昆山杜克大学
DUKE KUNSHAN UNIVERSITY

RNN architectures & applications

stats403_deep_learning

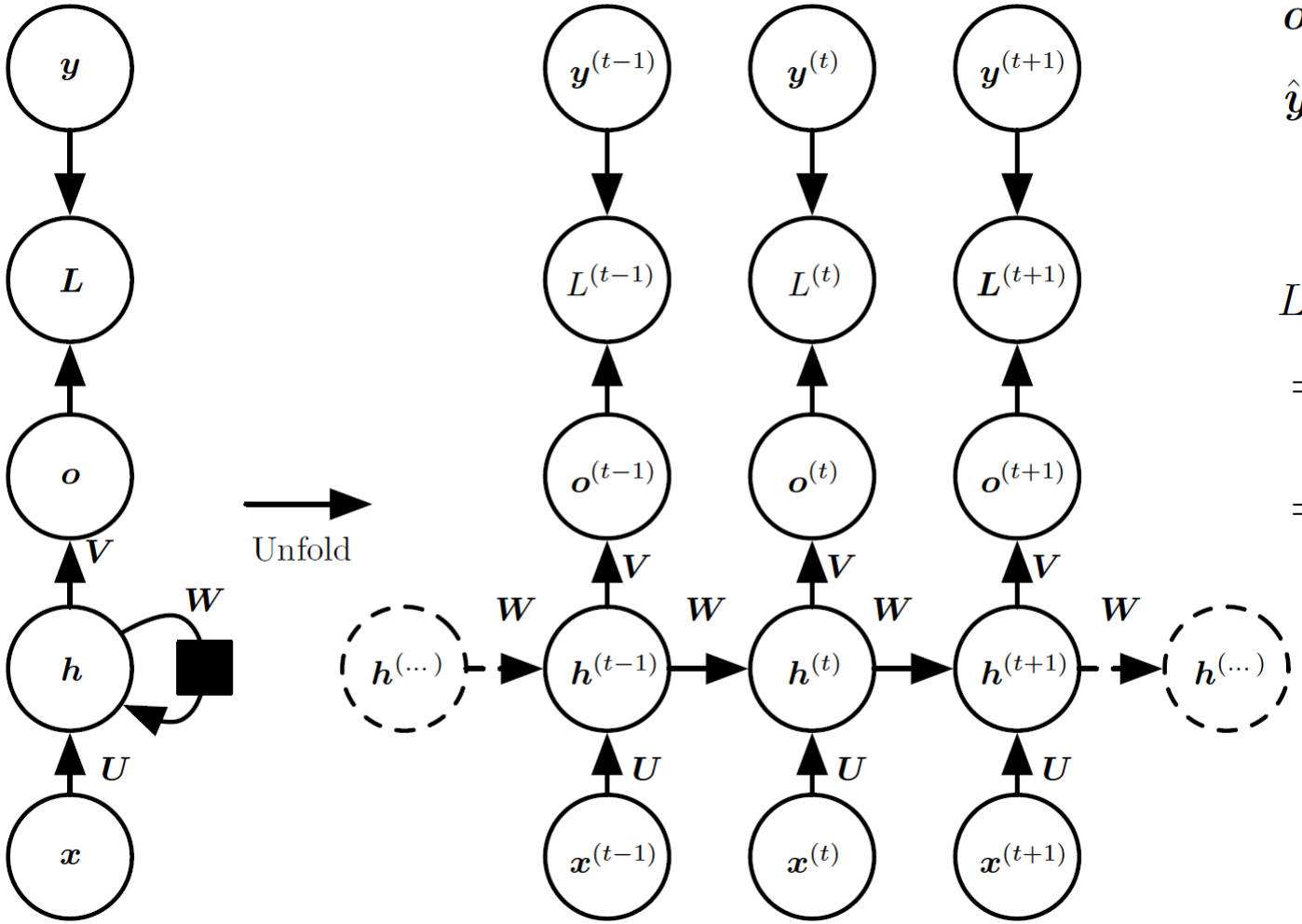
spring_2025

lecture_6

6.1 RNN architectures

backpropagation through time (BPTT)

$$
\begin{aligned}
\boldsymbol{a}^{(t)} &= \boldsymbol{b} + \boldsymbol{W}\boldsymbol{h}^{(t-1)} + \boldsymbol{U}\boldsymbol{x}^{(t)}, \\
\boldsymbol{h}^{(t)} &= \tanh(\boldsymbol{a}^{(t)}), \\
\boldsymbol{o}^{(t)} &= \boldsymbol{c} + \boldsymbol{V}\boldsymbol{h}^{(t)}, \\
\hat{\boldsymbol{y}}^{(t)} &= \operatorname{softmax}(\boldsymbol{o}^{(t)}),
\end{aligned}
$$

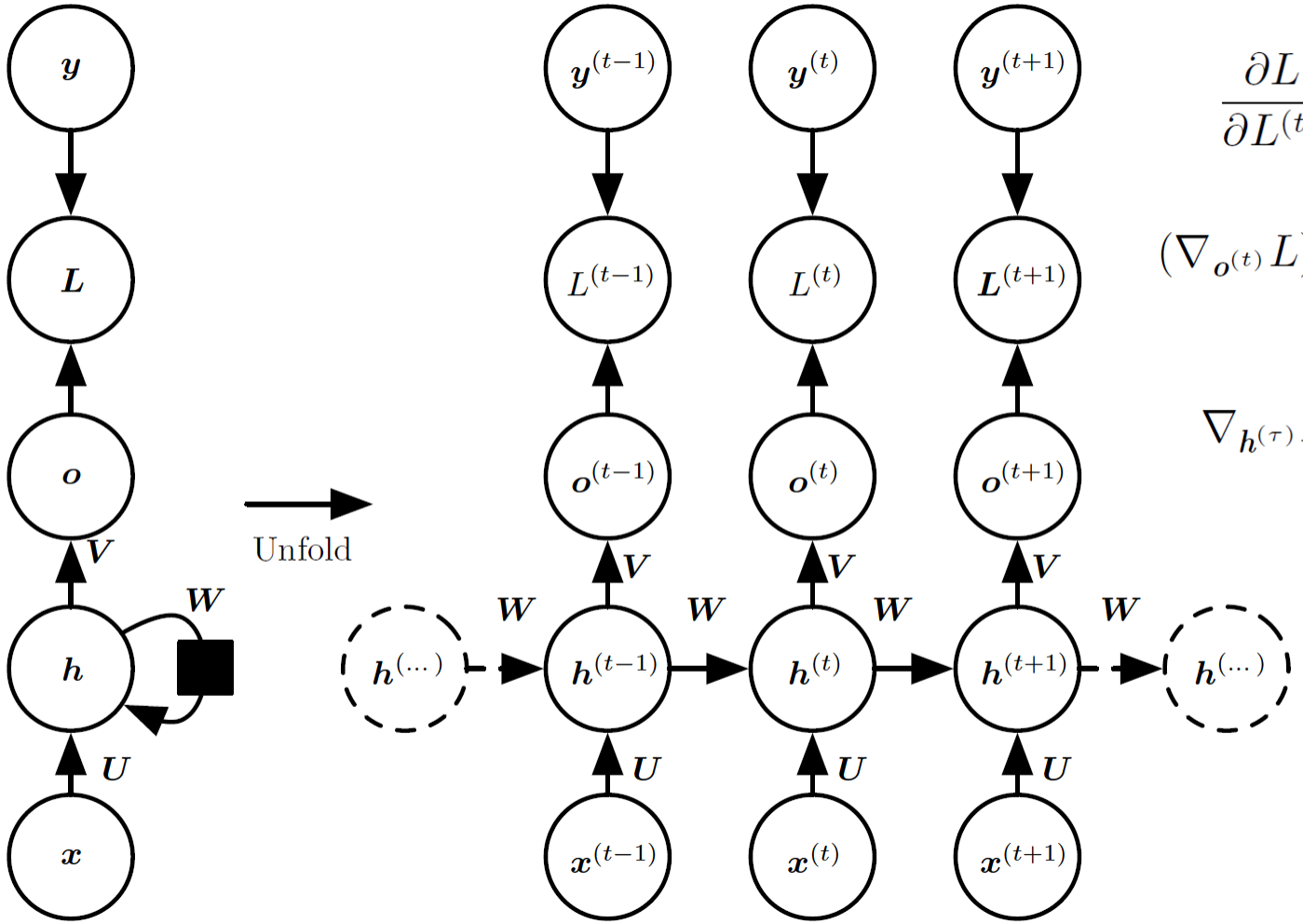


$$
\begin{aligned}
&L\big(\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(\tau)}\}, \{\boldsymbol{y}^{(1)}, \ldots, \boldsymbol{y}^{(\tau)}\}\big) \\
&= \sum_t L^{(t)} \\
&= -\sum_t \log p_{\text{model}}\big(y^{(t)} \mid \{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(t)}\}\big)
\end{aligned}
$$

BPTT

$$
\begin{aligned}
\boldsymbol{a}^{(t)} &= \boldsymbol{b} + \boldsymbol{W}\boldsymbol{h}^{(t-1)} + \boldsymbol{U}\boldsymbol{x}^{(t)}, \\
\boldsymbol{h}^{(t)} &= \tanh(\boldsymbol{a}^{(t)}), \\
\boldsymbol{o}^{(t)} &= \boldsymbol{c} + \boldsymbol{V}\boldsymbol{h}^{(t)}, \\
\hat{\boldsymbol{y}}^{(t)} &= \operatorname{softmax}(\boldsymbol{o}^{(t)}),
\end{aligned}
$$

$$
\begin{aligned}
&L\left(\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(\tau)}\}, \{\boldsymbol{y}^{(1)}, \ldots, \boldsymbol{y}^{(\tau)}\}\right) \\
&= \sum_t L^{(t)} \\
&= -\sum_t \log p_{\text{model}}\left(y^{(t)} \mid \{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(t)}\}\right)
\end{aligned}
$$

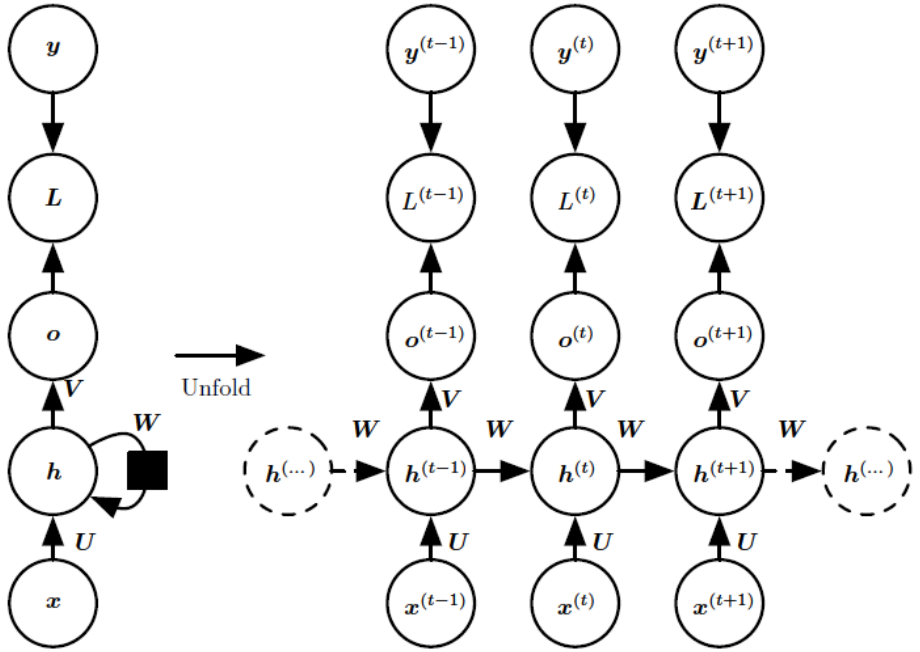


$$\frac{\partial L}{\partial L^{(t)}} = 1$$

$$(\nabla_{\boldsymbol{o}^{(t)}} L)_i = \frac{\partial L}{\partial o_i^{(t)}} = \frac{\partial L}{\partial L^{(t)}} \frac{\partial L^{(t)}}{\partial o_i^{(t)}} = \hat{y}_i^{(t)} - \mathbf{1}_{i,y^{(t)}}$$

$$\nabla_{\boldsymbol{h}^{(\tau)}} L = \boldsymbol{V}^\top \nabla_{\boldsymbol{o}^{(\tau)}} L$$

BPTT

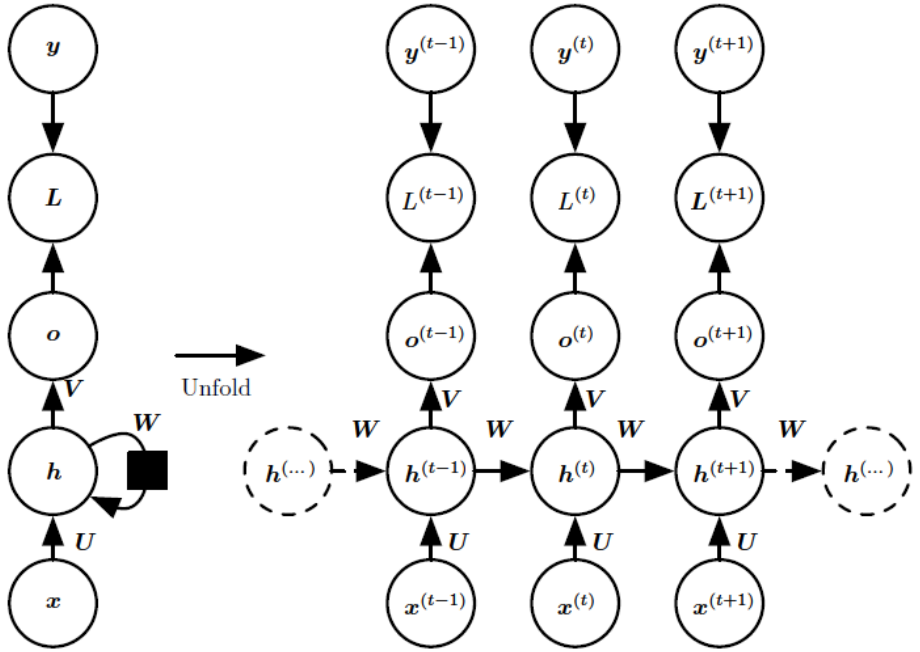


for $t \leq \tau - 1$,

$$\nabla_{\boldsymbol{h}^{(t)}} L = \left(\frac{\partial \boldsymbol{h}^{(t+1)}}{\partial \boldsymbol{h}^{(t)}}\right)^{\top} (\nabla_{\boldsymbol{h}^{(t+1)}} L) + \left(\frac{\partial \boldsymbol{o}^{(t)}}{\partial \boldsymbol{h}^{(t)}}\right)^{\top} (\nabla_{\boldsymbol{o}^{(t)}} L)$$

$$= \boldsymbol{W}^{\top} (\nabla_{\boldsymbol{h}^{(t+1)}} L) \operatorname{diag}\left(1 - (\boldsymbol{h}^{(t+1)})^2\right) + \boldsymbol{V}^{\top} (\nabla_{\boldsymbol{o}^{(t)}} L),$$

BPTT



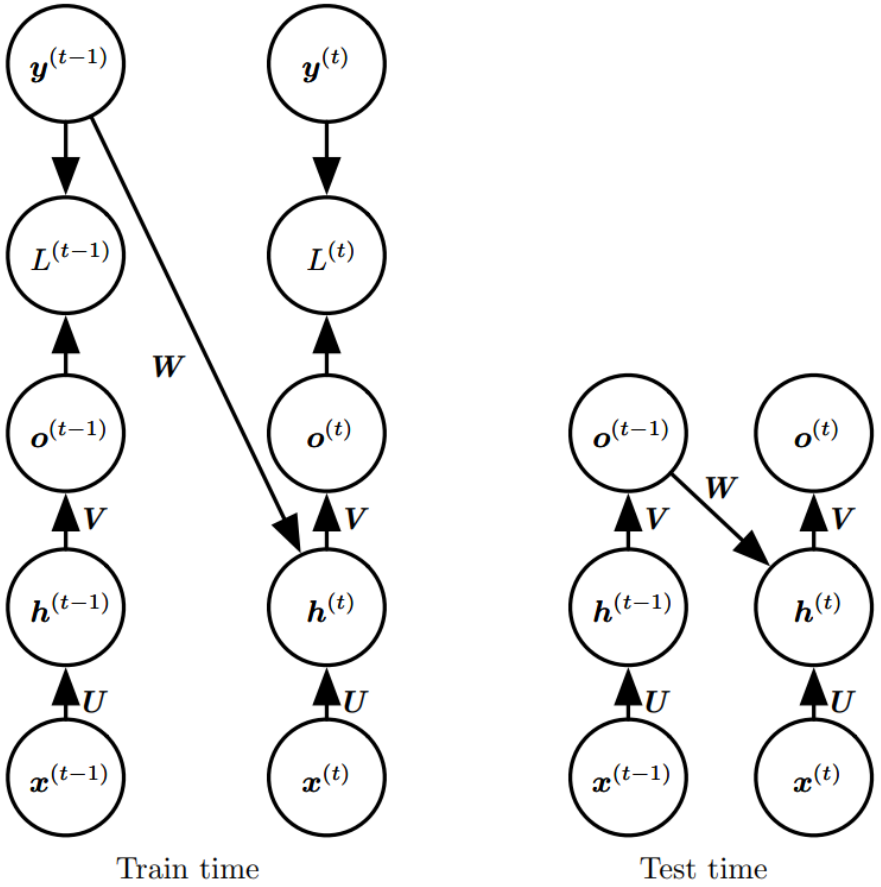
Once we get $\nabla_{\boldsymbol{h}^{(t)}} L$ for each t, we can calculate the gradient w.r.t. the parameters. For example,

$$\begin{aligned}\nabla_{\boldsymbol{W}} L &= \sum_t \sum_i \left(\frac{\partial L}{\partial h_i^{(t)}}\right) \nabla_{\boldsymbol{W}^{(t)}} h_i^{(t)} \\ &= \sum_t \operatorname{diag}\left(1 - \left(\boldsymbol{h}^{(t)}\right)^2\right) \left(\nabla_{\boldsymbol{h}^{(t)}} L\right) \boldsymbol{h}^{(t-1)^\top}\end{aligned}$$

teacher forcing

$$\log p(\boldsymbol{y}^{(1)}, \boldsymbol{y}^{(2)} \mid \boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)})$$
$$= \log p(\boldsymbol{y}^{(2)} \mid \boldsymbol{y}^{(1)}, \boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}) + \log p(\boldsymbol{y}^{(1)} \mid \boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)})$$

depends on
previous output



exploding and vanishing gradients

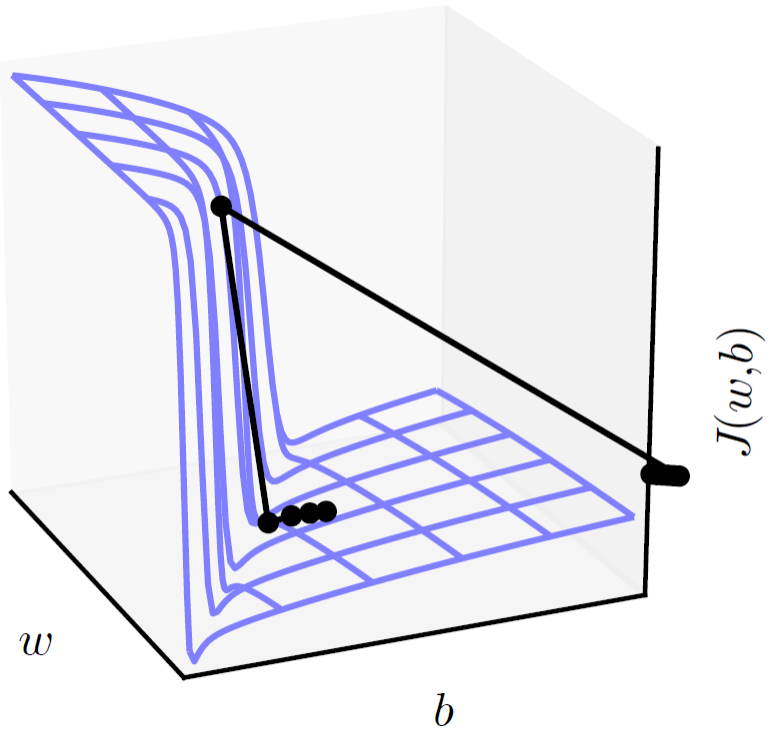
- consider a simple recurrence relation

$$\boldsymbol{h}^{(t)} = \boldsymbol{W}^{\top} \boldsymbol{h}^{(t-1)}$$

- we have $\boldsymbol{h}^{(t)} = (\boldsymbol{W}^{t})^{\top} \boldsymbol{h}^{(0)}$
- assume $\boldsymbol{W} = \boldsymbol{Q} \boldsymbol{\Lambda} \boldsymbol{Q}^{\top}$
- we can further simplify it to

