

Machine translation



Prof Dr Marko Robnik-Šikonja

Natural Language Processing, Edition 2023

Contents

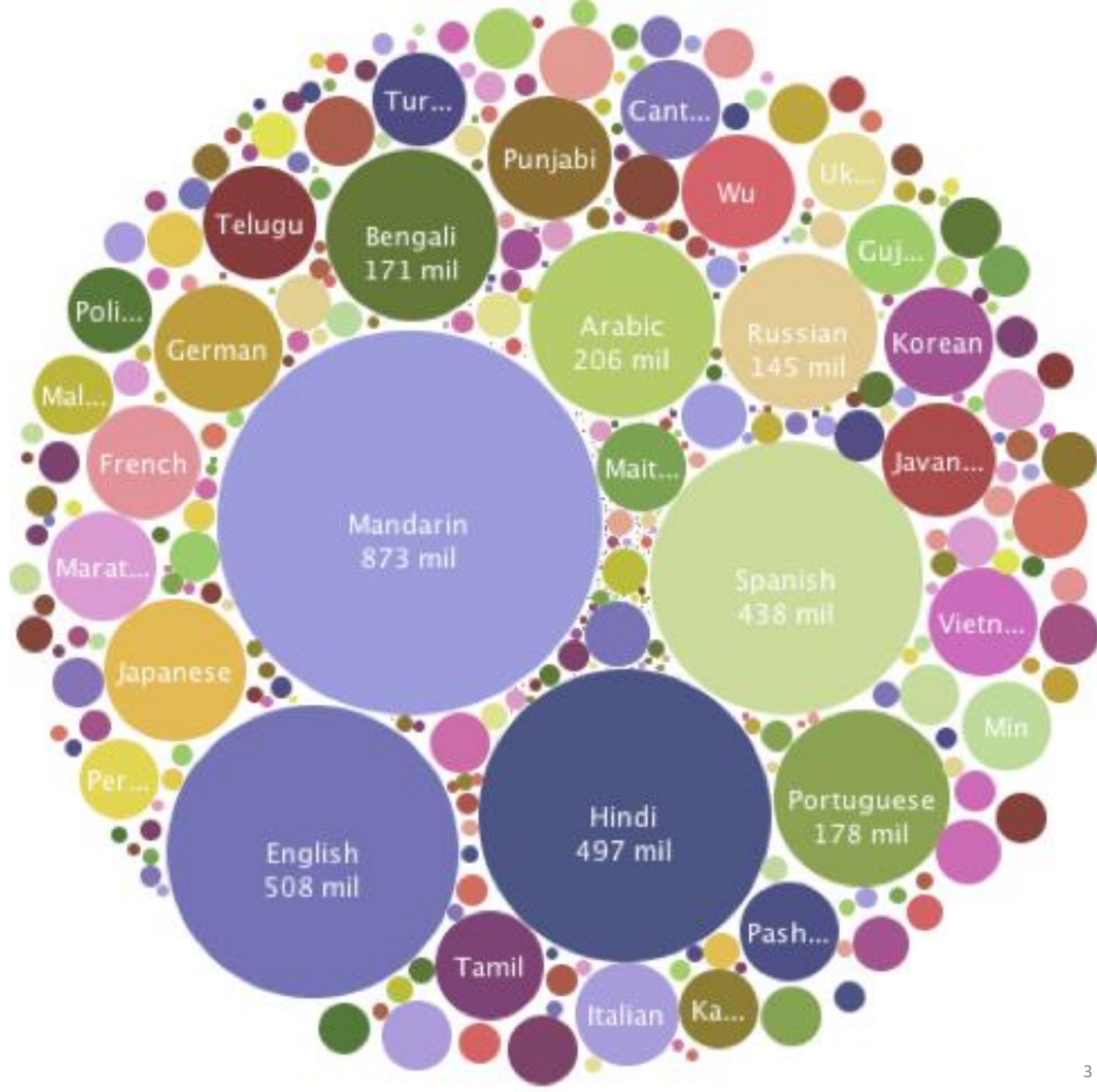
- statistical machine translation
- neural machine translation using sequence to sequence approach

Literature:

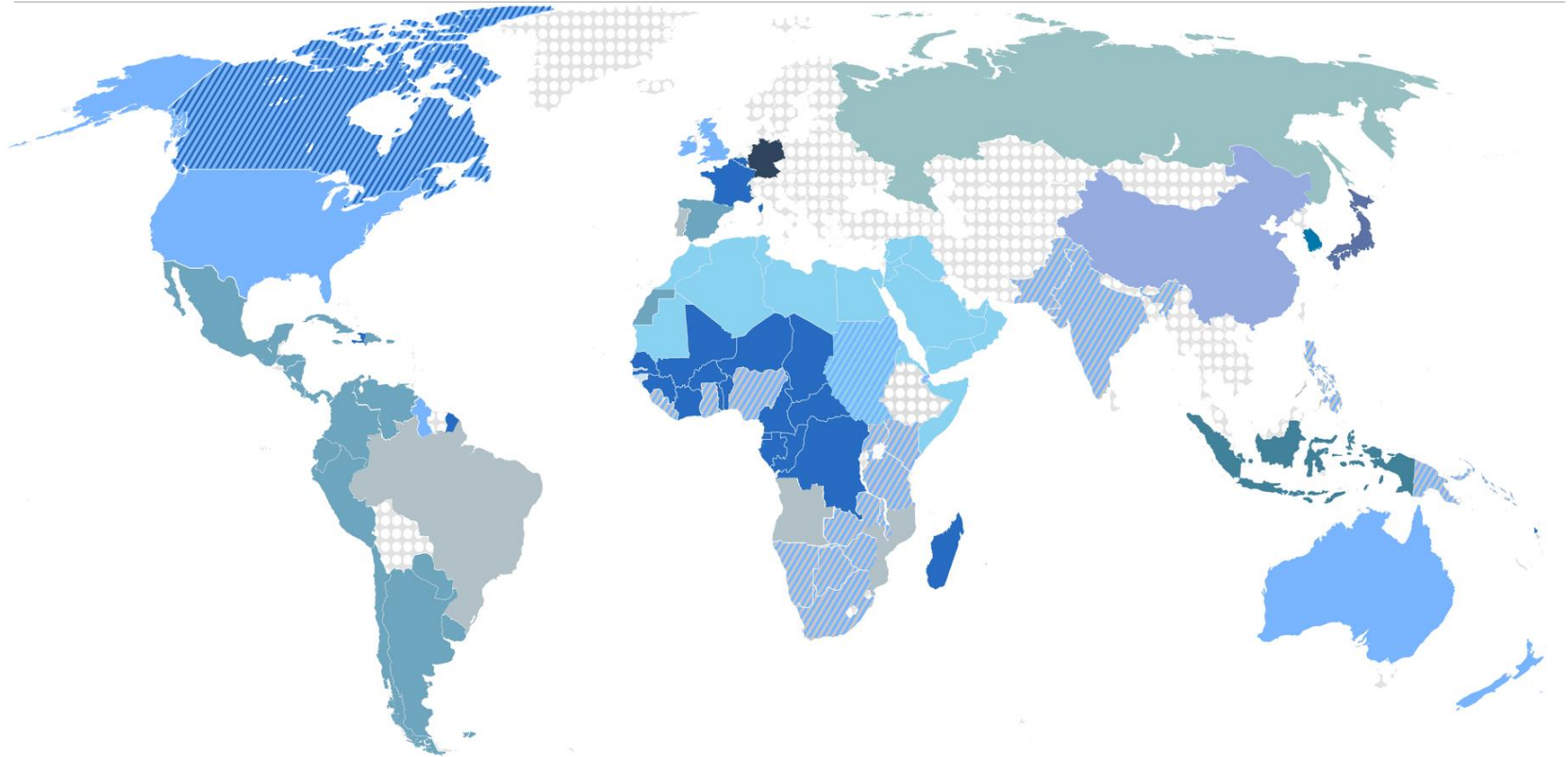
- Dan Jurafsky and James H. Martin. Speech and Language Processing (3rd ed. draft)
- Stanford course CS224n: Natural Language Processing with Deep Learning <https://web.stanford.edu/class/cs224n/>

World languages

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



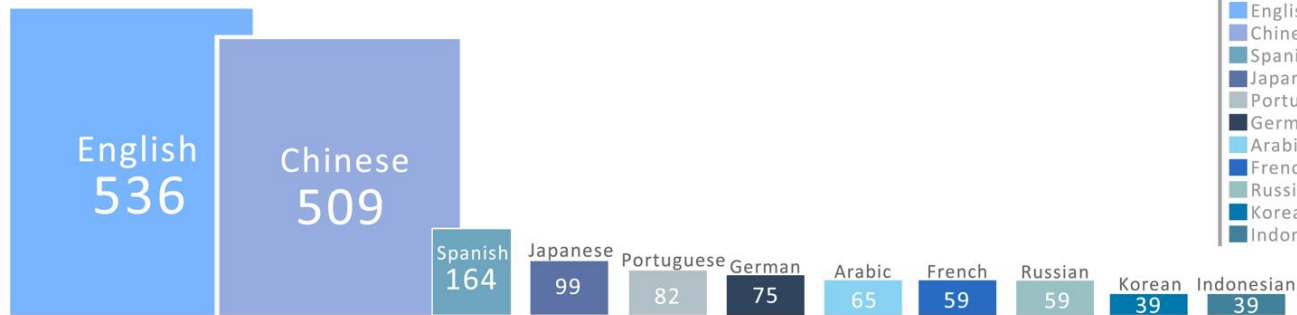
Top Languages on the Internet



■ 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%

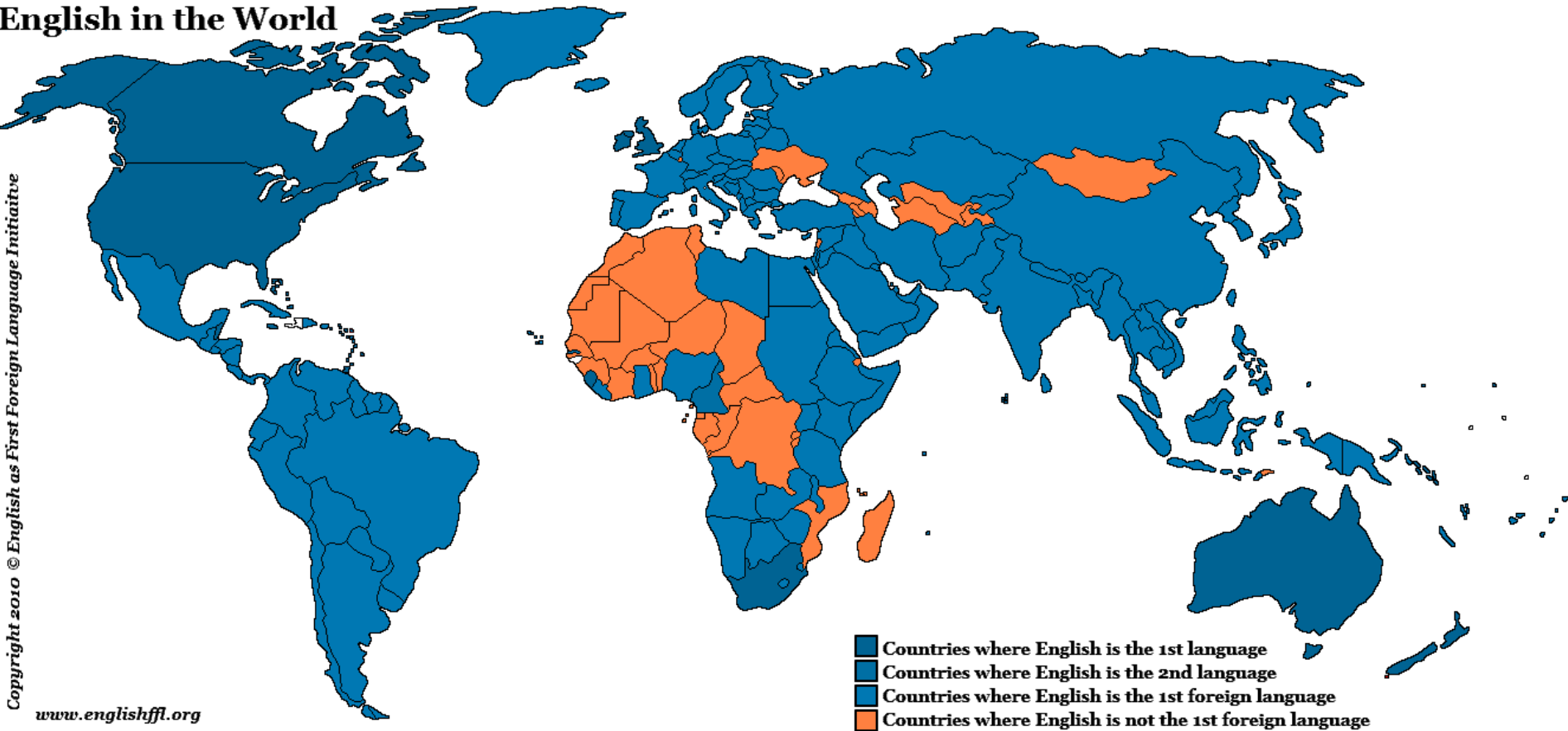
World population by Language (mln)

