Sequence Models & Attention Mechanism

Learning Objectives:

- Describe a basic sequence-to-sequence model
- Compare and contrast several different algorithms for language translation
- Optimize beam search and analyze it for errors
- Use beam search to identify likely translations
- Apply BLEU score to machine-translated text
- Implement an attention model
- Train a trigger word detection model and make predictions
- Synthesize and process audio recordings to create train/dev datasets

Sequence Models & Attention Mechanism

1 Basic models
2 Picking the most likely sentence
3 Beam search
4 Refinements to beam search
5 Error analysis on beam search
6 Bleu score (optional)
7 Attention model intuition
8 Attention model
9 Speech recognition
10 Trigger word detection

Sequence Models & Attention Mechanism

Basic models

Basic models

- Sequence-to-sequence models, applied in tasks like machine translation and image captioning, use an encoder-decoder architecture. For translation, an RNN encodes input, and another decodes output.
- In image captioning, a pre-trained CNN encodes the image, and an RNN generates the caption.
- Unlike language models aiming for random choices, sequence-to-sequence models seek the most likely translation or caption.
- Techniques like beam search and attention models are employed, as detailed in the next section.

Sequence to sequence model

$x^{<1>}$ $x^{<2>}$ $x^{<3>}$ $x^{<4>}$ $x^{<5>}$

Jane visite l'Afrique en septembre

⟶ Jane is visiting Africa in September.

$y^{<1>}$ $y^{<2>}$ $y^{<3>}$ $y^{<4>}$ $y^{<5>}$ $y^{<6>}$

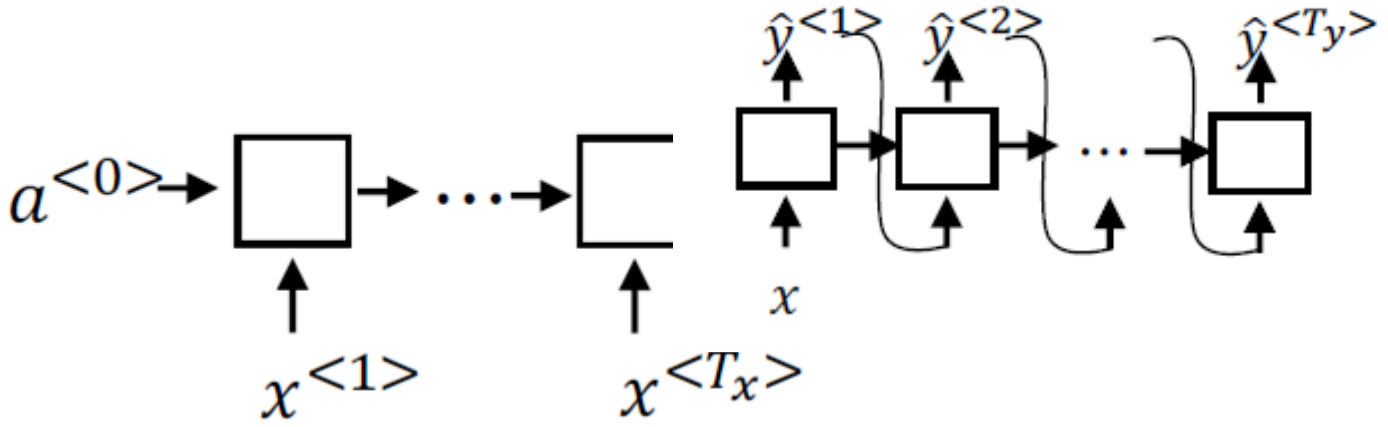
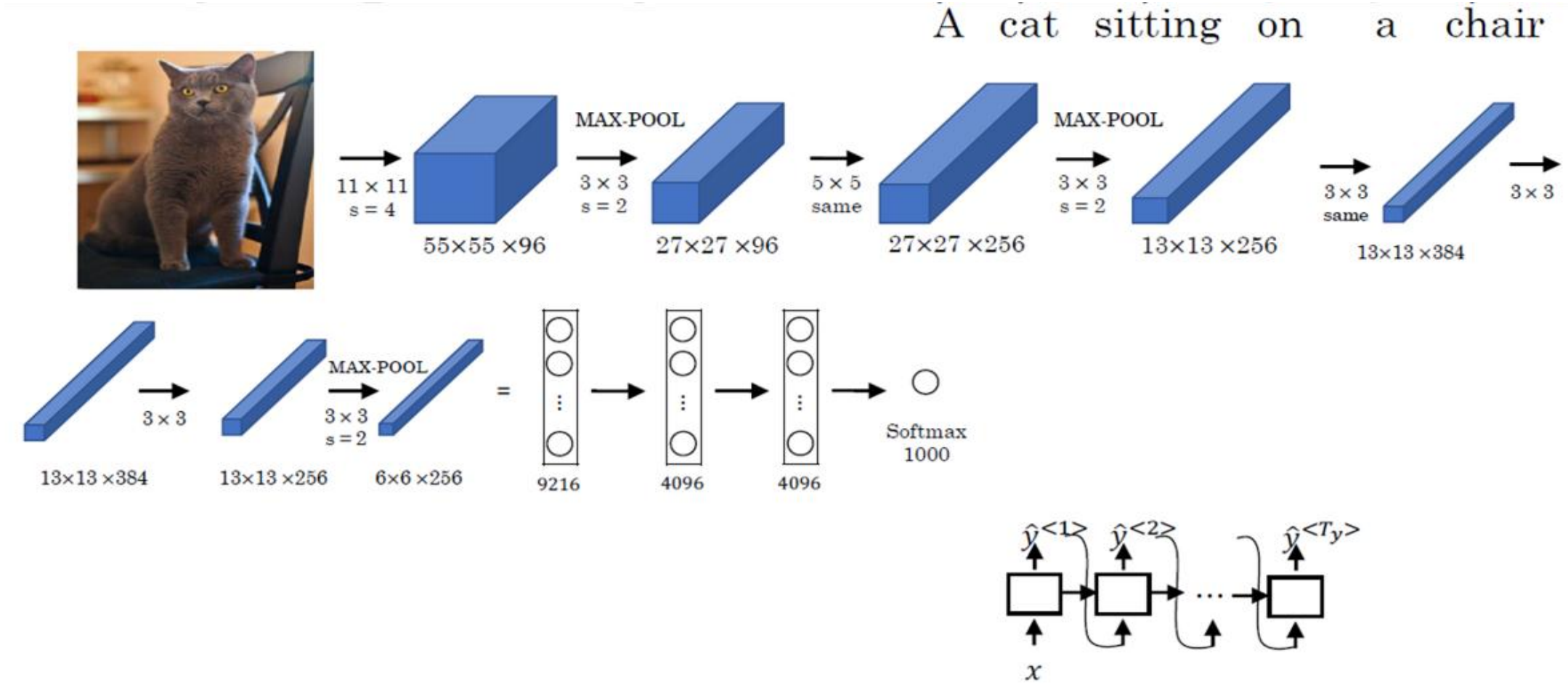


