Machine Translation

Prof. Luca Cagliero
Dipartimento di Automatica e Informatica
Politecnico di Torino

Politecnico
di Torino

Lecture goal

- Preliminaries
- Rule-based approaches
- Statistical Machine Translation
- Neural Machine Translation

Machine translation

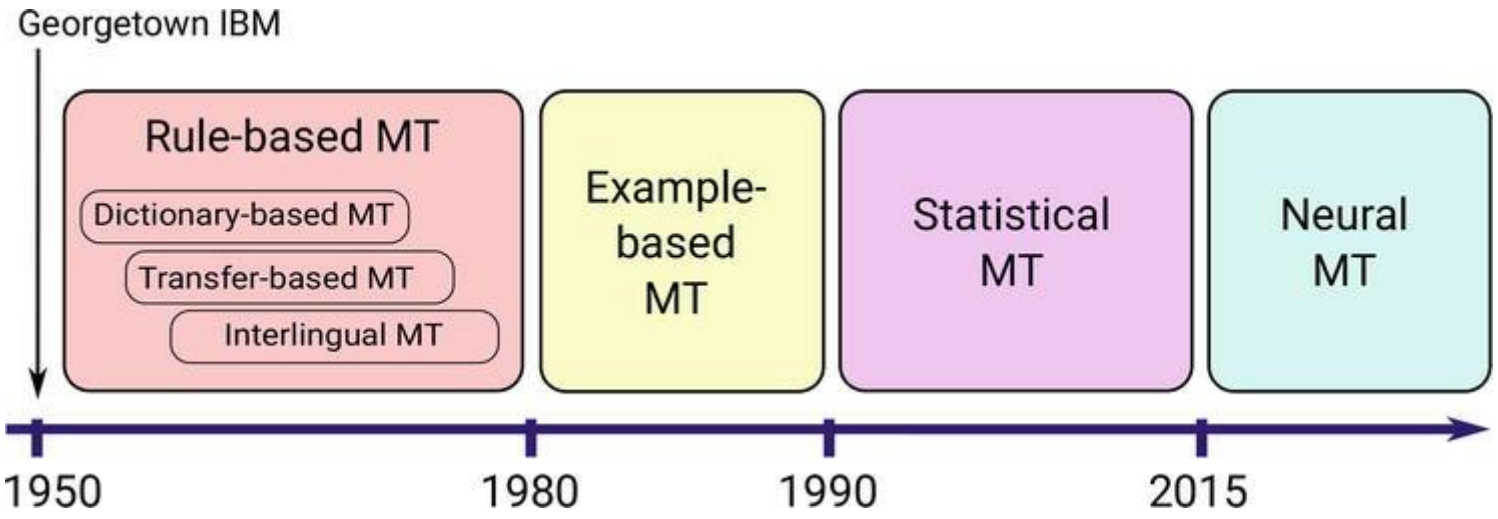
- Automatic translation of a sentence from a source language to a target language

Machine translation

- Approaches
  - Rule- or dictionary-based
  - Statistical
  - Neural

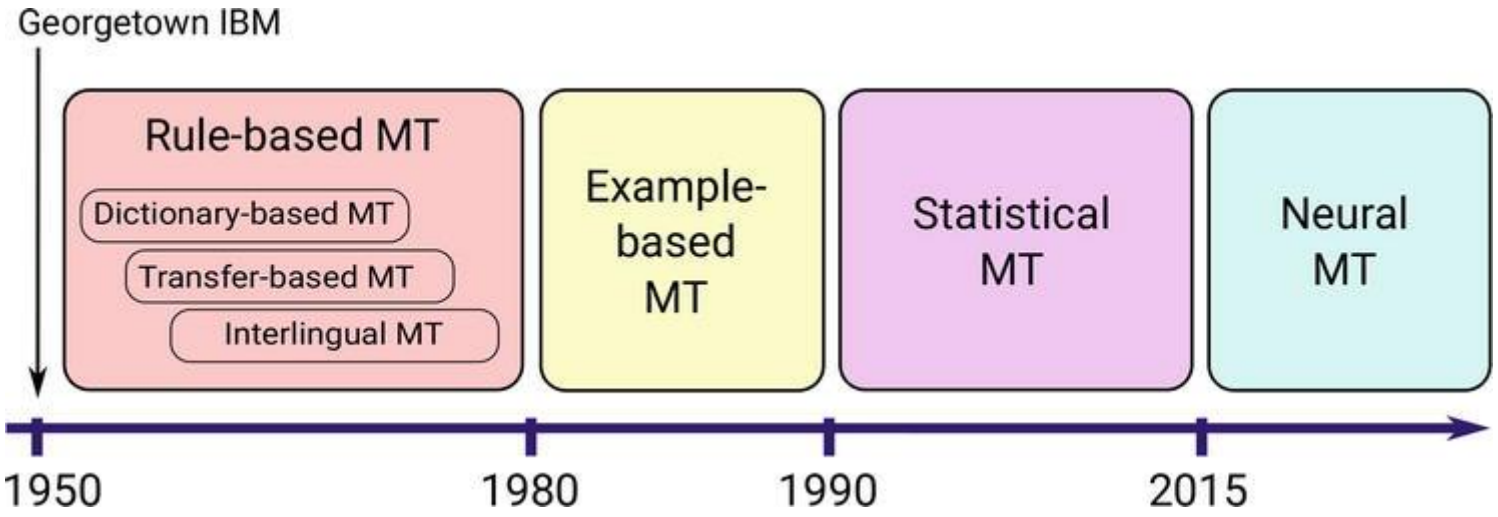
Rule-based Machine Translation

- Machine translation systems based on linguistic information about source and target languages basically retrieved from (bilingual) dictionaries and grammars
- Also known as **Knowledge-Based Machine Translation** or **Classical MT Approach**



Example-based Machine Translation

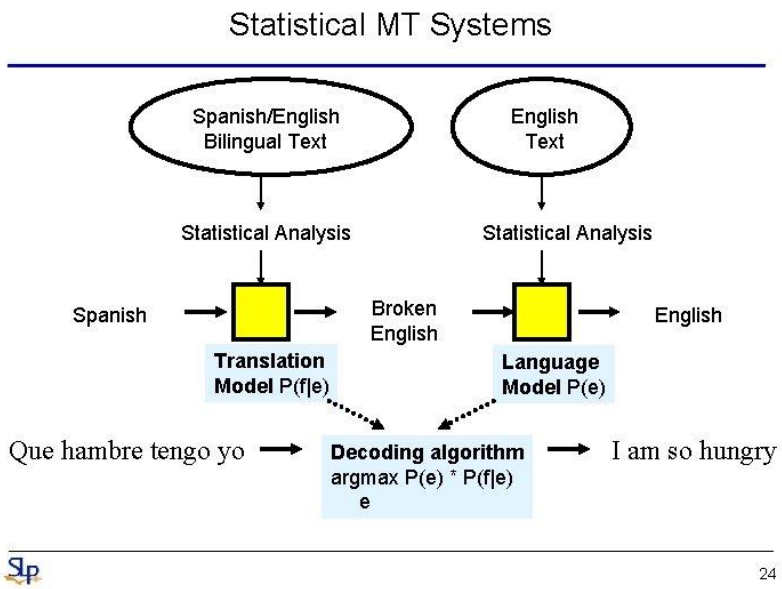
- Bilingual corpus with parallel texts
  - set of sentences in the source language (from which one is translating)
  - the corresponding translations of each sentence in the target language with point-to-point mapping
- Translate by analogy



Machine Translation Approaches: Issues and Challenges. M.D. Okpor. IJCSI International Journal of Computer Science Issues, Vol. 11, Issue 5, No 2, September 2014
ISSN (Print): 1694-0814 | ISSN (Online): 1694-0784

Statistical Machine Translation

- Based on statistical models whose parameters are derived from the analysis of bilingual text corpora.
- Key ideas
  - every sentence in one language is a possible translation of any sentence in the other
  - the most appropriate is the translation that is assigned the highest probability by the system



Introduction to Statistical Machine Learning. Raghav Bashyal.

Statistical Machine Translation

- Learn a probabilistic model from data
- Given a sentence x in the source language find the best sentence y in the target language

$$\mathbf{arg_y\ max\ P(y \mid x)}$$

- Apply Bayes theorem

$$\mathbf{arg_y\ max\ P(x \mid y)P(y)}$$

P(y): language model

P(x | y): translation model

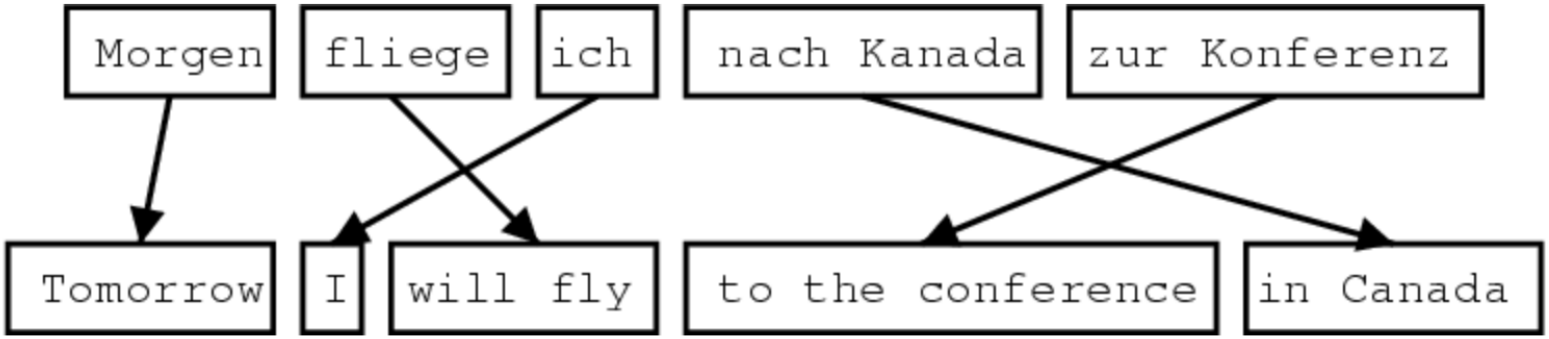
Statistical Machine Translation

- The **translation model** describes how words and phrases should be translated
  - Requires a bilingual collection of human-translated sentences
- The **language model** describes how to write in the target language
  - Requires a monolingual collection

Christopher Manning. Natural Language Processing with Deep Learning CS224N/Ling284

Translation model learning

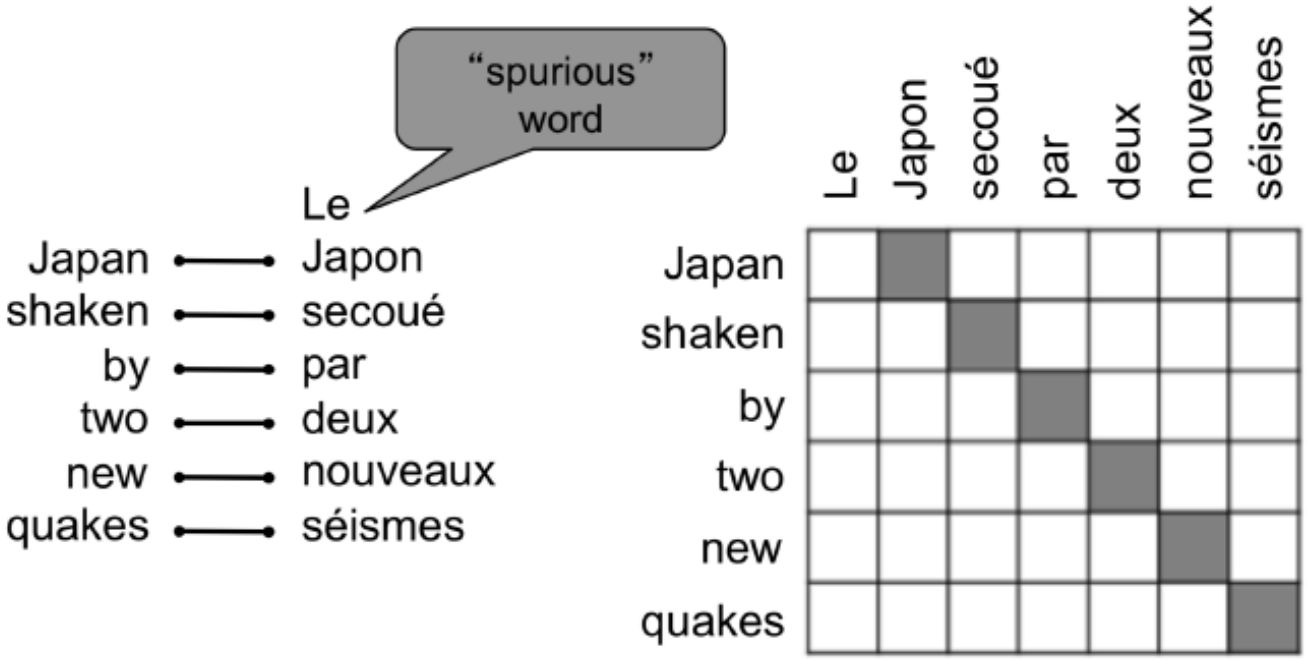
- $P(x \mid y) = P(x, a \mid y)$
- a: alignment term
  - word-level correspondence between source sentence x and target sentence y



Christopher Manning. Natural Language Processing with Deep Learning CS224N/Ling284

Alignment

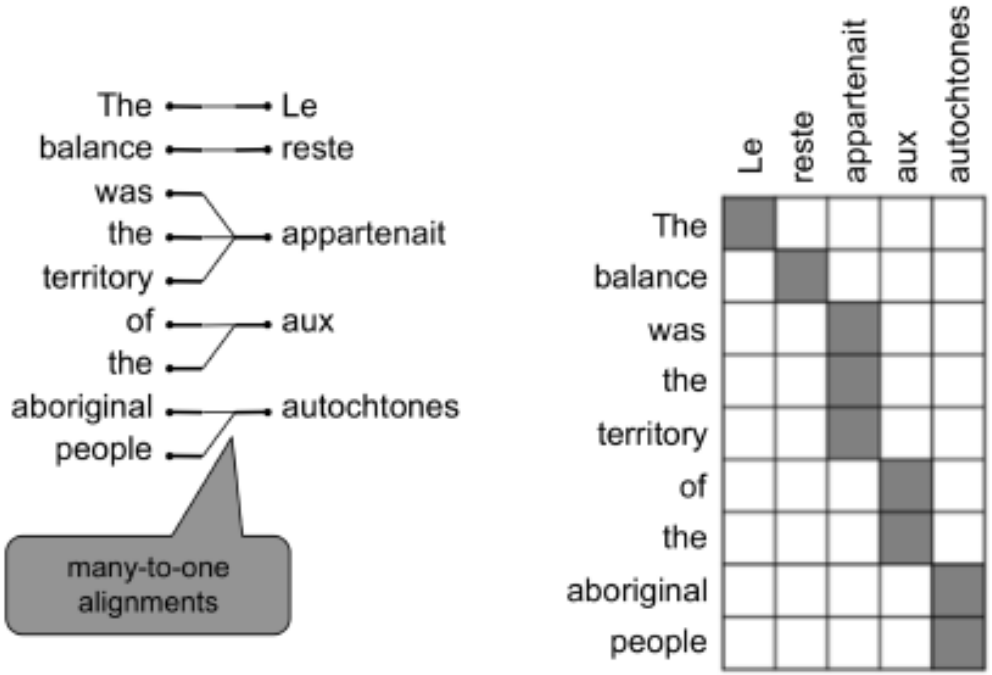
- Bilingual lexicon
  - correspondence between particular words
    - often incomplete



The Mathematics of Statistical Machine Translation: Parameter Estimation", Brown et al, 1993

Alignment

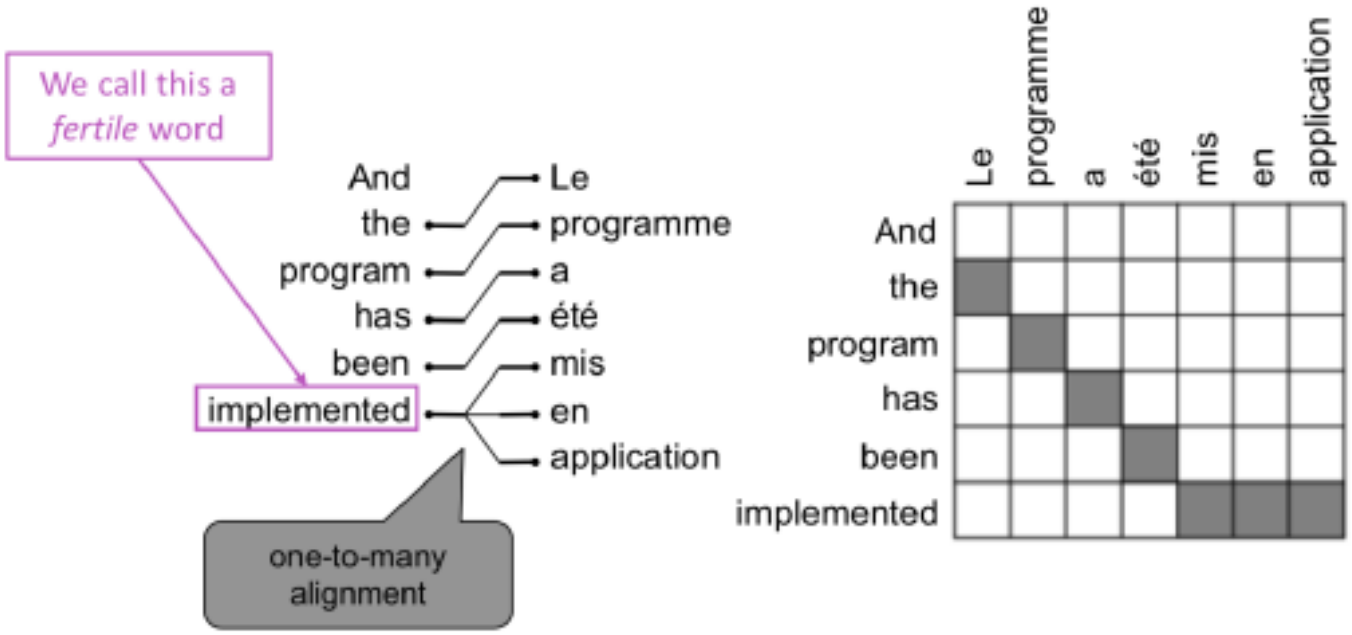
- **Many-to-one** correspondences are allowed



