Machine Translation

Vitaly Protasov

Introduction

Machine translation (MT) is the task of automatically translating text from one language to another.

According to The World Atlas of Language Structures (WALS), now we have 2662 alive languages.

We don’t have an ideal MT for high-resource languages, the situation for the rest ones quite worse.

Types of Machine Translation

- **Rule-based MT**

  relies on a set of linguistic rules that are designed to translate text and requires a significant amount of manual work.

- **Statistical MT**

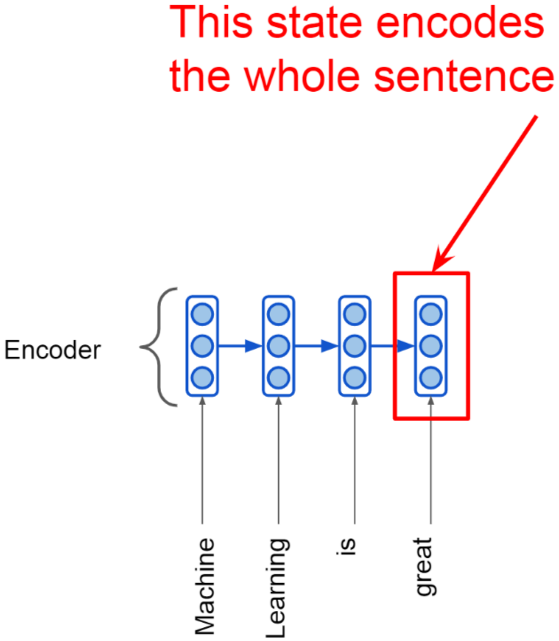
  based on the idea of training a model on a large corpus of bilingual data to identify patterns between words.

- **Neural MT**

  attempts to build and train a single, large neural network that reads a sentence and outputs a correct translation.