Image captioning



FPT UNIVERSITY

Sequence to sequence models

Picking the most likely sentence

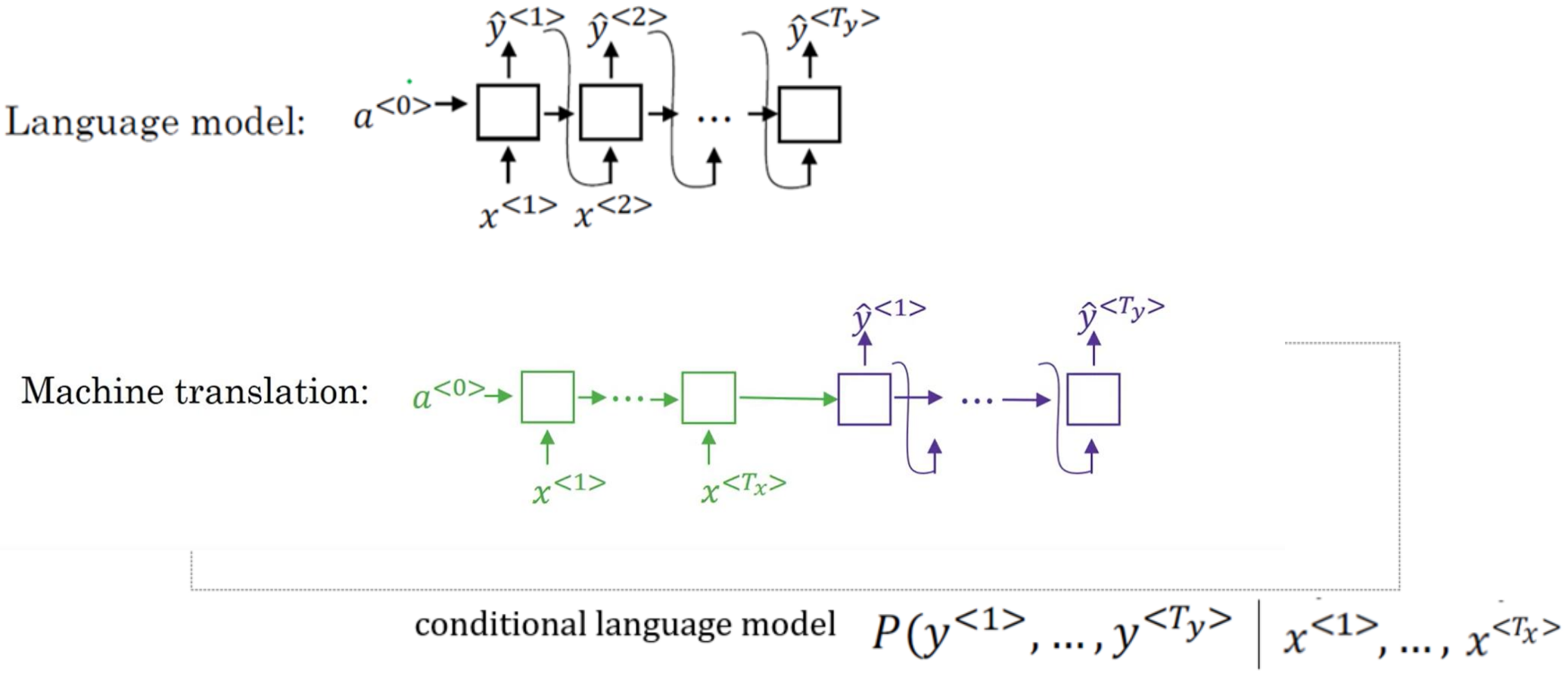
Picking the most likely sentence

- In summary, language modeling involves estimating the probability of a sentence and generating new sentences, while seq2seq modeling involves estimating the probability of an output sentence given an input sentence, which is a conditional language model.
- Seq2seq models consist of an encoder network that learns the representation of the input and a decoder network that generates the output sentence conditioned on the input representation, while language models do not require an explicit encoder-decoder structure.

Picking the most likely sentence

- To find the English sentence that maximizes the conditional probability given the input French sentence in machine translation, approximate search algorithms like beam search are used instead of exhaustive enumeration due to the exponentially large space of all possible sentences in a language.
- Greedy search picks the most likely word at each step, but it results in suboptimal translations.
- Beam search, on the other hand, is an approximate search algorithm that tries to find the sentence that maximizes the conditional probability and usually does a good enough job.

Machine translation as building a conditional language model



Finding the most likely translation

Jane visite l'Afrique en septembre.

$$P(y^{<1>}, \ldots, y^{<T_y>} \mid x)$$

→ Jane is visiting Africa in September.

→ Jane is going to be visiting Africa in September.

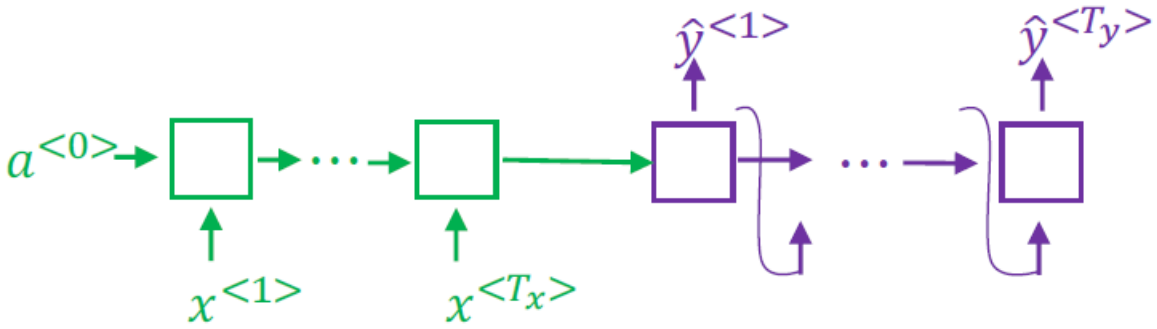
→ In September, Jane will visit Africa.

→ Her African friend welcomed Jane in September.

$$\arg\max_{y^{<1>},\ldots,y^{<T_y>}} P(y^{<1>}, \ldots, y^{<T_y>} \mid x)$$

Why not a greedy search?

- P(Jane is going | x > Jane is visiting | x)



$$\arg\max_{y^{<1>},\ldots,y^{<T_y>}} P(\hat{y}^{<1>}, \ldots, \hat{y}^{<T_y>} | x)$$

→ Jane is visiting Africa in September.

→ Jane is going to be visiting Africa in September.

FPT Education

FPT UNIVERSITY

Sequence to sequence models

Beam search

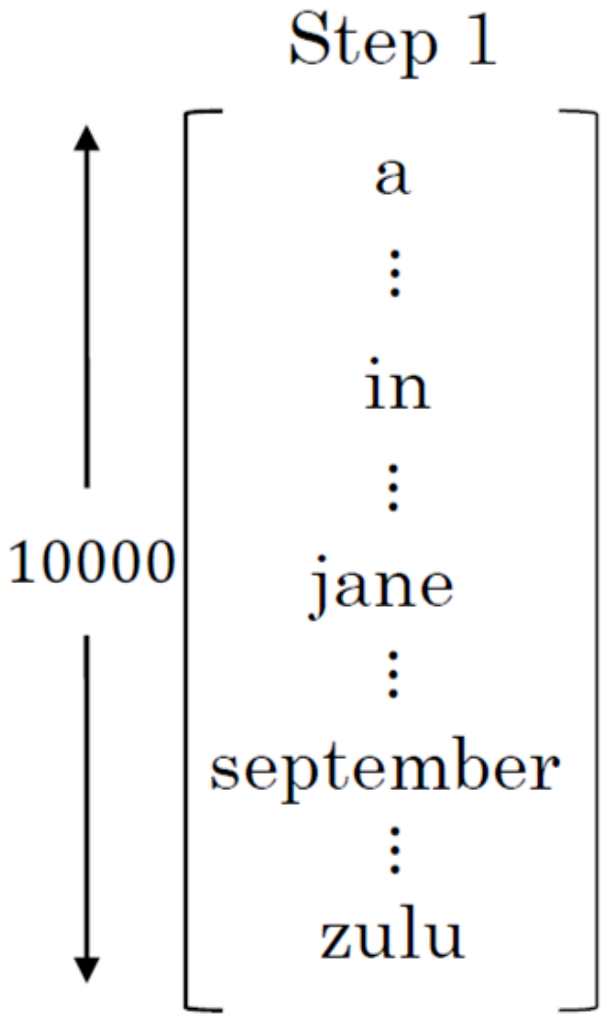
Beam search

- Beam search is a widely used search algorithm that is often used in natural language processing tasks such as machine translation, speech recognition, and text generation.
- It considers multiple alternatives at each step of the sequence generation process, rather than just the most likely one, which is what greedy search does. The algorithm has a parameter called B, which is the beam width, and it determines the number of candidates to consider at each step. The outcome of this process is the most likely output sequence, given a set of input sequences and a pre-trained language model.