The Mathematics of Statistical Machine Translation: Parameter Estimation", Brown et al, 1993

Alignment

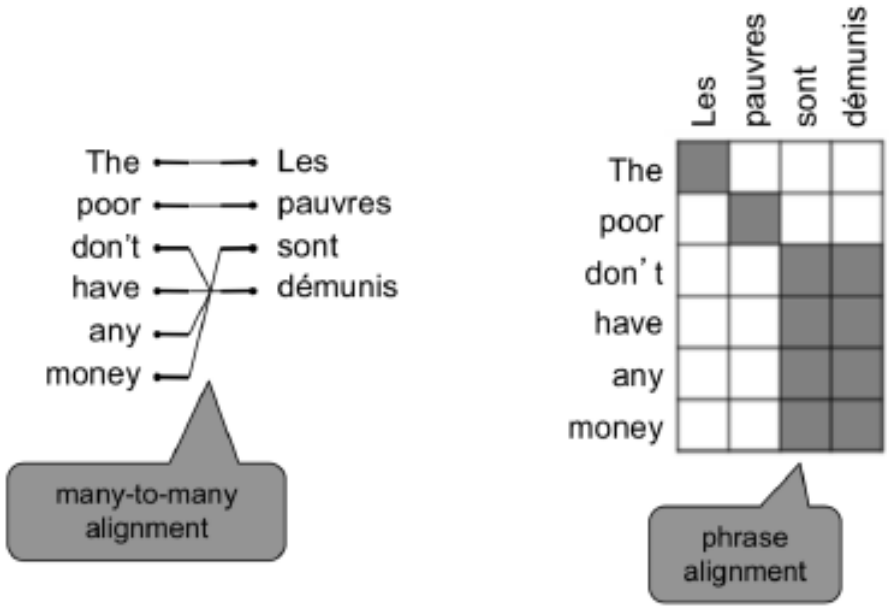
- **One-to-many** correspondences are allowed



The Mathematics of Statistical Machine Translation: Parameter Estimation", Brown et al, 1993

Alignment

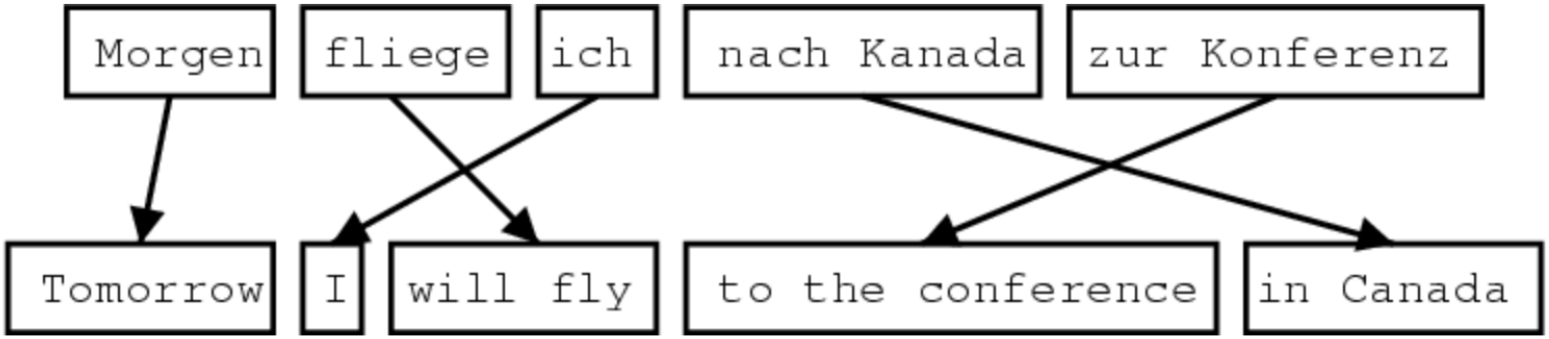
- **Many-to-many** correspondences are allowed



The Mathematics of Statistical Machine Translation: Parameter Estimation", Brown et al, 1993

Translation model learning

- P(x, a | y) is influenced by
  - Probability of word alignment
  - Text position
  - ...
- a is a latent variable
  - Expectation-Maximization



Christopher Manning. Natural Language Processing with Deep Learning CS224N/Ling284

Decoding

- Maximization process

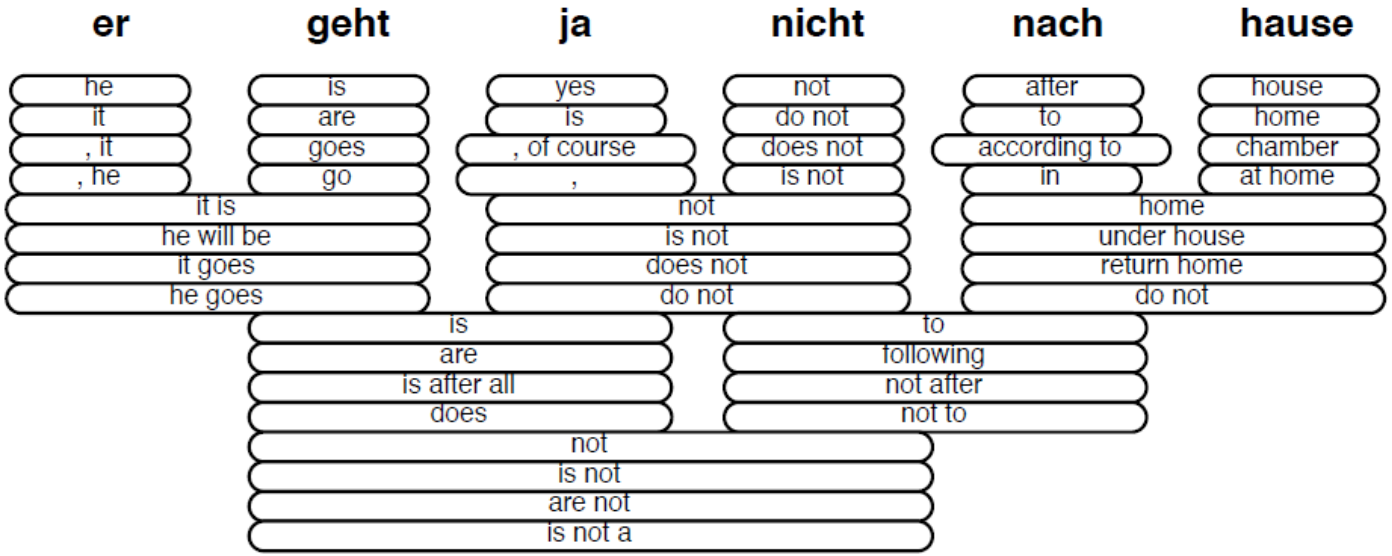
$$\mathbf{arg_y\ max\ P(x \mid y)P(y)}$$

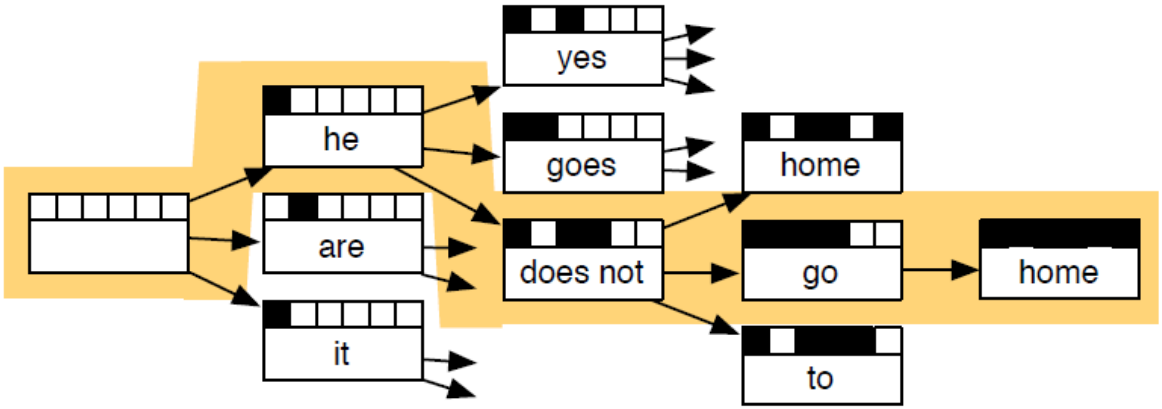
- Unfeasible without enforcinng independence assumptions

Koehn, 2009 Statistical Machjine Translation

Decoding

- Tree-based candidate exploration and evaluation





Koehn, 2009 Statistical Machjine Translation

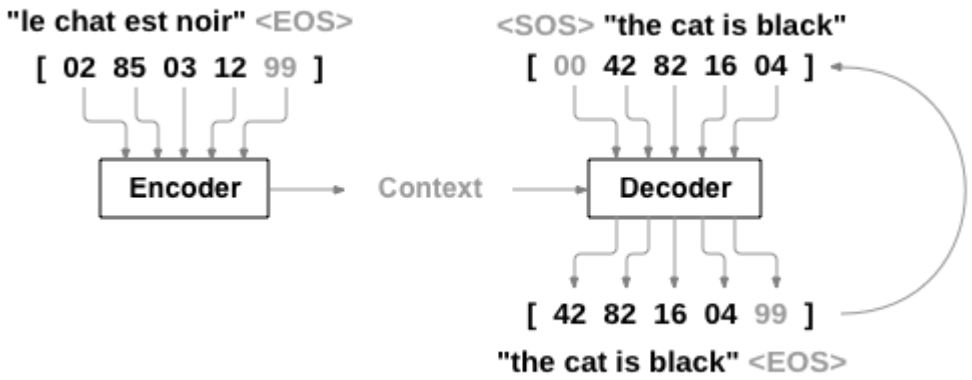
Challenges of Statistical Machine Translation

- Lots of feature engineering
  - Hand-crafted features used to capture peculiar language phenomena
- Limited scalability
  - Requires a large knowledge base
    - E.g., tables of equivalent phrases
- Lots of human efforts
  - Model maintenance
- Very limited portability
  - Ad hoc for a source/target language pair

Koehn, 2009 Statistical Machjine Translation

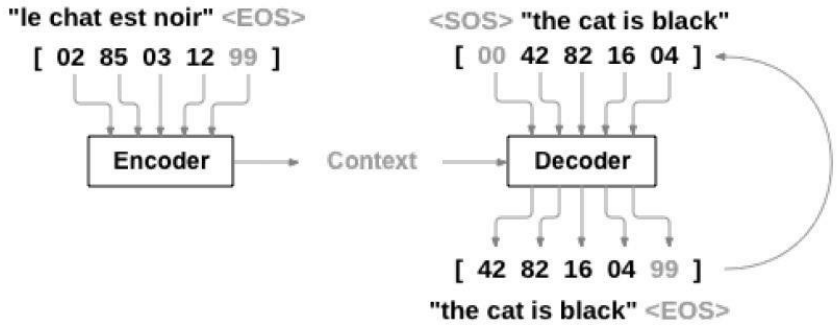
Neural Machine translation

- End-to-end Neural Network training on a large-scale bilingual corpus
  - Leverage Deep Learning architectures
- Sequence-to-sequence models (seq2seq)
  - End-to-End backpropagation

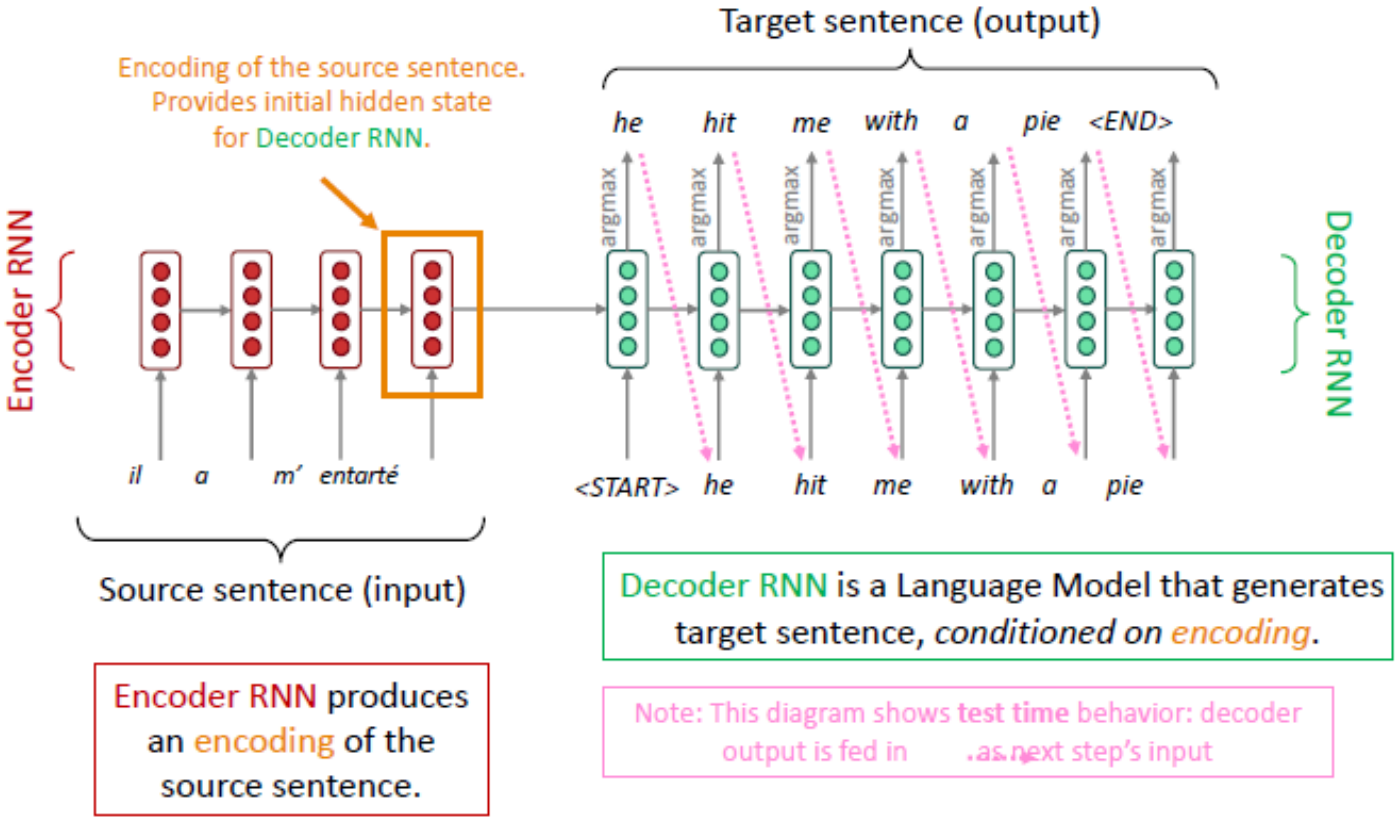


Neural Machine translation: the encoder decoder architecture

- Input and output are both variable-length sequences
- Encoder: it transforms the input into the encoded vector
  - The Encoder RNN produces an encoding of the source sentence
  - The vector encapsulates the information for all input elements in order to help the decoder make accurate predictions
- Decoder: maps the encoded state of a fixed shape to a variable-length sequence
  - The Decoder RNN is a Language Model that generates the target sentences conditioned on encoding
  - It codes the state to generate the translated sequence token by token as the output



Seq2Seq



Christopher Manning. Natural Language Processing with Deep Learning CS224N/Ling284

Seq2Seq

- Seq2Seq models are suitable for various NLP tasks
  - **Machine Translation**
    - From source sentence to target sentence
  - **Summarization**
    - from long text to shorter text
  - **Dialogue**
    - from the previous utterance to the next one
  - **Parsing**
    - from the input raw text to the parsed output
  - **Code generation**
    - from natural language to a programming language

