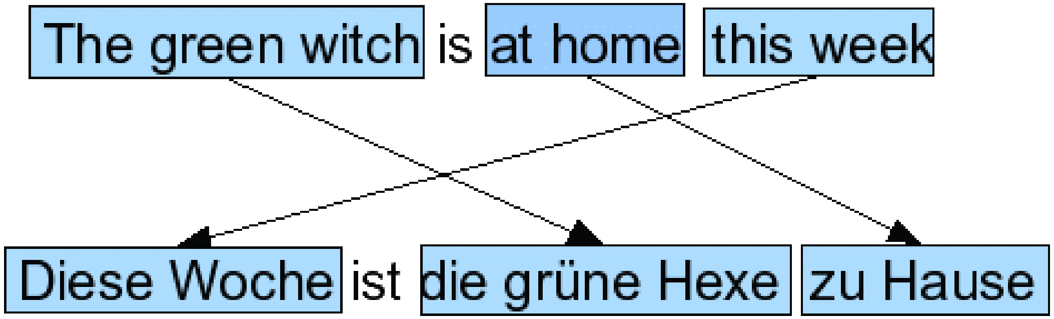
Phrase-based statistical MT

- the translation unit is not a word but a phrase



Phrase-Based Translation

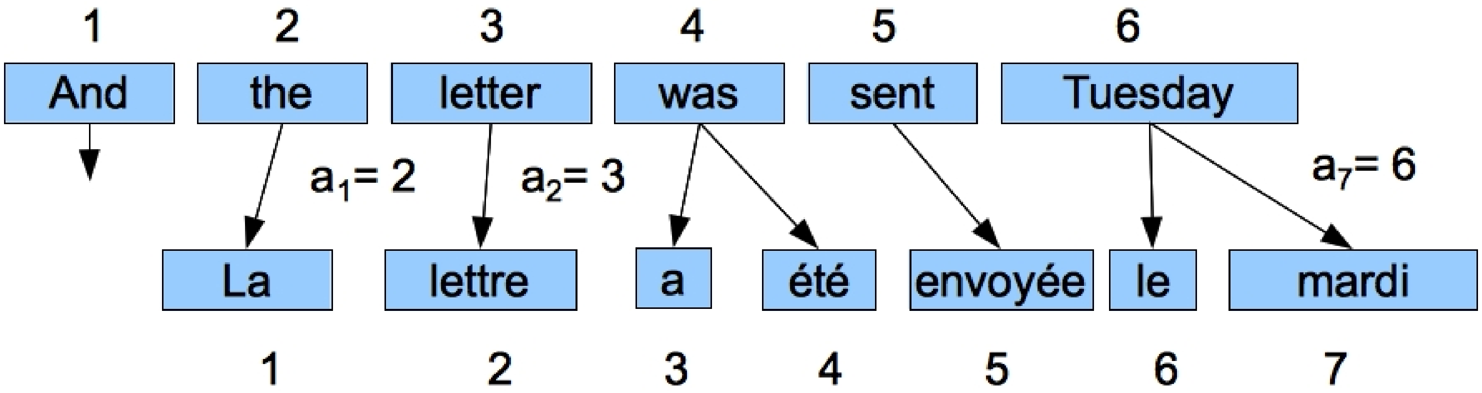
(Koehn et al. 2003)



- Remember the noisy channel model is backwards:
  - We translate German to English by pretending an English sentence generated a German sentence
    - Generative model gives us our probability P(F|E)
  - Given a German sentence, find the English sentence that generated it.

Word Alignment

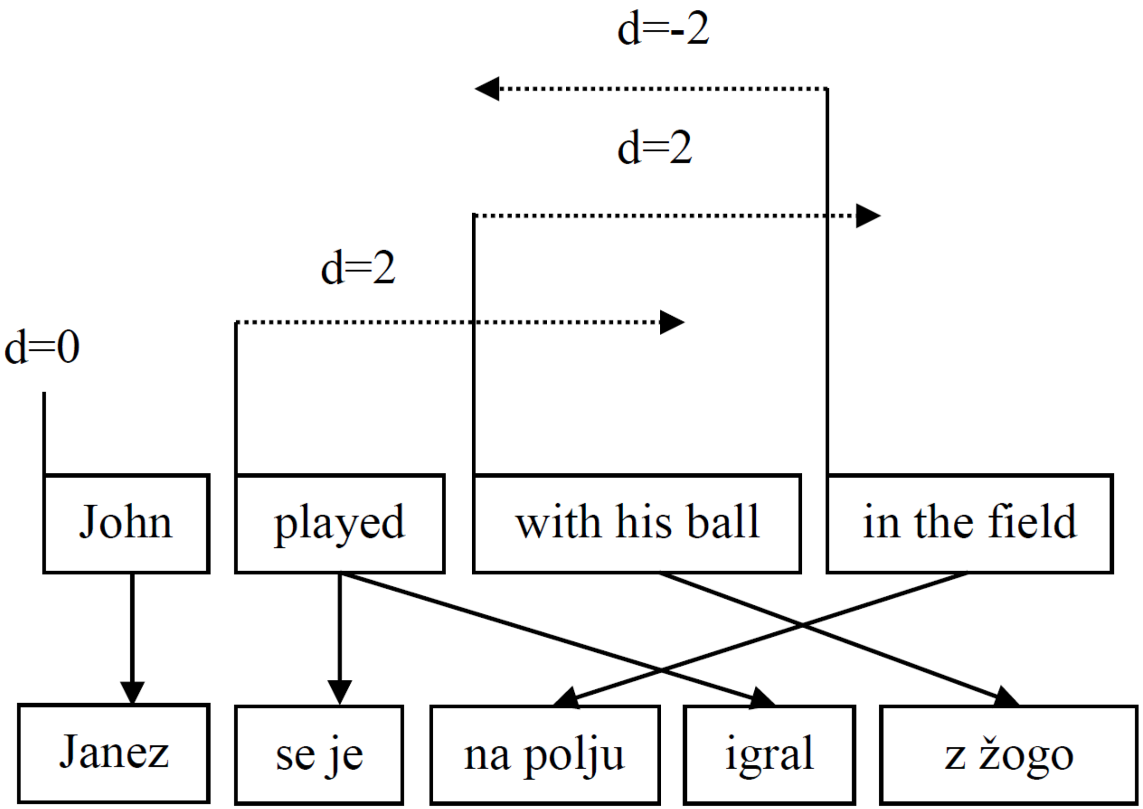
- A mapping between words in F and words in E



- Simplifying assumptions (for IBM Model 1 and HMM alignments):
  - one-to-many (not many-to-one or many-to-many)
    - each French word comes from exactly one English word
  - An alignment is a vector of length J, one cell for each French word
    - The index of the English word that the French word comes from
- Alignment above is thus the vector $A = [2, 3, 4, 4, 5, 6, 6]$ $a_1=2$, $a_2=3$, $a_3=4$, $a_4=4\ldots$

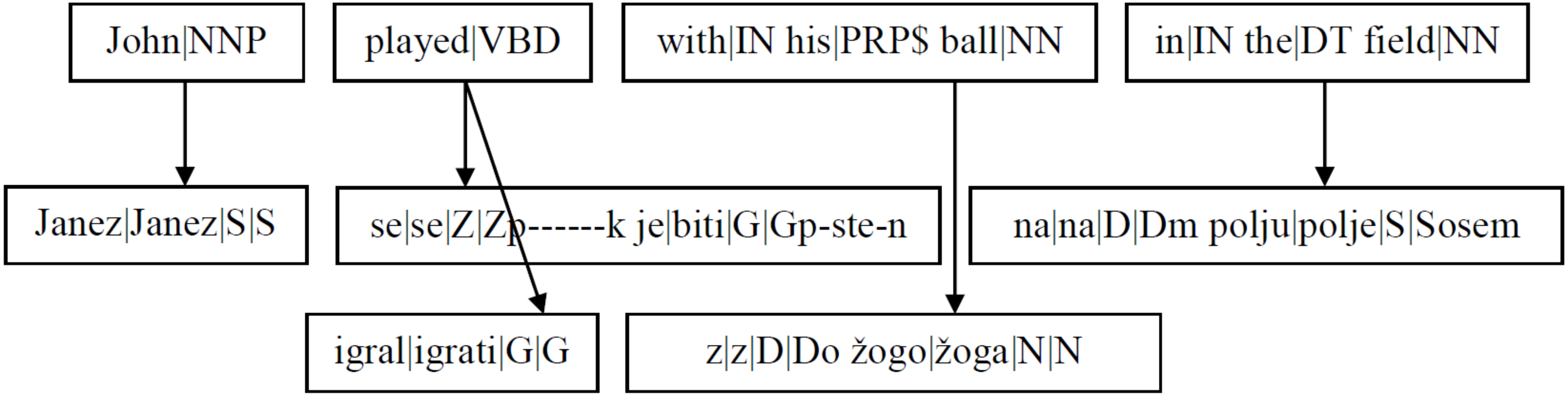
Phrase alignment

- alignment is based on distances
- longer distances are costlier



Factor based MT models

- we add features (tags) to words, called factors
- we use factors in alignments



John|NNP

played|VBD

with|IN his|PRP$ ball|NN

in|IN the|DT field|NN

Janez|Janez|S|S

se|se|Z|Zp------k je|biti|G|Gp-ste-n

na|na|D|Dm polju|polje|S|Sosem

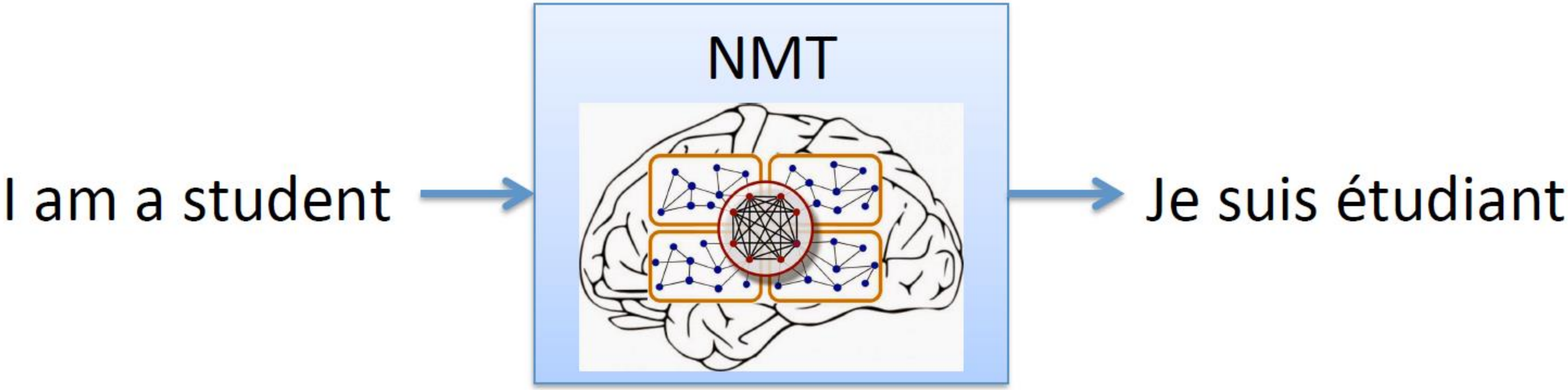
igral|igrati|G|G

z|z|D|Do žogo|žoga|N|N

Parallel corpora

- EuroParl: http://www.statmt.org/europarl/
  - A parallel corpus extracted from proceedings of the European Parliament.
  - Philipp Koehn. 2005. Europarl: A Parallel Corpus for Statistical Machine Translation. MT Summit
  - around 50 million words per EU language
    - Danish, Dutch, English, Finnish, French, German, Greek, Italian, Portuguese, Spanish, Swedish, Bulgarian, Czech, Estonian, Hungarian, Latvian, Lithuanian, Polish, Romanian, Slovak, and Slovene
- LDC: http://www.ldc.upenn.edu/
  - Large amounts of parallel English-Chinese and English-Arabic text
- Subtitles
- OPUS website

Neural machine translation (NMT)



I am a student

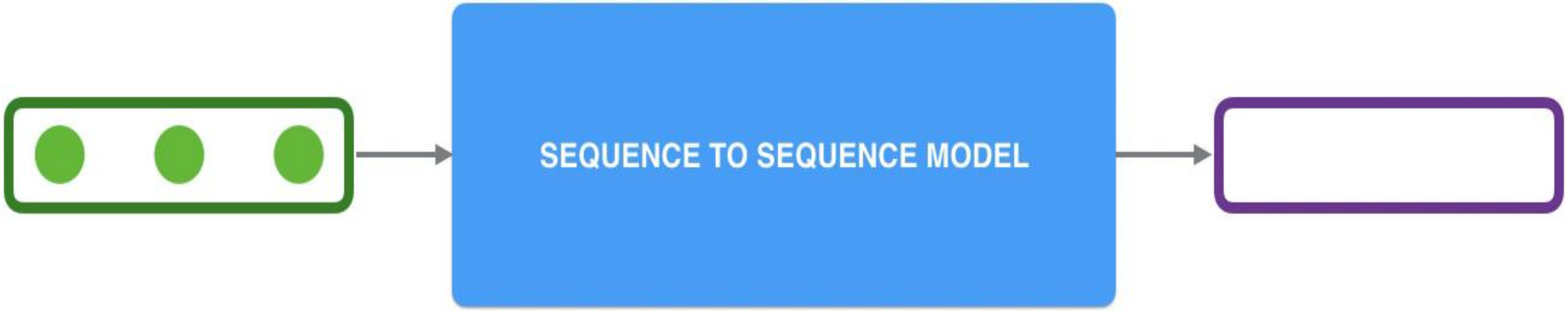
NMT

Je suis étudiant

(Sutskever et al., 2014; Cho et al., 2014)

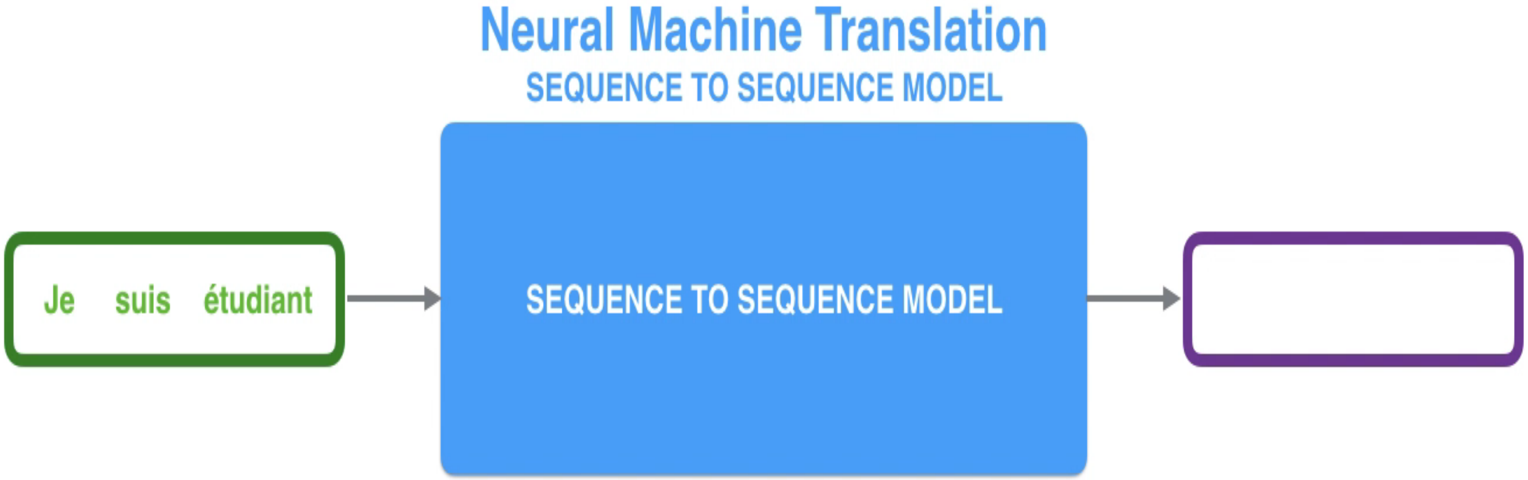
- direct translation based on sequences
- The neural network architecture is called sequence-to-sequence (aka seq2seq) and it involves *two* networks.

Seq2Seq model

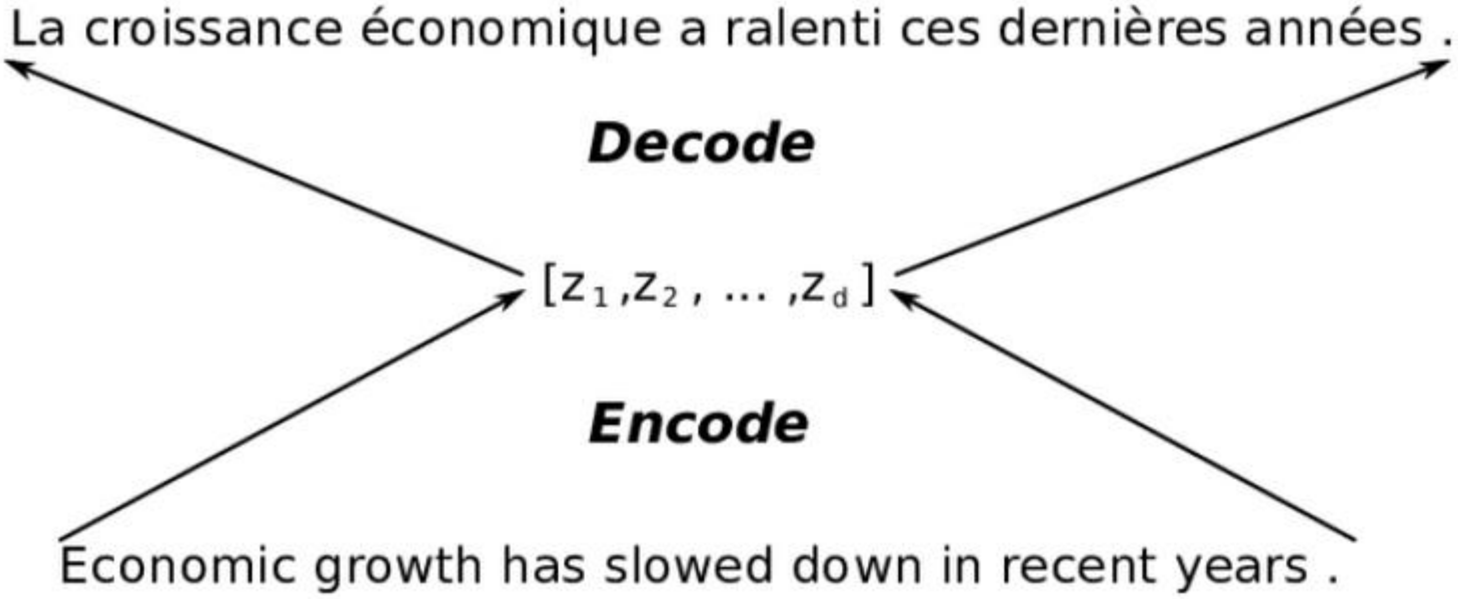


Videos by Jay Alammar: Visualizing A Neural Machine Translation Model (Mechanics of Seq2seq Models With Attention), 2018

Seq2Seq for NMT


Neural Machine Translation
SEQUENCE TO SEQUENCE MODEL
Je suis étudiant
SEQUENCE TO SEQUENCE MODEL

Encoder-Decoder Model



La croissance économique a ralenti ces dernières années .

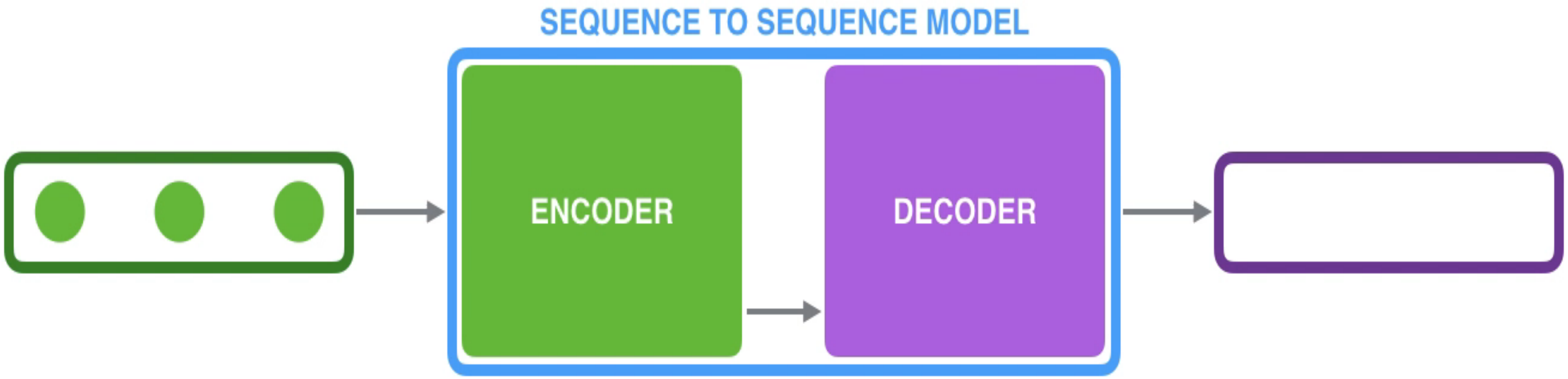
Decode

$[z_1, z_2, \ldots, z_d]$

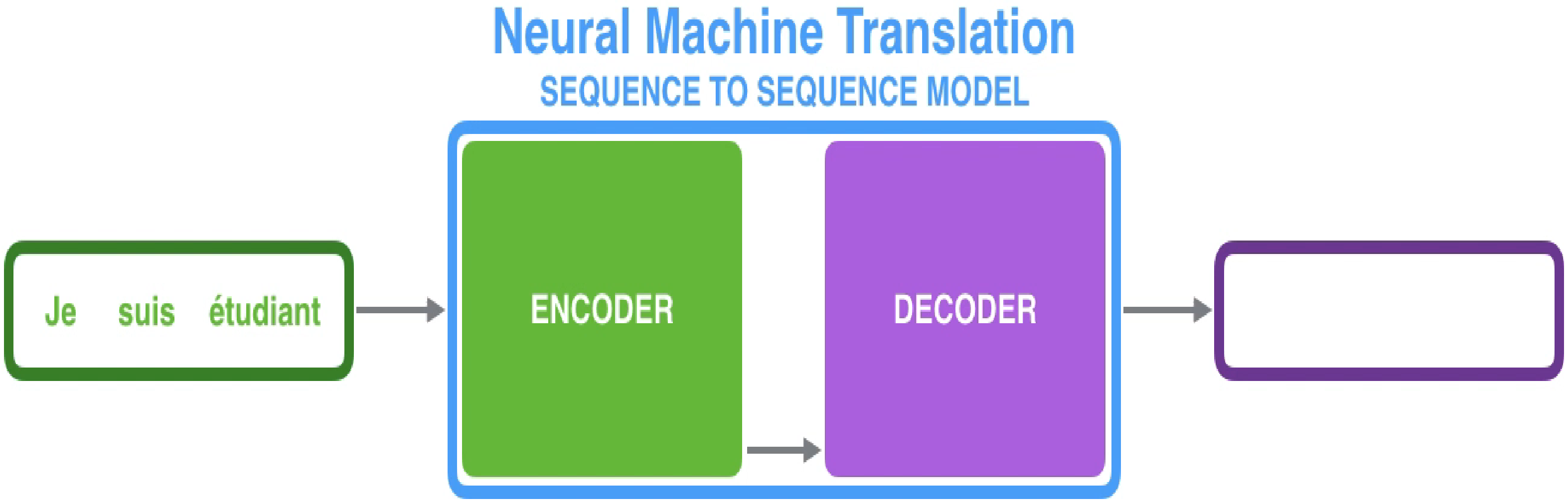
Encode

Economic growth has slowed down in recent years .

