University of Ljubljana, Faculty of Computer and Information Science

Machine translation

Prof Dr Marko Robnik-Šikonja

Natural Language Processing, Edition 2024

Contents

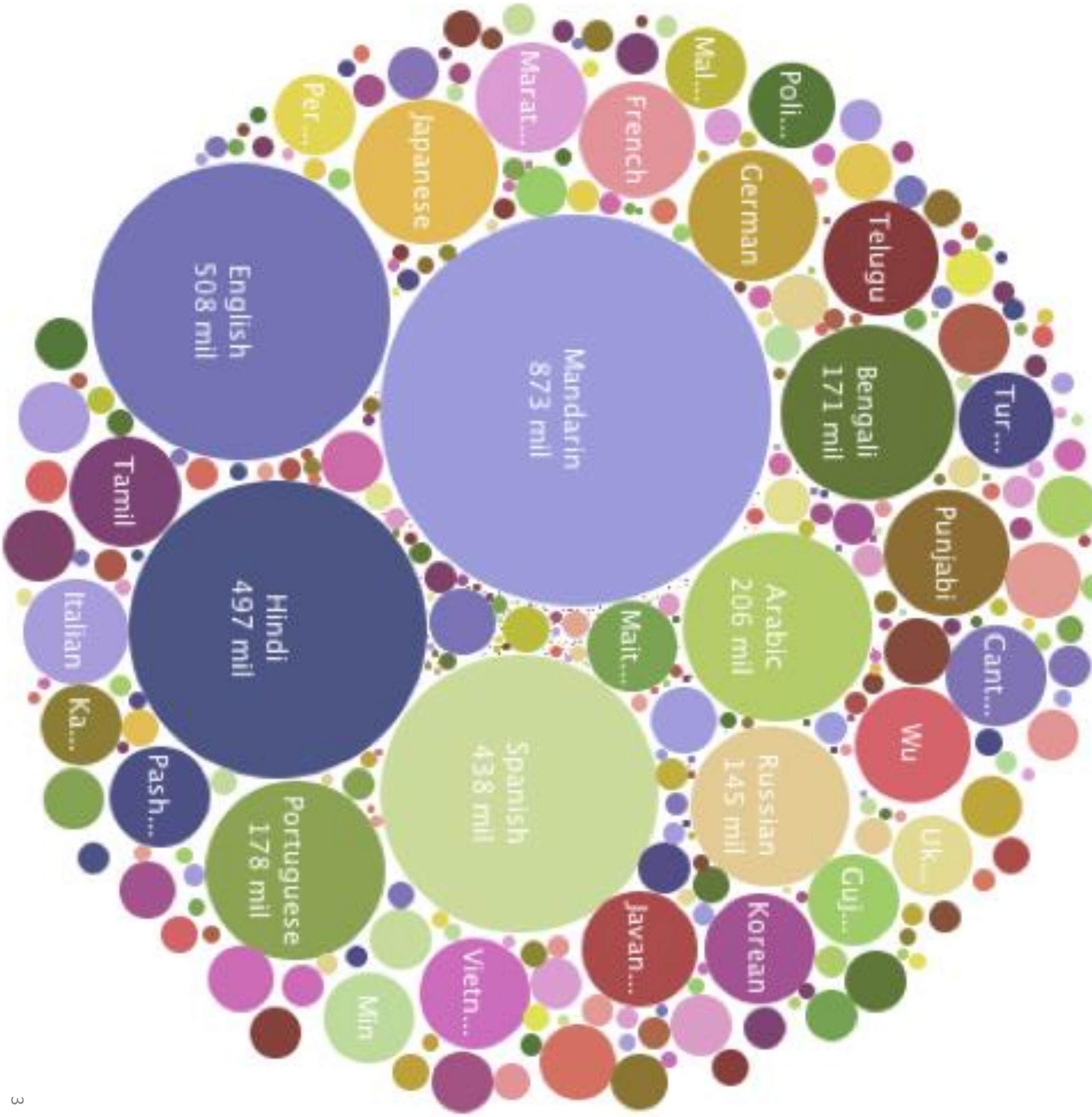
- statistical machine translation
- neural machine translation using sequence to sequence approach

Literature:

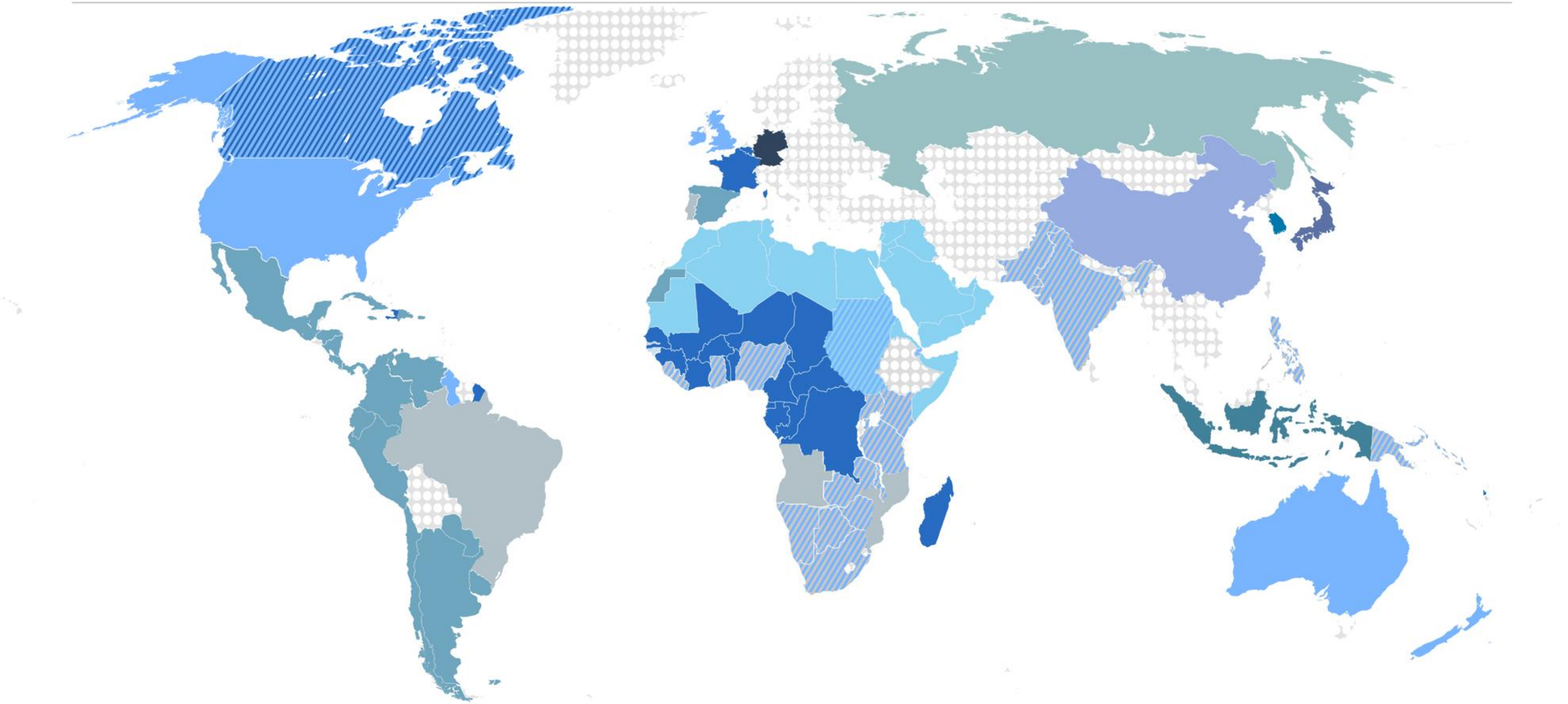
- Dan Jurafsky and James H. Martin. Speech and Language Processing (3rd ed. draft)

Word languages

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

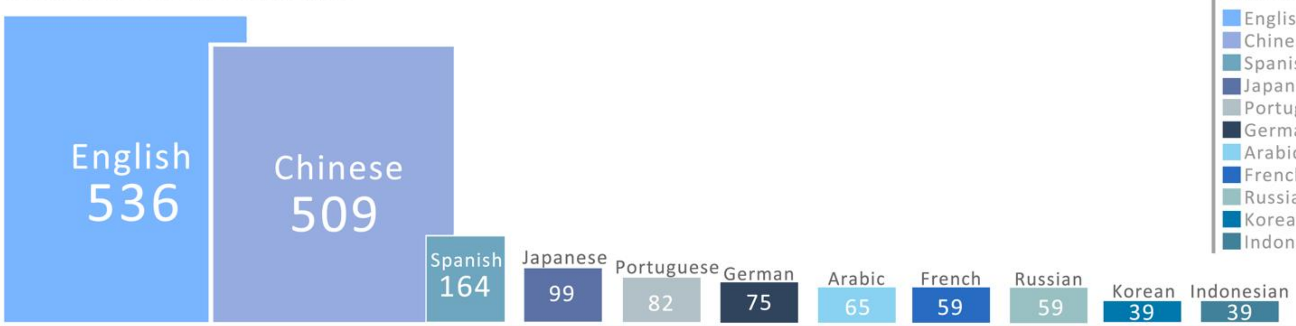


Top Languages on the Internet



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%

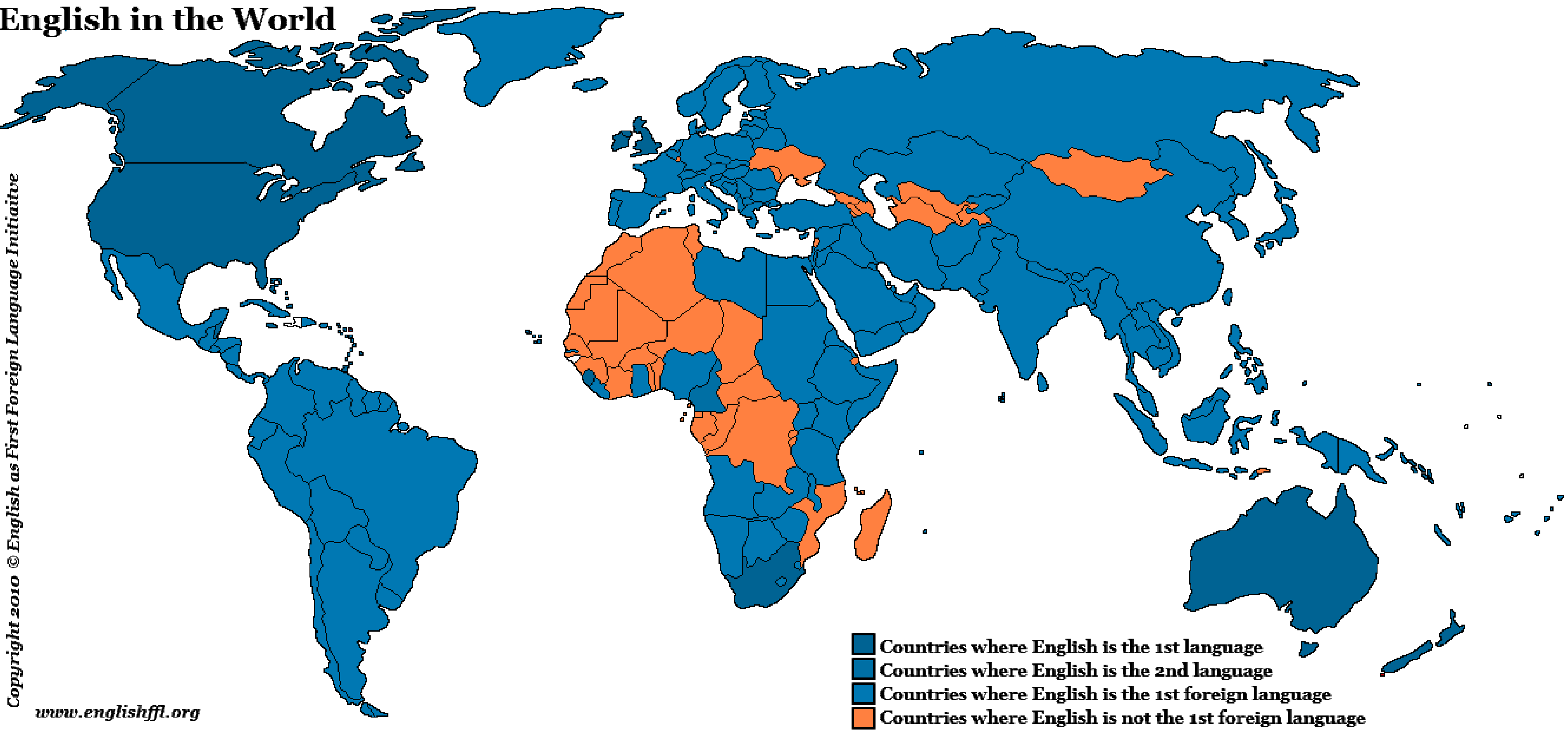
World population by Language (mln)

- 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

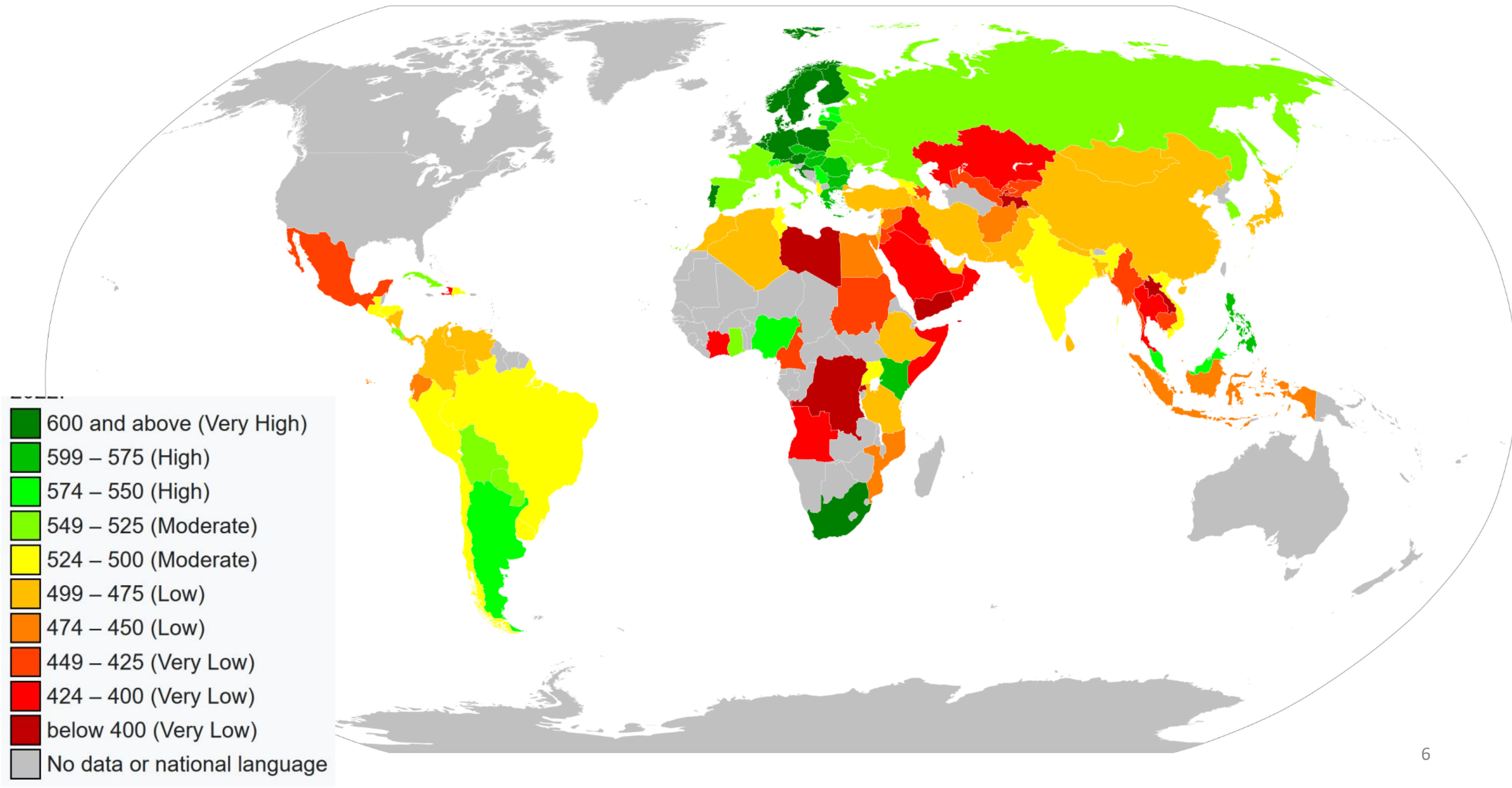
Web: languageconnect.net / E-mail: info@languageconnect.net / Follow us: lconnect

English as lingua franca?



