High Impact Skills Development Program

in Artificial Intelligence, Data Science, and Blockchain

Module 8: Natural Language Processing

Lecture 5: Neural Machine Translation

Instructor: Ahsan Jalal

Industry Trainer

Overview of Today's Lecture

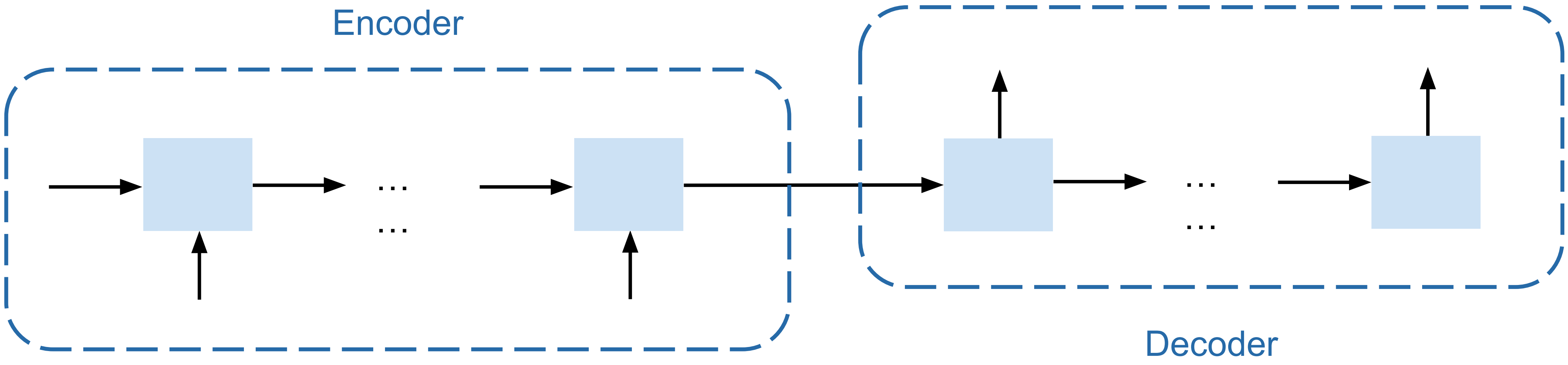
Machine Translation

- Neural network based machine translation
- Decoding Algorithms
- Attention in $seq2seq$ models

Neural Machine Translation

- Neural Machine Translation (NMT) is a way to do Machine Translation with a *single neural network*
- The neural network architecture is called sequence-to-sequence (aka seq2seq) and it involves *two* RNNs.

seq2seq models work with sequential inputs and outputs



Many-to-Many V2

seq2seq **models are quite versatile**

- Many other NLP tasks can be phrase as seq2seq problems.

  - Summarisation
  - Dialogue
  - Code Generation
  - Language Translation

Figure 26.13 A sample dialogue from the HIS System of Young et al. (2010) using the dialogue acts in Fig. 26.12.

Language translation is a prime application of *seq2seq* models

- The task of language translation requires converting an input sentence x from a source language to an output sentence y in the target language preserving the semantic information.

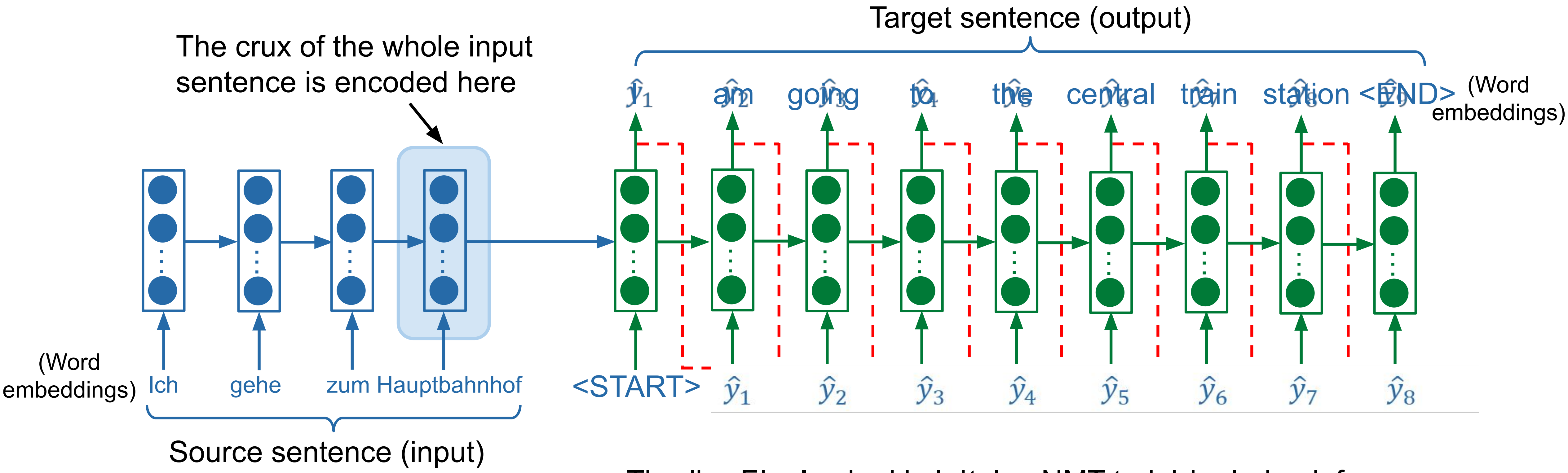
I play the flute.	⟹	میں بانسری بجاتا ہوں۔
I play cricket.	⟹	میں کرکٹ کھیلتا ہوں۔

- If the semantic information is not preserved, the objective of language translation is not achieved.