Seq2Seq

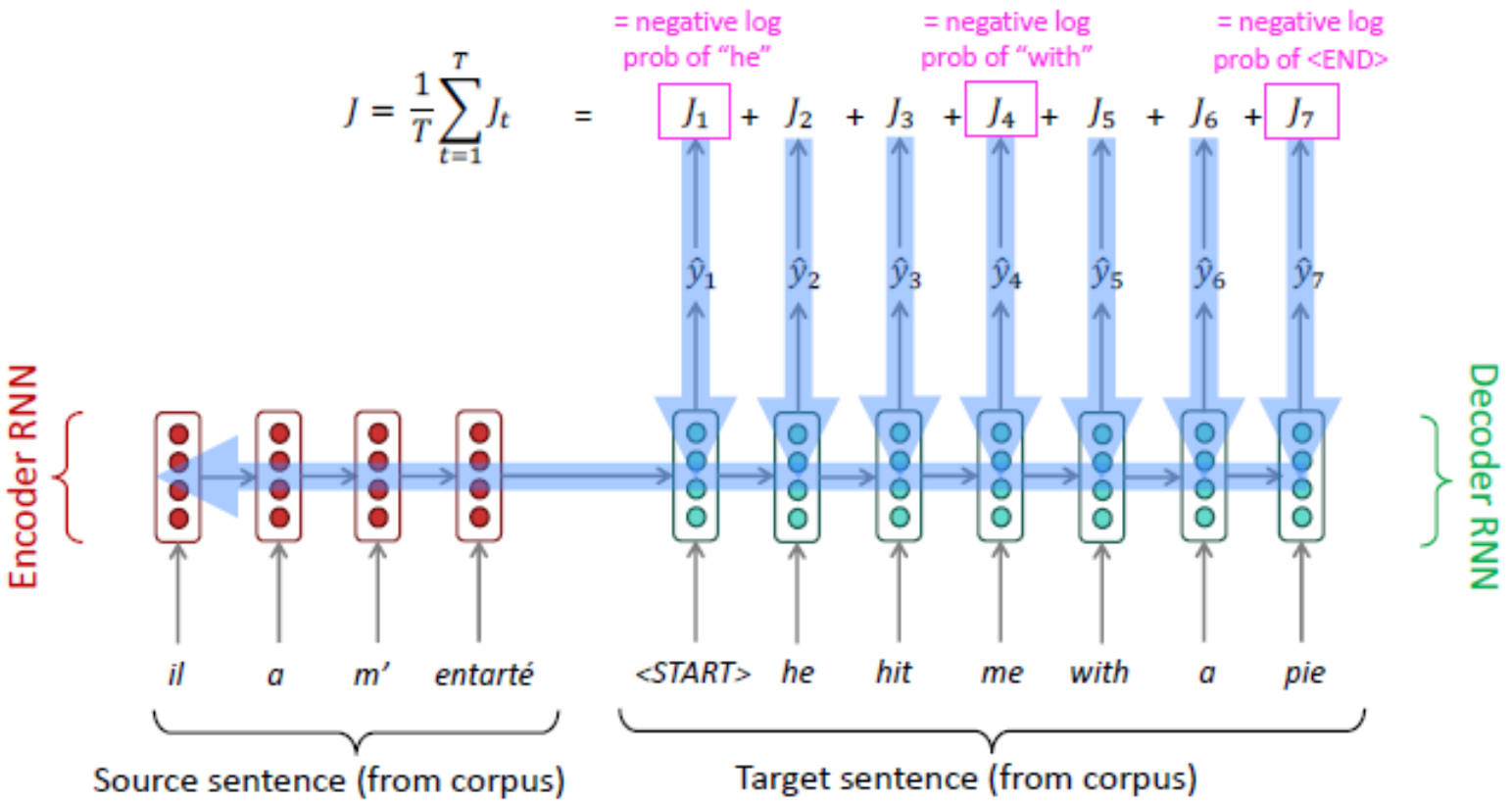
- Seq2Seq is a conditional language model
  - **Language model**
    - The decoder predicts the next word of the target sentence
  - **Conditional**
    - Predictions are conditioned on the source sentence

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

Probability of next target word, given target words so far and source sentence *x*

Christopher Manning. Natural Language Processing with Deep Learning CS224N/Ling284

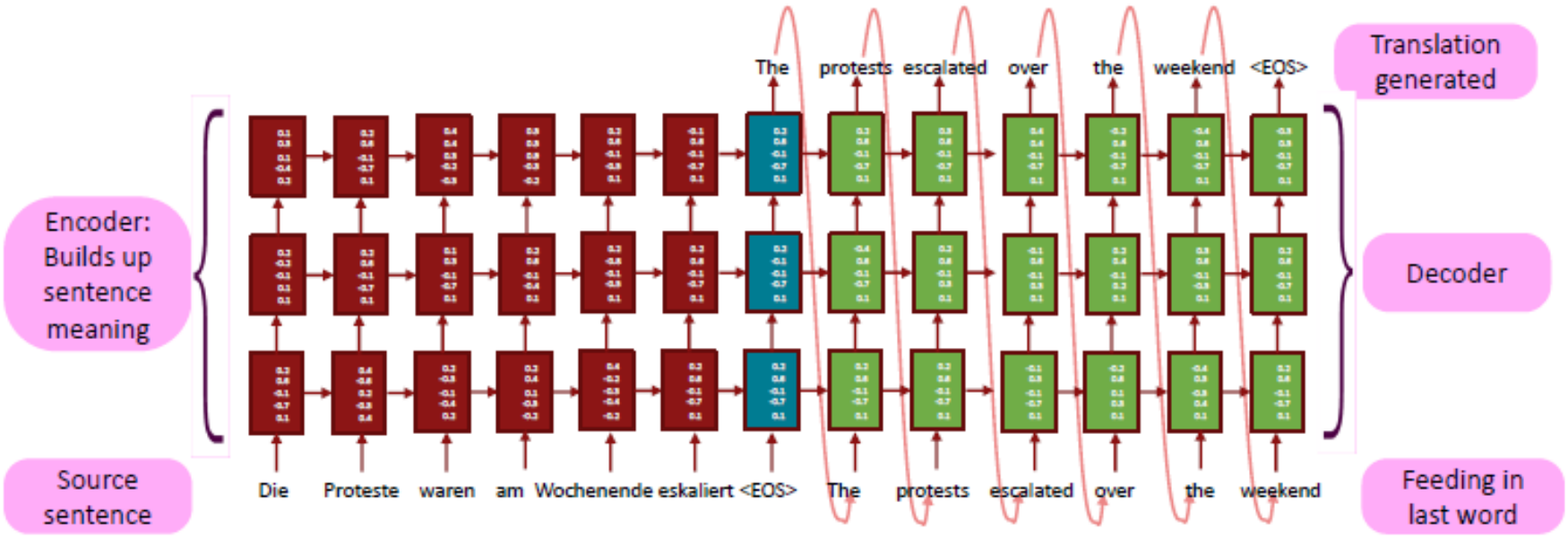
Seq2Seq: end-to-end optimization



Christopher Manning. Natural Language Processing with Deep Learning CS224N/Ling284

Multi-layer Encoder-Decoder

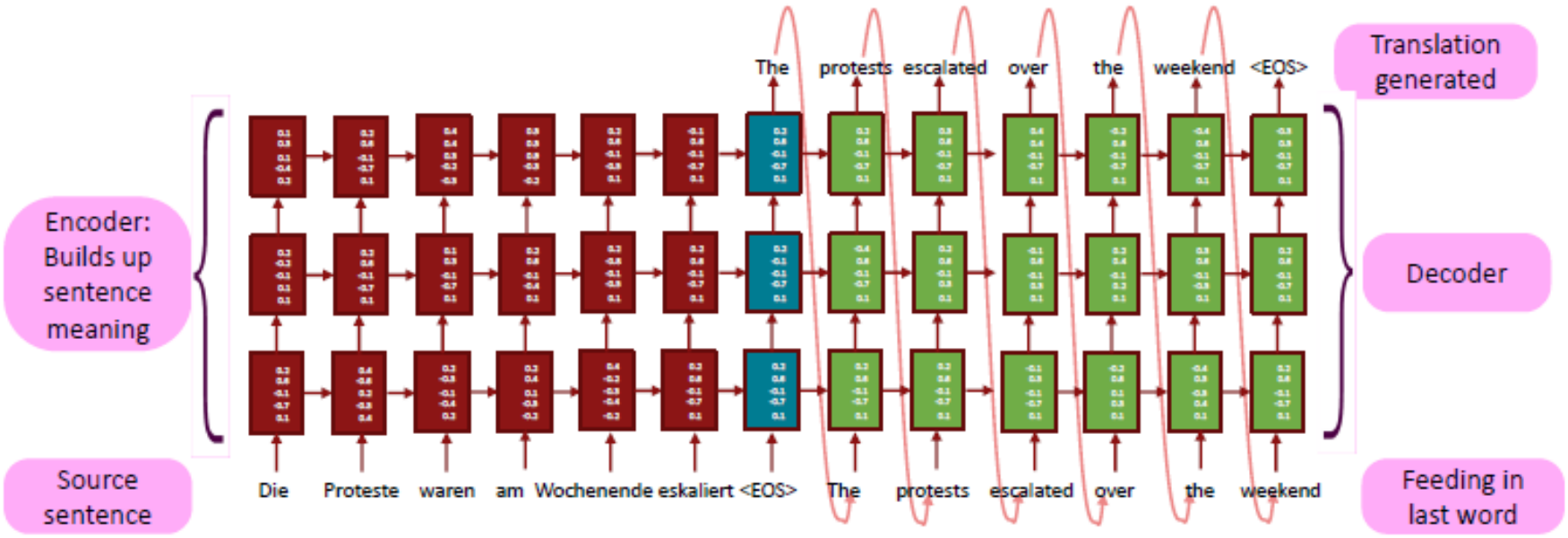
- Stacked RNNs
  - "Deep" not only in the temporal dimension
  - Apply multiple RNNs
  - The hidden states from RNN layer i are the input of RNN layer i+1



Massive exploration of Neural Machine Translation Architectures. Britz et al. 2007

Multi-layer Encoder-Decoder

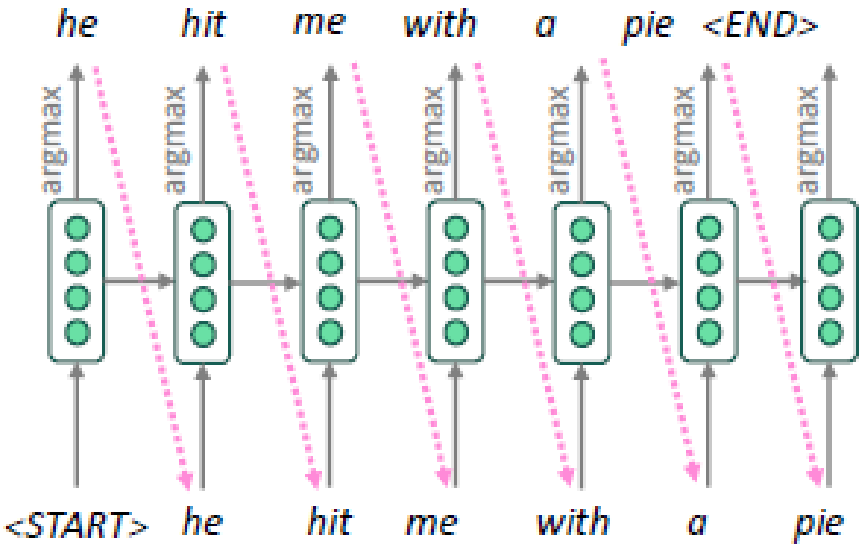
- Stacked RNNs
  - The lower RNNs compute lower-level features
  - The higher RNNs compute higher-level ones



Massive exploration of Neural Machine Translation Architectures. Britz et al. 2007

Greedy encoding

- Decode the target sentence by taking arg max on each step of the decoder
  - Take the most likely word on each step
  - No way to undo decisions



Massive exploration of Neural Machine Translation Architectures. Britz et al. 2007

Exhaustive encoding

- Find the length-T best translation by maximizing

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

- Compute all the possible sequences
  - Track all possible partial translations
  - Computationally intractable!
    - $O(V^T)$, where V is the vocabulary size

Massive exploration of Neural Machine Translation Architectures. Britz et al. 2007

Beam search encoding

- Keep track of the k most likely translations
  - K is the beam size (typically, between 5 and 10)
- Define the hypotheses y_1, ..., y_t and their scores
- Log probability

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

- Search for the top ranked hypotheses by considering only the top-k translations at each step

Massive exploration of Neural Machine Translation Architectures. Britz et al. 2007

Beam search encoding

- No guarantee to find the optimal solution
- Efficiency is significantly higher than exhaustive encoding
- Key steps
  - Calculate the probability distribution of the next word
  - Take k words and compute the scores
  - For each of the k hypothesis find top-k next words and calculate the scores
  - Among the K^2 available hypotheses keep the k hypothesis with highest score
  - Backtrack to obtain the full hypothesis
  - Stop decoding when the model produces an <END> token
    - Alternative stopping criteria
      - Until a predefined timestep *T* is reached
      - Until at least a predefined number *n* of hypotheses have been completed

Massive exploration of Neural Machine Translation Architectures. Britz et al. 2007

Beam search encoding

- Each hypotheses has a score

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

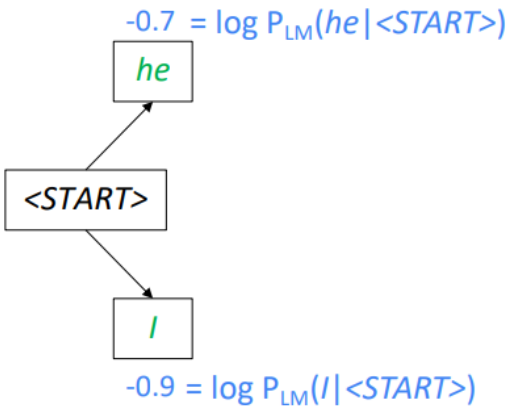
- Longer hypotheses have lower scores
  - Normalize by length:

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

Massive exploration of Neural Machine Translation Architectures. Britz et al. 2007

Beam search encoding

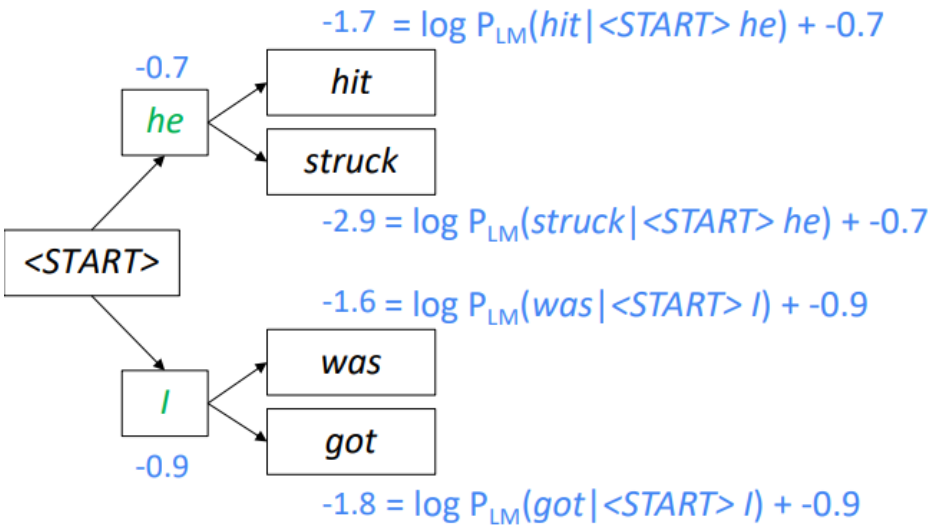
Beam size = k = 2. Blue numbers = $\text{score}(y_1, \ldots, y_t) = \sum_{i=1}^{t} \log P_{\text{LM}}(y_i | y_1, \ldots, y_{i-1}, x)$



Christopher Manning. Natural Language Processing with Deep Learning CS224N/Ling284

Beam search encoding

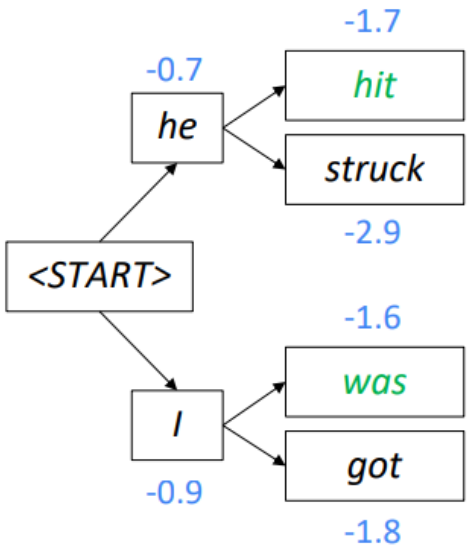
Beam size = k = 2. Blue numbers = $\text{score}(y_1, \ldots, y_t) = \sum_{i=1}^{t} \log P_{\text{LM}}(y_i | y_1, \ldots, y_{i-1}, x)$



Christopher Manning. Natural Language Processing with Deep Learning CS224N/Ling284

Beam search encoding

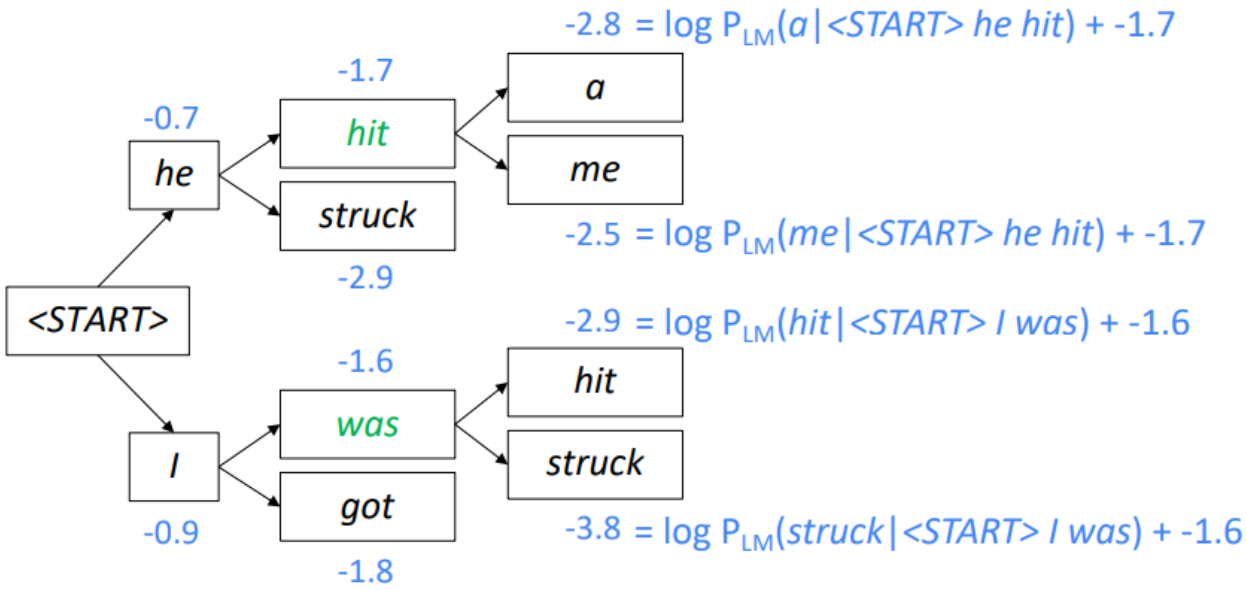
Beam size = k = 2. Blue numbers = $\text{score}(y_1, \ldots, y_t) = \sum_{i=1}^{t} \log P_{\text{LM}}(y_i | y_1, \ldots, y_{i-1}, x)$



Christopher Manning. Natural Language Processing with Deep Learning CS224N/Ling284