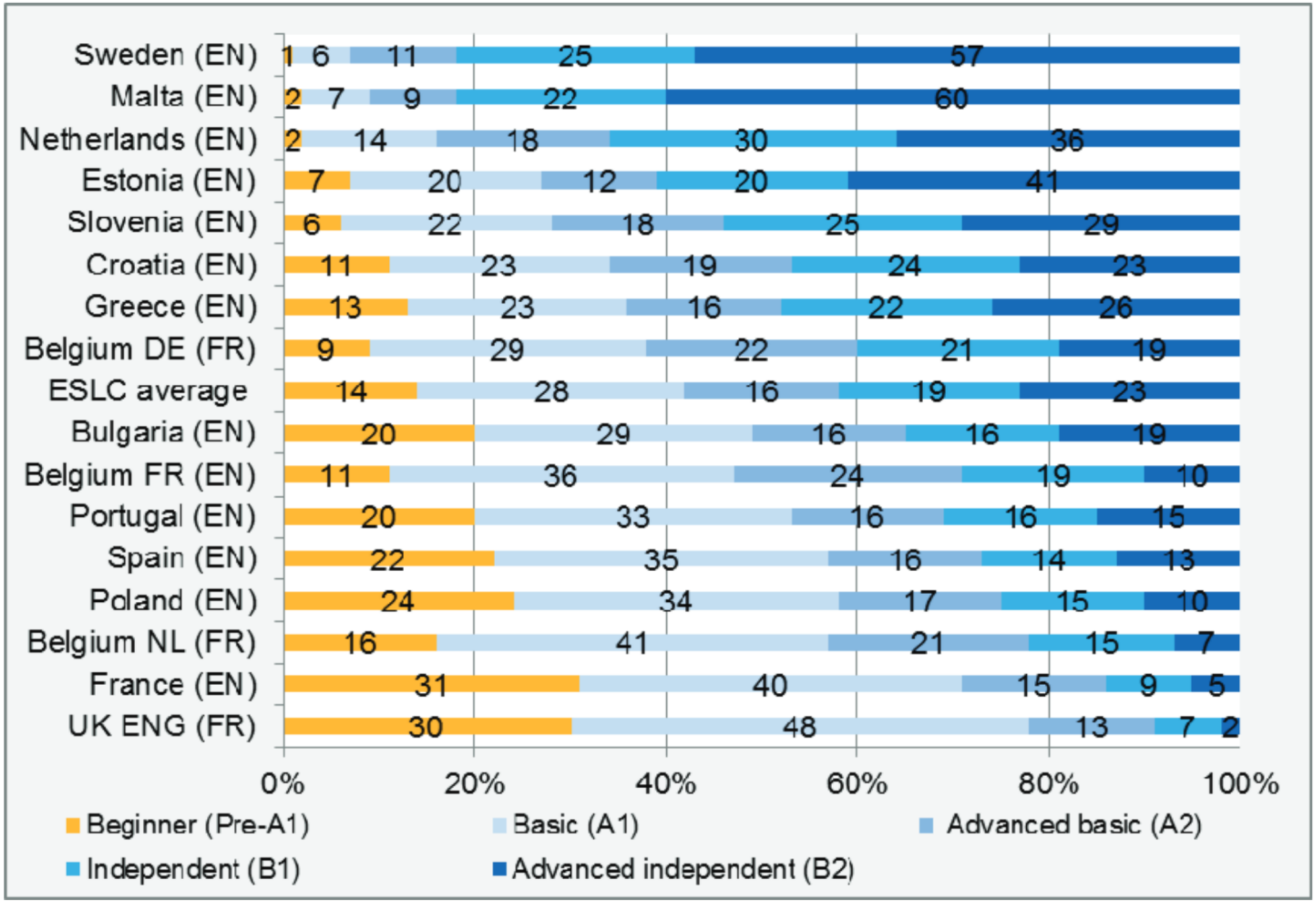
Global English proficiency index

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



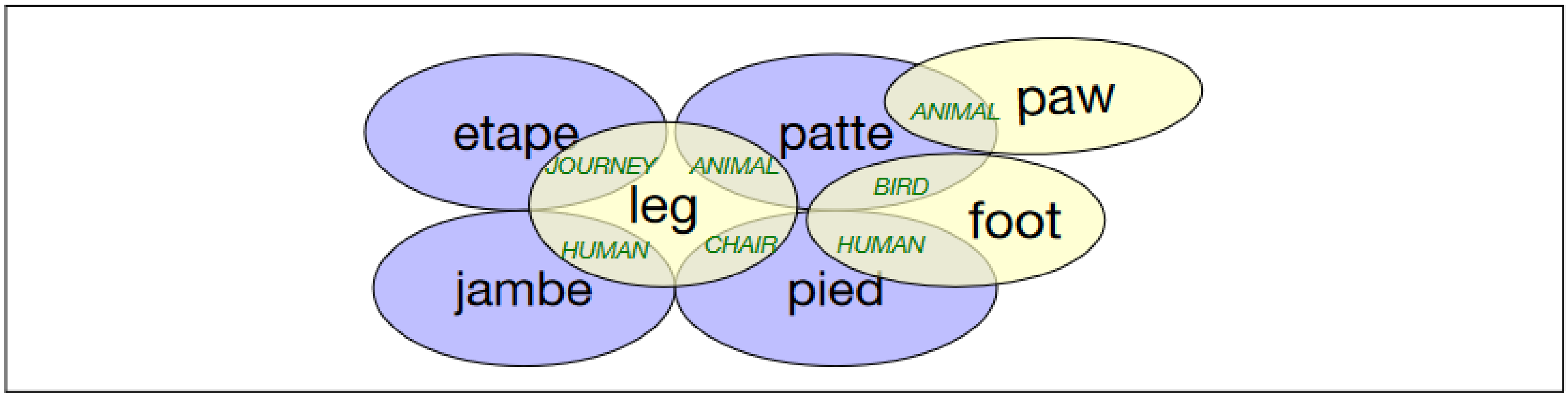
Language proficiency

- 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

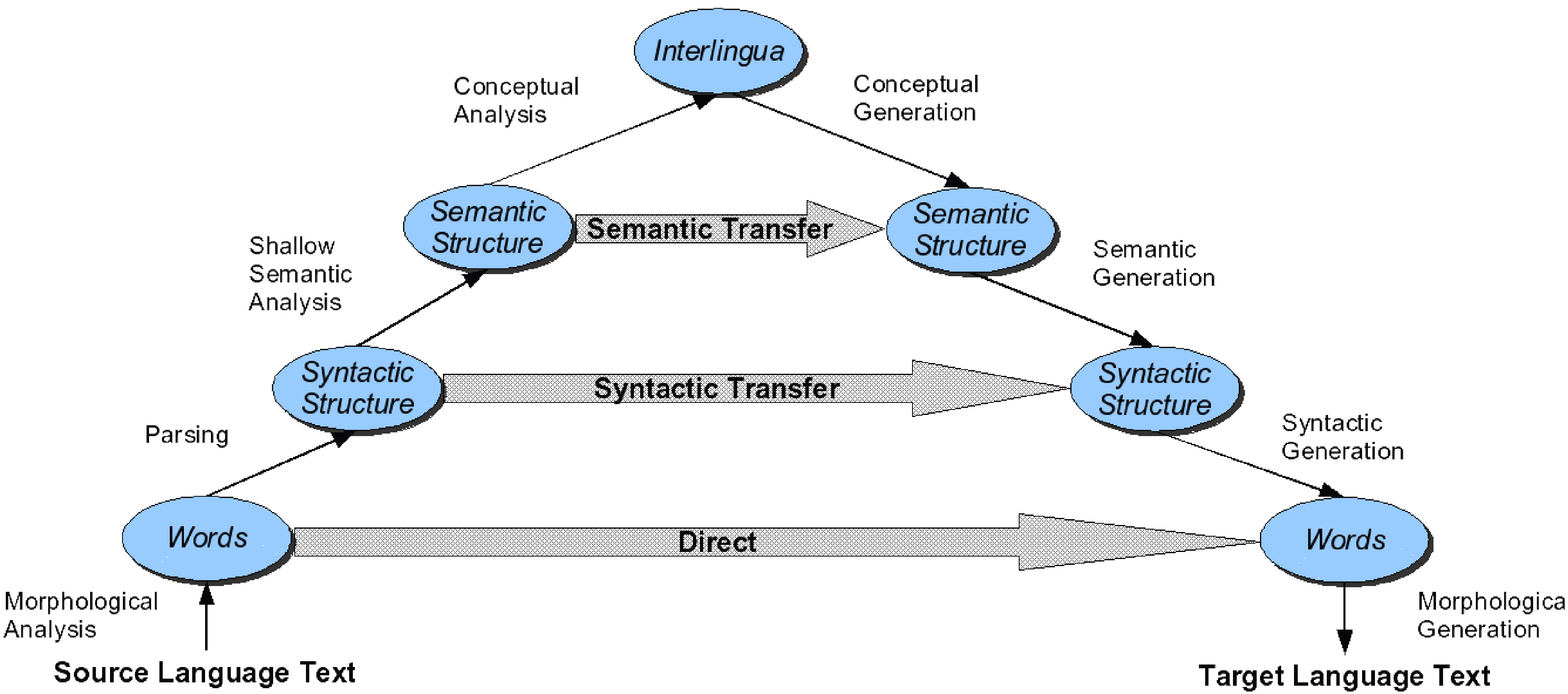
    `The Lord is for me like somebody who looks after animals with cotton-like hair`

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

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

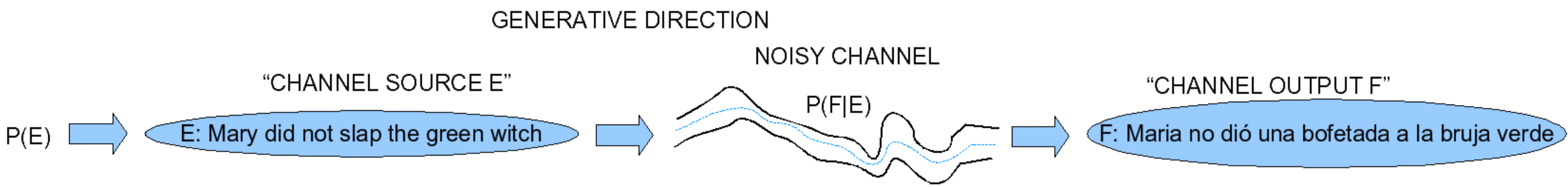
$$= \underset{E \in English}{\operatorname{argmax}} \frac{P(F \mid E) P(E)}{P(F)}$$

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

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



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)

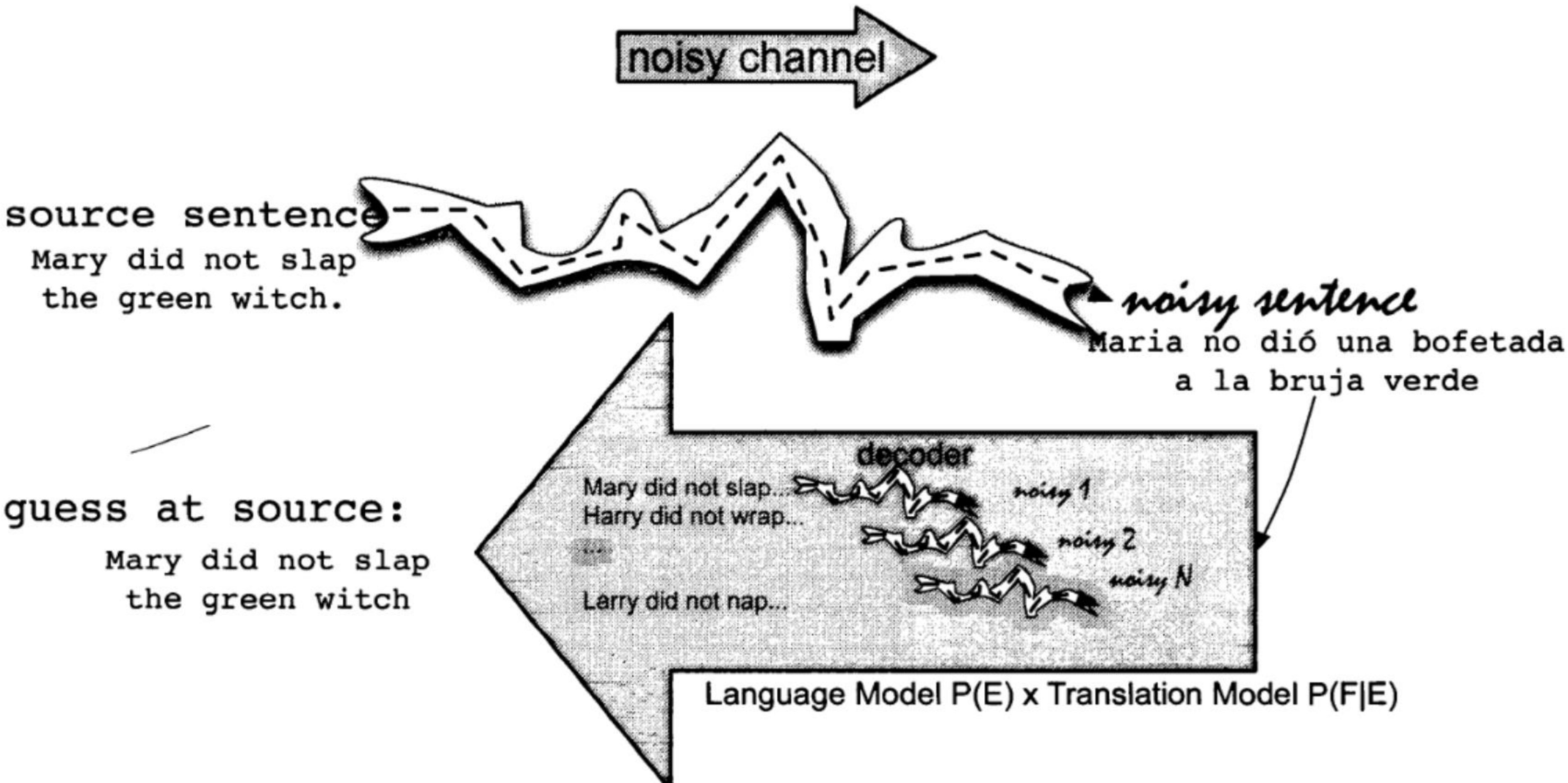
  (F,E) don't look like translations → low P(F | E)