The spirit is willing but the flesh is weak.	⟹	The liquor is good but the meat is spoiled.
Out of sight, out of mind.	⟹	Invisible, idiot.

Neural Machine Translation (NMT) made its entry in 2014

- Decoder generates target language sentence conditioned on the encoding of the source sentence.



Teacher Forcing is used during NMT training. The decoder is used in autoregressive mode during inference.

seq2seq models is a type of conditional language model

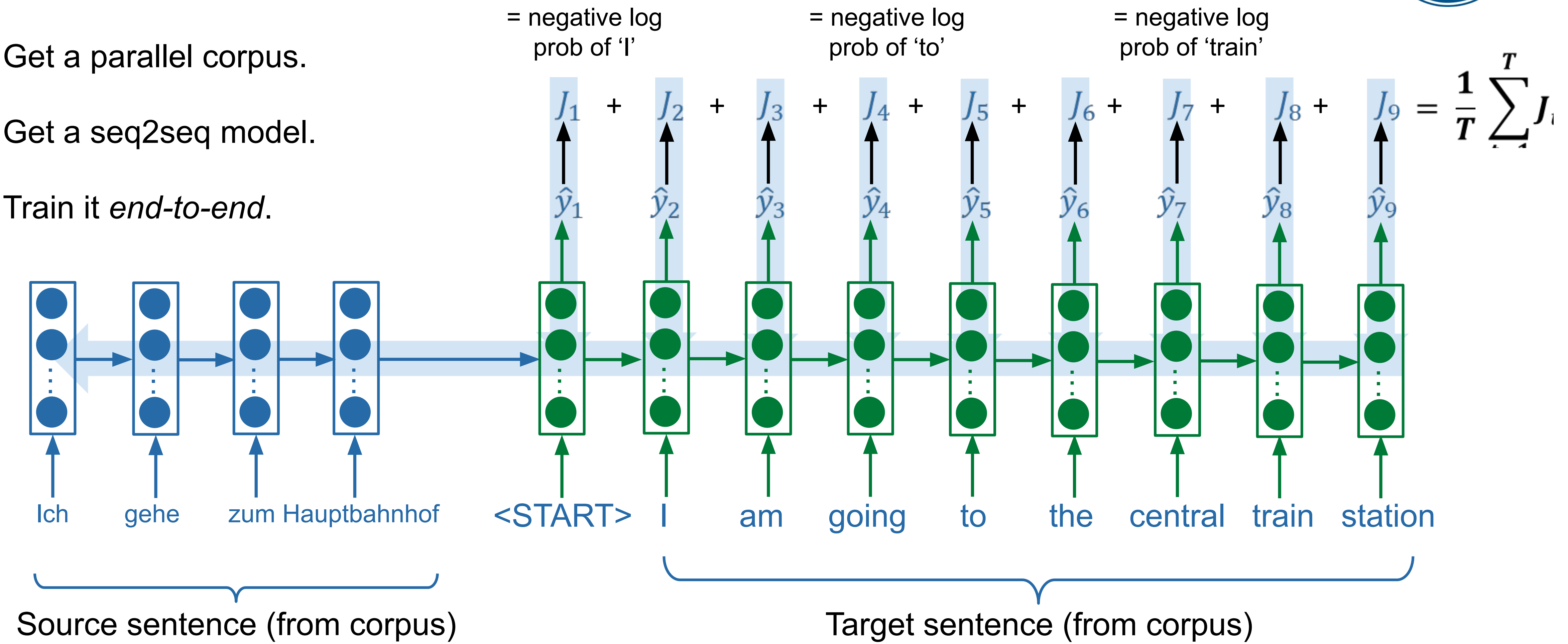
- Language Model generates coherent and grammatically correct sequence of words.
  - Predicts next word in the target sentence.
- Conditional Language Models generate output considering a given condition.
  - The next word in the target sequence is conditioned on the encoding of source sentence and the previously predicted word in the target sentence.
- Mathematically, the task of NMT is to predict;

$$P(e|g) = P(e_1|g)P(e_2|e_1, g)P(e_3|e_1, e_2, g) \ldots P(e_T|e_1, e_2, \ldots e_{T-1}, g)$$

Here g represents German sentence (target) and e stands for English sentence (source).

How to train an RNN-based Language Model?

- Get a parallel corpus.
- Get a seq2seq model.
- Train it *end-to-end*.