Beam search encoding

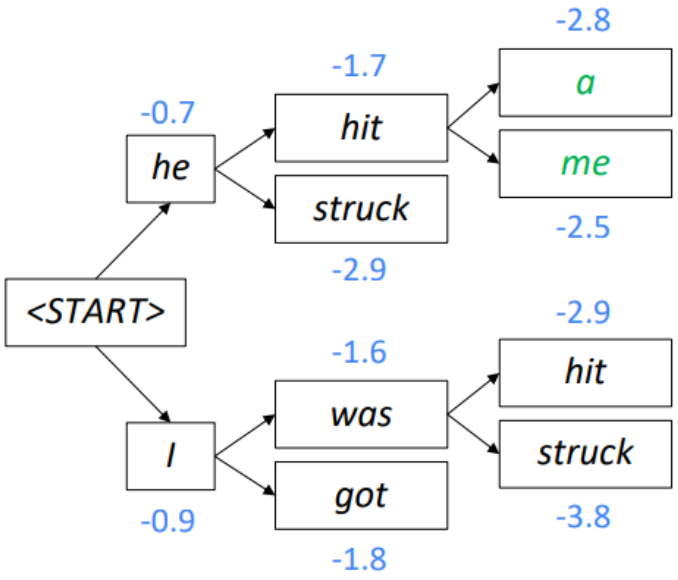
Beam size = k = 2. Blue numbers = $\text{score}(y_1, \ldots, y_t) = \sum_{i=1}^{t} \log P_{\text{LM}}(y_i | y_1, \ldots, y_{i-1}, x)$



Christopher Manning. Natural Language Processing with Deep Learning CS224N/Ling284

Beam search encoding

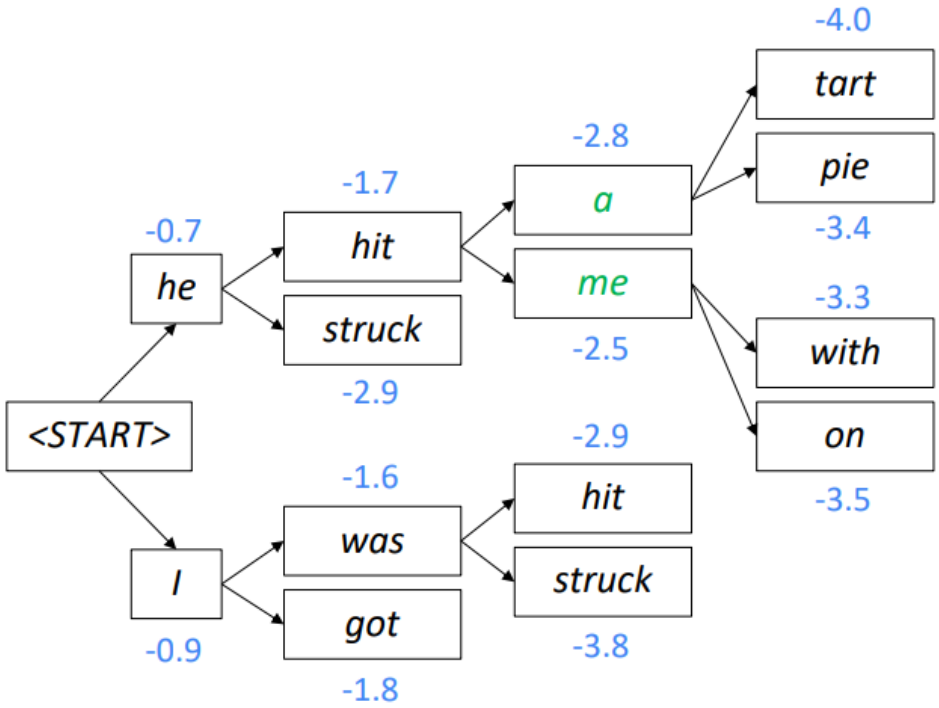
Beam size = k = 2. Blue numbers = $\text{score}(y_1, \ldots, y_t) = \sum_{i=1}^{t} \log P_{\text{LM}}(y_i | y_1, \ldots, y_{i-1}, x)$



Christopher Manning. Natural Language Processing with Deep Learning CS224N/Ling284

Beam search encoding

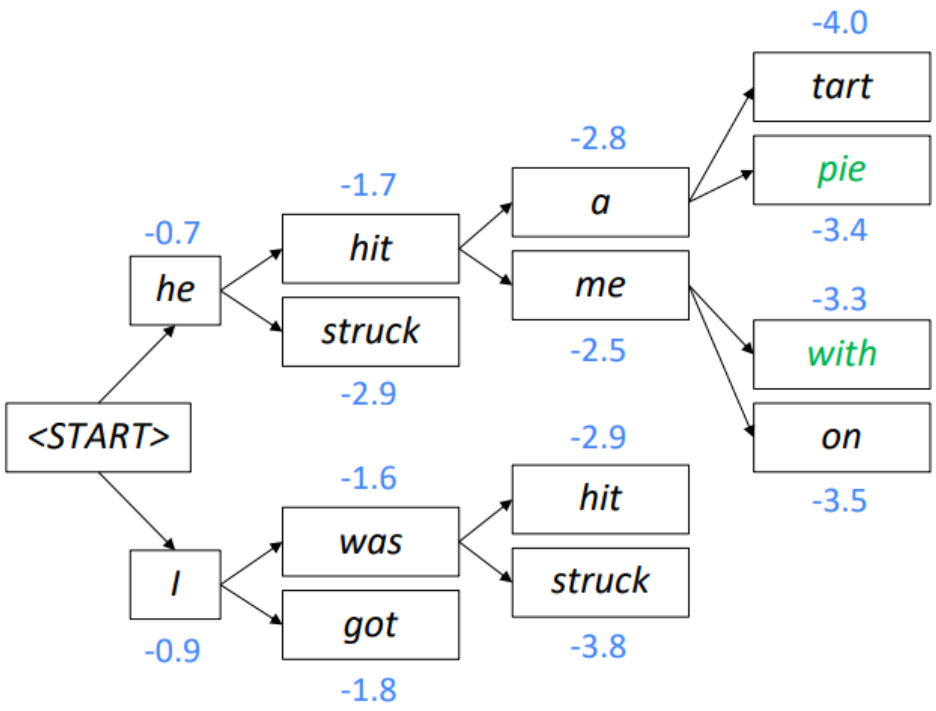
Beam size = k = 2. Blue numbers = $\text{score}(y_1, \ldots, y_t) = \sum_{i=1}^{t} \log P_{\text{LM}}(y_i | y_1, \ldots, y_{i-1}, x)$



Christopher Manning. Natural Language Processing with Deep Learning CS224N/Ling284

Beam search encoding

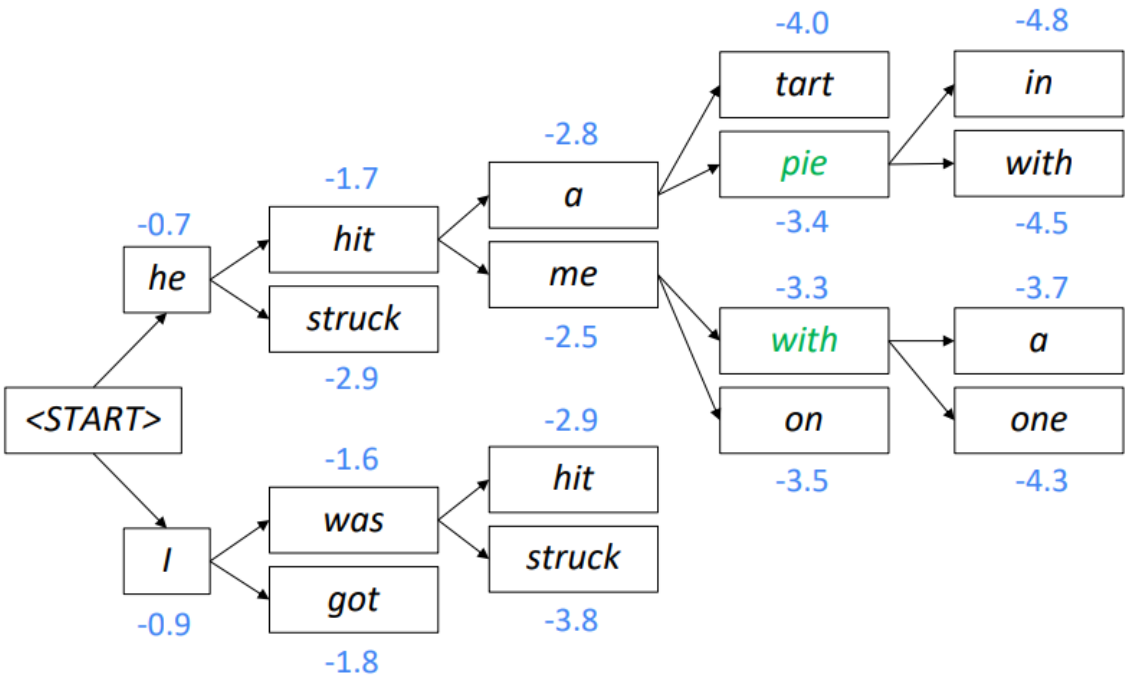
Beam size = k = 2. Blue numbers = $\text{score}(y_1, \ldots, y_t) = \sum_{i=1}^{t} \log P_{\text{LM}}(y_i | y_1, \ldots, y_{i-1}, x)$



Christopher Manning. Natural Language Processing with Deep Learning CS224N/Ling284

Beam search encoding

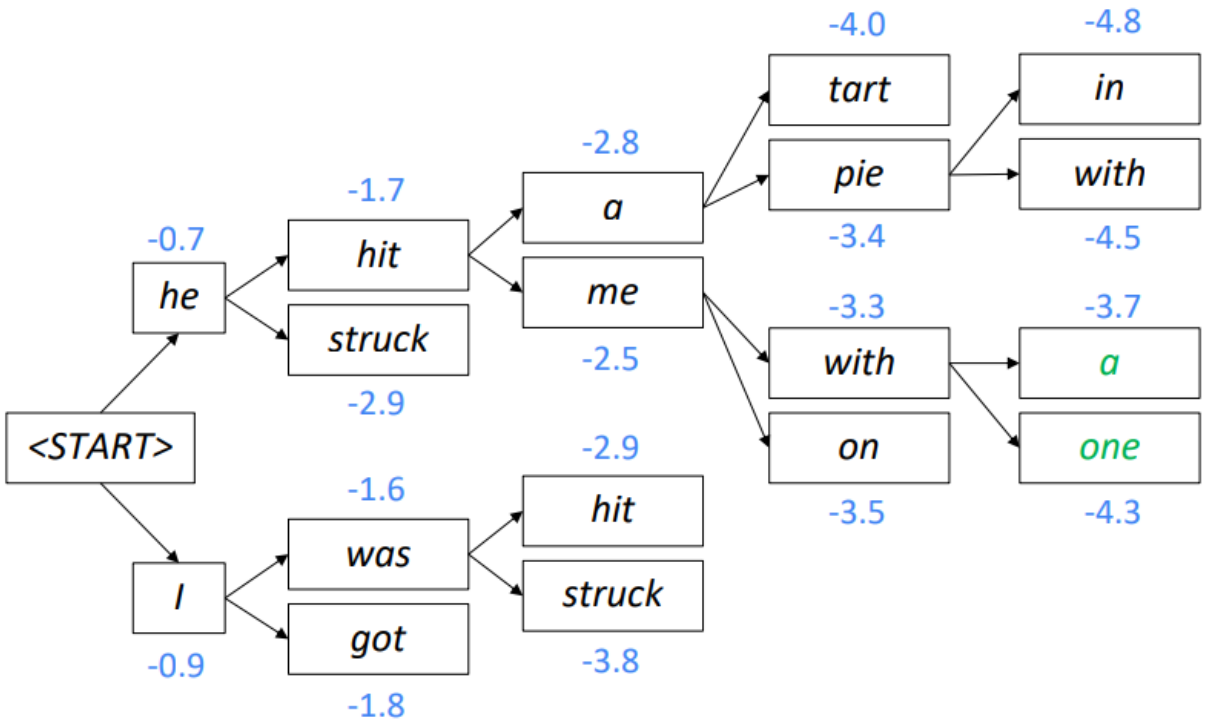
Beam size = k = 2. Blue numbers = $\text{score}(y_1, \ldots, y_t) = \sum_{i=1}^{t} \log P_{\text{LM}}(y_i | y_1, \ldots, y_{i-1}, x)$



Christopher Manning. Natural Language Processing with Deep Learning CS224N/Ling284

Beam search encoding

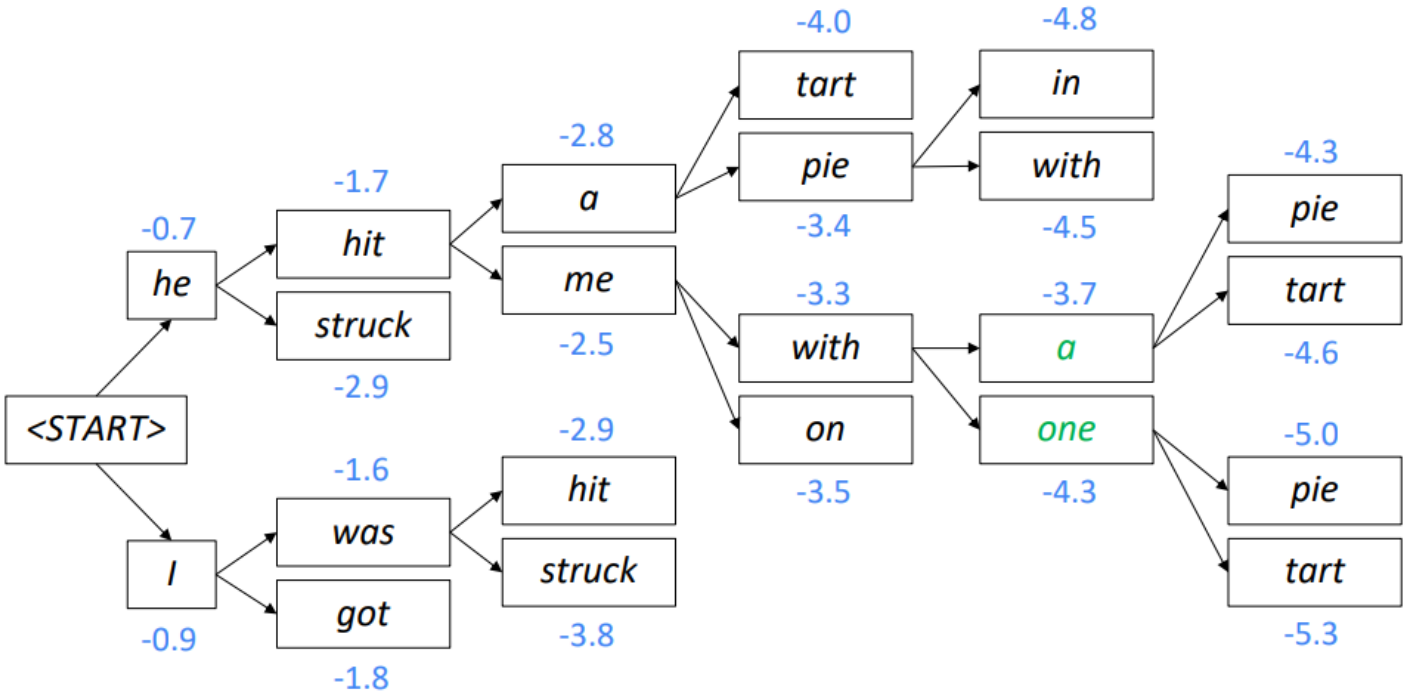
Beam size = k = 2. Blue numbers = $\text{score}(y_1, \ldots, y_t) = \sum_{i=1}^{t} \log P_{\text{LM}}(y_i | y_1, \ldots, y_{i-1}, x)$



Christopher Manning. Natural Language Processing with Deep Learning CS224N/Ling284

Beam search encoding

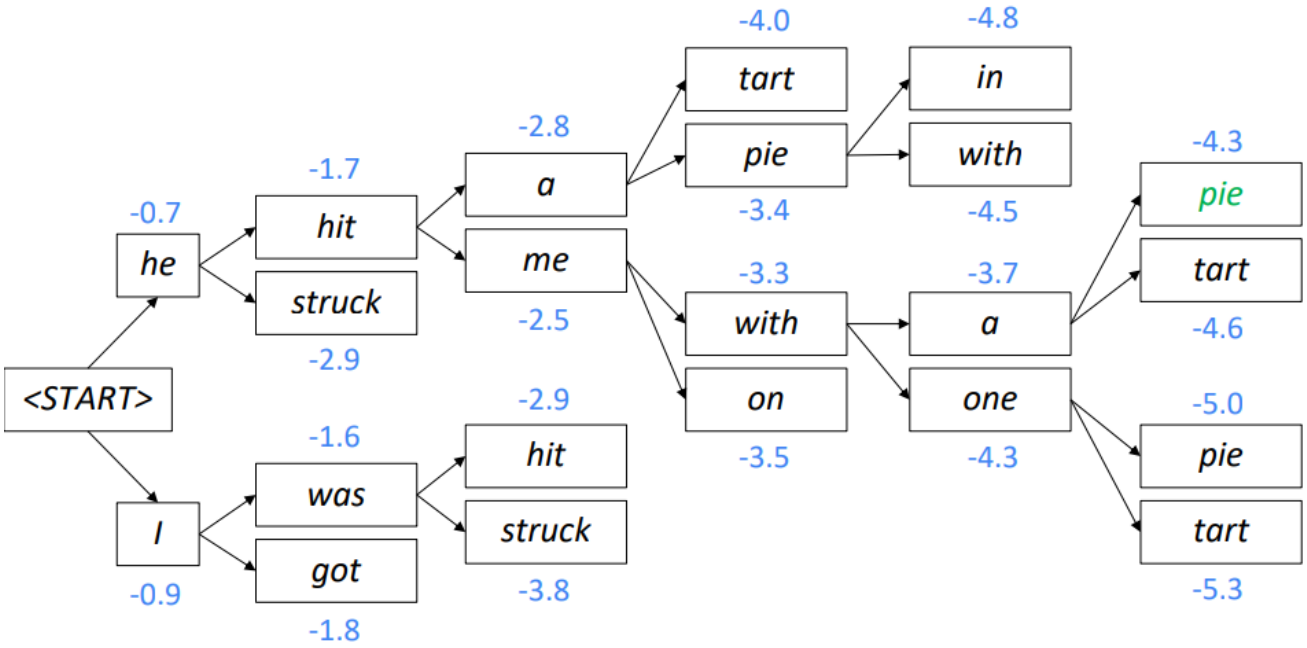
Beam size = k = 2. Blue numbers = $\text{score}(y_1, \ldots, y_t) = \sum_{i=1}^{t} \log P_{\text{LM}}(y_i | y_1, \ldots, y_{i-1}, x)$