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

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

Noisy channel model

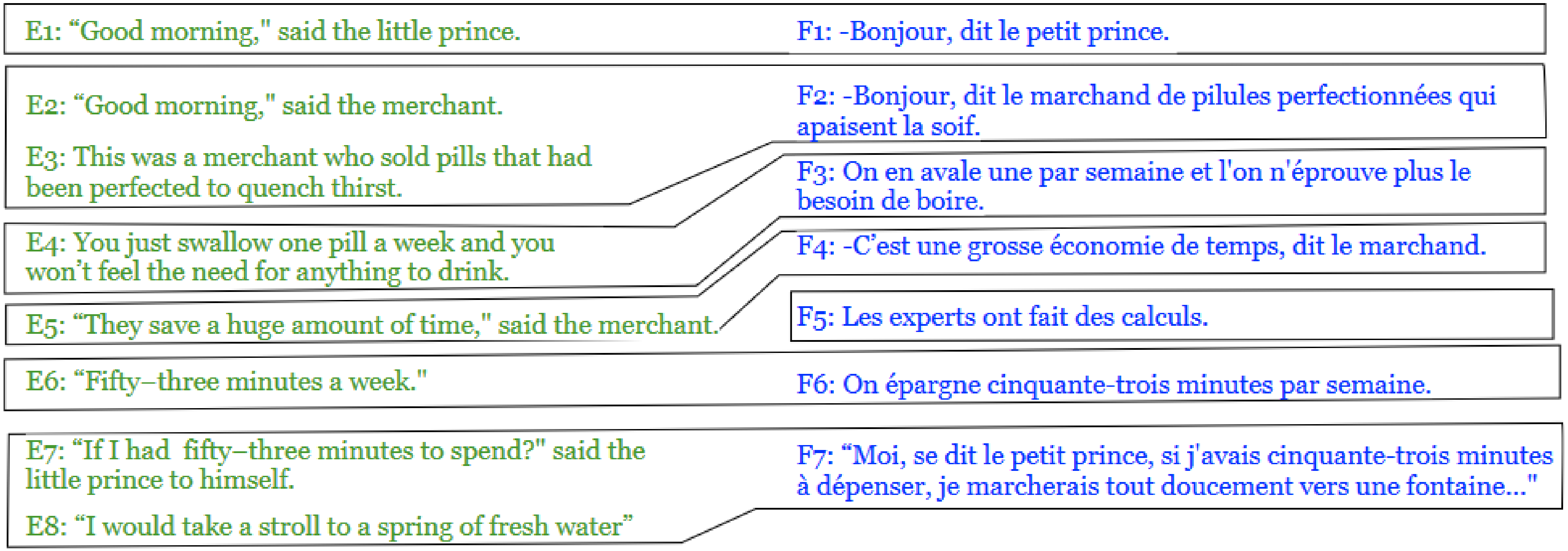
- inference goes backwards



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



- Sentence alignment takes sentences $E_1, \ldots, E_n$, and $F_1, \ldots, F_n$ 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

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

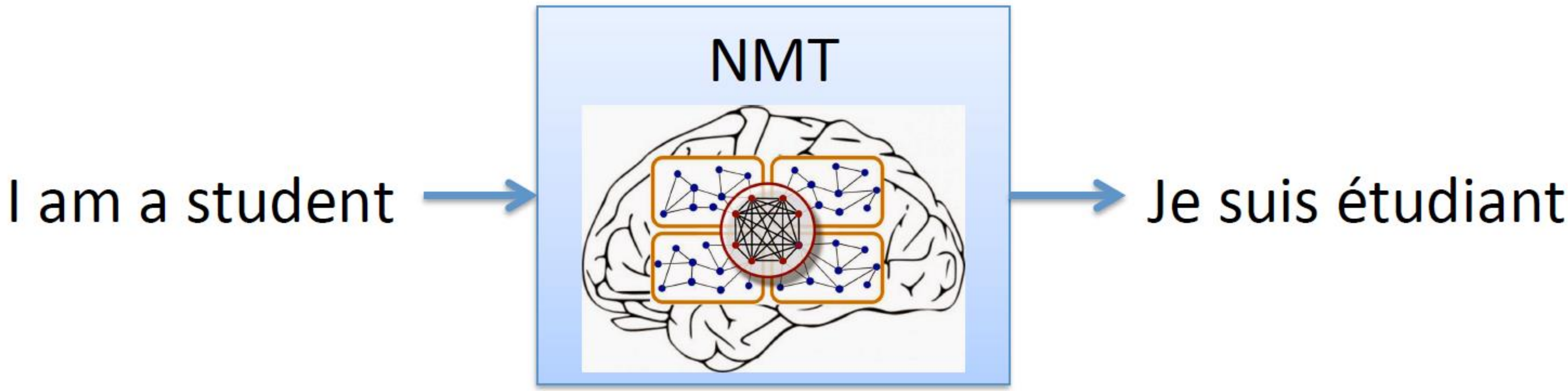
- where nSents() is the number of sentences (biases toward many alignments of single sentences instead of aligning very large spans).
- the denominator helps to normalize the similarities, so x_1, ..., x_S, y_1, ..., y_S are randomly selected sentences sampled from the respective documents.

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)



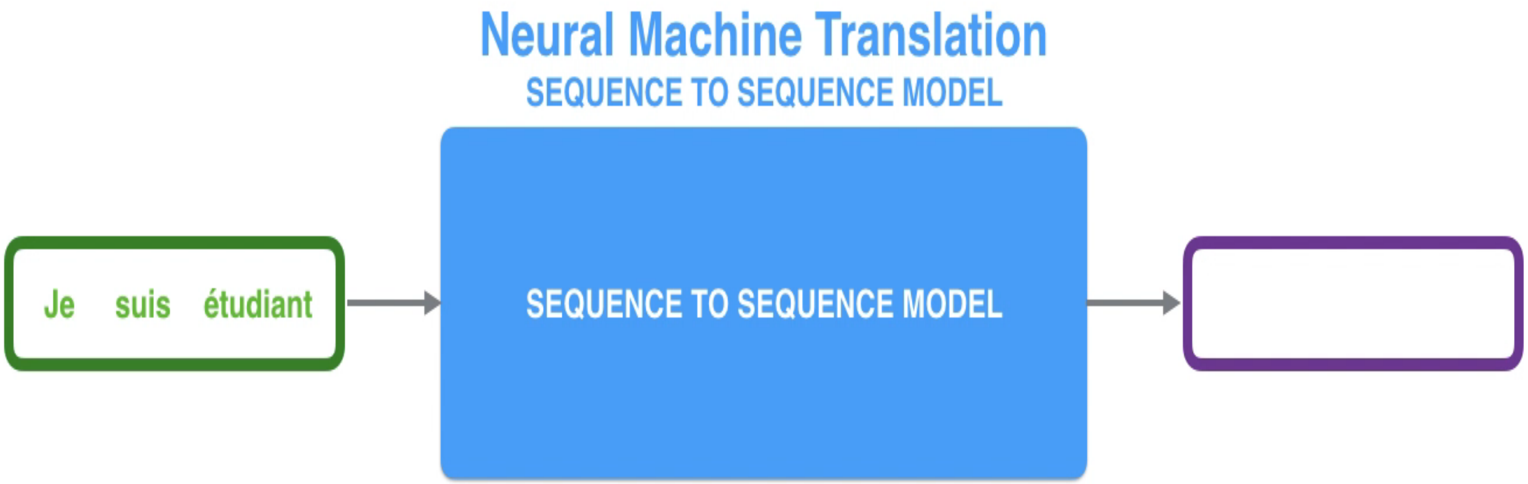
(Sutskever et al., 2014; Cho et al., 2014)