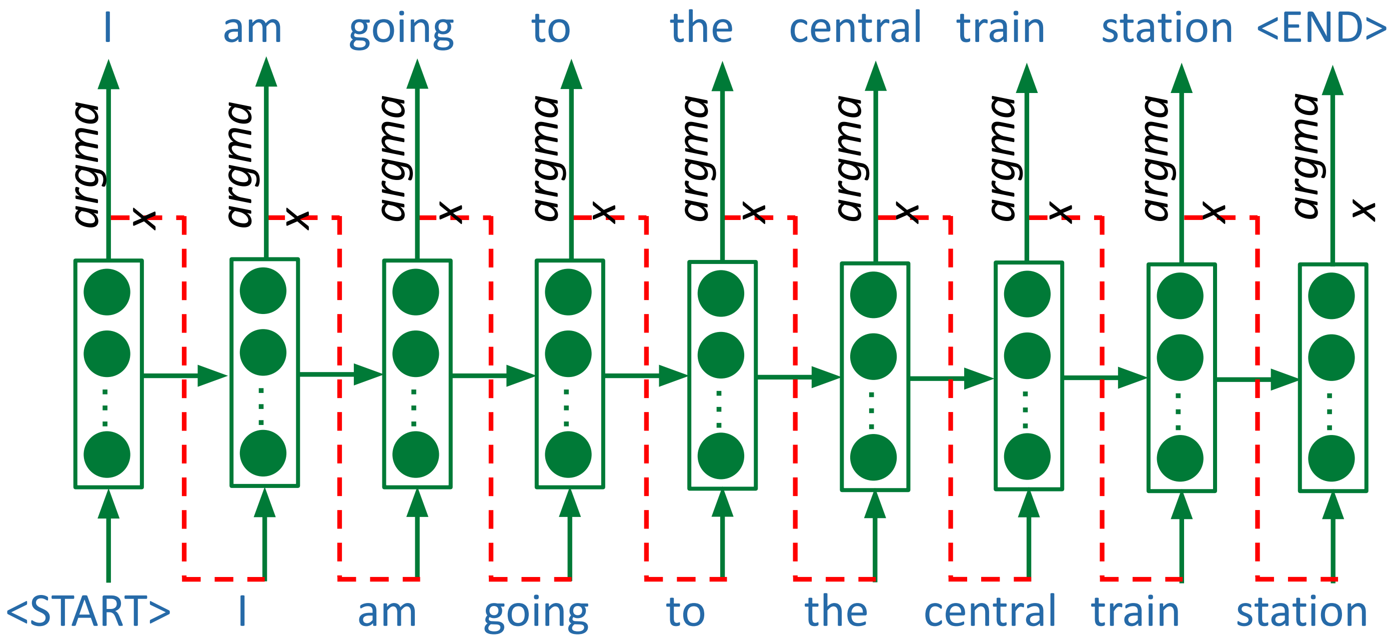


Greedy decoding picks the next word based on the highest score

- The use of $argmax$ ensures that only the words with the highest probability are chosen as expected output $\hat{y}_t$ at each time step.

- This greedy approach is not always desirable. Why?

$$P(g|e) = P(g_1|e)P(g_2|g_1, e)P(g_3|g_1, g_2, e) \dots P(g_T|g_1, g_2, \dots g_{T-1}, e)$$



Greedy decoding has some serious problems

- We cannot undo a choice that is made at a previous time step.
- Suppose the input is *Ich gehe zum Hauptbahnhof* (*I am going to the central train station*).

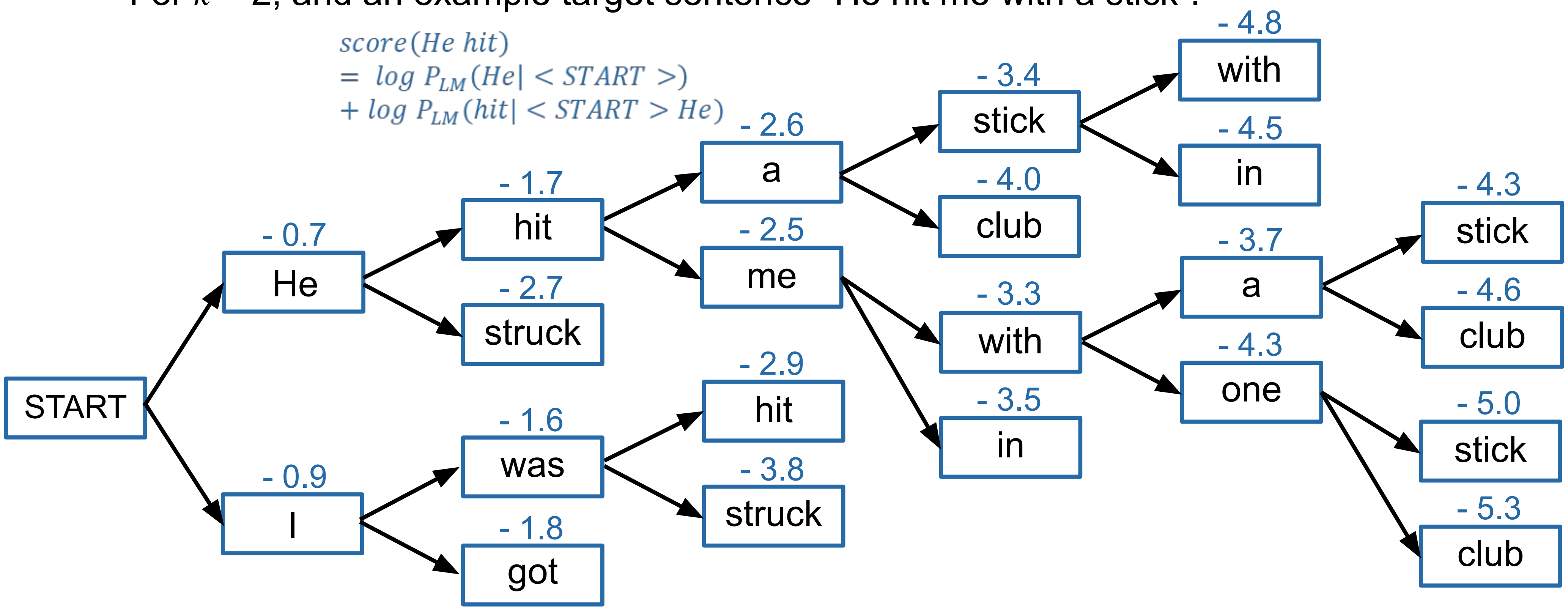
  I ___ I am ___ I am going ___ I am going back (made a mistake here)

