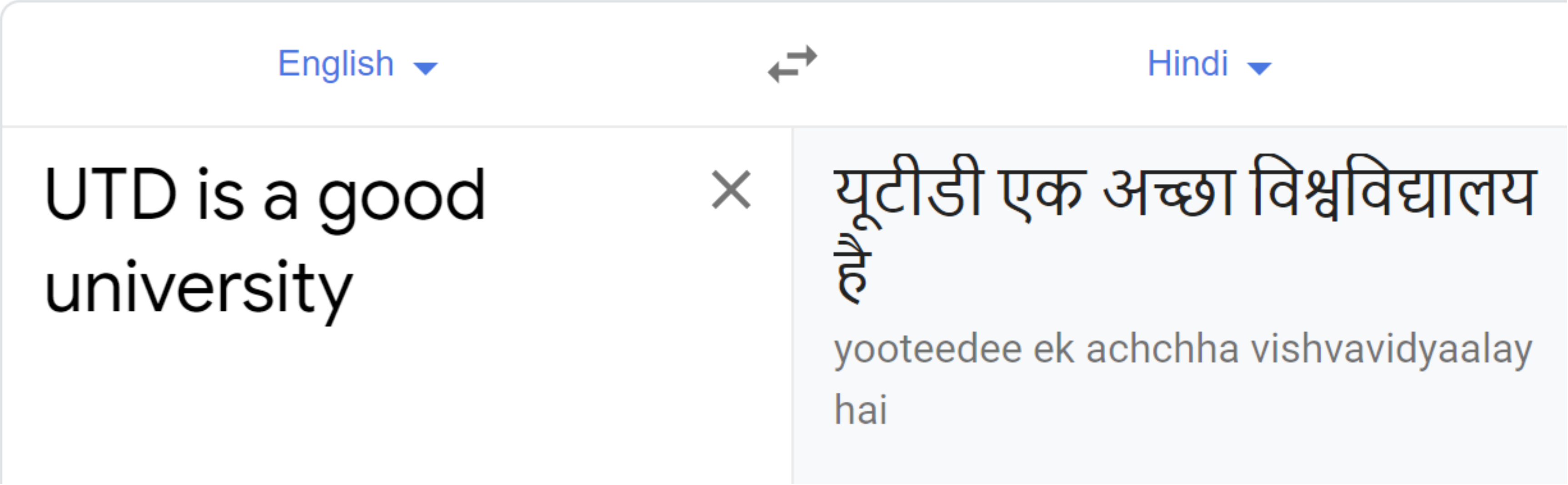
Encoder Decoder model (Sequence to Sequence Model)

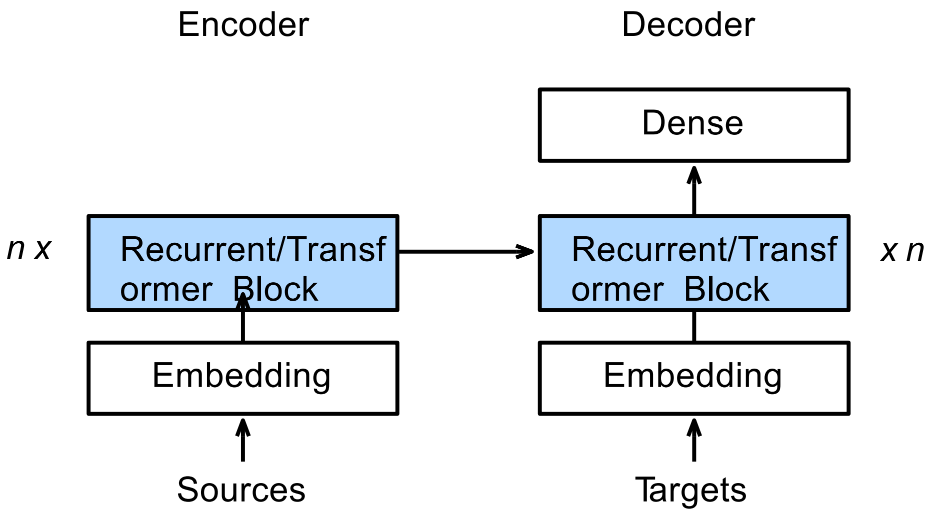
Machine Translation

- Given a sentence in a source language, translate into a target language
- These two sequences may have different lengths



Encoder/Decoder summary

- The encoder is a standard RNN/Transformer model without the output layer
- The output from last layer of encoder is an input to decoder



Predictions – Test Data

We model the probability of a sentence as follows:

$$y' = \underset{y}{\operatorname{argmax}}\, p(y|x) = \underset{y}{arg\mathrm{max}} \prod_{t=1}^{T} p(y_t|y_{<t}, x)$$

- We cannot find the exact solution.