- 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

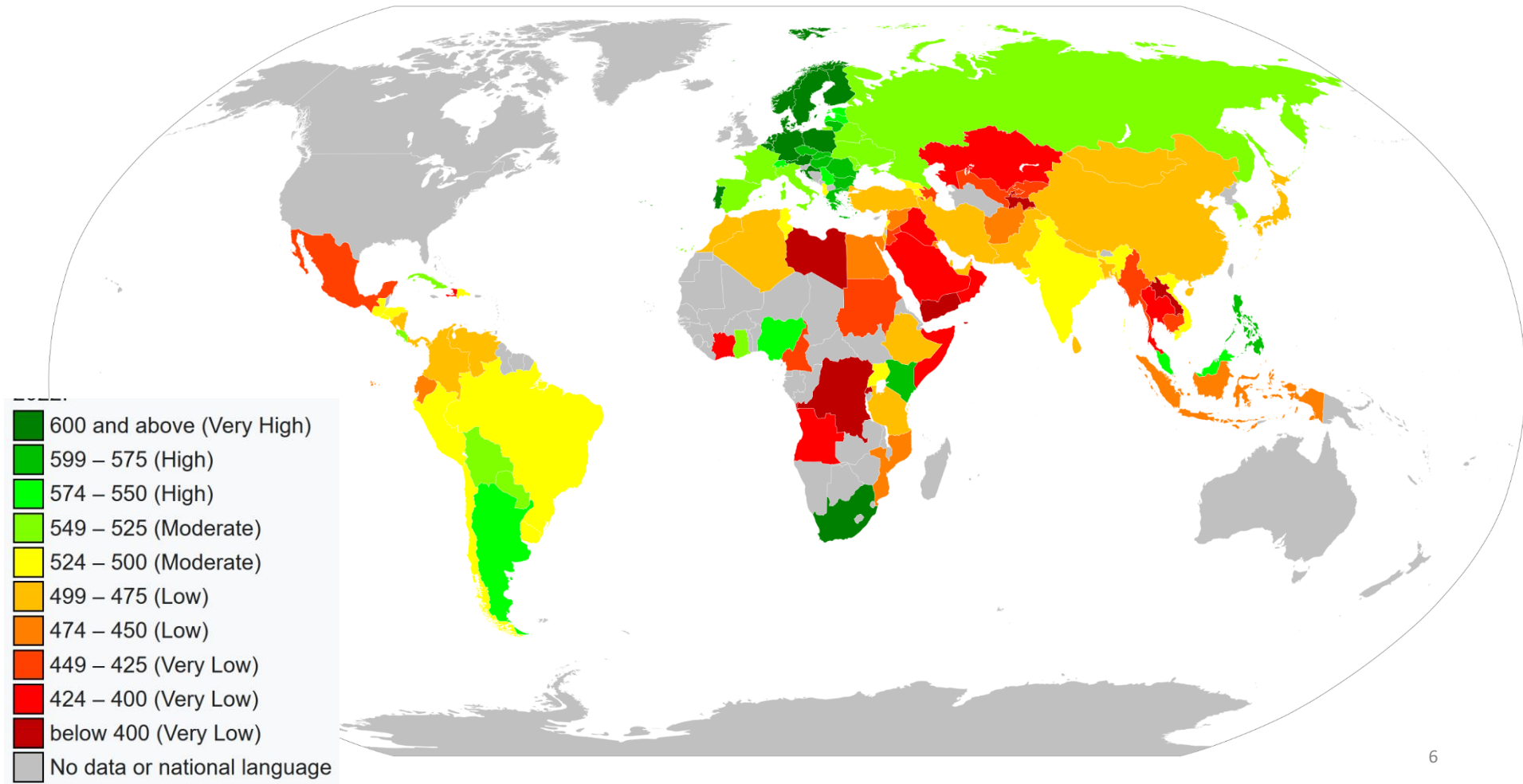
English as lingua franca?

English in the World



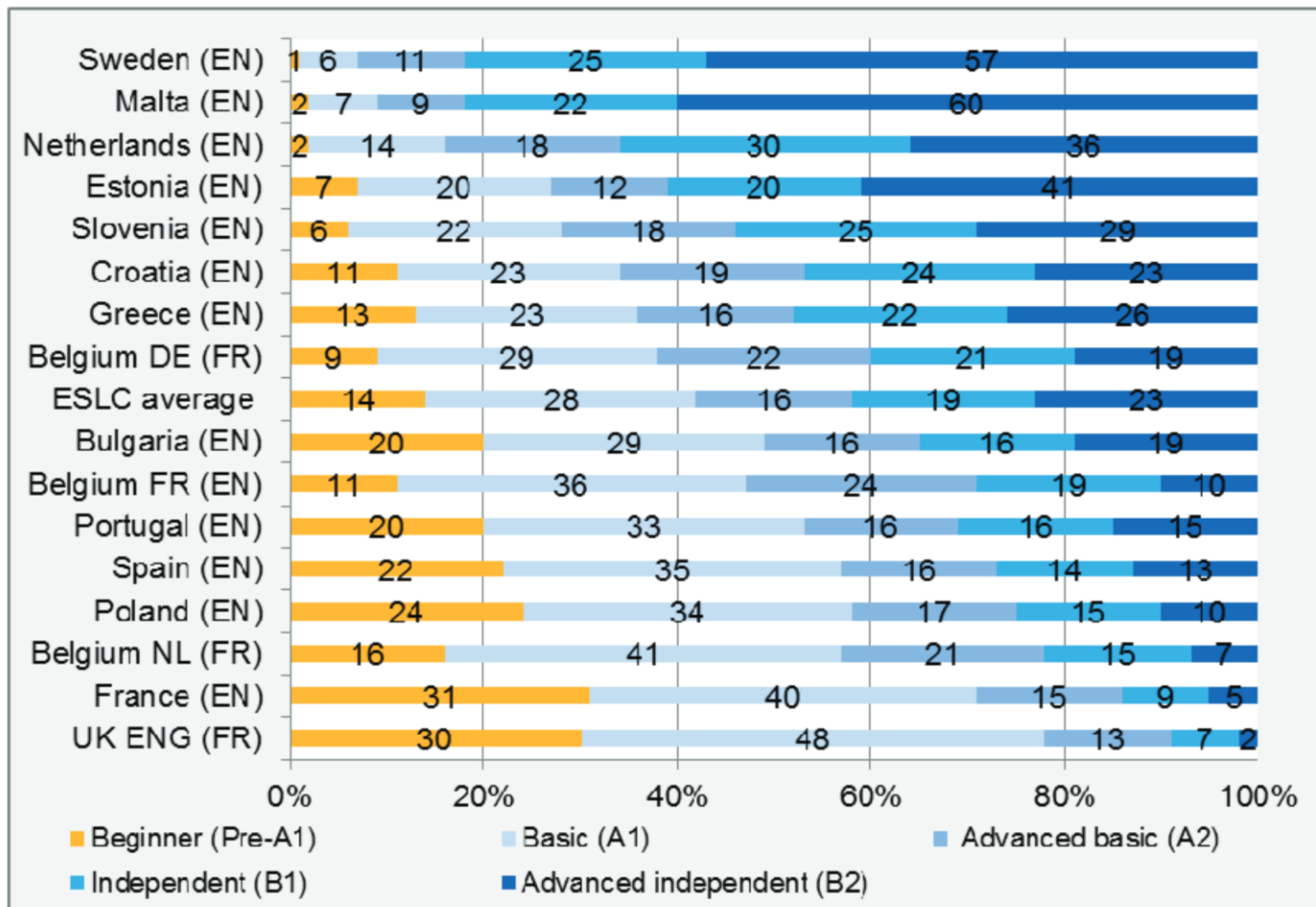
Global English proficiency index

English proficiency in the world in 2022, 2.1 million self-selected respondents



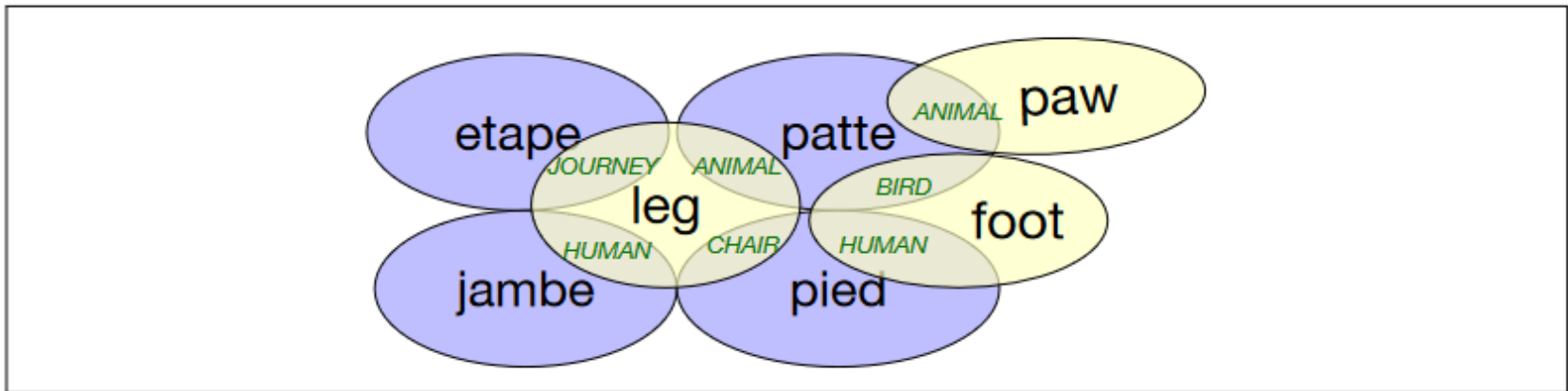
Language proficiency

- EU survey among pupils aged around 15, altogether 54,000 reponents



Lexical divergency

- Different languages have different definition of certain concepts



- The complex overlap between English leg, foot, etc., and various French translations as discussed by Hutchins and Somers (1992)

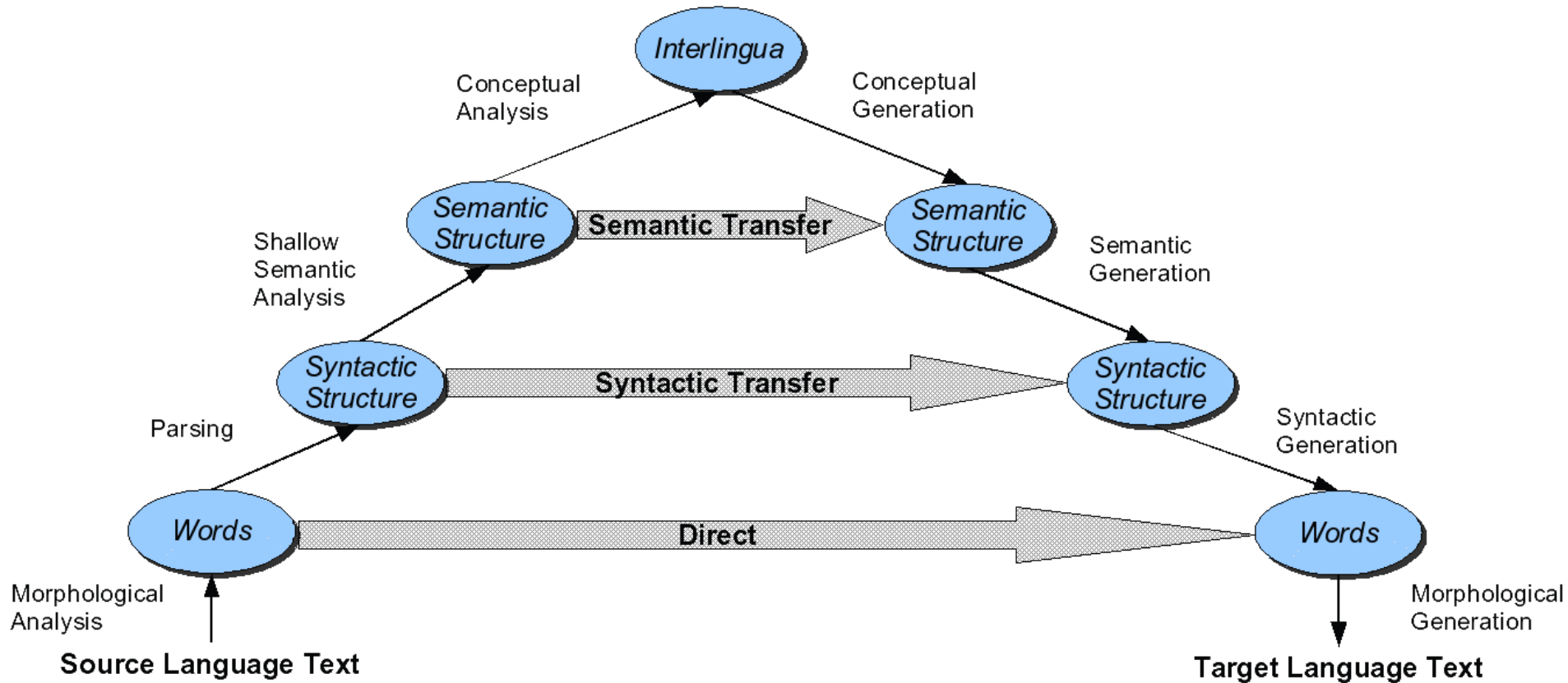
Statistical machine translation (SMT)

- 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"

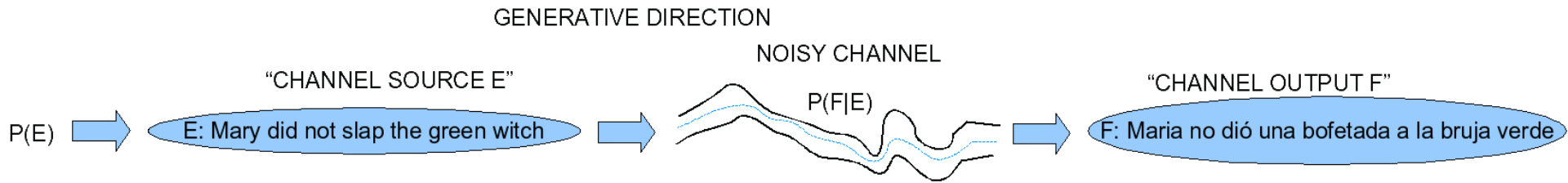
Given a French (foreign) sentence F , find an English sentence

$$\begin{aligned}\hat{E} &= \operatorname{argmax}_{E \hat{=} \text{English}} P(E | F) \\ &= \operatorname{argmax}_{E \hat{=} \text{English}} \frac{P(F | E)P(E)}{P(F)} \\ &= \operatorname{argmax}_{E \hat{=} \text{English}} \underbrace{P(F | E)}_{\text{Translation Model}} \underbrace{P(E)}_{\text{Language Model}}\end{aligned}$$

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



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)