- How can we fix this?
  - Exhaustive Search? Compute all possible sequences.
    - At each time step t of decoder, V^t partial translations are tracked, where V is vocabulary size.
    - This results in $O\ (V^T)$ complexity which is far too expensive.

Let’s take an example of beam search

- For k = 2, and an example target sentence “He hit me with a stick”.

$score(He\ hit)$
$= \log P_{LM}(He| < START >)$
$+ \log P_{LM}(hit| < START > He)$



$score(I\ got)$
$= \log P_{LM}(I| < START >)$
$+ \log P_{LM}(got| < START > I)$

How pick the most suitable hypothesis using beam search?

- In greedy search, target sentence is ended when $< END >$ token is generated.
- In beam search, different hypotheses may generate $< END >$ token at different time steps.
  - Set aside a completed hypothesis that has generated $< END >$ token and continue exploring others.
  - Beam search is stopped when either
    - T (predefined) time steps have arrived.
    - N (predefined) number of hypotheses have been completed.
- Absolute hypotheses scores can be deceiving.

$$score(\hat{y}^{<1>}, \hat{y}^{<2>}, \dots, \hat{y}^{<t>}) = \sum_{i=0}^{t} \log P_{LM}\,(\hat{y}^{i} | \hat{y}^{<1>}, \hat{y}^{<2>}, \dots, \hat{y}^{<i-1>}, x)$$

The NMT has some merits and demerits

- Provides better performance.
  - More fluent translation
  - Better use of context
- Single end-to-end system can be efficiently and conveniently optimised.
  - No subcomponents to optimise individually.
- Requires less human effort.
  - No feature engineering needed.
- Reusable
  - Same model different language pairs.
  - Requires bilingual data of course.

- Less interpretable
  - Difficult to track errors
- Hard to exert control
  - Cannot specify rules
- Safety concerns
  - Model can say whatever it wants.

MT models are evaluated using BLEU metric

- It compares machine translation with one or more human translations and computes a similarity score based on n-gram precision and brevity penalty.
  - Checks how many n-grams generated by MT are actually present in human translation.
  - Also evaluates if MT is significantly shorter than human translation. If c is length of candidate translation and r is the length of reference translation.