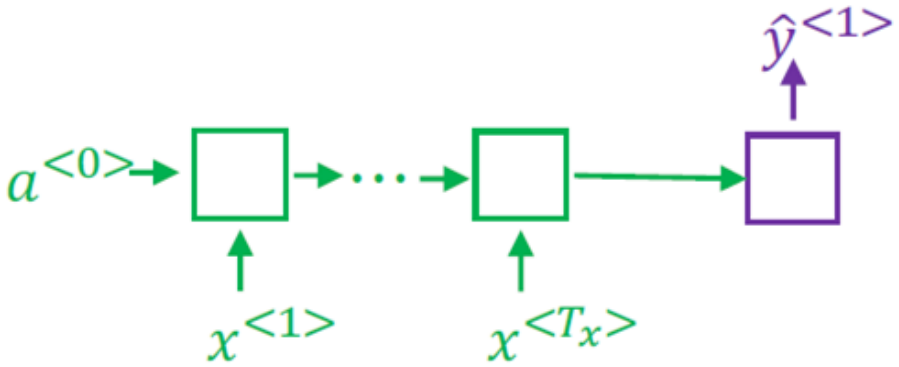
Beam search

- The process of beam search for the example French sentence "Jane, visite l'Afrique en Septembre" is summarized as follows:
- **Step 1:** Pick the first word of the English translation and use a network fragment with an encoder in green and a decoder in purple to evaluate the probability of the first output word given the French input sentence.
- **Step 2:** Consider the second word after the first and evaluate the possible choices using network fragments with hardwired first and second words. The most likely pair of the first and second words is selected by multiplying their probabilities.
- **Step 3:** Pick the top three possibilities for the third word. The most likely output English sentence is the outcome of this process. If the beam width was set to 1, beam search becomes greedy search, which considers only one possibility at a time.
- There are additional tips and tricks that can help make beam search work even better. The next section will discuss these tips and tricks.

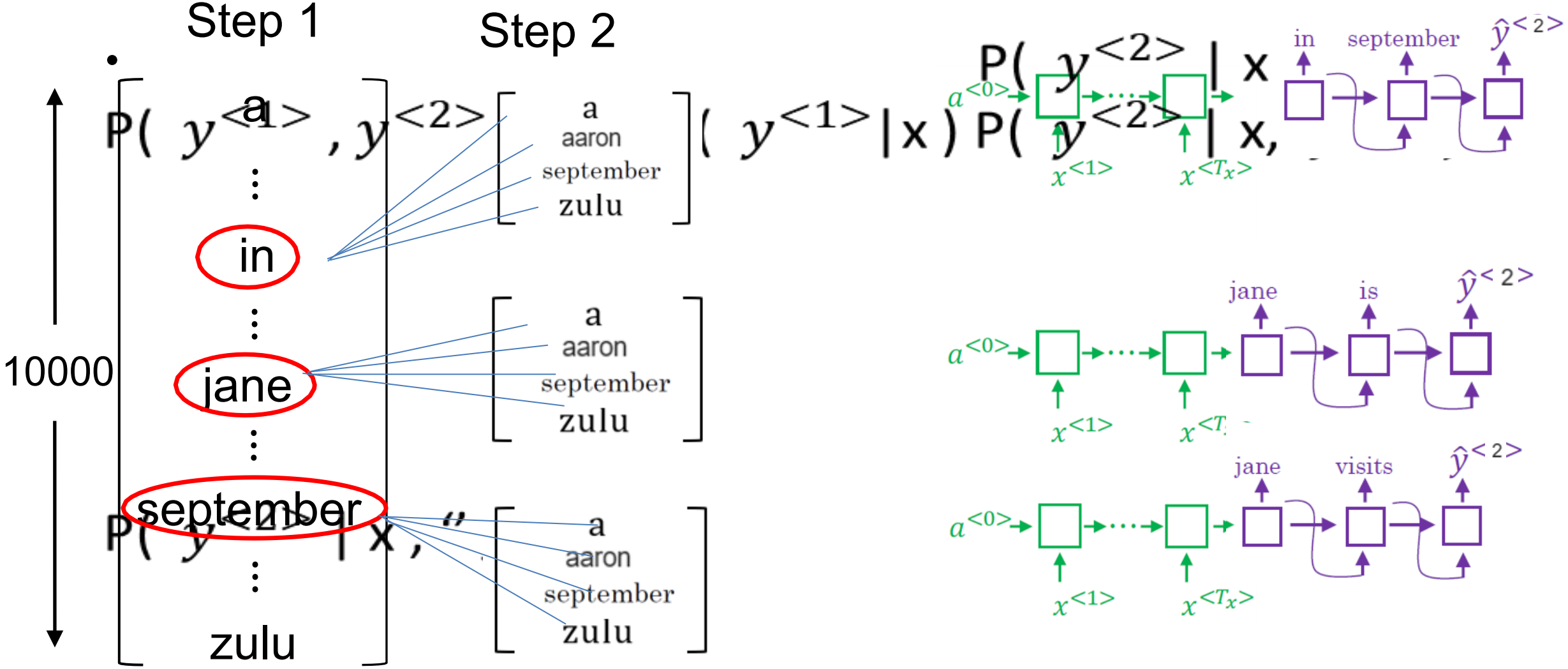
Beam search algorithm



$P(y^{<1>} \mid x)$



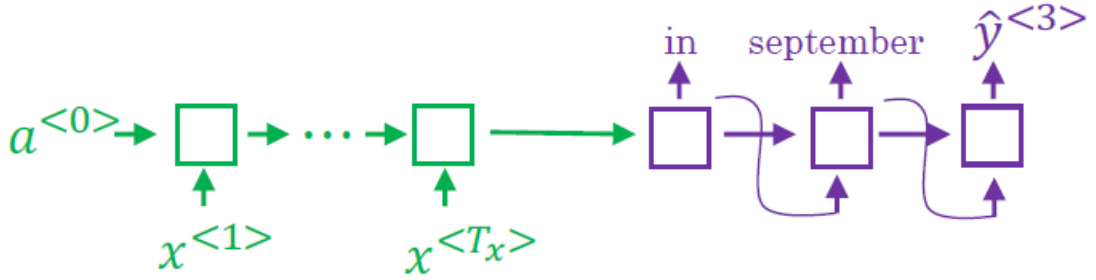
Beam search algorithm



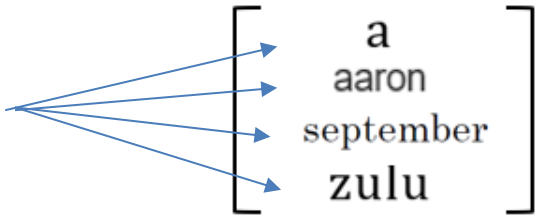
Beam search (B = 3)