Encoder-decoder for sequences

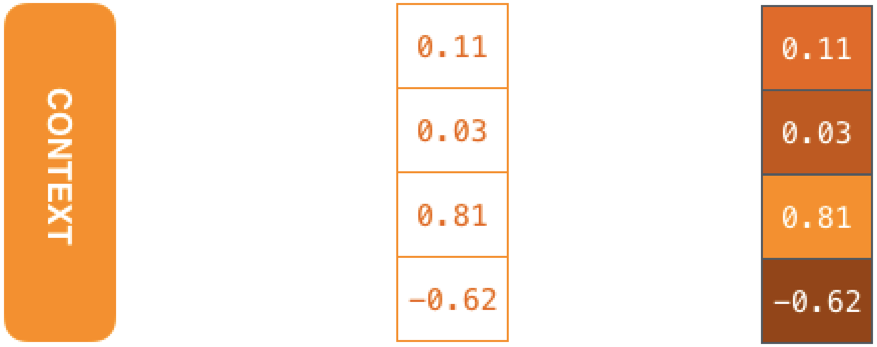


Encoder-decoder for NMT



Neural Machine Translation

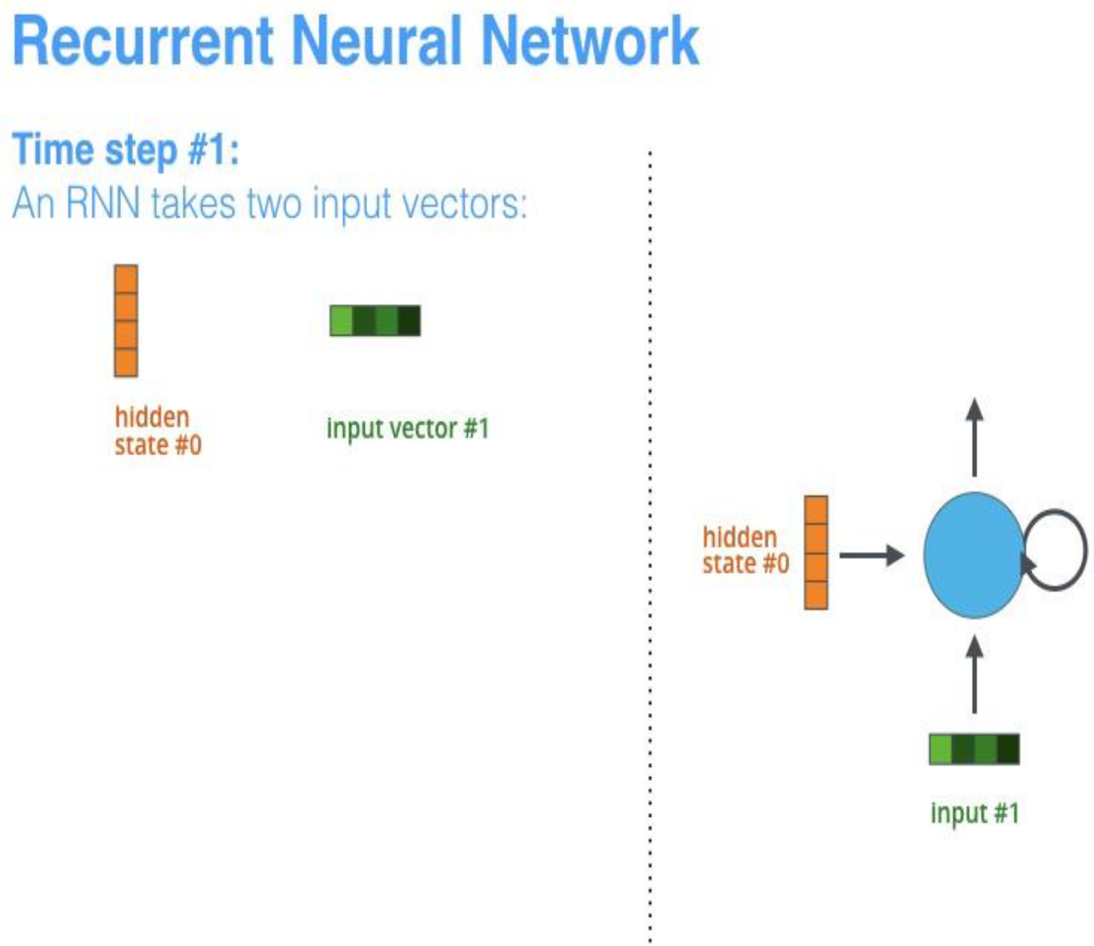
SEQUENCE TO SEQUENCE MODEL



0.11
0.03
0.81
-0.62

0.11
0.03
0.81
-0.62

RNN processing

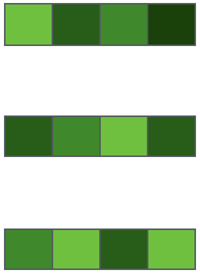


Recurrent Neural Network

Time step #1:
An RNN takes two input vectors:

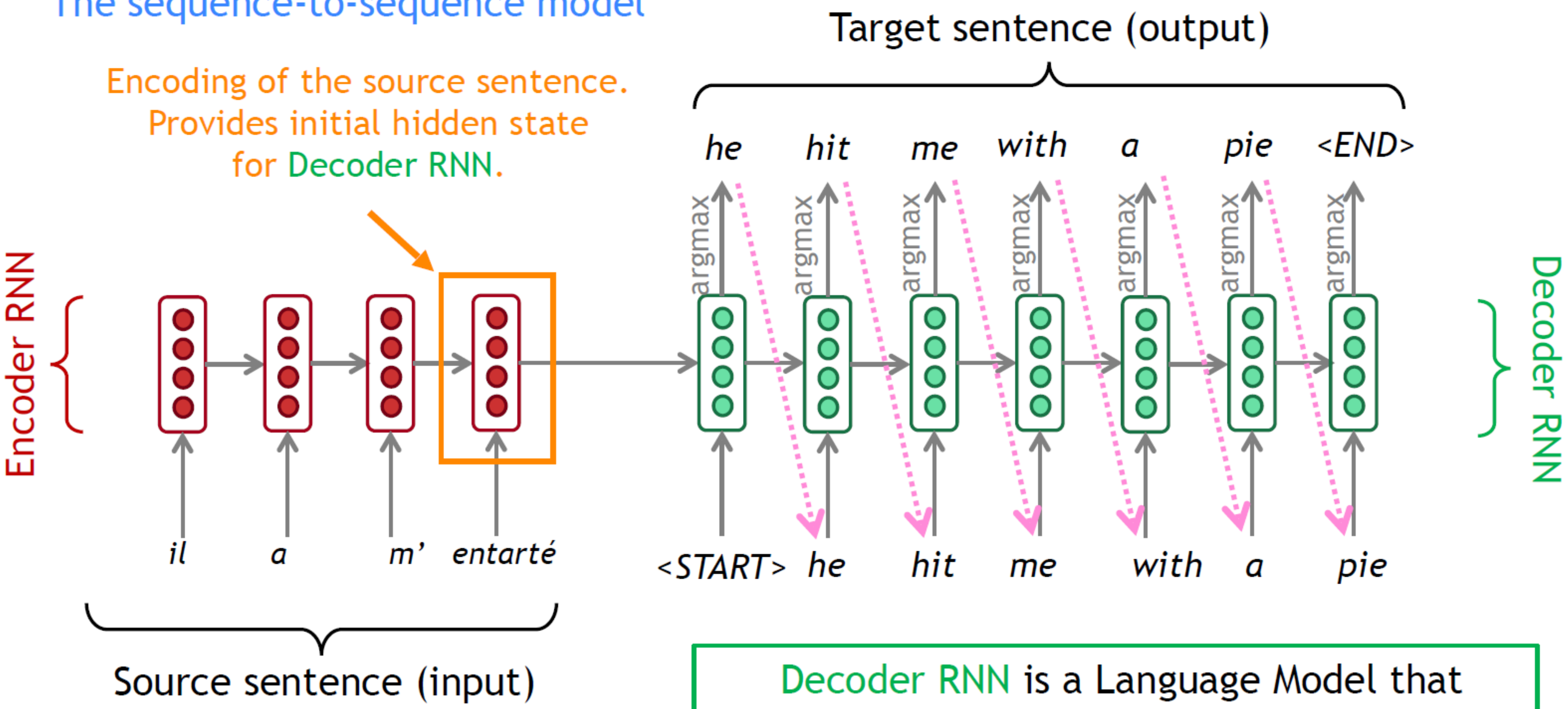
Representation

Input				
Je	0.901	-0.651	-0.194	-0.822
suis	-0.351	0.123	0.435	-0.200
étudiant	0.081	0.458	-0.400	0.480



NMT

The sequence-to-sequence model



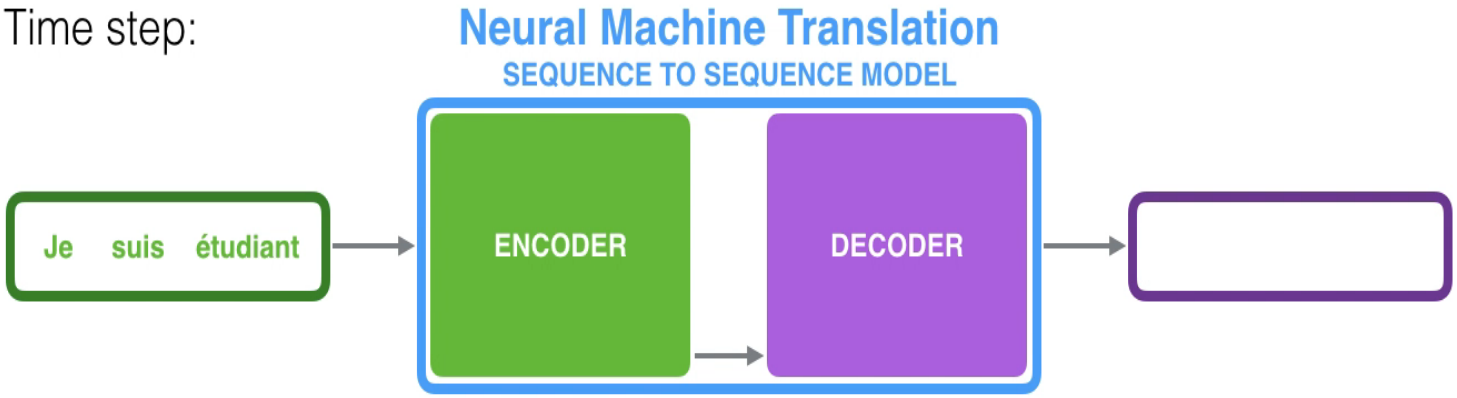
Encoding of the source sentence. Provides initial hidden state for Decoder RNN.

Encoder RNN produces an encoding of the source sentence.

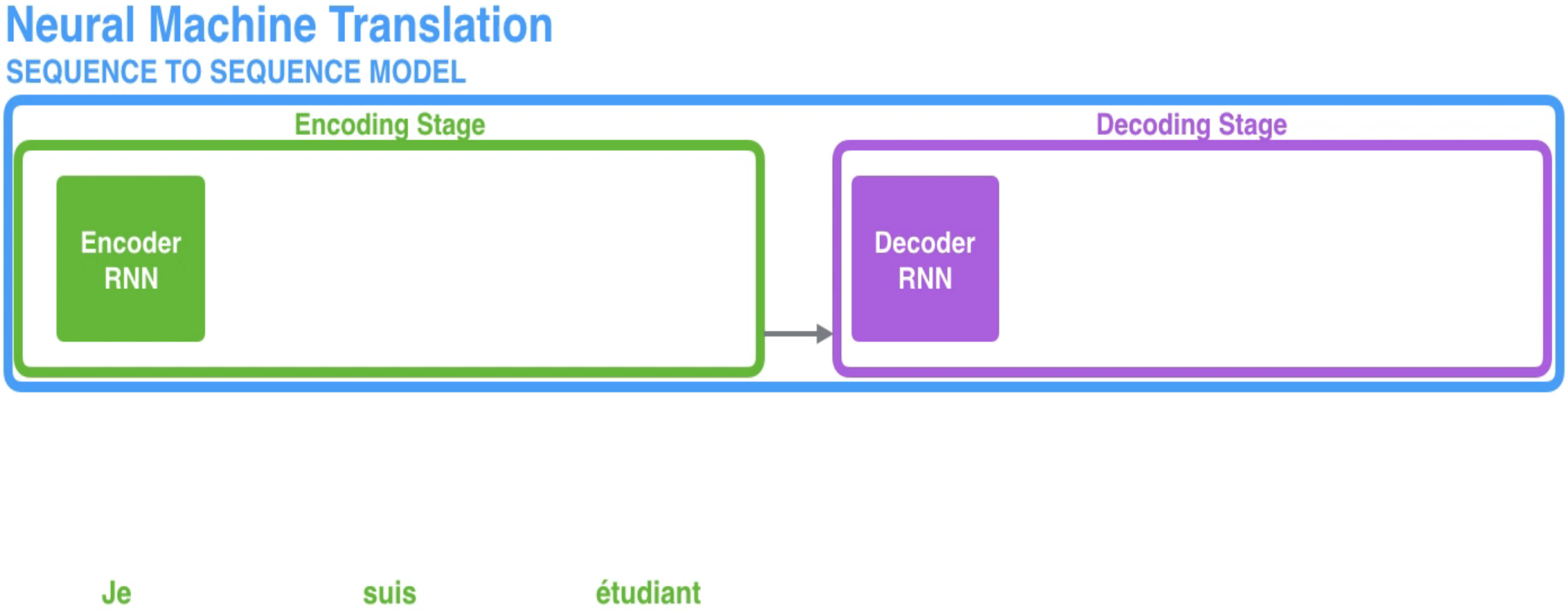
Decoder RNN is a Language Model that generates target sentence, *conditioned on encoding*.

Note: This diagram shows **test time** behavior: decoder output is fed in ·····➤ as next step's input

Encoder-decoder hidden states



Unrolled encoder-decoder



Sequence to sequence model

- Sequence-to-sequence is versatile!
- Sequence-to-sequence is useful for *more than just MT*
- Many NLP tasks can be phrased as sequence-to-sequence:
  - Summarization (long text → short text)
  - Dialogue (previous utterances → next utterance)
  - Parsing (input text → output parse as sequence)
  - Code generation (natural language → Python code)

Seq2seq NMT

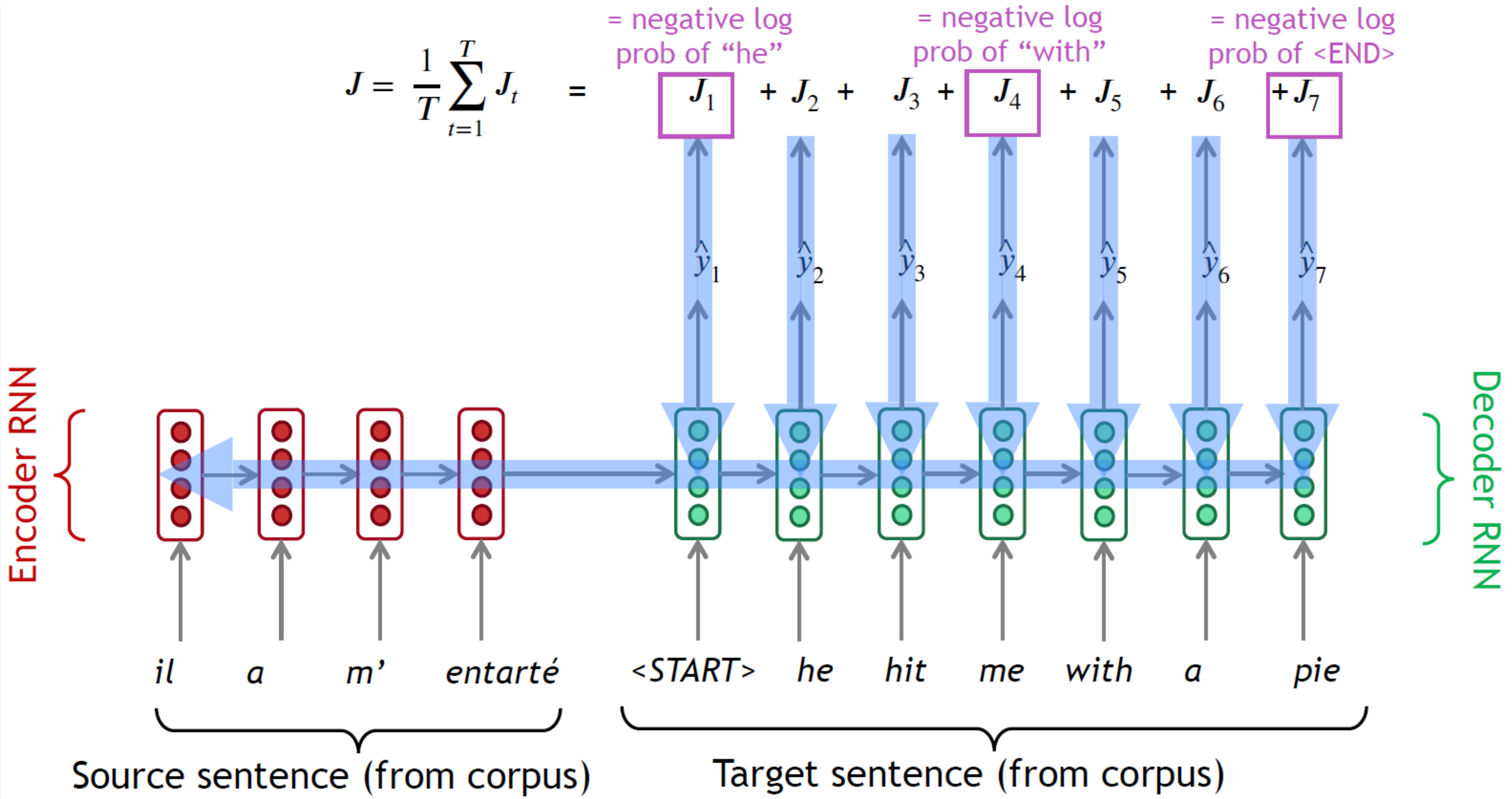
- The sequence-to-sequence model is an example of a **Conditional Language Model**.
  - **Language Model** because the decoder is predicting the next word of the target sentence *y*
  - **Conditional** because its predictions are *also* conditioned on the source sentence *x*

- NMT directly calculates $P(y|x)$:

$$P(y|x) = P(y_1|x)\,P(y_2|y_1,x)\,P(y_3|y_1,y_2,x)\ldots\underbrace{P(y_T|y_1,\ldots,y_{T-1},x)}_{\text{Probability of next target word, given target words so far and source sentence } x}$$

- **Question**: How to train a NMT system?
- **Answer**: Get a big parallel corpus…

Training NMT



$$J = \frac{1}{T}\sum_{t=1}^{T} J_t \quad = \quad \boxed{J_1} + J_2 + J_3 + \boxed{J_4} + J_5 + J_6 \boxed{+ J_7}$$

= negative log prob of "he"

= negative log prob of "with"

= negative log prob of <END>

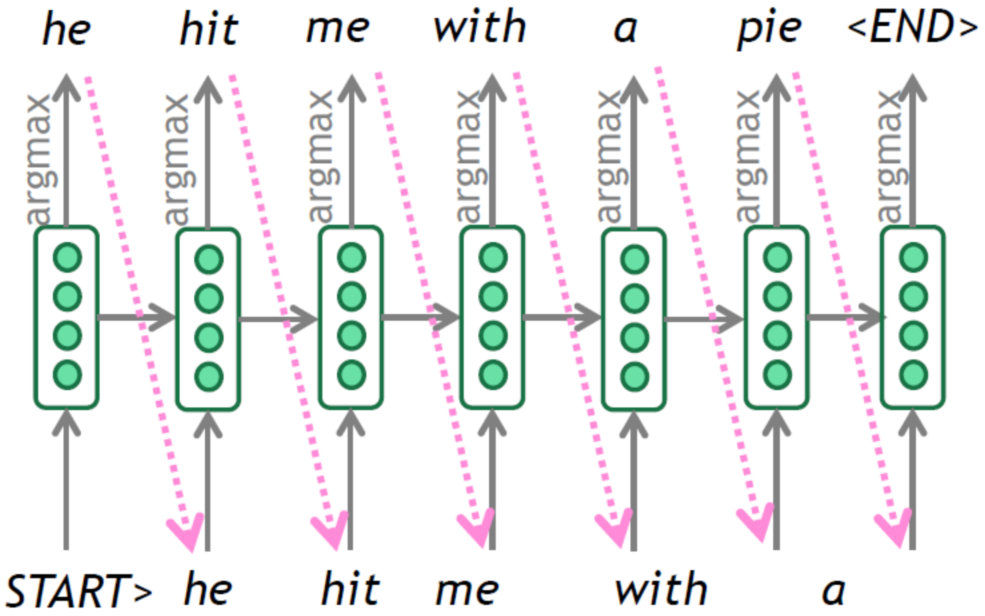
Source sentence (from corpus)

Target sentence (from corpus)

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

Decoding

- We saw how to generate (or “decode”) the target sentence by taking argmax on each step of the decoder
- This is greedy decoding (take most probable word on each step)
- Problems with this method?