Christopher Manning. Natural Language Processing with Deep Learning CS224N/Ling284

Beam search encoding

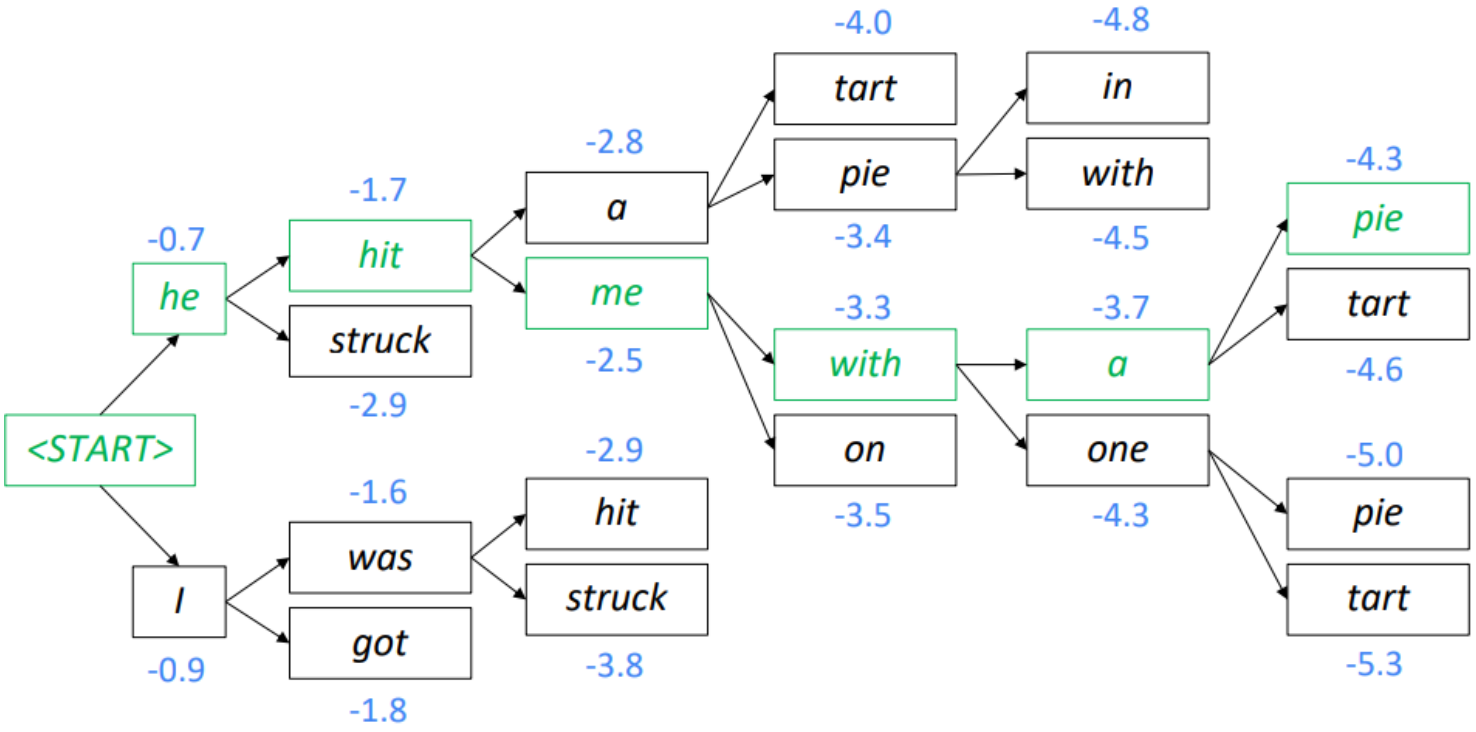
Beam size = k = 2. Blue numbers = $\text{score}(y_1, \ldots, y_t) = \sum_{i=1}^{t} \log P_{\text{LM}}(y_i | y_1, \ldots, y_{i-1}, x)$



Christopher Manning. Natural Language Processing with Deep Learning CS224N/Ling284

Beam search encoding

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> he hit me with a pie <END>

Christopher Manning. Natural Language Processing with Deep Learning CS224N/Ling284

BilinguaL Evaluation Under study (BLEU)

- Metric to compare the machine-generated translation to one or more human-written summaries
  - Based on co-occurrence-based metrics
    - N-gram precision
  - Hypothesis: a single translation per person
  - Low N-gram overlap with the human translation

BLEU: a method for automatic evaluation of Machine Translation. Papineni et al., 2002

BilinguaL Evaluation Under study (BLEU)

- Compute the geometric average of the modified n-gram precisions p_n using n-grams up to length N and positive weights w_n summing to one
  - Baseline: N=4, $w_n = 1/N$

$$p_n = \frac{\sum_{C \in \{Candidates\}} \sum_{n\text{-}gram \in C} Count_{clip}(n\text{-}gram)}{\sum_{C' \in \{Candidates\}} \sum_{n\text{-}gram' \in C'} Count(n\text{-}gram')}$$

BLEU: a method for automatic evaluation of Machine Translation. Papineni et al., 2002

BilinguaL Evaluation Under study (BLEU)

- Let c be the length of the candidate translation and r be the effective reference corpus length
- Compute the brevity penalty BP

$$\text{BP} = \begin{cases} 1 & \text{if } c > r \\ e^{(1-r/c)} & \text{if } c \leq r \end{cases}$$

BLEU: a method for automatic evaluation of Machine Translation. Papineni et al., 2002

BilinguaL Evaluation Under study (BLEU)

- Then

$$\text{BLEU} = \text{BP} \cdot \exp\left(\sum_{n=1}^{N} w_n \log p_n\right)$$

- The ranking is usually computed in the log domain

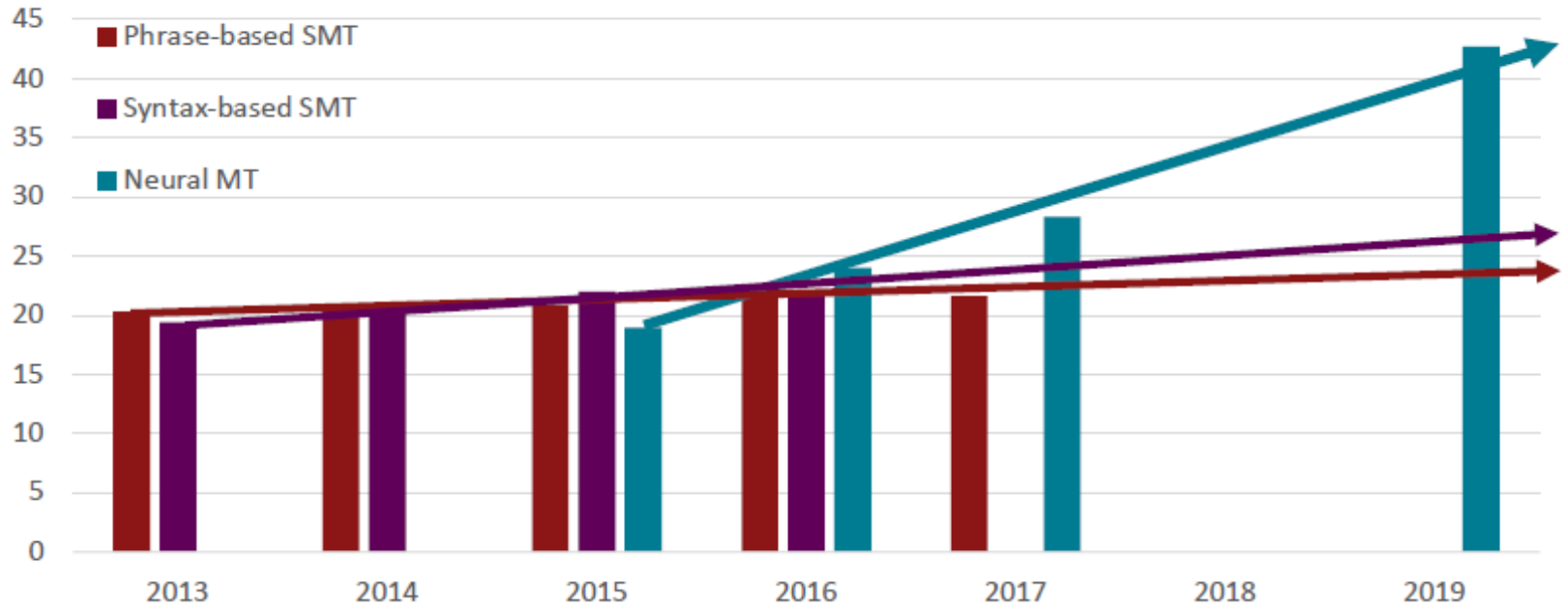
$$\log \text{BLEU} = \min(1 - \frac{r}{c}, 0) + \sum_{n=1}^{N} w_n \log p_n.$$

BLEU: a method for automatic evaluation of Machine Translation. Papineni et al., 2002

Additional reading on BLEU

- Kishore Papineni, Salim Roukos, Todd Ward, Wei-Jing Zhu. BLEU: a Method for Automatic Evaluation of Machine Translation. ACL 2002: 311-318
- Download and read the paper: https://aclanthology.org/P02-1040.pdf

Machine Translation evolution over time



Pros of Neural Machine Translation

- Text fluency
- Better contextualization
- No sub-components to be optimized
  - End-to-end training
- Limited human engineering effort
- Same method for all language pairs

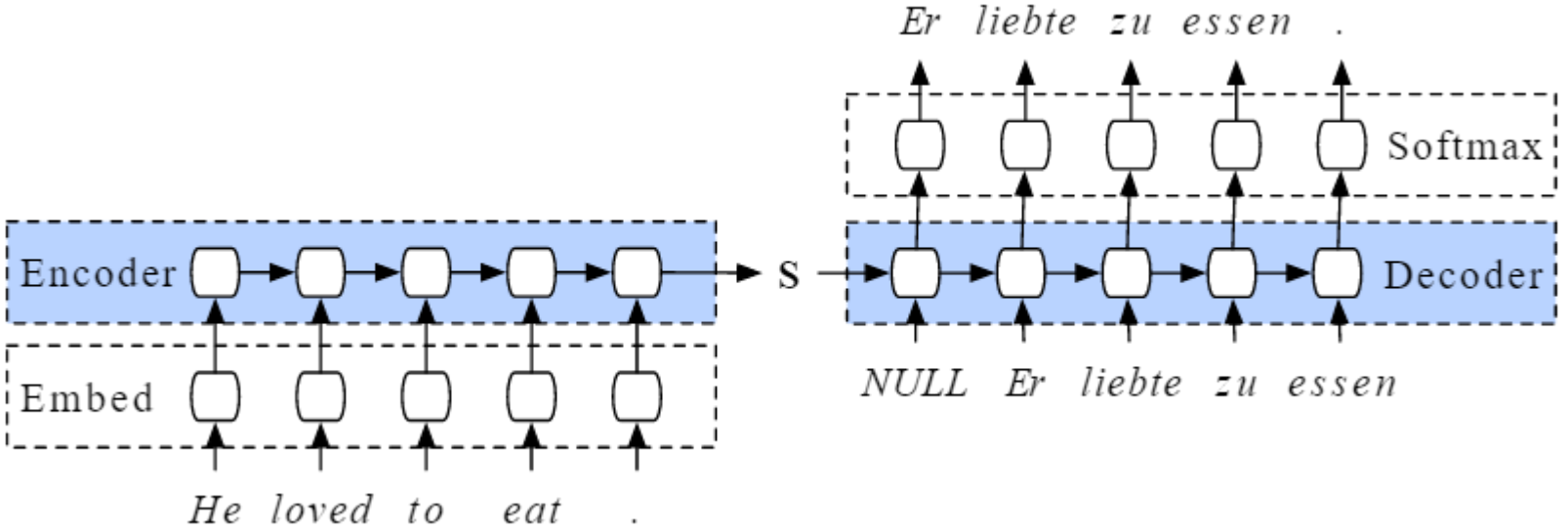
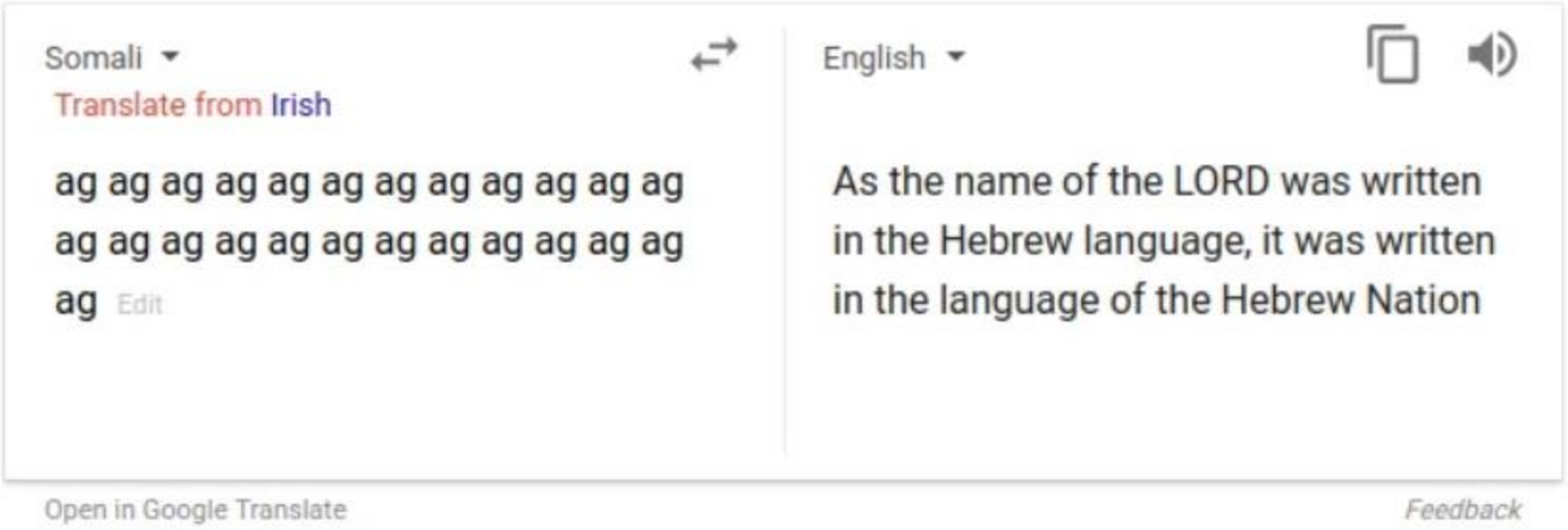


Image taken for smerity.com (latest access: June 2021)

Cons of Neural Machine Translation

- Hard to debug
- Hard to control
- Out-of-vocabulary words
- Domain mismatch from train to test data
- Low-resource language pairs
- Pronoun resolution errors
- Morphological agreement errors

Hard to control



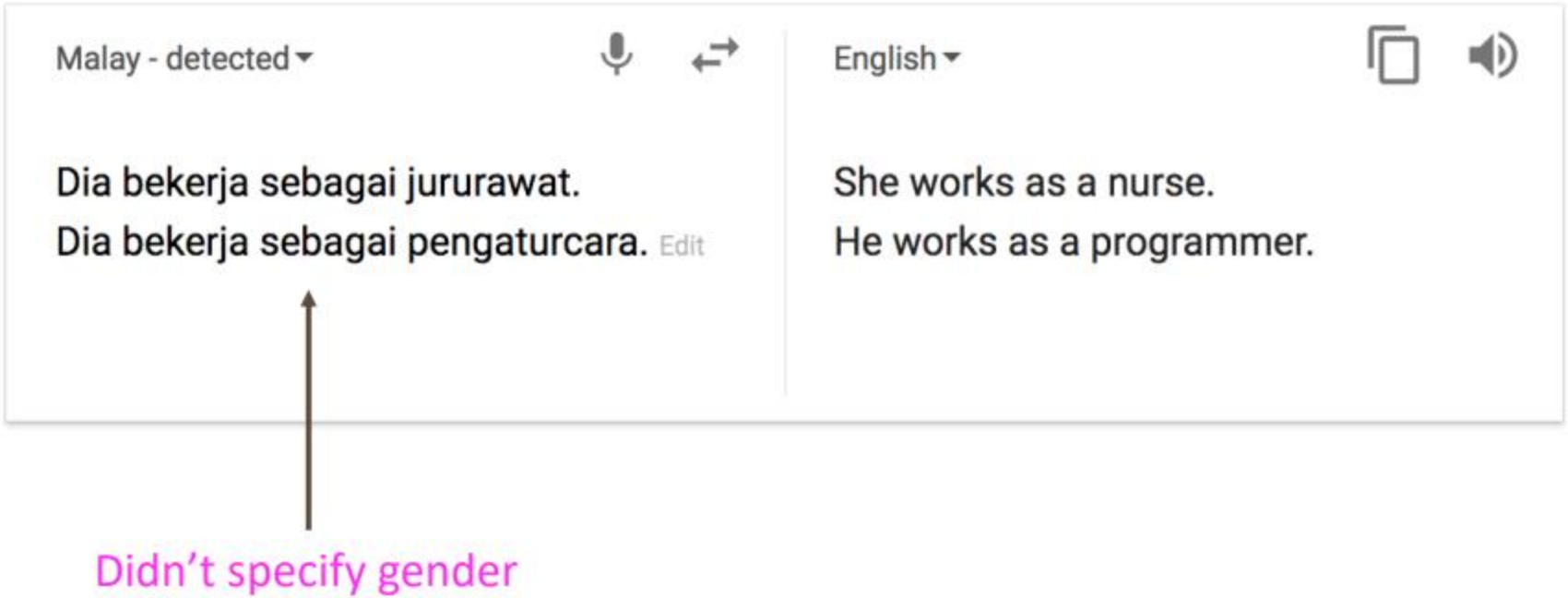
Christopher Manning. Natural Language Processing with Deep Learning CS224N/Ling284

Common sense

?