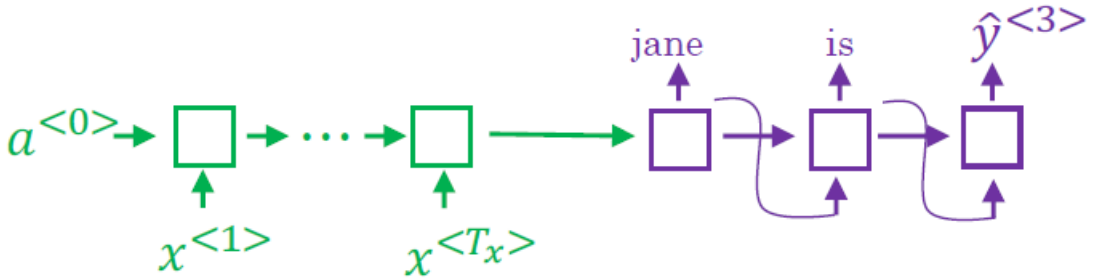
P($y^{<3>}$ | x , " in september ")

in september

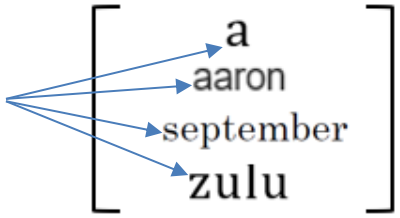


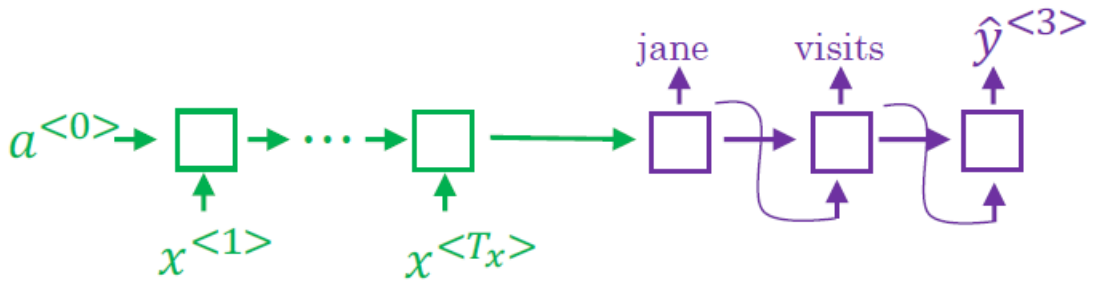
jane is





jane visits





$P(y^{<1>}, y^{<2>} | x)$ jane visits africa in september. <EOS>

FPT UNIVERSITY

Sequence to sequence models

Refinements to beam search

Refinements to beam search

- Two modifications to the beam search algorithm used for machine translation: length normalization and normalizing the objective by the number of words in the translation.
- Length normalization involves maximizing the sum of logarithms of probabilities instead of maximizing the product of probabilities.
- The second modification uses a heuristic formula that reduces the penalty for longer translations while still providing some normalization. The algorithm keeps track of the top three possibilities for each possible sentence length up to the maximum length considered, and the translation with the highest value on the normalized log-likelihood objective is selected.
- Trying different beam widths can help find the best value for a given task.
- While beam search is faster, it is not guaranteed to find the exact maximum compared to exact search algorithms like PAFs Breakfast or DFS DEFA search, and the next section will cover error analysis on beam search.

Length Normalization

$$\arg\max_{y} \prod_{t=1}^{T_y} P(y^{<t>} \mid x, y^{<1>}, \ldots, y^{<t-1>})$$

$$P(\ y^{<1>} \ldots . y^{<ty>} \mid x\ ) = P(\ y^{<1>} \mid x)\ P(\ y^{<2>} \mid x\ ,\ y^{<1>})$$
$$P(\ y^{<ty>} \mid x,\ y^{<1>} \ldots\ldots\ y^{<ty-1>}\ )$$

$$\arg\max_{y} \sum_{t=1}^{T_y} \log P(y^{<t>} \mid x, y^{<1>}, \ldots, y^{<t-1>})$$

$$\sum_{t=1}^{T_y} \log P(y^{<t>} \mid x, y^{<1>}, \ldots, y^{<t-1>})$$

Beam search discussion

- Beam width B?

- 1 > 3 > 10 , 100 , 1000 , > 3000
- Large B : better result , slower
- Small B : worse result , faster

Unlike exact search algorithms like BFS (Breadth First Search) or DFS (Depth First Search), Beam Search runs faster but is not guaranteed to find exact maximum for arg max)P(y|x).

FPT UNIVERSITY

Sequence to sequence models

Error analysis on beam search

Error analysis on beam search

- How can error analysis be used with beam search to identify whether errors in machine translation arise from the beam search algorithm or the RNN model that generates the translation?
- Beam search keeps track of the top B possibilities but is an approximate search algorithm that may not always output the most likely sentence. To determine if errors are due to beam search or the RNN model, the probabilities $P(y^{*} \mid x)$ and $P(\hat{y} \mid x)$ are computed using the RNN model to see which is greater.
  - If $P(y^{*} \mid x) \leq P(\hat{y} \mid x)$, the RNN model is at fault
  - if $P(y^{*} \mid x) > P(\hat{y} \mid x)$, the fault lies with beam search.
  - Through error analysis, the responsible component can be identified and improved upon, be it the beam width or the RNN model.

Example

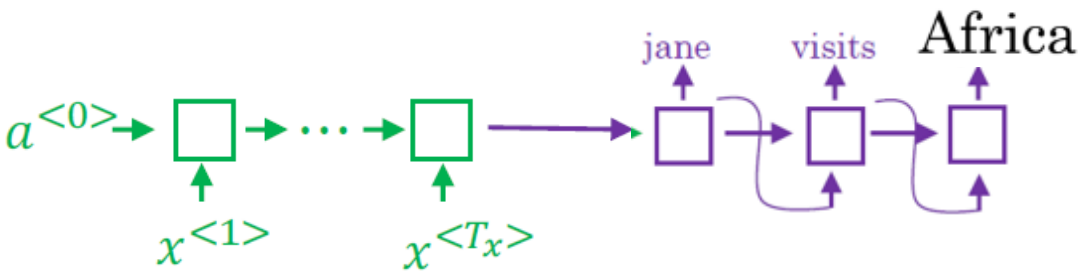
Jane visite l'Afrique en septembre.

Human: Jane visits Africa in September. $y^{<*>}$

Algorithm: Jane visited Africa last September. $\hat{y}$

$$RNN\ compute \quad P(y^{<*>}|x) > P(\hat{y}\,|x)$$

<=



Error analysis on beam search

FPT Education
FPT UNIVERSITY

Human: Jane visits Africa in September. $y^{<*>}$ $P(y^{<*>}|x)$

Algorithm: Jane visited Africa last September. $\hat{y}$ $P(\hat{y}|x)$

Case 1: $P(y^{<*>}|x) > P(\hat{y}|x)$

Beam search chose $\hat{y}$ But $y^{<*>}$ attains higher P(y|x)

Conclusion: Beam search is at fault.

Case 2: $P(y^{<*>}|x) \leq P(\hat{y}|x)$

$y^{<*>}$ is a better translation than $\hat{y}$ But RNN predicted $P(y^{<*>}|x) < P(\hat{y}|x)$

Conclusion: RNN model is at fault.

Error analysis process

Human	Algorithm	$P(y^{<*>}\|x)$	$P(\hat{y}\|x)$	At fault?
Jane visits Africa in September.	Jane visited Africa last September.	$2x10^{-10}$	$2x10^{-10}$	B R B R R . . .

Figures out what faction of errors are “due to” beam search vs. RNN model

FPT Education

FPT UNIVERSITY

Sequence to sequence models

Bleu score (optional)

Bleu score (optional)

- The BLEU score evaluates machine-generated translations' quality by measuring similarity to human-generated translations.
- It's based on precision, counting words in the machine output that appear in human references. To avoid favoring poor translations with the same words, a modified precision measure is used, limiting word counts to their maximum in the reference sentences.

Bleu score (optional)

- BLEU Score Components:
  - Involves unigrams, bigrams, trigrams, and higher-order n-grams.
  - Computes modified precision for each, considering occurrences in both machine-generated and reference texts.
- Calculation Method:
  - Modified precision for bigrams is calculated similarly to unigrams.

Counts are based on the frequency of each bigram in machine-generated output and references.

- Final BLEU Score:
  - Computed by averaging modified precision scores for all n-grams.
  - Includes brevity penalty adjustment for ...