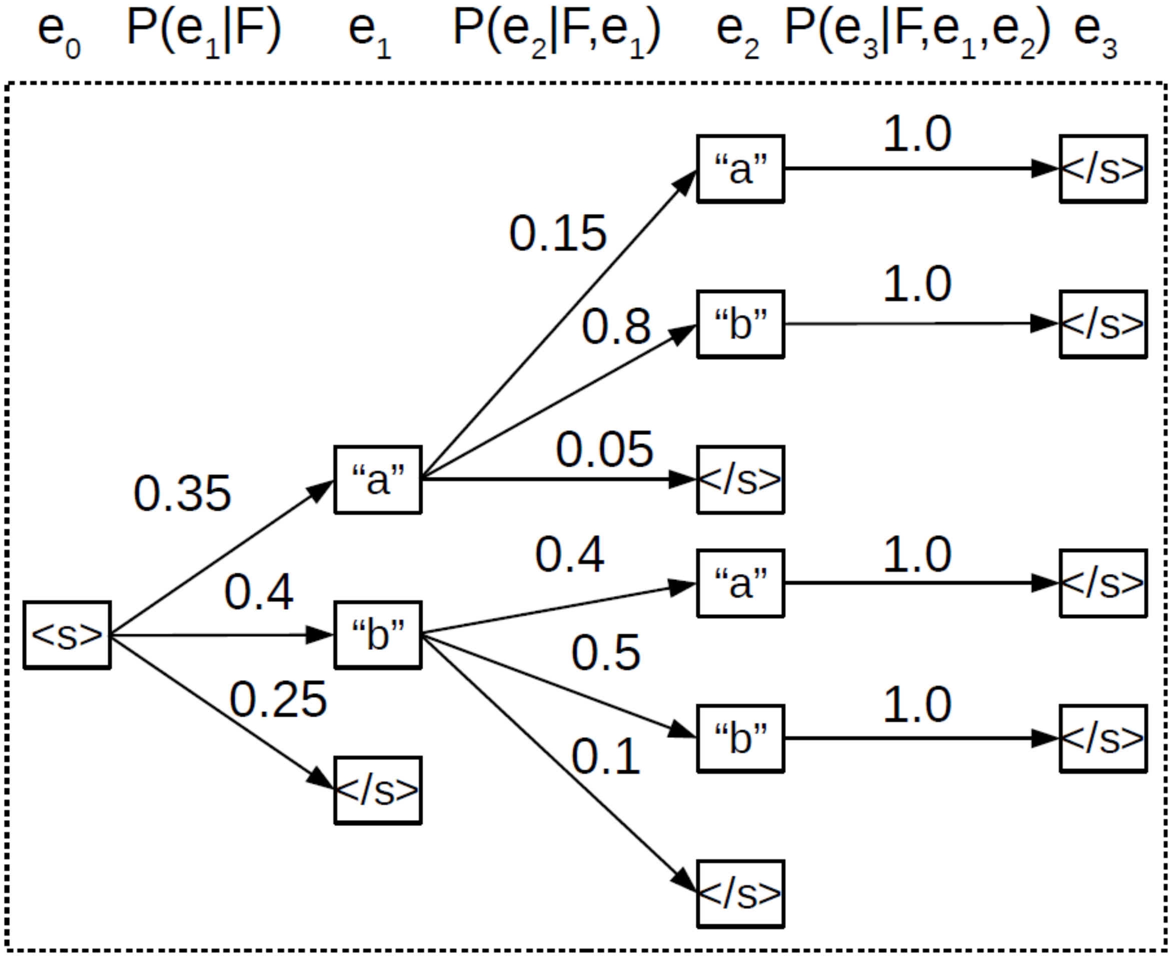


Problems with greedy decoding

- Greedy decoding has no way to undo decisions!
- Input: *il a m’entarté (he hit me with a pie)*
- → *he* ____
- → *he hit* ____
- → *he hit a* ____ (whoops! no going back now…)
- How to fix this?

Greedy prediction

- Example: greedy 1-best does not return the most probable sequence



Exhaustive search

- Ideally we want to find a (length *T*) translation *y* that maximizes

$$P(y|x) = P(y_1|x)\,P(y_2|y_1,x)\,P(y_3|y_1,y_2,x)\ldots,P(y_T|y_1,\ldots,y_{T-1},x)$$
$$= \prod_{t=1}^{T} P(y_t|y_1,\ldots,y_{t-1},x)$$

- We could try computing all possible sequences *y*
- This means that on each step *t* of the decoder, we're tracking Vt possible partial translations, where *V* is vocab size
- This O(VT) complexity is far too expensive!

Beam search decoding

- Core idea: On each step of decoder, keep track of the *k* most probable partial translations (which we call *hypotheses*)
- *k* is the beam size (in practice around 5 to 10)
- A hypothesis has a score which is its log probability:

$$\text{score}(y_1, \ldots, y_t) = \log P_{\text{LM}}(y_1, \ldots, y_t | x) = \sum_{i=1}^{t} \log P_{\text{LM}}(y_i | y_1, \ldots, y_{i-1}, x)$$

- Scores are all negative, and higher score is better
- We search for high-scoring hypotheses, tracking top *k* on each step
- Beam search is not guaranteed to find optimal solution
- But much more efficient than exhaustive search!

Beam search decoding: example

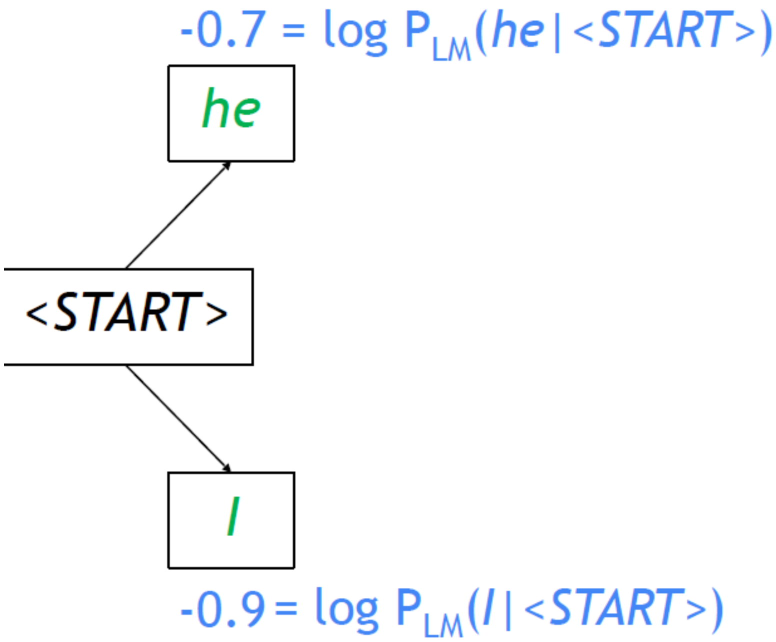
Beam size = k = 2. Blue numbers = $\text{score}(y_1, \ldots, y_t) = \sum_{i=1}^{t} \log P_{\text{LM}}(y_i | y_1, \ldots, y_{i-1}, x)$

<START>

Calculate prob dist of next word

Beam search decoding: example

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

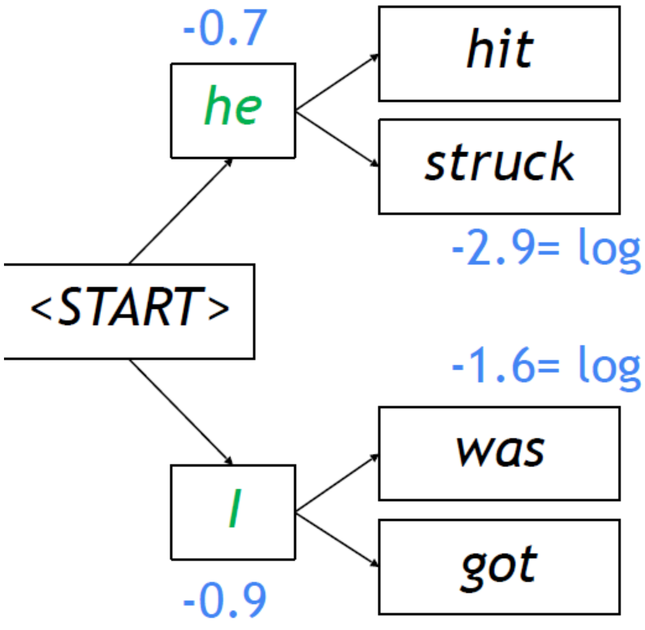

-0.7 = log P_{LM}(*he* | <*START*>)
he
<*START*>
I
-0.9 = log P_{LM}(*I* | <*START*>)

Take top *k* words and compute scores

Beam search decoding: example

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

-1.7 = log P_{LM}(*hit*|*<START> he*) + -0.7



-2.9= log P_{LM}(*struck*|*<START> he*) + -0.7

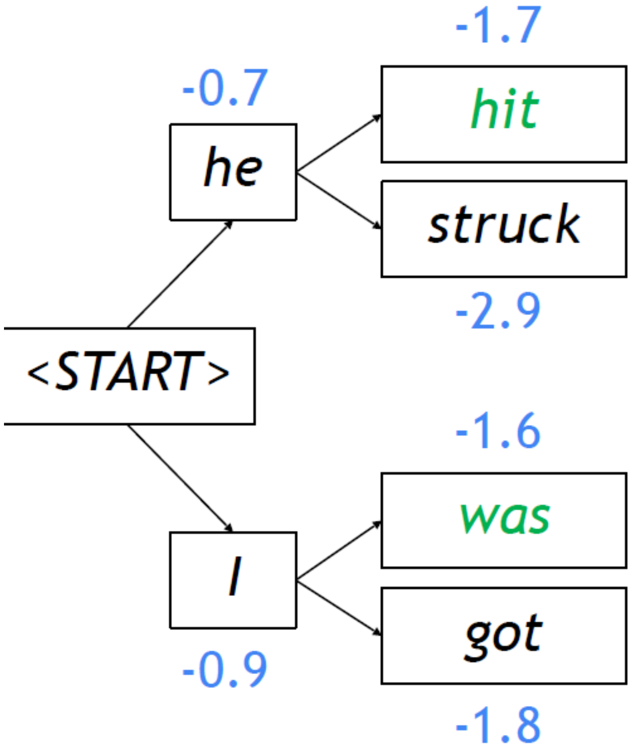
-1.6= log P_{LM}(*was*|*<START> I*) + -0.9

-1.8= log P_{LM}(*got*|*<START> I*) + -0.9

For each of the *k* hypotheses, find top *k* next words and calculate scores

Beam search decoding: example

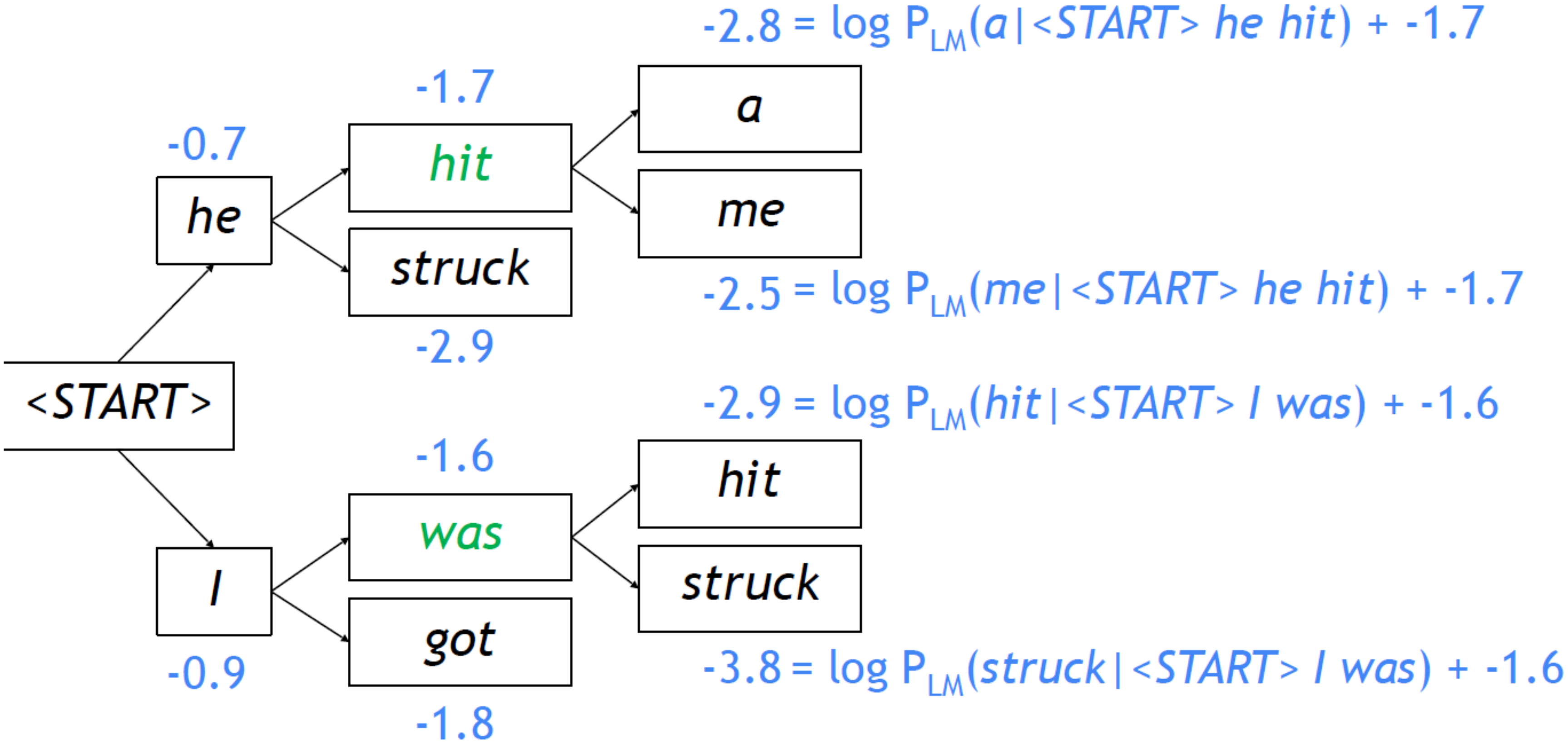
Beam size = k = 2. Blue numbers = $\text{score}(y_1, \ldots, y_t) = \sum_{i=1}^{t} \log P_{\text{LM}}(y_i | y_1, \ldots, y_{i-1}, x)$



Of these k^2 hypotheses, just keep k with highest scores

Beam search decoding: example

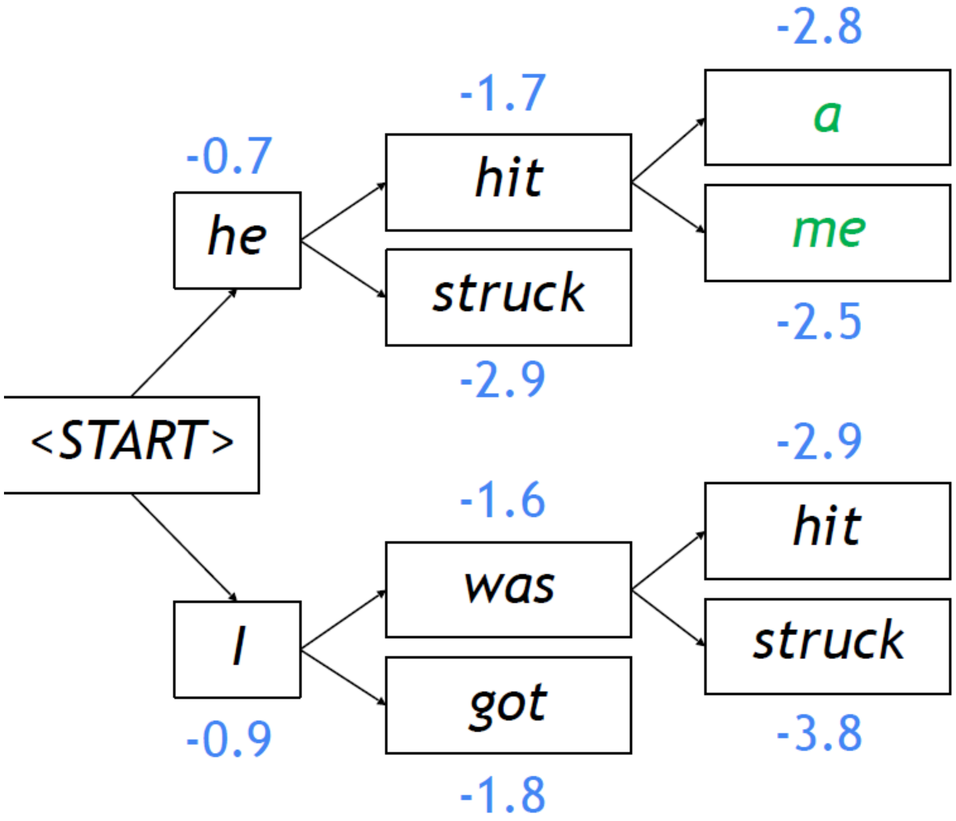
Beam size = k = 2. Blue numbers = $\text{score}(y_1, \ldots, y_t) = \sum_{i=1}^{t} \log P_{\text{LM}}(y_i | y_1, \ldots, y_{i-1}, x)$


-2.8 = log P_{LM}(*a* | *<START> he hit*) + -1.7
-1.7
a
-0.7
hit
he
me
struck
-2.5 = log P_{LM}(*me* | *<START> he hit*) + -1.7
-2.9
<START>
-2.9 = log P_{LM}(*hit* | *<START> I was*) + -1.6
-1.6
hit
was
struck
I
got
-3.8 = log P_{LM}(*struck* | *<START> I was*) + -1.6
-0.9
-1.8

For each of the *k* hypotheses, find top *k* next words and calculate scores

Beam search decoding: example

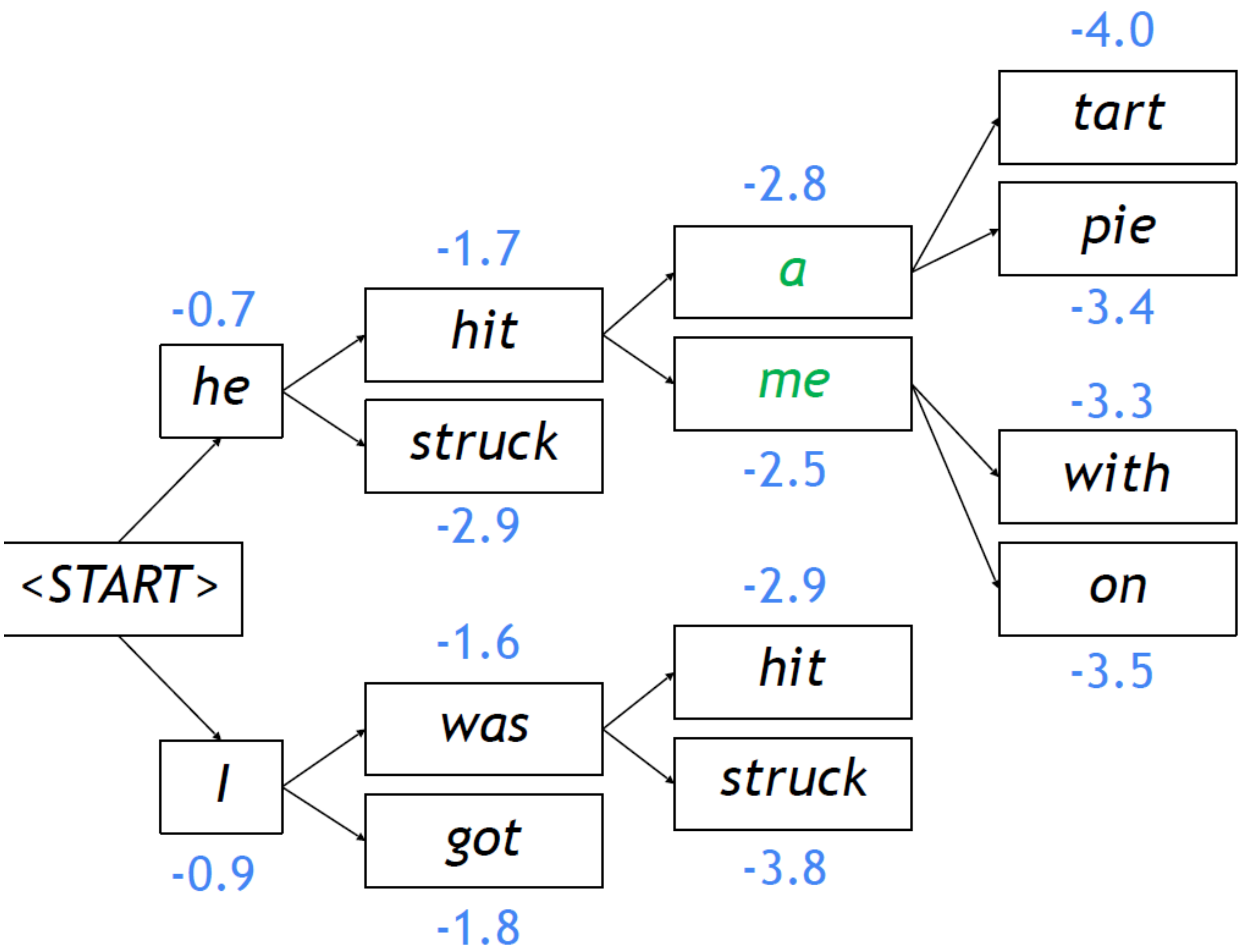
Beam size = k = 2. Blue numbers = $\text{score}(y_1, \ldots, y_t) = \sum_{i=1}^{t} \log P_{\text{LM}}(y_i | y_1, \ldots, y_{i-1}, x)$



Of these k^2 hypotheses, just keep *k* with highest scores

Beam search decoding: example

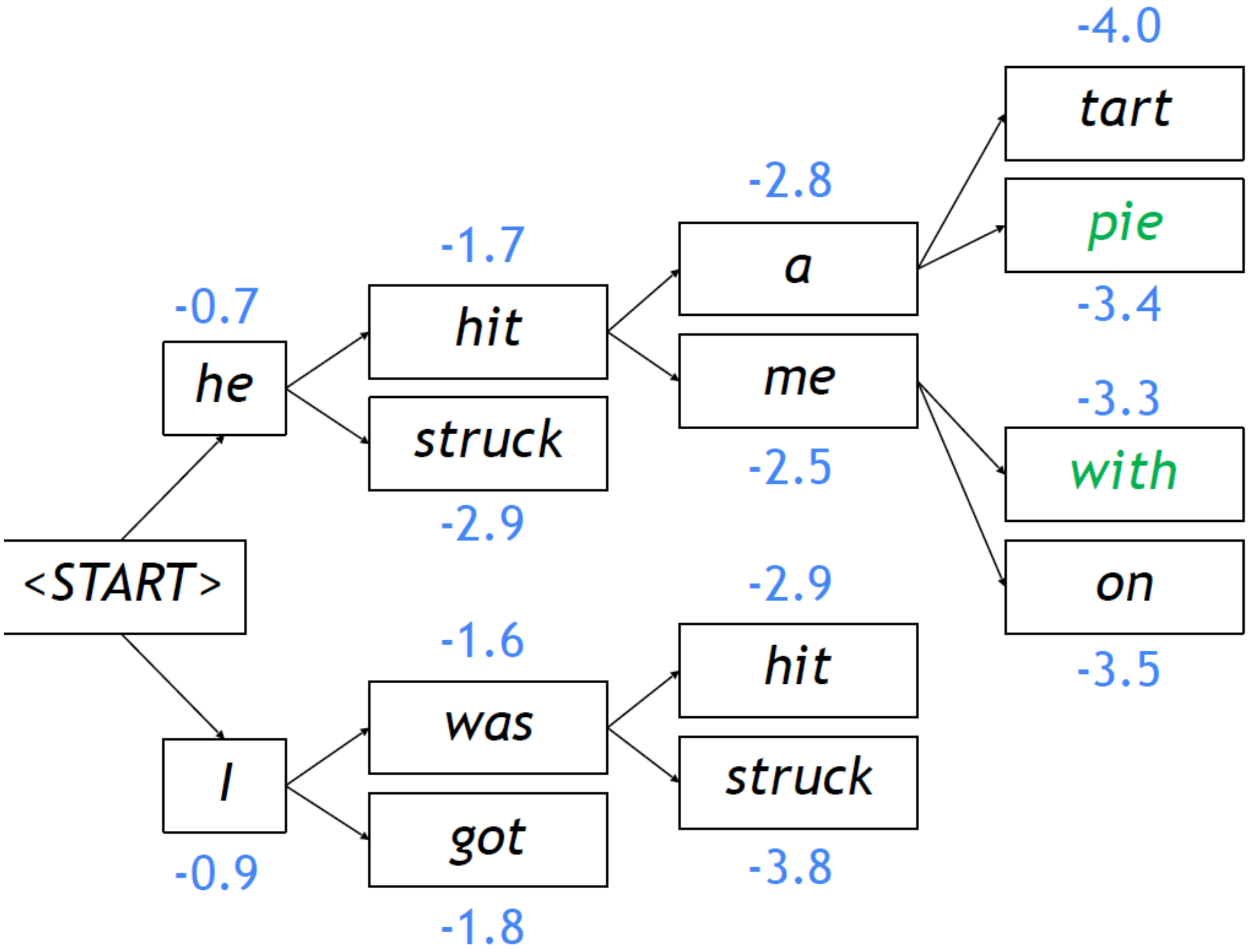
Beam size = k = 2. Blue numbers = $\text{score}(y_1, \ldots, y_t) = \sum_{i=1}^{t} \log P_{\text{LM}}(y_i | y_1, \ldots, y_{i-1}, x)$