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

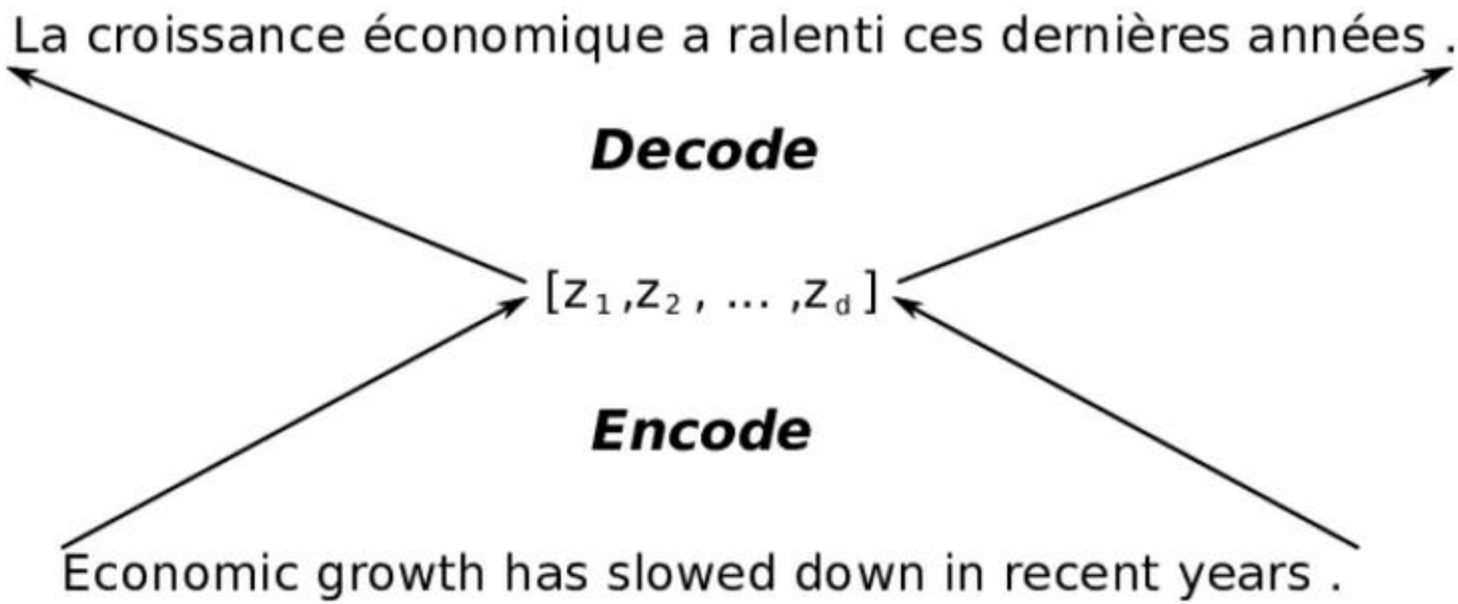
Seq2Seq model

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

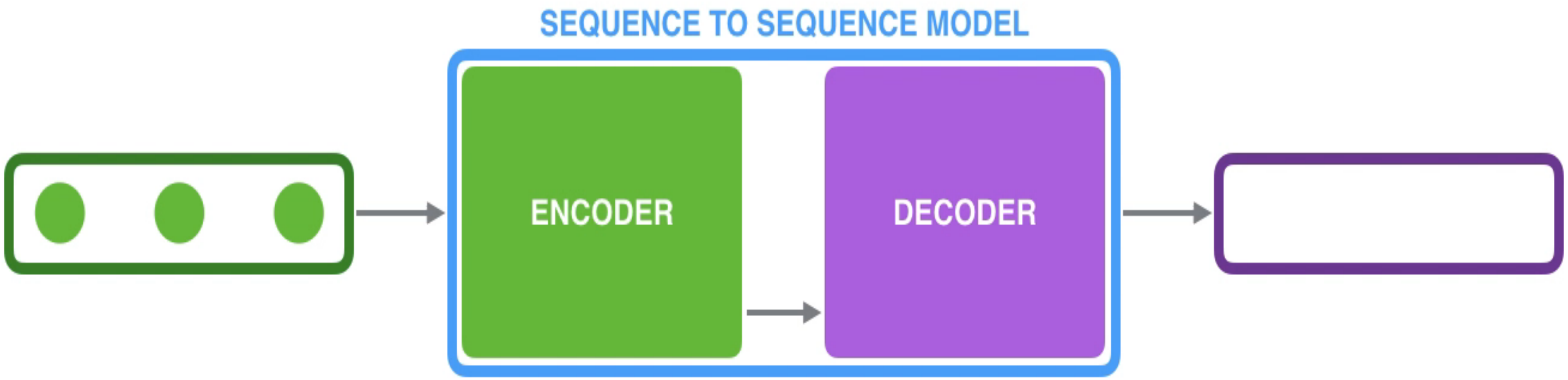
Seq2Seq for NMT



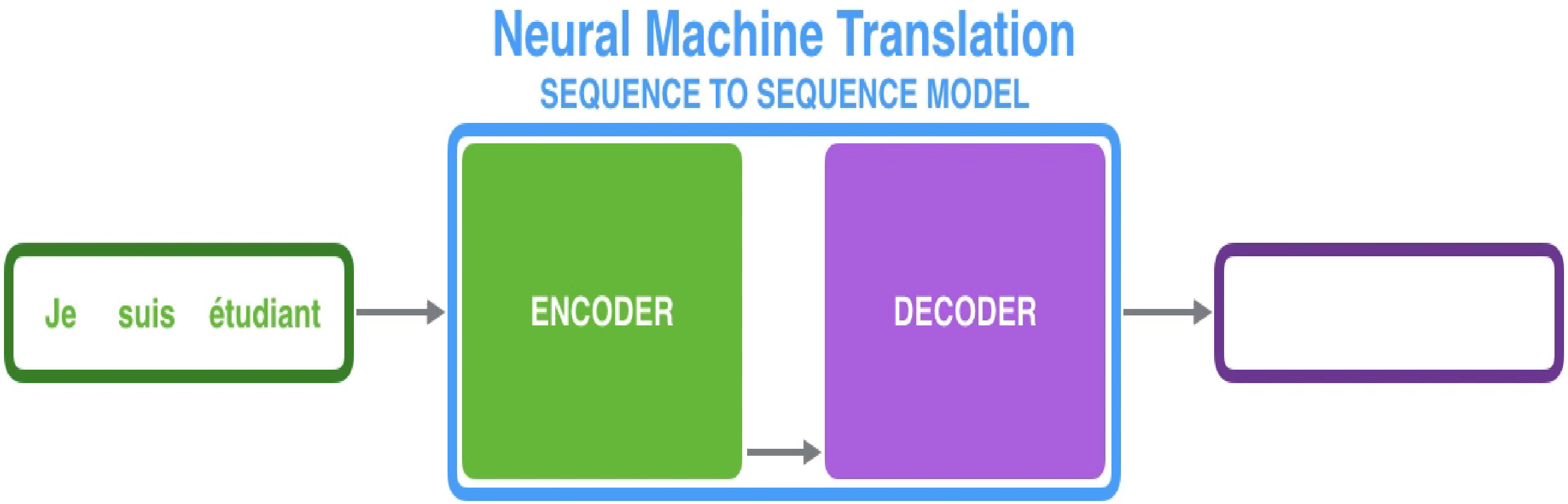
Encoder-Decoder Model

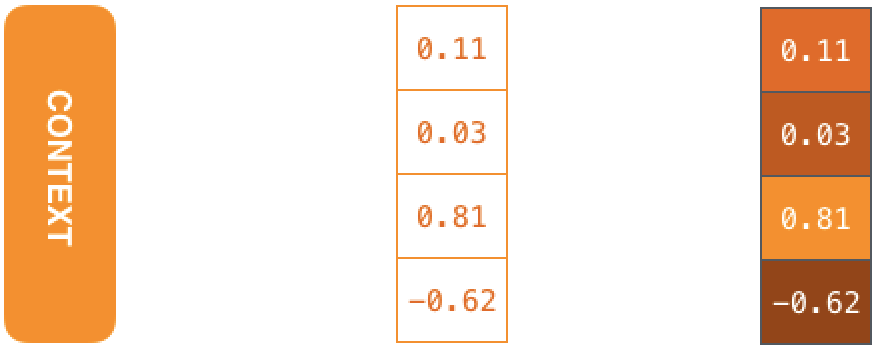


Encoder-decoder for sequences



Encoder-decoder for NMT





Seq2seq NMT

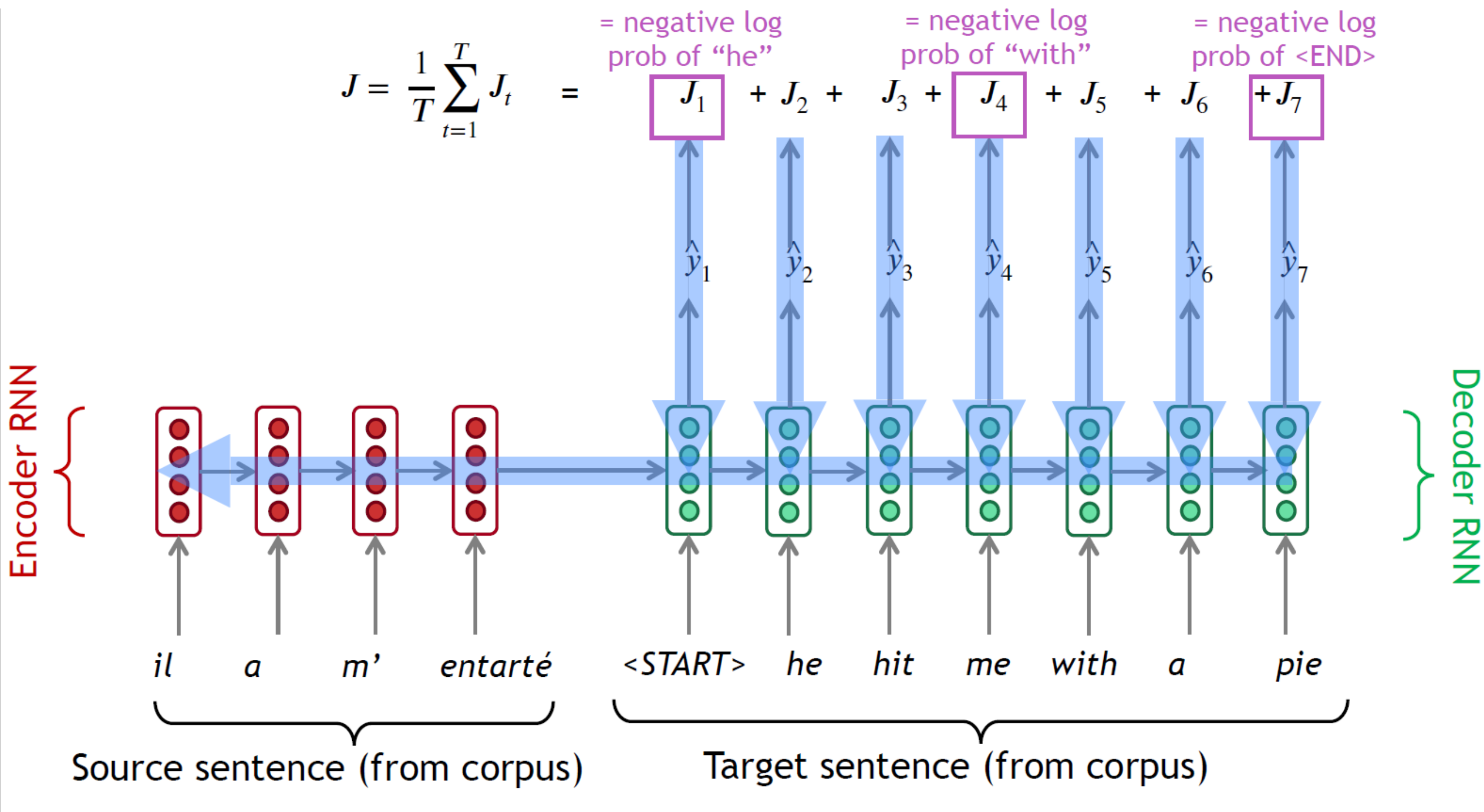
- The sequence-to-sequence model is an example of a **Conditional Language Model**.
  - **Language Model** because the decoder is predicting the next word of the target sentence *y*
  - **Conditional** because its predictions are *also* conditioned on the source sentence *x*

- 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…

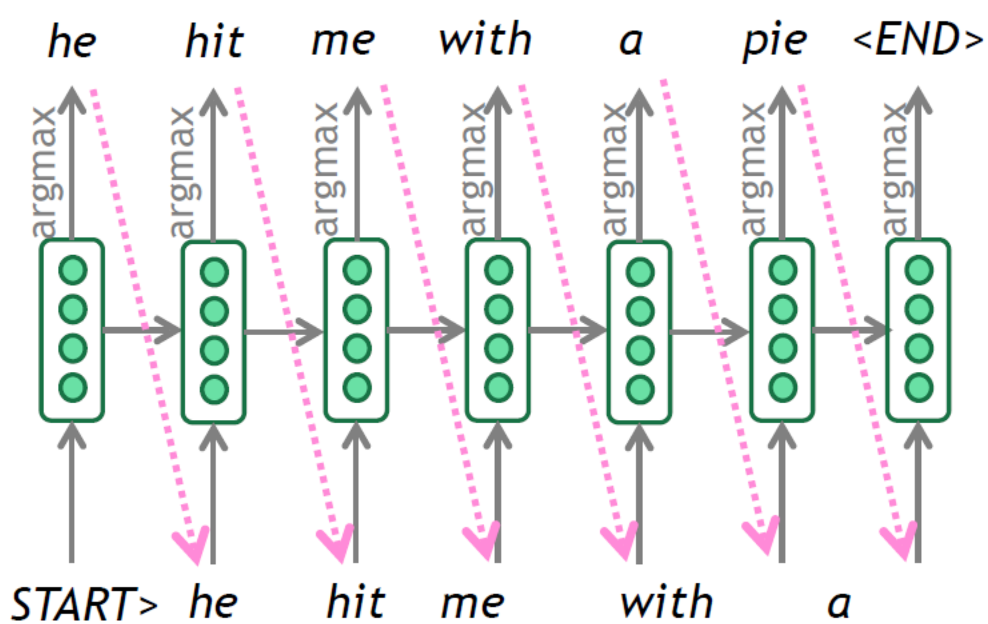
Training NMT



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?

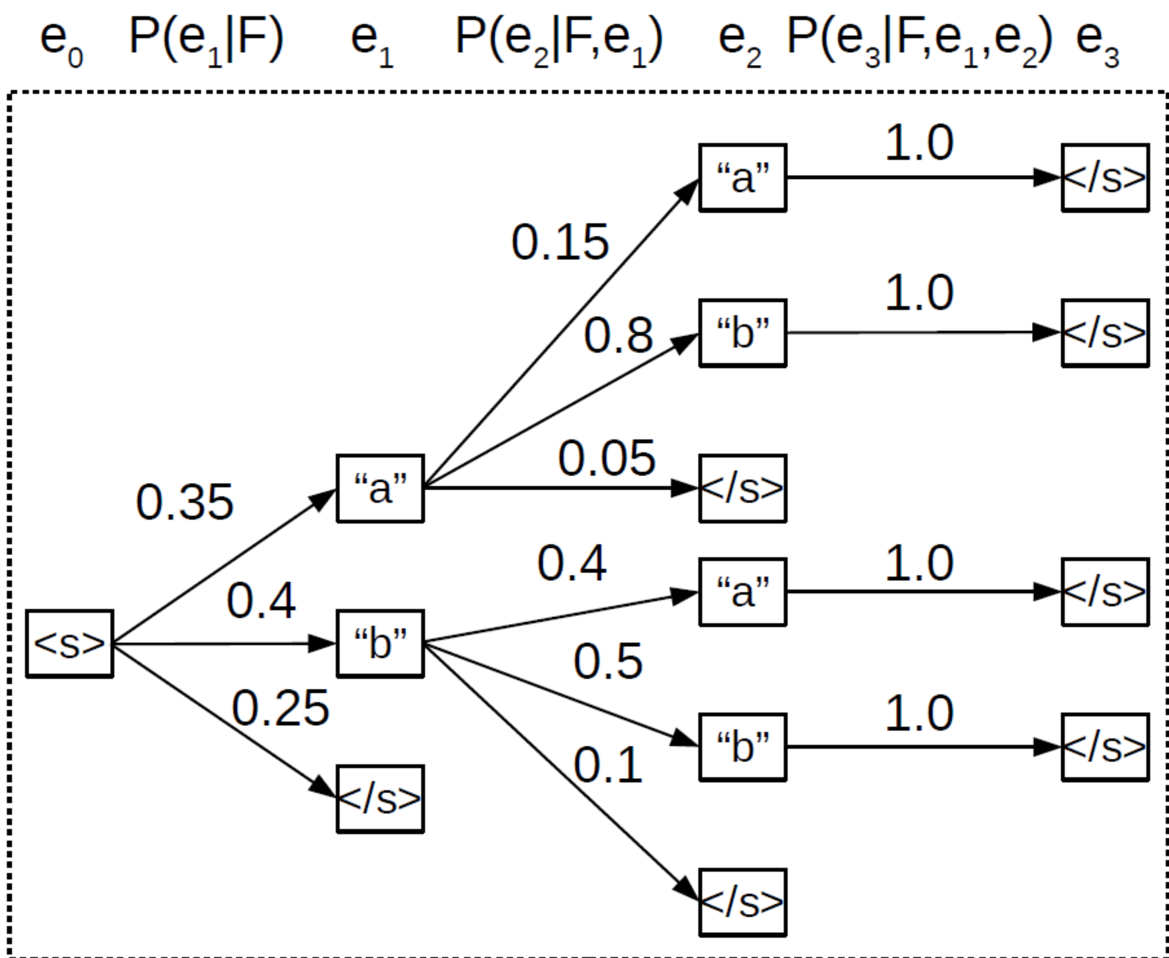


Problems with greedy decoding

- 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

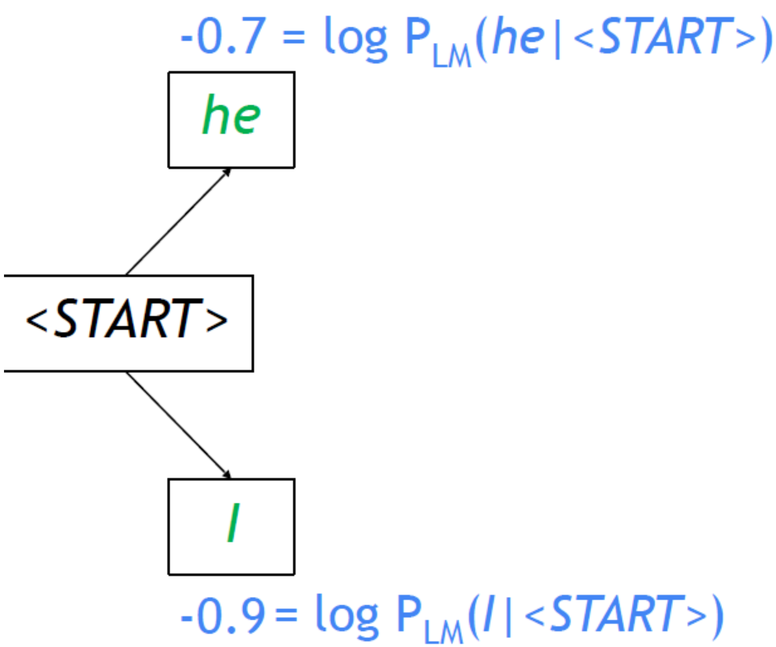
Beam size = k = 2. Blue numbers = $\text{score}(y_1, \ldots, y_t) = \sum_{i=1}^{t} \log P_{\text{LM}}(y_i | y_1, \ldots, y_{i-1}, x)$