$P(\text{me} | \text{hit}) > P(\text{to} | \text{crash})$

– 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

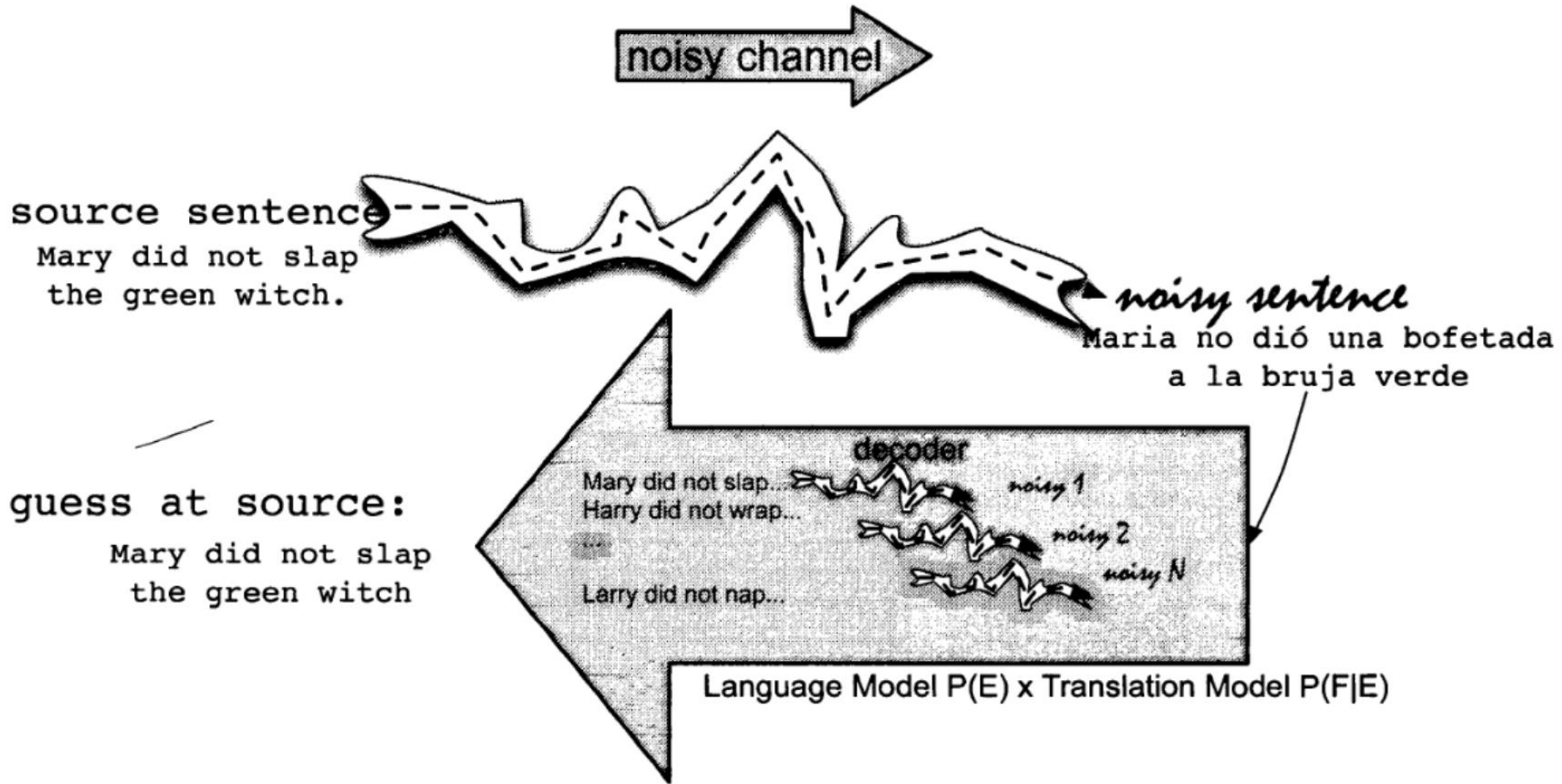
- $P(F|E)$'s job is to model “bag of words”; which words come from English to Spanish.
 - $P(F|E)$ doesn't have to worry about internal facts about English word order.
- $P(E)$'s job is to do bag generation: put the following words in order:
 - a ground there in the hobbit hole lived a in

Three Problems for Statistical MT

- **Language Model: given E , compute $P(E)$**
 - good English string \rightarrow high $P(E)$
 - random word sequence \rightarrow low $P(E)$
- **Translation Model: given (F,E) compute $P(F | E)$**
 - (F,E) look like translations \rightarrow high $P(F | E)$
 - (F,E) don't look like translations \rightarrow low $P(F | E)$
- **Decoding algorithm: given LM, TM, F , find \hat{E}**
 - Find translation E that maximizes $P(E) * P(F | E)$

Noisy channel model

- inference goes backwards



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

Sentence alignment

E1: "Good morning," said the little prince.	F1: -Bonjour, dit le petit prince.
E2: "Good morning," said the merchant.	F2: -Bonjour, dit le marchand de pilules perfectionnées qui apaisent la soif.
E3: This was a merchant who sold pills that had been perfected to quench thirst.	F3: On en avale une par semaine et l'on n'éprouve plus le besoin de boire.
E4: You just swallow one pill a week and you won't feel the need for anything to drink.	F4: -C'est une grosse économie de temps, dit le marchand.
E5: "They save a huge amount of time," said the merchant.	F5: Les experts ont fait des calculs.
E6: "Fifty-three minutes a week."	F6: On épargne cinquante-trois minutes par semaine.
E7: "If I had fifty-three minutes to spend?" said the little prince to himself.	F7: "Moi, se dit le petit prince, si j'avais cinquante-trois minutes à dépenser, je marcherais tout doucement vers une fontaine..."
E8: "I would take a stroll to a spring of fresh water"	

- Sentence alignment takes sentences E_1, \dots, E_n , and F_1, \dots, F_n and finds minimal sets of sentences that are translations of each other, including
 - single sentence mappings like (E_1, F_1) , (E_4, F_3) , (E_5, F_4) , (E_6, F_6)
 - many-to-one (2-1) alignments: $(E_2/E_3, F_2)$, $(E_7/E_8, F_7)$,
 - null alignments (F_5) .

Alignment procedure 1/2

- compute cost function that takes a span of source sentences and a span of target sentences and returns a score measuring how likely these spans are to be translations
- for that we use multilingual embedding space of both languages

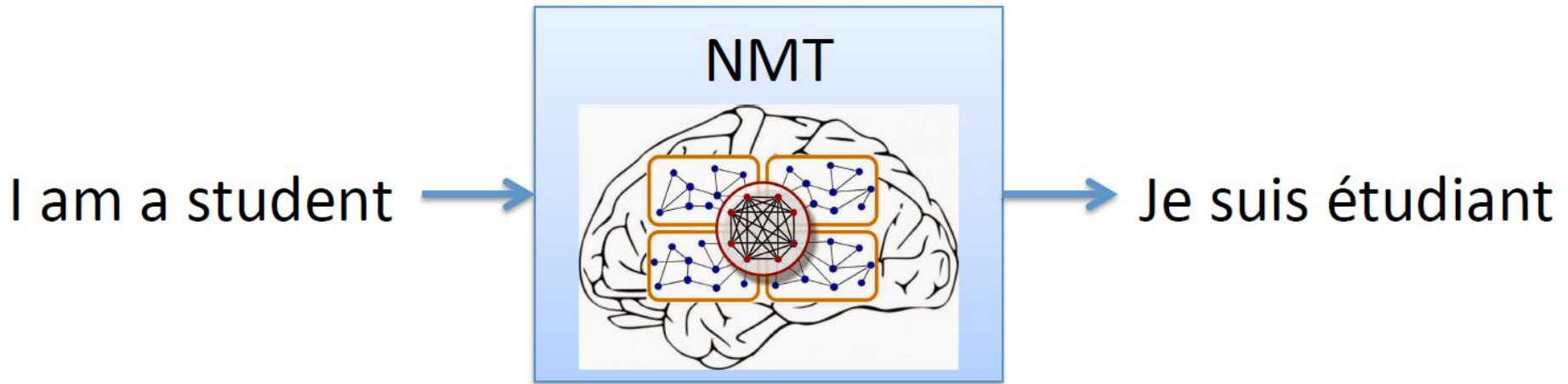
$$c(x, y) = \frac{(1 - \cos(x, y)) \text{nSents}(x) \text{nSents}(y)}{\sum_{s=1}^S 1 - \cos(x, y_s) + \sum_{s=1}^S 1 - \cos(x_s, y)}$$

- where $\text{nSents}()$ is the number of sentences (biases toward many alignments of single sentences instead of aligning very large spans).
- the denominator helps to normalize the similarities, so $x_1, \dots, x_S, y_1, \dots, y_S$ are randomly selected sentences sampled from the respective documents.

Alignment procedure 1/2

- an alignment algorithm that takes the alignment scores to find a good alignment between the documents
- Usually dynamic programming is used as the alignment algorithm, i.e. an extension of the minimum edit distance algorithm
- Finally, corpus cleanup:
 - remove noisy sentence pairs, e.g., too long or too short sentences,
 - too similar sentences (just copies instead of translations),
 - rank by the multilingual embedding cosine score and remove low-scoring pairs

Neural machine translation (NMT)



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

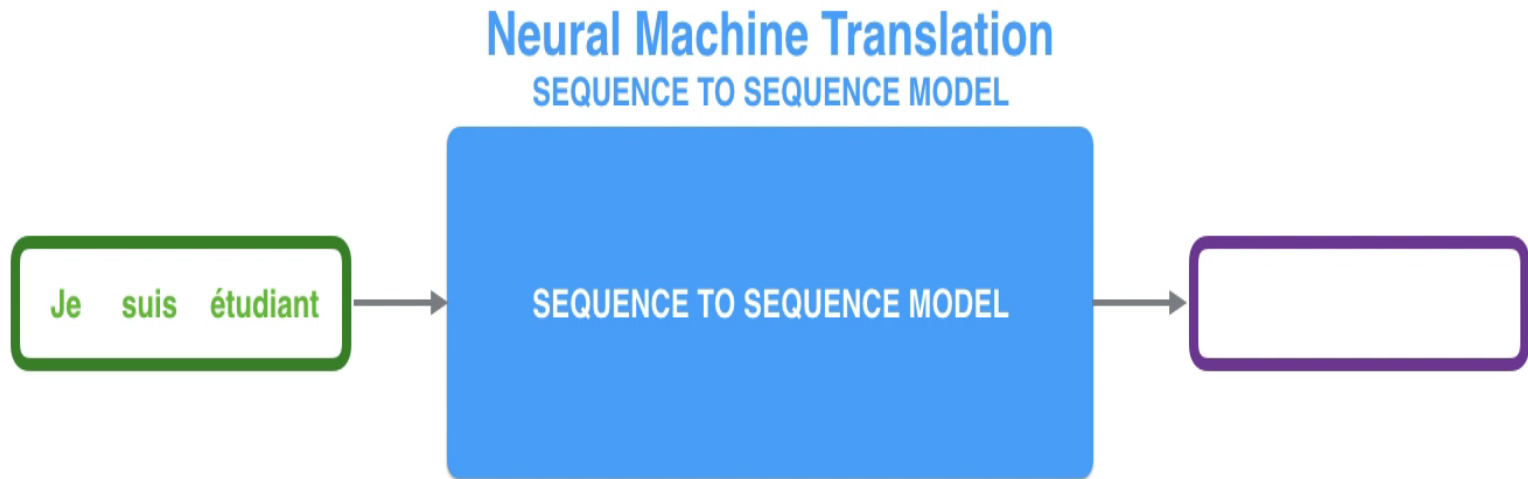
- direct translation based on sequences
- The neural network architecture is called sequence-to-sequence (aka seq2seq) and it involves *two* networks.

Seq2Seq model

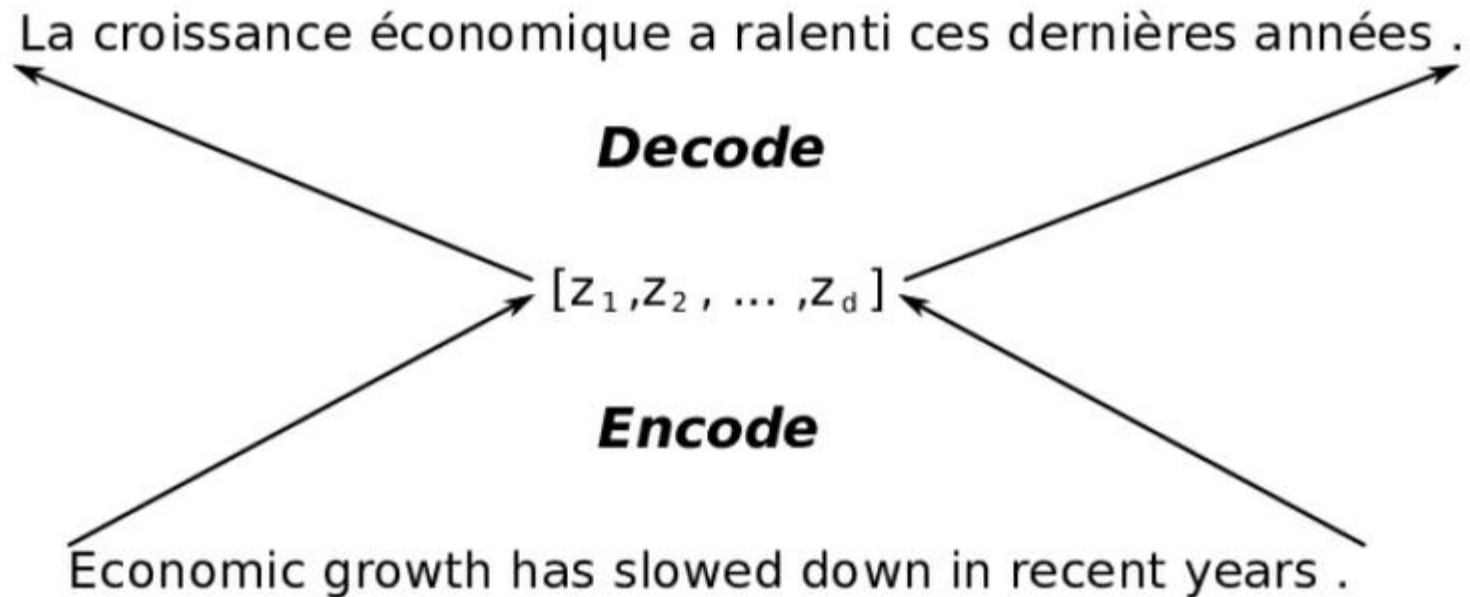


Videos by Jay Alammar: [Visualizing A Neural Machine Translation Model \(Mechanics of Seq2seq Models With Attention\)](#), 2018