Evaluating machine translation

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: Modified precision:

Bleu score on bigrams

Example: Reference 1: The cat is on the mat.

Reference 2: There is a cat on the mat.

MT output: The cat the cat on the mat.

	Count	countclip
the cat	2	1
cat the	1	0
cat on	1	1
on the	1	1
the mat	1	1

$$\frac{4}{6}$$

Bleu score on unigrams

Example: Reference 1: The cat is on the mat.

Reference 2: There is a cat on the mat.

MT output: The cat the cat on the mat.

$$p_1 = \frac{\sum_{unigram \in \hat{y}} count_{clip}\,(unigram)}{\sum_{unigram \in \hat{y}} count\,(unigram)}$$

$$p_n = \frac{\sum_{ngram \in \hat{y}} count_{clip}\,(ngram)}{\sum_{ngram \in \hat{y}} count\,(ngram)}$$

Bleu details

p_n = Bleu score on n-grams only

Combined Bleu score:

$$\text{BP} = \begin{cases} 1 & \text{if MT\_output\_length} > \text{reference\_output\_length} \\ \exp(1 - \text{MT\_output\_length}/\text{reference\_output\_length}) & \text{otherwise} \end{cases}$$

FPT Education

FPT UNIVERSITY

Sequence to sequence models

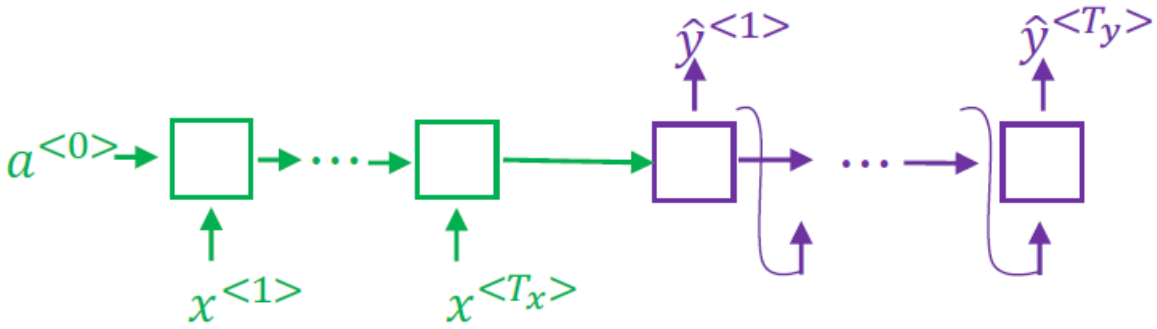
Attention model intuition

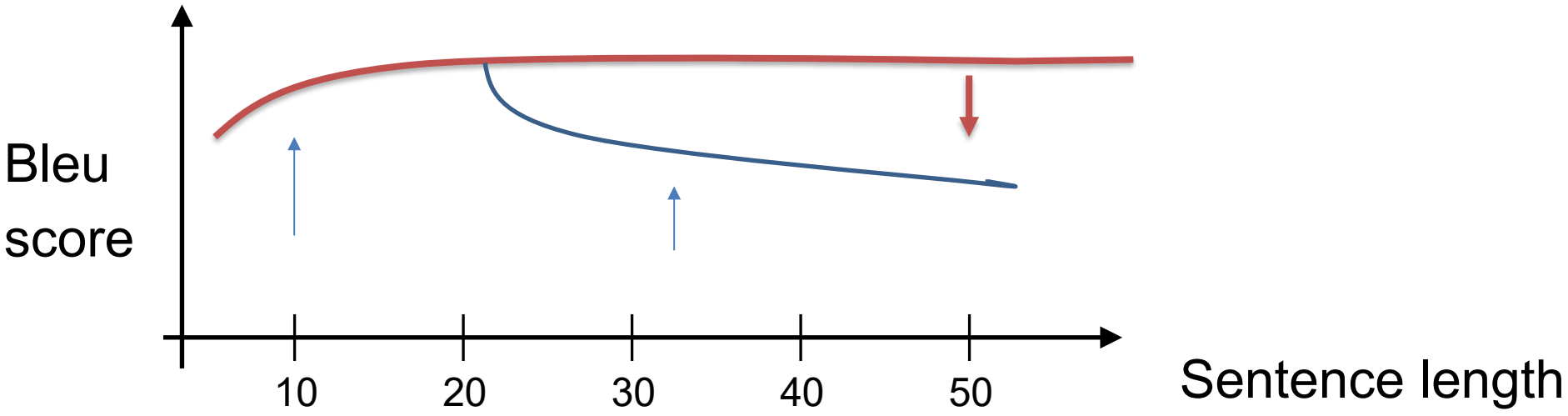
Attention model intuition

- The Encoder-Decoder architecture is a commonly used model for machine translation, but struggles to effectively translate long sentences as it tries to memorize the whole sentence before generating the translation.

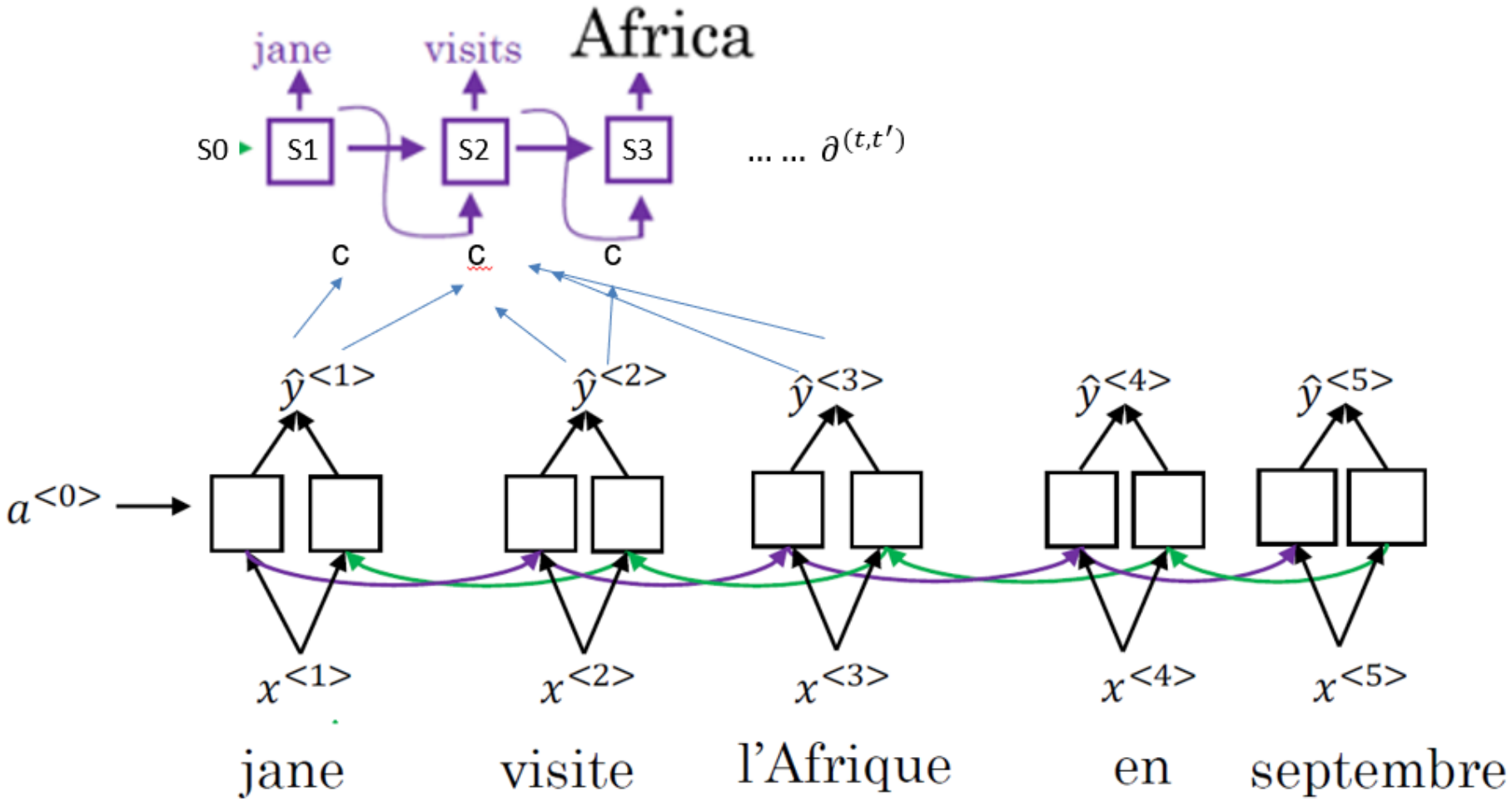
- The Attention Model addresses this issue by allowing the model to focus on different parts of the input sentence at different times while generating the translation. This is achieved through attention weights that determine how much attention the model should pay to each part of the input sentence when generating each word in the output sentence.