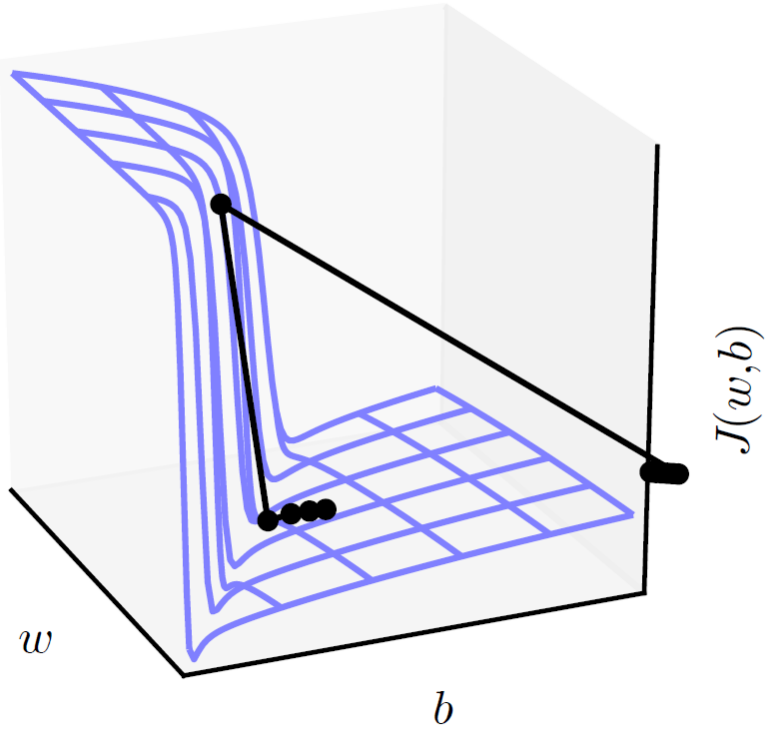
$$\boldsymbol{h}^{(t)} = \boldsymbol{Q}^{\top} \boldsymbol{\Lambda}^{t} \boldsymbol{Q} \boldsymbol{h}^{(0)}$$

exploding gradients



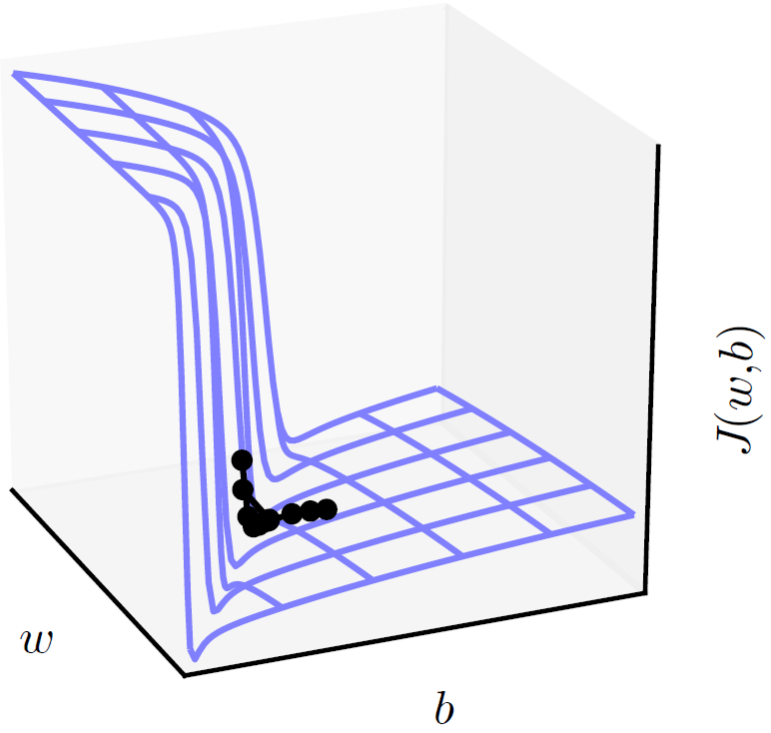
gradient clipping

Without clipping



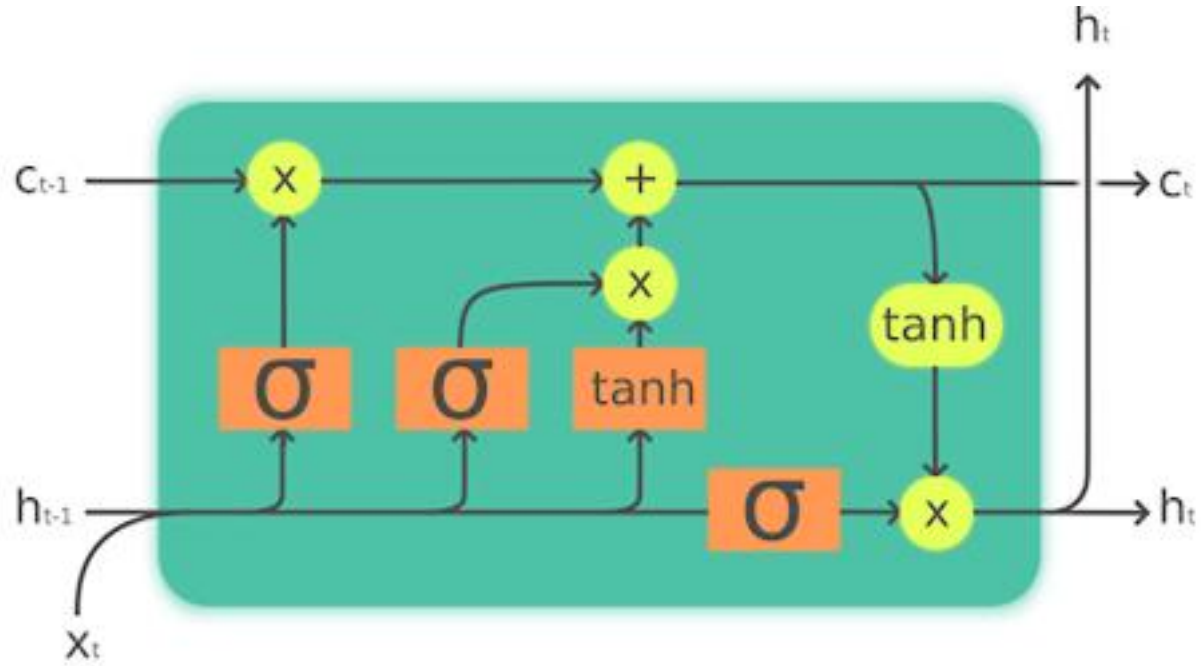
With clipping

$$\text{if } \|\boldsymbol{g}\| > v$$
$$\boldsymbol{g} \leftarrow \frac{\boldsymbol{g}v}{\|\boldsymbol{g}\|}$$



Long Short-Term Memory (LSTM)

Hochreiter & Schmidhuber (1997)



$$\boldsymbol{f}_t = \sigma_g(\boldsymbol{W}_f \boldsymbol{x}_t + \boldsymbol{U}_f \boldsymbol{h}_{t-1} + \boldsymbol{b}_f)$$

$$\boldsymbol{i}_t = \sigma_g(\boldsymbol{W}_i \boldsymbol{x}_t + \boldsymbol{U}_i \boldsymbol{h}_{t-1} + \boldsymbol{b}_i)$$

$$\boldsymbol{o}_t = \sigma_g(\boldsymbol{W}_o \boldsymbol{x}_t + \boldsymbol{U}_o \boldsymbol{h}_{t-1} + \boldsymbol{b}_o)$$

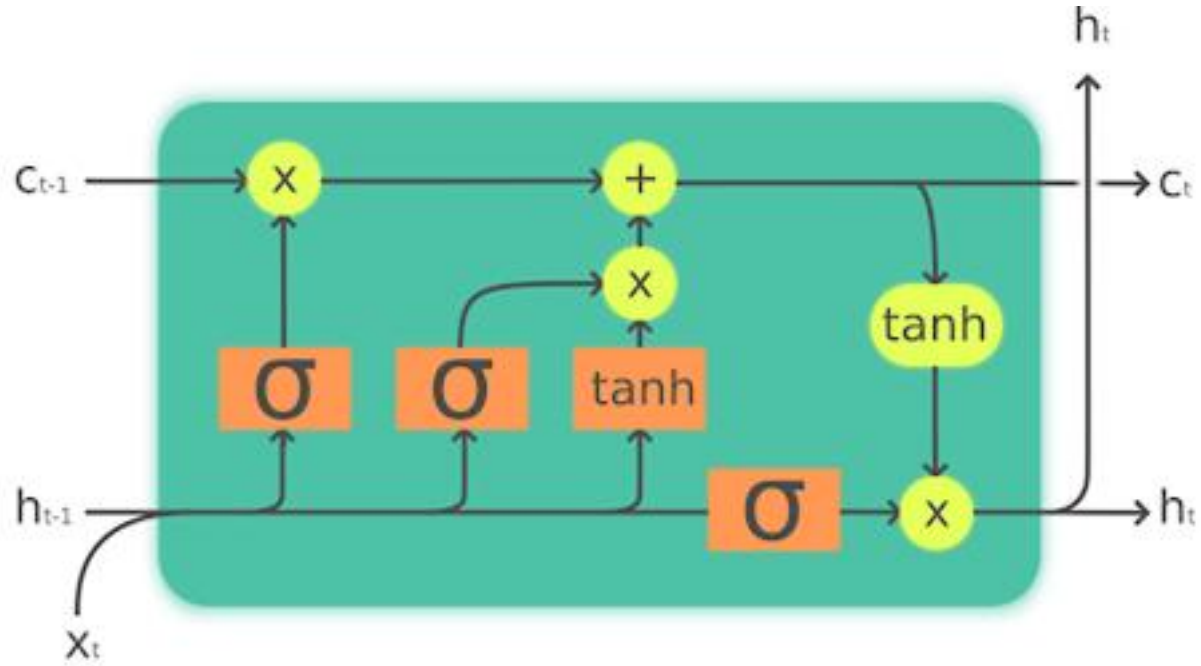
$$\boldsymbol{c}_t = \boldsymbol{f}_t \circ \boldsymbol{c}_{t-1} + \boldsymbol{i}_t \circ \tanh(\boldsymbol{W}_c \boldsymbol{x}_t + \boldsymbol{U}_c \boldsymbol{h}_{t-1} + \boldsymbol{b}_c)$$

$$\boldsymbol{h}_t = \boldsymbol{o}_t \circ \tanh(\boldsymbol{c}_t)$$

image credit: https://colah.github.io/posts/2015-08-Understanding-LSTMs/

Long Short-Term Memory (LSTM)

Hochreiter & Schmidhuber (1997)



forget gate

$$\boldsymbol{f}_t = \sigma_g(\boldsymbol{W}_f \boldsymbol{x}_t + \boldsymbol{U}_f \boldsymbol{h}_{t-1} + \boldsymbol{b}_f)$$

input gate

$$\boldsymbol{i}_t = \sigma_g(\boldsymbol{W}_i \boldsymbol{x}_t + \boldsymbol{U}_i \boldsymbol{h}_{t-1} + \boldsymbol{b}_i)$$

output gate

$$\boldsymbol{o}_t = \sigma_g(\boldsymbol{W}_o \boldsymbol{x}_t + \boldsymbol{U}_o \boldsymbol{h}_{t-1} + \boldsymbol{b}_o)$$

internal state

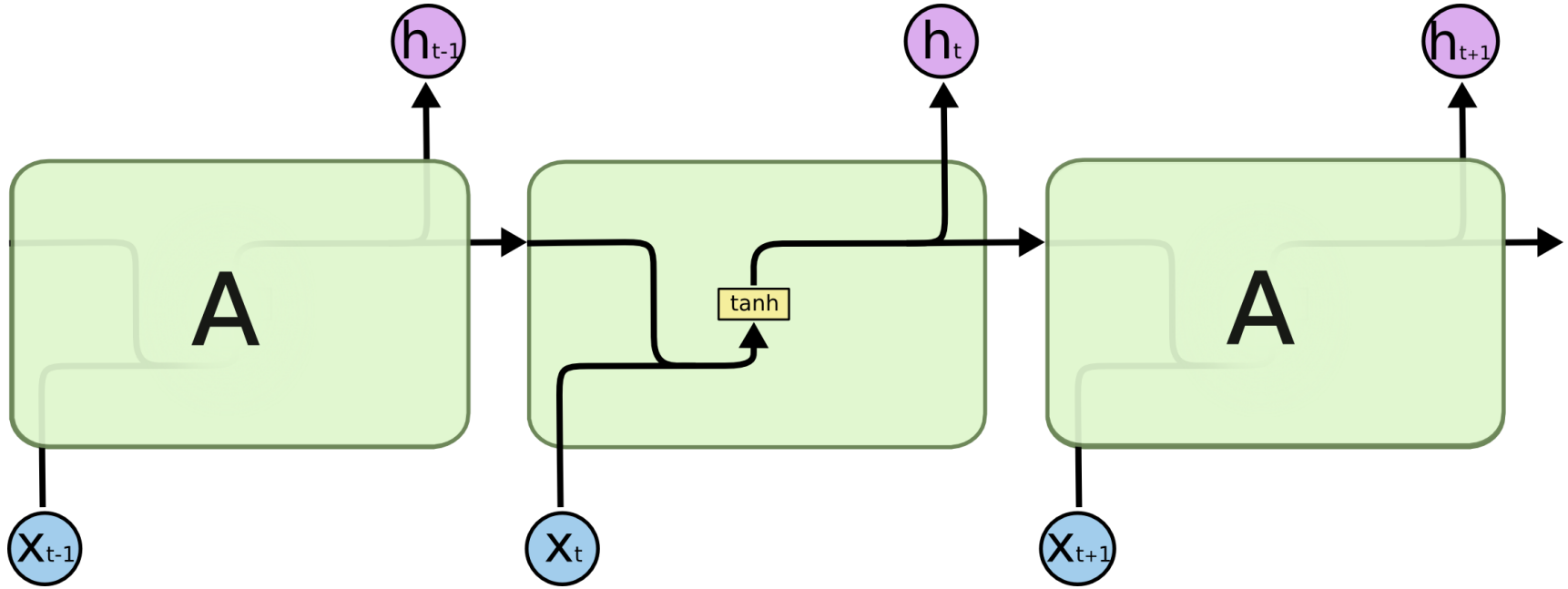
$$\boldsymbol{c}_t = \boldsymbol{f}_t \circ \boldsymbol{c}_{t-1} + \boldsymbol{i}_t \circ \tanh(\boldsymbol{W}_c \boldsymbol{x}_t + \boldsymbol{U}_c \boldsymbol{h}_{t-1} + \boldsymbol{b}_c)$$

output

$$\boldsymbol{h}_t = \boldsymbol{o}_t \circ \tanh(\boldsymbol{c}_t)$$

image credit: https://colah.github.io/posts/2015-08-Understanding-LSTMs/

RNN



LSTM

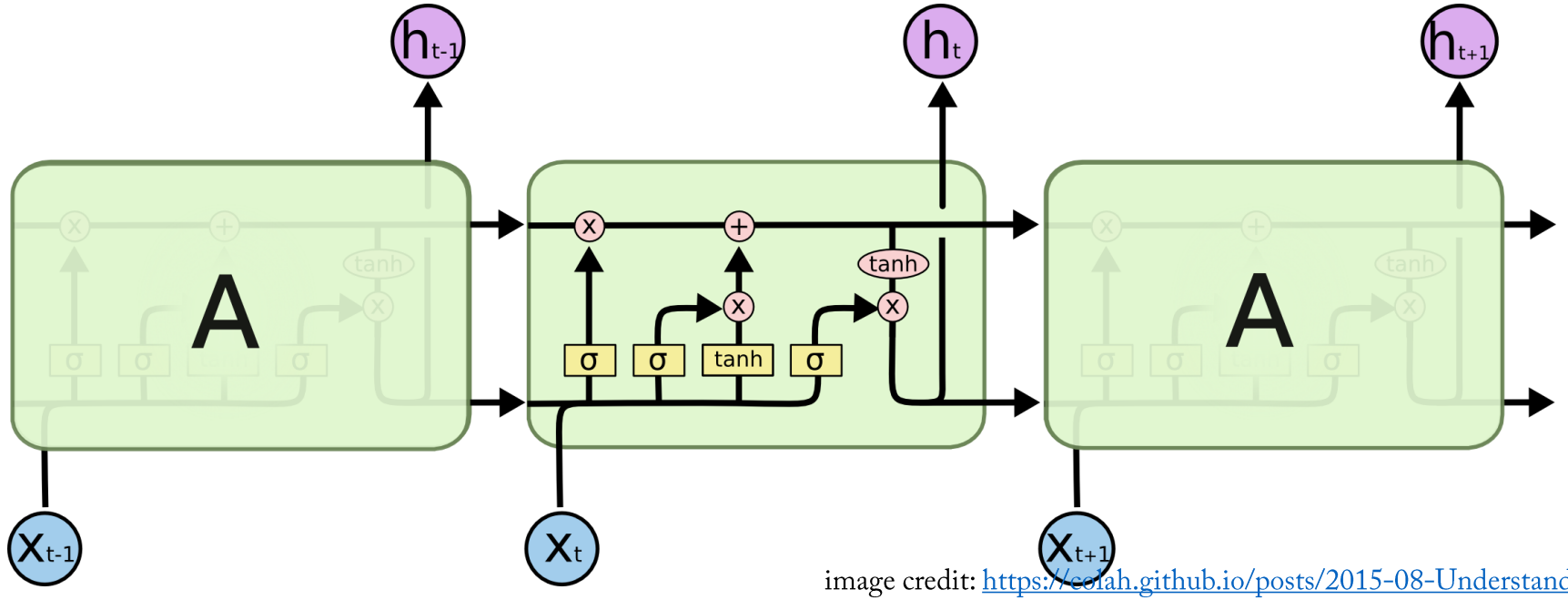
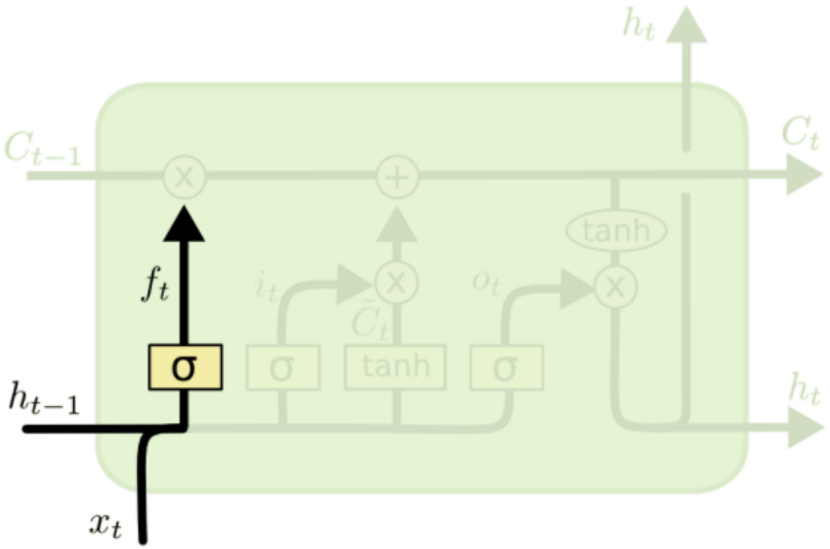


image credit: https://colah.github.io/posts/2015-08-Understanding-LSTMs/

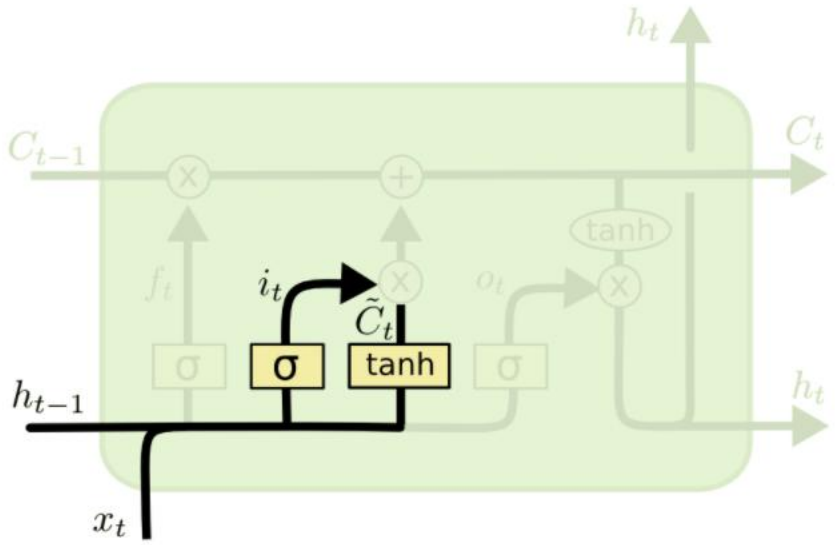
LSTM



forget gate

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \; + \; b_f\right)$$

LSTM

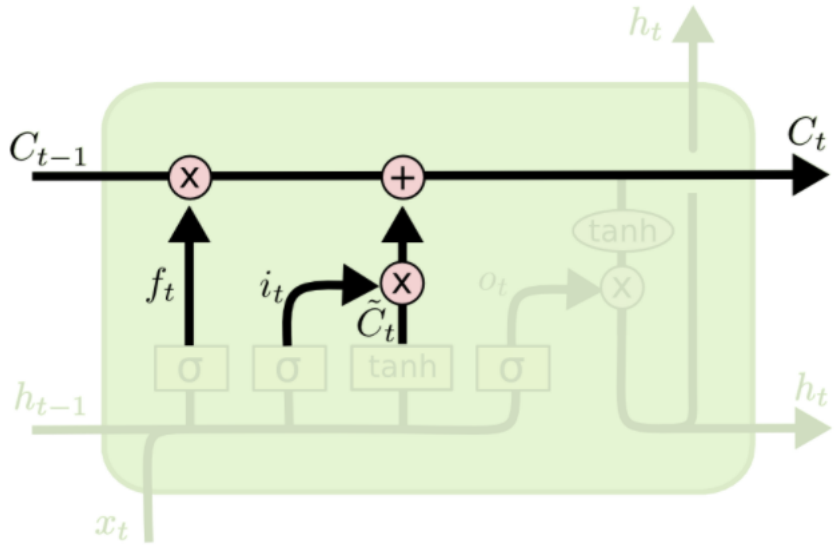


input gate

$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \; + \; b_i\right)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \; + \; b_C)$$