<START>

Calculate prob dist of next word

Beam search decoding: example

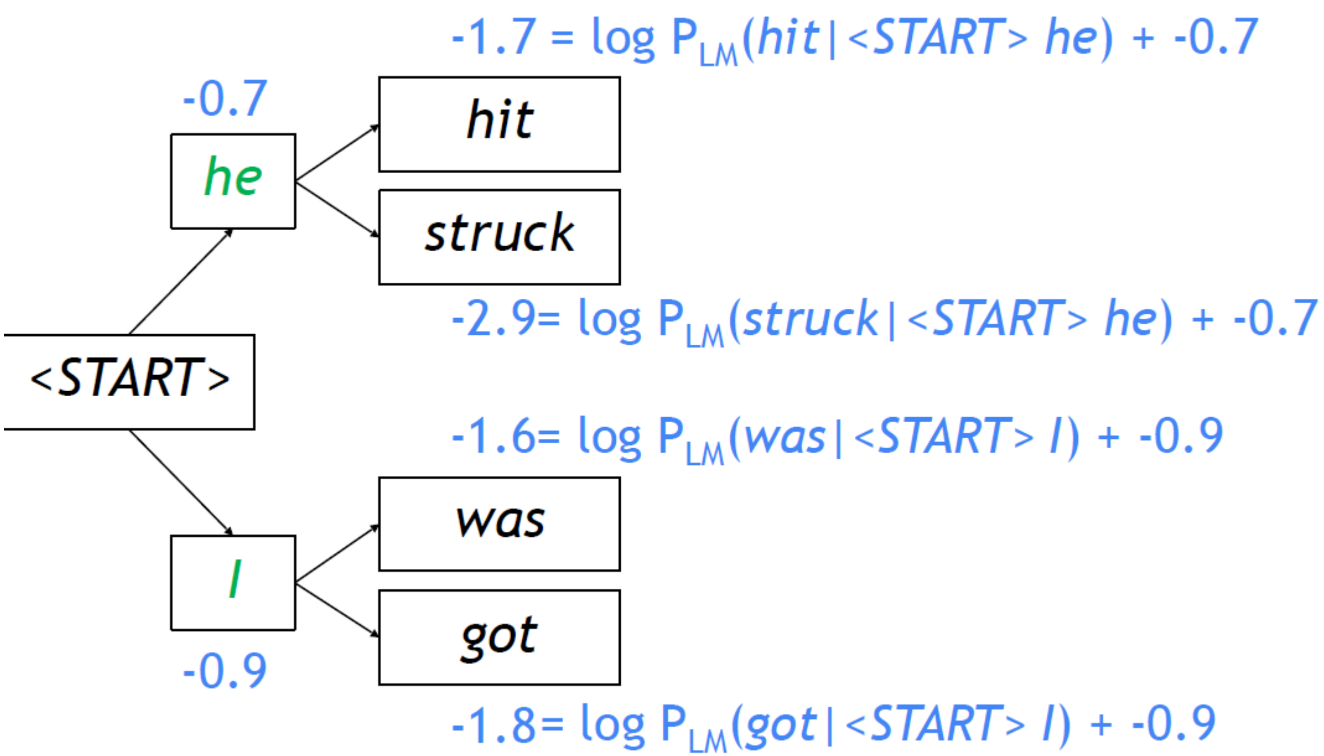
Beam size = k = 2. Blue numbers = $\text{score}(y_1, \ldots, y_t) = \sum_{i=1}^{t} \log P_{\text{LM}}(y_i | y_1, \ldots, y_{i-1}, x)$



Take top *k* words
and compute scores

Beam search decoding: example

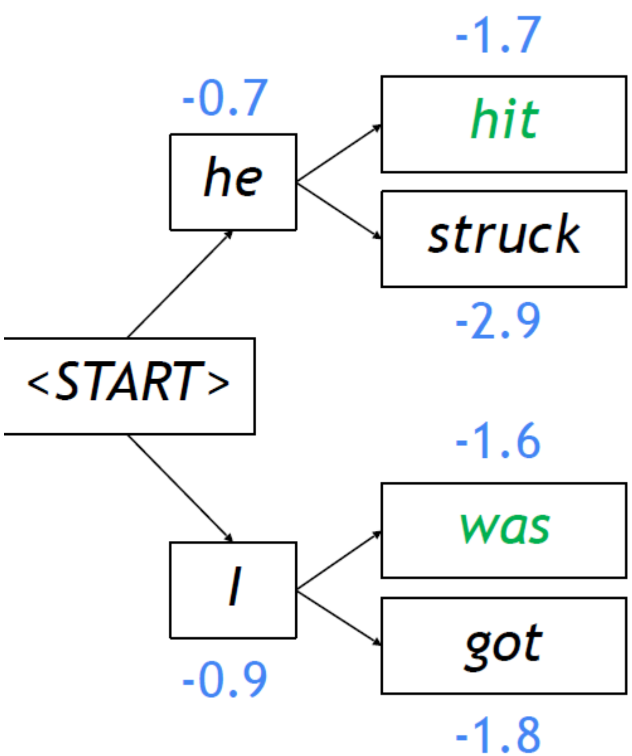
Beam size = k = 2. Blue numbers = $\text{score}(y_1, \ldots, y_t) = \sum_{i=1}^{t} \log P_{\text{LM}}(y_i | y_1, \ldots, y_{i-1}, x)$



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

Beam search decoding: example

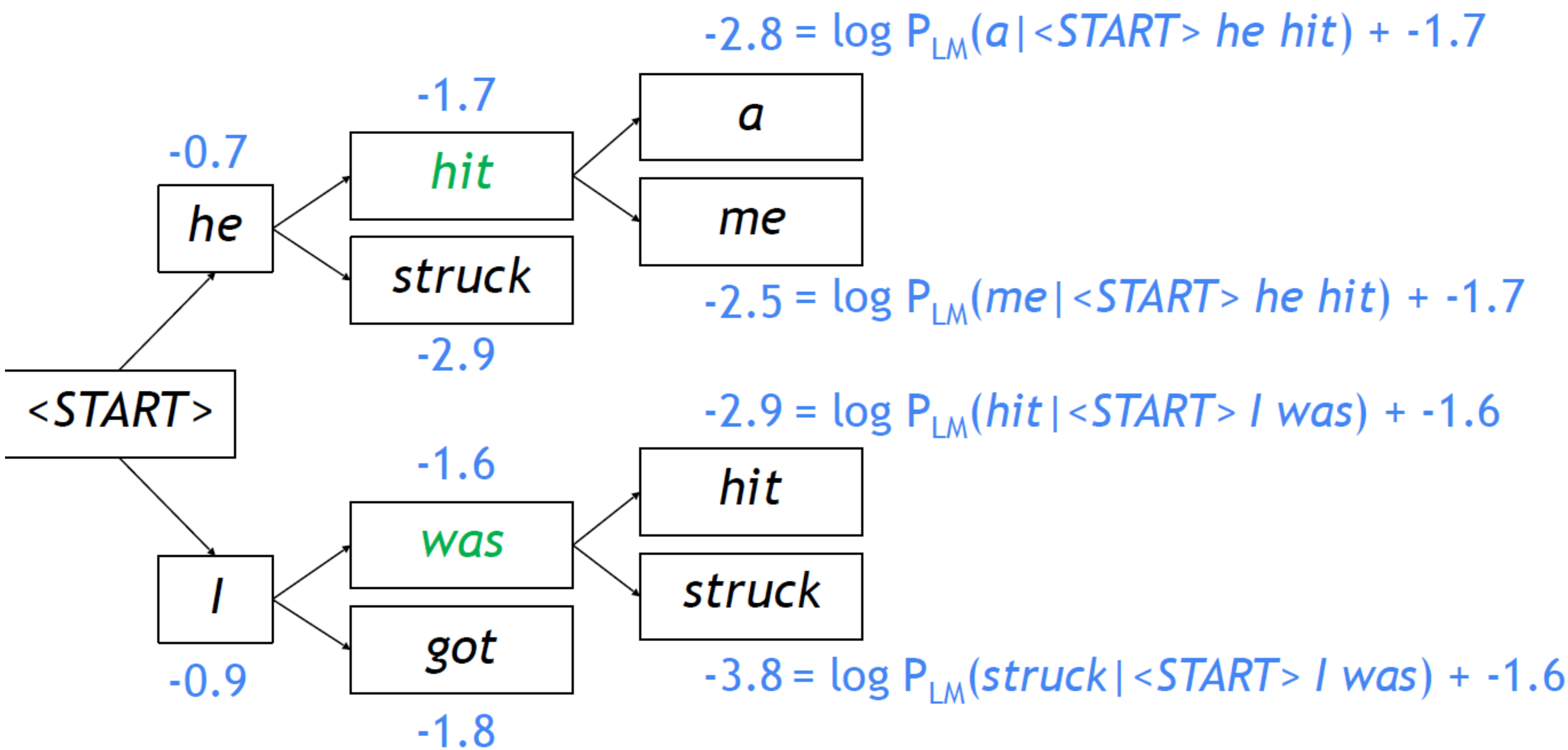
Beam size = k = 2. Blue numbers = $\text{score}(y_1, \ldots, y_t) = \sum_{i=1}^{t} \log P_{\text{LM}}(y_i | y_1, \ldots, y_{i-1}, x)$



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

Beam search decoding: example

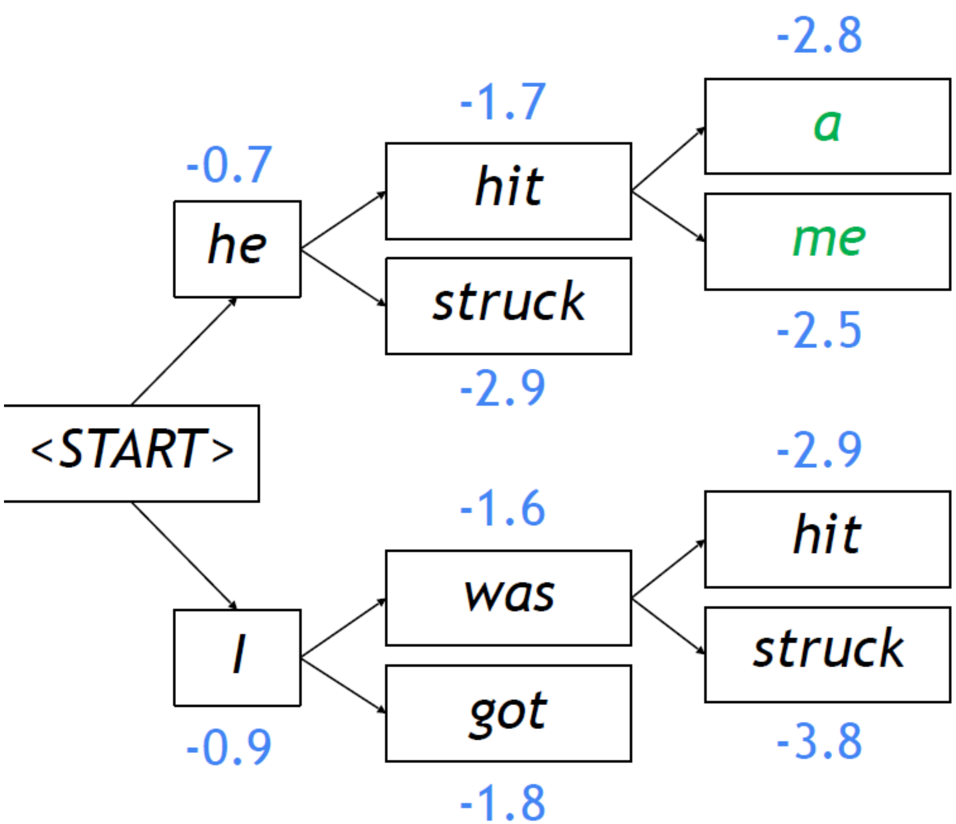
Beam size = k = 2. Blue numbers = $\text{score}(y_1, \ldots, y_t) = \sum_{i=1}^{t} \log P_{\text{LM}}(y_i | y_1, \ldots, y_{i-1}, x)$



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

Beam search decoding: example

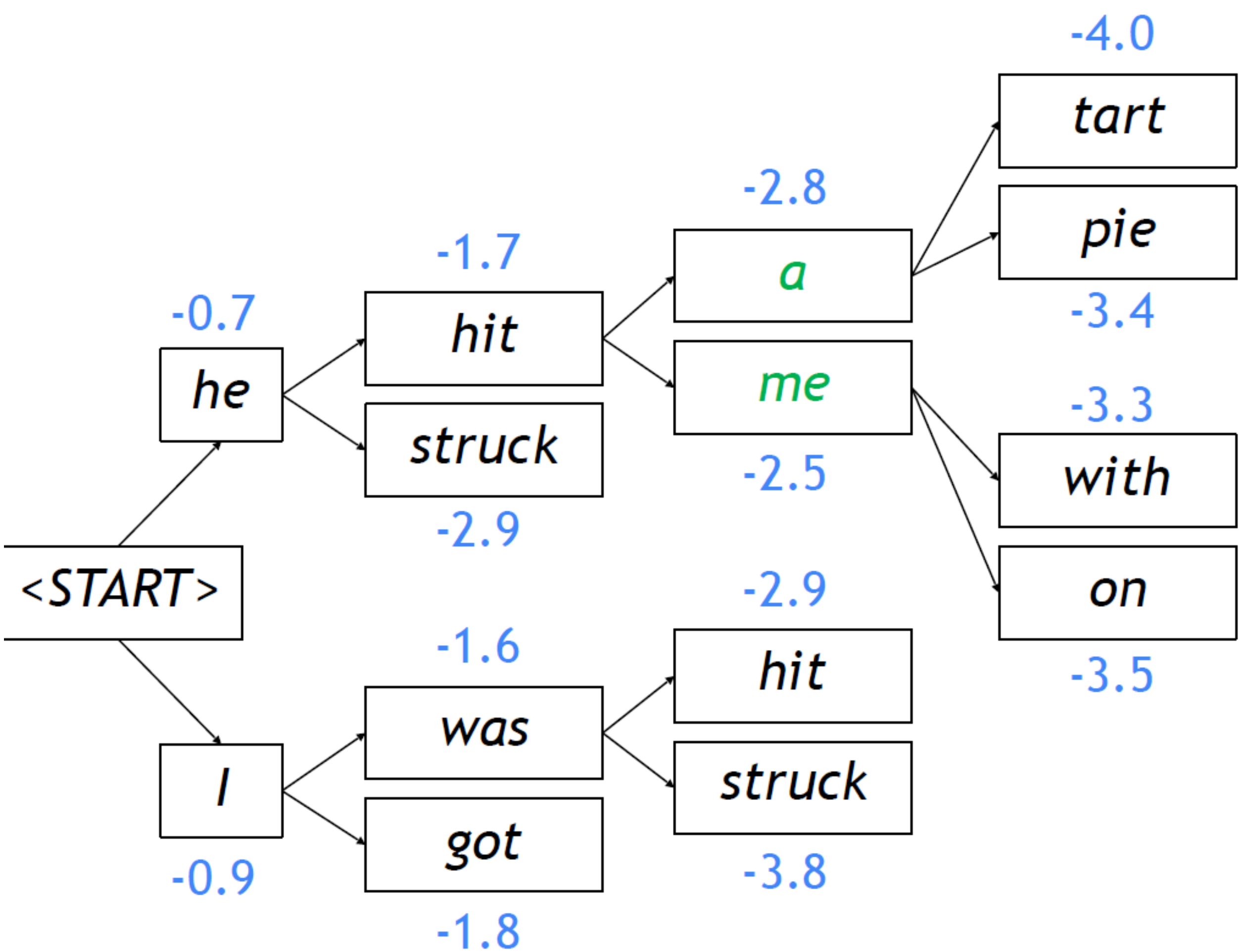
Beam size = k = 2. Blue numbers = $\text{score}(y_1, \ldots, y_t) = \sum_{i=1}^{t} \log P_{\text{LM}}(y_i | y_1, \ldots, y_{i-1}, x)$