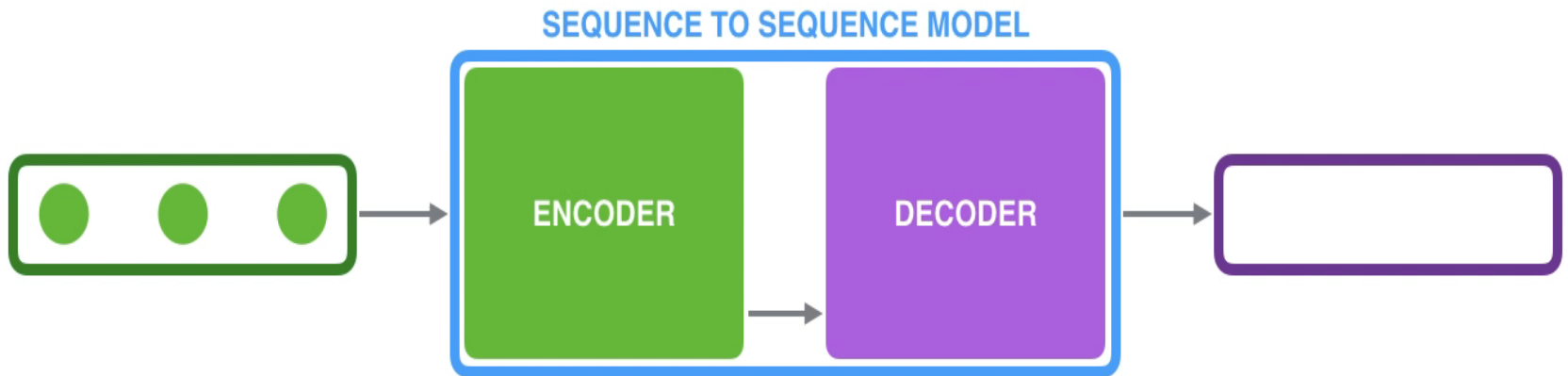
Seq2Seq for NMT



Encoder-Decoder Model

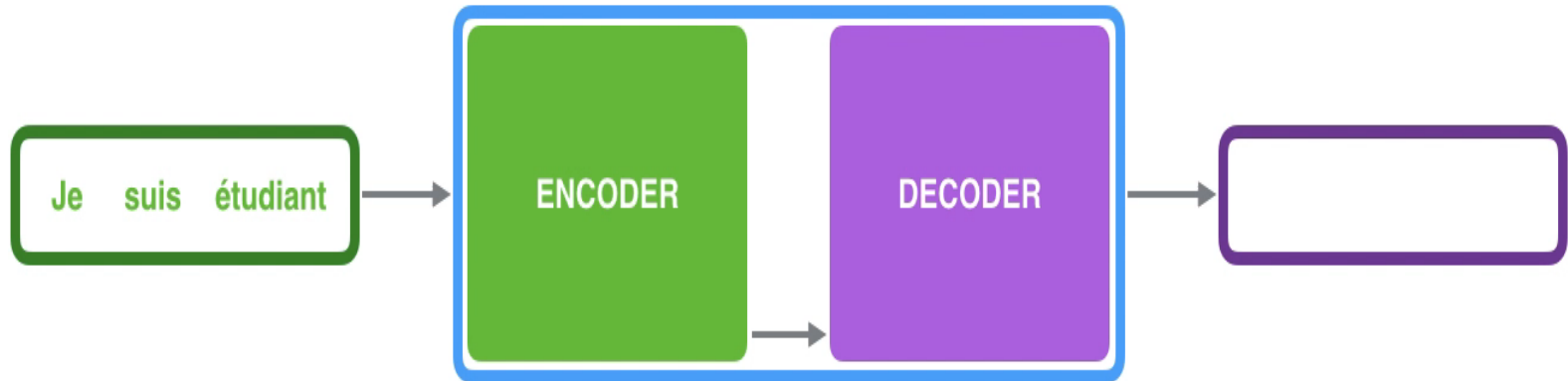


Encoder-decoder for sequences



Encoder-decoder for NMT

Neural Machine Translation SEQUENCE TO SEQUENCE MODEL



CONTEXT

0.11
0.03
0.81
-0.62

0.11
0.03
0.81
-0.62

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

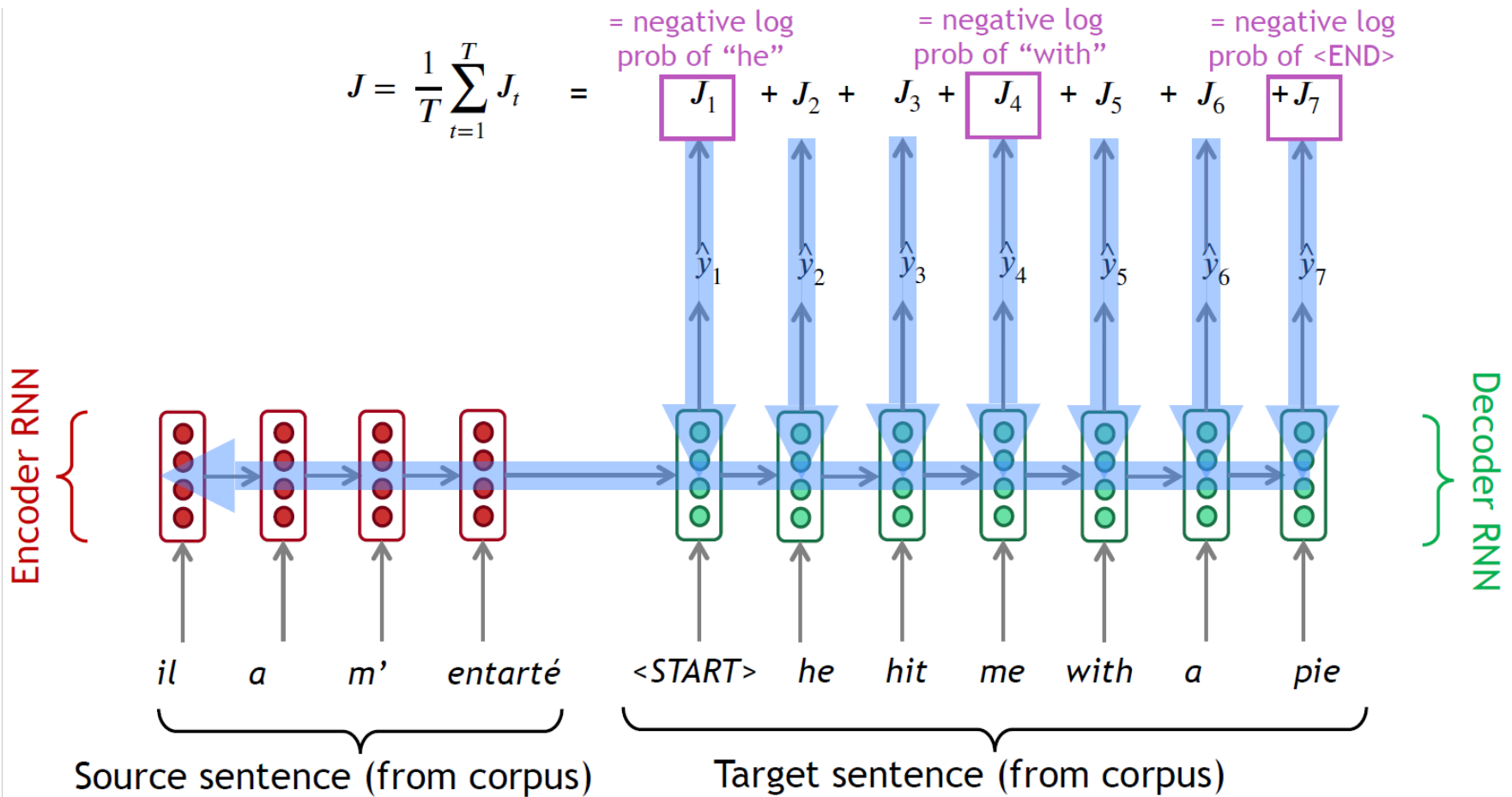
- NMT directly calculates $P(y|x)$:

$$P(y|x) = P(y_1|x) P(y_2|y_1, x) P(y_3|y_1, y_2, x) \dots P(y_T|y_1, \dots, y_{T-1}, x)$$

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...

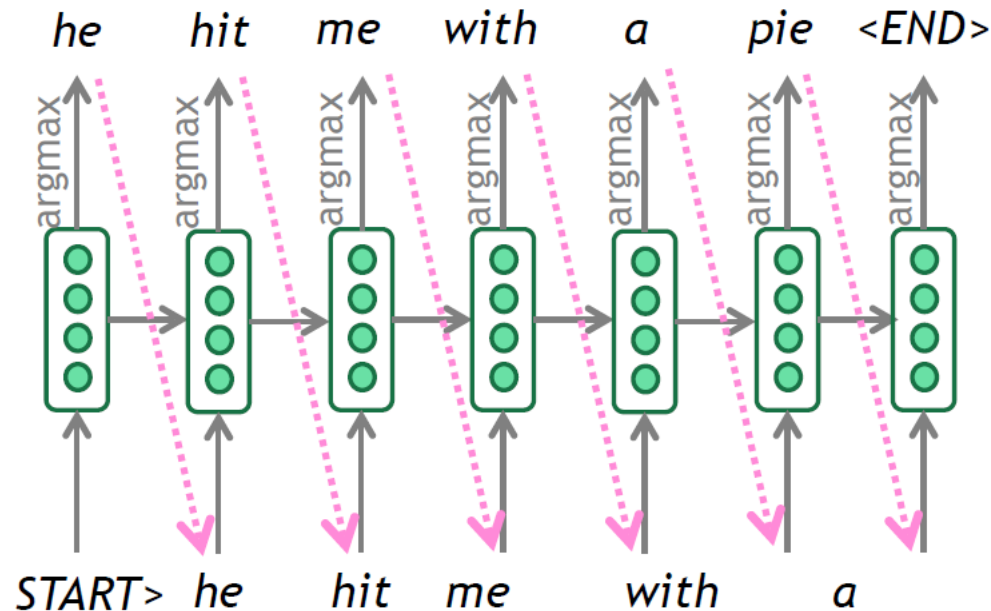
Training NMT



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?

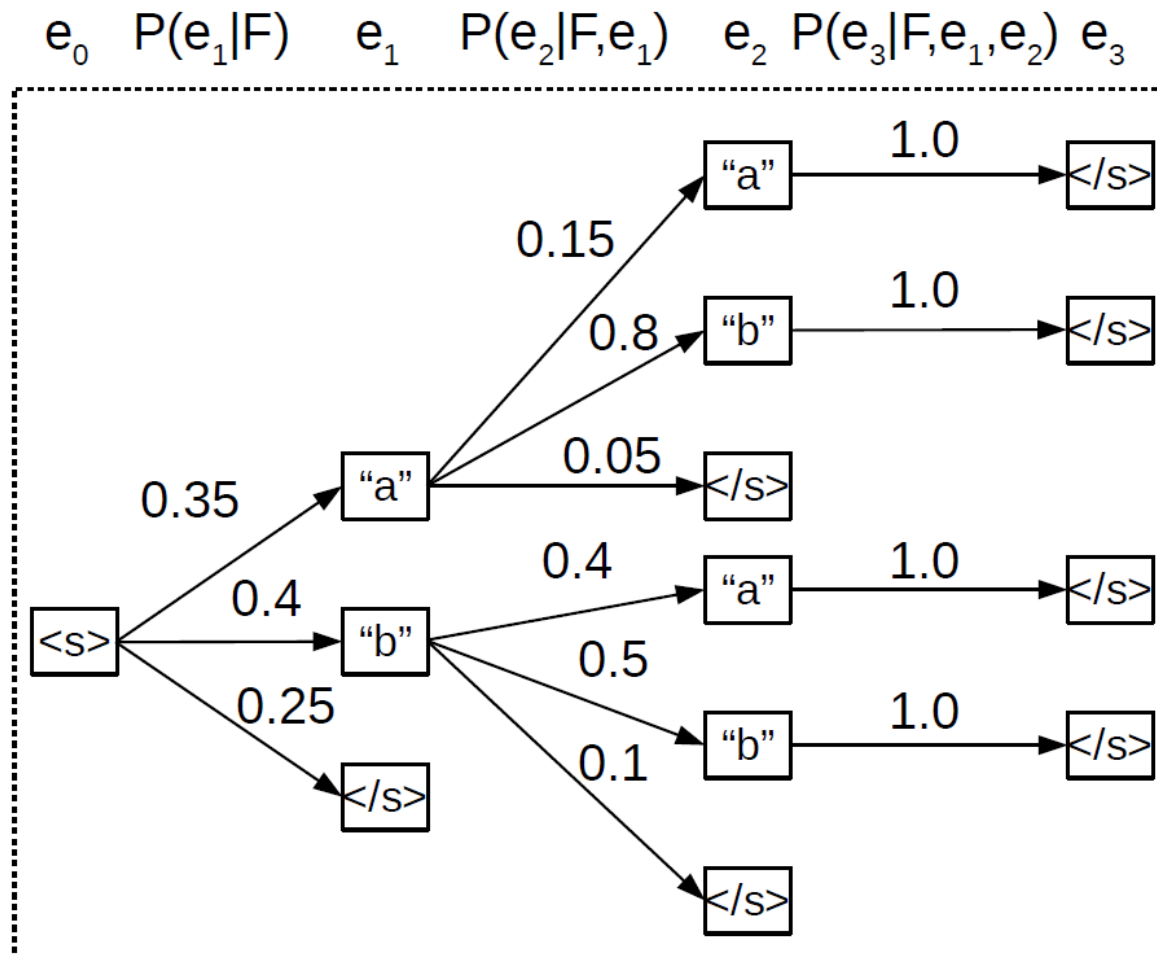


Problems with greedy decoding

- 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

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

- We could try computing all possible sequences y
- This means that on each step t of the decoder, we're tracking V^t possible partial translations, where V is vocab size
- This $O(V^T)$ 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:

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

- 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

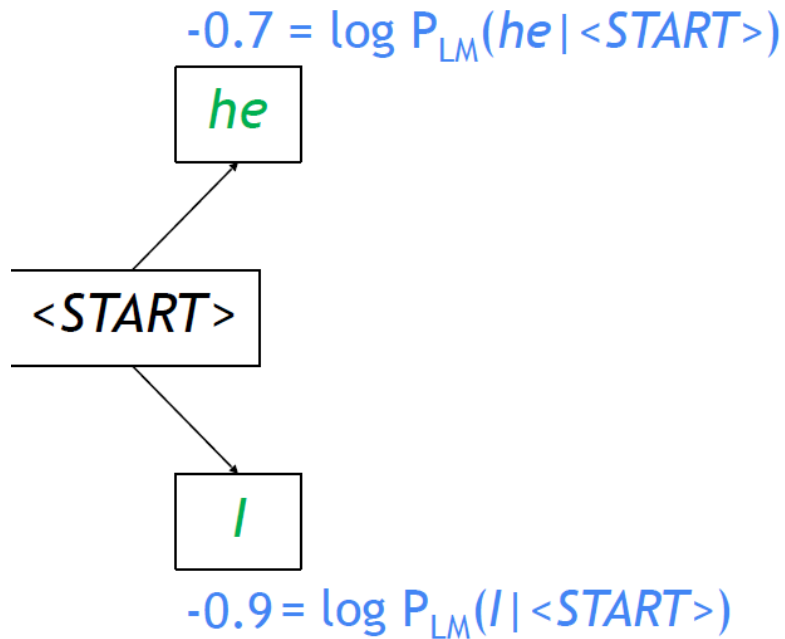
Beam size = $k = 2$. Blue numbers = $\text{score}(y_1, \dots, y_t) = \sum_{i=1}^t \log P_{\text{LM}}(y_i | y_1, \dots, y_{i-1}, x)$

<START>

Calculate prob
dist of next word

Beam search decoding: example

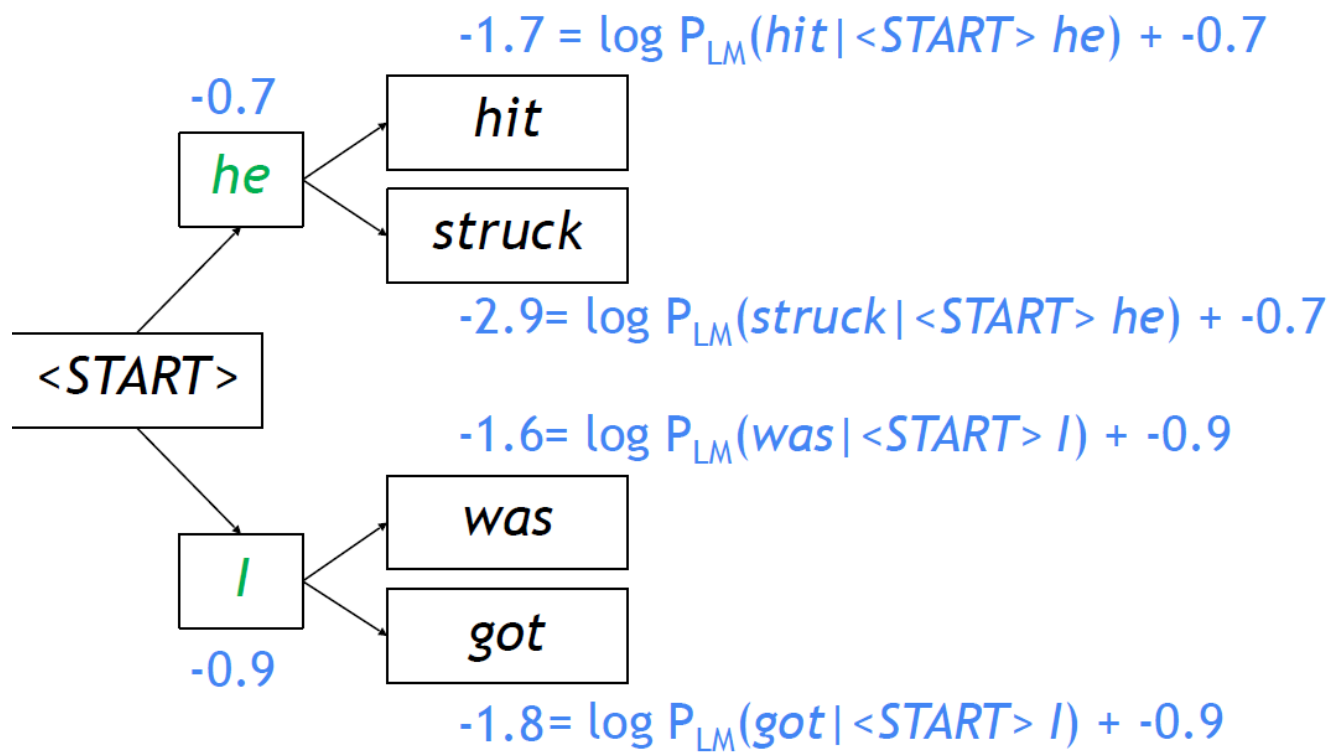
Beam size = $k = 2$. Blue numbers = $\text{score}(y_1, \dots, y_t) = \sum_{i=1}^t \log P_{\text{LM}}(y_i | y_1, \dots, y_{i-1}, x)$



Take top k words
and compute scores

Beam search decoding: example

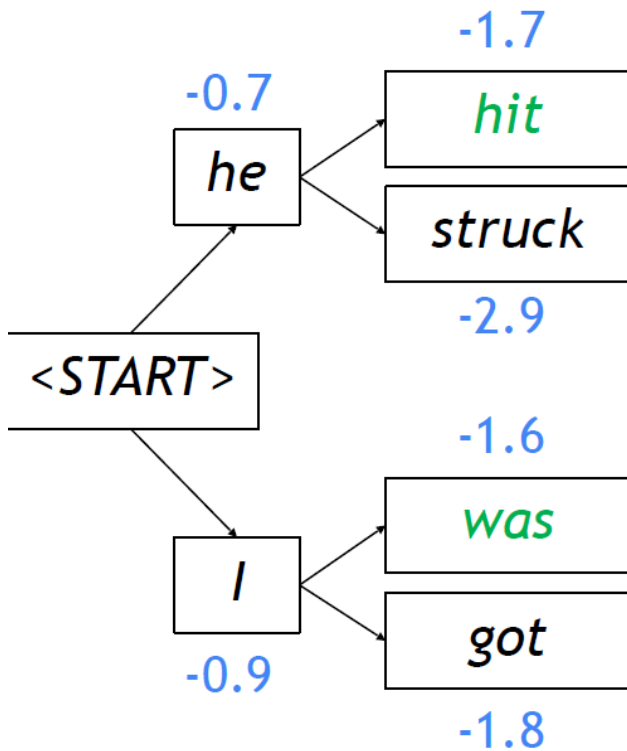
Beam size = $k = 2$. Blue numbers = $\text{score}(y_1, \dots, y_t) = \sum_{i=1}^t \log P_{\text{LM}}(y_i | y_1, \dots, y_{i-1}, x)$



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

Beam search decoding: example

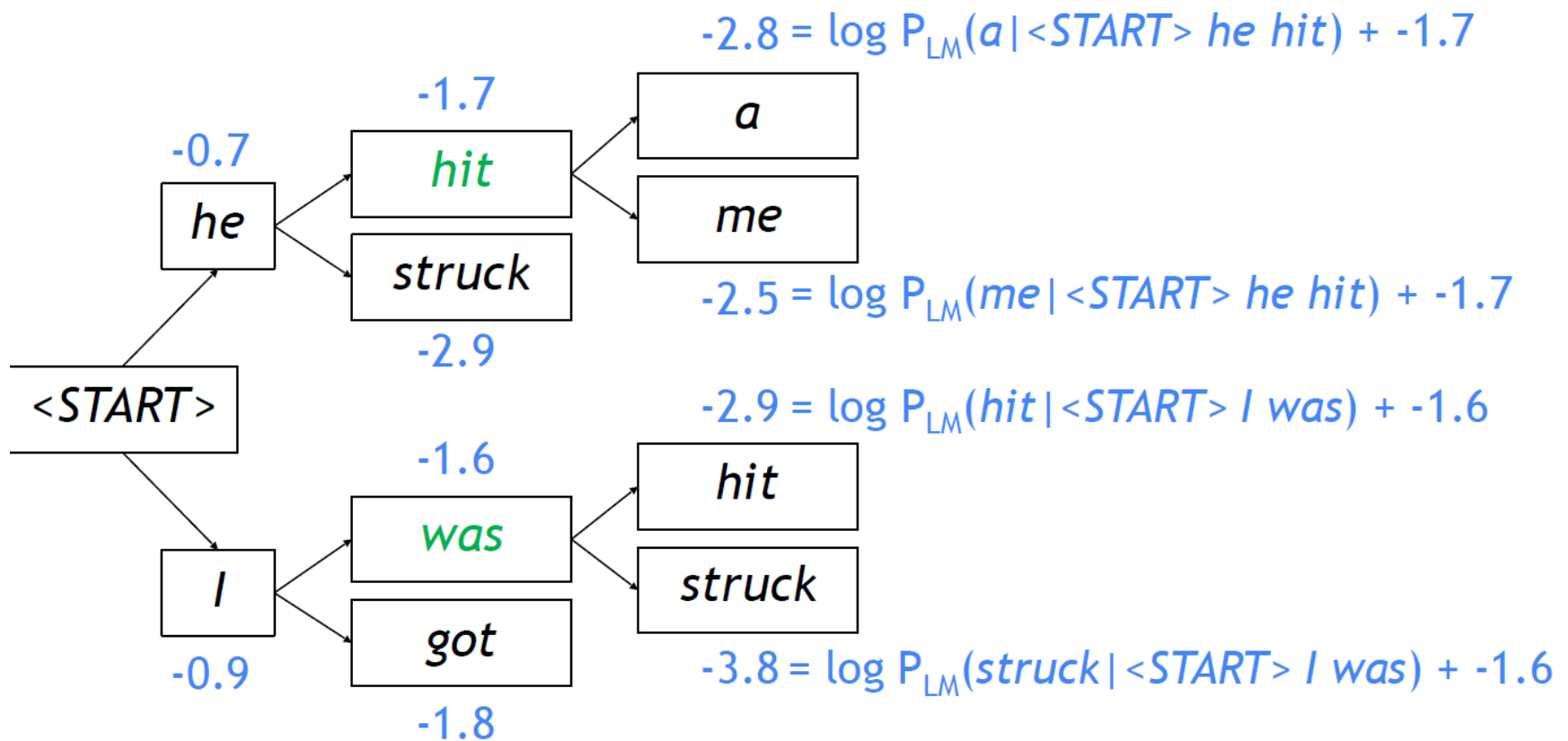
Beam size = $k = 2$. Blue numbers = $\text{score}(y_1, \dots, y_t) = \sum_{i=1}^t \log P_{\text{LM}}(y_i | y_1, \dots, y_{i-1}, x)$



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

Beam search decoding: example

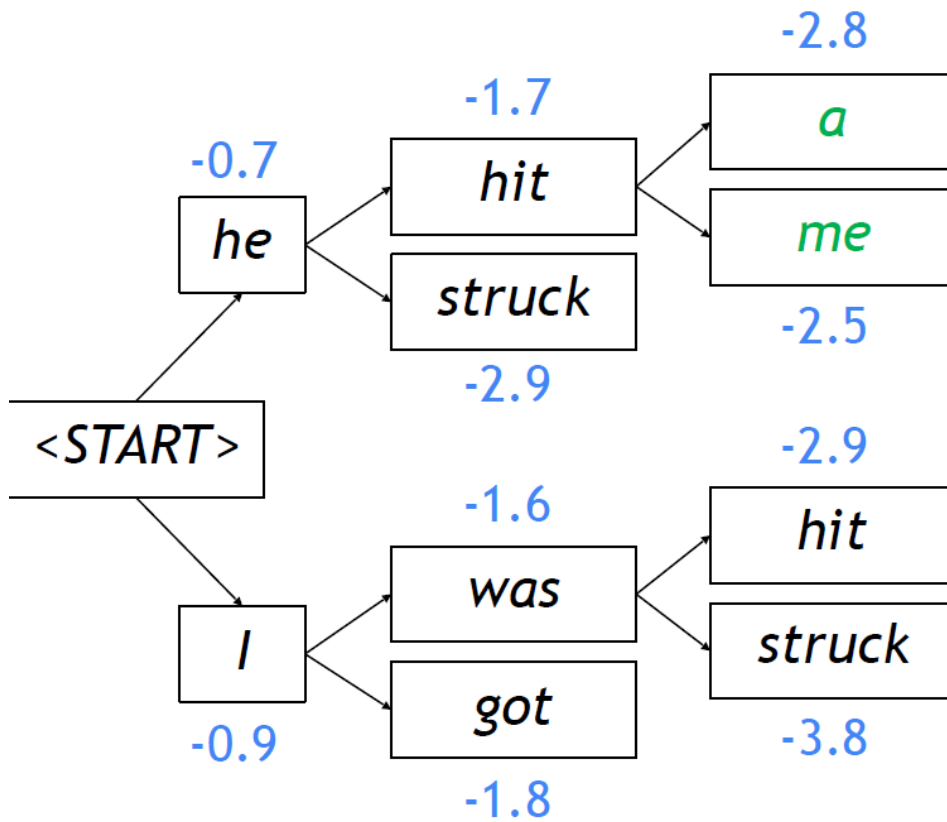
Beam size = $k = 2$. Blue numbers = $\text{score}(y_1, \dots, y_t) = \sum_{i=1}^t \log P_{\text{LM}}(y_i | y_1, \dots, y_{i-1}, x)$