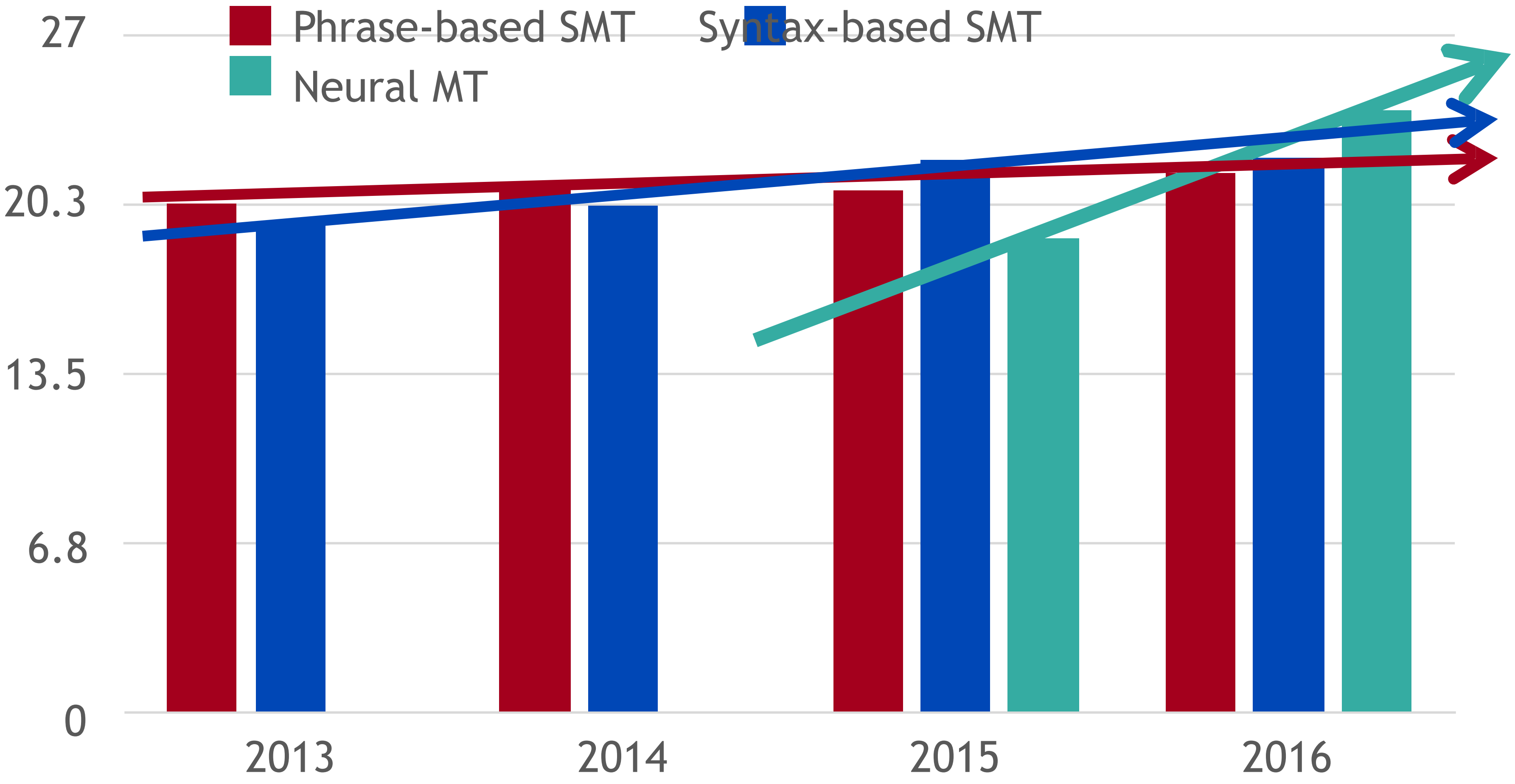
$$Brevity\ Penalty = \begin{cases} 1, & if\ c > r \\ e^{\left(1-\frac{r}{c}\right)}, & if\ c \leq r \end{cases}$$

- BLEU is useful but imperfect.
  - There can be many valid translations. It does not consider semantic similarity between words, or

Papineni, Kishore, et al. "BLEU: a method for automatic evaluation of machine translation." Proceedings of the 40th annual meeting of the Association for Computational Linguistics. 2002.

MT progress over time

[Edinburgh En-De WMT newstest2013 Cased BLEU; NMT 2015 from U. Montréal]



Source:
http://www.meta-net.eu/events/meta-forum-2016/slides/09_sennrich.pdf

NMT: the biggest success story of NLP Deep Learning

Neural Machine Translation went from a fringe research activity in **2014** to the leading standard method in **2016**

- **2014**: First seq2seq paper published
- **2016**: Google Translate switches from SMT to NMT
- This is amazing!
  - **SMT** systems, built by hundreds of engineers over many years, outperformed by NMT systems trained by a handful of engineers in a few months

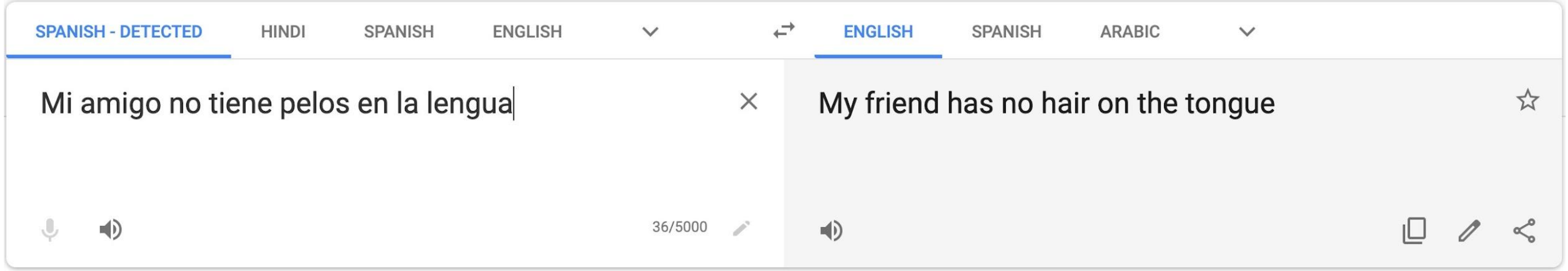
The problem of machine translation is far from solved