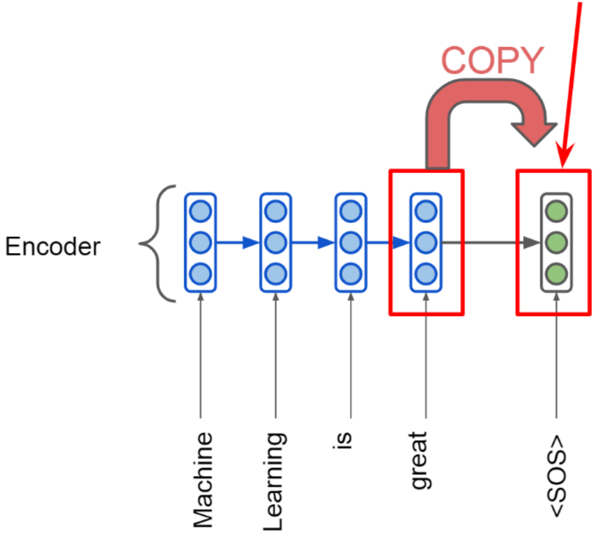
Neural Machine Translation. Seq2seq

- RNN-based



Neural Machine Translation. Seq2seq

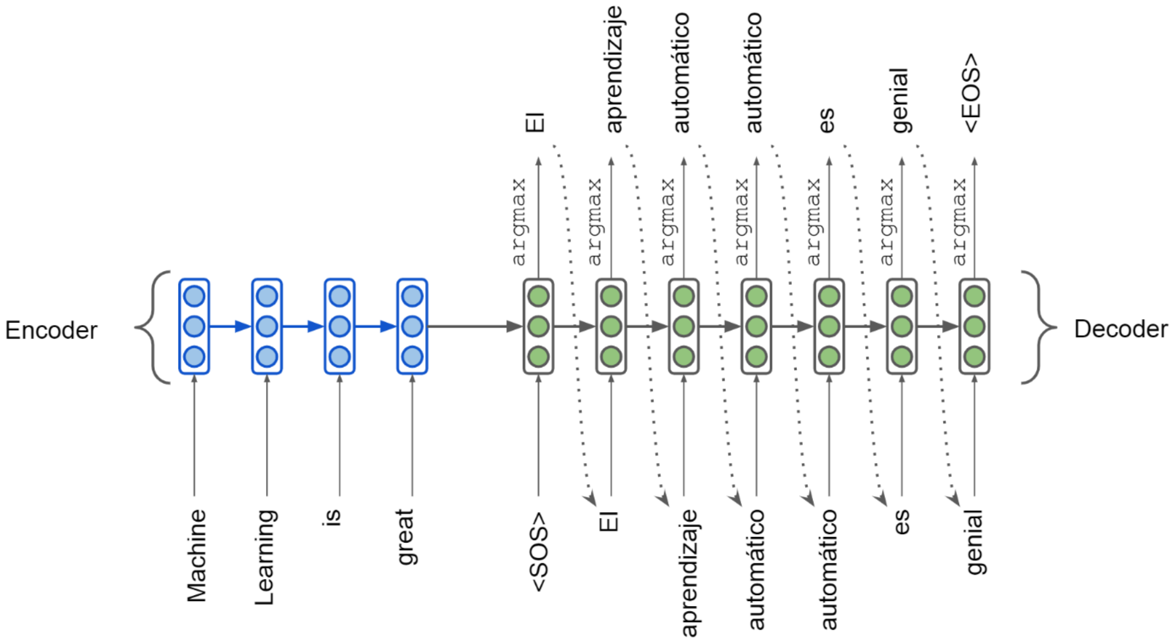
- RNN-based



Forwarded as initial hidden state to decoder

Neural Machine Translation. Seq2seq

- RNN-based



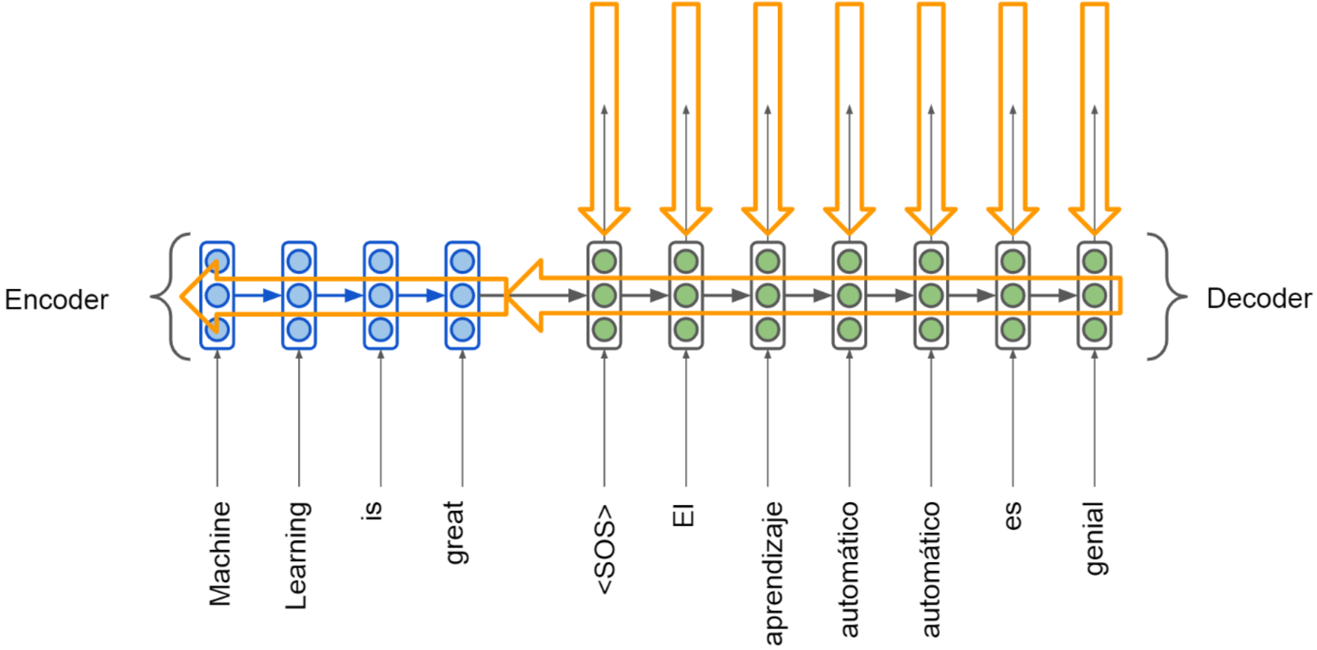
Training process

- Manually, we train to maximize the probability of the translation in the target language T, given the source sentence S, the source language l_s, and the target language l_t, i.e., $P(T|S, l_s, l_t\ )$.

$$P(T|S) = P(t_2|\underset{\overset{\uparrow}{\text{<sos>}}}{t_1}, \mathrm{S})P(t_3|t_2, t_1, \mathrm{S}) \ldots P(\underset{\overset{\uparrow}{\text{<eos>}}}{t_{last}}|t_1, t_2, \ldots, \mathrm{S})$$

- From a probabilistic perspective, translation is equivalent to finding a target sentence T that maximizes the conditional probability of T given a source sentence S, i.e., $argmax_T P(T|S)$