- WHY ?

Predictions – Test Data

We model the probability of a sentence as follows:

$$y' = \underset{y}{\operatorname{argmax}}\, p(y|x) = \underset{y}{arg\max} \prod_{t=1}^{T} p(y_t|y_{<t}, x)$$

- We cannot find the exact solution.
- We need to check $|V|^T$ possible hypothesis.

Predictions – Test Data

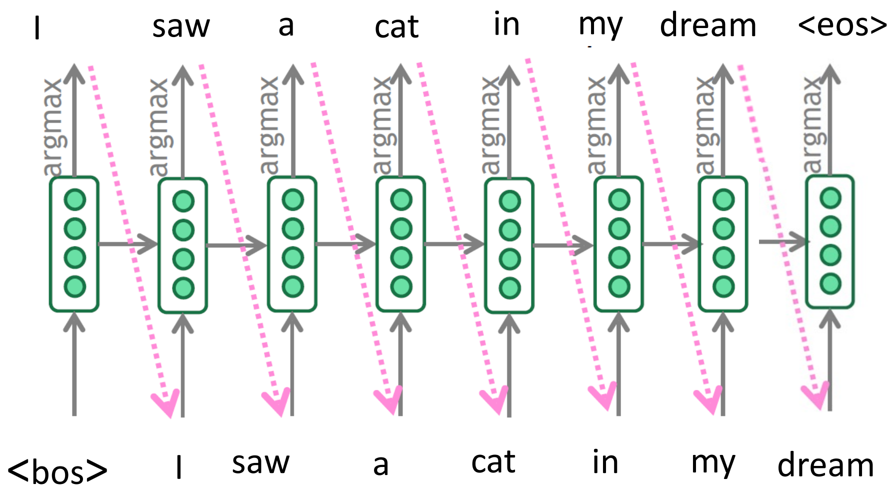
- For every possible sequence, compute its probability and pick the best one

- If output vocabulary size is V, and max sequence length T, then we need to examine V^T sequences

- It's computationally infeasible

- For Example if, $V = 10000, T = 10$, $V^T = 10^{40}$

Predictions – Greedy Decoding



- At each step, pick the most probable token

Problems ?

Predictions – Greedy Decoding

- Cannot undo incorrect decision
    - I ____
    - I saw ___
    - I saw a ____
    - I saw a cat___
    - I saw a cat in __

Greedy search:
0.5×0.4×0.4×0.6=0.048

Time step	1	2	3	4
A	0.5	0.1	0.2	0.0
B	0.2	0.4	0.2	0.2
C	0.2	0.3	0.4	0.2
<HRV!	0.1	0.2	0.2	0.6

A better choice:
0.5×0.3×0.6×0.6=0.054

Time step	1	2	3	4
A	0.5	0.1	0.1	0.1
B	0.2	0.4	0.6	0.2
C	0.2	0.3	0.2	0.1
<HRV!	0.1	0.2	0.1	0.6

$$\underset{y}{\operatorname{argmax}} \prod_{t=1}^{T} p(y_t|y_{<t}, x) \neq \underset{y_t}{\operatorname{argmax}} \prod_{t=1}^{T} p(y_t|y_{<t}, x)$$

Beam Search

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.