The problem of long sequences





Attention model intuition

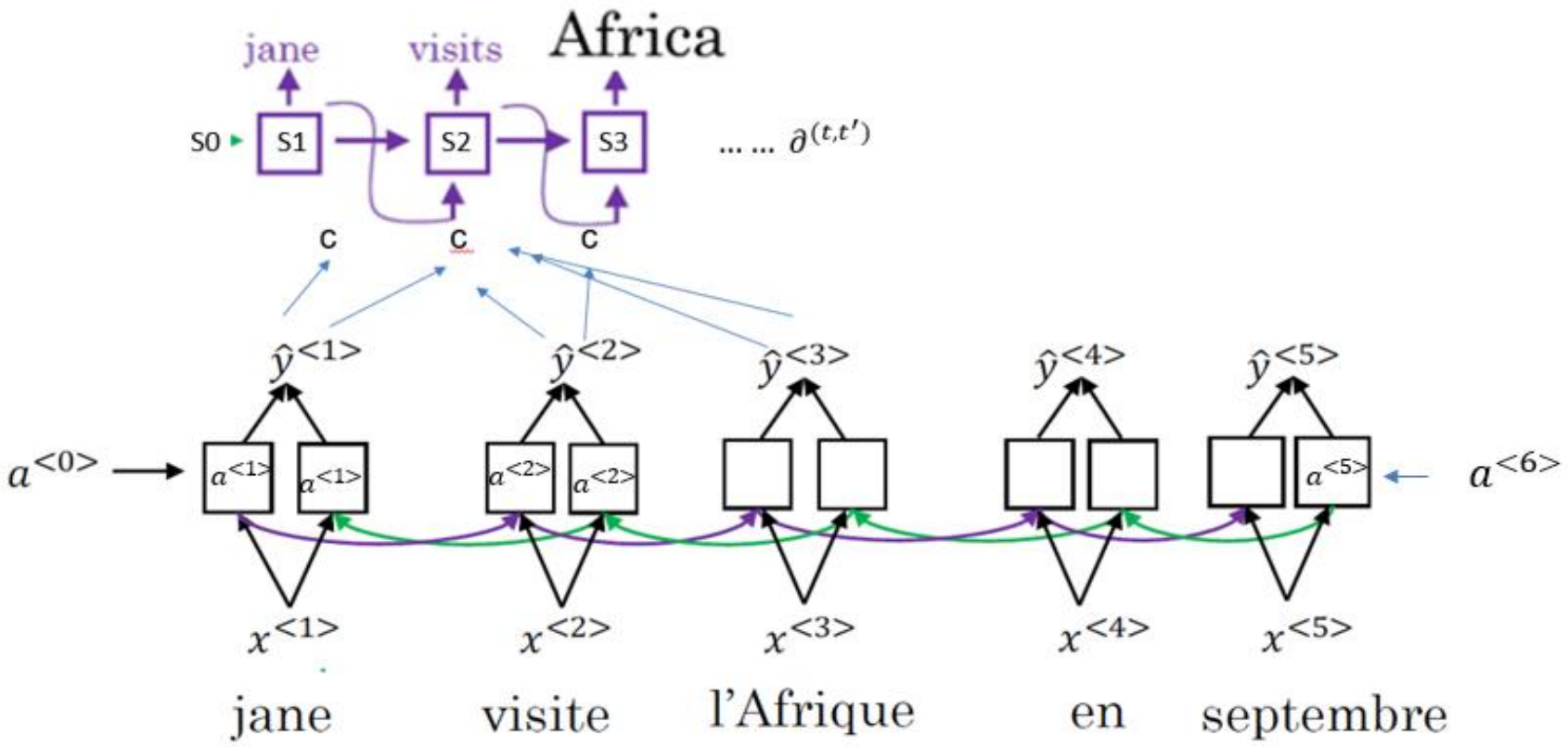


Sequence to sequence models

Attention model

- The Attention model, crucial in machine translation, enables a neural network to focus on specific input sequence parts while generating an output.
- Implemented with bidirectional RNNs, GRUs, or LSTMs, it computes features for each input sentence word. A forward-only RNN generates the translation, utilizing a context vector for each time step. This vector sums features from different steps, weighted by attention weights computed through a small neural network.
- The Attention model finds applications beyond translation, like image captioning, despite its quadratic time complexity, prompting ongoing research for cost reduction.

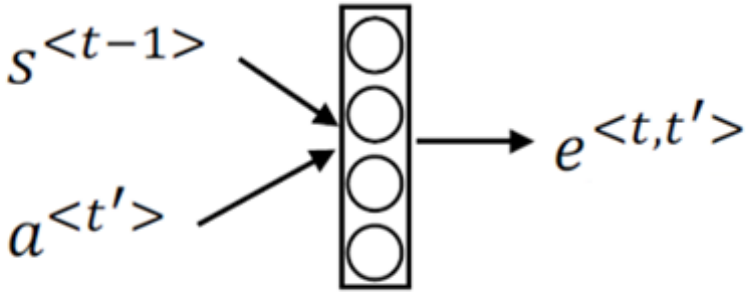
Attention model

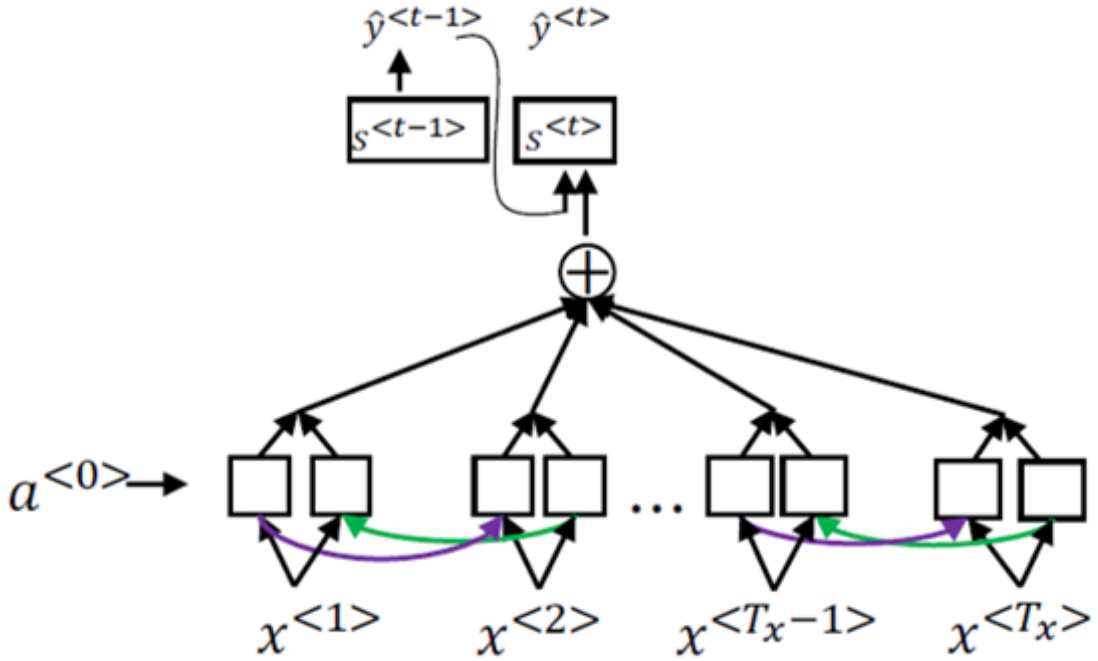


Computing attention $\alpha^{<t,t'>}$

$$\alpha^{<t,t'>} = \text{amount of attention } y^{<t>} \text{ should pay to } a^{<t'>}$$

$$\alpha^{<t,t'>} = \frac{\exp(e^{<t,t'>})}{\sum_{t'=1}^{T_x} \exp(e^{<t,t'>})}$$