Training process

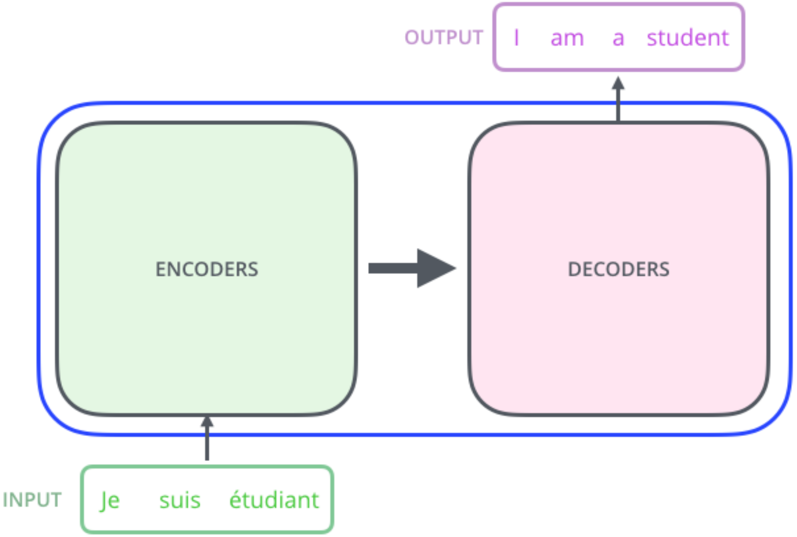


Loss functions:

- Negative log-likelihood
- Cross Entropy
- Mixed CE

Neural Machine Translation

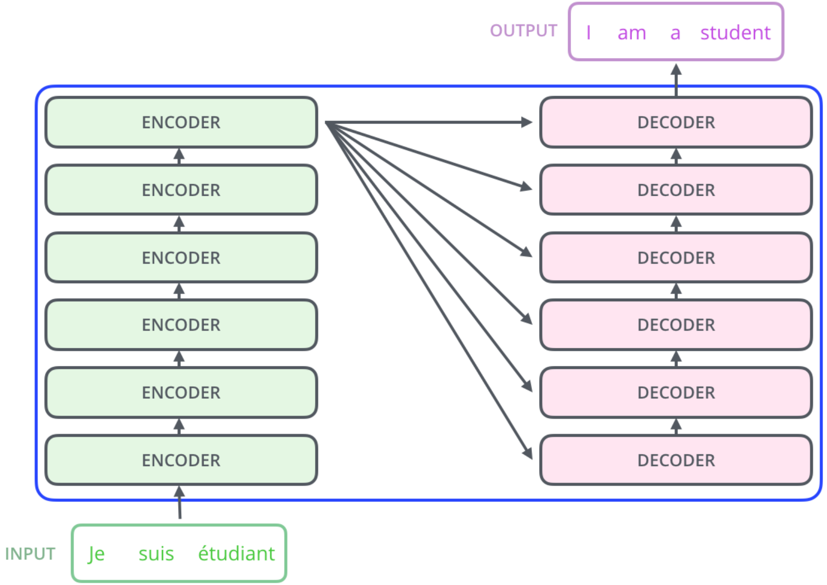
- Transformer-based



Source: https://jalammar.github.io/illustrated-transformer/

Neural Machine Translation

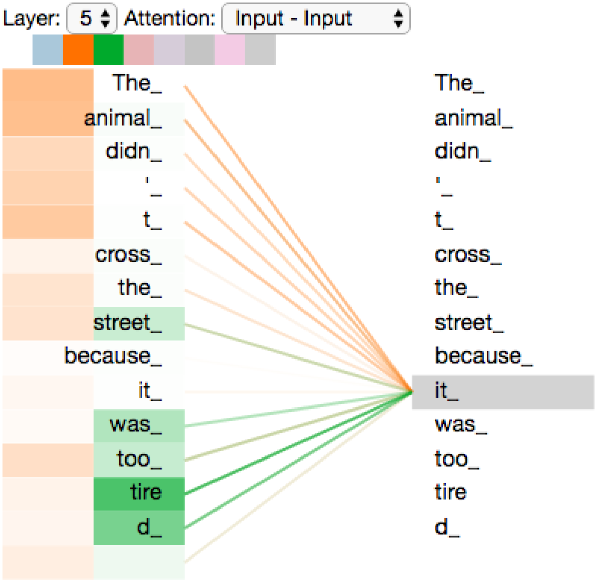
- Transformer-based



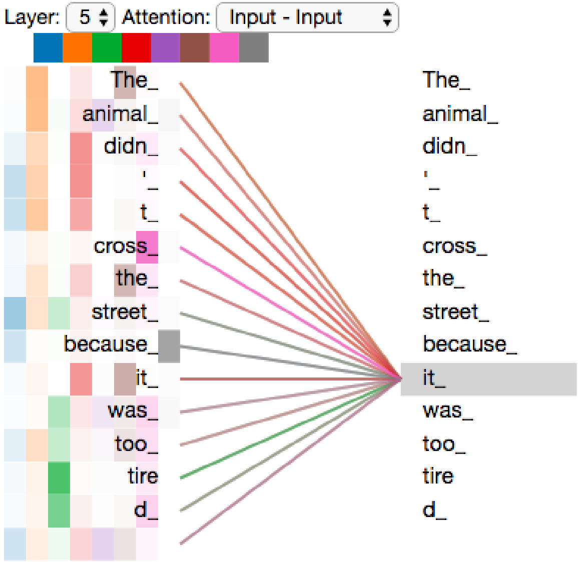
Source: https://jalammar.github.io/illustrated-transformer/