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

Beam search decoding: example

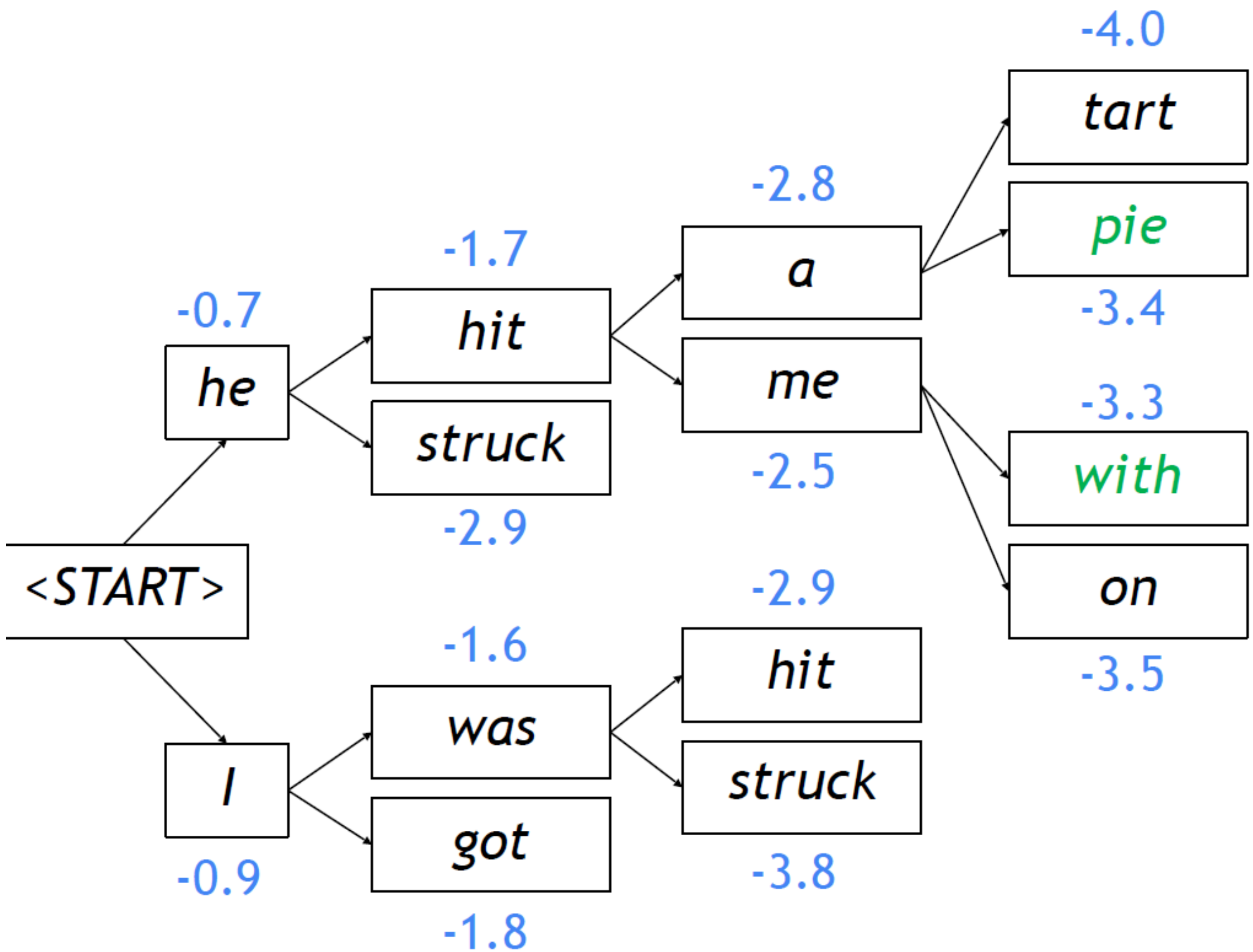
Beam size = k = 2. Blue numbers = $\text{score}(y_1, \ldots, y_t) = \sum_{i=1}^{t} \log P_{\text{LM}}(y_i | y_1, \ldots, y_{i-1}, x)$



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

Beam search decoding: example

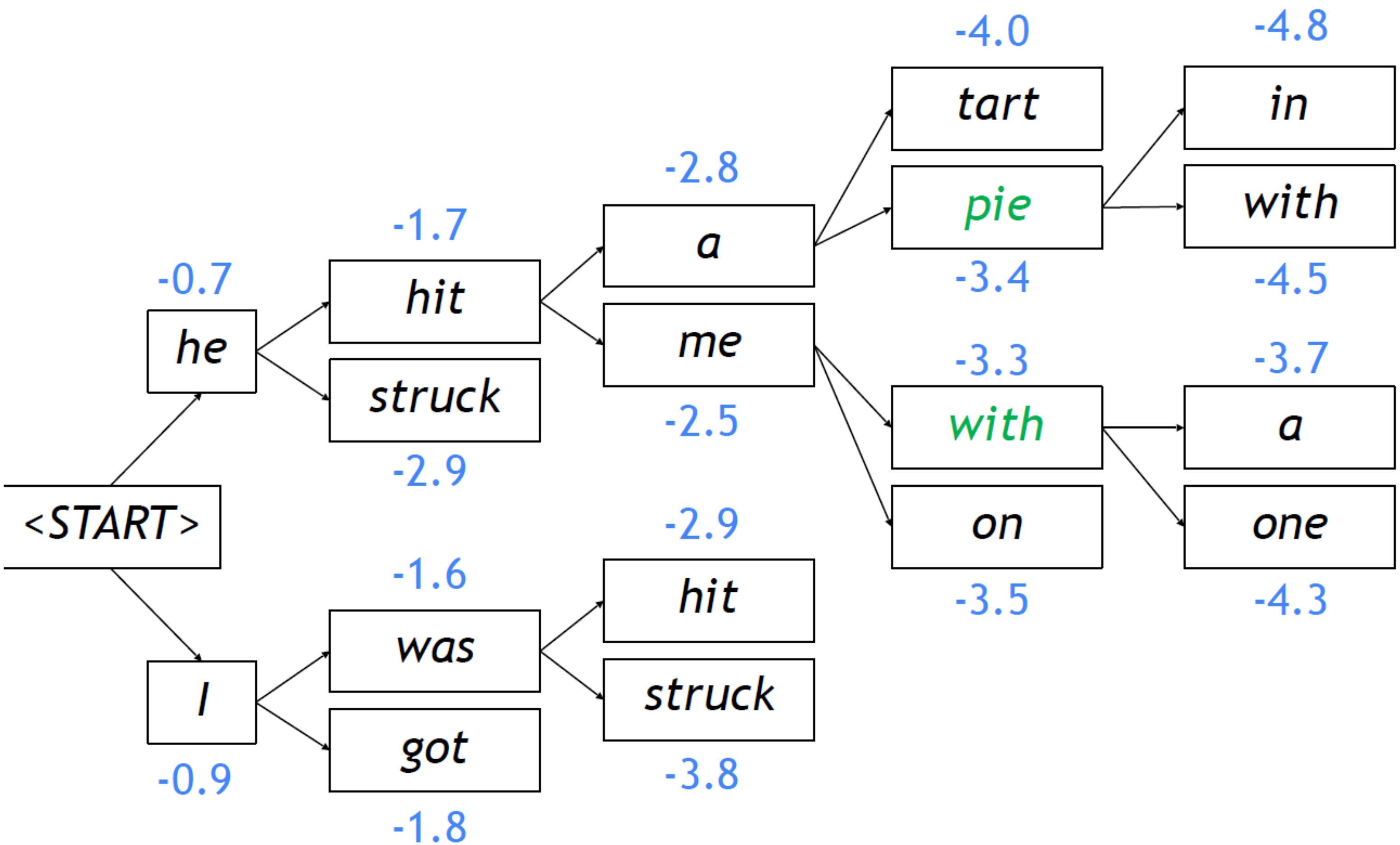
Beam size = k = 2. Blue numbers = $\text{score}(y_1, \ldots, y_t) = \sum_{i=1}^{t} \log P_{\text{LM}}(y_i | y_1, \ldots, y_{i-1}, x)$



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

Beam search decoding: example

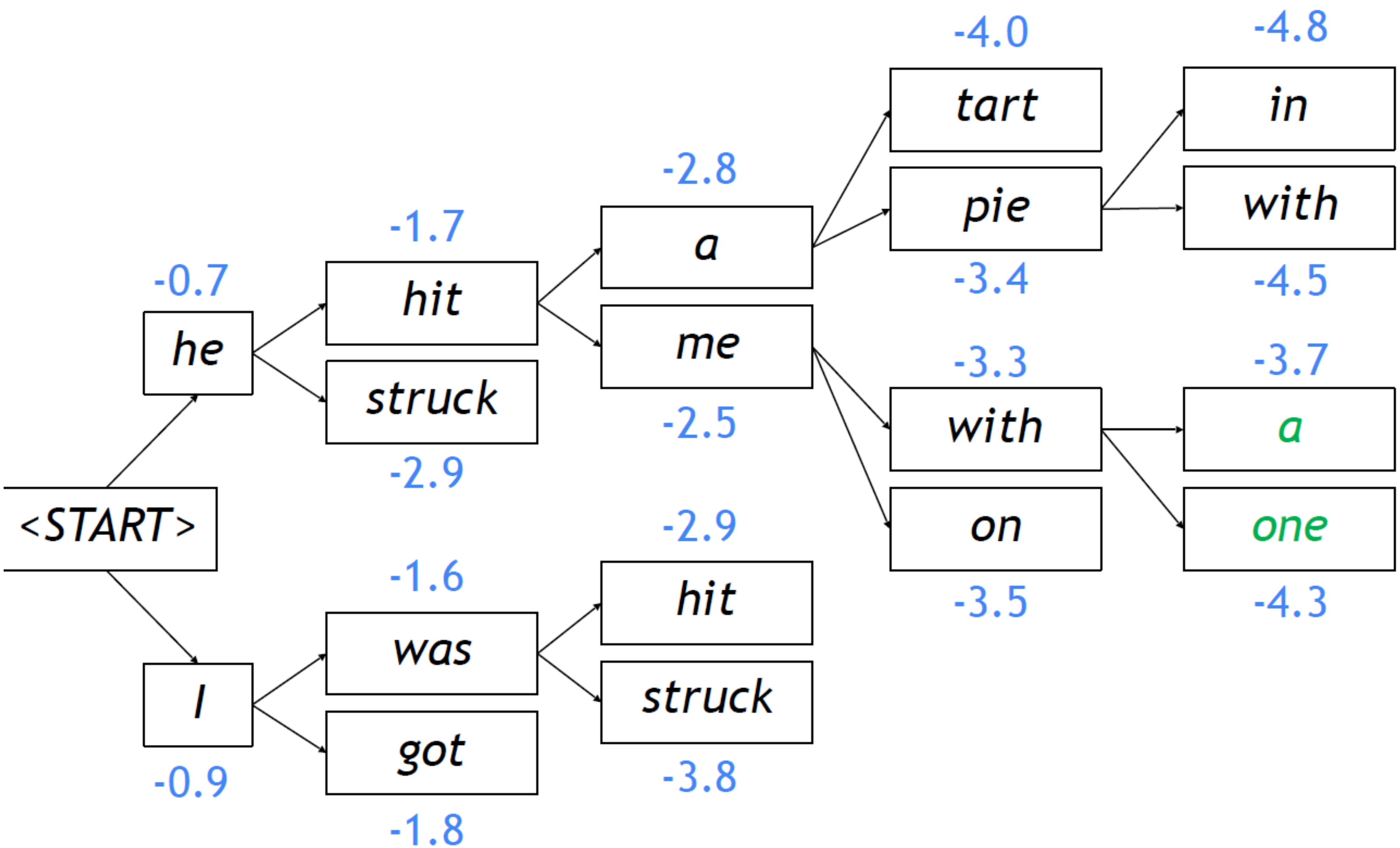
Beam size = k = 2. Blue numbers = $\text{score}(y_1, \ldots, y_t) = \sum_{i=1}^{t} \log P_{\text{LM}}(y_i | y_1, \ldots, y_{i-1}, x)$