For each of the *k* hypotheses, find top *k* next words and calculate scores

Beam search decoding: example

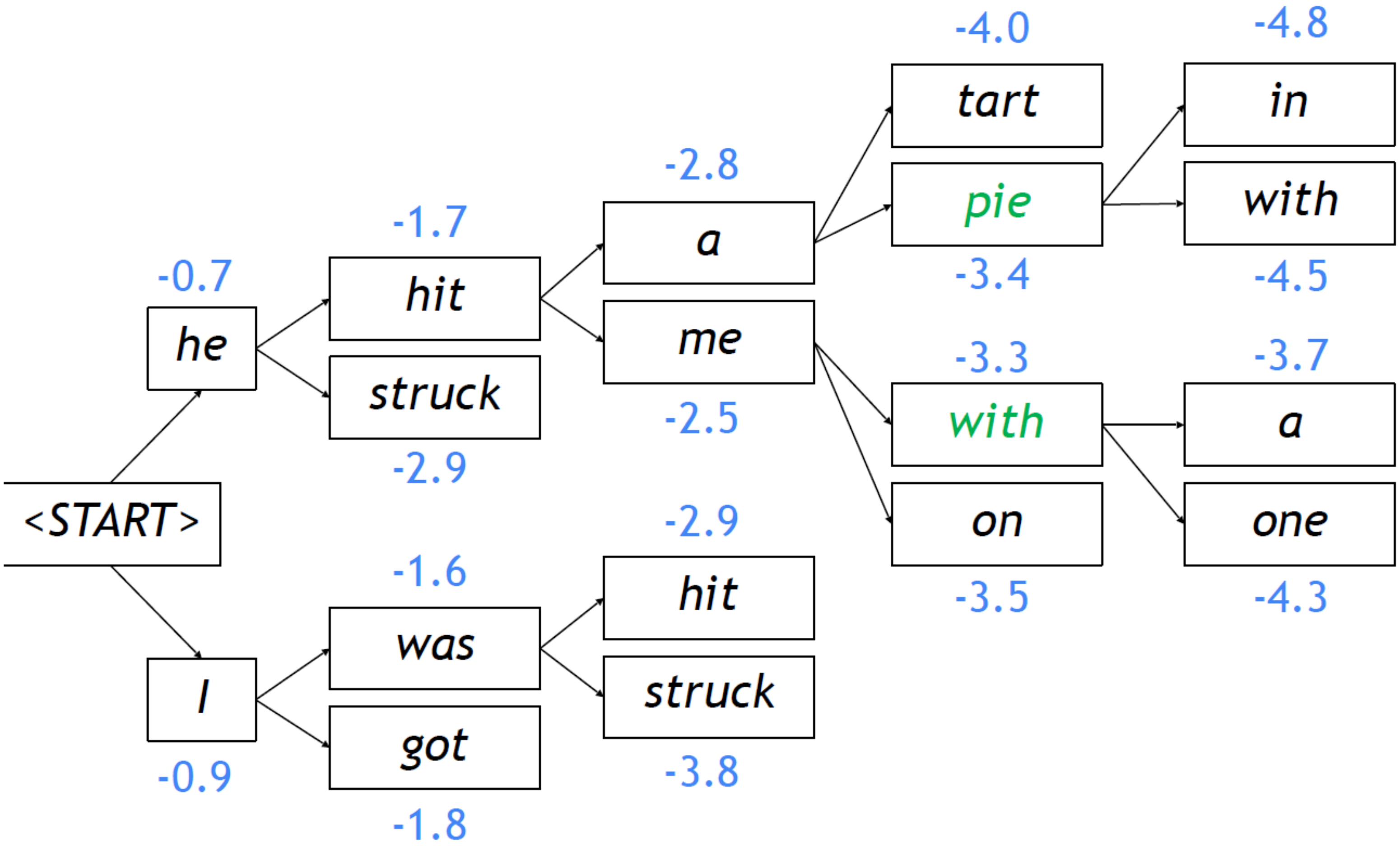
Beam size = k = 2. Blue numbers = $\text{score}(y_1, \ldots, y_t) = \sum_{i=1}^{t} \log P_{\text{LM}}(y_i | y_1, \ldots, y_{i-1}, x)$



Of these k^2 hypotheses, just keep *k* with highest scores

Beam search decoding: example

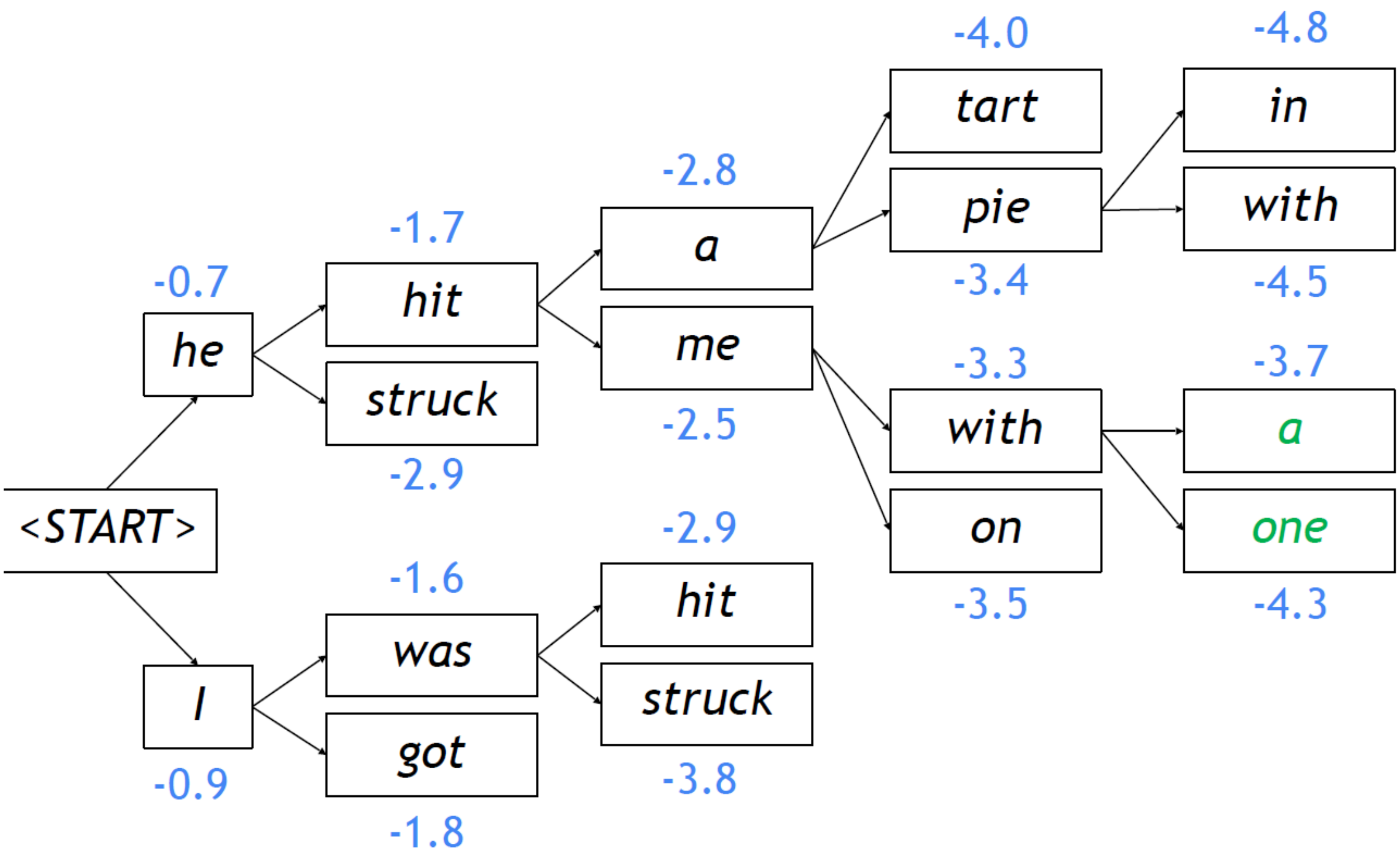
Beam size = k = 2. Blue numbers = $\text{score}(y_1, \ldots, y_t) = \sum_{i=1}^{t} \log P_{\text{LM}}(y_i | y_1, \ldots, y_{i-1}, x)$



For each of the *k* hypotheses, find top *k* next words and calculate scores

Beam search decoding: example

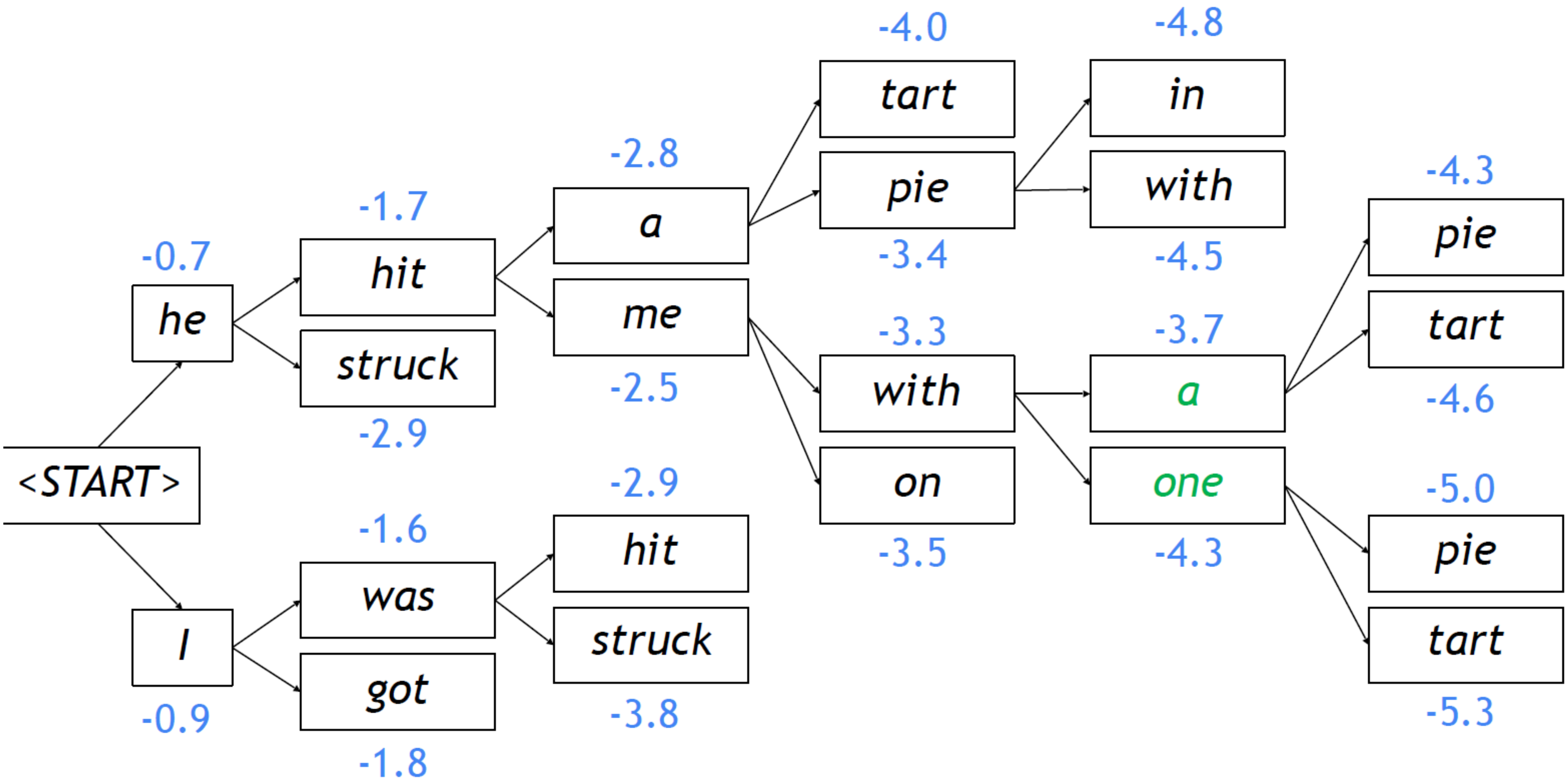
Beam size = k = 2. Blue numbers = $\text{score}(y_1, \ldots, y_t) = \sum_{i=1}^{t} \log P_{\text{LM}}(y_i | y_1, \ldots, y_{i-1}, x)$



Of these k^2 hypotheses, just keep k with highest scores

Beam search decoding: example

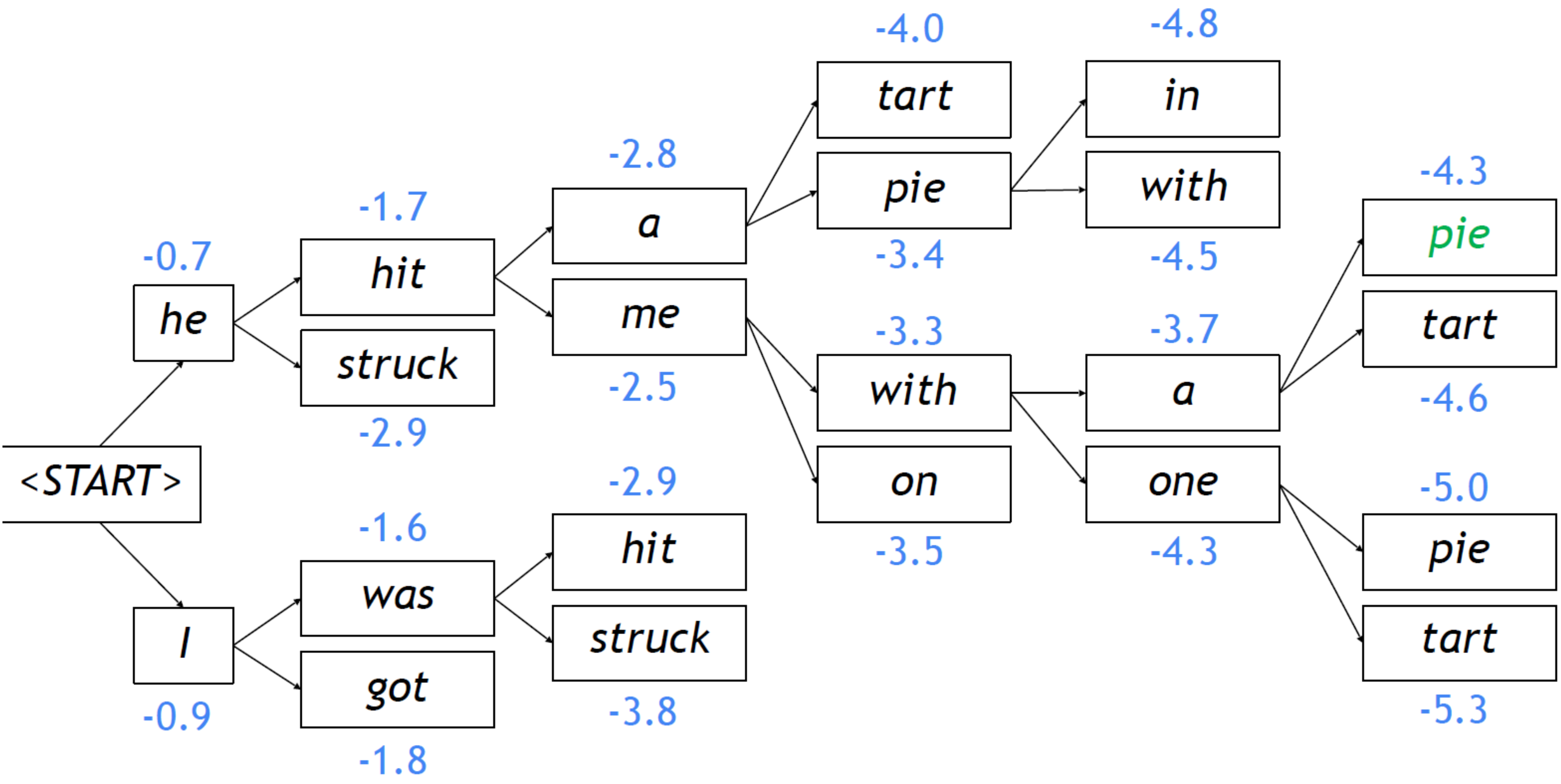
Beam size = k = 2. Blue numbers = $\text{score}(y_1, \ldots, y_t) = \sum_{i=1}^{t} \log P_{\text{LM}}(y_i | y_1, \ldots, y_{i-1}, x)$



For each of the *k* hypotheses, find top *k* next words and calculate scores

Beam search decoding: example

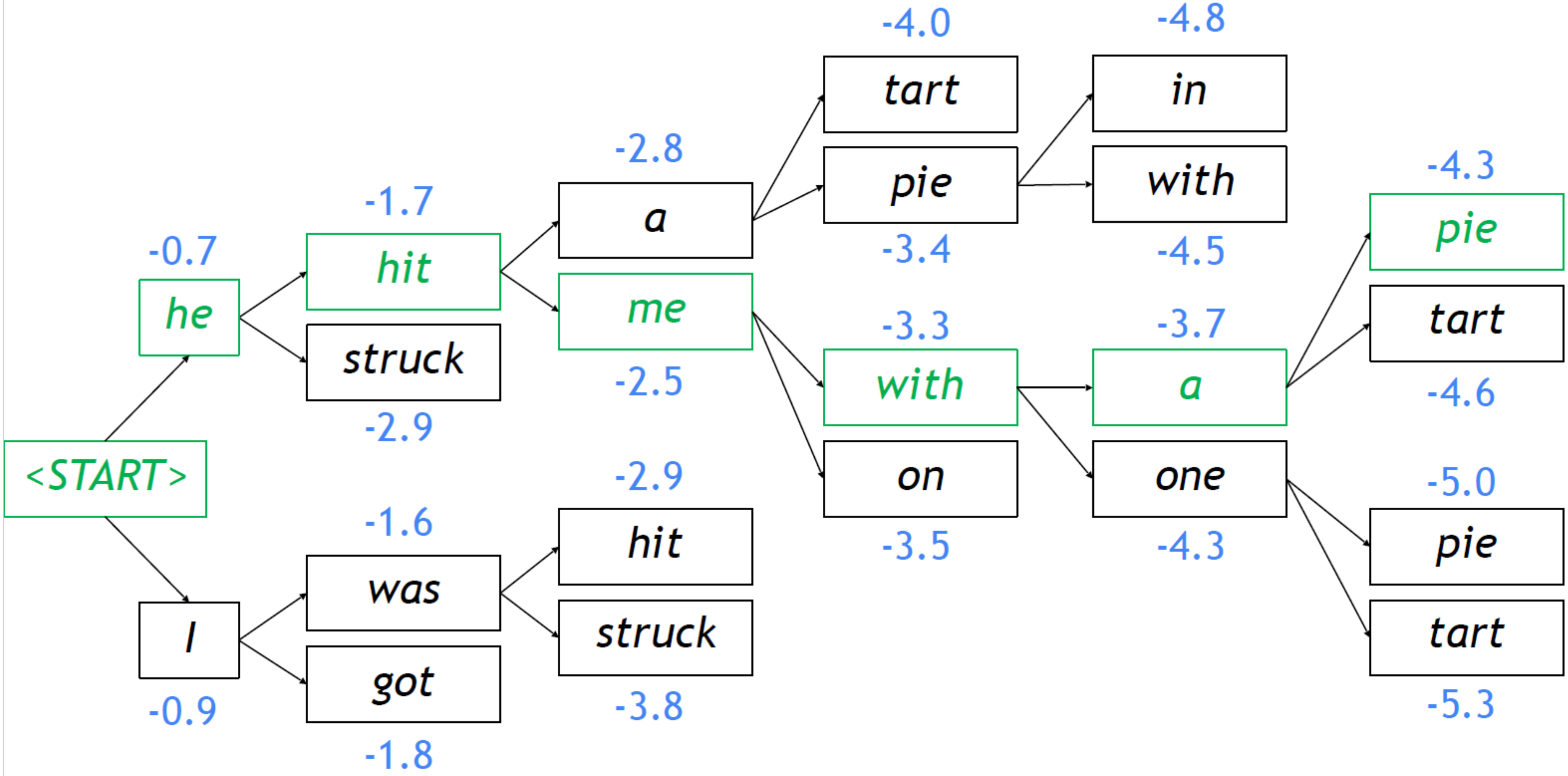
Beam size = k = 2. Blue numbers = $\mathrm{score}(y_1, \ldots, y_t) = \sum_{i=1}^{t} \log P_{\mathrm{LM}}(y_i | y_1, \ldots, y_{i-1}, x)$



This is the top-scoring hypothesis!

Beam search decoding: example

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



Backtrack to obtain the full hypothesis

Beam search decoding: stopping criterion

- In greedy decoding, usually we decode until the model produces a <END> token
  - For example: *<START> he hit me with a pie <END>*
- In beam search decoding, different hypotheses may produce <END> tokens on different time steps
  - When a hypothesis produces <END>, that hypothesis is complete.
  - Place it aside and continue exploring other hypotheses via beam search.
- Usually we continue beam search until:
  - We reach time step *T* (where *T* is some pre-defined cutoff), or
  - We have at least *n* completed hypotheses (where *n* is pre-defined cutoff)

Beam search decoding: finishing up

- We have our list of completed hypotheses.
- How to select top one with highest score?
- Each hypothesis $y_1, \ldots, y_t$ on our list has a score

$$\text{score}(y_1, \ldots, y_t) = \log P_{\text{LM}}(y_1, \ldots, y_t | x) = \sum_{i=1}^{t} \log P_{\text{LM}}(y_i | y_1, \ldots, y_{i-1}, x)$$

- Problem with this: longer hypotheses have lower scores
- Fix: normalize by length. Use this to select top one instead:

$$\frac{1}{t} \sum_{i=1}^{t} \log P_{\text{LM}}(y_i | y_1, \ldots, y_{i-1}, x)$$

What’s the effect of changing beam size *k*?