Christopher Manning. Natural Language Processing with Deep Learning CS224N/Ling284

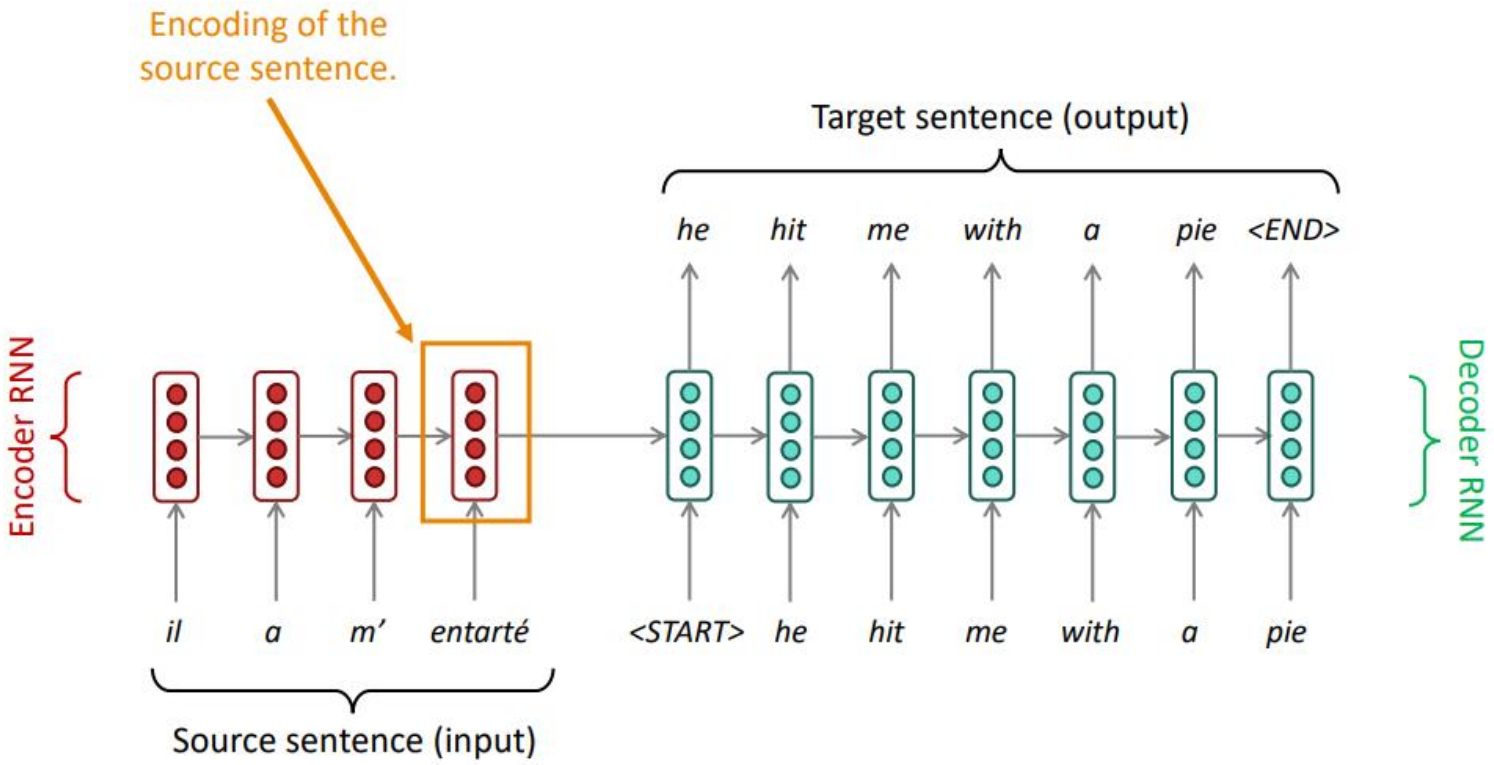
Gender bias



Christopher Manning. Natural Language Processing with Deep Learning CS224N/Ling284

Transformers in Neural Machine Translation

- Encoding of the source allows us to capture all information about the source sentence
  - Given a set of vector values, and a vector query, attention is a technique to compute a weighted sum of the values, dependent on the query



Christopher Manning. Natural Language Processing with Deep Learning CS224N/Ling284

Attention in a nutshell

- Given a set of vector values, and a vector query, attention is a technique to compute a weighted sum of the values, dependent on the query
- The weighted sum is a selective summary of the information contained in the values, where the query determines which values to focus on.
- Attention is a way to obtain a fixed-size representation of an arbitrary set of representations (the values), dependent on some other representation (the query)

Christopher Manning. Natural Language Processing with Deep Learning CS224N/Ling284

Attention in Seq2Seq (1/3)

- Encoder hidden states:

$$h_1, \ldots, h_N \in \mathbb{R}^h$$

- On timestep t, we have decoder hidden state $s_t \in \mathbb{R}^h$
- The attention score on timestep t:

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

Christopher Manning. Natural Language Processing with Deep Learning CS224N/Ling284

Attention in Seq2Seq (2/3)

- To get the attention probability distribution on timestep t

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

- The attention output on timestep t is a weighted sum of the encoder hidden states:

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

Christopher Manning. Natural Language Processing with Deep Learning CS224N/Ling284

Attention in Seq2Seq (3/3)

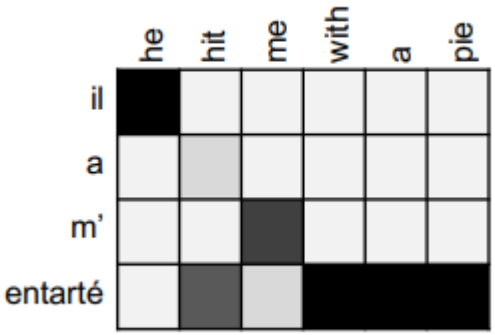
- Attention outputs are concatenated with the decoder hidden states as in a traditional seq2seq model

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

Christopher Manning. Natural Language Processing with Deep Learning CS224N/Ling284

Pros of using attention

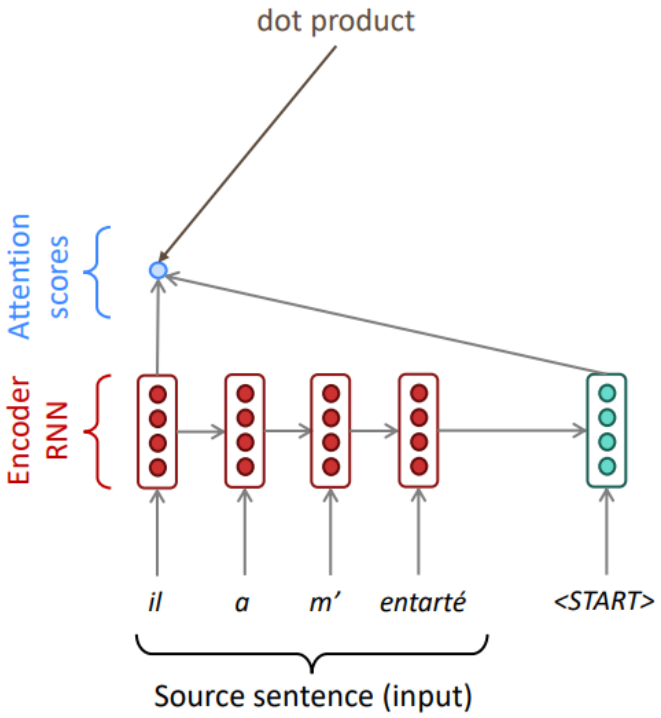
- Attention is useful for improving translation performance
  - Focus on certain parts of the source
  - Solve the bottleneck problem
    - the decoder looks directly at source
  - Overcome the vanishing gradient
    - Shortcut to faraway states
  - Saliency map



Christopher Manning. Natural Language Processing with Deep Learning CS224N/Ling284

Transformers in Neural Machine Translation

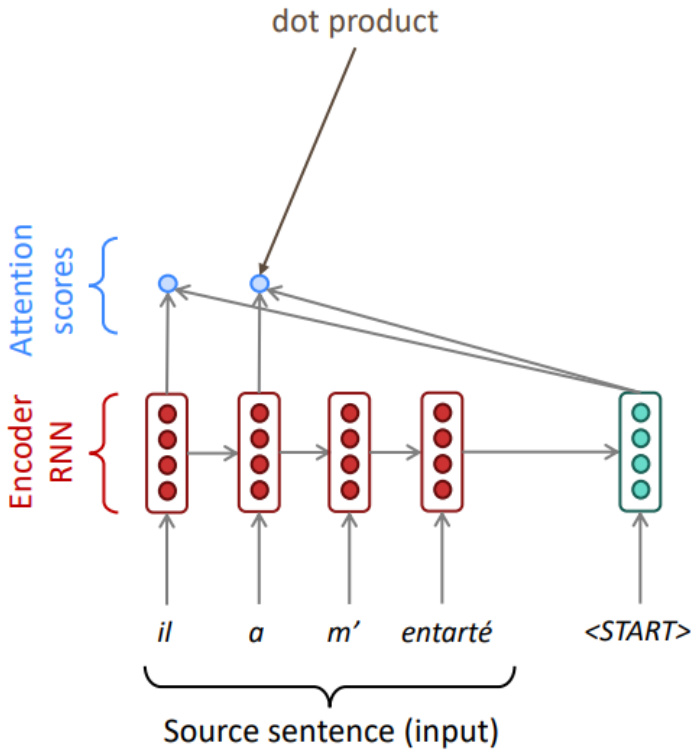
- On each decoder step, attention uses the direct connection to the encoder to attend particular tokens of the source sequence



Christopher Manning. Natural Language Processing with Deep Learning CS224N/Ling284

Transformers in Neural Machine Translation

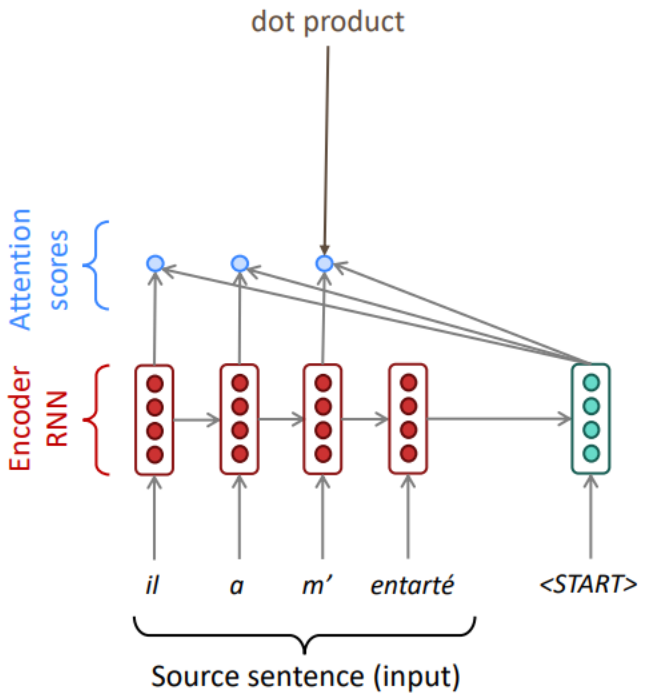
- On each decoder step, attention uses the direct connection to the encoder to attend particular tokens of the source sequence



Christopher Manning. Natural Language Processing with Deep Learning CS224N/Ling284

Transformers in Neural Machine Translation

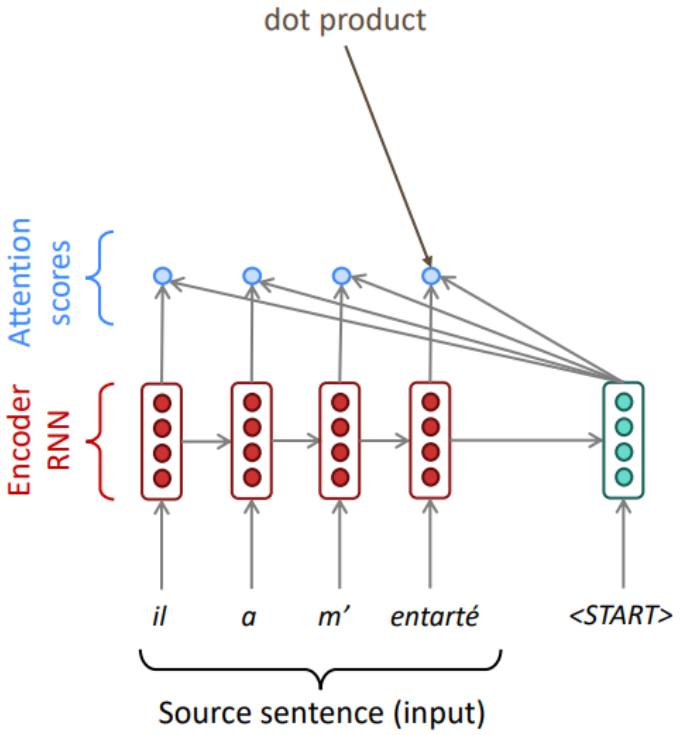
- On each decoder step, attention uses the direct connection to the encoder to attend particular tokens of the source sequence



Christopher Manning. Natural Language Processing with Deep Learning CS224N/Ling284

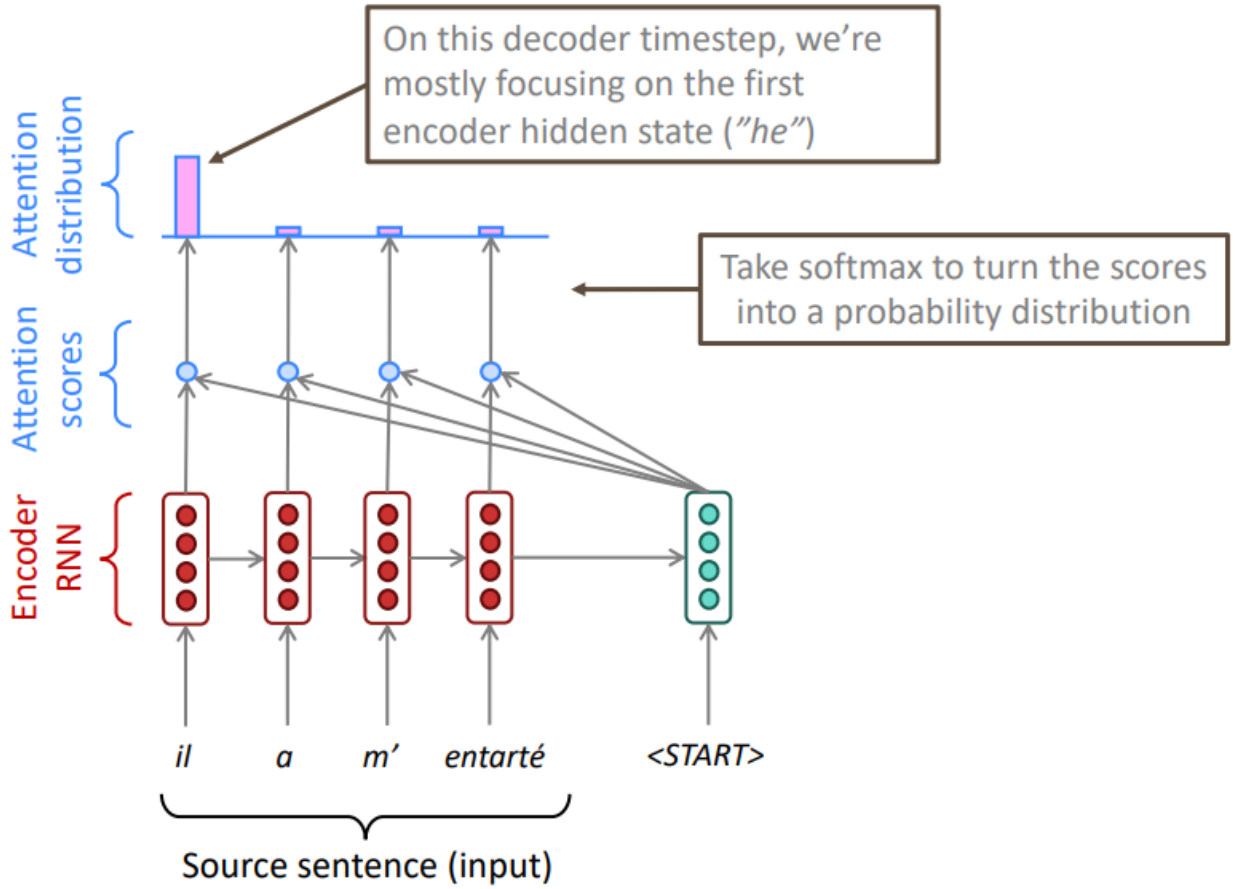
Transformers in Neural Machine Translation

- On each decoder step, attention uses the direct connection to the encoder to attend particular tokens of the source sequence