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

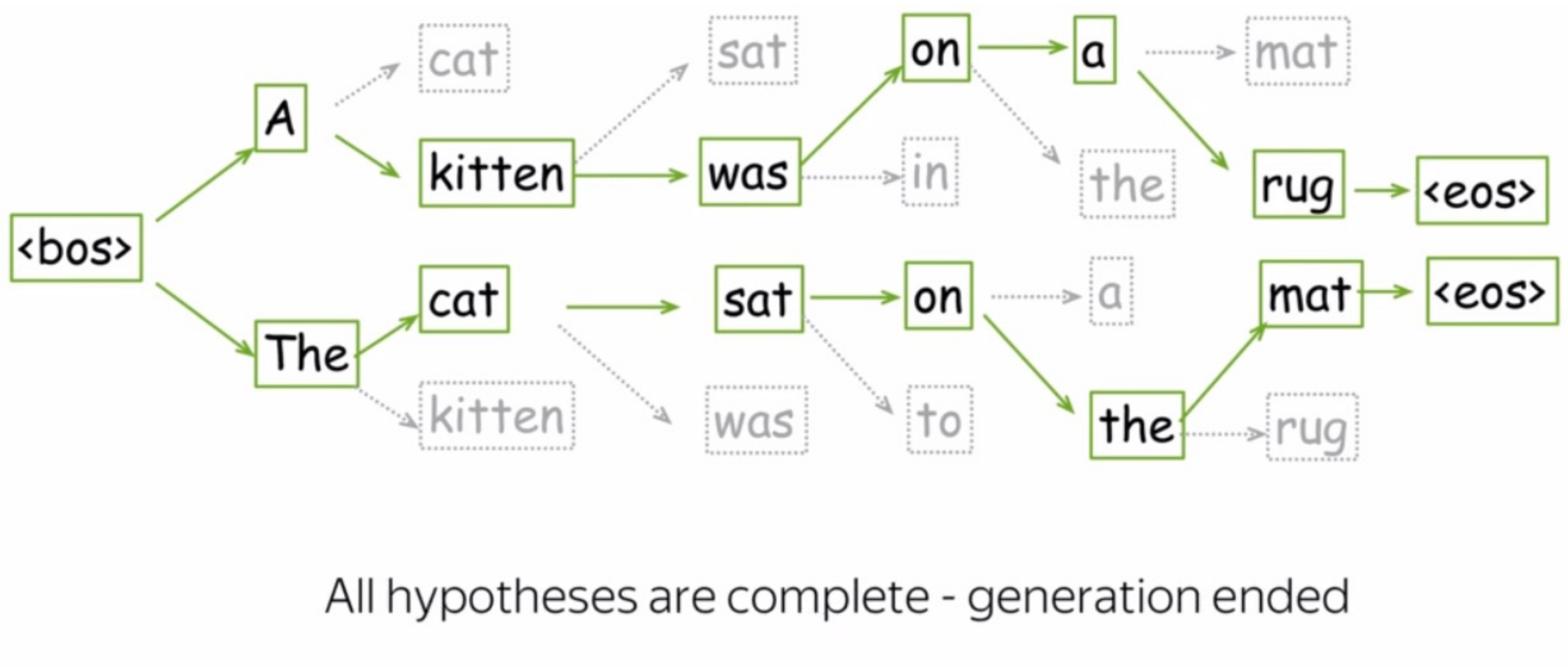
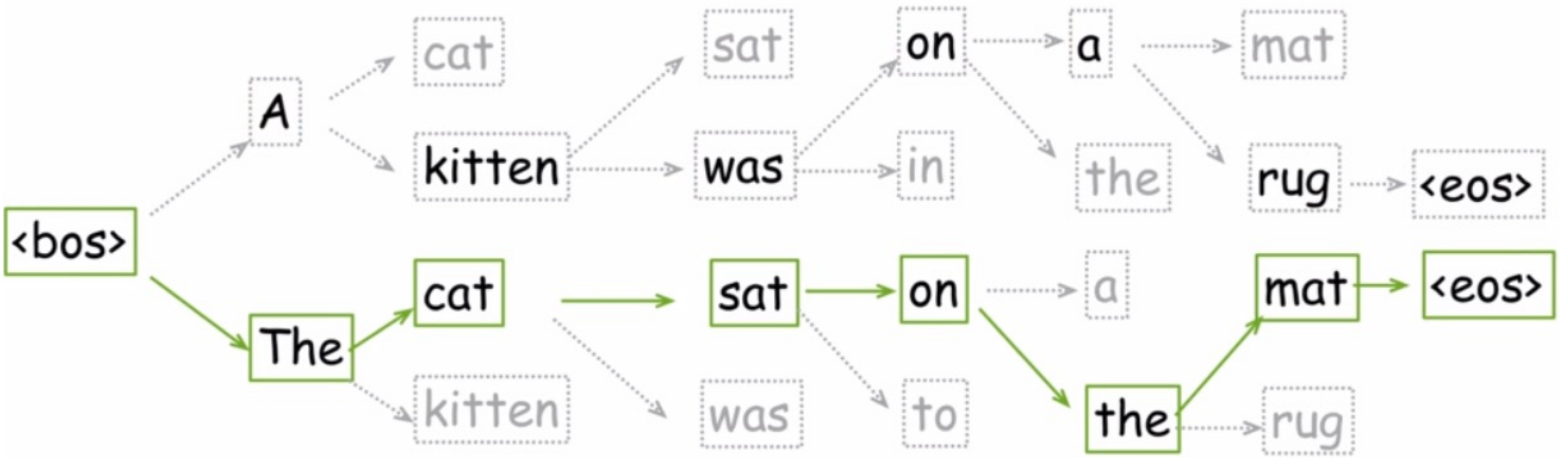