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

Beam search decoding: example

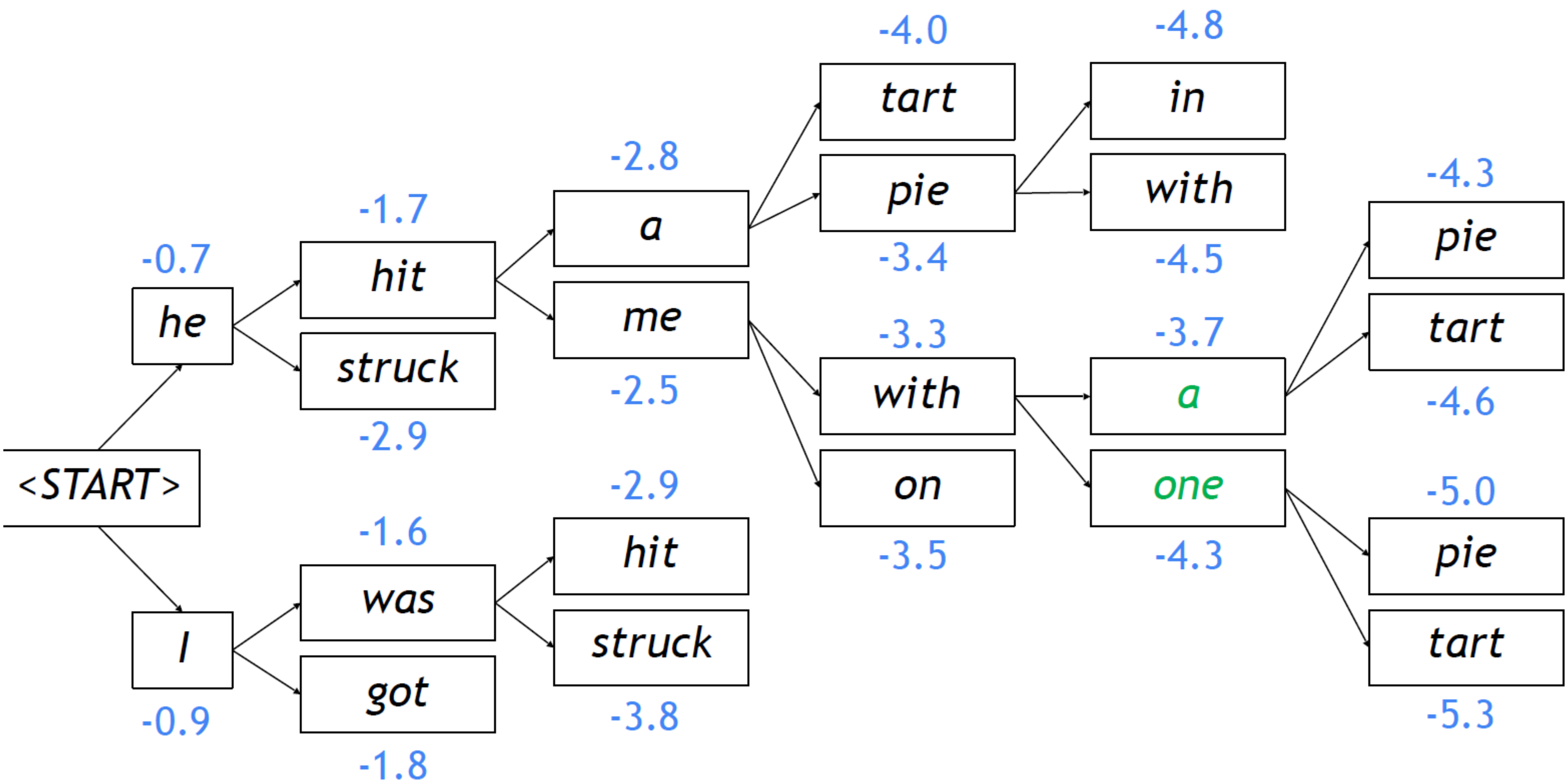
Beam size = k = 2. Blue numbers = $\text{score}(y_1, \ldots, y_t) = \sum_{i=1}^{t} \log P_{\text{LM}}(y_i | y_1, \ldots, y_{i-1}, x)$



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

Beam search decoding: example

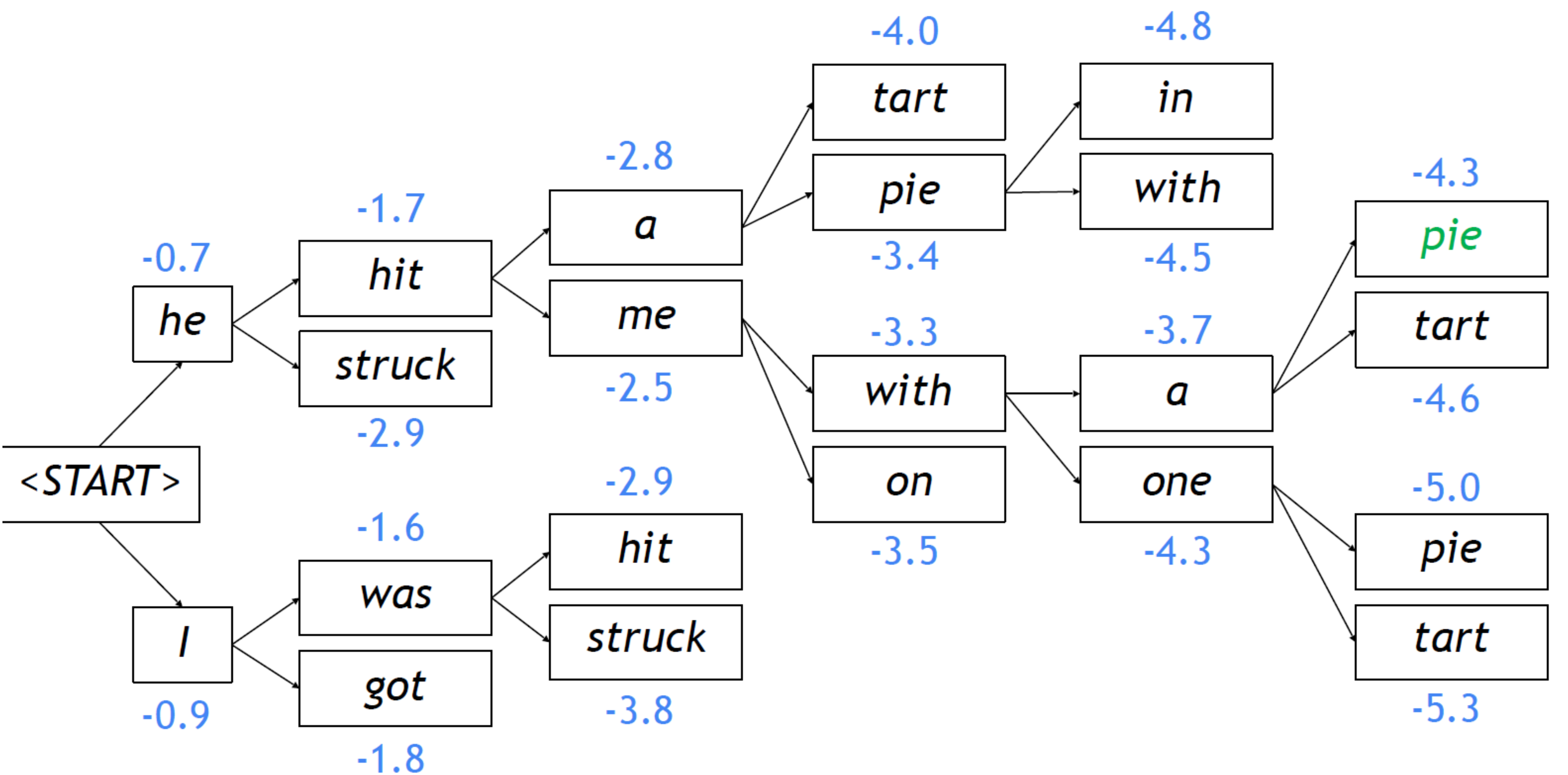
Beam size = k = 2. Blue numbers = $\text{score}(y_1, \ldots, y_t) = \sum_{i=1}^{t} \log P_{\text{LM}}(y_i | y_1, \ldots, y_{i-1}, x)$



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

Beam search decoding: example

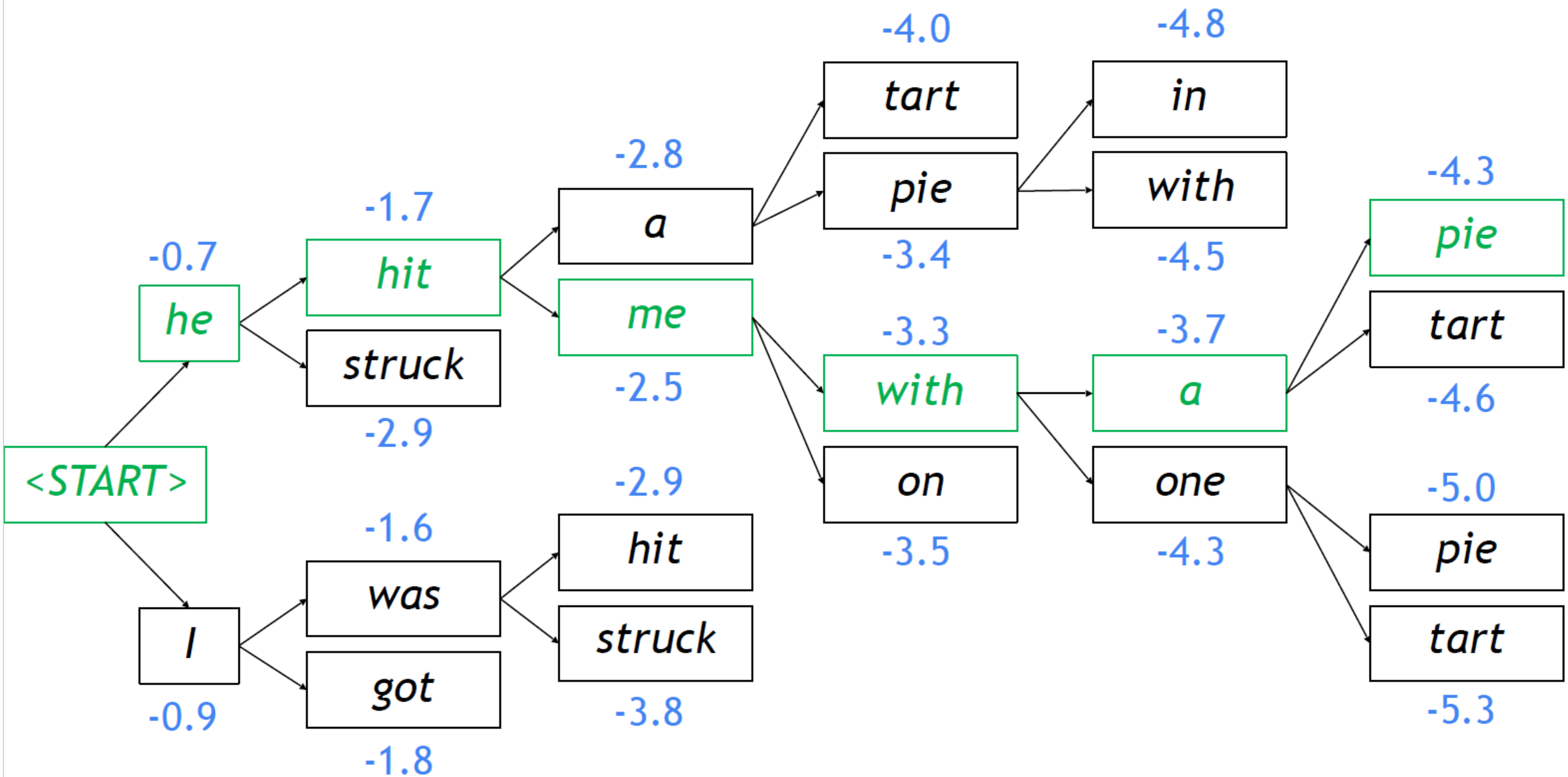
Beam size = k = 2. Blue numbers = $\text{score}(y_1, \ldots, y_t) = \sum_{i=1}^{t} \log P_{\text{LM}}(y_i | y_1, \ldots, y_{i-1}, x)$