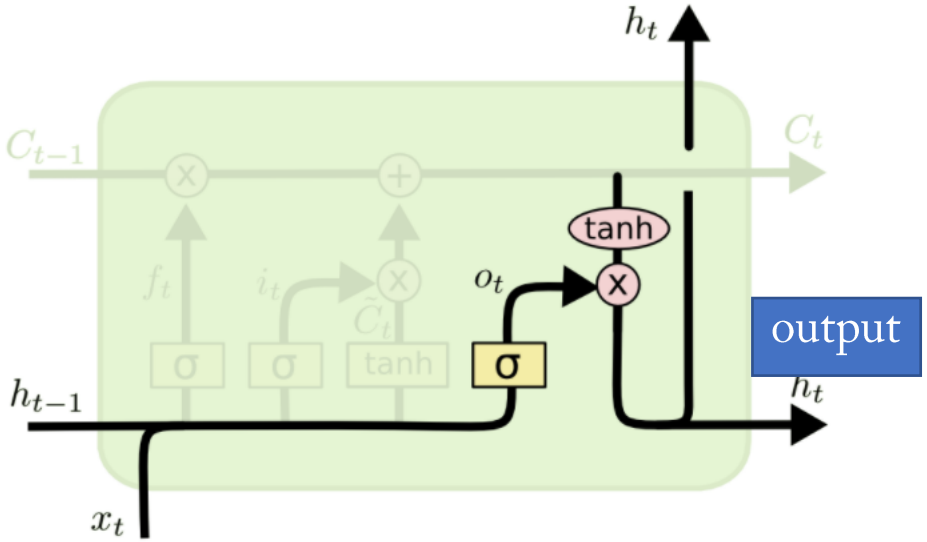
LSTM



internal state

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

LSTM



output gate

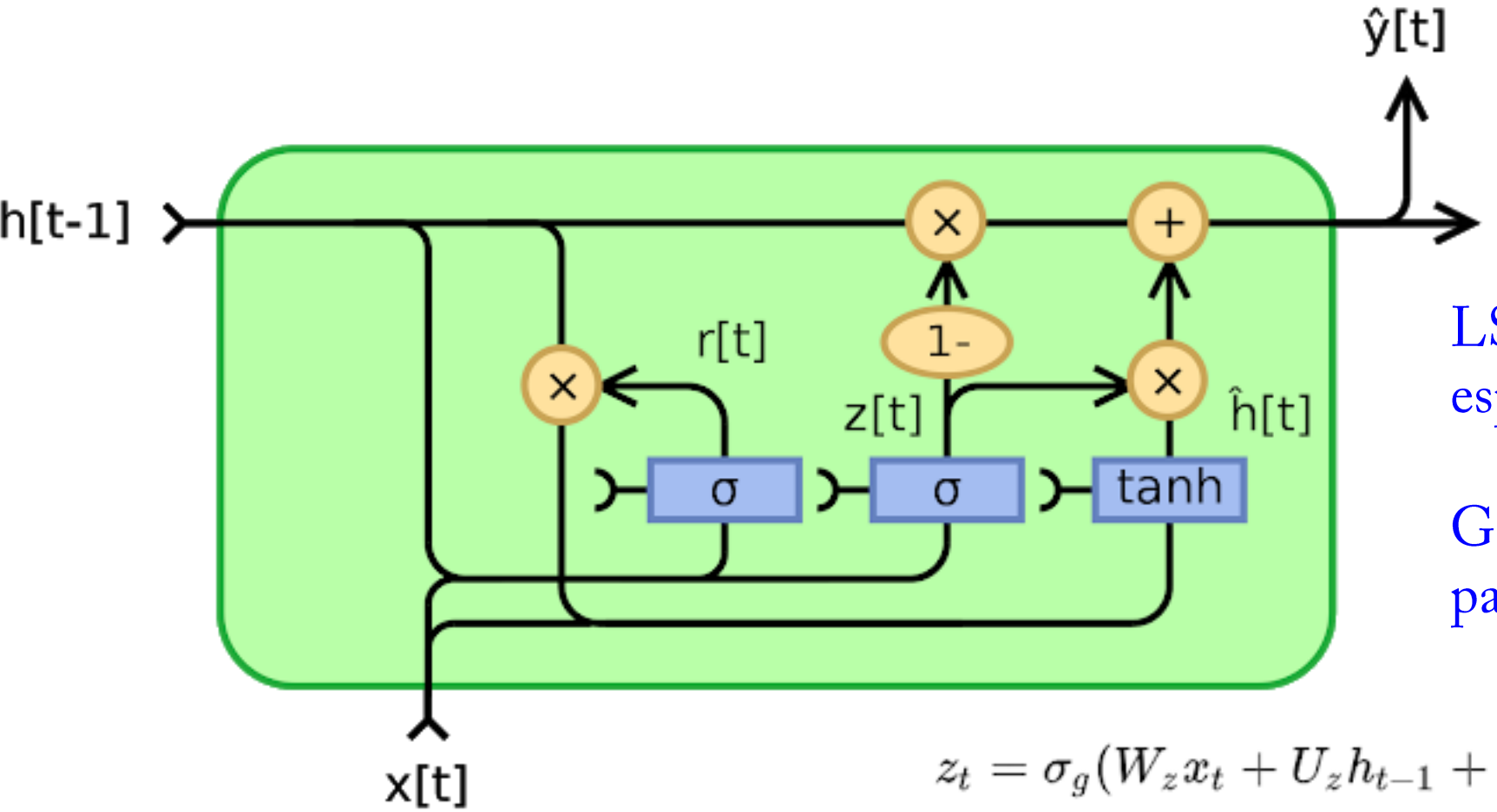
$$o_t = \sigma\left(W_o\left[h_{t-1}, x_t\right] + b_o\right)$$

output

$$h_t = o_t * \tanh\left(C_t\right)$$

Gated Recurrent Unit (GRU)

Cho (2004)



LSTM maybe the first to try especially for long dependence

GRU can be faster and fewer parameters

$$z_t = \sigma_g(W_z x_t + U_z h_{t-1} + b_z)$$
$$r_t = \sigma_g(W_r x_t + U_r h_{t-1} + b_r)$$
$$h_t = z_t \odot h_{t-1} + (1 - z_t) \odot \phi_h(W_h x_t + U_h(r_t \odot h_{t-1}) + b_h)$$

6.2 part-of-speech (POS) tagging

Ch 8, Jurafsky, D., & Martin, J. H. (2018). *Speech and language processing (draft)*. https://web. stanford. edu/~ jurafsky/slp3.

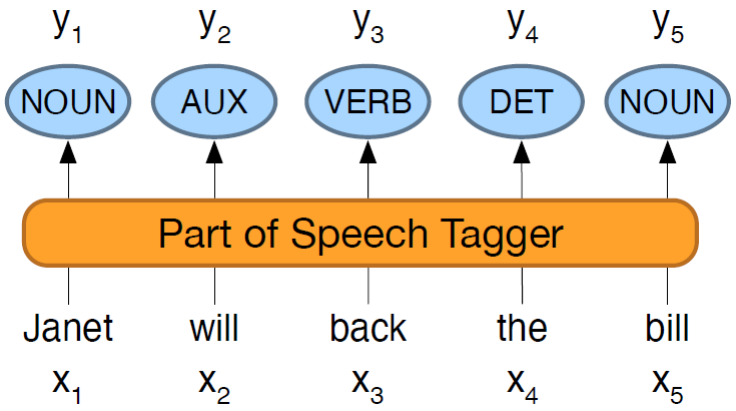
parts of speech

	Tag	Description	Example
Open Class	**ADJ**	Adjective: noun modifiers describing properties	*red*, *young*, *awesome*
	ADV	Adverb: verb modifiers of time, place, manner	*very*, *slowly*, *home*, *yesterday*
	NOUN	words for persons, places, things, etc.	*algorithm*, *cat*, *mango*, *beauty*
	VERB	words for actions and processes	*draw*, *provide*, *go*
	PROPN	Proper noun: name of a person, organization, place, etc..	*Regina*, *IBM*, *Colorado*
	INTJ	Interjection: exclamation, greeting, yes/no response, etc.	*oh*, *um*, *yes*, *hello*
Closed Class Words	**ADP**	Adposition (Preposition/Postposition): marks a noun's spacial, temporal, or other relation	*in, on, by, under*
	AUX	Auxiliary: helping verb marking tense, aspect, mood, etc.,	*can, may, should, are*
	CCONJ	Coordinating Conjunction: joins two phrases/clauses	*and*, *or*, *but*
	DET	Determiner: marks noun phrase properties	*a, an, the, this*
	NUM	Numeral	*one, two, first, second*
	PART	Particle: a function word that must be associated with another word	*'s, not, (infinitive) to*
	PRON	Pronoun: a shorthand for referring to an entity or event	*she, who, I, others*
	SCONJ	Subordinating Conjunction: joins a main clause with a subordinate clause such as a sentential complement	*that*, *which*
Other	**PUNCT**	Punctuation	; , ()
	SYM	Symbols like $ or emoji	$, %
	X	Other	asdf, qwfg

parts of speech: Penn Treebank tags

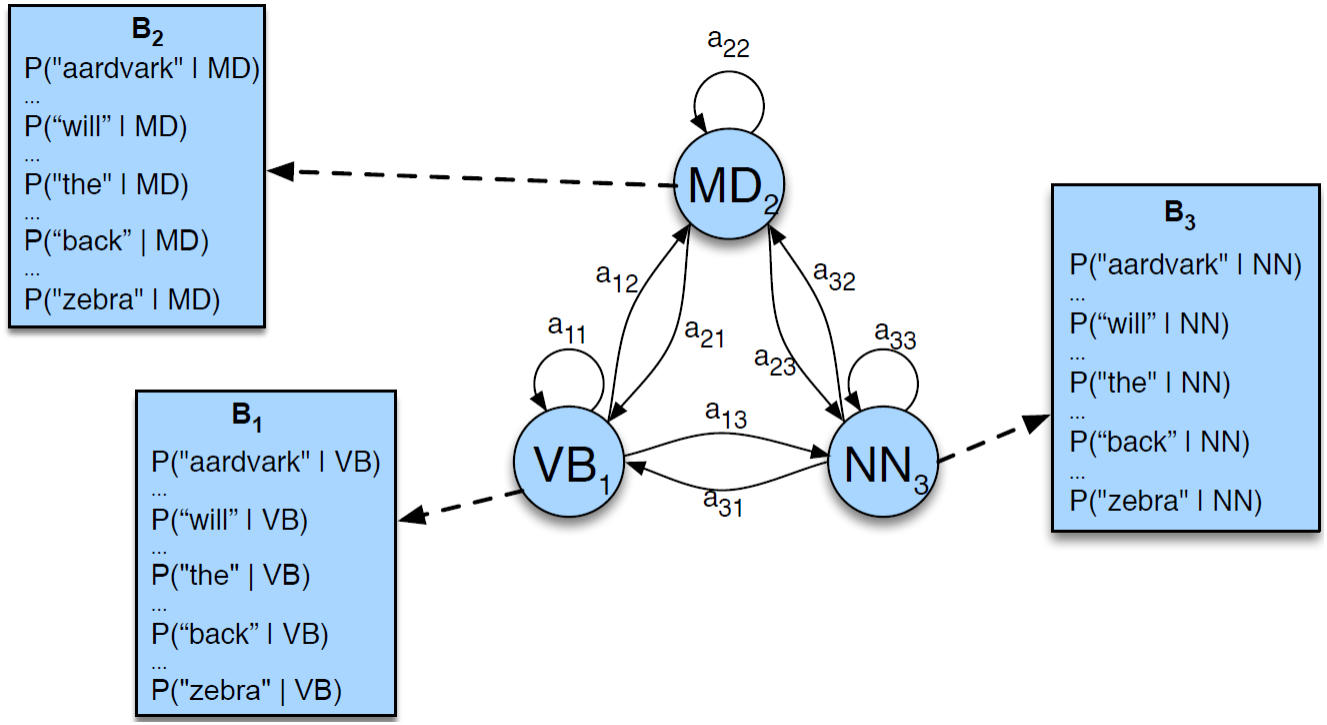
Tag	Description	Example	Tag	Description	Example	Tag	Description	Example
CC	coord. conj.	*and, but, or*	NNP	proper noun, sing.	*IBM*	TO	"to"	*to*
CD	cardinal number	*one, two*	NNPS	proper noun, plu.	*Carolinas*	UH	interjection	*ah, oops*
DT	determiner	*a, the*	NNS	noun, plural	*llamas*	VB	verb base	*eat*
EX	existential 'there'	*there*	PDT	predeterminer	*all, both*	VBD	verb past tense	*ate*
FW	foreign word	*mea culpa*	POS	possessive ending	*'s*	VBG	verb gerund	*eating*
IN	preposition/ subordin-conj	*of, in, by*	PRP	personal pronoun	*I, you, he*	VBN	verb past participle	*eaten*
JJ	adjective	*yellow*	PRP$	possess. pronoun	*your, one's*	VBP	verb non-3sg-pr	*eat*
JJR	comparative adj	*bigger*	RB	adverb	*quickly*	VBZ	verb 3sg pres	*eats*
JJS	superlative adj	*wildest*	RBR	comparative adv	*faster*	WDT	wh-determ.	*which, that*
LS	list item marker	*1, 2, One*	RBS	superlatv. adv	*fastest*	WP	wh-pronoun	*what, who*
MD	modal	*can, should*	RP	particle	*up, off*	WP$	wh-possess.	*whose*
NN	sing or mass noun	*llama*	SYM	symbol	+,%, &	WRB	wh-adverb	*how, where*

tagging



- ❑ Naïve idea: check the dictionary for tagging
- ❑ Problem: not unique
- ✓ Sequential labeling: tagging according to the context

Hidden-Markov Model (HMM)



Hidden-Markov Model (HMM)

$Q = q_1 q_2 \ldots q_N$	a set of N **states**
$A = a_{11} \ldots a_{ij} \ldots a_{NN}$	a **transition probability matrix** A, each a_{ij} representing the probability of moving from state i to state j, s.t. $\sum_{j=1}^{N} a_{ij} = 1 \quad \forall i$
$O = o_1 o_2 \ldots o_T$	a sequence of T **observations**, each one drawn from a vocabulary $V = v_1, v_2, \ldots, v_V$
$B = b_i(o_t)$	a sequence of **observation likelihoods**, also called **emission probabilities**, each expressing the probability of an observation o_t being generated from a state q_i
$\pi = \pi_1, \pi_2, \ldots, \pi_N$	an **initial probability distribution** over states. π_i is the probability that the Markov chain will start in state i. Some states j may have $\pi_j = 0$, meaning that they cannot be initial states. Also, $\sum_{i=1}^{n} \pi_i = 1$

Hidden-Markov Model (HMM)

- Markov assumption:

$$P(q_i|q_1,...,q_{i-1}) = P(q_i|q_{i-1})$$

- output independence:

$$P(o_i|q_1,\ldots q_i,\ldots,q_T,o_1,\ldots,o_i,\ldots,o_T) = P(o_i|q_i)$$

Hidden-Markov Model (HMM)

- transition probability
  - e.g. modal verbs [will] are very likely to be followed by a verb in the base form [eat]
  - In the WSJ corpus, we can calculate

$$P(VB|MD) = \frac{C(MD,VB)}{C(MD)} = \frac{10471}{13124} = .80$$

Hidden-Markov Model (HMM)

- emission probability
  - ❑In the WSJ corpus, we can calculate

$$P(will|MD) = \frac{C(MD, will)}{C(MD)} = \frac{4046}{13124} = .31$$

Hidden-Markov Model (HMM)

- **decoding**: given as input an HMM and a sequence of observations, find the most probable sequence of states

$$\hat{t}_{1:n} = \underset{t_1 \ldots t_n}{\operatorname{argmax}} P(t_1 \ldots t_n | w_1 \ldots w_n)$$

Hidden-Markov Model (HMM)

$$\hat{t}_{1:n} = \underset{t_1 \dots t_n}{\operatorname{argmax}} P(t_1 \dots t_n | w_1 \dots w_n)$$

$$\hat{t}_{1:n} = \underset{t_1 \dots t_n}{\operatorname{argmax}} \frac{P(w_1 \dots w_n | t_1 \dots t_n) P(t_1 \dots t_n)}{P(w_1 \dots w_n)}$$

$$\hat{t}_{1:n} = \underset{t_1 \dots t_n}{\operatorname{argmax}} P(w_1 \dots w_n | t_1 \dots t_n) P(t_1 \dots t_n)$$

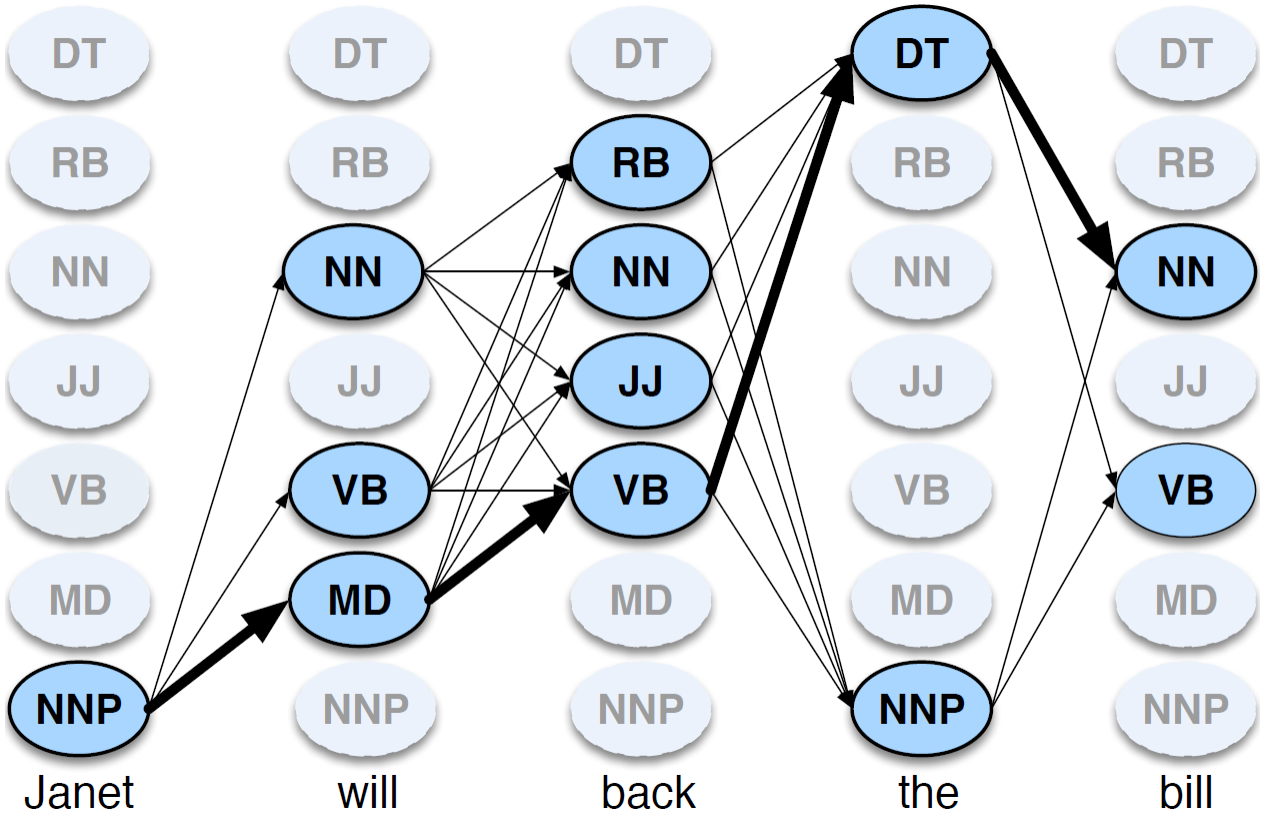
$$\prod_{i=1}^{n} P(w_i | t_i)$$

$$\prod_{i=1}^{n} P(t_i | t_{i-1})$$

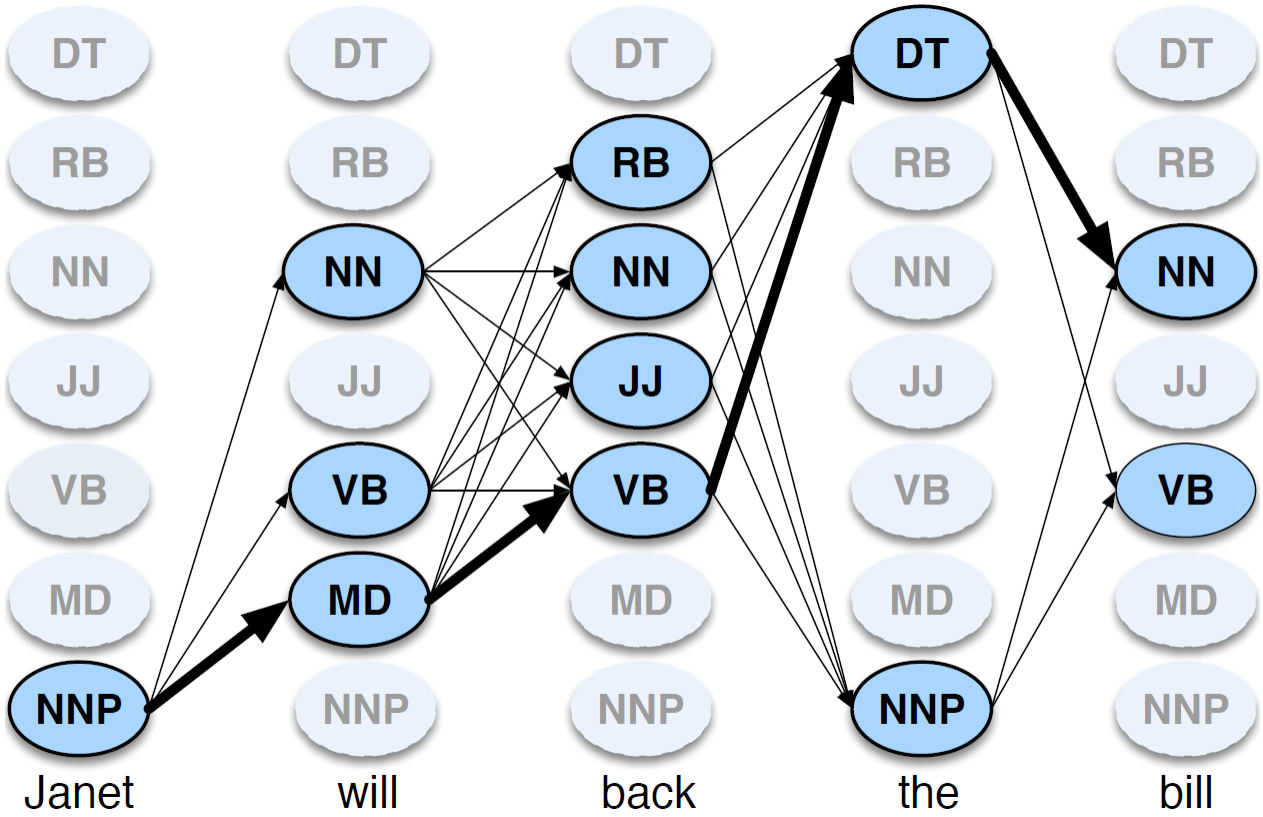
Hidden-Markov Model (HMM)