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

Beam search decoding: example

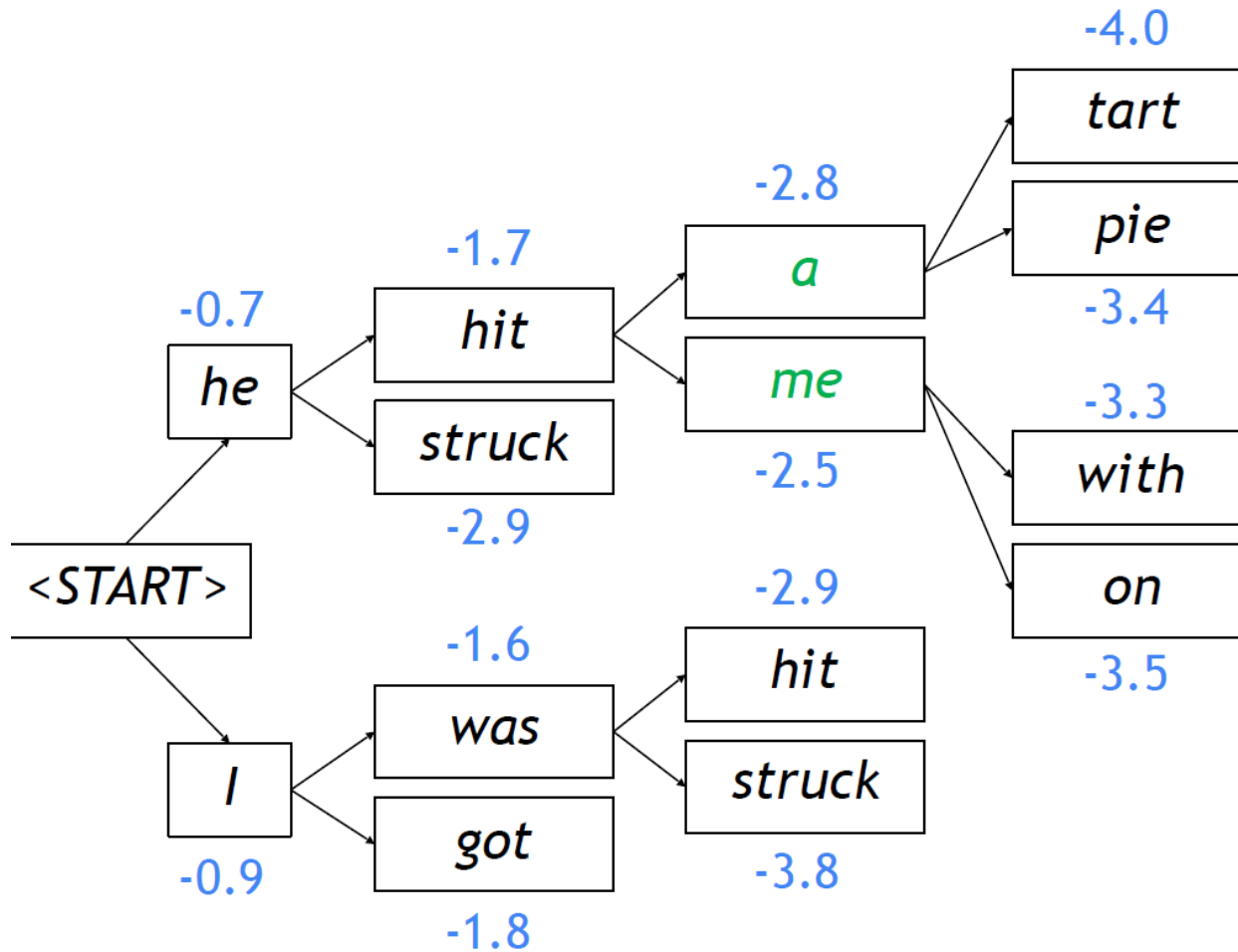
Beam size = $k = 2$. Blue numbers = $\text{score}(y_1, \dots, y_t) = \sum_{i=1}^t \log P_{\text{LM}}(y_i | y_1, \dots, y_{i-1}, x)$



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

Beam search decoding: example

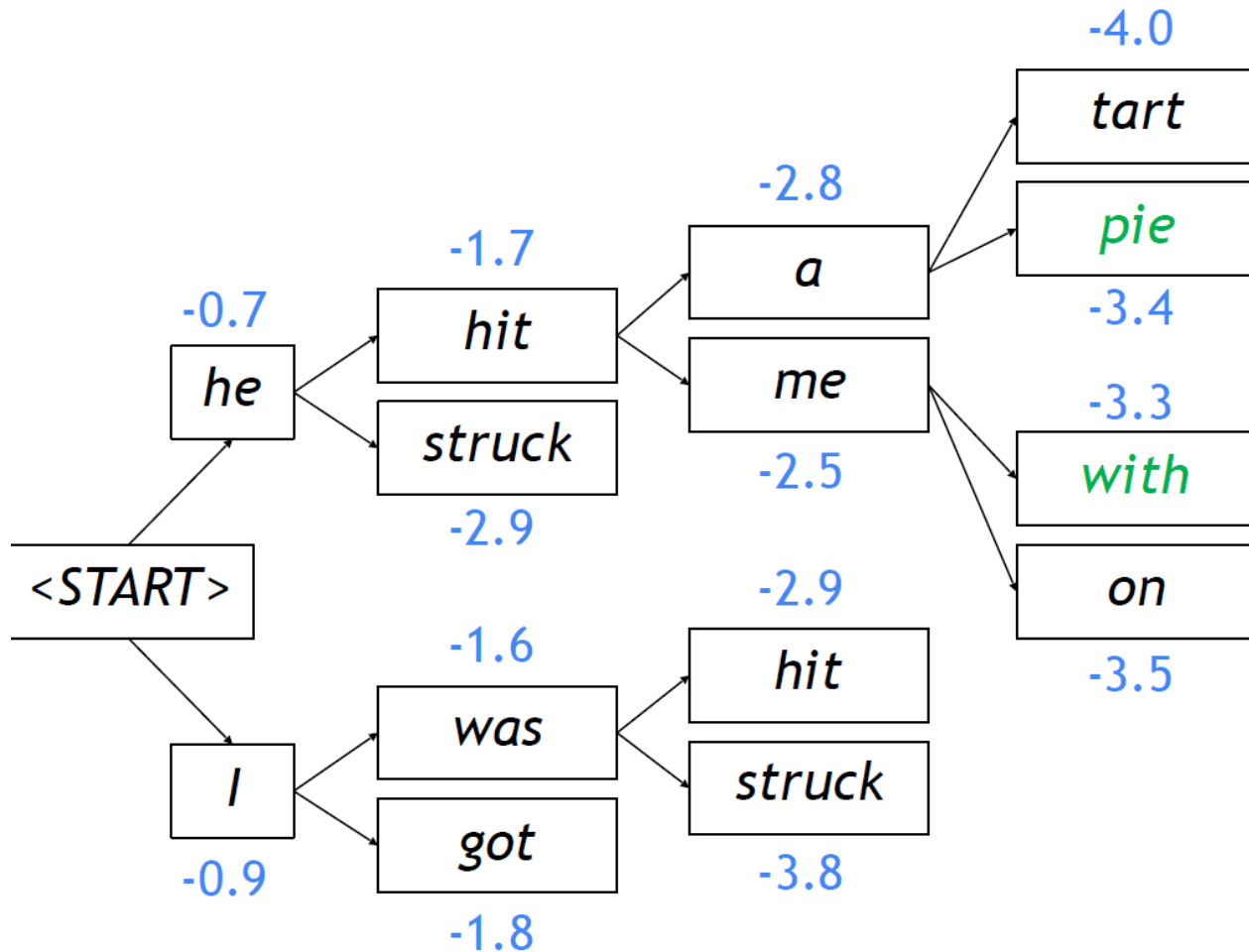
Beam size = $k = 2$. Blue numbers = $\text{score}(y_1, \dots, y_t) = \sum_{i=1}^t \log P_{\text{LM}}(y_i | y_1, \dots, y_{i-1}, x)$



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

Beam search decoding: example

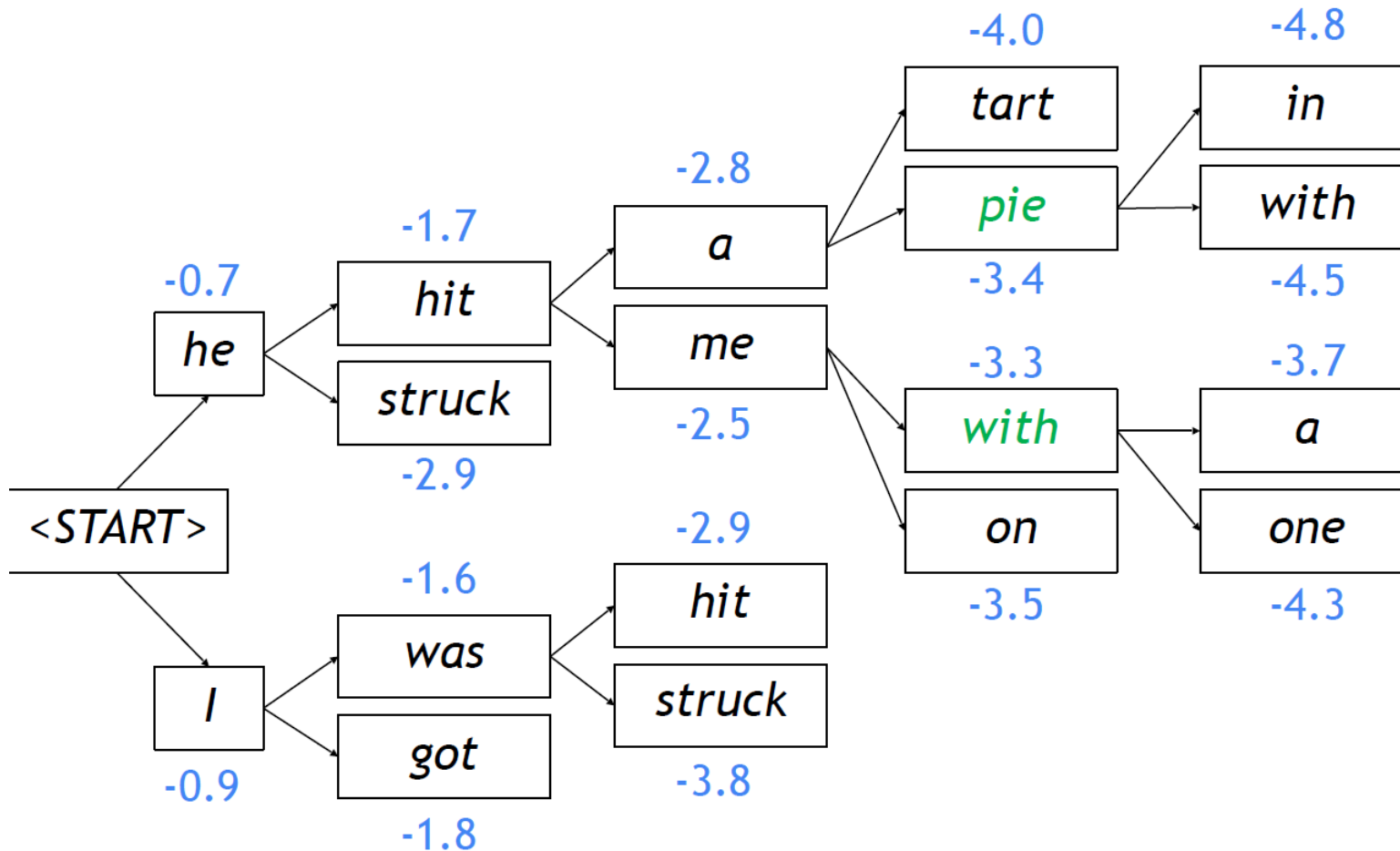
Beam size = $k = 2$. Blue numbers = $\text{score}(y_1, \dots, y_t) = \sum_{i=1}^t \log P_{\text{LM}}(y_i | y_1, \dots, y_{i-1}, x)$



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

Beam search decoding: example

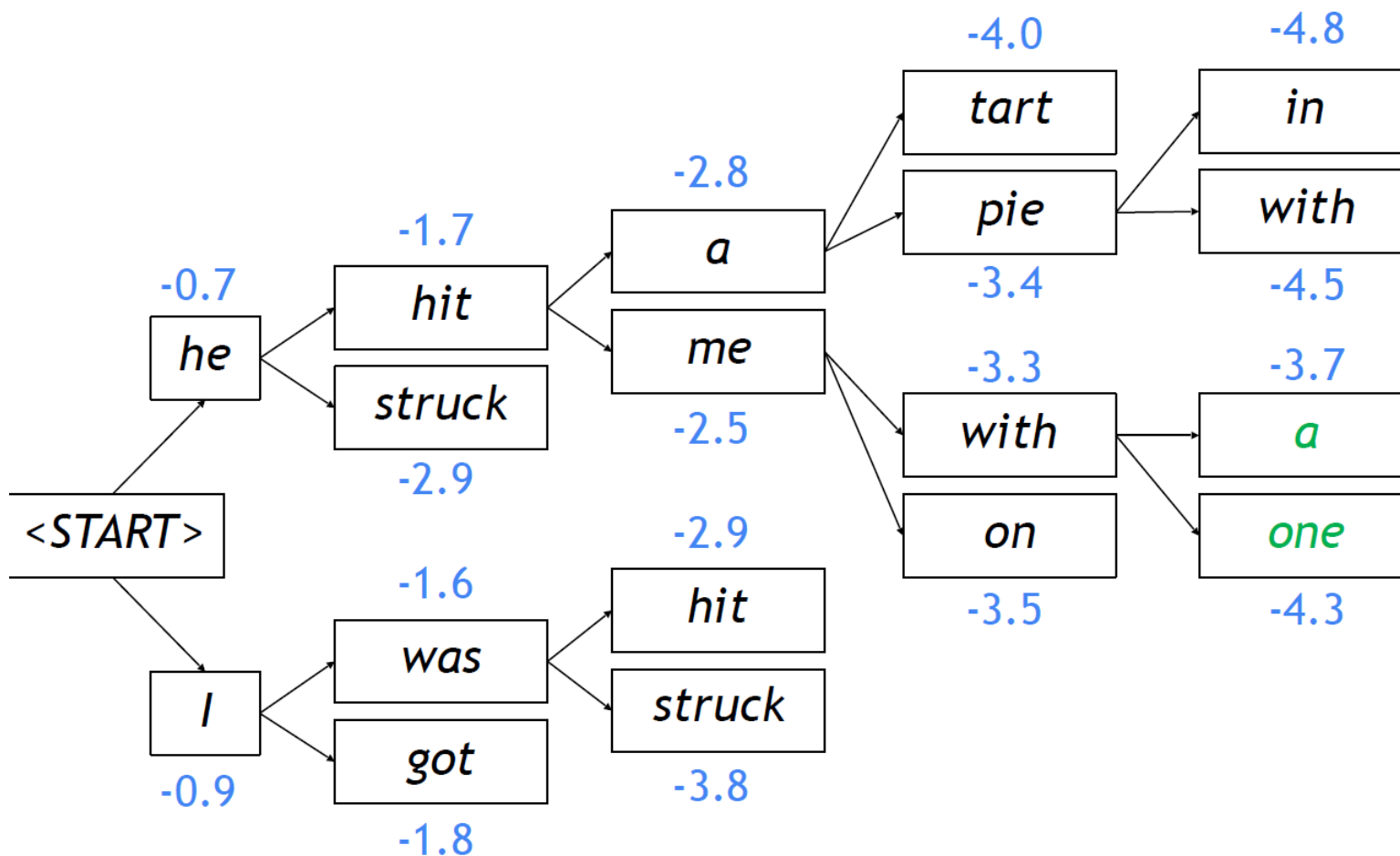
Beam size = $k = 2$. Blue numbers = $\text{score}(y_1, \dots, y_t) = \sum_{i=1}^t \log P_{\text{LM}}(y_i | y_1, \dots, y_{i-1}, x)$