Neural Machine Translation

- Transformer-based



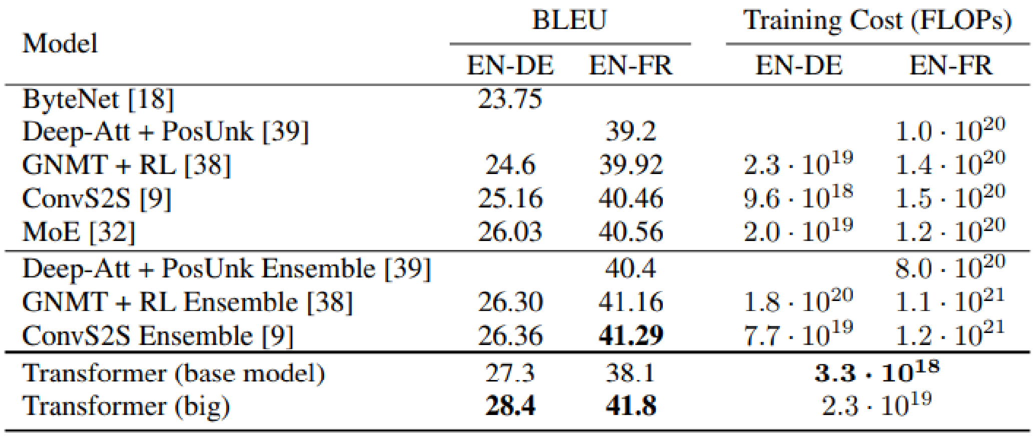
all the attention heads are harder to interpret



Source: https://jalammar.github.io/illustrated-transformer/

Neural Machine Translation

- Thanks to all combined techniques such as Multi-head attention, Positional encoding, etc. transformer-based models produce high-quality translation!