$$\hat{t}_{1:n} = \underset{t_1 \dots t_n}{\operatorname{argmax}} P(t_1 \dots t_n | w_1 \dots w_n) \approx \underset{t_1 \dots t_n}{\operatorname{argmax}} \prod_{i=1}^{n} \overbrace{P(w_i|t_i)}^{\text{emission}} \overbrace{P(t_i|t_{i-1})}^{\text{transition}}$$

Viterbi algorithm (used on top of HMM in this case)



Viterbi algorithm



each cell $v_t(j) := \max_{q_1,\ldots,q_{t-1}} P(q_1,\ldots,q_{t-1}; o_1,\ldots,o_t, q_t = j|\text{HMM}) = \max_{i=1,\ldots,N} v_{t-1}(i) a_{ij} b_j(o_t)$

Viterbi algorithm

transition probability a_{ij}

	NNP	MD	VB	JJ	NN	RB	DT
$<s>$	0.2767	0.0006	0.0031	0.0453	0.0449	0.0510	0.2026
NNP	0.3777	0.0110	0.0009	0.0084	0.0584	0.0090	0.0025
MD	0.0008	0.0002	0.7968	0.0005	0.0008	0.1698	0.0041
VB	0.0322	0.0005	0.0050	0.0837	0.0615	0.0514	0.2231
JJ	0.0366	0.0004	0.0001	0.0733	0.4509	0.0036	0.0036
NN	0.0096	0.0176	0.0014	0.0086	0.1216	0.0177	0.0068
RB	0.0068	0.0102	0.1011	0.1012	0.0120	0.0728	0.0479
DT	0.1147	0.0021	0.0002	0.2157	0.4744	0.0102	0.0017

Viterbi algorithm

state observation likelihood $b_j(o)$

	Janet	will	back	the	bill
NNP	0.000032	0	0	0.000048	0
MD	0	0.308431	0	0	0
VB	0	0.000028	0.000672	0	0.000028
JJ	0	0	0.000340	0	0
NN	0	0.000200	0.000223	0	0.002337
RB	0	0	0.010446	0	0
DT	0	0	0	0.506099	0

Viterbi algorithm

- Let's illustrate the algorithm using a simpler and unrealistic example:

"Cats cannot fly."

Initial distribution and transition probabilities

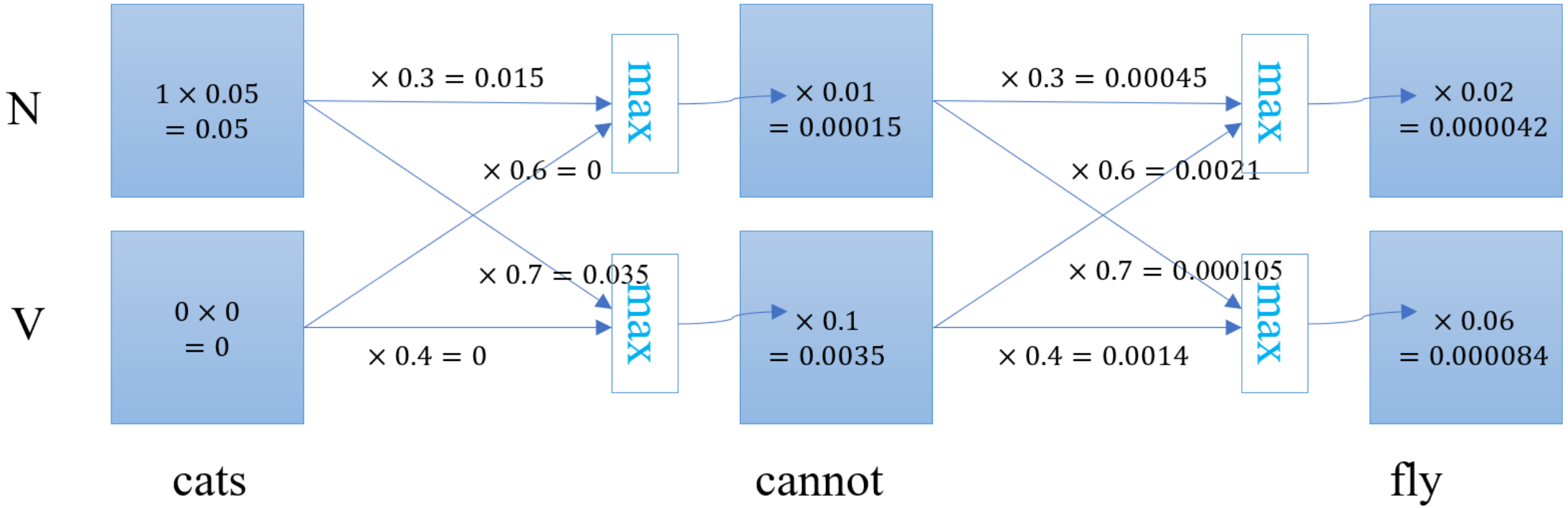
	N	V
initial	1	0
N	0.3	0.7
V	0.6	0.4

Emission probabilities

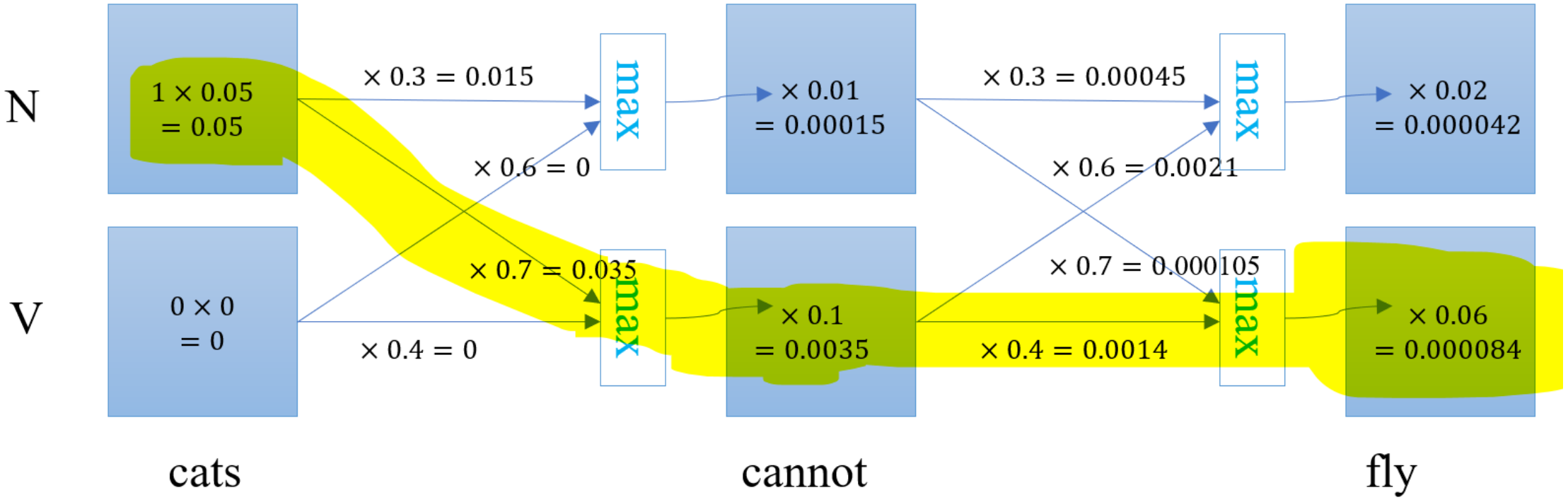
	cats	cannot	fly
N	0.05	0.01	0.02
V	0	0.1	0.06

Viterbi algorithm

each cell $v_t(j) := \max_{q_1,\ldots,q_{t-1}} P(q_1,\ldots,q_{t-1};o_1,\ldots,o_t,q_t=j|\text{HMM}) = \max_{i=1,\ldots,N} v_{t-1}(i)a_{ij}b_j(o_t)$



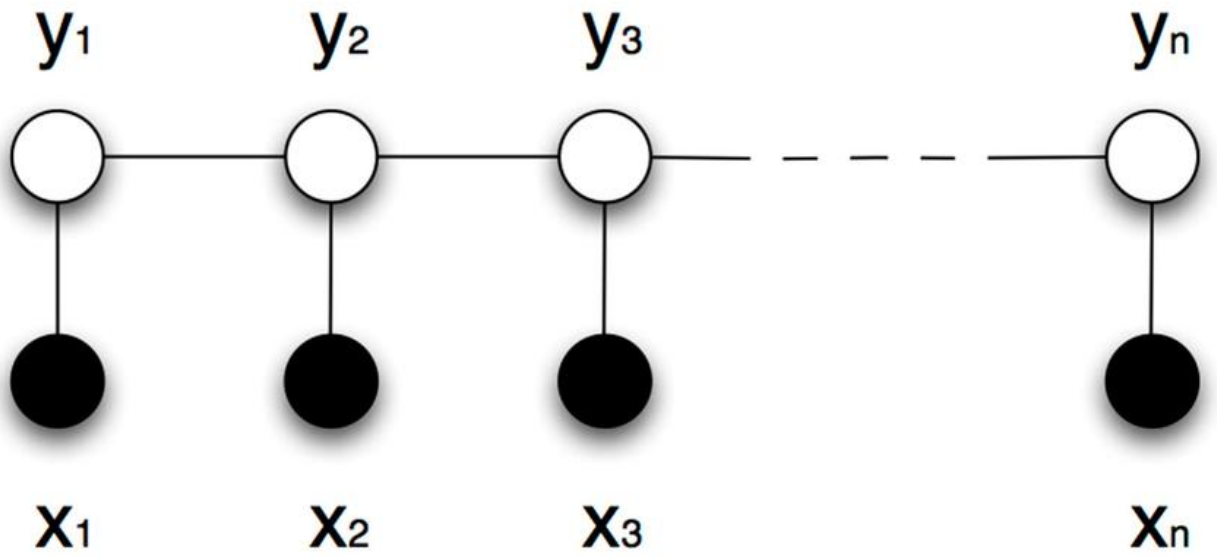
Viterbi algorithm



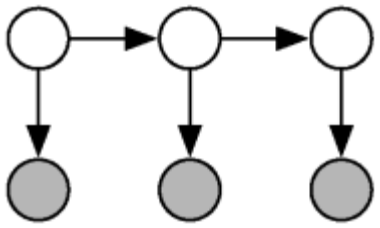
conditional random field (CRF)

sequence of output tags

sequence of input words

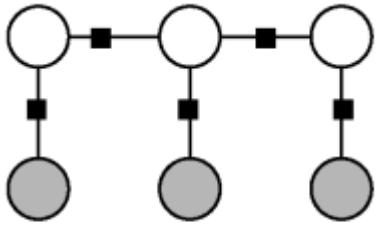


generative

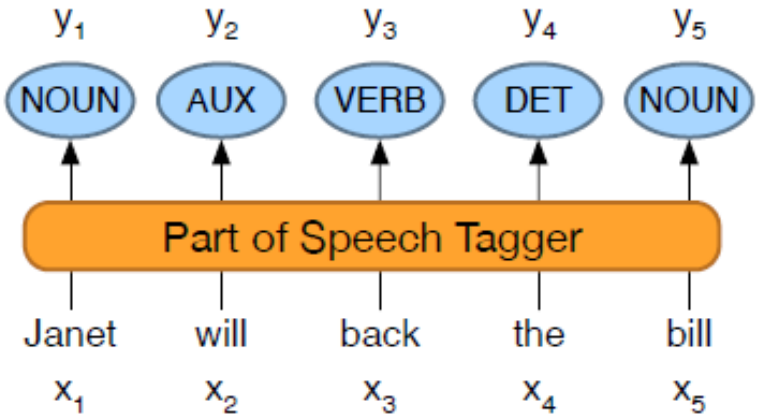


HMMs

discriminative

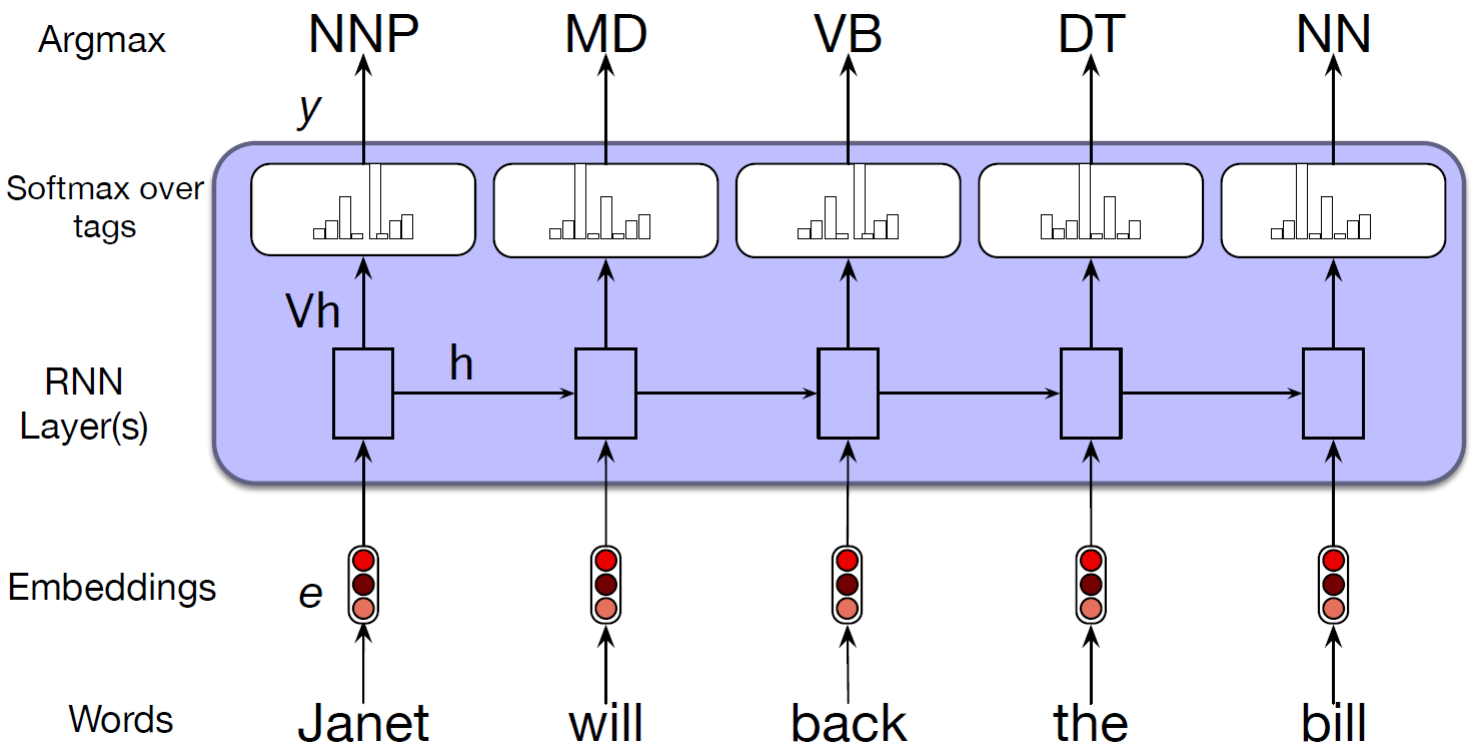


Linear-chain CRFs



§8.5, Jurafsky, D., & Martin, J. H. (2018). *Speech and language processing (draft)*. https://web. stanford. edu/~ jurafsky/slp3.

RNN for POS tagging



Let’s use LSTM for POS tagging!

- click here

6.3 RNN for forcasting

RNN for forecasting

https://eng.uber.com/m4-forecasting-competition/

RNN for forecasting

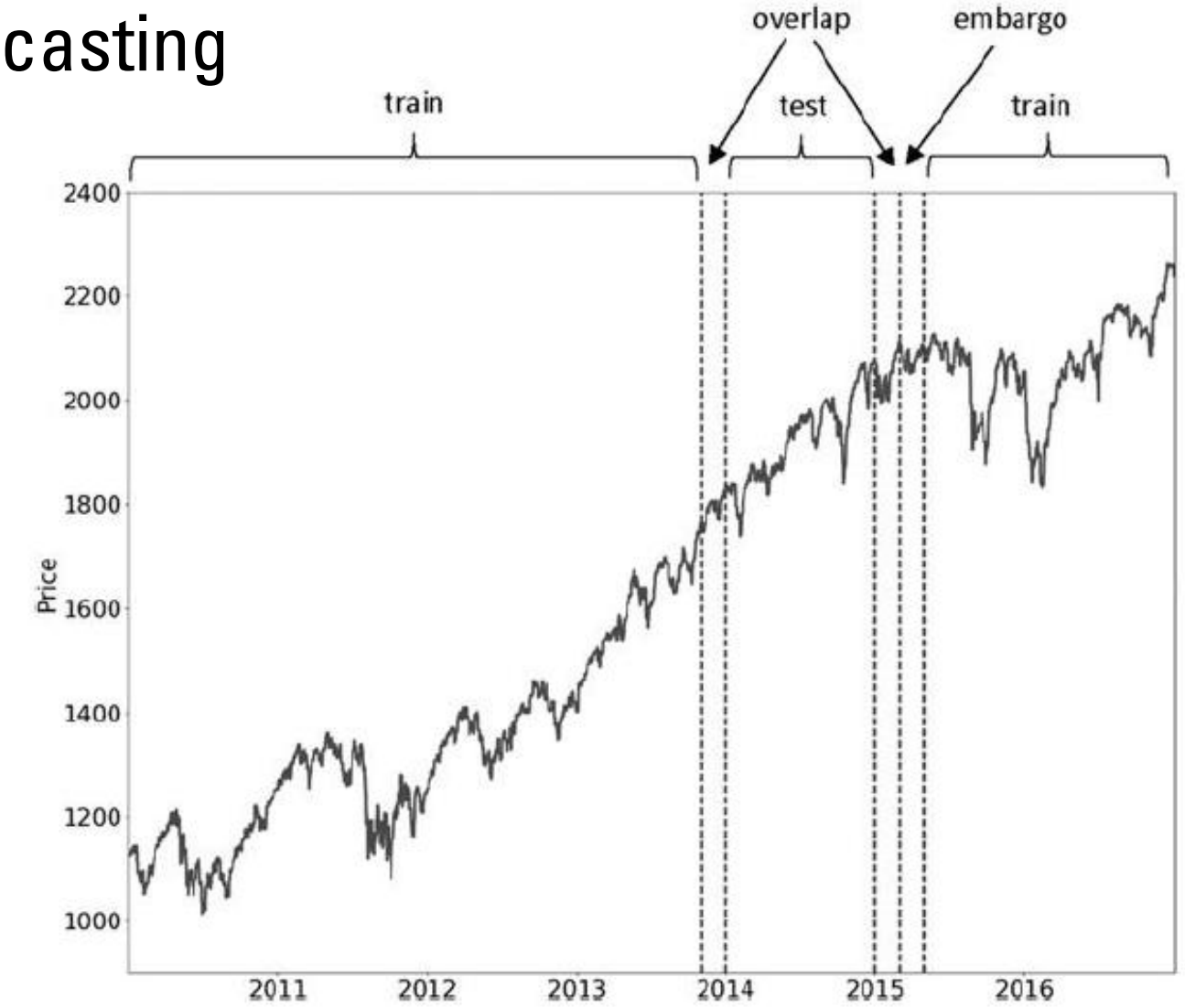


image credit: De Prado, M. L. (2018). *Advances in financial machine learning*. John Wiley & Sons.