- Small *k* has similar problems to greedy decoding (*k*=1)
  - Ungrammatical, unnatural, nonsensical, incorrect
- Larger *k* means you consider more hypotheses
  - Increasing *k* reduces some of the problems above
  - Larger *k* is more computationally expensive
    - But increasing *k* can introduce other problems:
    - For NMT, increasing *k* too much decreases BLEU score (Tu et al, Koehn et al). This is primarily because large-k beam search produces too short translations (even with score normalization!)
    - It can even produce empty translations (Stahlberg & Byrne 2019)
    - In open-ended tasks like chit-chat dialogue, large *k* can make output more generic

Neural Machine Translation with Reconstruction, Tu et al, 2017 https://arxiv.org/pdf/1611.01874.pdf

Six Challenges for Neural Machine Translation, Koehn et al, 2017 https://arxiv.org/pdf/1706.03872.pdf

Effect of beam size in chit-chat dialogue

I mostly eat a fresh and raw diet, so I save on groceries

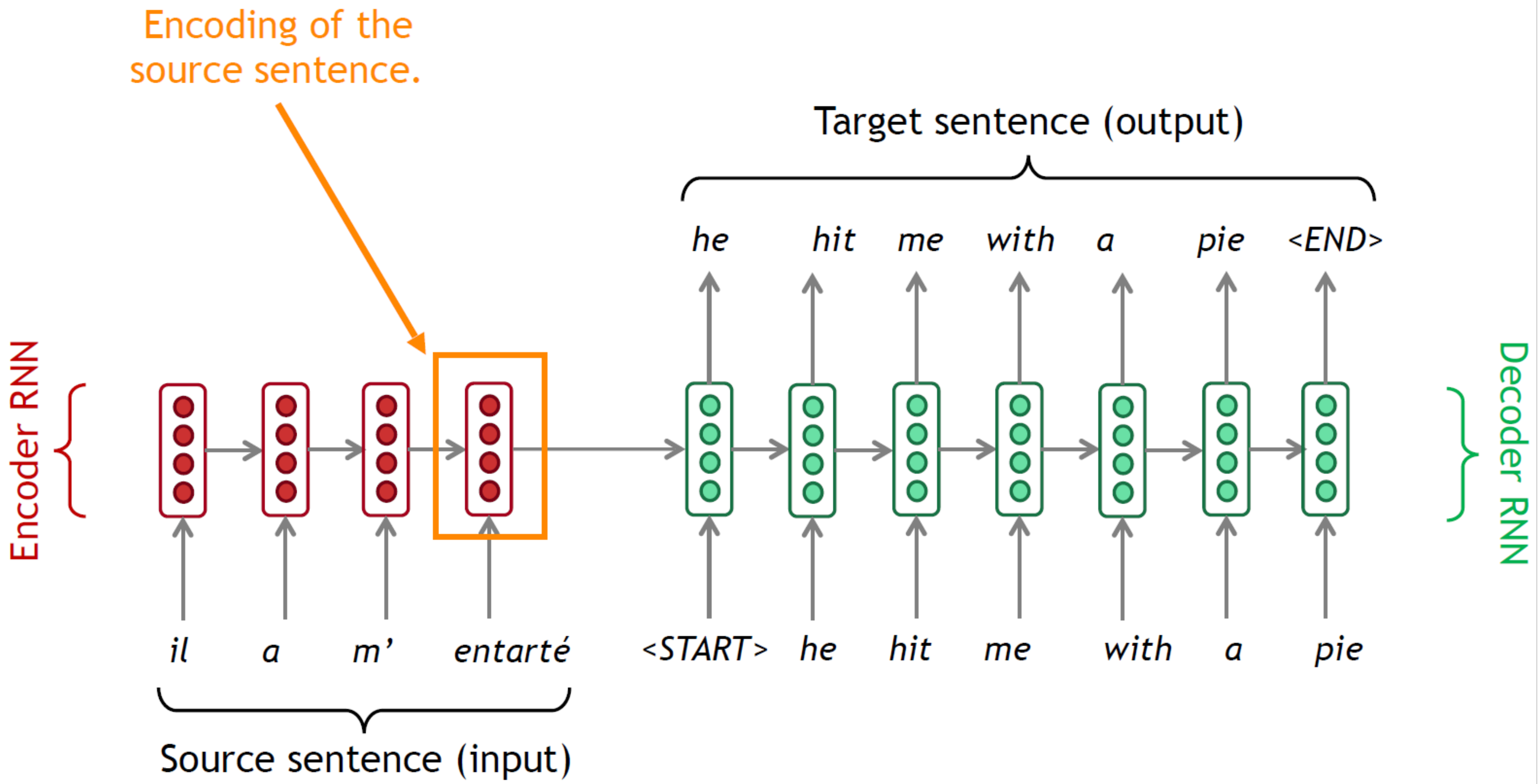
Human chit-chat partner

Beam size	Model response
1	*I love to eat healthy and eat healthy*
2	*That is a good thing to have*
3	*I am a nurse so I do not eat raw food*
4	*I am a nurse so I am a nurse*
5	*Do you have any hobbies?*
6	*What do you do for a living?*
7	*What do you do for a living?*
8	*What do you do for a living?*

Low beam size:
More on-topic but nonsensical;
bad English

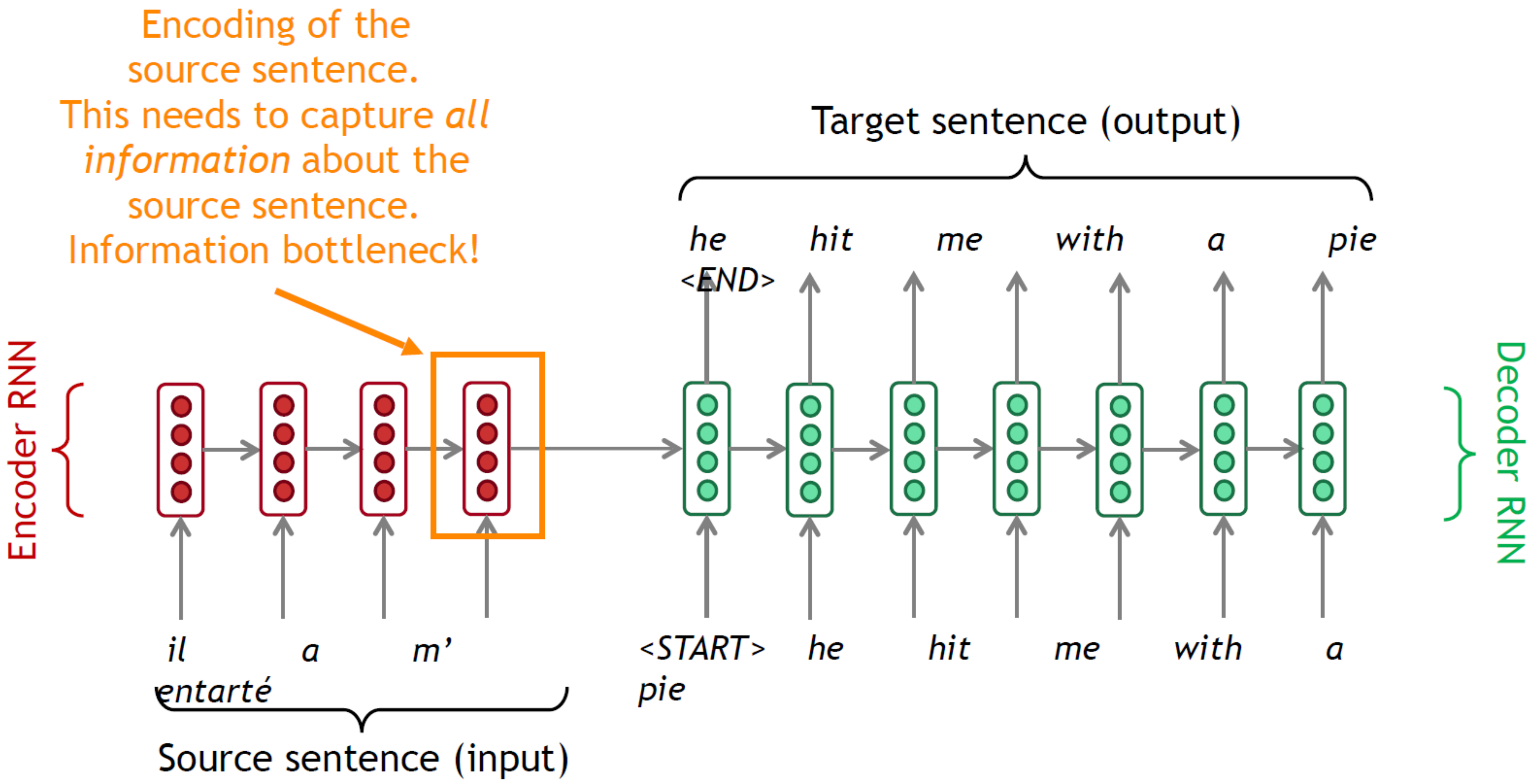
High beam size:
Converges to safe, "correct" response, but it's generic and less relevant

Sequence-to-sequence: the bottleneck problem



Problems with this architecture?

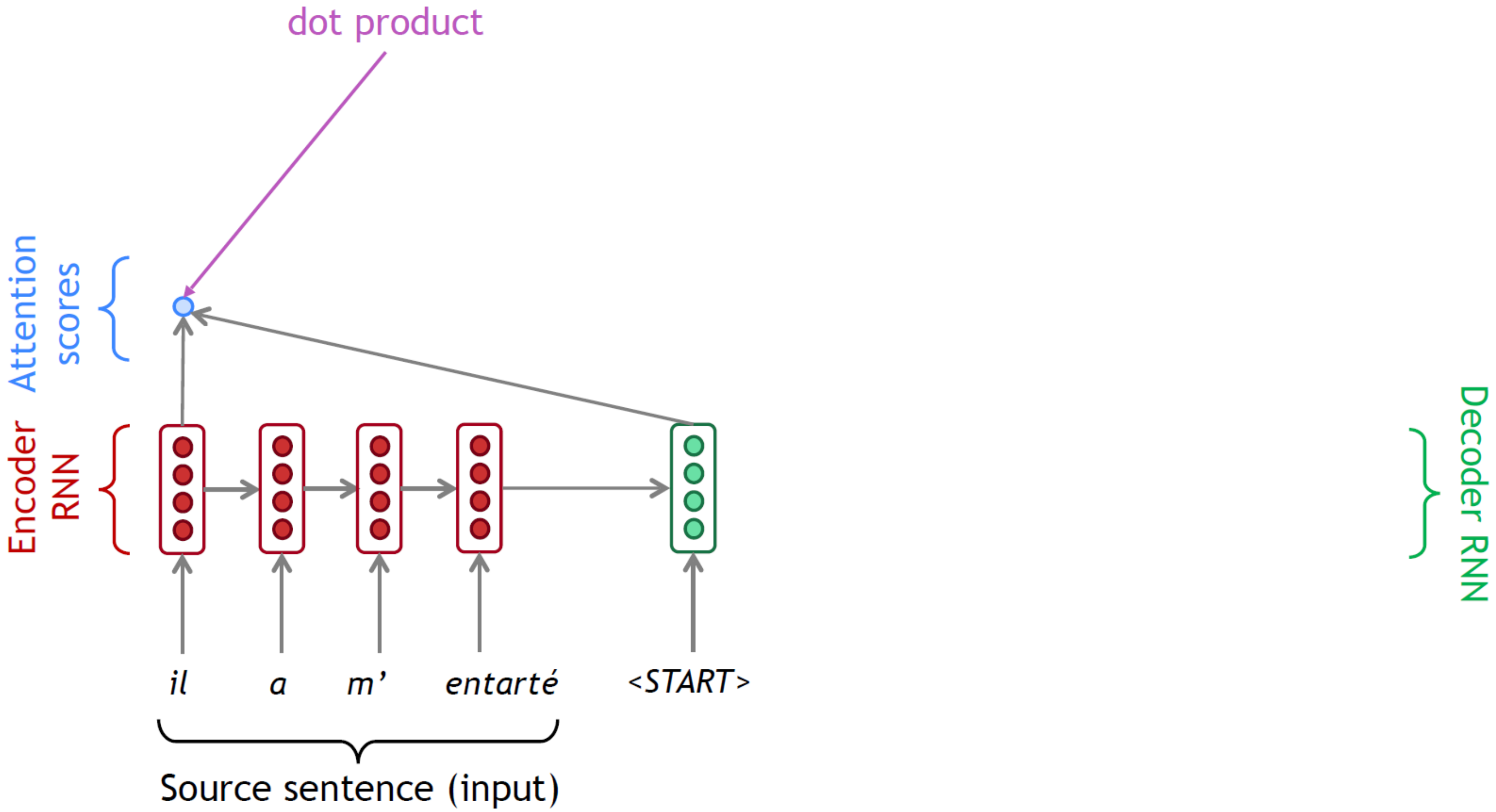
Sequence-to-sequence: the bottleneck problem



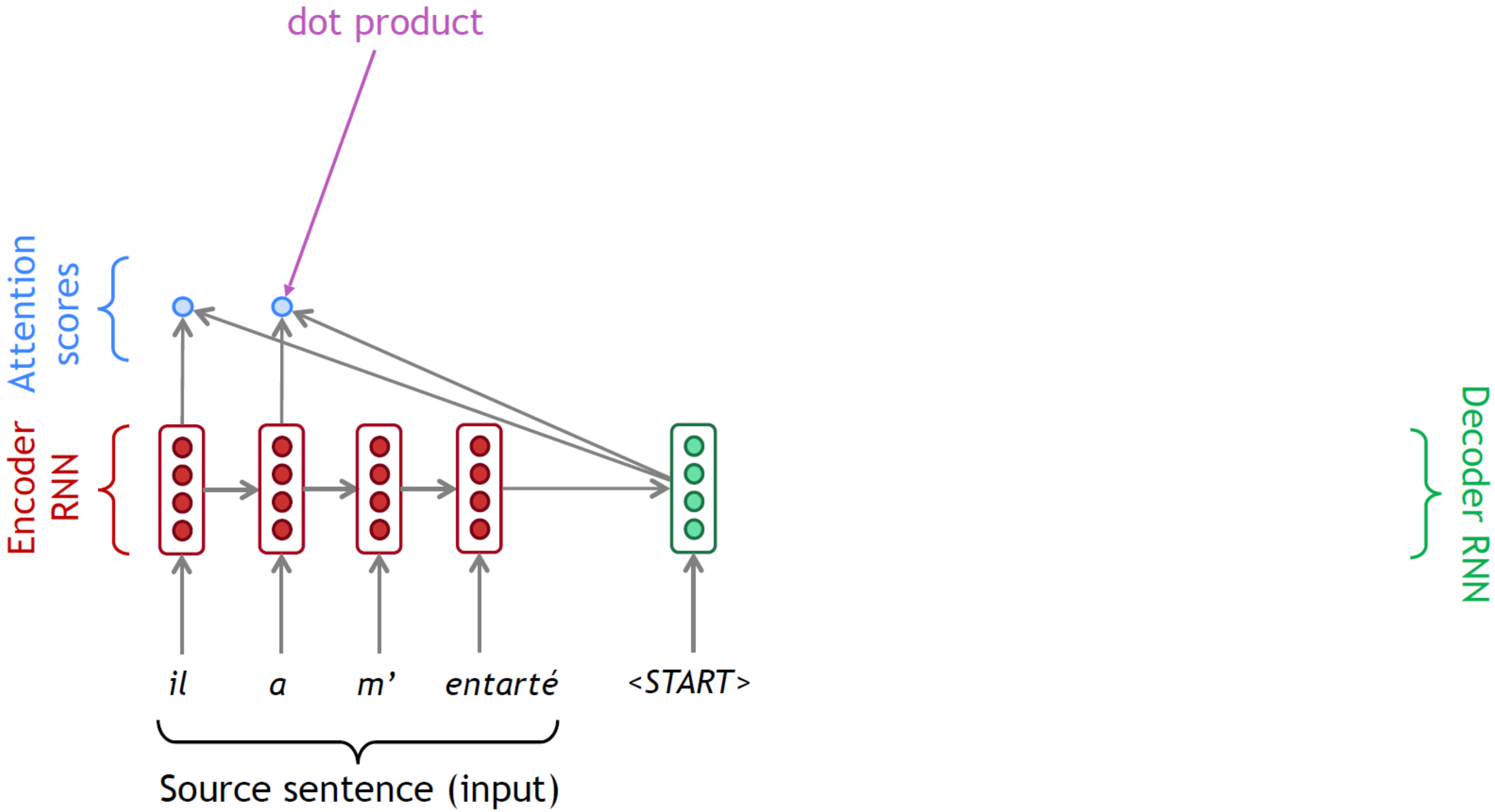
Attention

- Attention provides a solution to the bottleneck problem.
- Core idea: on each step of the decoder, use *direct connection to the encoder* to *focus on a particular part* of the source sequence