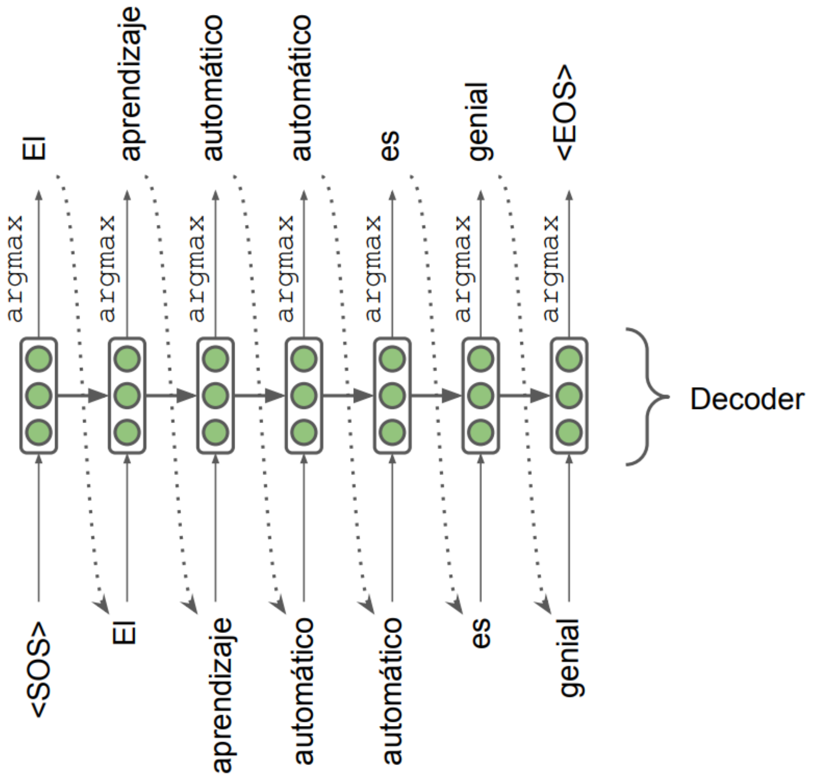


Model	BLEU		Training Cost (FLOPs)	
	EN-DE	EN-FR	EN-DE	EN-FR
ByteNet [18]	23.75			
Deep-Att + PosUnk [39]		39.2		$1.0 \cdot 10^{20}$
GNMT + RL [38]	24.6	39.92	$2.3 \cdot 10^{19}$	$1.4 \cdot 10^{20}$
ConvS2S [9]	25.16	40.46	$9.6 \cdot 10^{18}$	$1.5 \cdot 10^{20}$
MoE [32]	26.03	40.56	$2.0 \cdot 10^{19}$	$1.2 \cdot 10^{20}$
Deep-Att + PosUnk Ensemble [39]		40.4		$8.0 \cdot 10^{20}$
GNMT + RL Ensemble [38]	26.30	41.16	$1.8 \cdot 10^{20}$	$1.1 \cdot 10^{21}$
ConvS2S Ensemble [9]	26.36	**41.29**	$7.7 \cdot 10^{19}$	$1.2 \cdot 10^{21}$
Transformer (base model)	27.3	38.1	$\mathbf{3.3 \cdot 10^{18}}$	
Transformer (big)	**28.4**	**41.8**	$2.3 \cdot 10^{19}$	

Source: https://arxiv.org/pdf/1706.03762.pdf

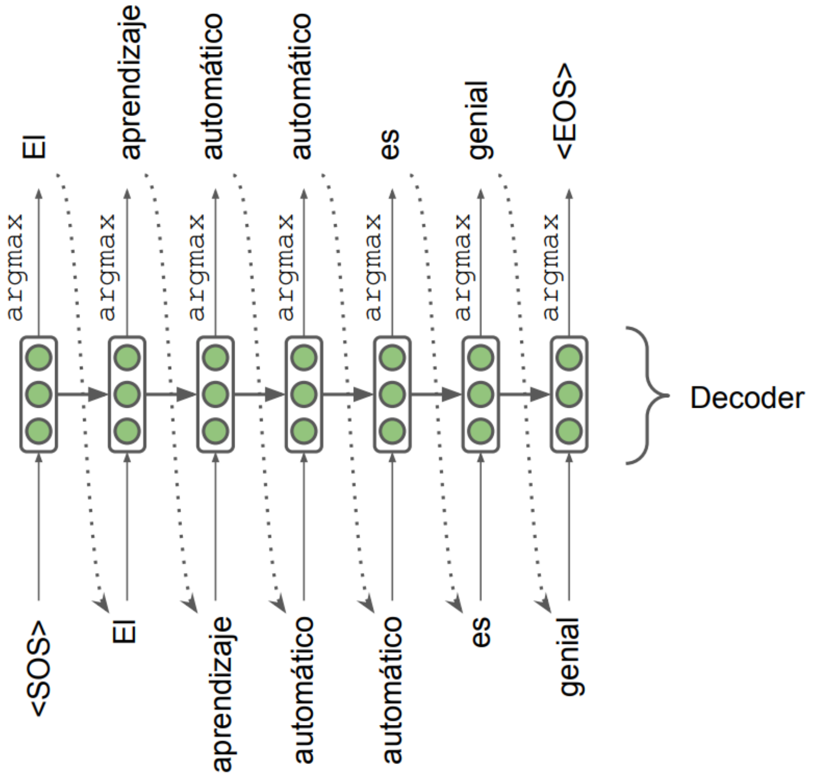
Why not greedy search?

- **Greedy** - decoder predicts the most probable token (argmax) on each step.



Why not greedy search?

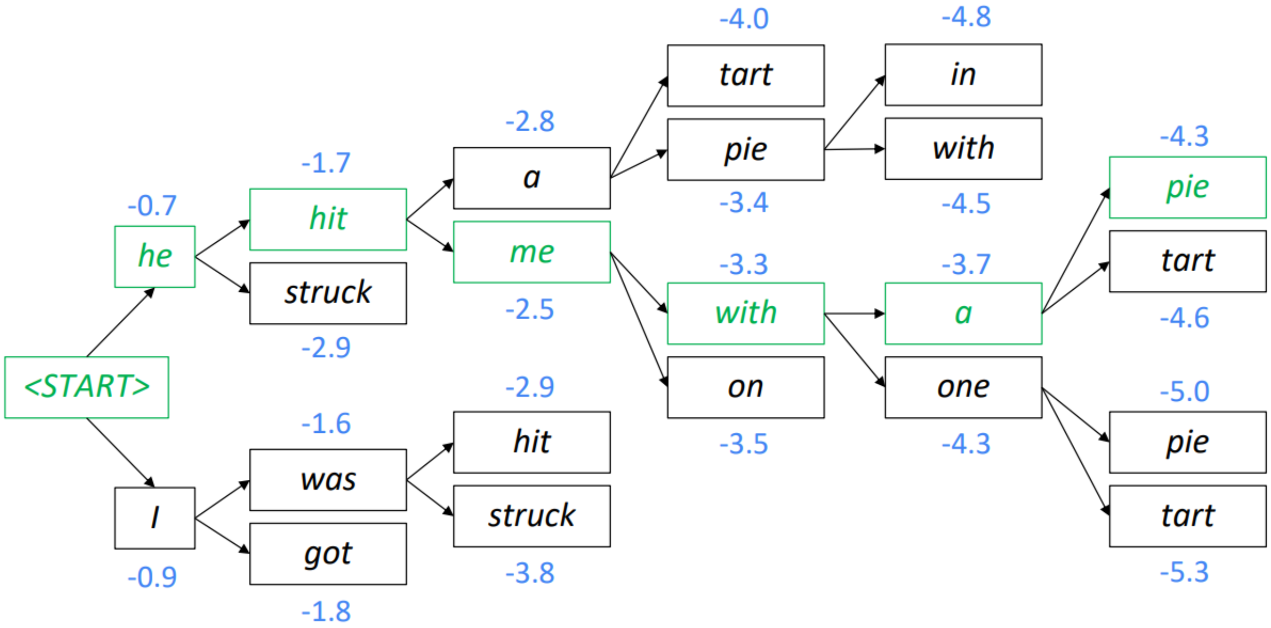
- **Greedy** - decoder predicts the most probable token (argmax) on each step.