6.4 machine translation

machine translation: difficulty

English: *He wrote a letter to a friend*

Japanese: *tomodachi ni tegami-o kaita*

friend to letter wrote

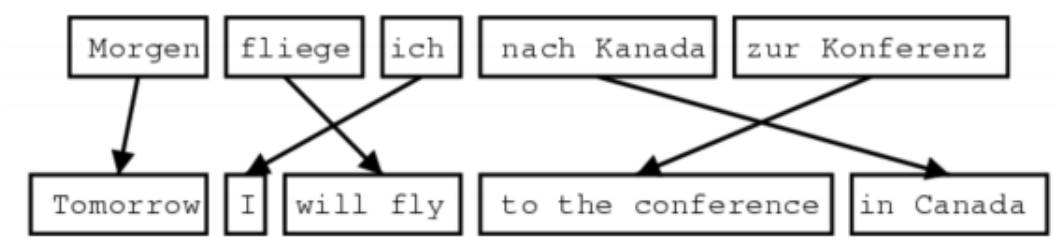
machine translation: difficulty

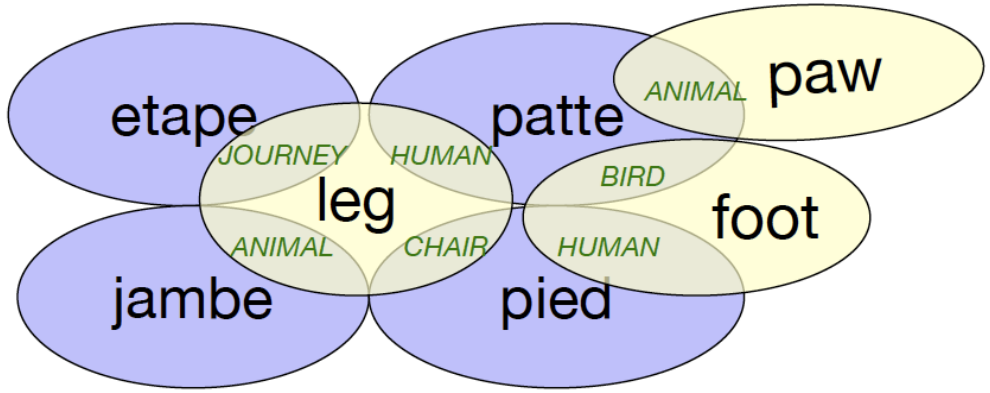
大会/General Assembly 在/on 1982年/1982 12月/December 10日/10 通过了/adopted 第37号/37th 决议/resolution， 核准了/approved 第二次/second 探索/exploration 及/and 和平peaceful 利用/using 外层空间/outer space 会议/conference 的/of 各项/various 建议/suggestions 。

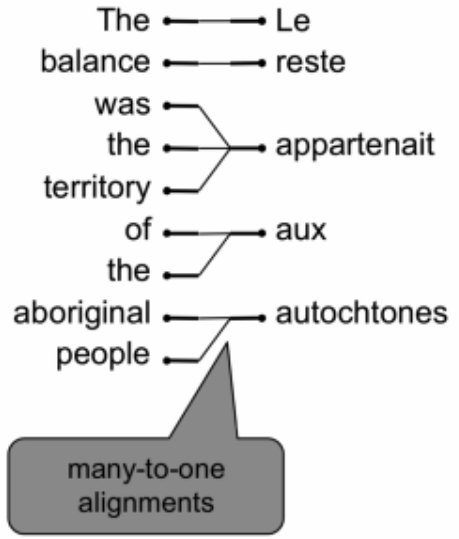
On 10 December 1982 , the General Assembly adopted resolution 37 in which it endorsed the recommendations of the Second United Nations Conference on the Exploration and Peaceful Uses of Outer Space .

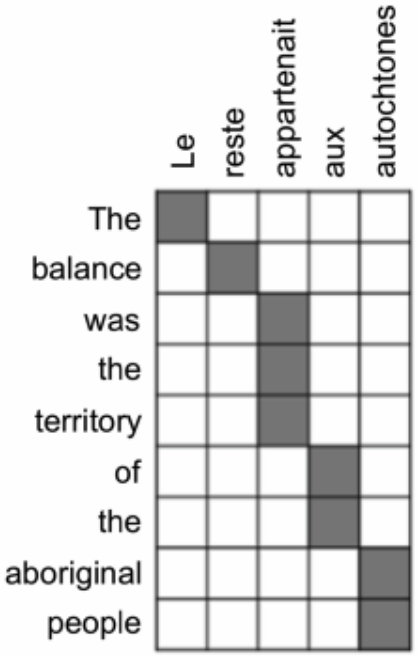
encoder-decoder handles well this sort of difficulty

machine translation: difficulty



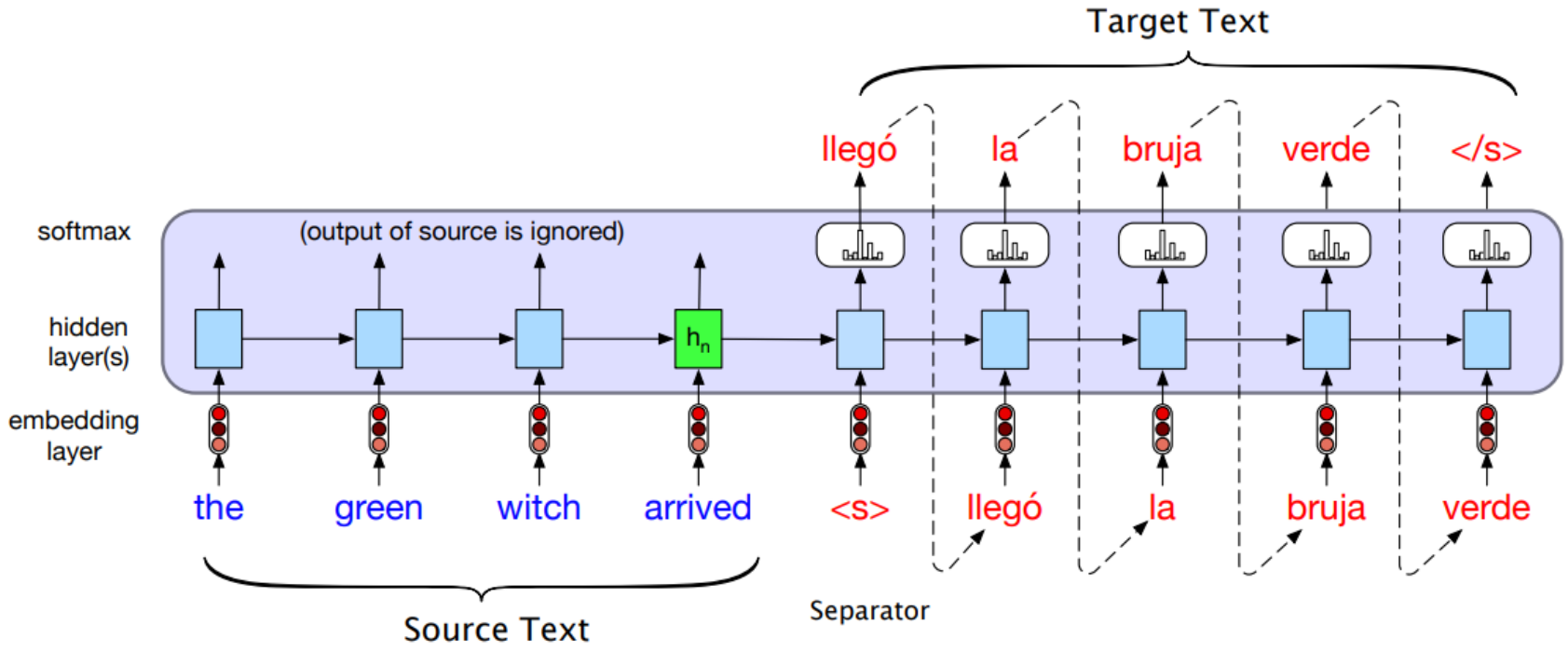




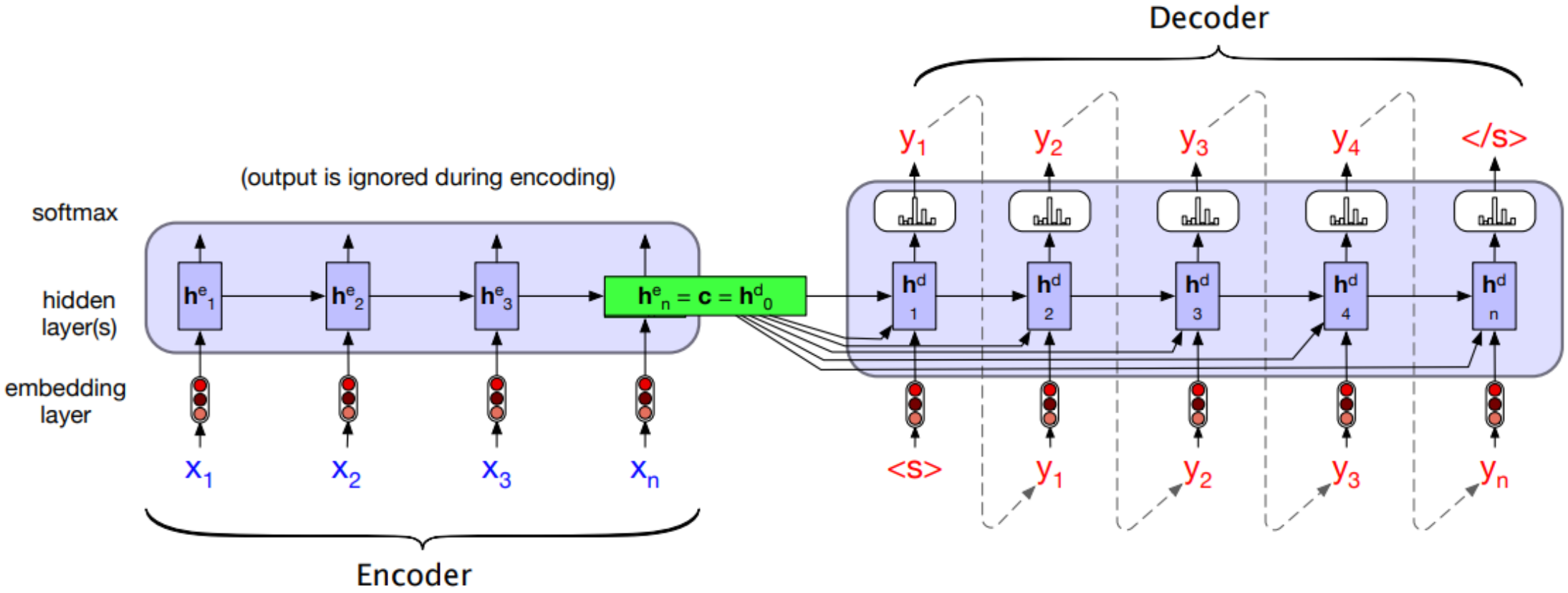


machine translation

Translating a single sentence (inference time) in the basic RNN version of encoder-decoder (sequence-to-sequence) approach to machine translation. Source and target sentences are concatenated with a separator token in between, and the decoder uses context information from the encoder's last hidden state.



machine translation

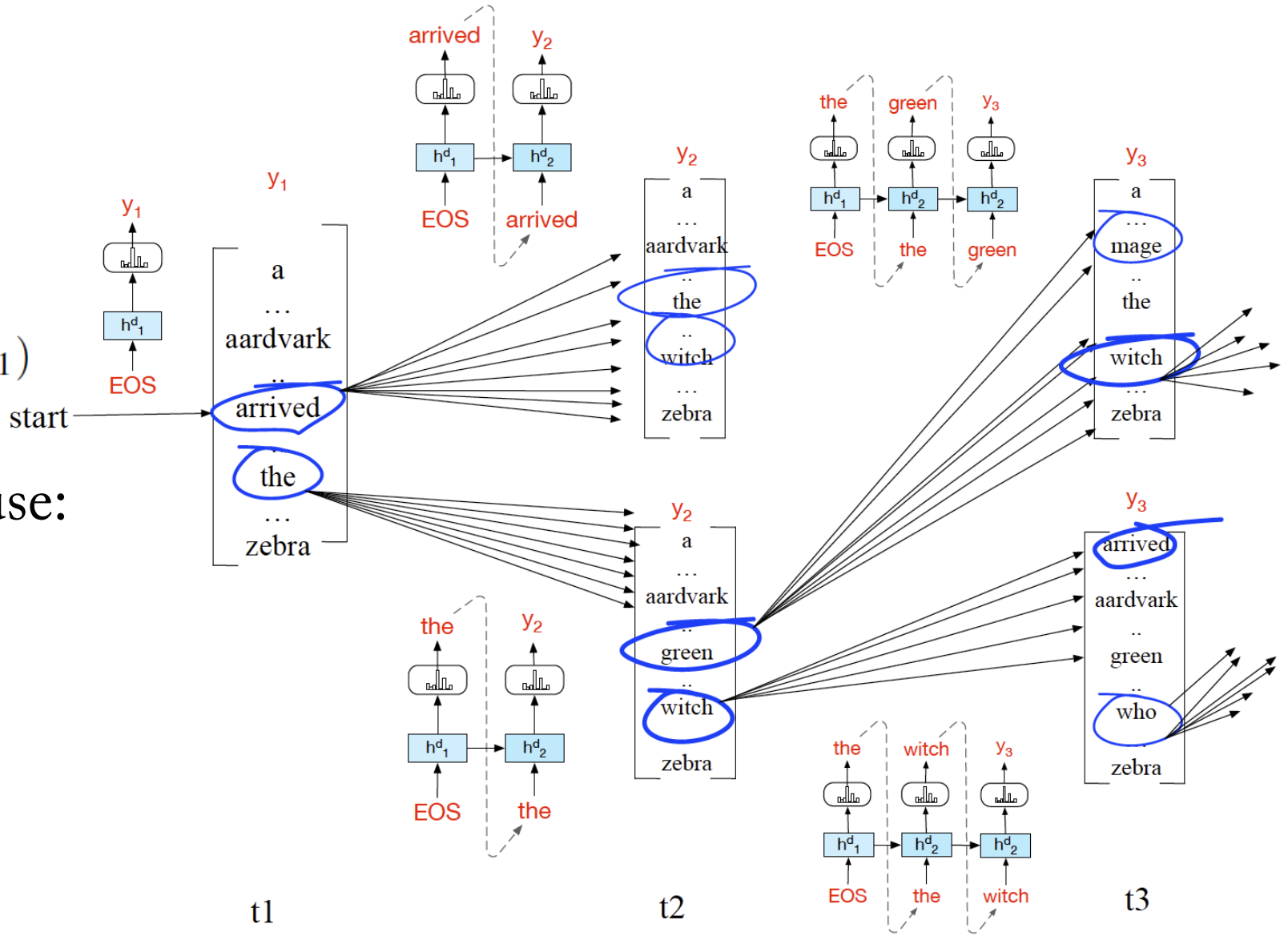


decoding

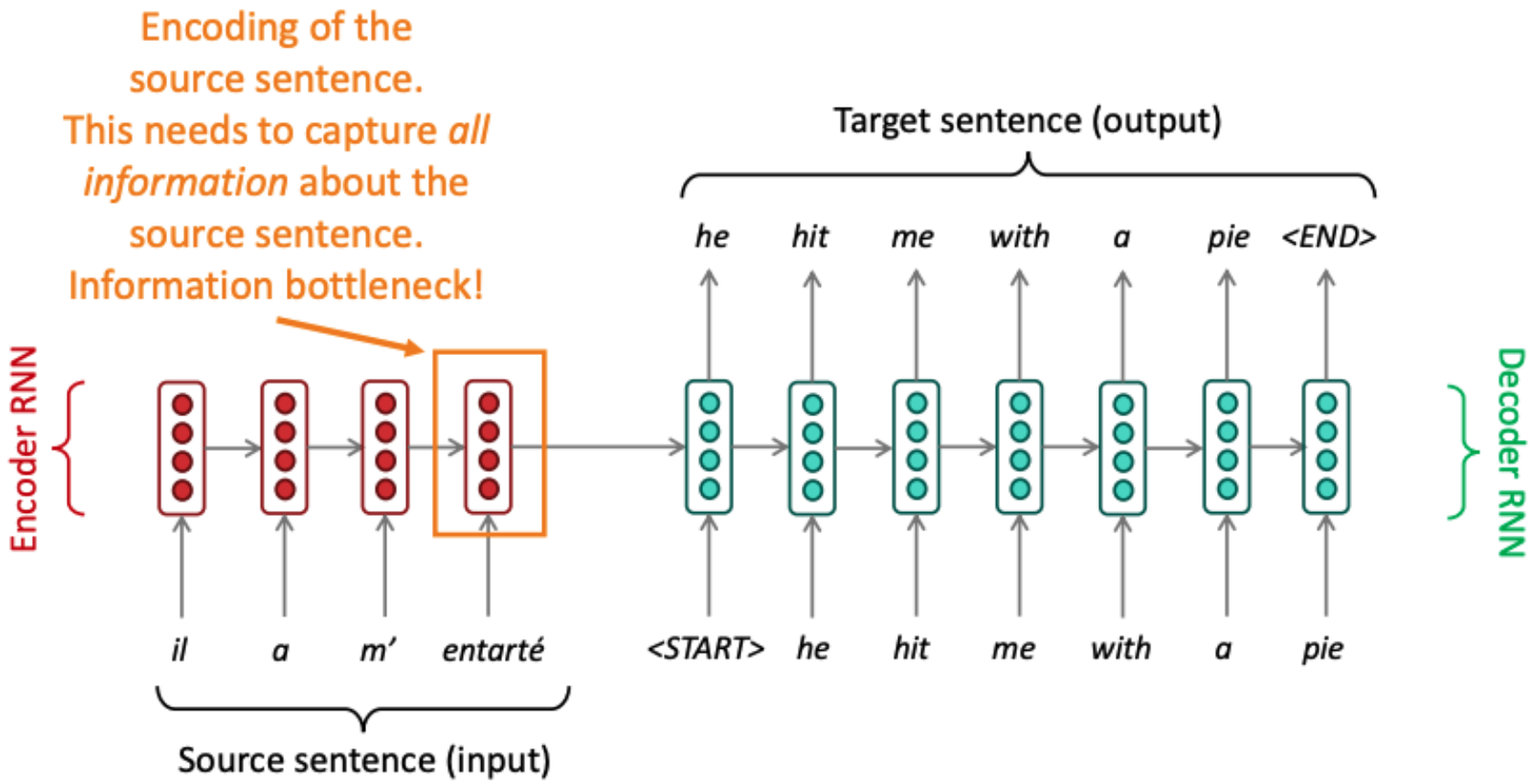
- greedy:

$$\hat{y}_t = \text{argmax}_{w \in V} P(w|x, y_1 \ldots y_{t-1})$$

- greedy is not optimal, use:
  - beam-search

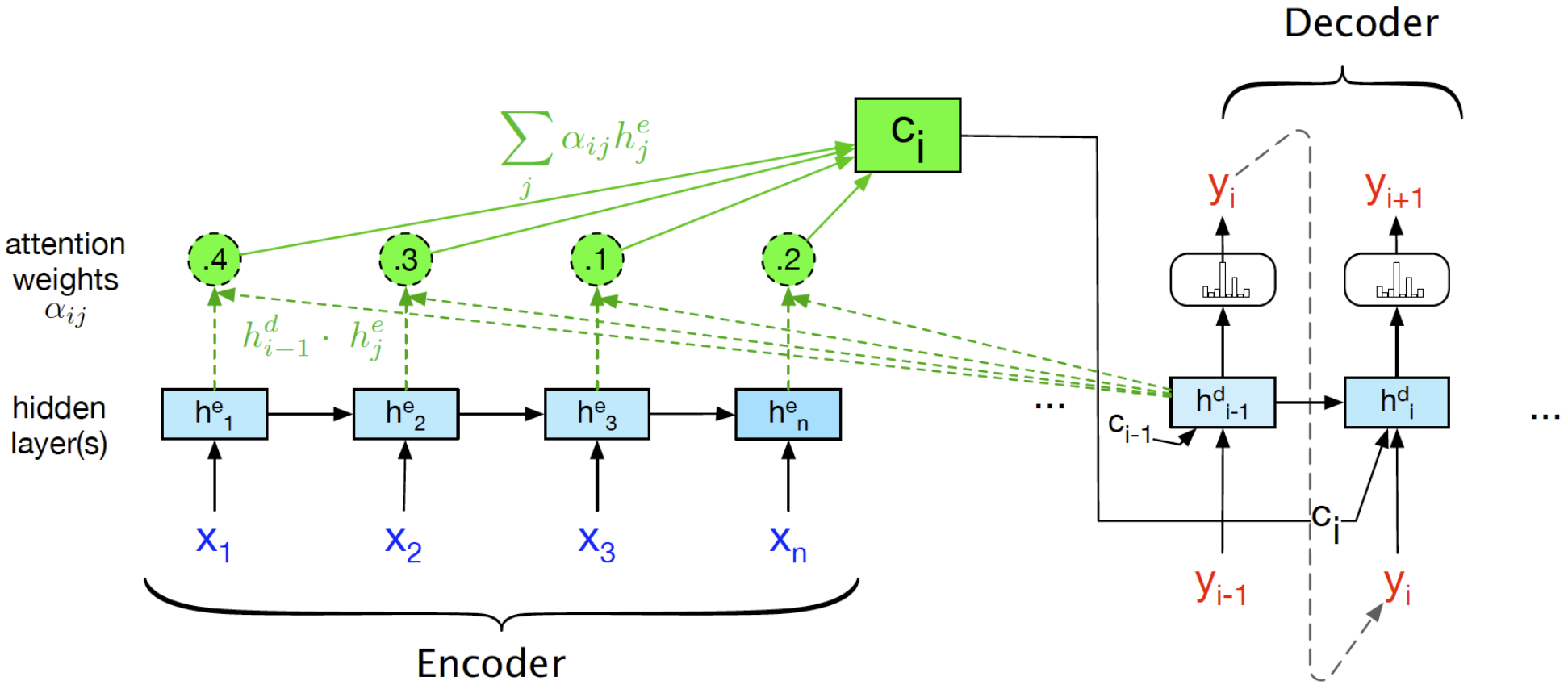


drawback of encoder-decoder



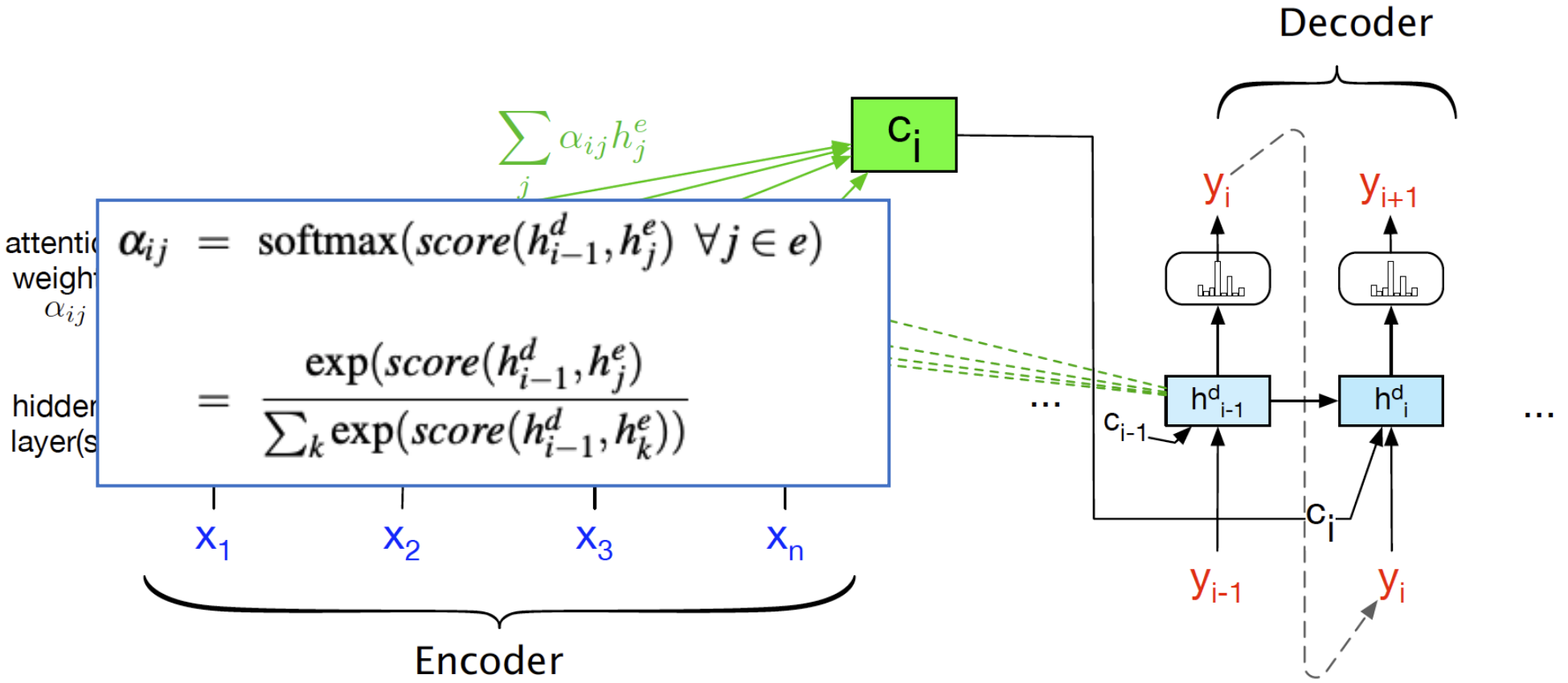
machine translation: bottleneck

- solved by "attention" – to be covered

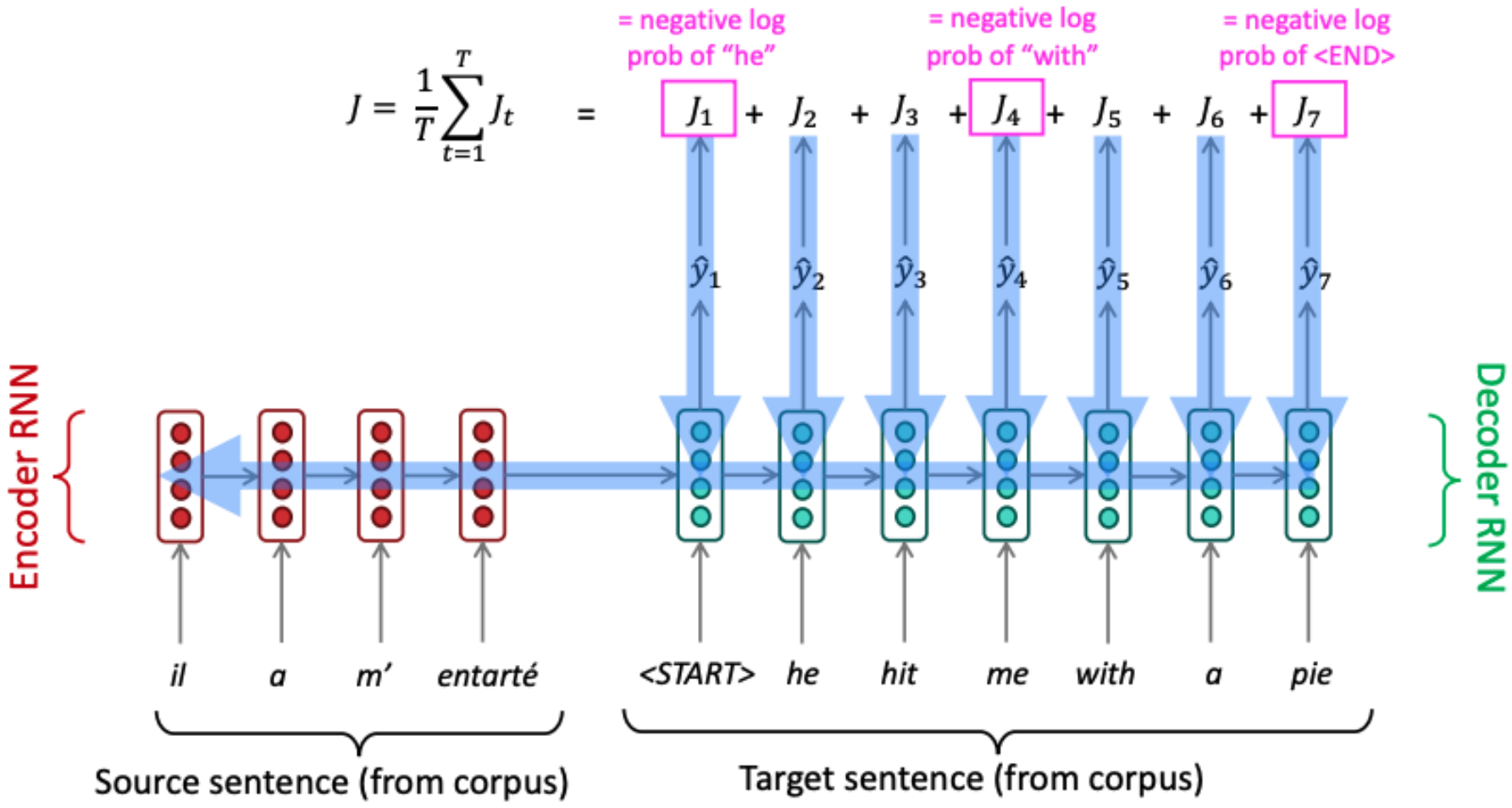


machine translation: bottleneck

- solved by "attention" – to be covered



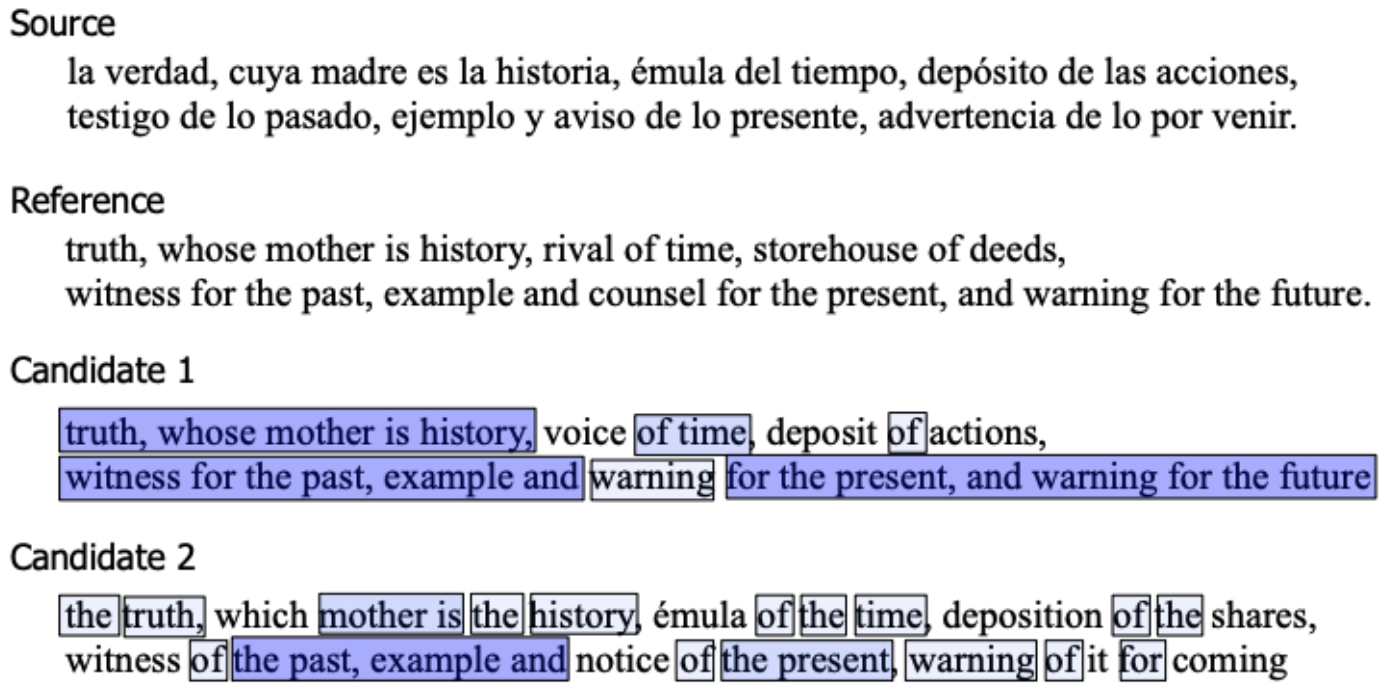
machine translation: training



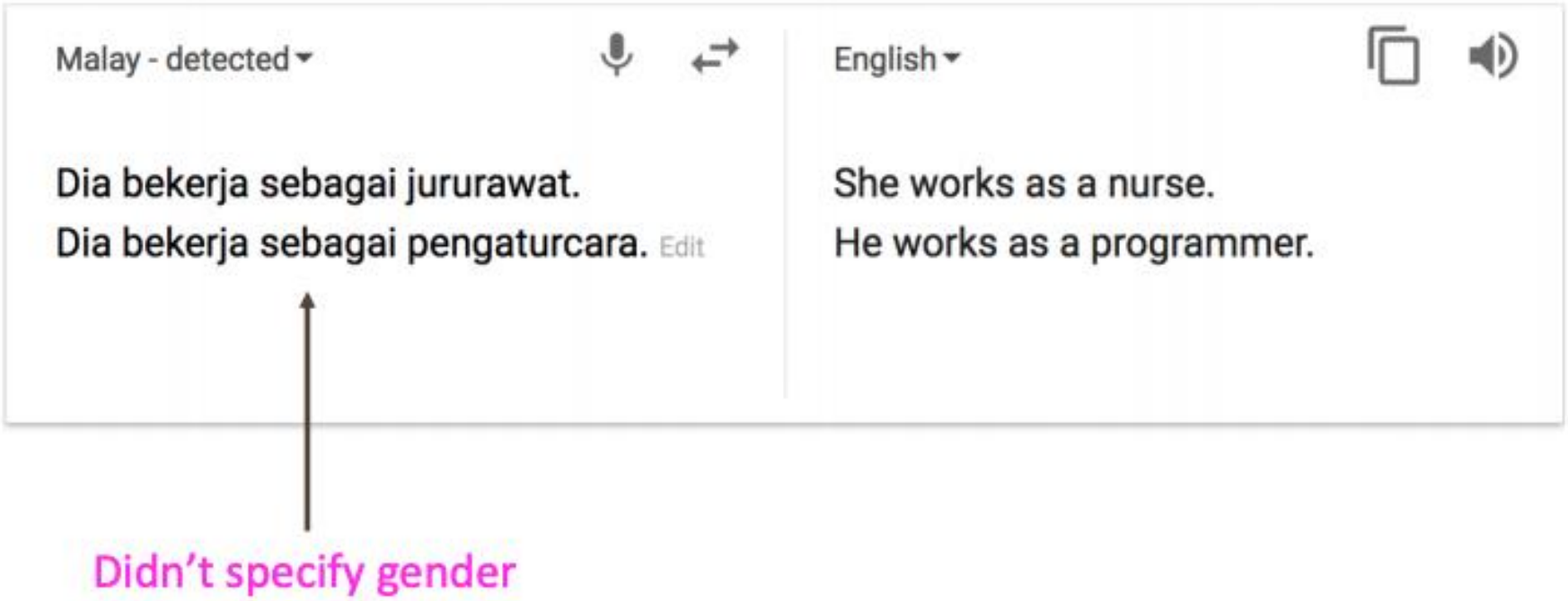
machine translation: evaluation

BLEU (bilingual evaluation understudy)

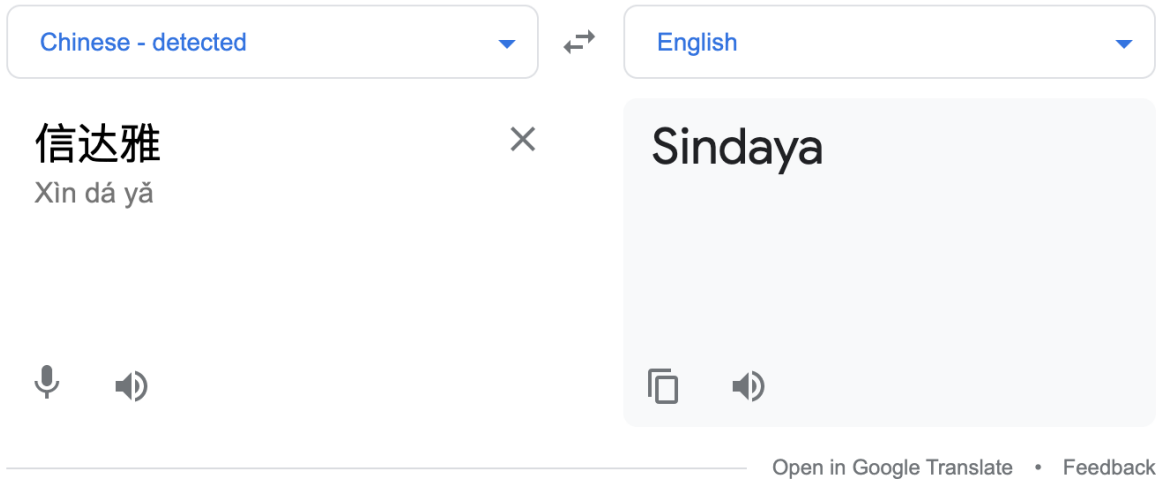
- compare the machine translation with human translation, and compute a similarity score
  - n-gram precision
  - penalty for too-short translation

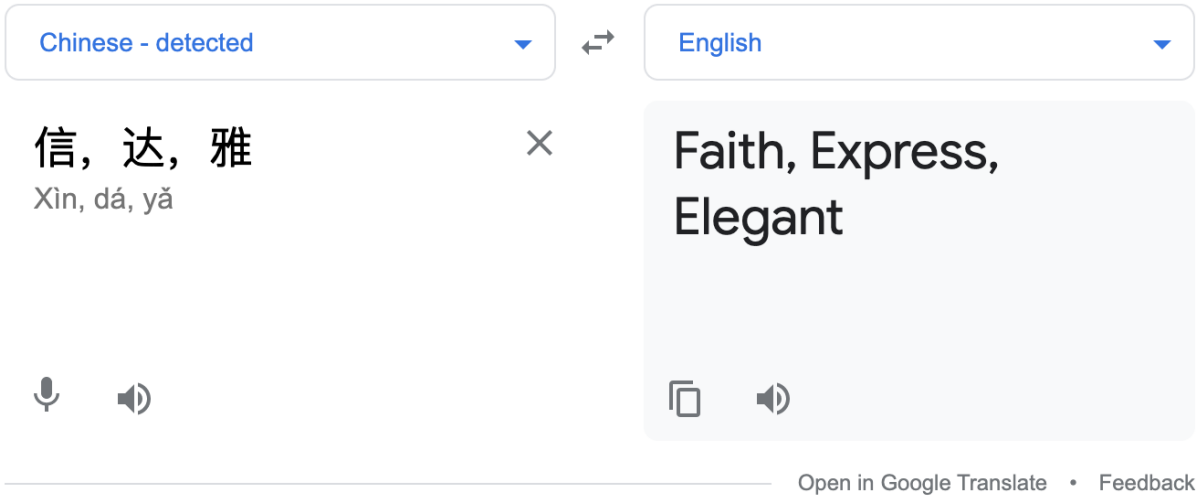


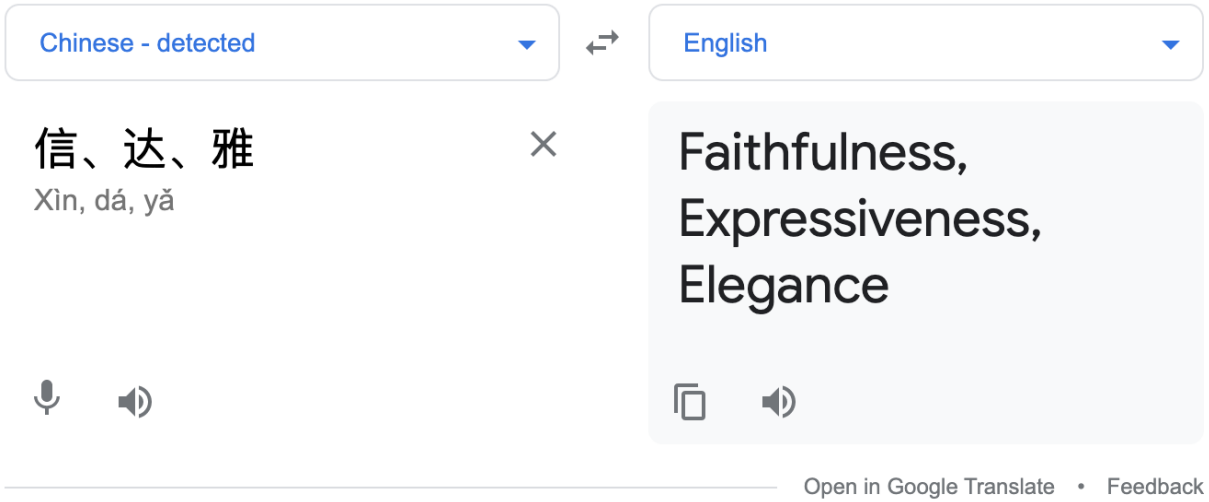
machine translation: bias



machine translation: bias

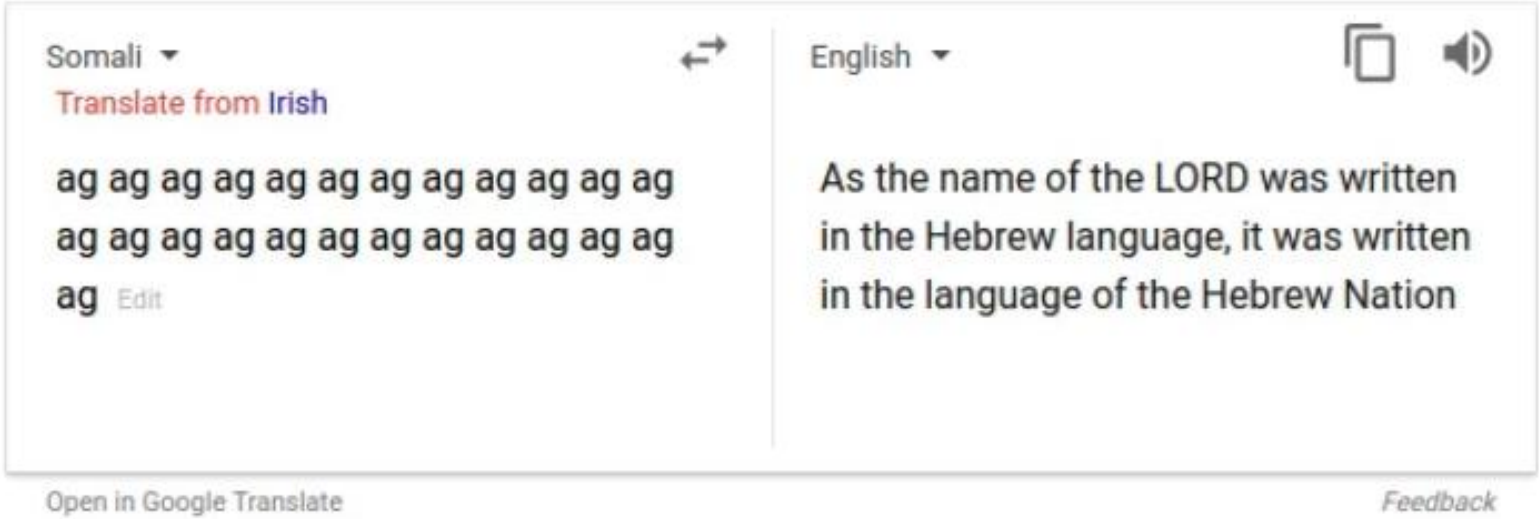






This example was taken in 2021. Google translate has been evolving and you may see a different result.