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

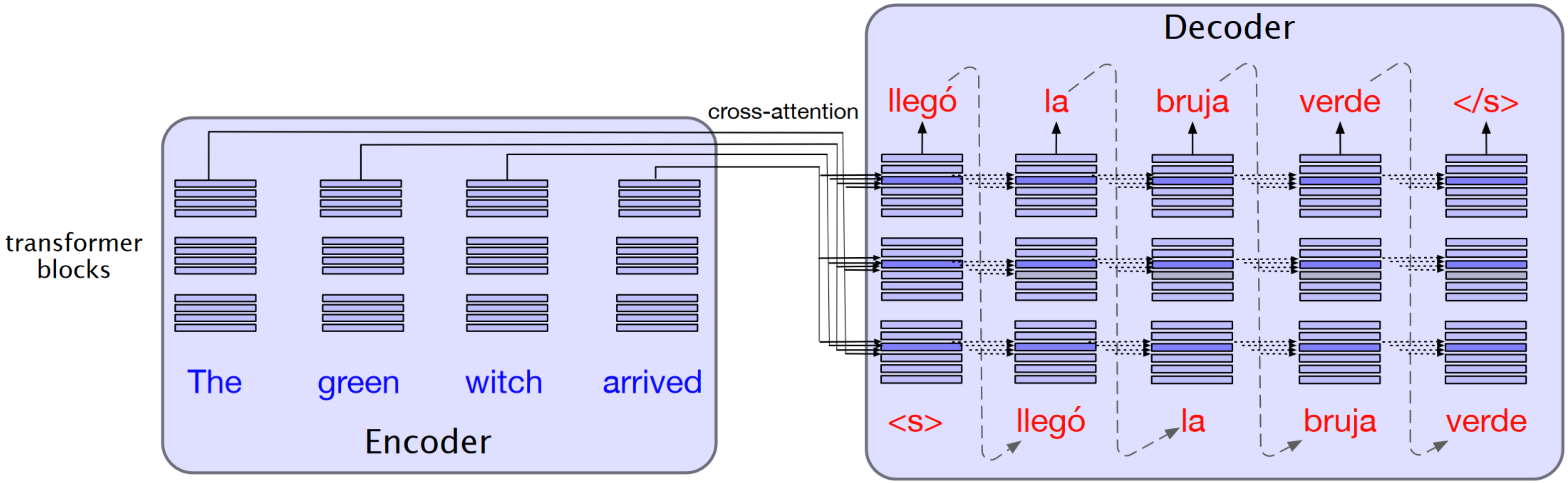
Human chit-chat partner

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?*

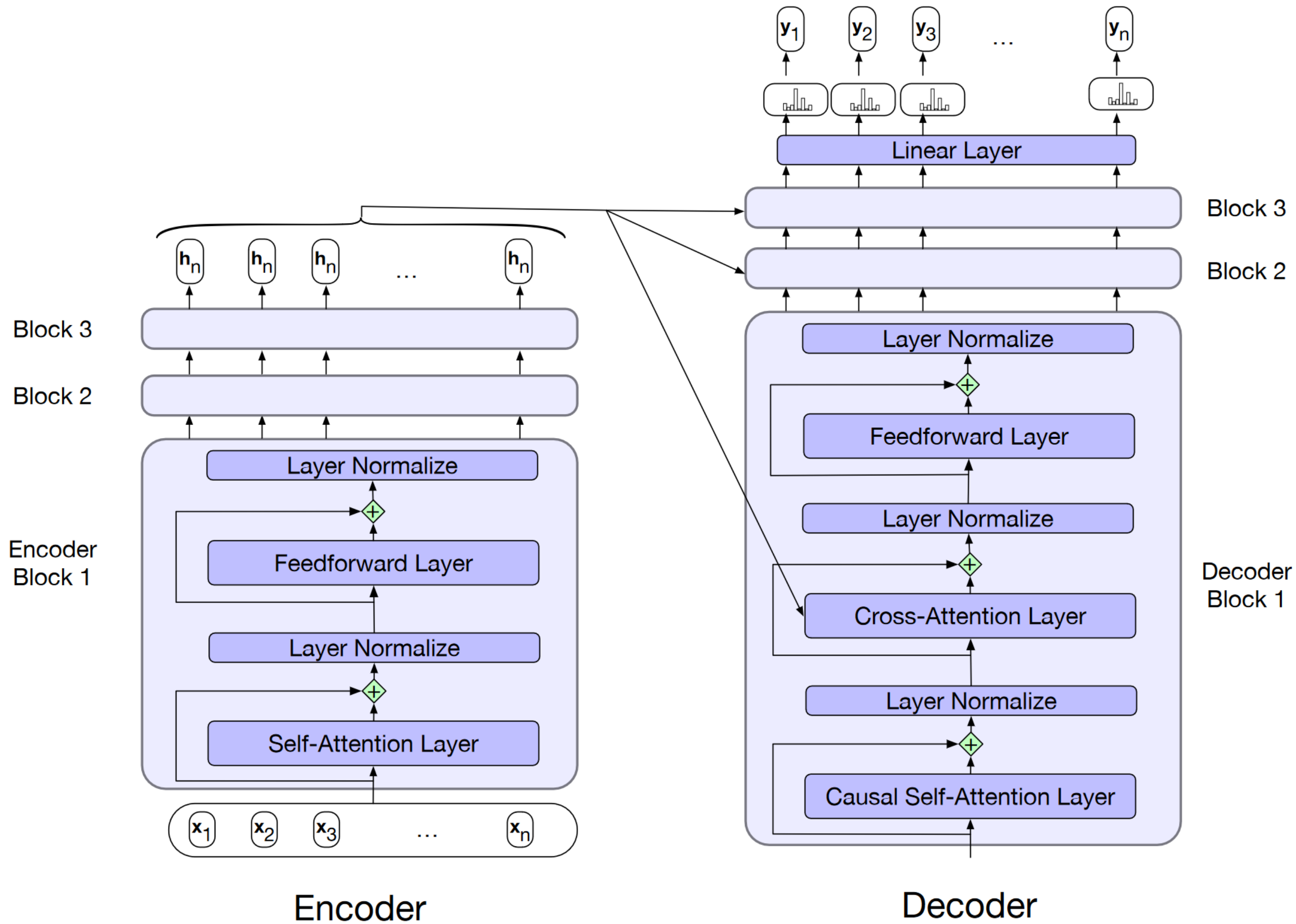
Low beam size:
More on-topic but nonsensical;
bad English

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

Transformer is encoder-decoder



Attention in transformer



The final output of the encoder $H_{enc} = h_1, ..., h_T$ 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

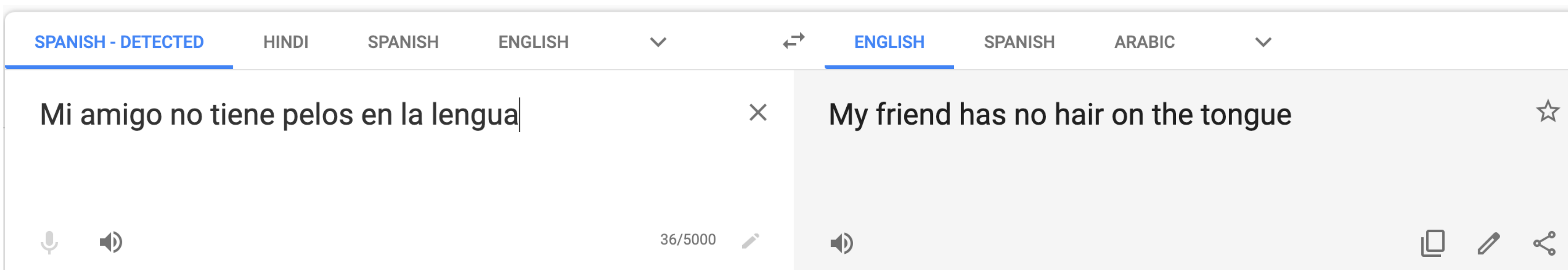
- Compared to SMT, NMT has many advantages:
  - Better performance
  - More fluent
  - Better use of context
  - Better use of phrase similarities
- A single neural network to be optimized end-to-end
  - No subcomponents to be individually optimized
- Requires much less human engineering effort
  - No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

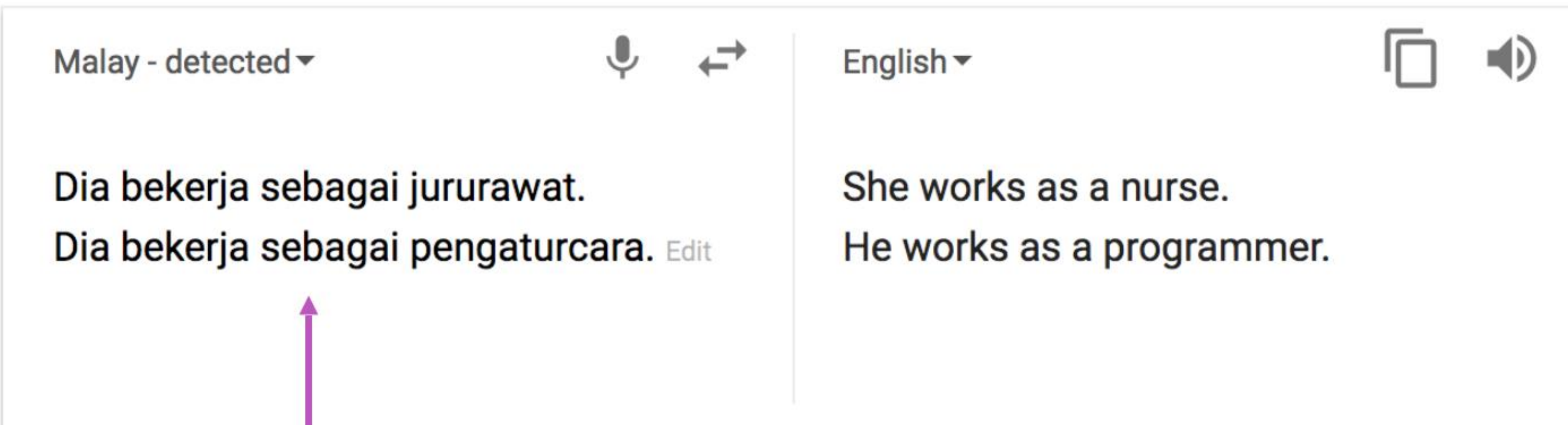
So is Machine Translation solved?

- 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



So is Machine Translation solved?

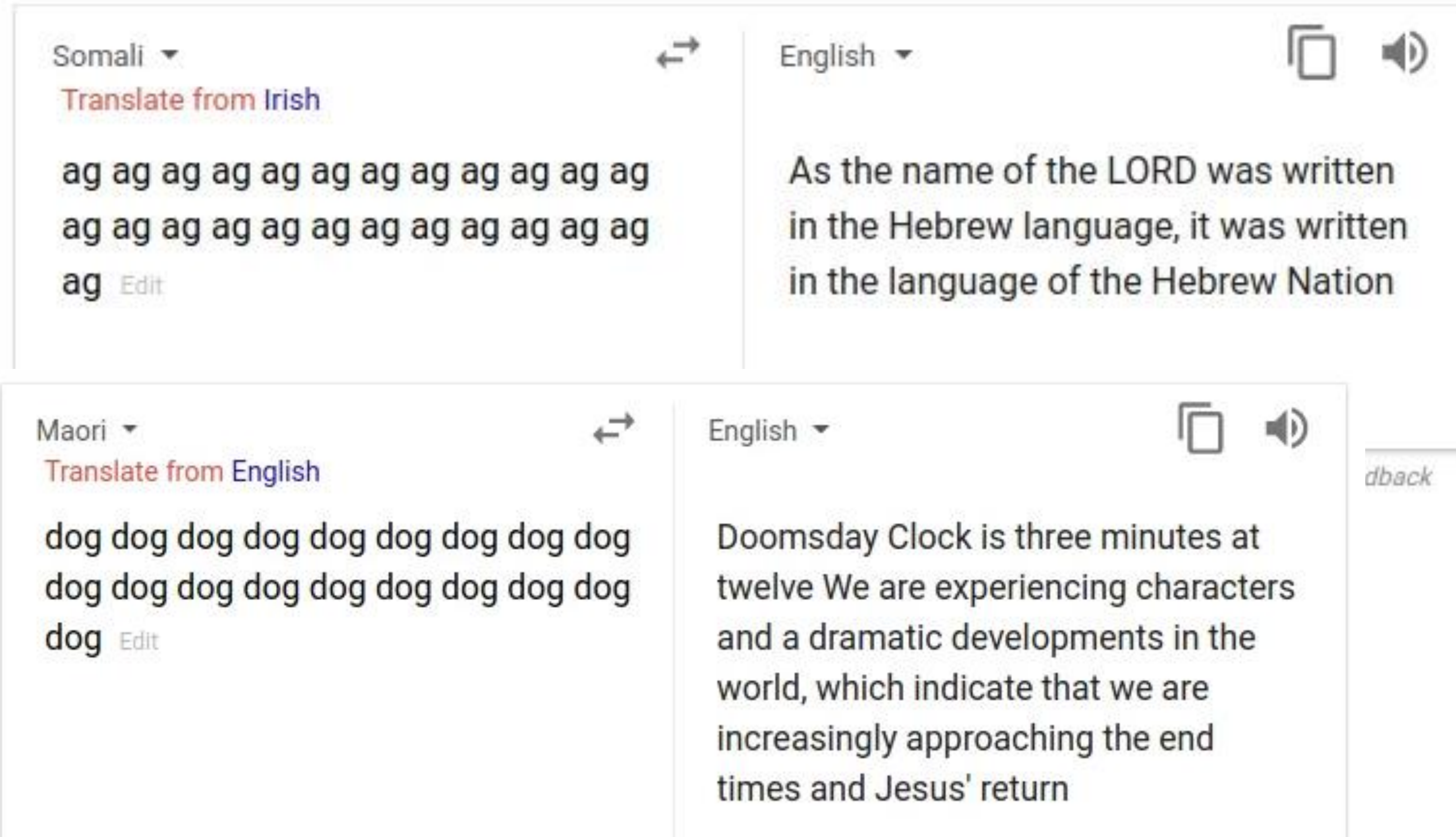
- NMT picks up biases in training data



Didn’t specify gender

So is Machine Translation solved?

- Uninterpretable systems do strange things



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

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

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

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

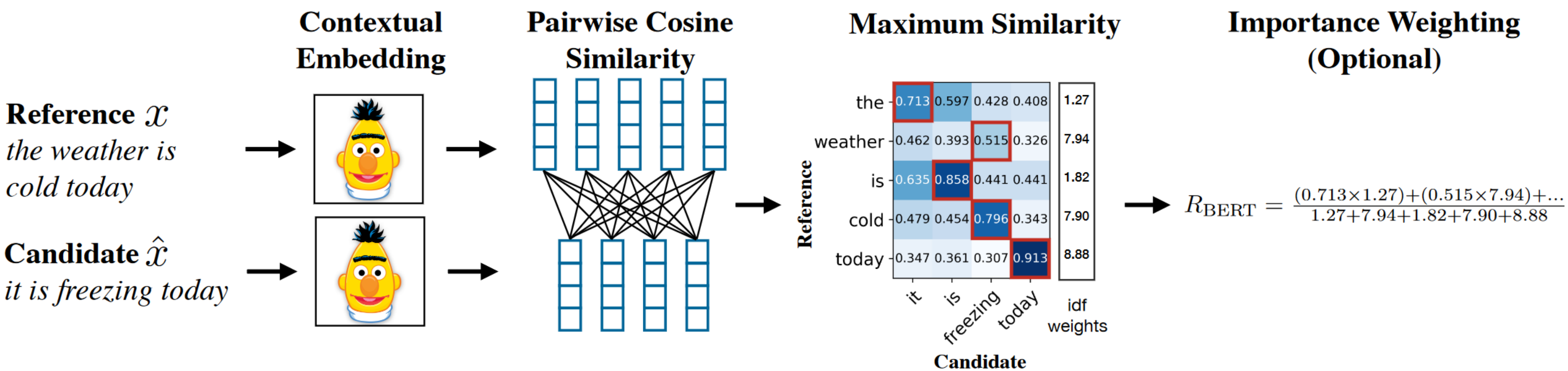
- no good measure exists on the level of sentence
- no good measure exists of an absolute translation quality

BERTScore

- for the reference x and the candidate $\tilde{x}$, compute a BERT embedding for each token x_i and $\tilde{x}_j$.
- For each pair of tokens, compute their 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

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

BERTScore illustration



Improvements in MT

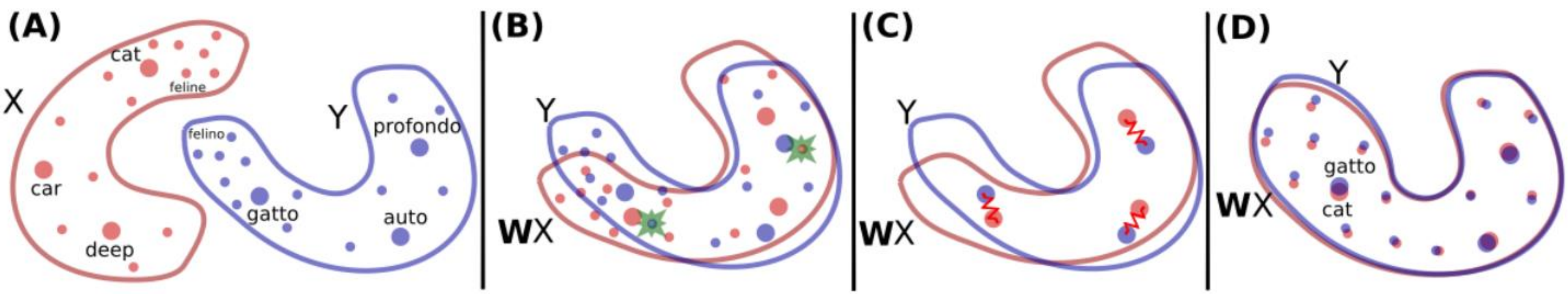
- large corpora
- adaptations to specific domains, e.g., IT, pharmacy, automotive industry
- terminological dictionaries, terminology lists, translation memories

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

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