We could try computing all possible sequences T

- This means that on each step i of the decoder, we are tracking V^i possible partial translations, where V is vocab size.
- This $O(V^I)$ complexity is far too expensive!

Beam search



Source: https://web.stanford.edu/class/cs224n/slides/cs224n-2021-lecture07-nmt.pdf

Beam search

- **Core idea**: On each step of decoder, keep track of the *k* most probable partial, where *k* is the beam size (in practice around 5 to 10)
- For each possible translation sentence, we calculate:

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

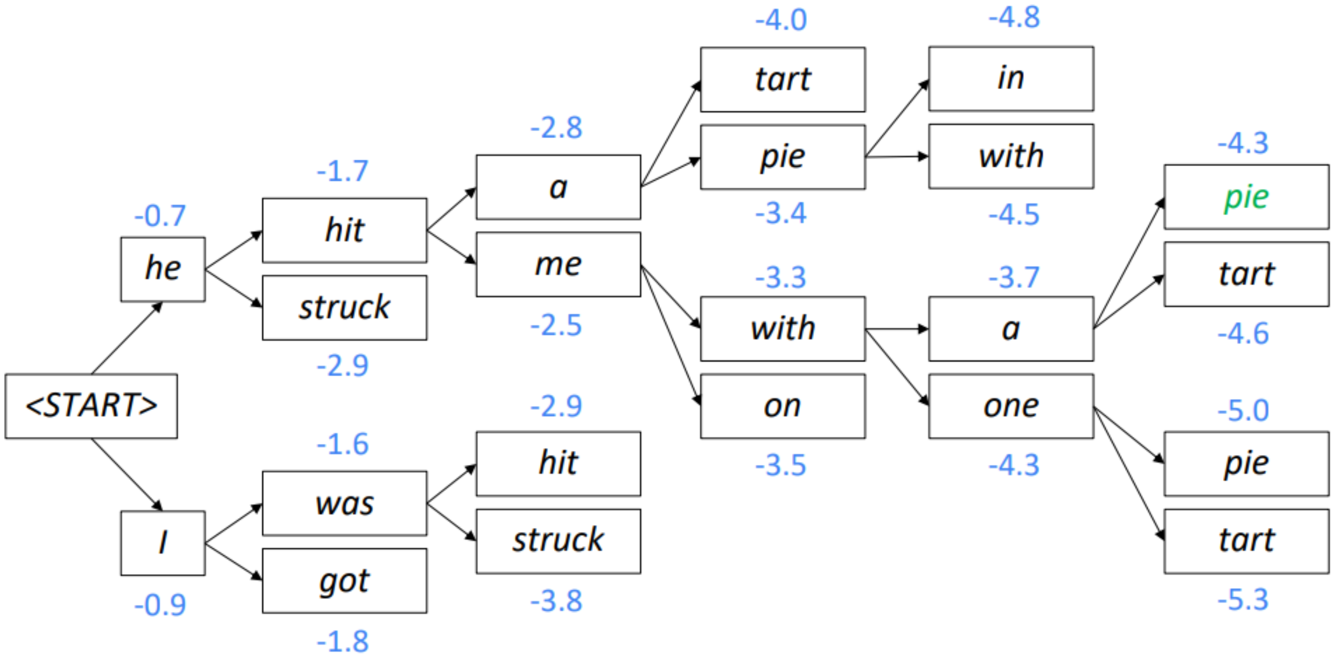
  - Scores are all negative, and higher score is better
  - We search for high-scoring sequence, tracking top *k* on each step