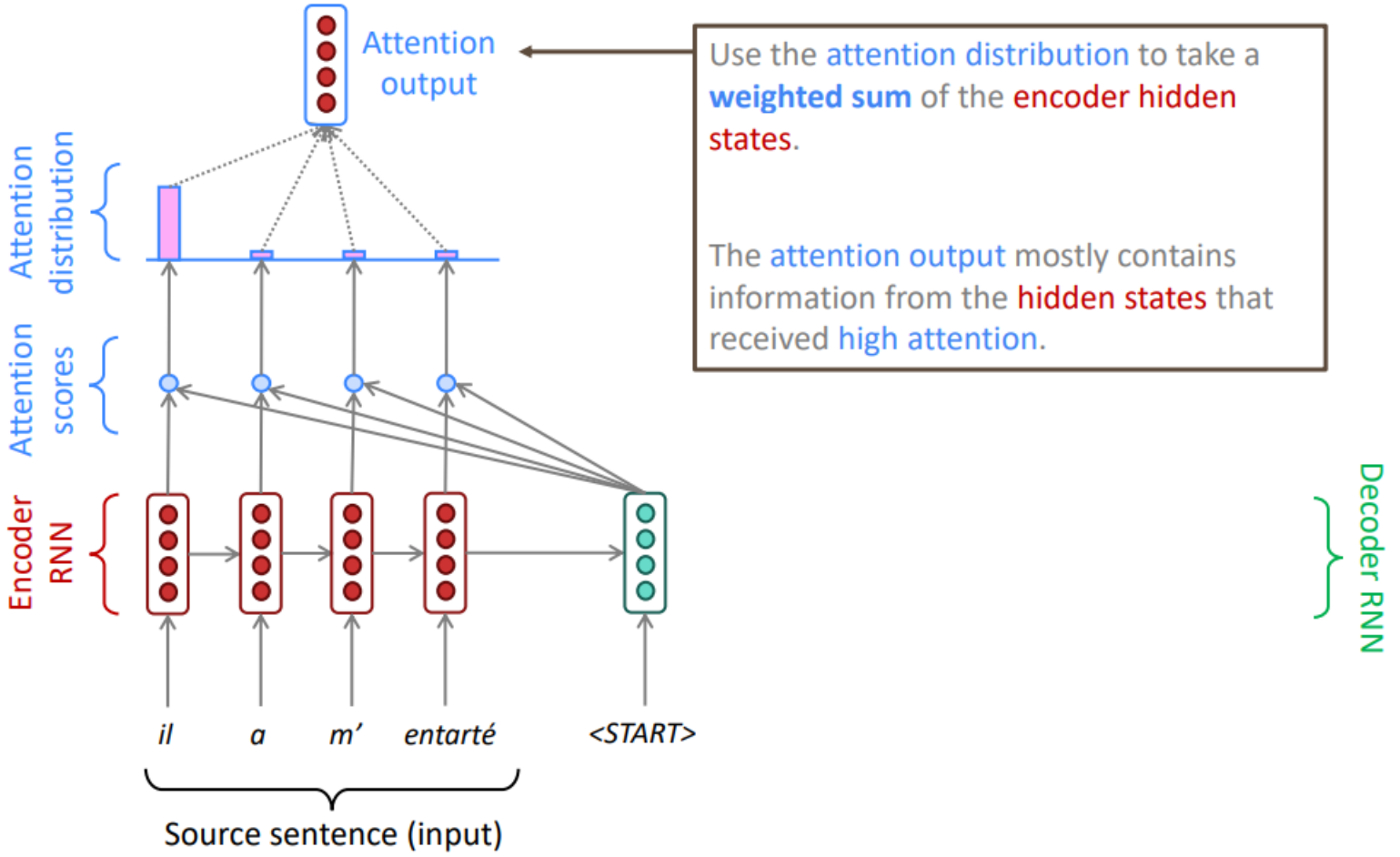
Christopher Manning. Natural Language Processing with Deep Learning CS224N/Ling284

Transformers in Neural Machine Translation



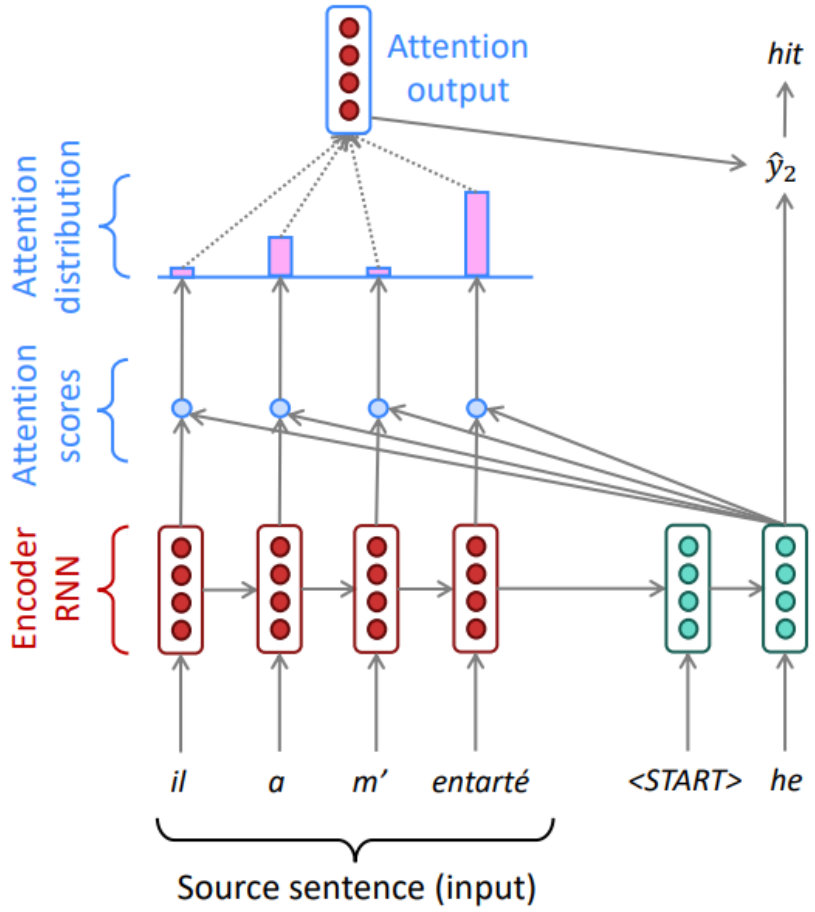
Christopher Manning. Natural Language Processing with Deep Learning CS224N/Ling284

Transformers in Neural Machine Translation



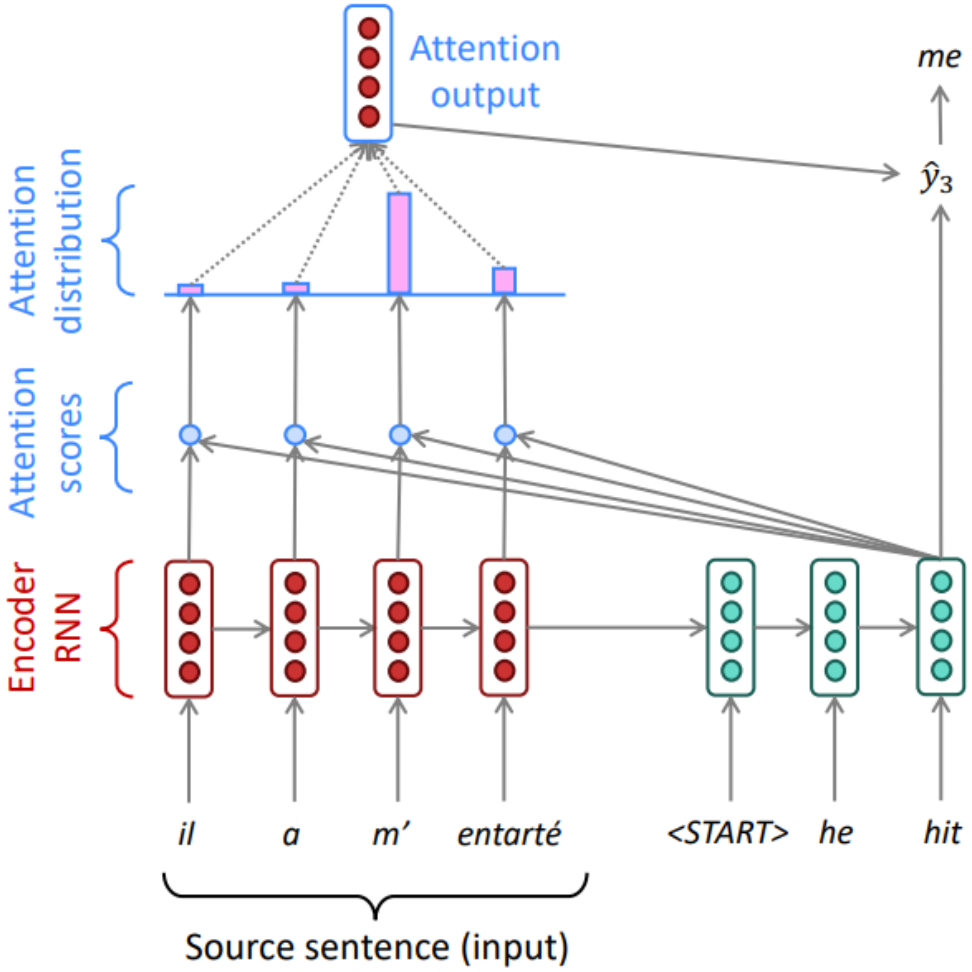
Christopher Manning. Natural Language Processing with Deep Learning CS224N/Ling284

Transformers in Neural Machine Translation



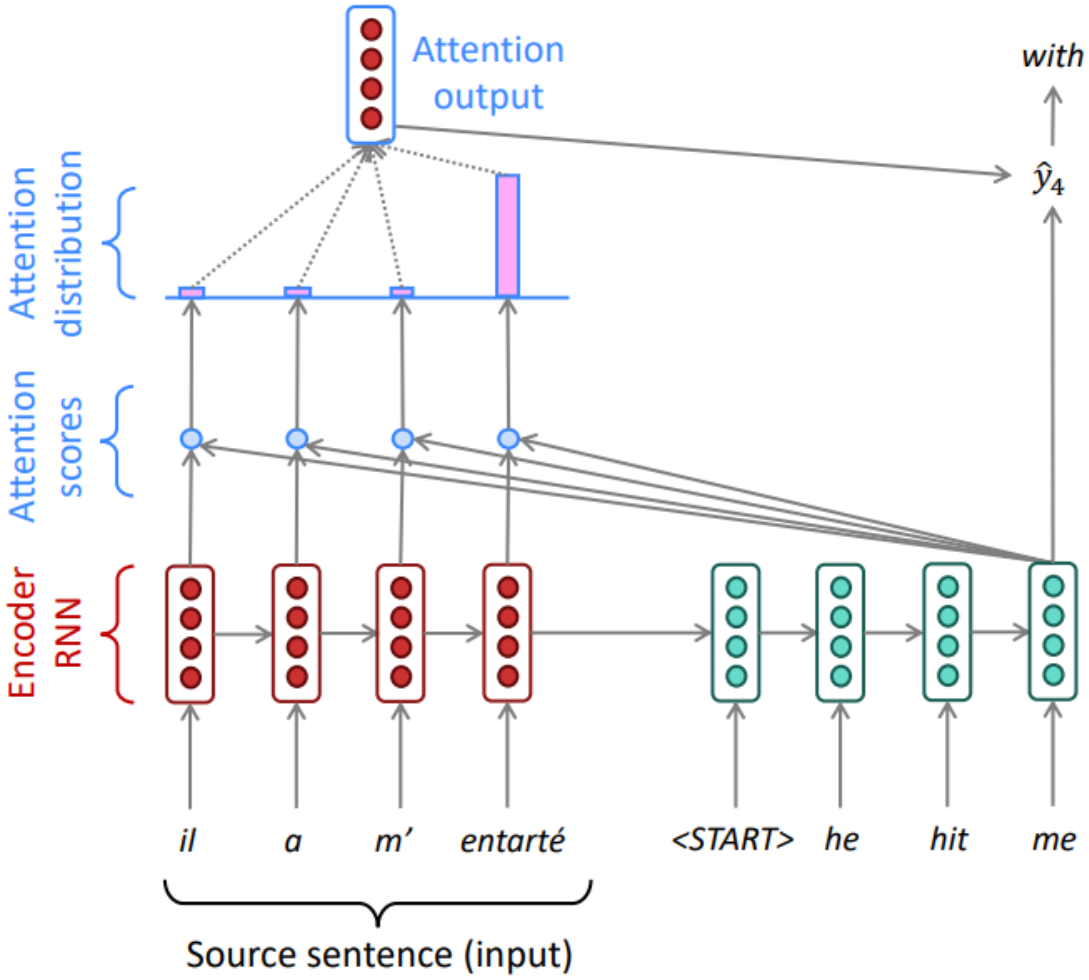
Christopher Manning. Natural Language Processing with Deep Learning CS224N/Ling284

Transformers in Neural Machine Translation



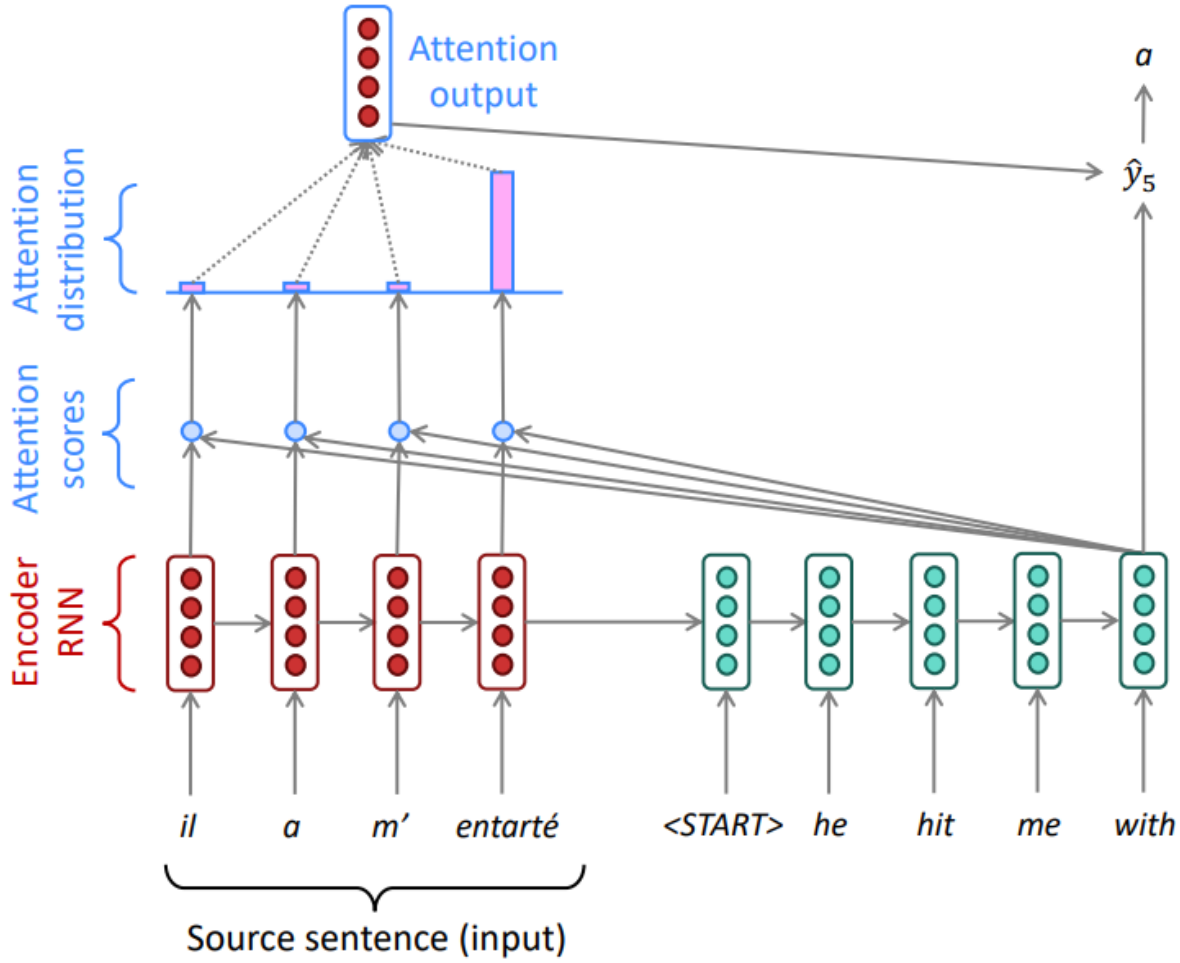
Christopher Manning. Natural Language Processing with Deep Learning CS224N/Ling284

Transformers in Neural Machine Translation



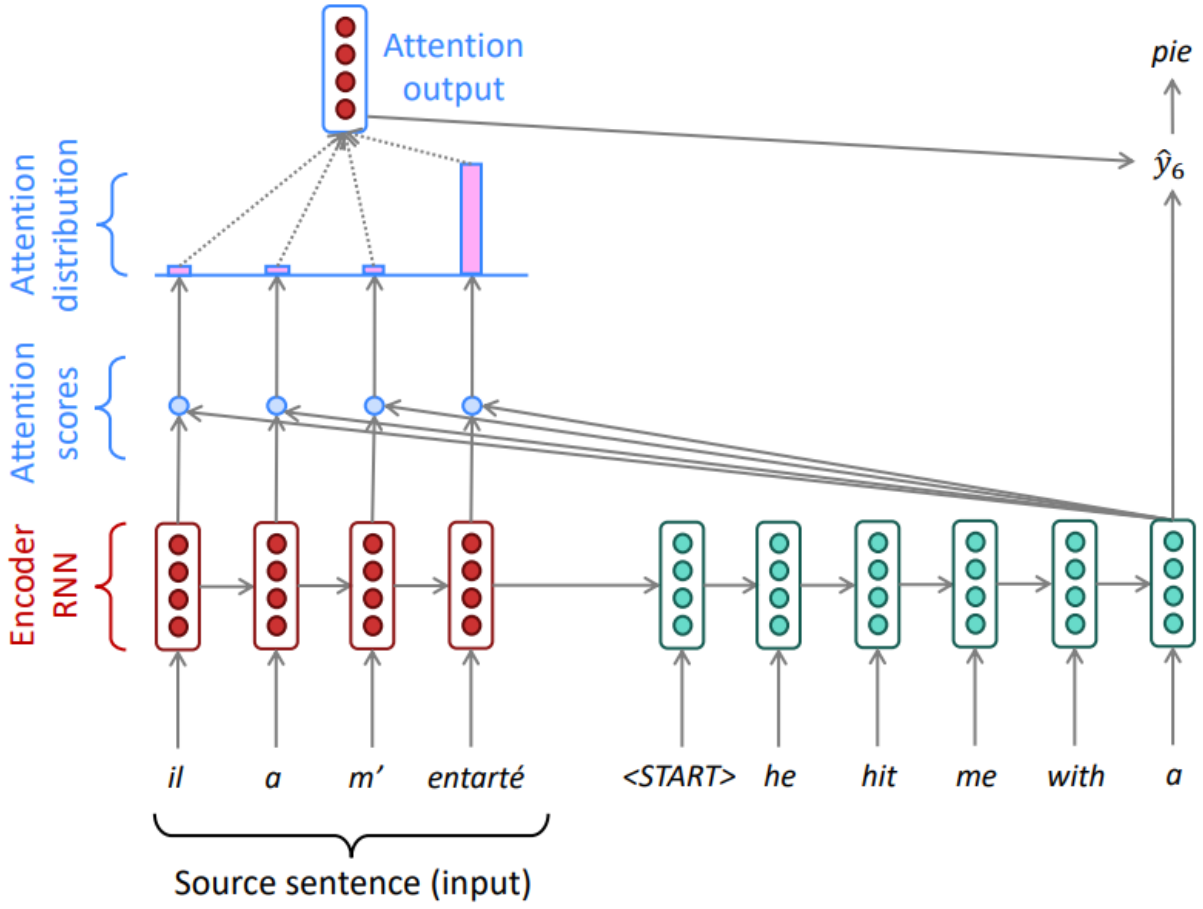
Christopher Manning. Natural Language Processing with Deep Learning CS224N/Ling284