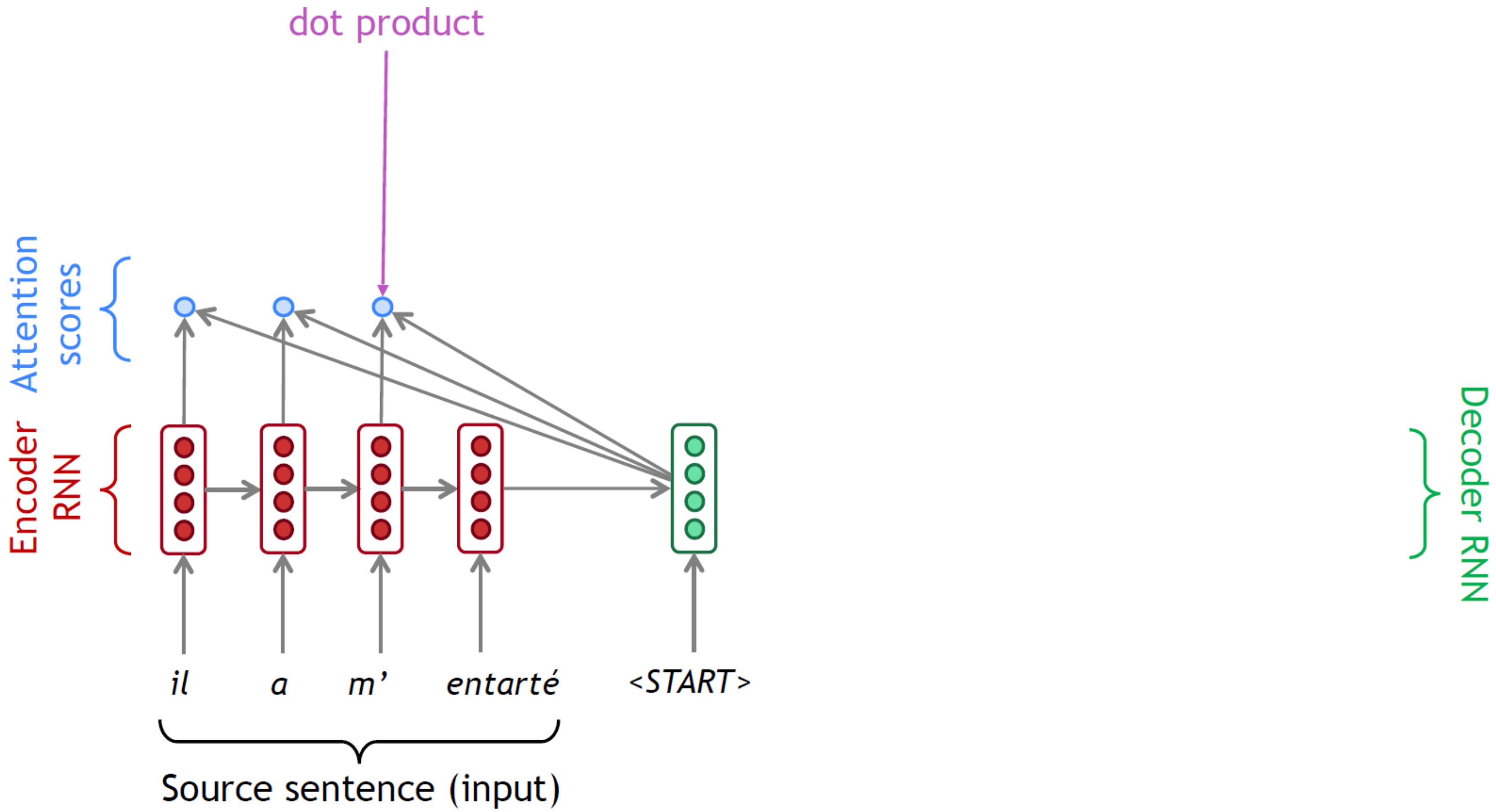
Sequence-to-sequence with attention



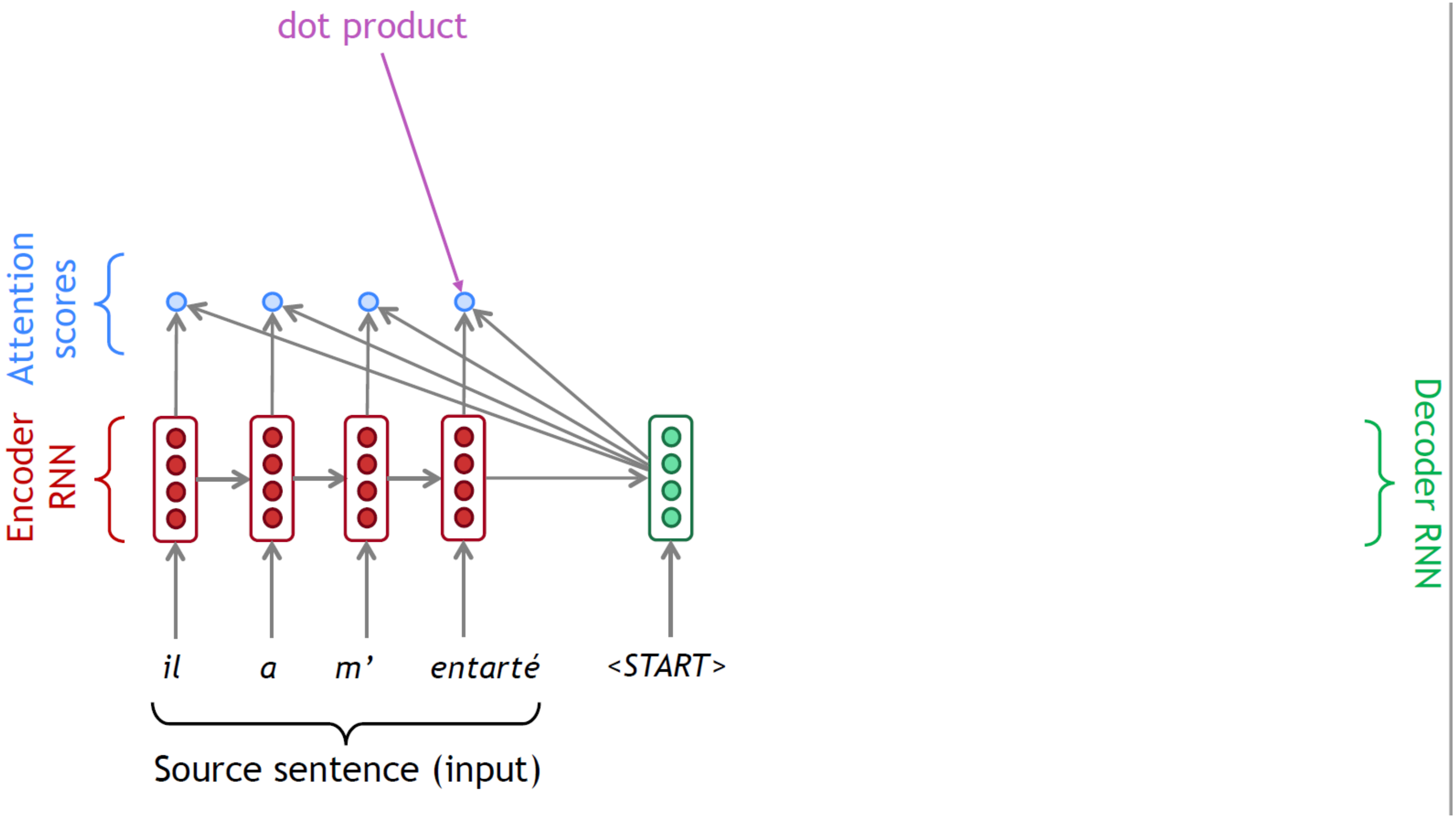
Sequence-to-sequence with attention



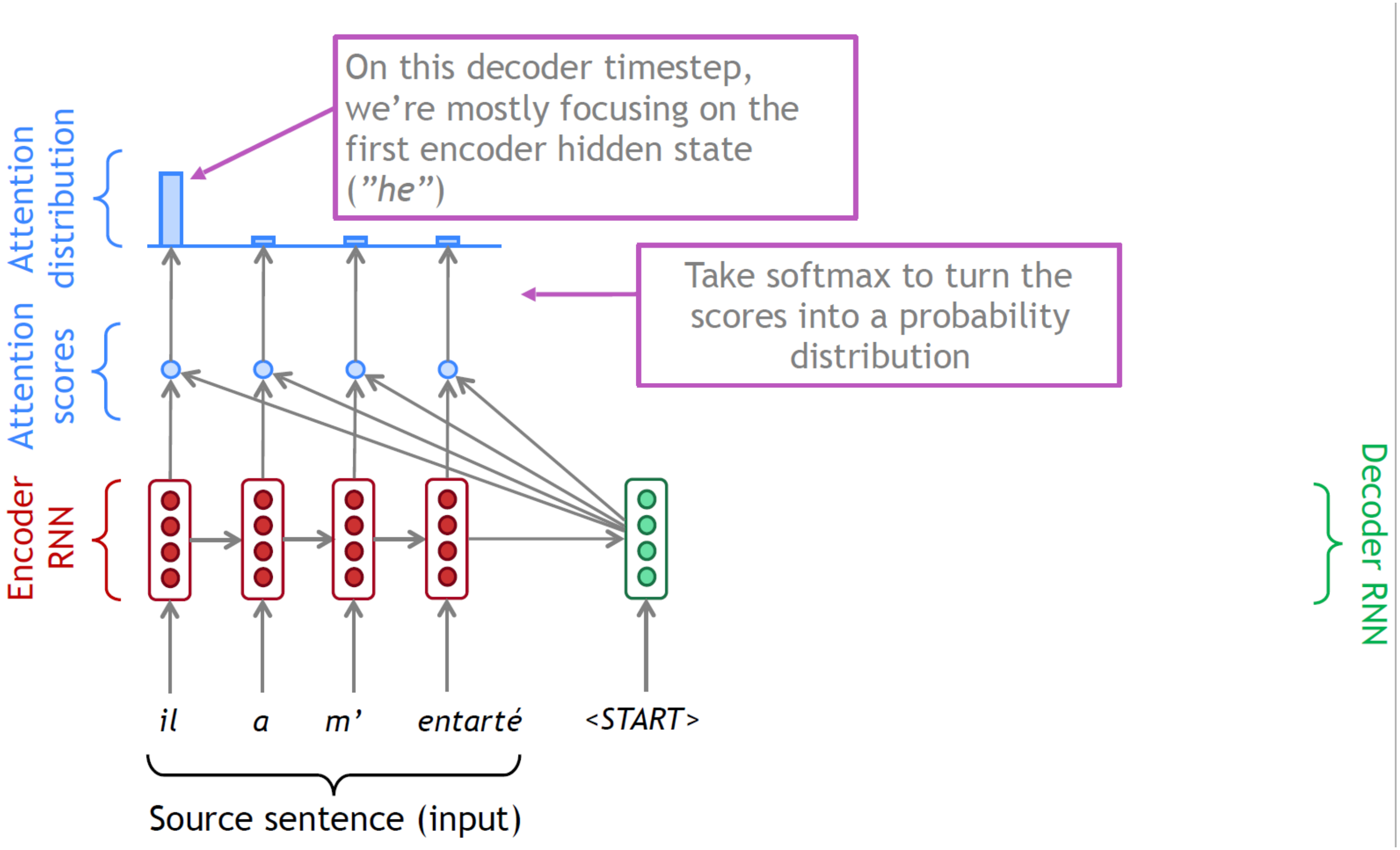
Sequence-to-sequence with attention



Sequence-to-sequence with attention



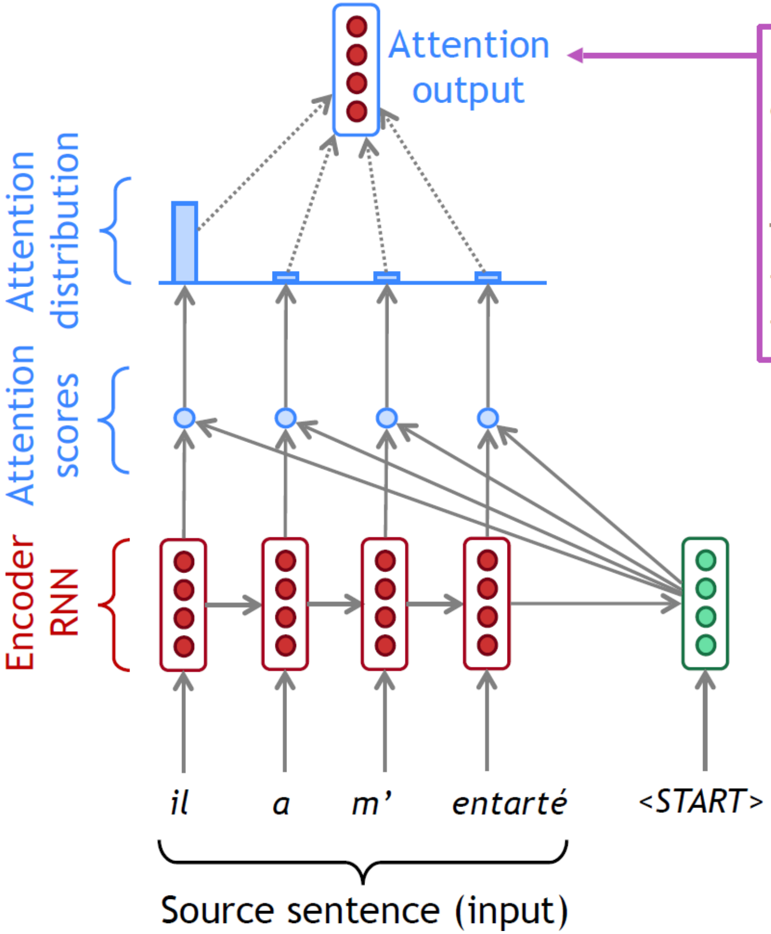
Sequence-to-sequence with attention



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

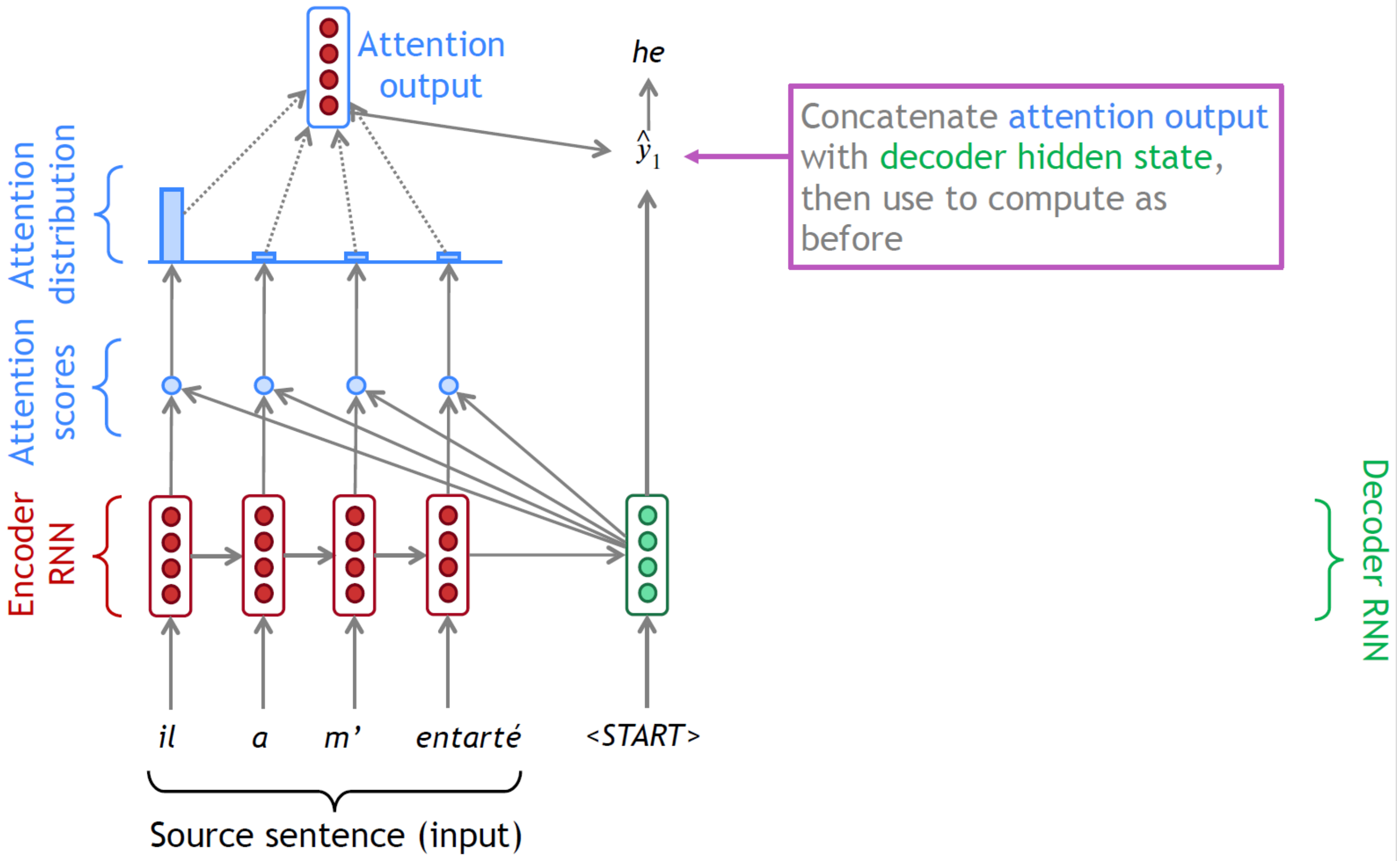
Sequence-to-sequence with attention



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

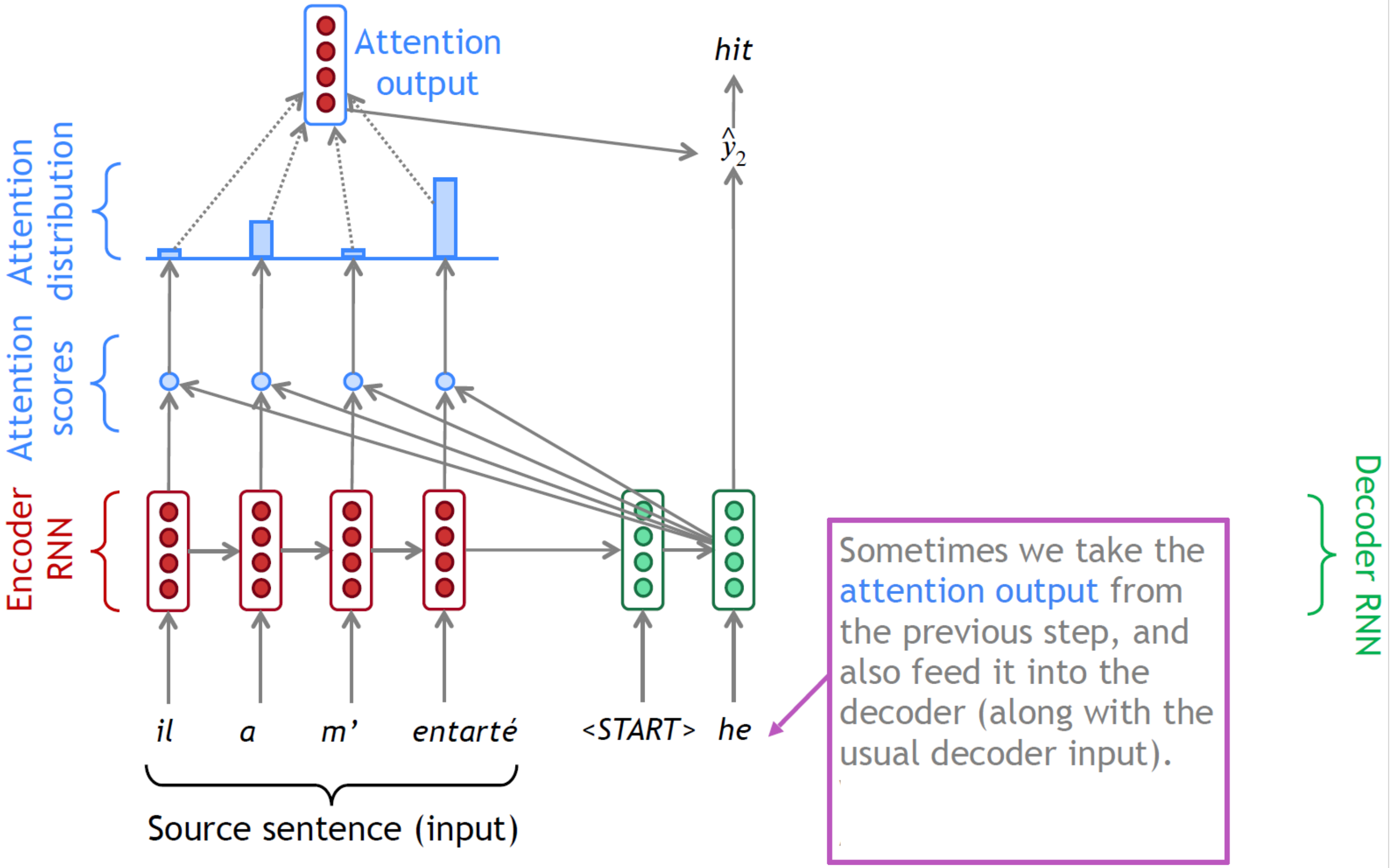
The attention output mostly contains information from the hidden states that received high attention.

Sequence-to-sequence with attention



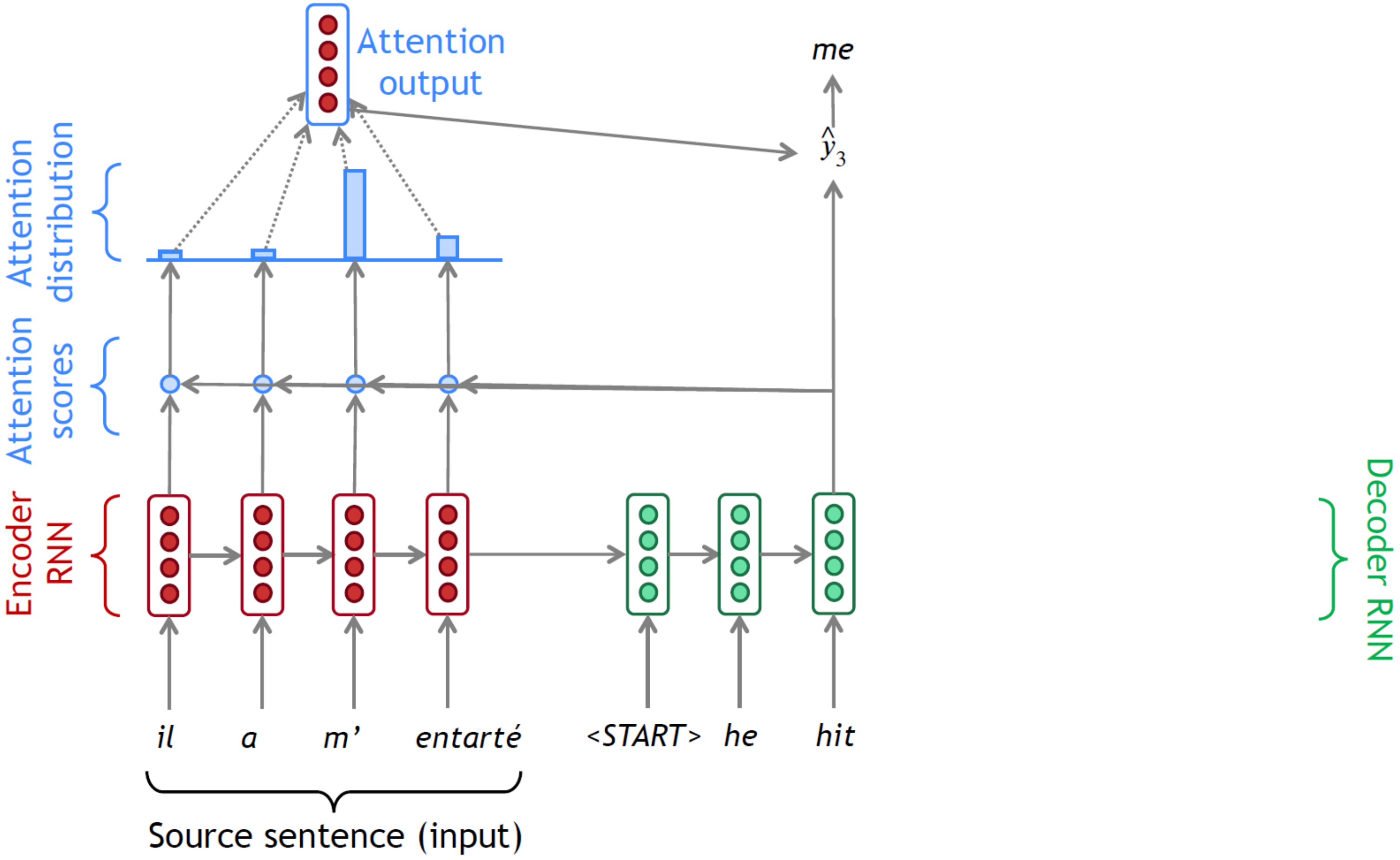
Concatenate attention output with decoder hidden state, then use to compute $\hat{y}_1$ as before

Sequence-to-sequence with attention

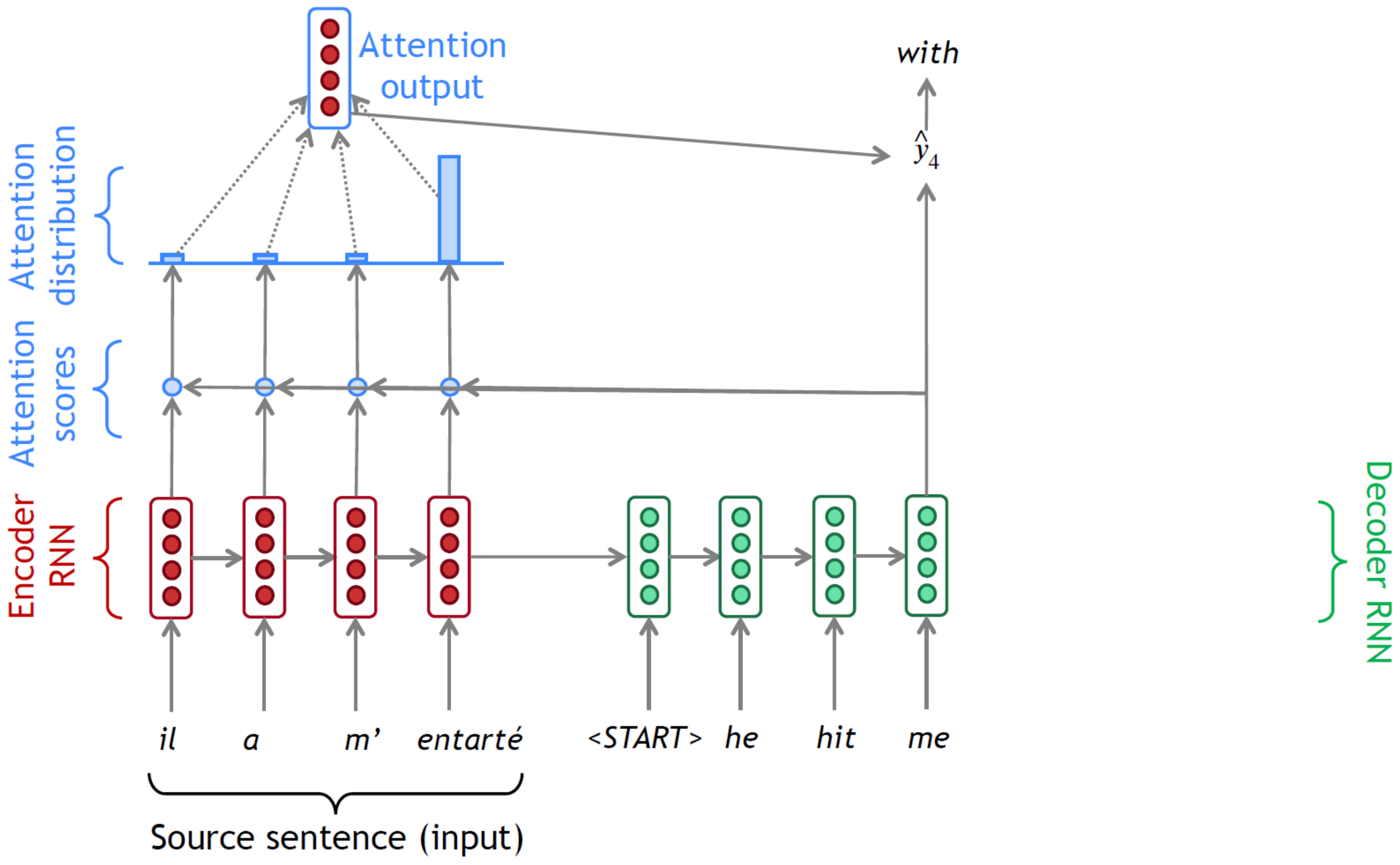


Sometimes we take the attention output from the previous step, and also feed it into the decoder (along with the usual decoder input).