Image source https://lena-voita.github.io/nlp_course/seq2seq_and_attention.html

Beam Search



Pick the hypothesis with the highest probability

Image source https://lena-voita.github.io/nlp_course/seq2seq_and_attention.html

Beam search decoding: stopping criterion

- In beam search decoding, different hypotheses may produce <END> tokens on different timesteps
  - 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 timestep *T* (where *T* is some pre-defined cutoff), or
  - We have at least *n* completed hypotheses (where *n* is pre-defined cutoff)

Slide Source : http://web.stanford.edu/class/cs224n/slides/cs224n-2020-lecture08-nmt.pdf

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

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

- <u>Problem with this:</u> longer hypotheses have lower scores

- <u>Fix:</u> Normalize by length. Use this to select top one instead:

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

Slide Source : http://web.stanford.edu/class/cs224n/slides/cs224n-2020-lecture08-nmt.pdf

Evaluation

BLEU (Papineni et al. 2002): what fraction of {1-4}-grams in the system translation appear in the reference translations?

$$p_n = \frac{Number\ of\ ngrams\ in\ system\ and\ refernce\ translations}{Number\ of\ ngrams\ in\ system\ translation}$$

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

Evaluation

French: Le chat est sur le tapis.

Reference 1: The cat is on the mat.

Reference 2: There is a cat on the mat.

MT output: the the the the the the the.

Precision:

$$p_1 = \frac{7}{7}$$

Modified precision:

$$\frac{2}{7}$$

$$clip\ count = \min(count, \max\ ref\ count)$$

Evaluation

Hypothesis translation
Appeared calm when he was taken to the American plane, which will to Miami, Florida.

Reference translations
Orejuela appeared calm as he was led to the American plane which will take him to Miami, Florida.
Orejuela appeared calm while being escorted to the plane that would take him to Miami, Florida.
Orejuela appeared calm as he was being led to the American plane that was to carry him to Miami in Florida.
Orejuela seemed quite calm as he was being led to the American plane that would take him to Miami in Florida.

Appeared
calm
when
he
was
taken
to
the
American
plane
,
which
will
to
Miami
,
Florida
.

$$p_1 = \frac{15}{18} = 0.833$$

Ngrams appearing >1 time in the hypothesis can match up to the max number of times they appear in a single reference — e.g., two commas in hypothesis but one max in any single reference.

Callison-Burch et al. (2006), Re-evaluating the Role of BLEU in Machine Translation Research

Slide Credit: David Bamman (UC Berkley)

Evaluation

Hypothesis translation
Appeared calm when he was taken to the American plane, which will to Miami, Florida.

Reference translations
Orejuela appeared calm as he was led to the American plane which will take him to Miami, Florida.
Orejuela appeared calm while being escorted to the plane that would take him to Miami, Florida.
Orejuela appeared calm as he was being led to the American plane that was to carry him to Miami in Florida.
Orejuela seemed quite calm as he was being led to the American plane that would take him to Miami in Florida.

Appeared calm
calm when
when he
he was
was taken
taken to
to the
the American
American plane
plane ,
, which
which will
will to
to Miami
Miami ,
, Florida
Florida .

$$p_2 = \frac{10}{17} = 0.588$$

Callison-Burch et al. (2006), Re-evaluating the Role of BLEU in Machine Translation Research

Slide Credit: David Bamman (UC Berkley)

Evaluation

We could optimize the score by minimizing the denominator (the number of ngrams generated)

$$p_n = \frac{Number\ of\ ngrams\ in\ system\ and\ refernce\ translations}{Number\ of\ ngrams\ in\ system\ translation}$$