Transformers in Neural Machine Translation



Christopher Manning. Natural Language Processing with Deep Learning CS224N/Ling284

Transformers in Neural Machine Translation



Christopher Manning. Natural Language Processing with Deep Learning CS224N/Ling284

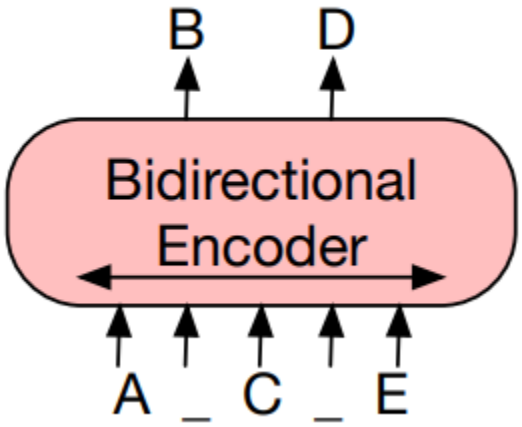
BART

- A state-of-the-art denoising autoencoder for pretraining sequence-to-sequence models
- Based on Transformers
  - Can be seen as a generalization of BERT and GPT
- Trained by
  - corrupting text with arbitrary noising function
  - learning a model to reconstruct the original text

(https://marian-nmt.github.io/) (latest access: April 2021)

BERT encoder

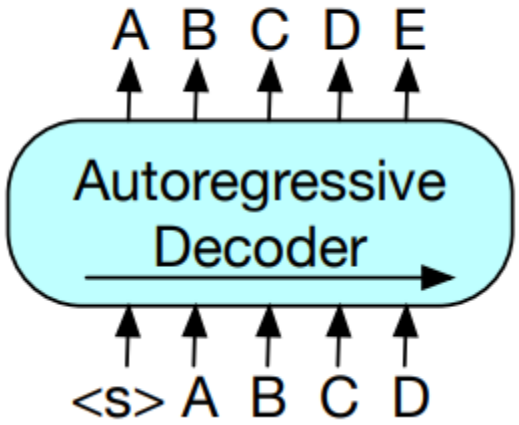
- BERT encoder
  - Random tokens are replaced with masks, and the document is encoded bidirectionally.
  - Missing tokens are predicted independently
    - Not suitable for text generation!



Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, Luke Zettlemoyer.
BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension.

BART

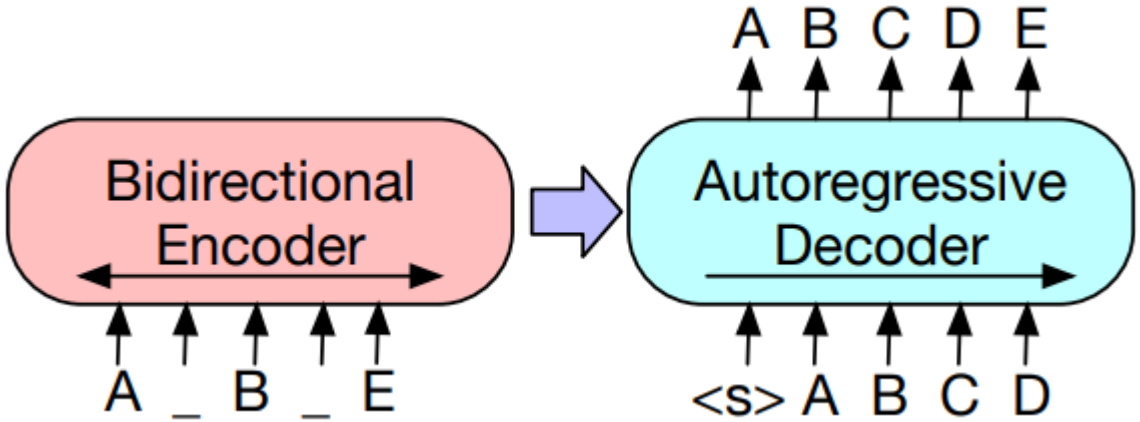
- GPT decoder
  - Tokens are predicted auto-regressively
    - Suitable for text generation.
  - Words are conditioned on leftward context
    - it cannot learn bidirectional interactions



Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, Luke Zettlemoyer.
BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension.

BART

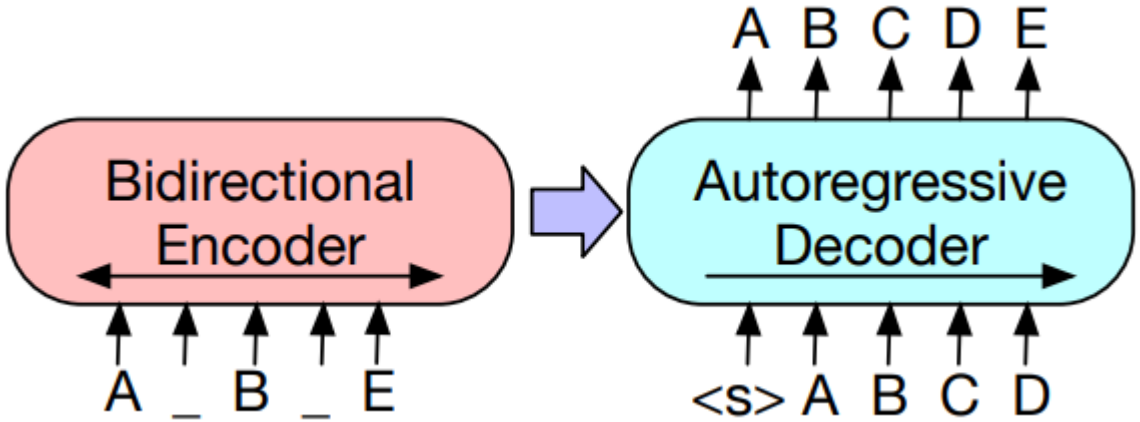
- BART encoder decoder
  - Inputs to the encoder need not be aligned with decoder outputs
    - Arbitrary noise transformations are allowed
  - A document is corrupted by replacing spans of text with mask symbols
    - The corrupted document (left) is encoded with a bidirectional model
    - the likelihood of the original document (right) is calculated with an autoregressive decoder



Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, Luke Zettlemoyer.
BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension.

BART

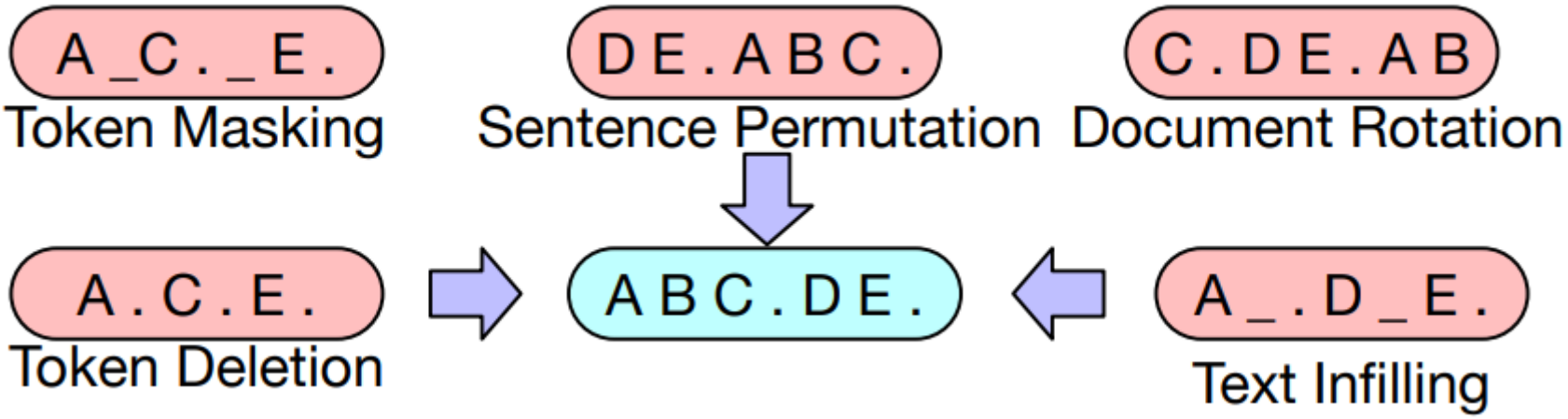
- BART encoder decoder
  - For fine-tuning, an uncorrupted document is input to both the encoder and decoder, and we use representations from the final hidden state of the decoder



Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, Luke Zettlemoyer.
BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension.

BART

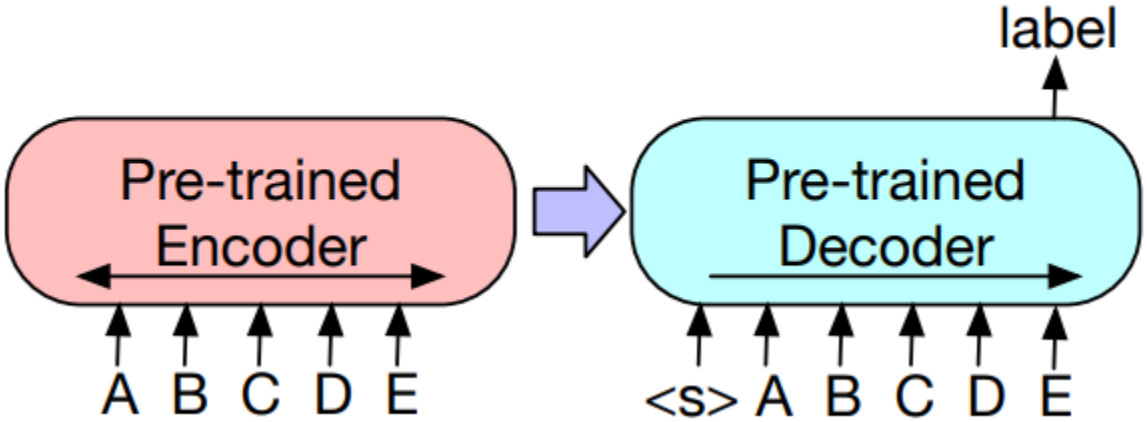
- Transformation for input noising



Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, Luke Zettlemoyer.
BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension.

BART

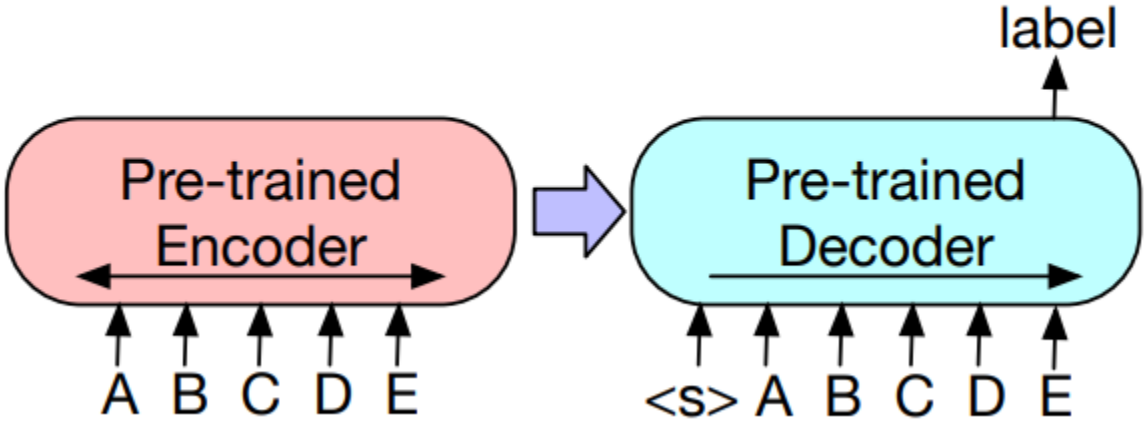
- BART for sequence classification
  - The same input is fed into encoder and decoder
  - The representation from the final output is used



Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, Luke Zettlemoyer.
BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension.

BART

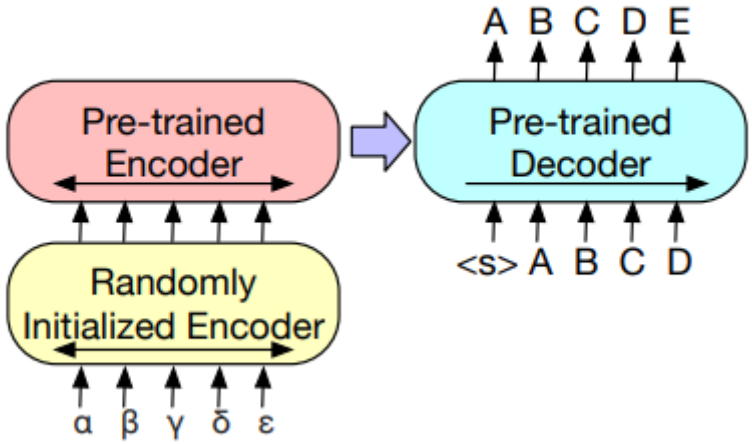
- The same architecture can be exploited for text summarization



Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, Luke Zettlemoyer. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension.

BART

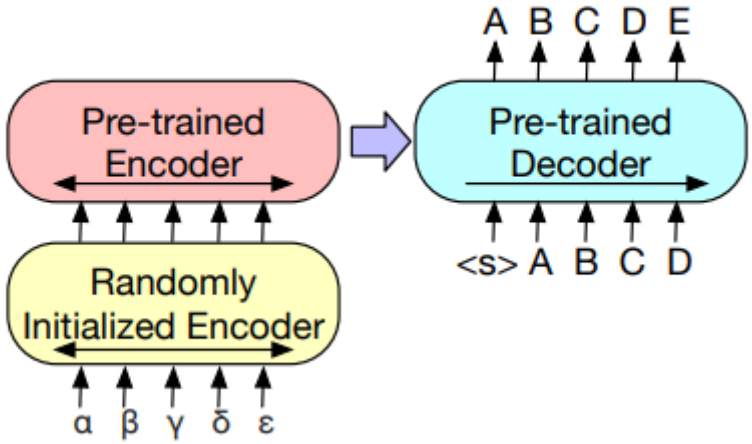
- BART for Machine Translation
  - learn a small additional encoder that replaces the word embeddings in BART
  - The new encoder can use a disjoint vocabulary



Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, Luke Zettlemoyer.
BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension.

BART

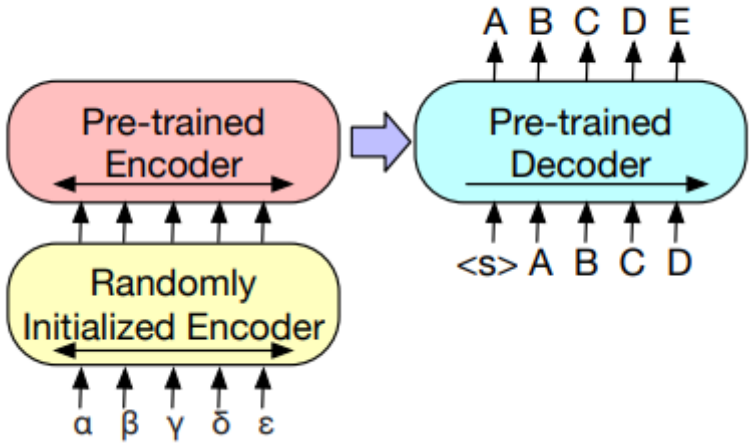
- BART for Machine Translation
  - Use the entire BART model (both encoder and decoder) as a single pretrained decoder for machine translation by adding a new set of encoder parameters that are learned from bilingual corpus



Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, Luke Zettlemoyer.
BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension.

BART

- BART for Machine Translation
  - Replace BERT's encoder embedding layer with a new randomly initialized encoder
  - The model is trained end-to-end
    - Train a new encoder to map words in the source language into an input that BART can denoise to the target language.
    - The new encoder can use a separate vocabulary from the original BART model



Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, Luke Zettlemoyer. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension.

Additional reading on BART

- Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, Luke Zettlemoyer. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. CoRR abs/1910.13461 (2019)
- Download and read the paper: https://arxiv.org/pdf/1910.13461.pdf

Acknowlegdements and copyright license

- Copyright licence
  - Attribution + Noncommercial + NoDerivatives

- Acknowledgements
  - I would like to thank Dr. Moreno La Quatra, who collaborated to the writing and revision of the teaching content
- Affiliation
  - The author and his staff are currently members of the Database and Data Mining Group at Dipartimento di Automatica e Informatica (Politecnico di Torino) and of the SmartData interdepartmental centre
    - https://dbdmg.polito.it
    - https://smartdata.polito.it

Thank you!