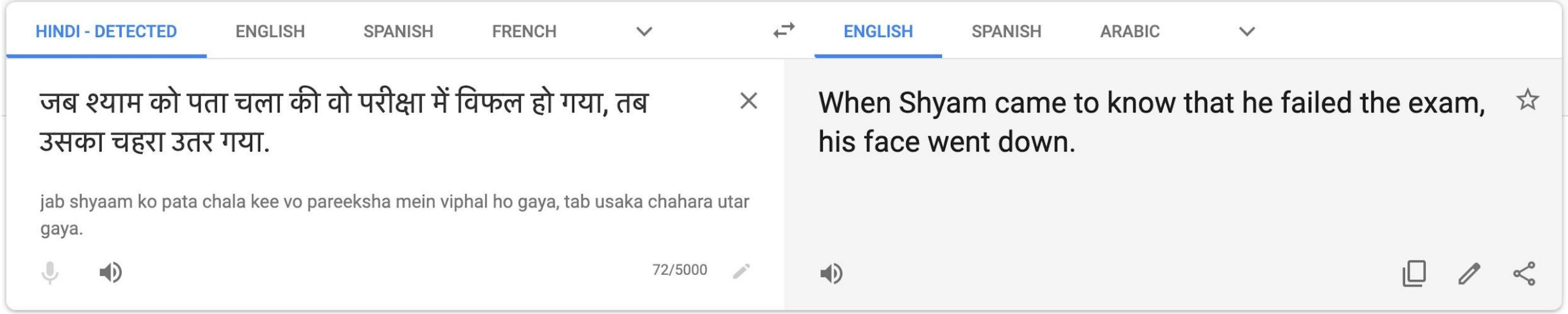
- Machine translation has achieved a lot but many challenges still remain.
    - Out of vocabulary words.
    - Domain mismatch.
    - Maintaining wider context.
    - Low-resource language pairs.

Interesting Read: https://www.skynettoday.com/editorials/state_of_nmt

So is Machine Translation solved?

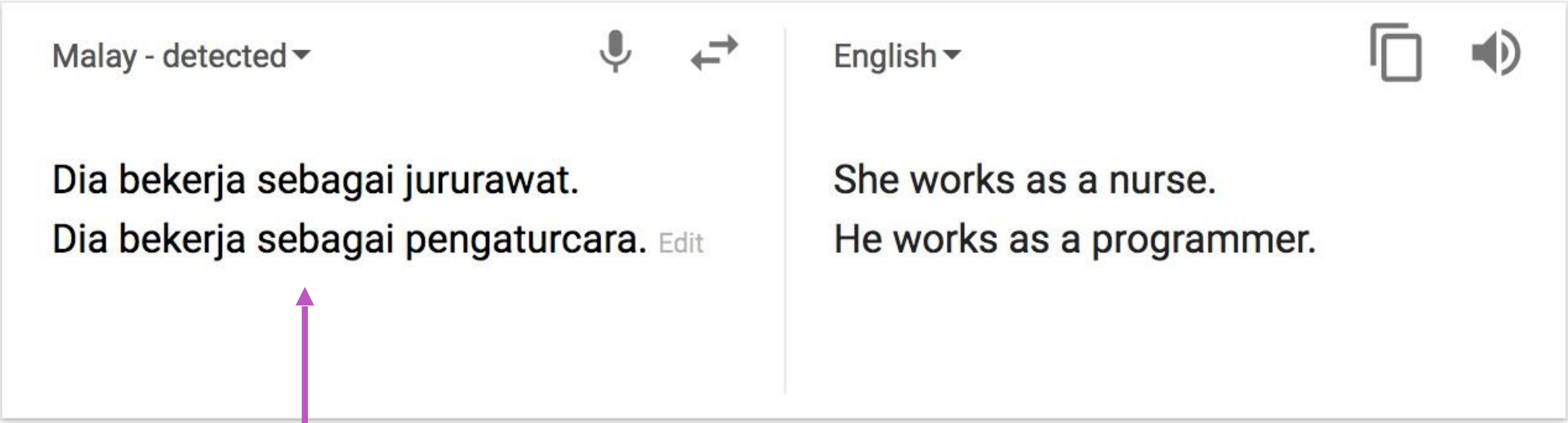
- **Nope!**
- Using common sense is still hard
- Idioms are difficult to translate





So is Machine Translation solved?