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

Beam search decoding: example

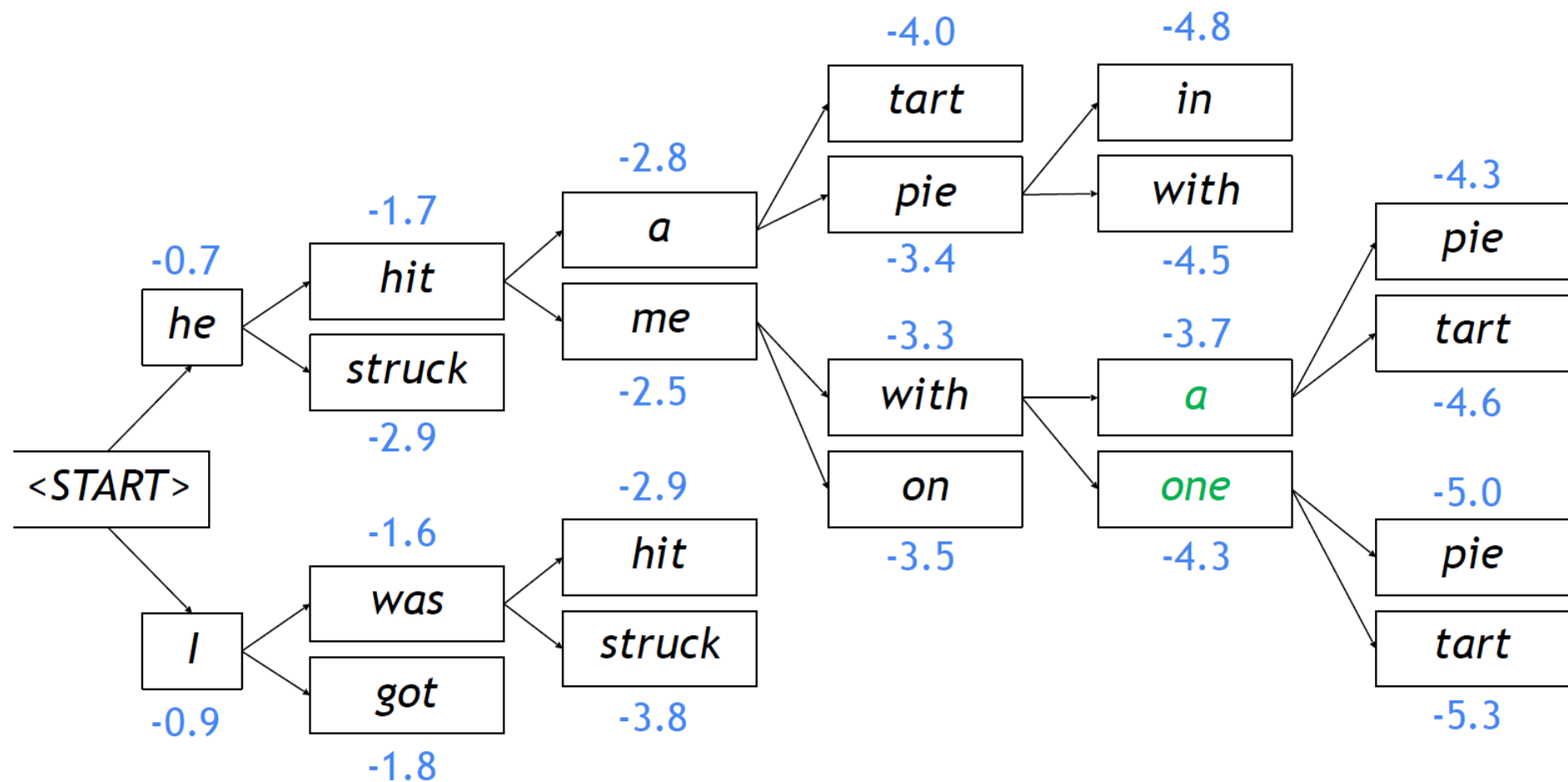
Beam size = $k = 2$. Blue numbers = $\text{score}(y_1, \dots, y_t) = \sum_{i=1}^t \log P_{\text{LM}}(y_i | y_1, \dots, y_{i-1}, x)$



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

Beam search decoding: example

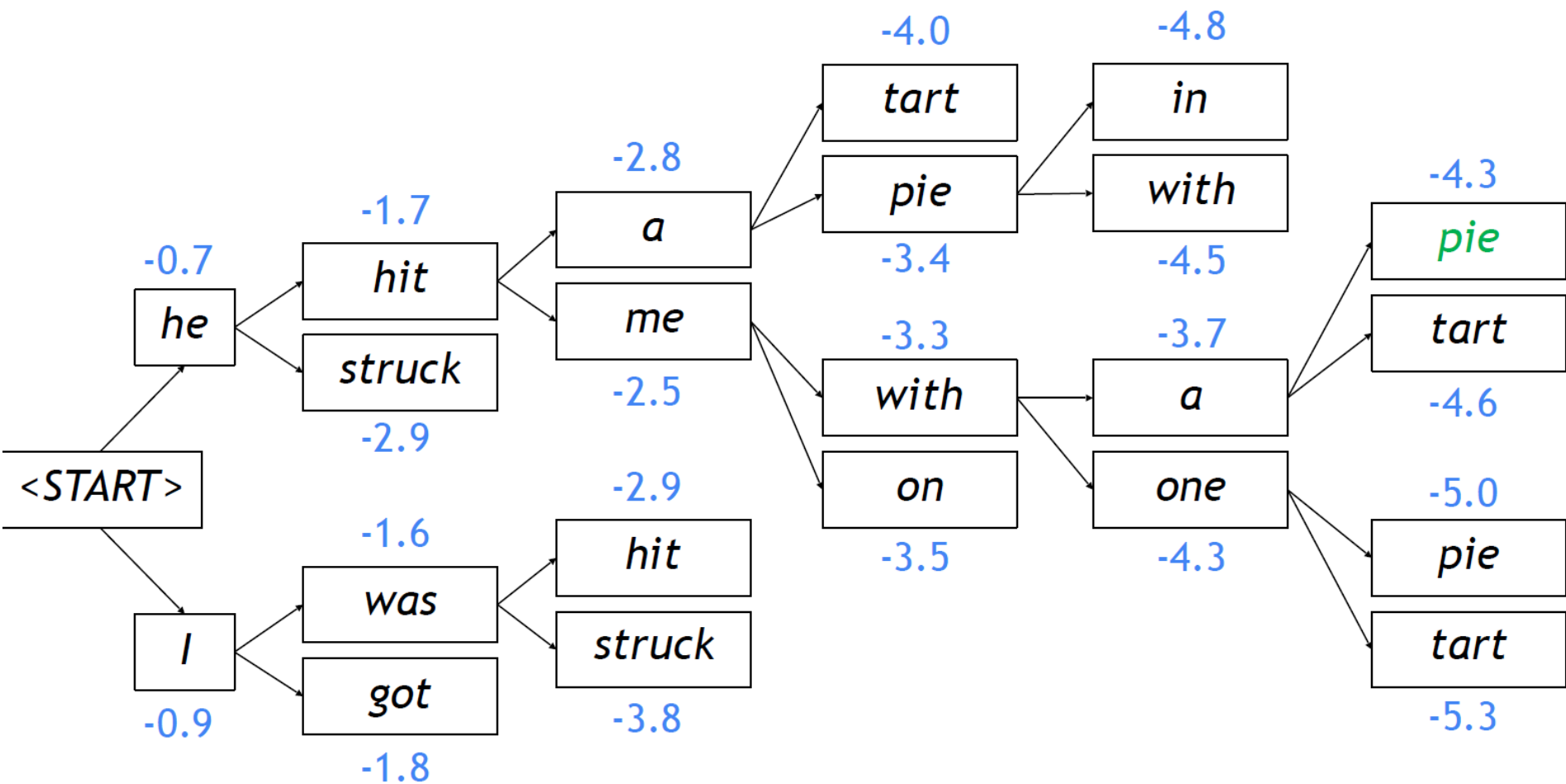
Beam size = $k = 2$. Blue numbers = $\text{score}(y_1, \dots, y_t) = \sum_{i=1}^t \log P_{\text{LM}}(y_i | y_1, \dots, y_{i-1}, x)$



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

Beam search decoding: example

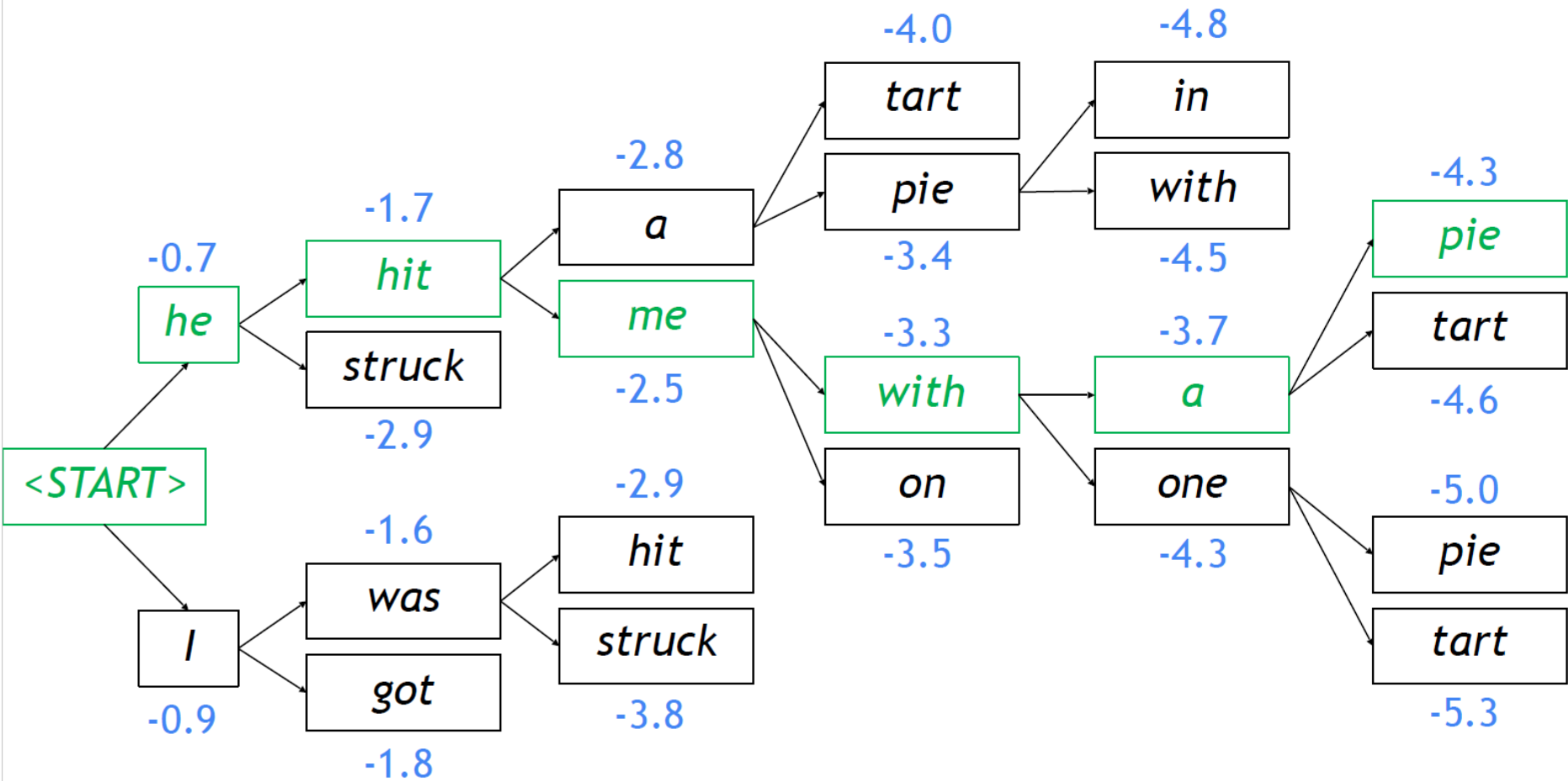
Beam size = $k = 2$. Blue numbers = $\text{score}(y_1, \dots, y_t) = \sum_{i=1}^t \log P_{\text{LM}}(y_i | y_1, \dots, y_{i-1}, x)$



This is the top-scoring hypothesis!