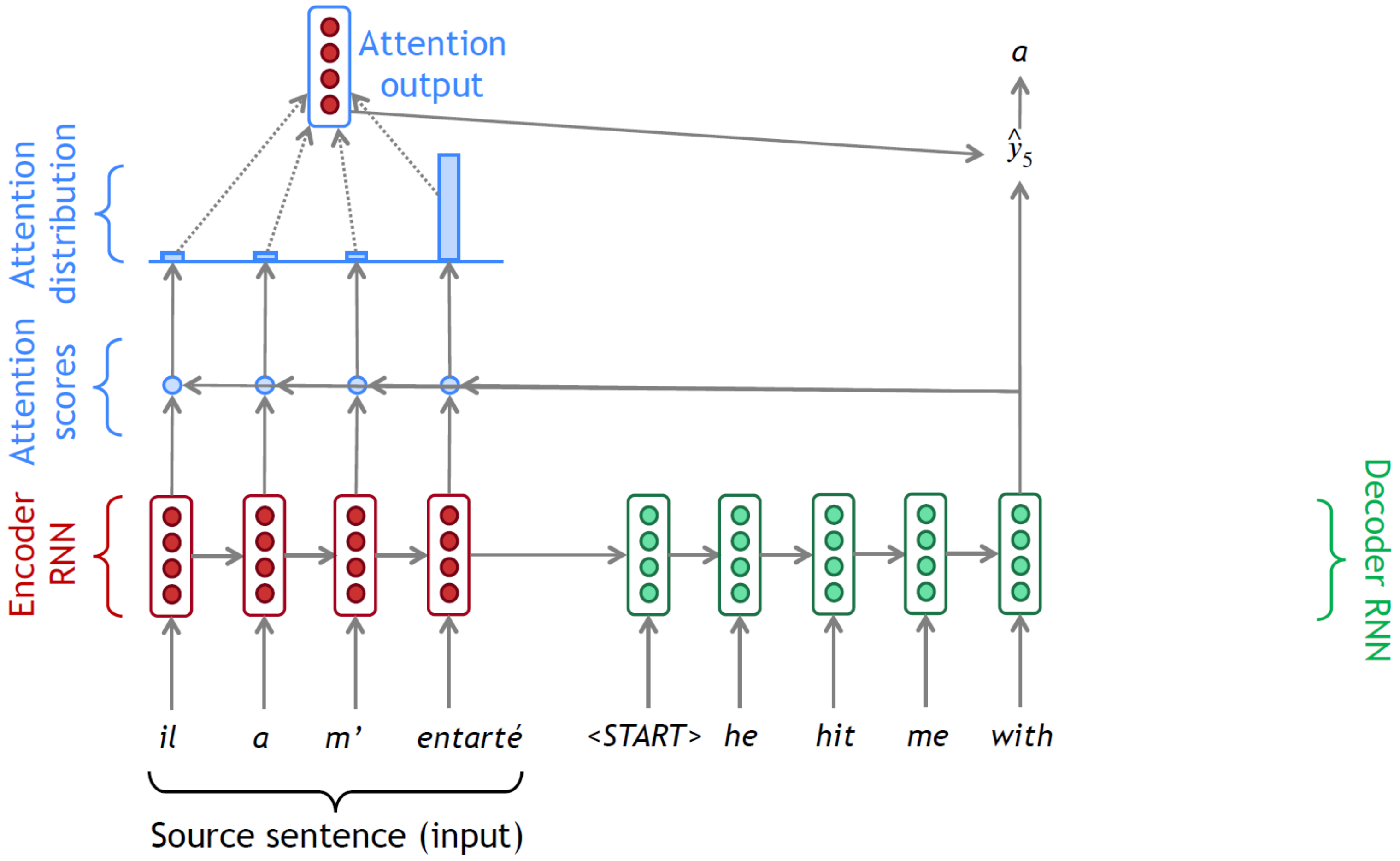
Sequence-to-sequence with attention



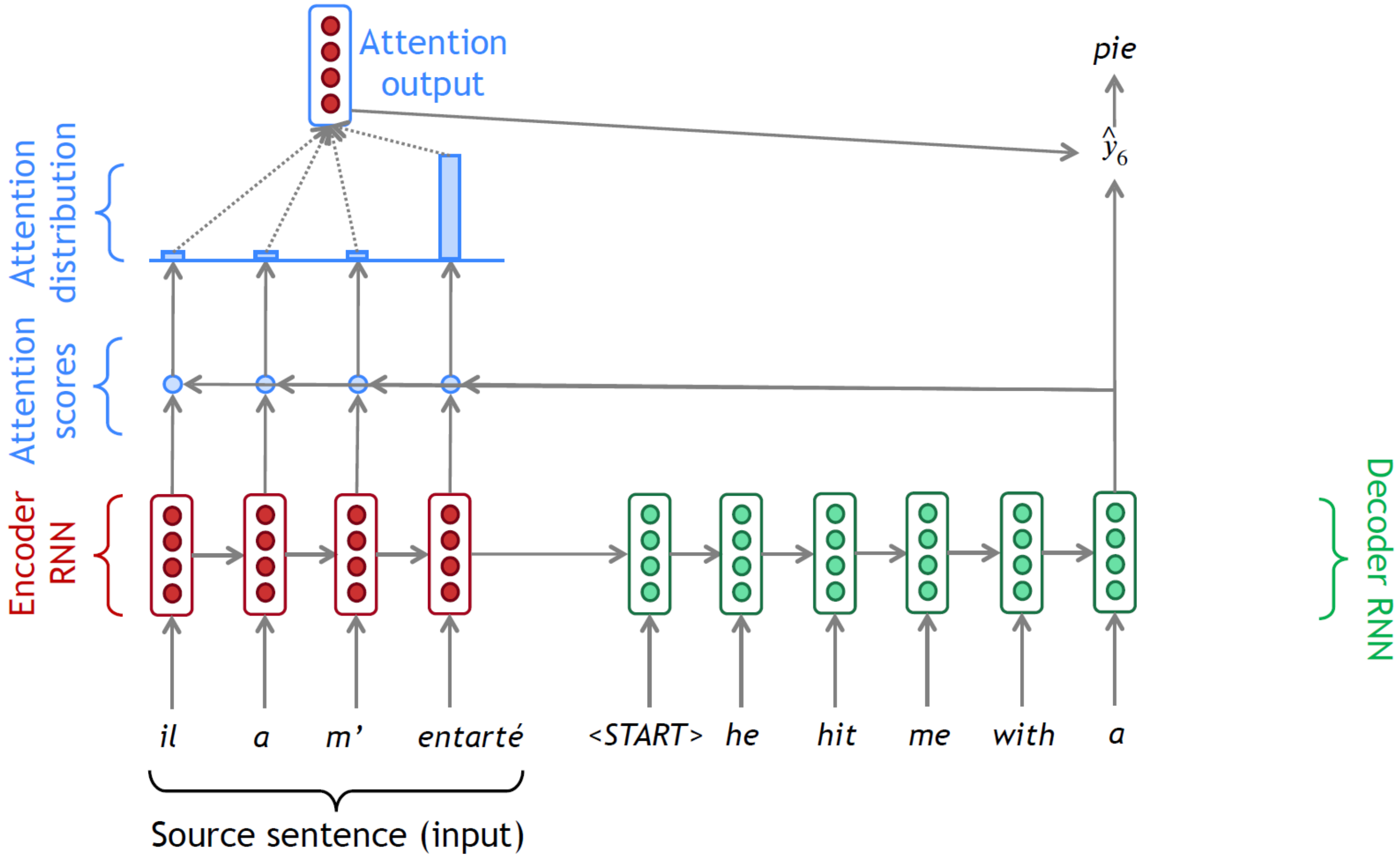
Sequence-to-sequence with attention



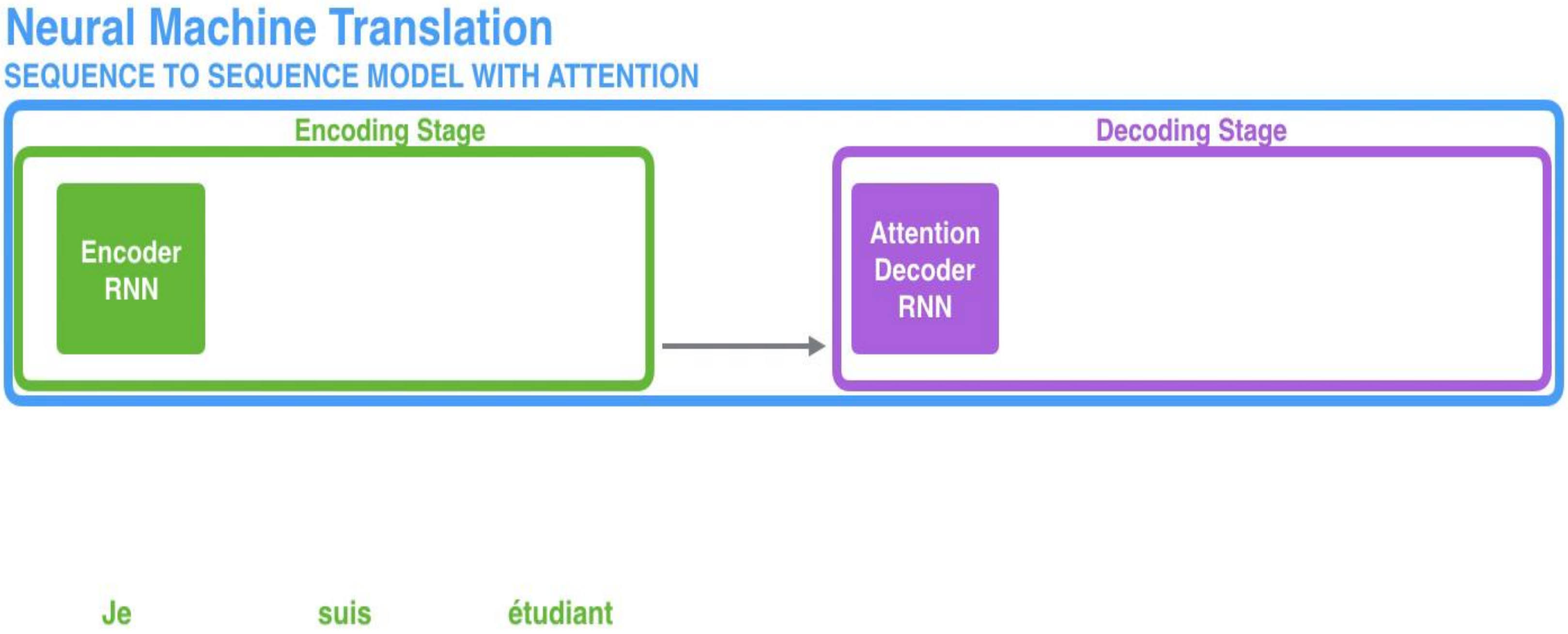
Sequence-to-sequence with attention



Sequence-to-sequence with attention



NMT with attention



Attention: in equations

- We have encoder hidden states $h_1, \ldots, h_N \in \mathbb{R}^h$
- On timestep *t*, we have decoder hidden state $s_t \in \mathbb{R}^h$
- We get the attention scores for this step:

$$\boldsymbol{e}^t = [\boldsymbol{s}_t^T \boldsymbol{h}_1, \ldots, \boldsymbol{s}_t^T \boldsymbol{h}_N] \in \mathbb{R}^N$$

- We take softmax to get the attention distribution for this step (this is a probability distribution and sums to 1)

$$\alpha^t = \mathrm{softmax}(\boldsymbol{e}^t) \in \mathbb{R}^N$$

- We use to take a weighted sum of the encoder hidden states to get the attention output

$$\boldsymbol{a}_t = \sum_{i=1}^{N} \alpha_i^t \boldsymbol{h}_i \in \mathbb{R}^h$$

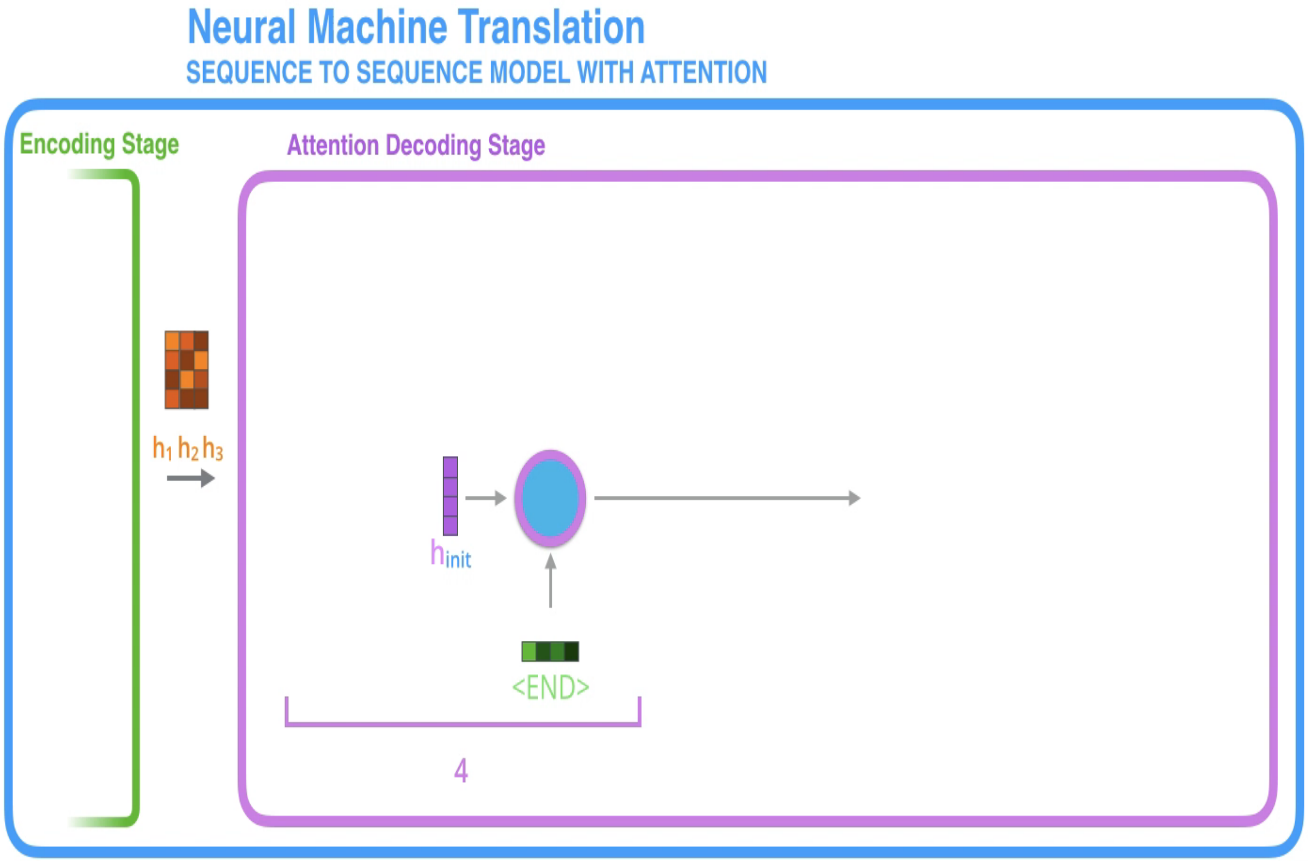
- Finally we concatenate the attention output with the decoder hidden state and proceed as in the non-attention seq2seq model

$$[\boldsymbol{a}_t; \boldsymbol{s}_t] \in \mathbb{R}^{2h}$$

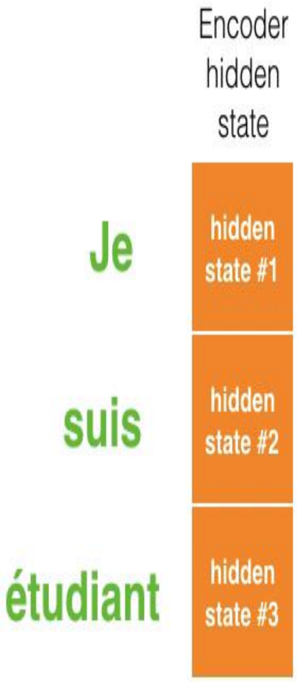
Illustration of attention

Attention at time step 4

Decoder with attention

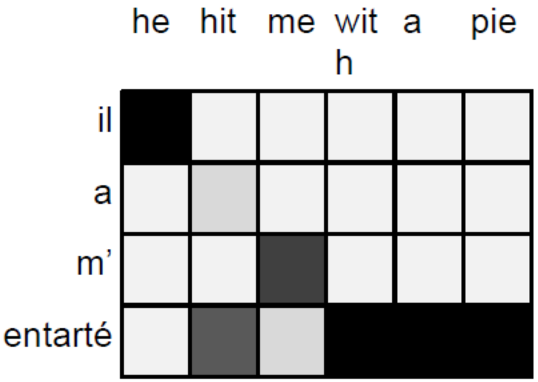


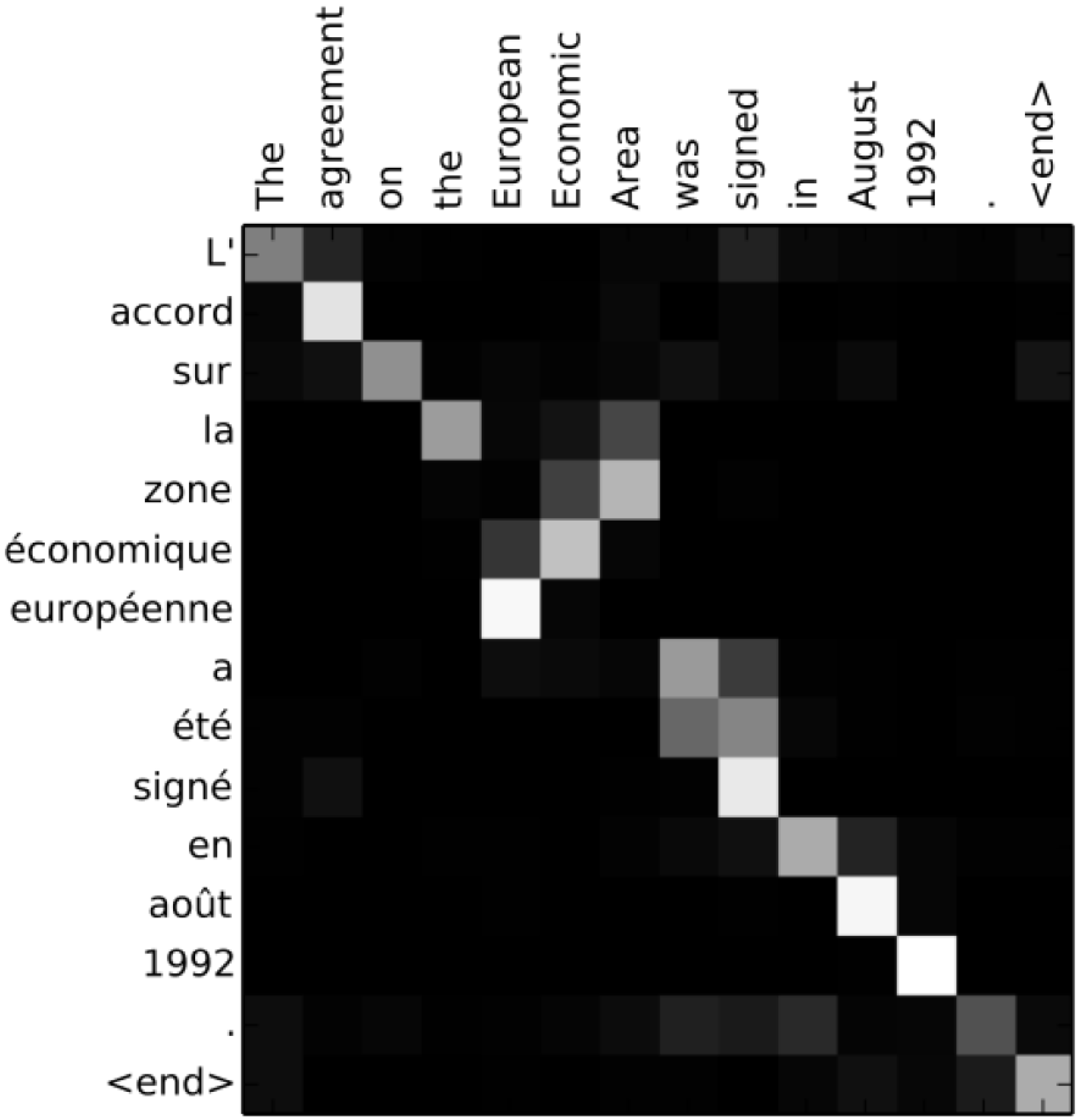
Attention produces alignments



Advantages of attention

- Attention significantly improves NMT performance
  - It’s very useful to allow decoder to focus on certain parts of the source
- Attention solves the bottleneck problem
  - Attention allows decoder to look directly at source; bypass bottleneck
- Attention helps with vanishing gradient problem
  - Provides shortcut to faraway states
- Attention provides some interpretability
  - By inspecting attention distribution, we can see what the decoder was focusing on
  - We get (soft) alignment for free!
  - This is great because we never explicitly trained an alignment system
  - The network just learned alignment by itself




The
agreement
on
the
European
Economic
Area
was
signed
in
August
1992
.
<end>
L'
accord
sur
la
zone
économique
européenne
a
été
signé
en
août
1992
.
<end>

Attention and unknown words

- using the attention, we know alignment of words
- unknown words on the output <ukn> can be translated from the dictionary, e.g., max $p_{dict}(e|f)$ or copied from the input to the output

Attention is a general deep learning technique

- We've seen that attention is a great way to improve the sequence-to-sequence model for Machine Translation.
- However: You can use attention in many architectures (not just seq2seq) and many tasks (not just MT)
- More general definition of attention:
  - Given a set of vector *values*, and a vector *query*, attention is a technique to compute a weighted sum of the values, dependent on the query.
- We sometimes say that the query *attends to* the values.
- For example, in the seq2seq + attention model, each decoder hidden state (query) *attends to* all the encoder hidden states (values).
- Intuition:
  - The weighted sum is a *selective summary* of the information contained in the values, where the query determines which values to focus on.
  - Attention is a way to obtain a *fixed-size representation of an arbitrary set of representations* (the values), dependent on some other representation (the query).

Advantages of NMT

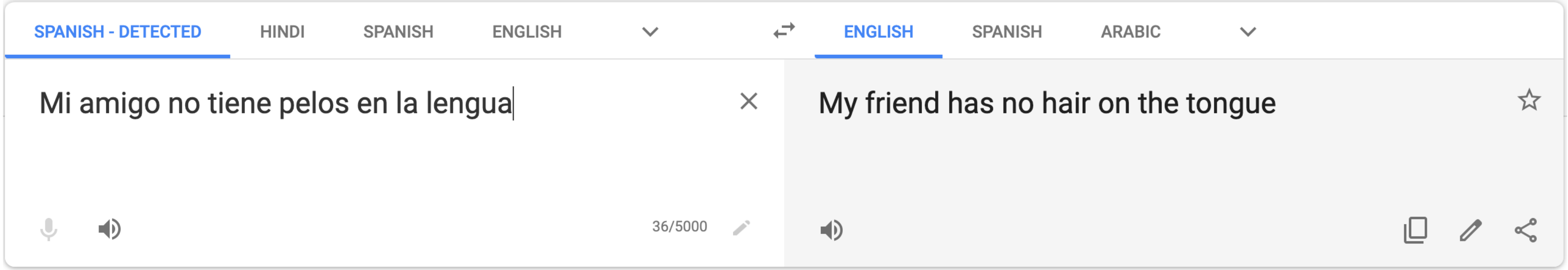
- Compared to SMT, NMT has many advantages:
  - Better performance
  - More fluent
  - Better use of context
  - Better use of phrase similarities
- A single neural network to be optimized end-to-end
  - No subcomponents to be individually optimized
- Requires much less human engineering effort
  - No feature engineering
  - Same method for all language pairs

Disadvantages of NMT?

- Compared to SMT:
- NMT is less interpretable
  - Hard to debug
- NMT is difficult to control
  - For example, can’t easily specify rules or guidelines for translation
  - Safety concerns!

So is Machine Translation solved?