- **Nope!**
- NMT picks up biases in training data

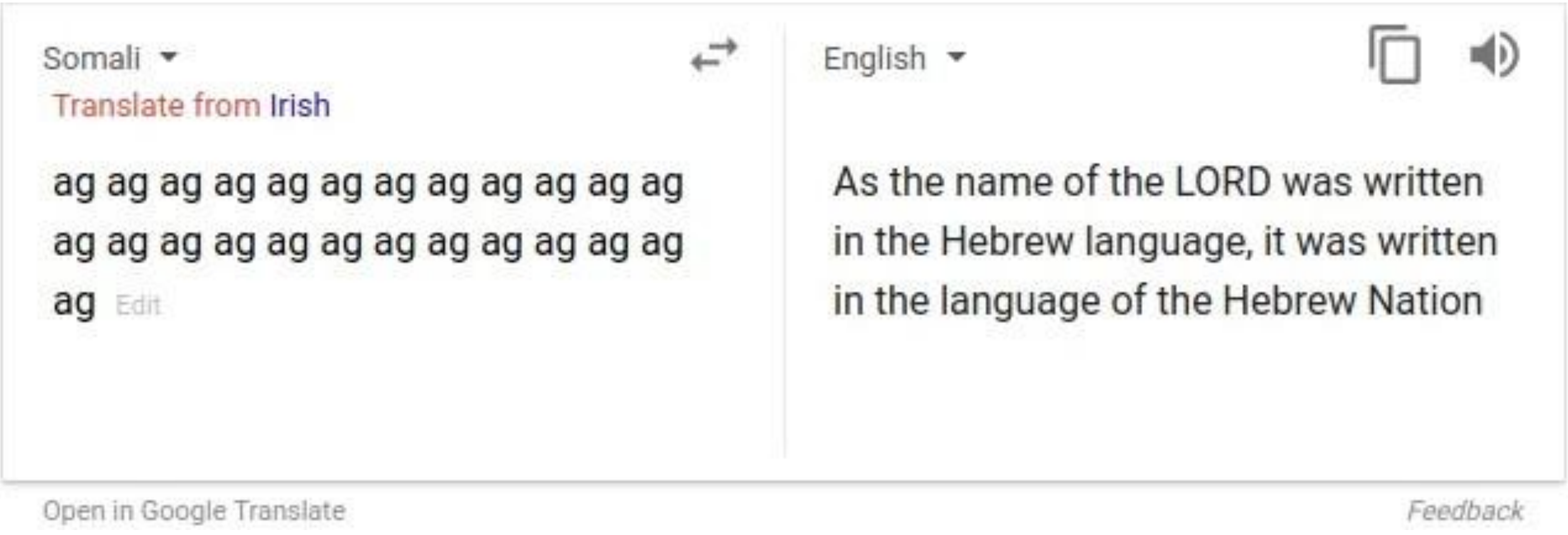


Didn't specify gender

Source: https://hackernoon.com/bias-sexist-or-this-is-the-way-it-should-be-

So is Machine Translation solved?

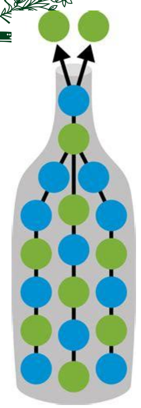
- Nope!
- Uninterpretable systems do strange things



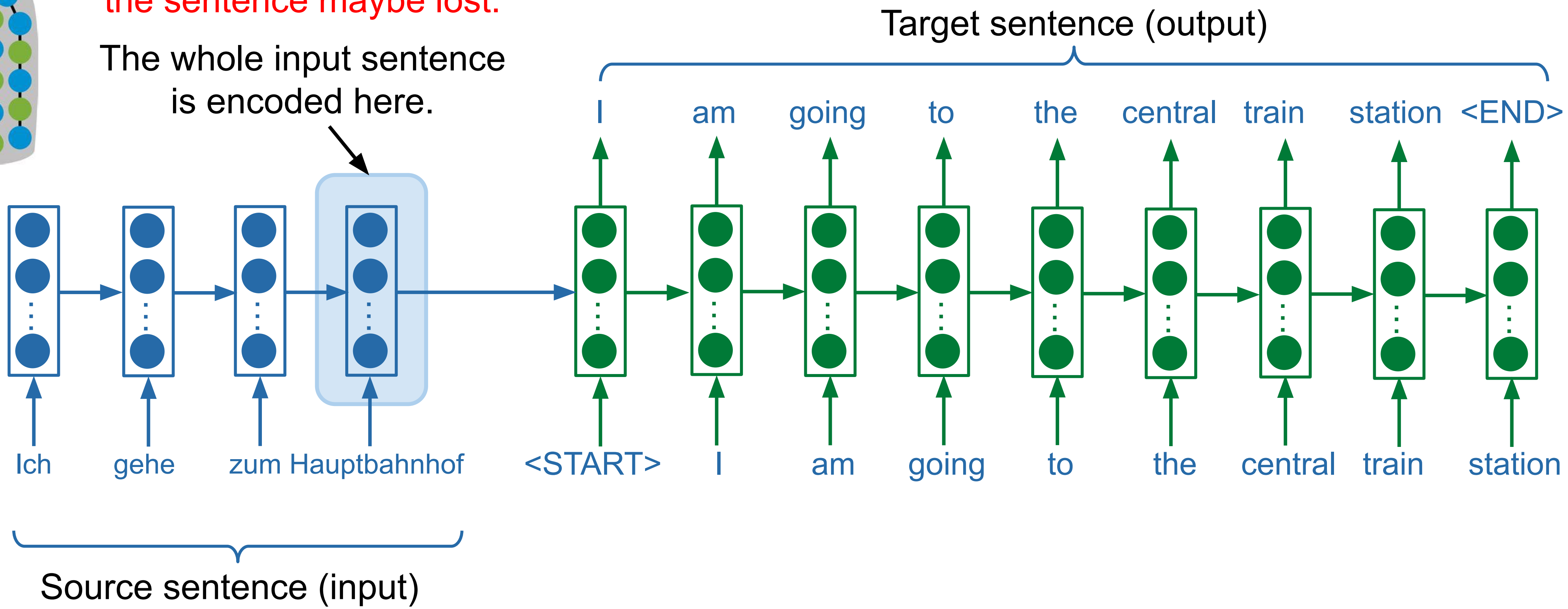
Picture source: https://www.vice.com/en_uk/article/j5npeg/why-is-google-translate-spitting-out-sinister-religious-prophecies