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

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

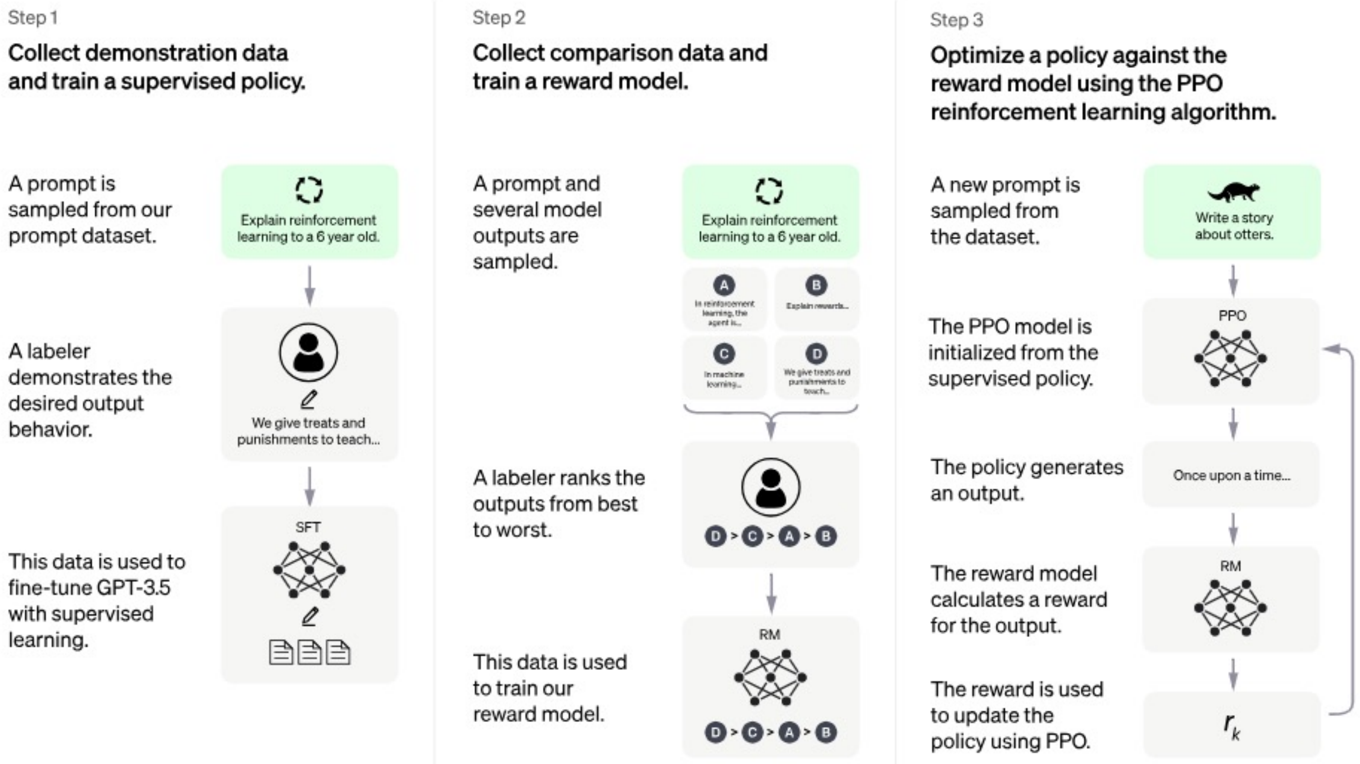
c = length of hypothesis translation

r = length of closest reference translation

Important variables in Transformers

- **Vocabulary Size (V):** The number of unique tokens that the model recognizes.
- **Embedding/Model Size (D):** The dimensionality of the word embeddings, also known as the hidden size.
- **Sequence/Context Length (L):** The maximum number of tokens that the model can process in a single pass.
- **Number of Attention Heads (H):** In the multi-head attention mechanism, the input is divided into H different parts.
- **Intermediate Size (I):** The feed-forward network has an intermediate layer whose size is typically larger than the embedding size.
- **Number of Layers (N):** The number of Transformer blocks/layers.
- **Batch Size (B):** The number of examples processed together in one forward/backward pass during training.
- **Tokens Trained on (T):** The total number of tokens that a model sees during training. This is normally reported more than the number of epochs.

ChatGPT



https://www.youtube.com/watch?v=zjkBMFhNj_g

The video is part of the syllabus

Instruction tuning(SFT)

RLHF