- Many difficulties remain:
- Out-of-vocabulary words
- Domain mismatch between train and test data
- Maintaining context over longer text
- Low-resource language pairs
- Using common sense is still hard
- Idioms are difficult to translate

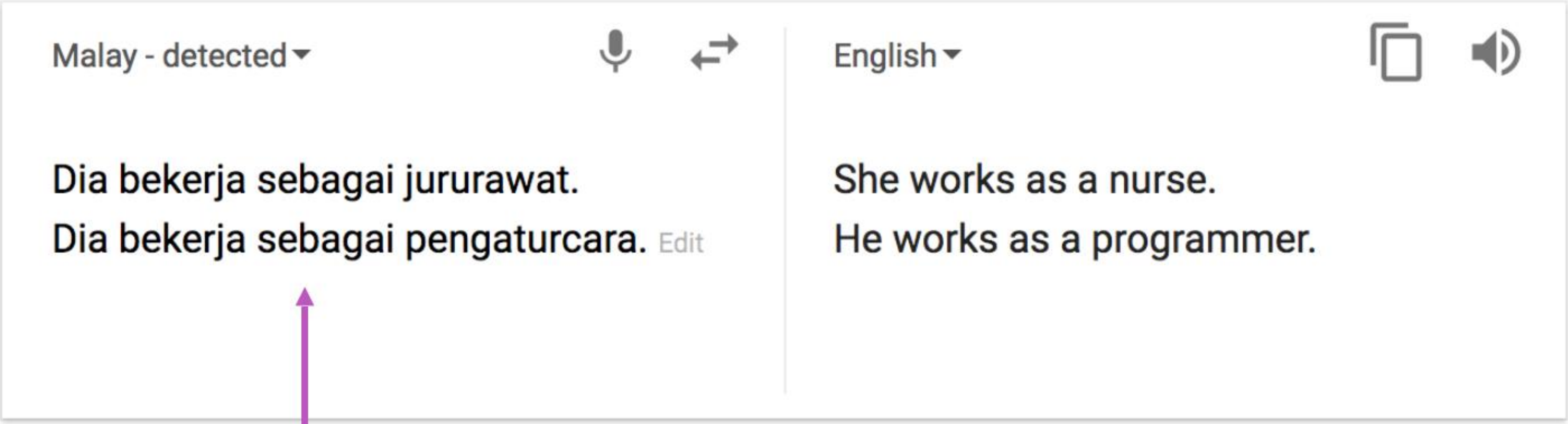


Mi amigo no tiene pelos en la lengua

My friend has no hair on the tongue

So is Machine Translation solved?

- NMT picks up biases in training data



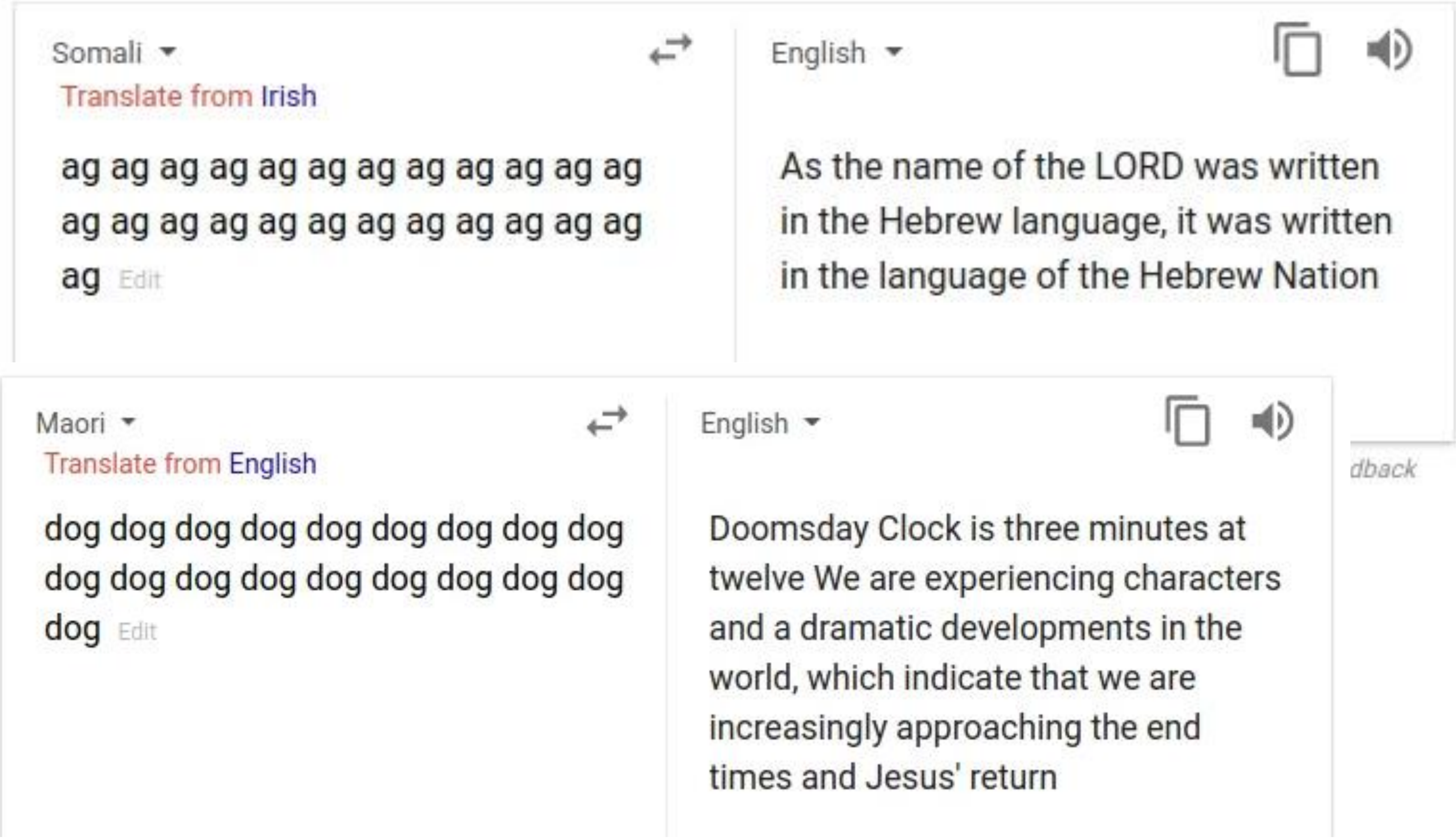
Dia bekerja sebagai jururawat.
Dia bekerja sebagai pengaturcara.

She works as a nurse.
He works as a programmer.

Didn’t specify gender

So is Machine Translation solved?

- Uninterpretable systems do strange things



ag ag ag ag ag ag ag ag ag ag ag ag ag ag ag ag ag ag ag ag ag ag ag ag ag

As the name of the LORD was written in the Hebrew language, it was written in the language of the Hebrew Nation

dog dog dog dog dog dog dog dog dog dog dog dog dog dog dog dog dog dog dog

Doomsday Clock is three minutes at twelve We are experiencing characters and a dramatic developments in the world, which indicate that we are increasingly approaching the end times and Jesus' return

Picture source: https://www.vice.com/en_uk/article/j5npeg/why-is-google-translate-spitting-out-sinister-religious-prophecies

Explanation: https://www.skynettoday.com/briefs/google-nmt-prophecies

Evaluating MT: Using human evaluators

- **Fluency**: How intelligible, clear, readable, or natural in the target language is the translation?
- **Fidelity**: Does the translation have the same meaning as the source?
  - **Adequacy**: Does the translation convey the same information as source?
    - Bilingual judges given source and target language, assign a score
      - Monolingual judges given reference translation and MT result.
  - **Informativeness**: Does the translation convey enough information as the source to perform a task?
    - What % of questions can monolingual judges answer correctly about the source sentence given only the translation.

Automatic Evaluation of MT

George A. Miller and J. G. Beebe-Center. 1958. Some Psychological Methods for Evaluating the Quality of Translations. Mechanical Translation 3:73-80.

- Human evaluation is expensive and very slow
- Need an evaluation metric that takes seconds, not months
- Intuition: MT is good if it looks like a human translation

1. Collect one or more human ***reference translations*** of the source.
2. Score MT output based on its similarity to the reference translations.
   - BLEU
   - NIST
   - TER
   - METEOR

Human evaluation

INPUT: Ich bin müde. **(INPUT: Je suis fatigué.)**

	Fidelity	Fluency
Tired is I.	5	2
Cookies taste good!	1	5
I am tired.	5	5

WER measure

- Word Error Rate (WER): Levenhstein distance to the reference translation (insert, delete, substitute)
- good for fluency
- not so well for fidelity
- inflexible
- Hypothesis 1 = „he saw a man and a woman“

  Reference = „he saw a woman and a man“

  WER does not take into account „woman“ or „man“ !

PER measure

- Position-Independent Word Error Rate (PER)
- PER: matching on the level of unigrams
- not good for fluency
- too flexible for fidelity

Hypothesis 1 = „he saw a man“

Hypothesis 2 = „a man saw he“

Reference = „he saw a man“

Both hypotheses have the same value of PER!

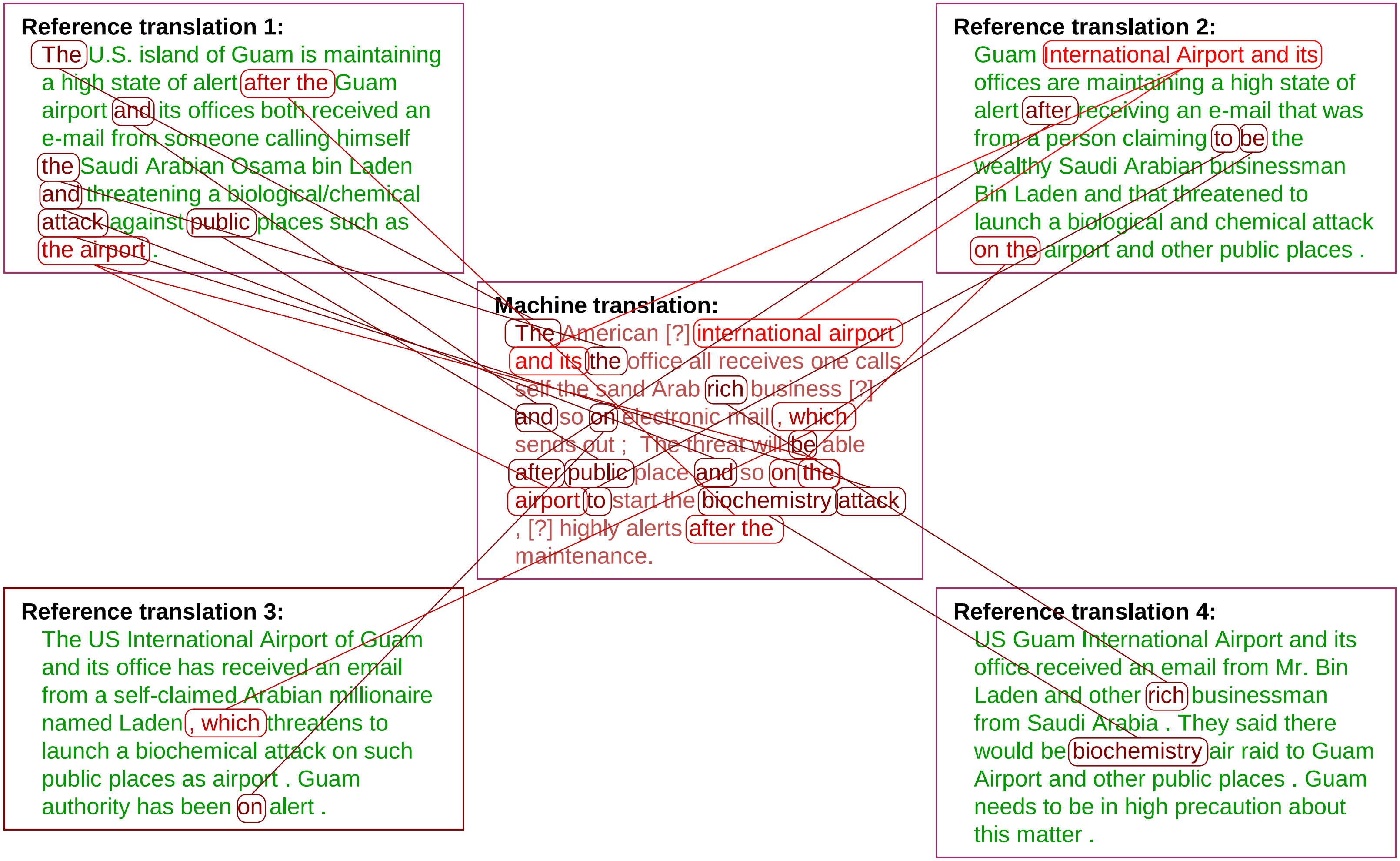
BLEU (Bilingual Evaluation Understudy)

Kishore Papineni, Salim Roukos, Todd Ward and Wei-Jing Zhu. 2002. BLEU: A
method for automatic evaluation of machine translation. Proceedings of ACL 2002.

- “n-gram precision”
- Ratio of correct n-grams to the total number of output n-grams
  - Correct: Number of *n*-grams (unigram, bigram, etc.) the MT output shares with the reference translations.
  - Total: Number of *n*-grams in the MT result.
- The higher the precision, the better the translation
- Recall is ignored

Multiple Reference Translations

Slide from Bonnie Dorr



Reference translation 1:
The U.S. island of Guam is maintaining a high state of alert after the Guam airport and its offices both received an e-mail from someone calling himself the Saudi Arabian Osama bin Laden and threatening a biological/chemical attack against public places such as the airport .

Reference translation 2:
Guam International Airport and its offices are maintaining a high state of alert after receiving an e-mail that was from a person claiming to be the wealthy Saudi Arabian businessman Bin Laden and that threatened to launch a biological and chemical attack on the airport and other public places .

Machine translation:
The American [?] international airport and its the office all receives one calls self the sand Arab rich business [?] and so on electronic mail , which sends out ; The threat will be able after public place and so on the airport to start the biochemistry attack , [?] highly alerts after the maintenance.

Reference translation 3:
The US International Airport of Guam and its office has received an email from a self-claimed Arabian millionaire named Laden , which threatens to launch a biochemical attack on such public places as airport . Guam authority has been on alert .

Reference translation 4:
US Guam International Airport and its office received an email from Mr. Bin Laden and other rich businessman from Saudi Arabia . They said there would be biochemistry air raid to Guam Airport and other public places . Guam needs to be in high precaution about this matter .

Computing BLEU: Unigram precision

Slides from Ray Mooney

Cand 1: Mary no slap the witch green

Cand 2: Mary did not give a smack to a green witch.

Ref 1: Mary did not slap the green witch.

Ref 2: Mary did not smack the green witch.

Ref 3: Mary did not hit a green sorceress.

Candidate 1 Unigram Precision: 5/6

Computing BLEU: Bigram Precision

Cand 1: Mary no slap the witch green.
Cand 2: Mary did not give a smack to a green witch.

Ref 1: Mary did not slap the green witch.
Ref 2: Mary did not smack the green witch.
Ref 3: Mary did not hit a green sorceress.

Candidate 1 Bigram Precision: 1/5

Computing BLEU: Unigram Precision

Cand 1: Mary no slap the witch green.
Cand 2: Mary did not give a smack to a green witch.

Ref 1: Mary did not slap the green witch.
Ref 2: Mary did not smack the green witch.
Ref 3: Mary did not hit a green sorceress.

Clip the count of each *n*-gram
to the maximum count of the *n*-gram in any single reference

Candidate 2 Unigram Precision: 7/10

Computing BLEU: Bigram Precision

Cand 1: Mary no slap the witch green.
Cand 2: Mary did not give a smack to a green witch.

Ref 1: Mary did not slap the green witch.
Ref 2: Mary did not smack the green witch.
Ref 3: Mary did not hit a green sorceress.

Candidate 2 Bigram Precision: 4/9

Brevity Penalty

- BLEU is precision-based: no penalty for dropping words
- Instead, we use a brevity penalty for translations that are shorter than the reference translations.

$$\text{brevity-penalty} = \min\left(1\,,\ \frac{\text{output-length}}{\text{reference-length}}\right)$$

Computing BLEU

- Precision$_1$, precision$_2$, etc., are computed over all candidate sentences C in the test set

$$\text{precision}_n = \frac{\sum_{C \in \textit{corpus}} \sum_{\text{n-gram} \in C} \text{count-in-reference}_{\text{clip}}(\text{n-gram})}{\sum_{C \in \textit{corpus}} \sum_{\text{n-gram} \in C} \text{count}(\text{n-gram})}$$

$$\text{BLEU-4} = \min\left(1, \frac{\text{output-length}}{\text{reference-length}}\right) \prod_{i=1}^{4} \text{precision}_i$$

BLEU-2:

Candidate 1: Mary no slap the witch green.
Best Reference: Mary did not slap the green witch.

$$\frac{6}{7} \times \frac{5}{6} \times \frac{1}{5} = .14$$

Candidate 2: Mary did not give a smack to a green witch.
Best Reference: Mary did not smack the green witch.

$$\frac{7}{10} \times \frac{4}{9} = .31$$

Properties of BLEU

- BLEU works well in comparing similar MT systems , e.g., competing variants or using different parameters
- not so good in comparison of different systems

- no good measure exists on the level of sentence
- no good measure exists of an absolute translation quality

Improvements in MT

- large corpora
- adaptations to specific domains, e.g., IT, pharmacy, automotive industry
- terminological dictionaries, terminology lists, translation memories

A few results of NMT systems

- Adamič: Louis Adamič translation of Ivan Cankar’s “A cup of coffee” to English (this is the source text for MT)
- Nematus: open source NMT (EdinburghNLP)
- translations provided by Matjaž Rihtar and Simon Krek
- GT: Google Translate

A cup of coffee - Skodelica kave – originals

- ADAMIČ: I have often been unjust, unfair to people whom I loved. Such injustice is an unpardonable sin, permanent, enduring, unforgettable in one's conscience. Sometimes the sin is forgotten, eroded from your life, drowned in the eventfulness of the days; but suddenly, perhaps in the middle of a beautiful enjoyable day, perhaps at night, it comes back upon you, to weigh down your soul, to pain and burn your conscience as though you have just committed it.

- CANKAR: Velikokrat v svojem življenju sem storil krivico človeku, ki sem ga ljubil. Taka krivica je kakor greh zoper svetega duha: ne na tem ne na onem svetu ni odpuščena. Neizbrisljiva je, nepozabljiva. Časih počiva dolga leta, kakor da je bila ugasnila v srcu, izgubila se, utopila v nemirnem življenju. Nenadoma, sredi vesele ure, ali ponoči, ko se prestrašen vzdramiš iz hudih sanj, pade v dušo težak spomin, zaboli in zapeče s toliko silo, kakor da je bil greh šele v tistem trenutku storjen.

A cup of coffee: ANG → SLO

- NEMATUS: Pogosto sem bil nepravičen, nepošten do ljudi, ki sem jih imel rad. Takšna krivica je neodpustljiv greh, trajen, trajen, nepozaben, v vesti. Včasih se greh pozabi, ovrže iz tvojega življenja, utopi se v vseh dnevih, ampak na lepem, morda sredi lepega lepega dne, se ti vrne, da bi lahko tehtal svojo dušo, bolečino in sežgal svojo vest, kot da si jo pravkar zagrešil.

- GT: Pogosto sem bil krivičen, nepošten do ljudi, ki sem jih ljubil. Takšna krivica je nepreklicni greh, trajna, trajna, nepozabna v svoji vesti. Včasih je pozabljen greh, erodiran iz tvojega življenja, utopil v dogodnost dni; ampak nenadoma, morda sredi čudovitega prijetnega dne, morda ponoči, se vrne na vas, da tehta dušo, bolečino in vžge svojo vest, kot da ste jo pravkar storili.

A cup of coffee: SLO → ANG

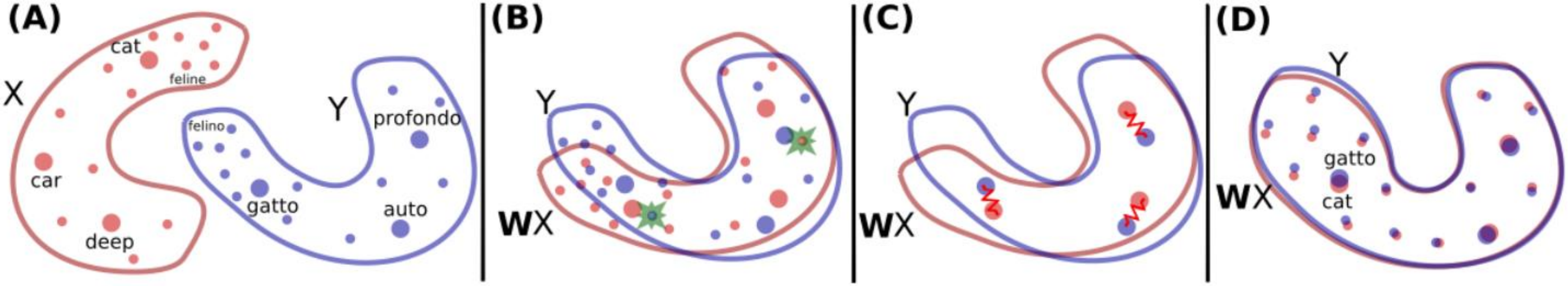
- CANKAR: Velikokrat v svojem življenju sem storil krivico človeku, ki sem ga ljubil. Taka krivica je kakor greh zoper svetega duha: ne na tem ne na onem svetu ni odpuščena. Neizbrisljiva je, nepozabljiva. Časih počiva dolga leta, kakor da je bila ugasnila v srcu, izgubila se, utopila v nemirnem življenju. Nenadoma, sredi vesele ure, ali ponoči, ko se prestrašen vzdramiš iz hudih sanj, pade v dušo težak spomin, zaboli in zapeče s toliko silo, kakor da je bil greh šele v tistem trenutku storjen.
- NEMATUS: Many times in my life, I have wronged the man I loved. Such an injustice is like a sin against the Holy Spirit: Not on this earth is fired. She's indelible, unforgettable. Times have been resting for many years, as if she was lost in her heart, lost, drowned in a restless life. Suddenly, in the middle of a happy hour, or at night, when you're scared of a terrible dream, it falls into the soul of a difficult memory, pain and sting with as much force as it was committed in that moment.
- GT: Many times in my life, I did injustice to the man I loved. Such guilt is like a sin against the Holy Ghost: it is not forsaken in this world. It is indelible, unforgettable. It rests for many years, as if it had been turned off in the heart, lost, drowned in a troubled life. Suddenly, in the middle of a cheerful hour, or at night, when frightened by the whimpering of bad dreams, a heavy memory falls into the soul, gets sick and burns with so much force as if the sin was only done at that moment.

Are translators an endangered profession?

- Will translators soon be just quality controllers of MT systems and only fix minor details?
- Douglas Hofstadter: The Shallowness of Google Translate. The Atlantic, Jan 30, 2018
- Conclusion: Translation requires understanding the text, not only syntactic manipulation.
- But: many different purposes of translation, using modern tools.

Unsupervised translation from word embeddings

- alignment of two languages for low-resource languages



- Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, Hervé Jégou (2017): Word Translation Without Parallel Data. arXiv:1710.04087

Nematus

- Attention-based encoder-decoder model for neural machine translation built in Tensorflow.
- support for RNN and Transformer architectures
- arbitrary input features (factored neural machine translation)
- multi-GPU support
- batch decoding
- n-best output
- https://github.com/EdinburghNLP/nematus

OpenNMT

- good open source choice is also OpenNMT

http://opennmt.net

- implementations in lua (luaTorch), python (pyTorch), TensorFlow
- Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, Alexander M. Rush (2017): OpenNMT: Open-Source Toolkit for Neural Machine Translation. ArXiv:1701.02810