machine translation: non-interpretable



6.5 speech recognition and text to speech

Speech recognition and text to speech

speech recognition

It’s time for lunch!

text to speech

Ch 26, Jurafsky, D., & Martin, J. H. (2018). *Speech and language processing.*

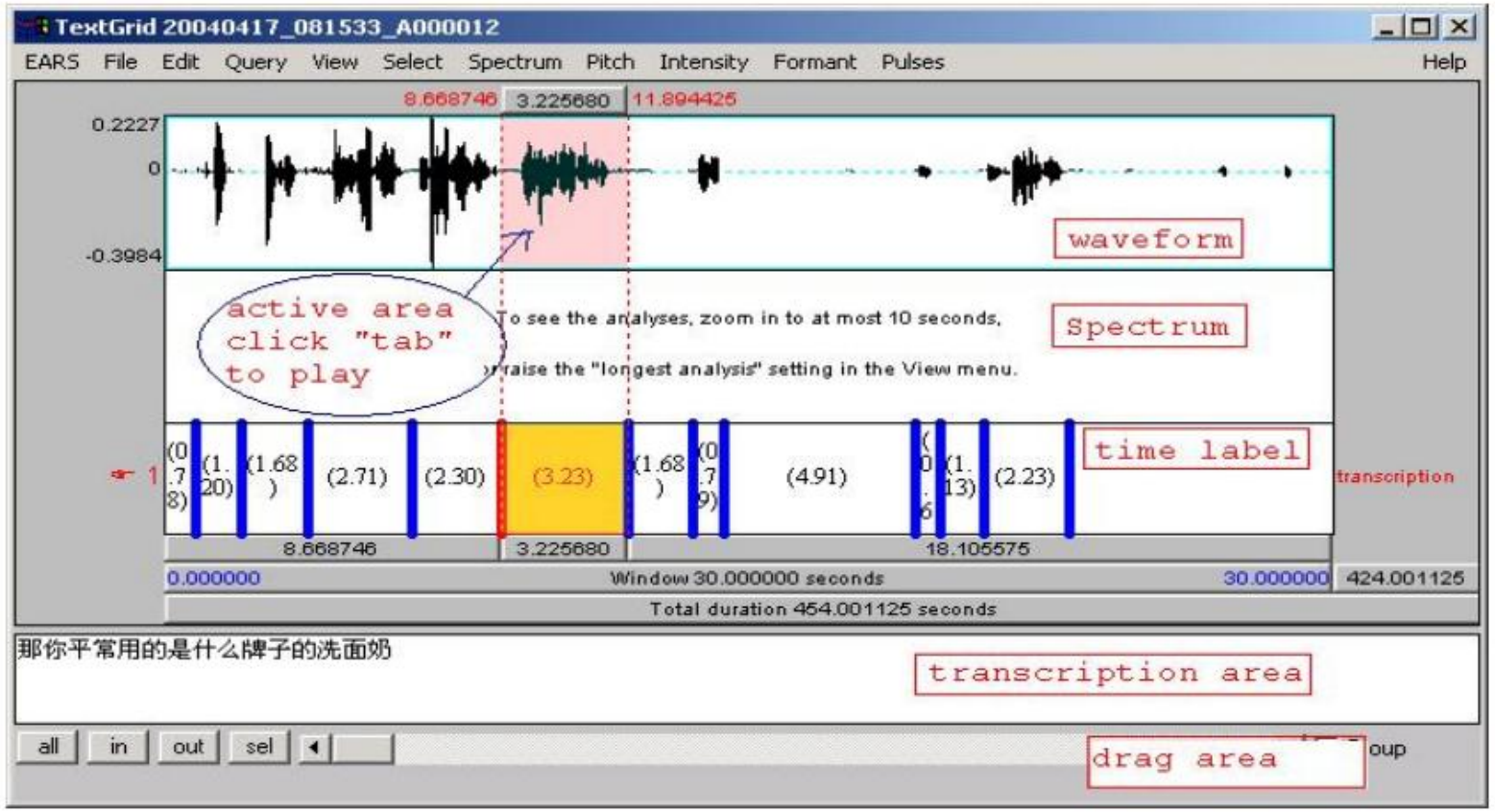
datasets

- **Open Speech and Language Resources** https://www.openslr.org/resources.php
  - LibriSpeech: audiobooks
  - Switchboard: telephone conversation
- CORAAL: African American speakers

datasets

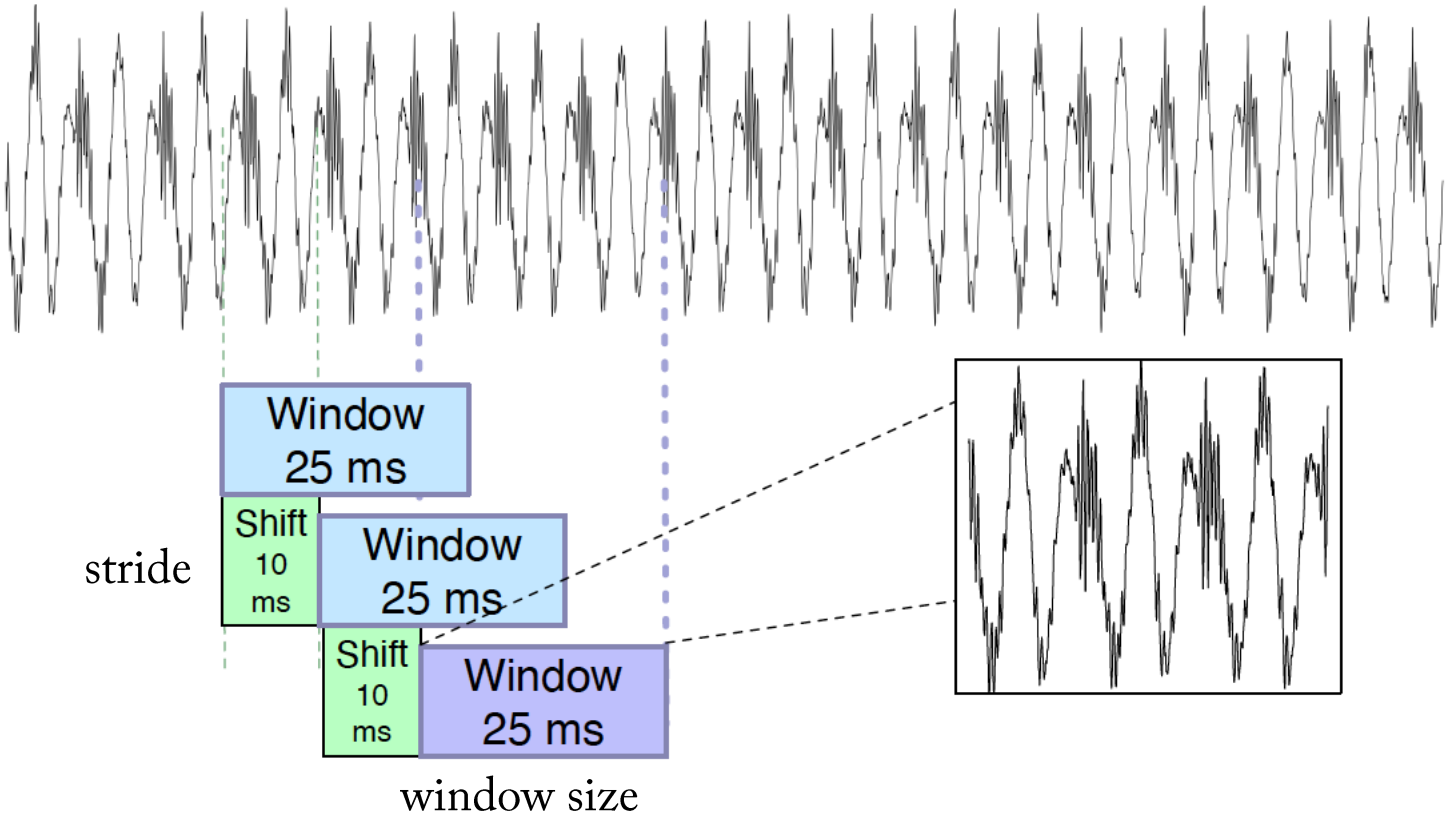
- HKUST Mandarin Telephone Speech

官话 Mandarin	闽 Min	湘 Xiang	赣 Gan	粤 Yue	客家 Kejia	吴 Wu
北东 N.E.	闽南 Minnan	土话 Tuhua	南昌 Nanchang	广州 G.Zh.	梅县 Meixian	太湖 Taihu
冀鲁 Jilu	蒲仙 Puxian	新湘语 N. X	鹰潭 Yingtan	五邑 Wuyi		台州 Taizhou
胶辽 Jiaoliao	闽东 Mindong	老湘语 O. X	抚州 Fuzhou			婺州 Wuzhou
北京 Beijing	闽北 Minbei		宜春 Yichun			处衢 Chuqu
中原 Central	闽中 Minzhong		吉安 Ji'an			瓯江 Oujiang
兰银 Lanyin	琼文 Qiongwen					宣州 Xuanzhou
西南 S.W.	邵将 Shaojiang					
江淮 Jianghuai						
桂柳片 Gui-Liu						

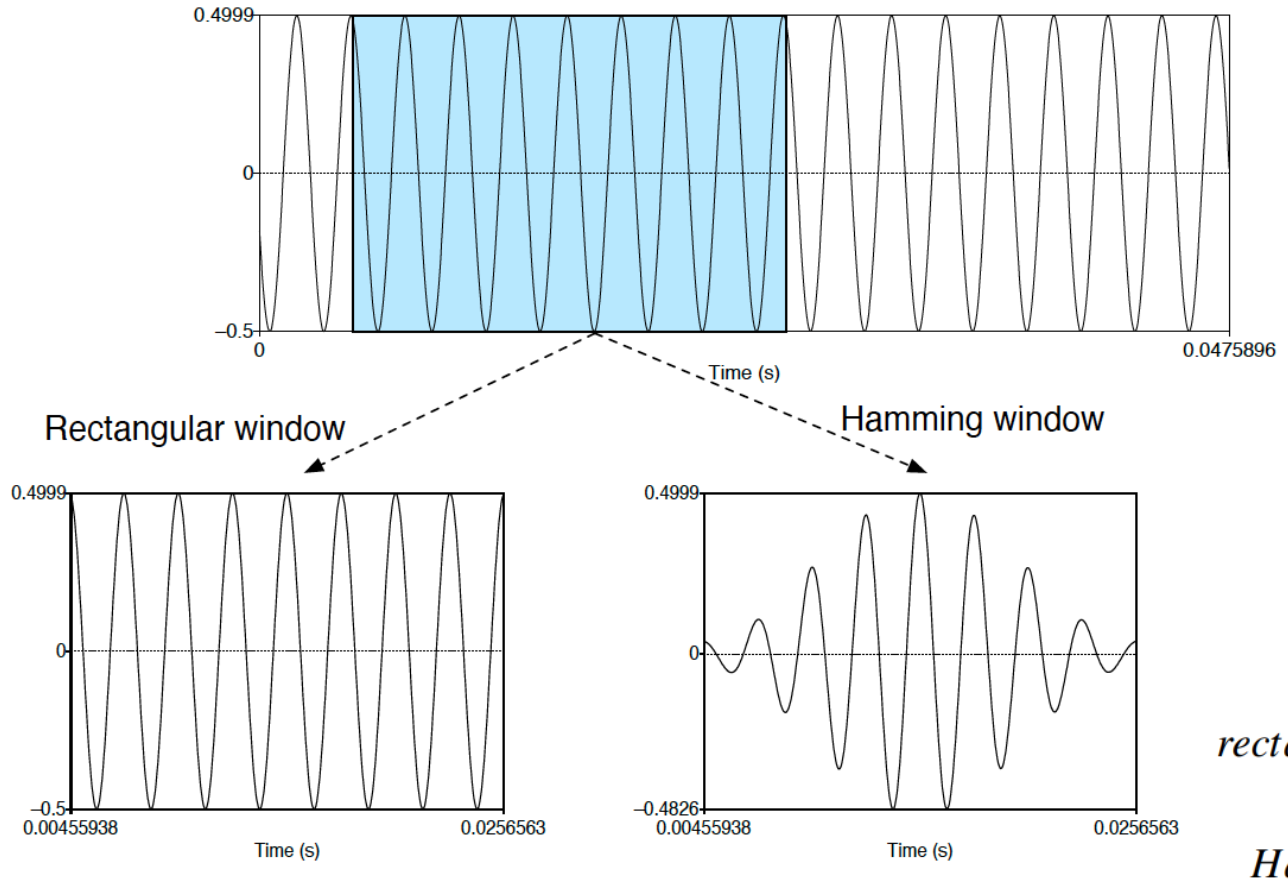


windowing

- need to extract features (frames) from small windows



windowing

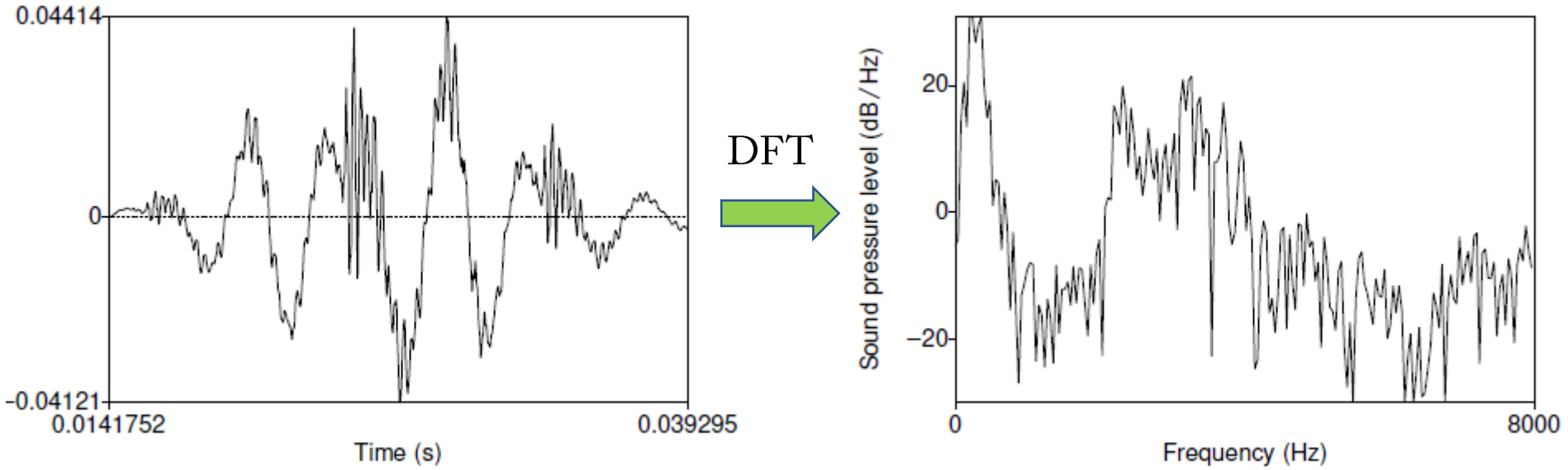


$$y[n] = w[n]\, s[n]$$

$$rectangular \qquad w[n] = \begin{cases} 1 & 0 \le n \le L-1 \\ 0 & \text{otherwise} \end{cases}$$

$$Hamming \qquad w[n] = \begin{cases} 0.54 - 0.46\cos\left(\frac{2\pi n}{L}\right) & 0 \le n \le L-1 \\ 0 & \text{otherwise} \end{cases}$$

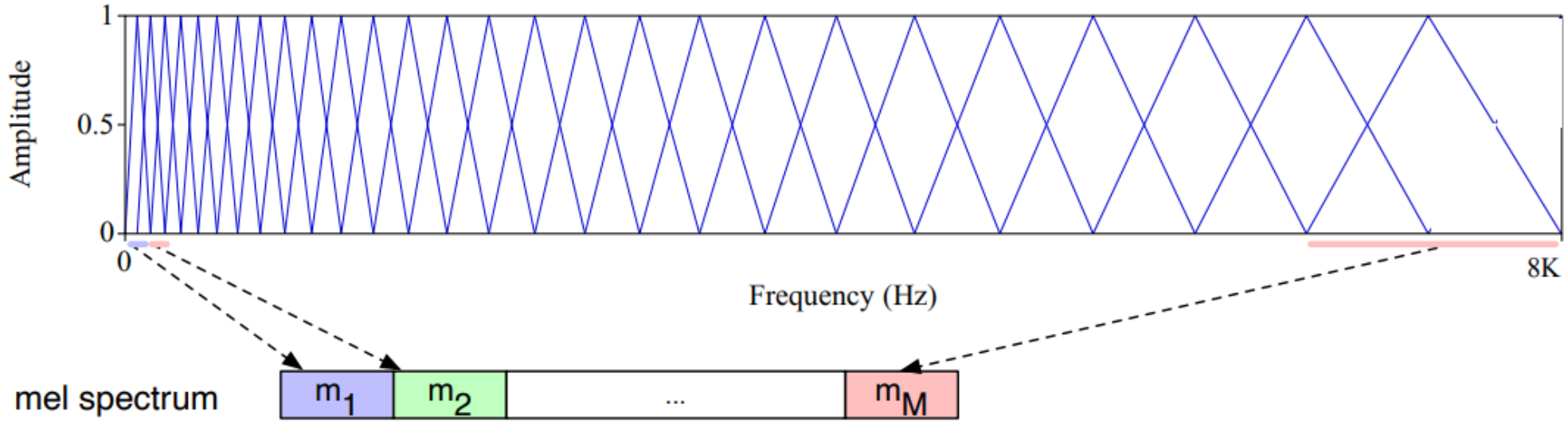
spectrum


0.04414
0
-0.04121
0.0141752
0.039295
Time (s)
DFT
Sound pressure level (dB/Hz)
20
0
-20
0
8000
Frequency (Hz)