Beam search decoding: example

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



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, \dots, y_t on our list has a score

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

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

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

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

Effect of beam size in chit-chat dialogue

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



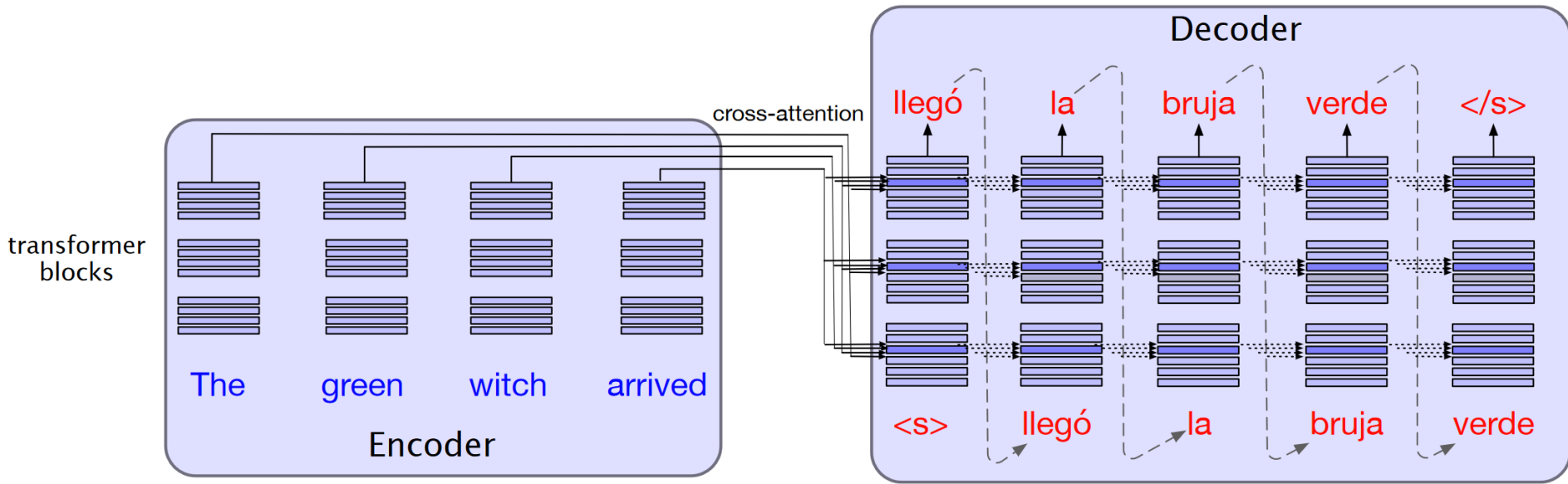
Human
chit-chat
partner

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

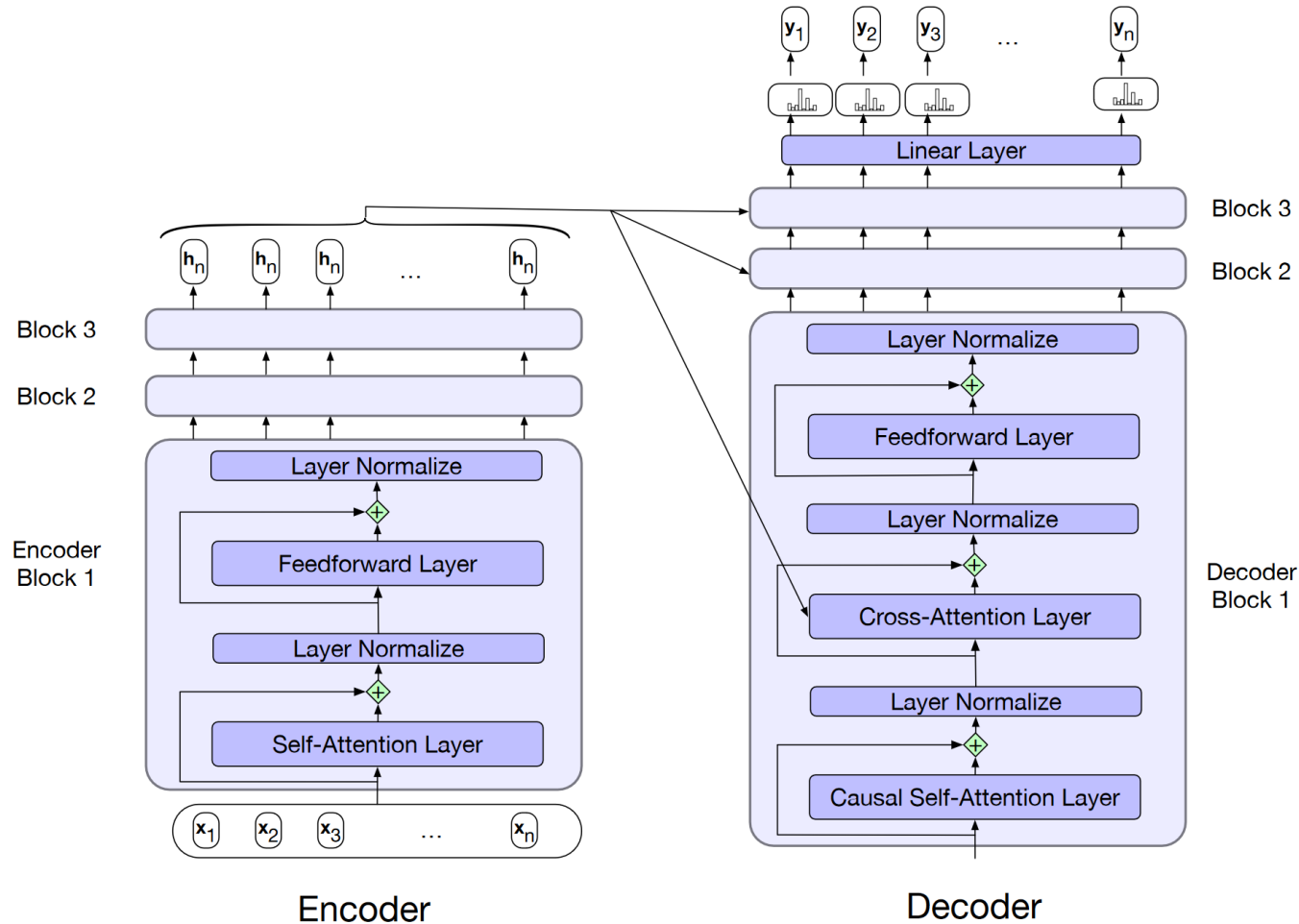
Low beam size:
More on-topic but
nonsensical;
bad English

High beam size:
Converges to safe,
“correct” response,
but it’s generic and
less relevant

Transformer is encoder-decoder



Attention in transformer



The final output of the encoder $H_{\text{enc}} = h_1, \dots, h_T$ is the context used in the decoder. The decoder is a standard transformer except for the cross-attention layer, which takes the decoder output H_{enc} and uses it to form its K and V inputs.

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!

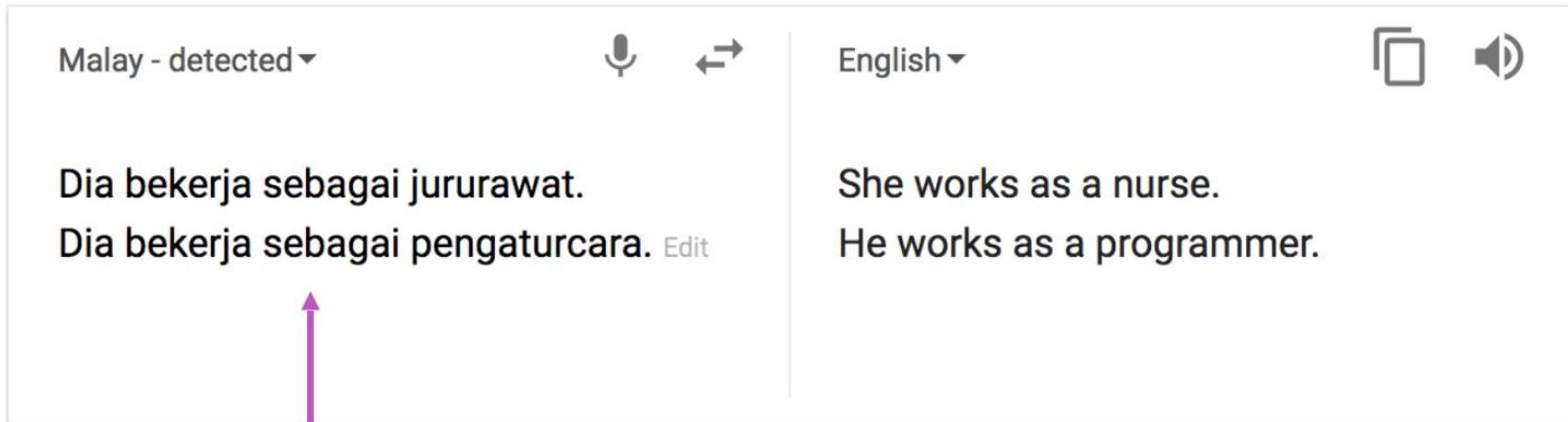
So is Machine Translation solved?

- 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

The screenshot shows a Google Translate interface. The source language is Spanish (detected) and the target language is English. The input text is "Mi amigo no tiene pelos en la lengua" and the output text is "My friend has no hair on the tongue". The interface includes a microphone icon, a speaker icon, a character count "36/5000", and a star icon for saving the translation.

So is Machine Translation solved?

- NMT picks up biases in training data



The screenshot shows a machine translation interface with two columns. The left column is labeled 'Malay - detected' and contains the text: 'Dia bekerja sebagai jururawat.' followed by 'Dia bekerja sebagai pengaturcara. Edit'. The right column is labeled 'English' and contains the text: 'She works as a nurse.' followed by 'He works as a programmer.'. A purple arrow points from the text 'Didn't specify gender' below to the Malay text 'Dia bekerja sebagai pengaturcara. Edit'.

Didn't specify gender

So is Machine Translation solved?

- Uninterpretable systems do strange things

The image displays two examples of Google Translate's output. The top example shows a Somali input of 'ag ag ag ag ag ag ag ag ag ag ag ag ag ag ag ag ag' being translated into English as 'As the name of the LORD was written in the Hebrew language, it was written in the language of the Hebrew Nation'. The bottom example shows a Maori input of 'dog dog dog dog dog dog dog dog dog dog dog dog dog dog dog' being translated into English as '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'. Both examples include interface elements like language dropdowns, a bidirectional arrow, and copy/speak icons.

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.)

Tired is I.

Cookies taste good!

I am tired.

Fidelity	Fluency
5	2
1	5
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

- Precision₁, precision₂, etc., are computed over all candidate sentences C in the test set

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

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

BLEU-2:

Candidate 1: Mary no slap the witch green.

Best Reference: Mary did not slap the green witch.

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

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

Best Reference: Mary did not smack the green witch.

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

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

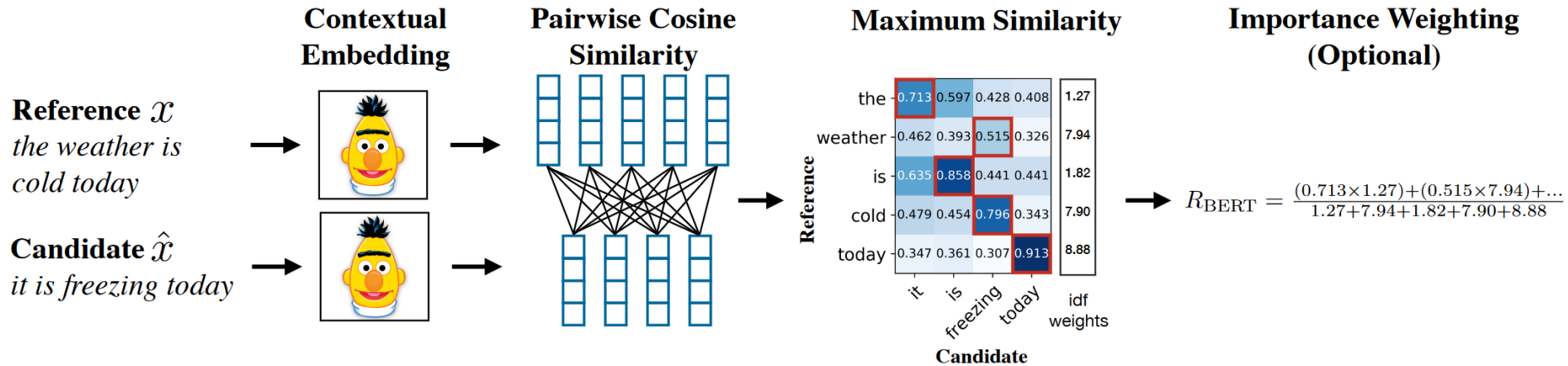
- no good measure exists on the level of sentence
- no good measure exists of an absolute translation quality

BERTScore

- for the reference x and the candidate \tilde{x} , compute a BERT embedding for each token x_i and \tilde{x}_j .
- Each pair of tokens its cosine similarity.
Each token in x is matched to a token in \tilde{x} to compute recall, and each token in \tilde{x} is matched to a token in x to compute precision (with each token greedily matched to the most similar token in the corresponding sentence).
- BERTSCORE provides precision, recall, and F_1

$$R_{\text{BERT}} = \frac{1}{|x|} \sum_{x_i \in x} \max_{\tilde{x}_j \in \tilde{x}} x_i \cdot \tilde{x}_j \quad P_{\text{BERT}} = \frac{1}{|\tilde{x}|} \sum_{\tilde{x}_j \in \tilde{x}} \max_{x_i \in x} x_i \cdot \tilde{x}_j$$

BERTScore illustration



Improvements in MT

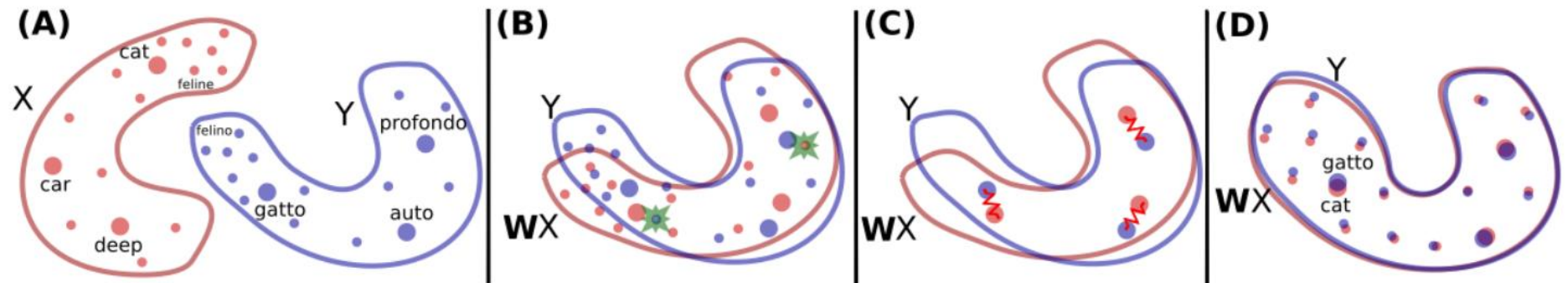
- large corpora
- adaptations to specific domains, e.g., IT, pharmacy, automotive industry
- terminological dictionaries, terminology lists, translation memories

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

NMT in Slovene

- RSDO project
- English-Slovene and Slovene-English
- Demo at <https://www.slovenscina.eu/prevajalnik>
- following the NVIDIA NeMo NMT AAYN recipe
- the training corpus Parallel corpus EN-SL RSDO4 1.0 (<https://www.clarin.si/repository/xmlui/handle/11356/1457>)
- training 32.638.758 translation pairs
- validation: 8.163 translation pairs.
- BLEU score: 48.3191 Slovene to English
- BLEU score: 53.8191 English to Slovene