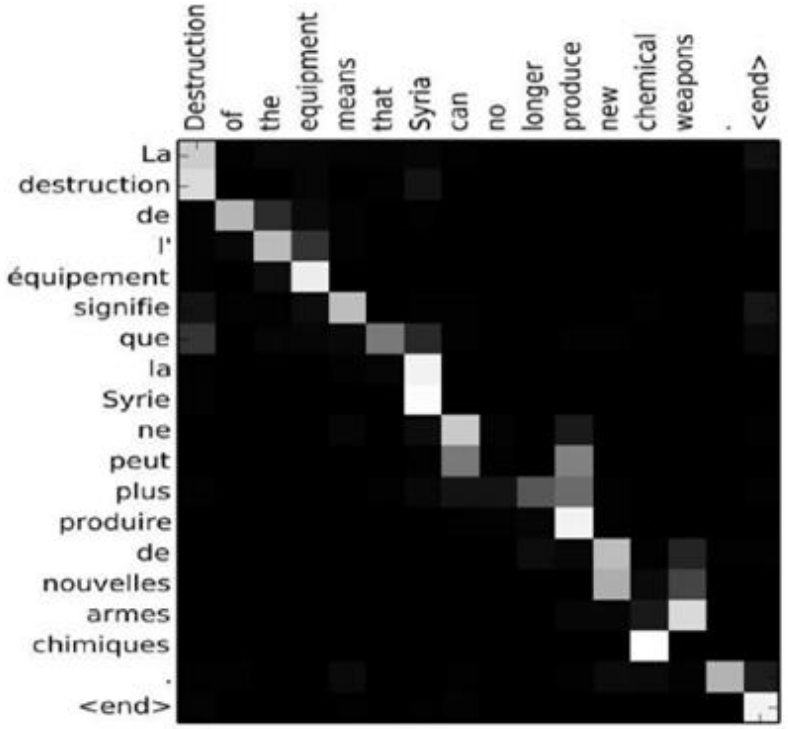


Attention examples

July 20th 1969 $\longrightarrow$ 1969 – 07 – 20

23 April, 1564 $\longrightarrow$ 1564 – 04 – 23

Visualization of $\alpha^{<t,t'>}$:



FPT Education

FPT UNIVERSITY

Speech recognition

Audio data

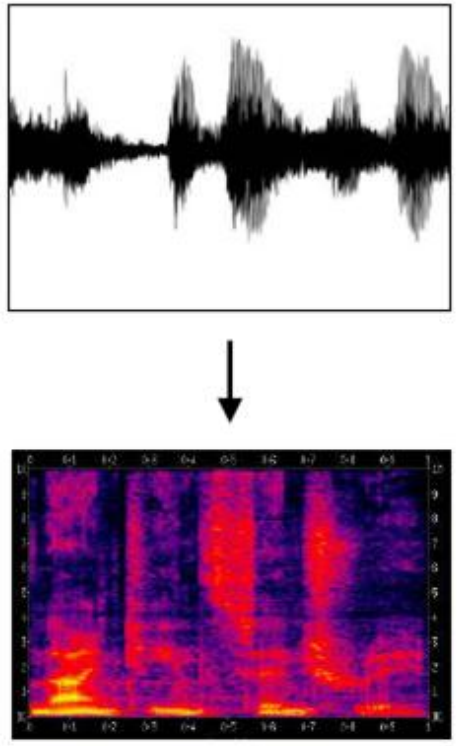
Speech recognition

- Speech recognition involves converting an audio clip into a text transcript, typically using a spectrogram as input to a deep learning model.
- End-to-end deep learning models can input raw audio and directly output the transcript.
- One approach is to use an attention model, while another is to use the CTC cost for speech recognition.
- A trigger word detection system can be built with a smaller dataset than is required for full speech recognition.

Speech recognition problem

x

audio clip

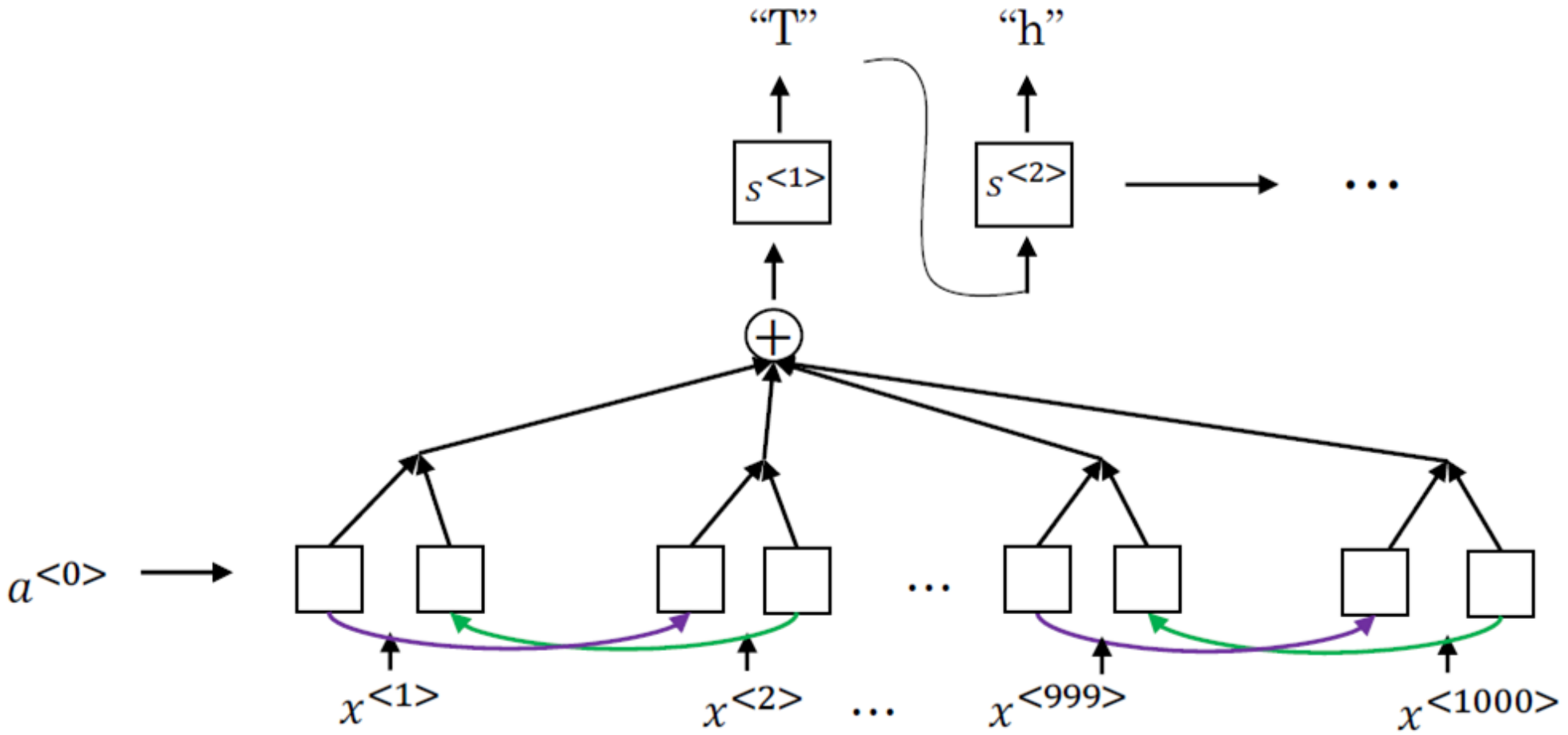


y

transcript

“the quick brown fox”