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

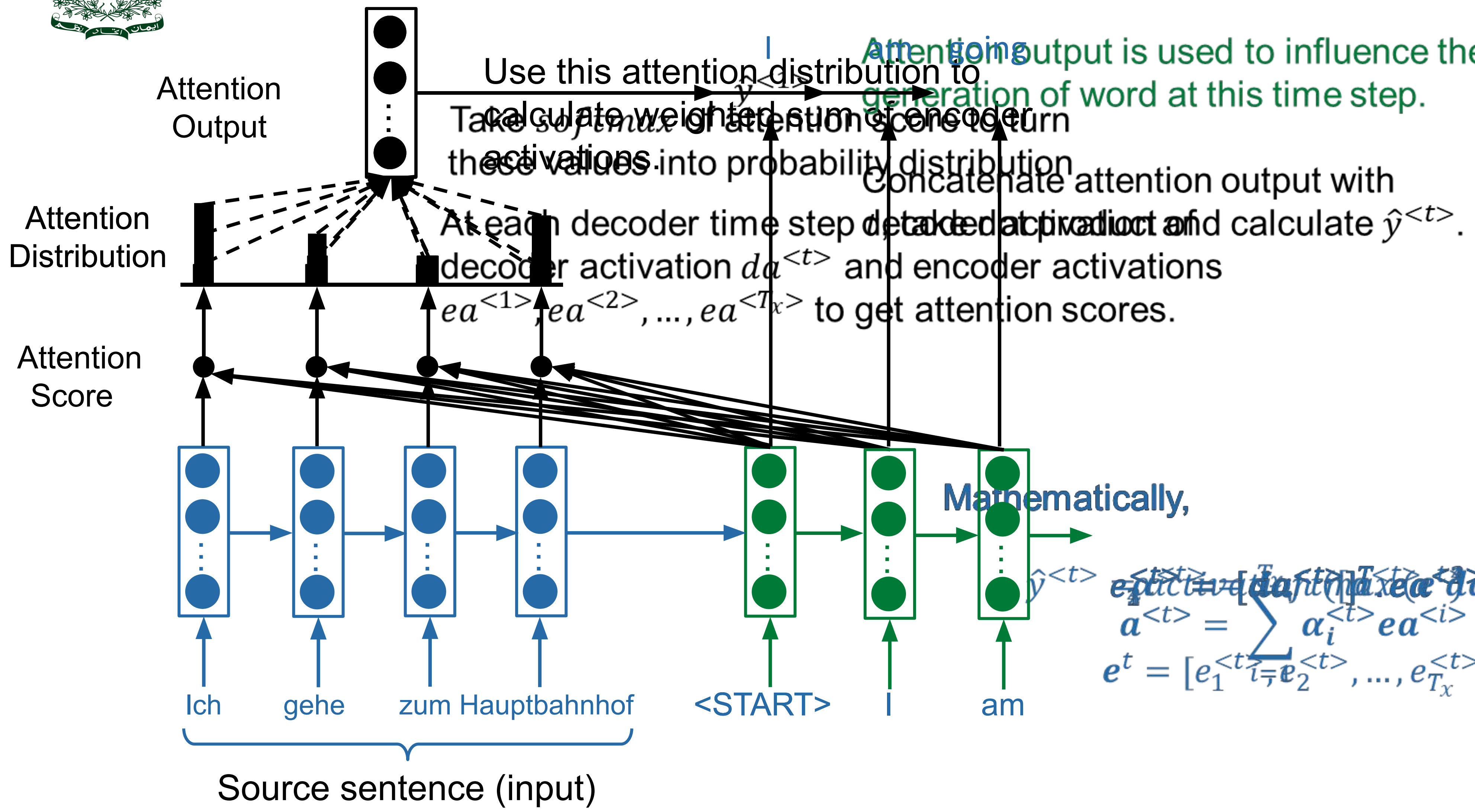
Classical seq2seq model has a few shortcomings



Information from the start of the sentence maybe lost.



The bottleneck in seq2sec models can be removed using Attention



Mathematically,

$$a^{<t>} = \sum_{i=1}^{T_x} \alpha_i^{<t>} ea^{<i>}$$

$$e^t = [e_1^{<t>}, e_2^{<t>}, \ldots, e_{T_x}^{<t>}]$$

Adding attention to seq2seq models has many advantages

- Attention helps decoder focus on relevant parts in the source sentence.
- It resolves information bottleneck problem.
  - Instead of relying on a single vector to capture the whole source sentence, now decoder has two vectors for guidance.
- It also helps with vanishing gradient.
  - Direct connections between encoder and decoder are helpful especially in longer sentences.
- Attention may provide some interpretability.
  - Analysis of attention output can help understand what the decoder was fixating at while predicting a certain target word.
  - Soft alignment is achieved for free without even explicitly training for it.

Summary of today's lecture

- Since 2014, Neural MT rapidly replaced intricate Statistical MT

- Sequence-to-sequence is the architecture for NMT (uses 2 RNNs)

- Attention is a way to *focus on particular parts* of the input
  - Improves sequence-to-sequence alot!

Happy Learning!