- Beam search **is not guaranteed to find optimal solution**

Source: https://web.stanford.edu/class/cs224n/slides/cs224n-2021-lecture07-nmt.pdf

Beam search

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



Source: https://web.stanford.edu/class/cs224n/slides/cs224n-2021-lecture07-nmt.pdf

Beam search

- **Main problem:** longer sequences have lower scores

- **Solution**: normalize by length. Use this to select top one instead:

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

Evaluation metrics

- n-gram-based metrics:
  - ROUGE
  - BLEU, spBLEU
  - METEOR
  - chrF++
- contextual embeddings-based:
  - BERTScore
- trainable metrics:
  - COMET
  - BLEURT
- **The best one**: human evaluation!

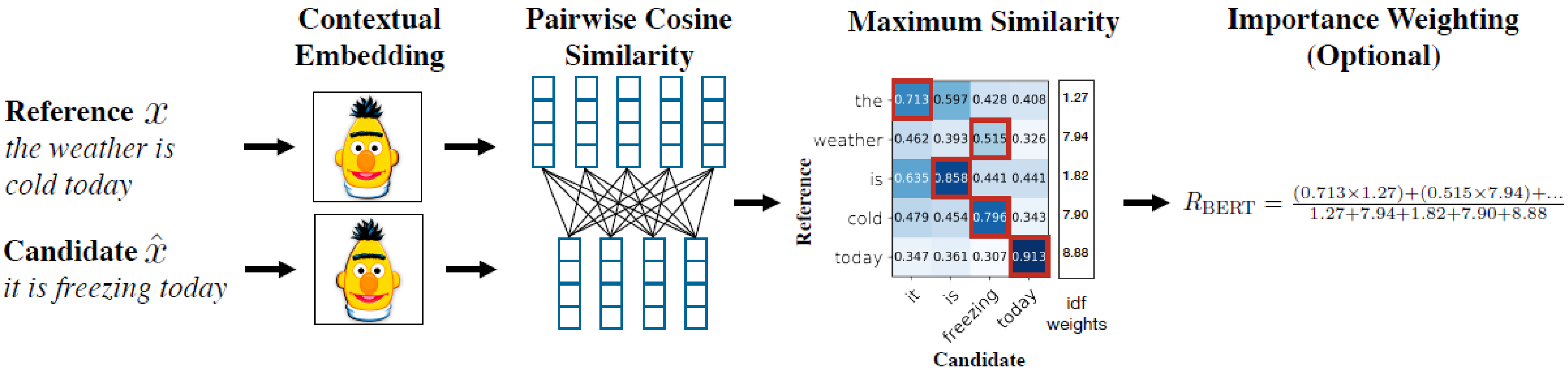
ROUGE

- a set of metrics specifically designed for evaluating automatic summarization, but that can be also used for machine translation and text generation.

- ROUGE-N:
  - $Precision = \frac{num\ n{-}gram\ matches}{num\ n{-}grams\ in\ generated\ translation}$
  - $Recall = \frac{num\ n{-}gram\ matches}{num\ n{-}grams\ in\ original\ translation}$

- **Variations:** ROUGE-1, ROUGE-2, ROUGE-L where *L* is determined as the longest subsequence between generated and original translations.

BERTScore



Reference	it	is	freezing	today	idf weights
the	0.713	0.597	0.428	0.408	1.27
weather	0.462	0.393	0.515	0.326	7.94
is	0.635	0.858	0.441	0.441	1.82
cold	0.479	0.454	0.796	0.343	7.90
today	0.347	0.361	0.307	0.913	8.88

$$R_{\mathrm{BERT}} = \frac{(0.713\times1.27)+(0.515\times7.94)+\ldots}{1.27+7.94+1.82+7.90+8.88}$$

Source: https://openreview.net/pdf?id=SkeHuCVFDr

Advantages of NMT

Compared to SMT, NMT has many advantages:

- Better performance
  - More fluent
  - Better use of context

- A single neural network to be optimized end-to-end
  - No subcomponents to be individually optimized