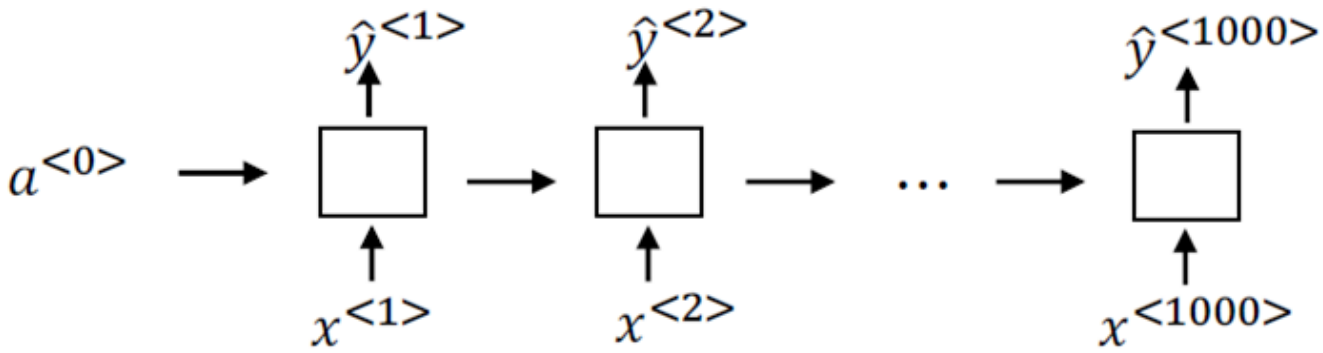
Attention model for speech recognition



CTC cost for speech recognition

(Connectionist temporal classification)

"the quick brown fox"



Basic rule: collapse repeated characters not separated by "blank"

FPT Education

FPT UNIVERSITY

Trigger word detection

Audio data

Trigger word detection

- To build a trigger word detection system, an RNN model is trained using spectrogram features of audio clips and target labels indicating the presence or absence of the trigger word.
- The target label is set to 1 right after the trigger word is spoken, and possibly multiple 1s are outputted for a fixed period of time to balance the ratio of 1s to 0s in the training set.

- There is no consensus on the best algorithm for trigger word detection, but an RNN is one example presented in this module.

What is trigger word detection?

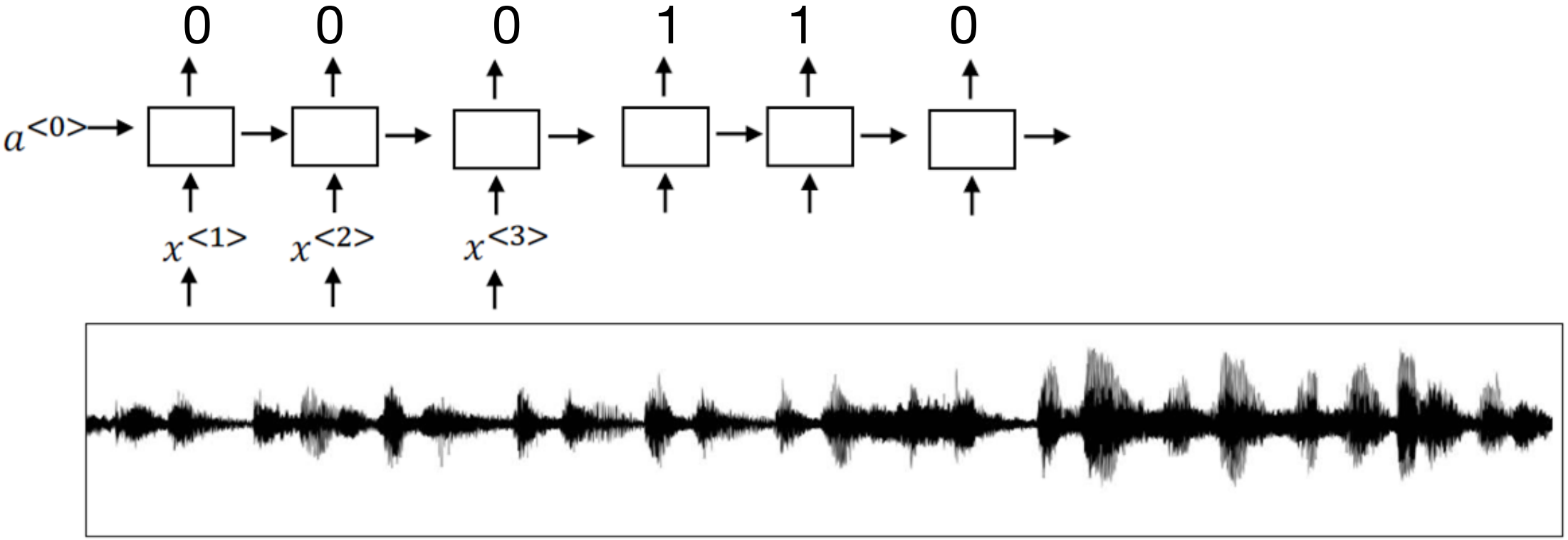
Amazon Echo
(Alexa)

Baidu DuerOS
(xiaodunihao)

Apple Siri
(Hey Siri)

Google Home
(Okay Google)

Trigger word detection algorithm



Summarization

- Basic Models:
  - Introduction to basic models used in natural language processing and machine translation, often including encoder-decoder architectures.
- Picking the Most Likely Sentence:
  - Discusses how to choose the most likely translation or sentence from a set of possible candidates generated by a machine translation system.
- Beam Search:
  - Beam search is a decoding algorithm used in machine translation and other sequence generation tasks. It explores multiple possible sequences simultaneously, narrowing down the search space.
- Refinements to Beam Search:
  - Describes improvements and refinements to the basic beam search algorithm, such as length normalization and diverse beam search.
- Error Analysis on Beam Search:
  - Examines common errors made by beam search and discusses strategies for addressing these errors to improve translation quality.

Summarization

- Bleu Score evaluates machine translation quality by comparing n-grams.
- Attention in neural networks focuses on input parts during output generation.
- Attention models enhance sequence-to-sequence models, improving translation.
- Speech recognition converts spoken language to text using RNNs and attention.
- Trigger word detection identifies specific words in audio, e.g., for voice assistants.

Question

FPT Education
FPT UNIVERSITY

1. Is the encoder modeling the probability of the input sentence in a conditional language model?
2. What are the effects of increasing beam width in beam search?
3. Without sentence normalization, does beam search tend to output shorter translations?
4. If $P(y^*) \leq P(\hat{y})$, will increasing beam width help improve the translation?
5. If $P(y^*) > P(\hat{y})$ in most mistakes, should you focus on improving the search algorithm?
6. What do we expect of attention weights $\alpha^{<t,t'>}$?
7. Why can't we use $s^{<t>}$ instead of $s^{<t-1>}$ to compute attention scores $e^{<t,t'>}$?
8. When does attention outperform encoder-decoder models without attention?
9. What does the CTC model collapse the string "c–oo–kk–bb–oo–oo–kk" to?
10. In trigger word detection, what is $x^{<t>}$?