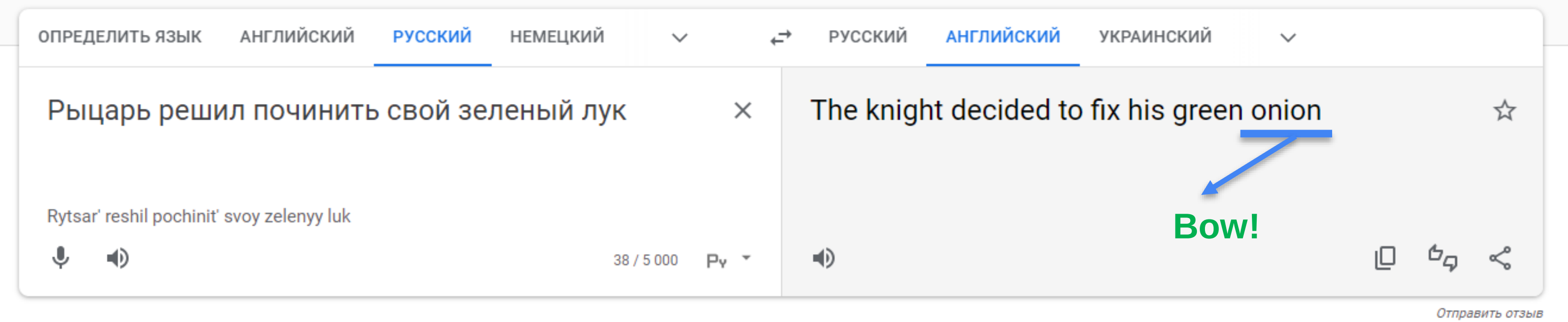
- Requires much less human engineering efforts
  - Same method for all language pairs

Disadvantages of NMT

- 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!

Polysemy words problem



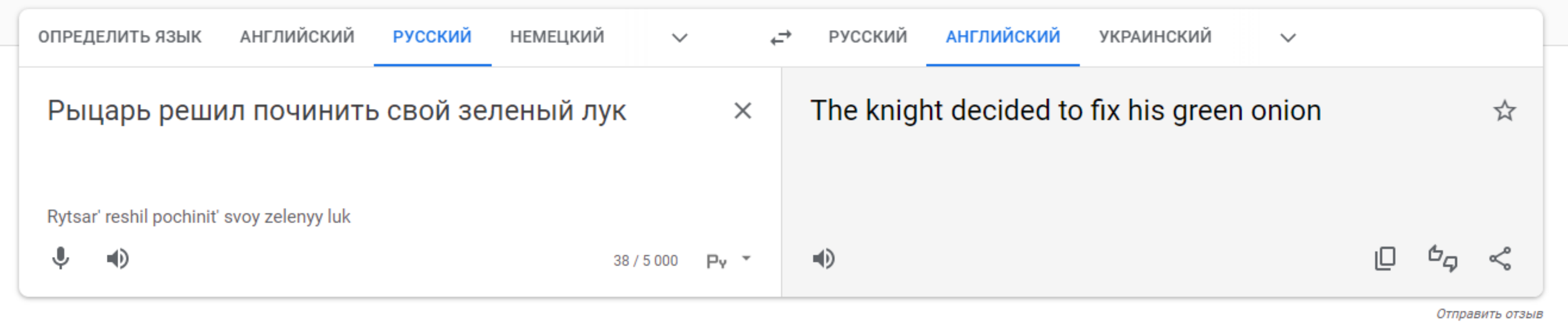
Рыцарь решил починить свой зеленый лук

Rytsar' reshil pochinit' svoy zelenyy luk

The knight decided to fix his green onion

Bow!

Polysemy words problem



Рыцарь решил починить свой зеленый лук

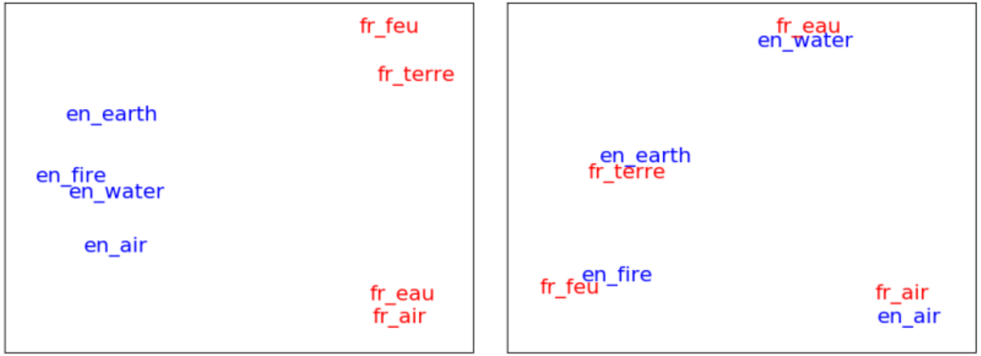
Rytsar' reshil pochinit' svoy zelenyy luk

38 / 5 000

The knight decided to fix his green onion

- We need to make internal representation of languages more cross-lingual!

Application of Procrustes alignment for word vectors



Modern Research

- Workshop on Machine Translation

- No Language Left Behind: Scaling Human-Centered Machine Translation

NLLB-200 project consider improvement MT for all possible languages including low-resource ones.
No intermediate language(s)!

Source: https://arxiv.org/ftp/arxiv/papers/2207/2207.04672.pdf