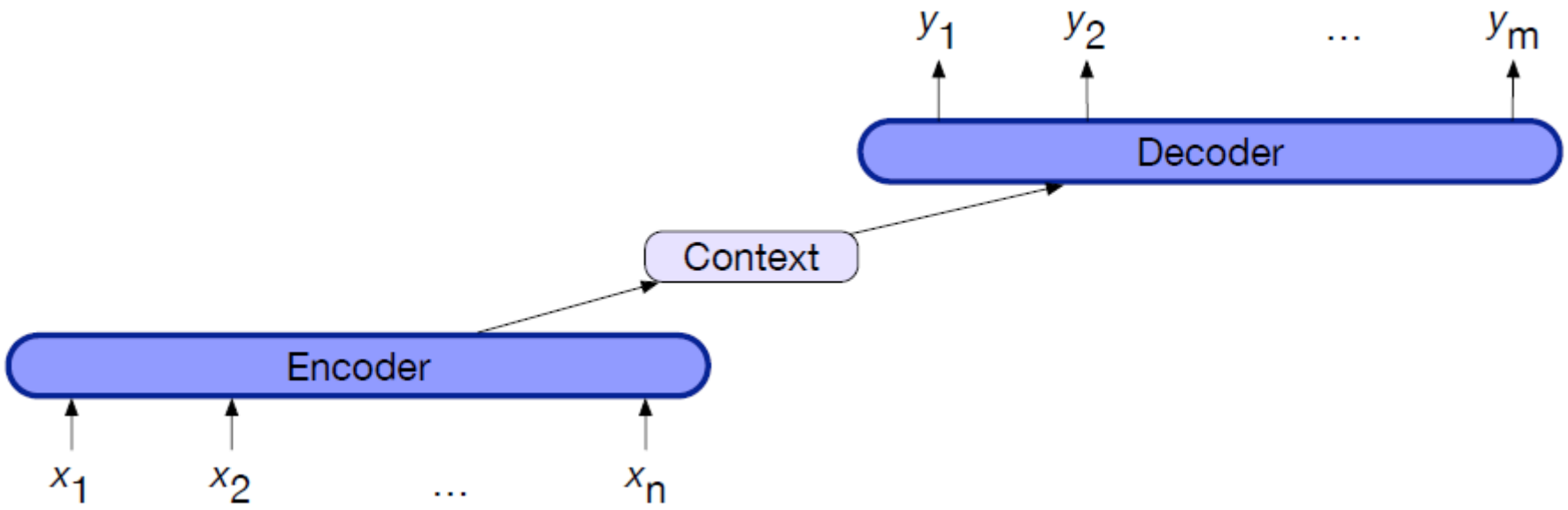
spectrum

- mel scale

$$\text{mel}(f) = 1127 \ln\left(1 + \frac{f}{700}\right)$$



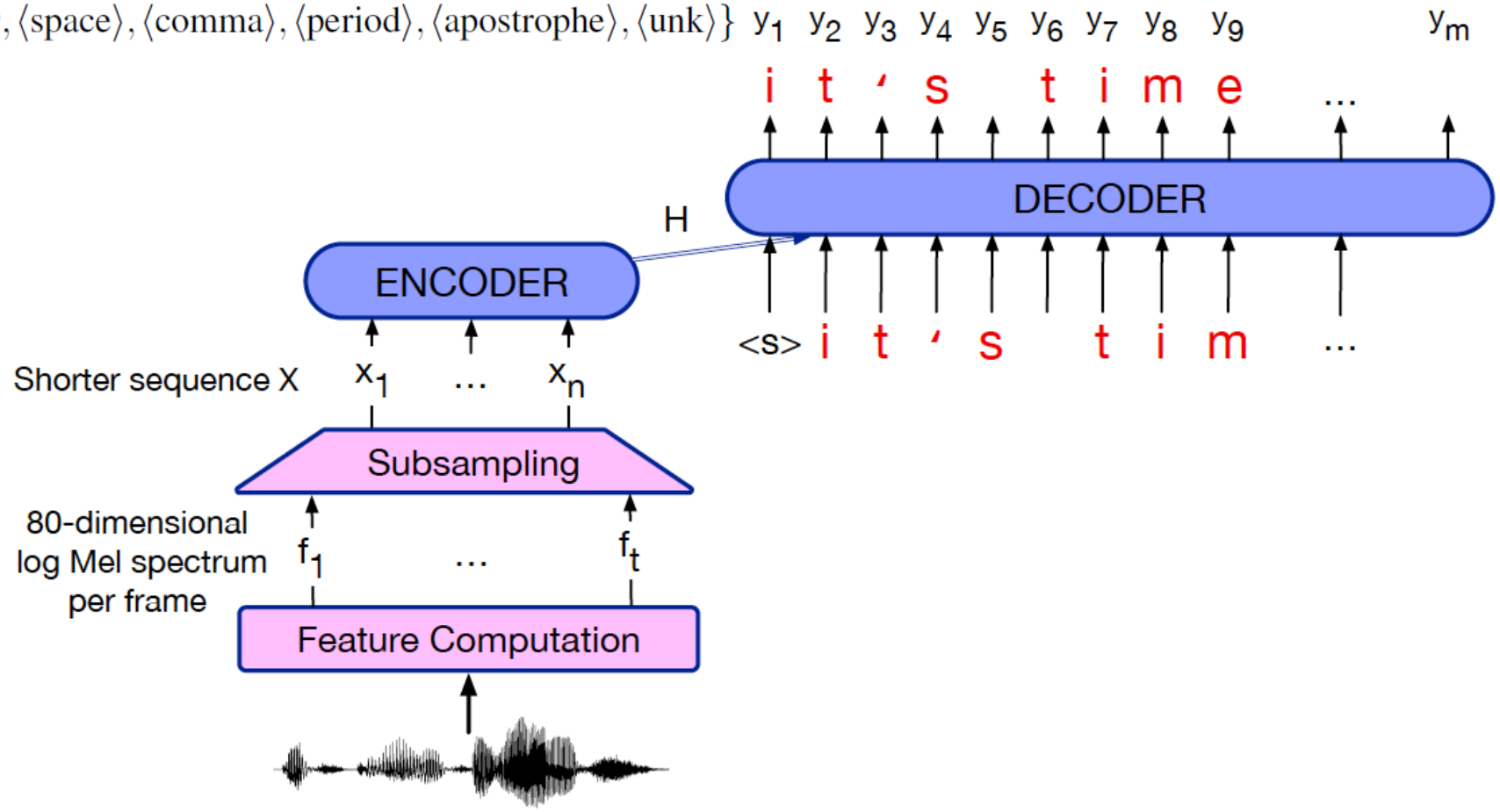
encoder-decoder RNN



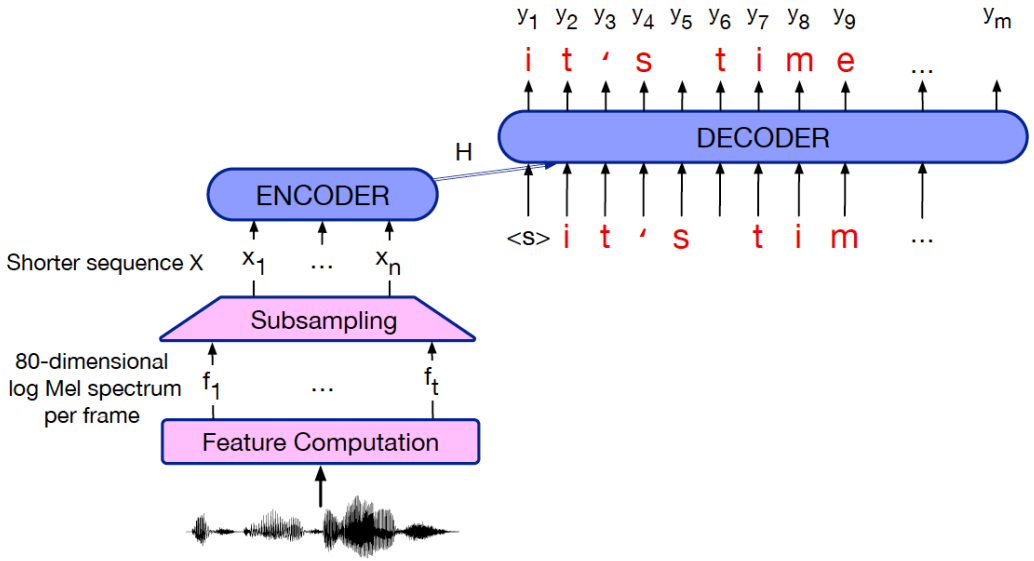
encoder-decoder speech recognizer

$$Y = (\langle \text{SOS} \rangle, y_1, \ldots, y_m \langle \text{EOS} \rangle)$$

$y_i \in \{a, b, c, \ldots, z, 0, \ldots, 9, \langle \text{space} \rangle, \langle \text{comma} \rangle, \langle \text{period} \rangle, \langle \text{apostrophe} \rangle, \langle \text{unk} \rangle\}$



training



- inference

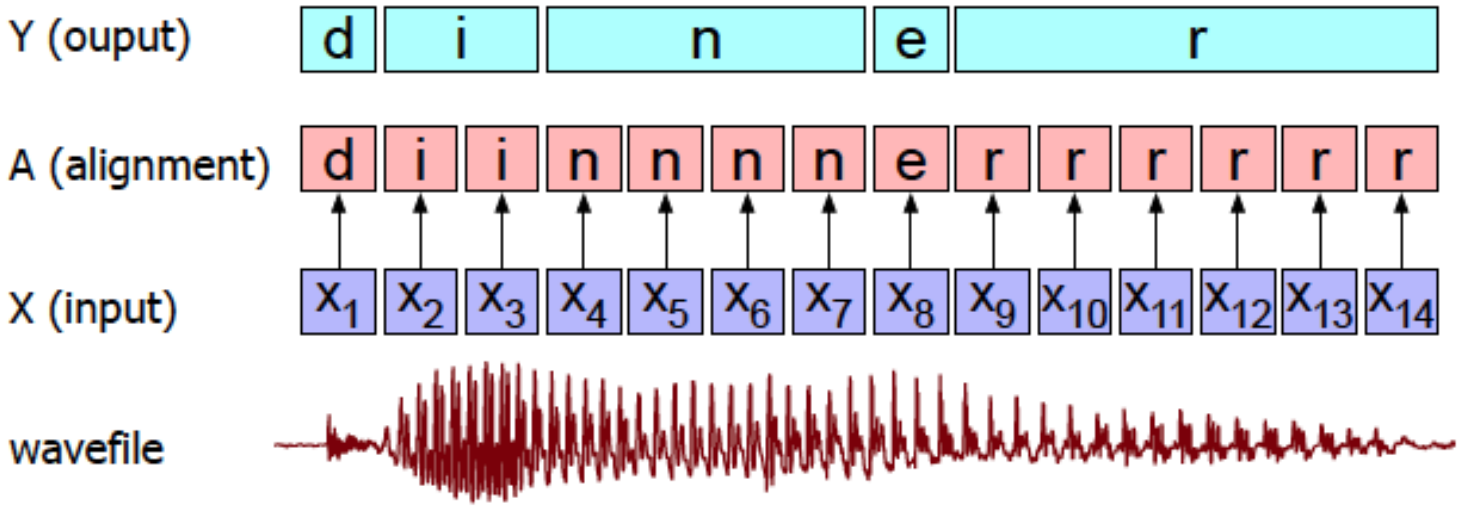
$$p(y_1,\ldots,y_n)=\prod_{i=1}^{n}p(y_i|y_1,\ldots,y_{i-1},X)$$

$$\hat{y}_i \;=\; \text{argmax}_{\text{char}\in\text{ Alphabet}}P(\text{char}|y_1\ldots y_{i-1},X)$$

- learning: cross-entropy loss
- insufficient data: use a large language model for improved training, e.g.,

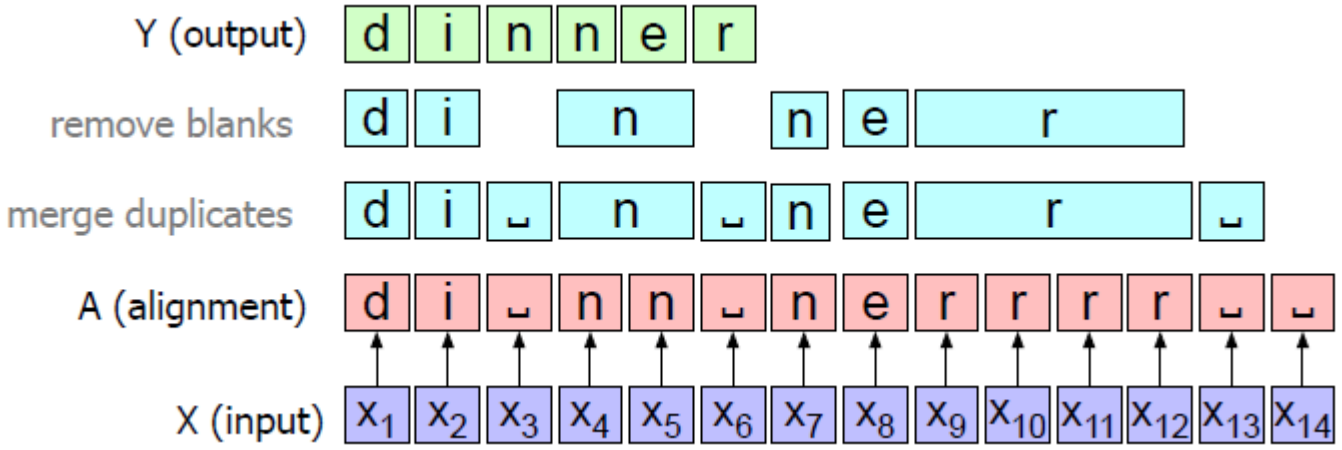
$$score(Y|X)=\frac{1}{|Y|_c}\log P(Y|X)+\lambda\log P_{LM}(Y)$$

CTC speech recognizer



Connectionist Temporal Classification: an alternative to encoder-decoder

CTC speech recognizer



non-one-to-one-ness

d i i n ␣ n n e e e r r r ␣

d d i n n ␣ n e r r ␣ ␣ ␣ ␣

d d d i n ␣ n n ␣ ␣ ␣ e r r

evaluation

$$\text{Word Error Rate} = 100 \times \frac{\text{Insertions} + \text{Substitutions} + \text{Deletions}}{\text{Total Words in Correct Transcript}}$$

REF:	i	***	**	UM	the	PHONE	IS		i	LEFT	THE	portable	****	PHONE	UPSTAIRS	last	night
HYP:	i	GOT	IT	TO	the	*****	FULLEST		i	LOVE	TO	portable	FORM	OF	STORES	last	night
Eval:		I	I	S		D	S			S	S		I	S	S		

evaluation

English Tasks	**WER%**
LibriSpeech audiobooks 960hour clean	1.4
LibriSpeech audiobooks 960hour other	2.6
Switchboard telephone conversations between strangers	5.8
CALLHOME telephone conversations between family	11.0
Sociolinguistic interviews, CORAAL (AAL)	27.0
CHiMe5 dinner parties with body-worn microphones	47.9
CHiMe5 dinner parties with distant microphones	81.3
Chinese (Mandarin) Tasks	**CER%**
AISHELL-1 Mandarin read speech corpus	6.7
HKUST Mandarin Chinese telephone conversations	23.5

TTS text-to-speech

- spectrum prediction

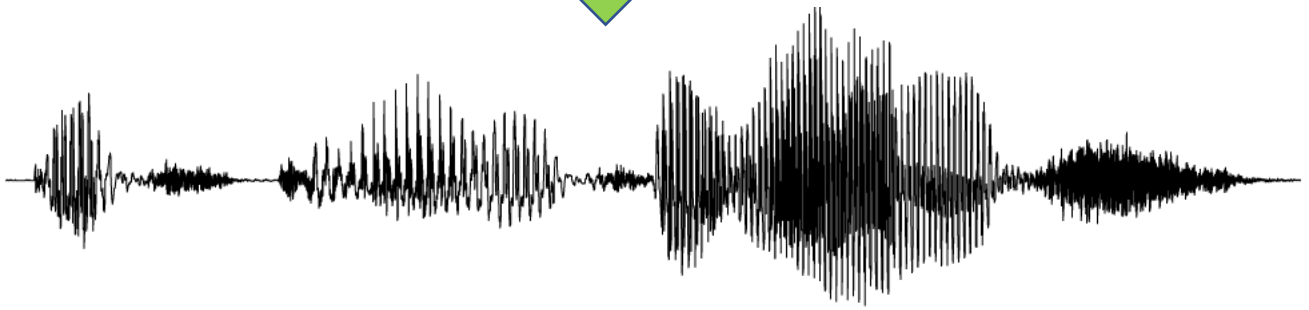
It's time for lunch!

TTS text-to-speech

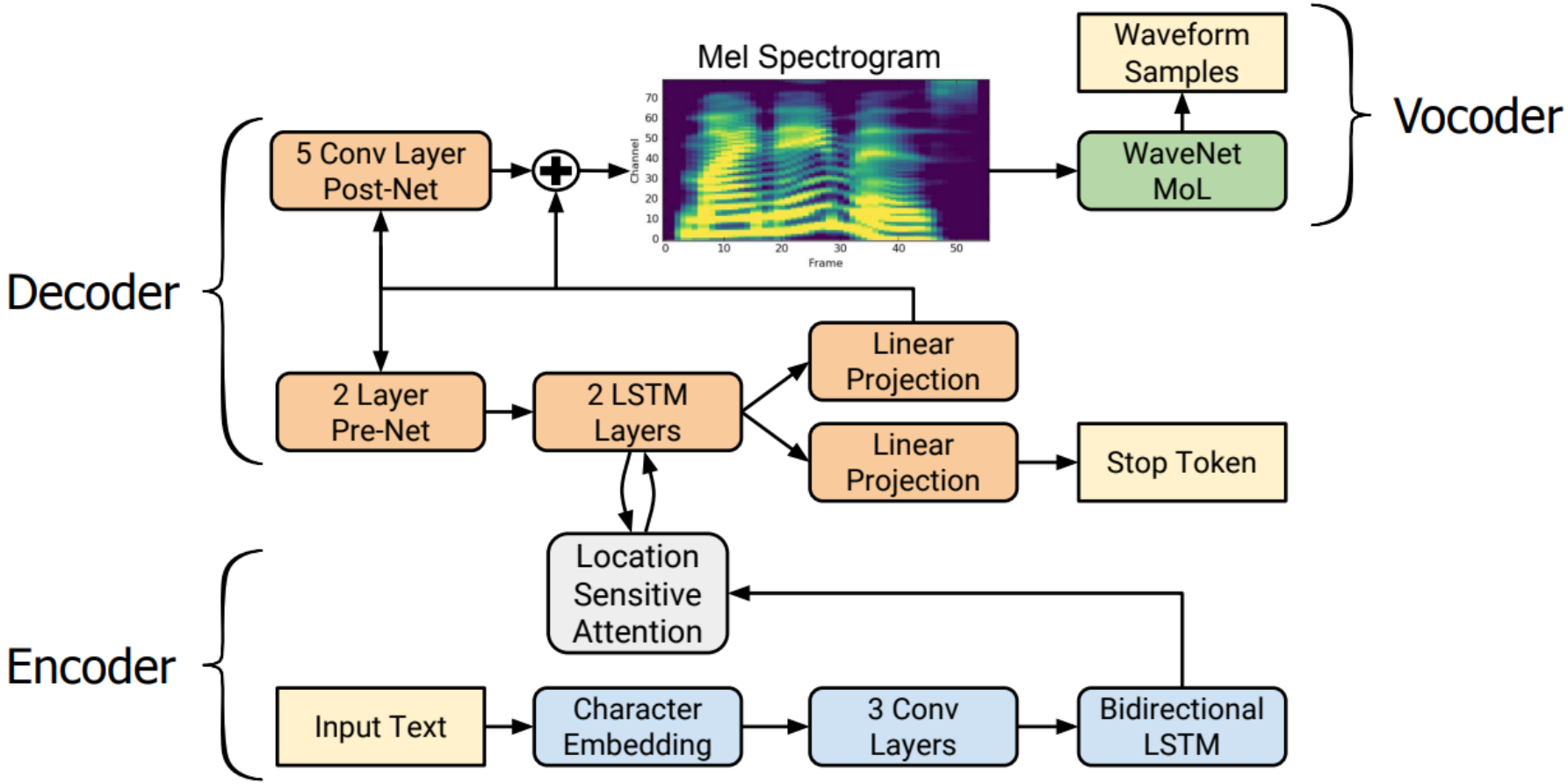
It's time for lunch!

encoder-
decoder

vocoder



Tacotron2



visit https://pytorch.org/hub/nvidia_deeplearningexamples_tacotron2/
or play here

Thank you!

Reference

- Ch 10, *Deep Learning*.
- Pascanu, R., Mikolov, T., & Bengio, Y. (2013, May). On the difficulty of training recurrent neural networks. In ICML (pp. 1310-1318).
- Ch 8, Jurafsky, D., & Martin, J. H. (2018). *Speech and language processing (draft)*. https://web. stanford. edu/~ jurafsky/slp3.
- Goldberg, Y. (2016). A primer on neural network models for natural language processing. *Journal of Artificial Intelligence Research*, 57, 345-420.

Thank you!

Reference

- Ch 11 & 26, Jurafsky, D., & Martin, J. H. (2018). *Speech and language processing (draft)*. https://web. stanford. edu/~ jurafsky/slp3.
- Goldberg, Y. (2016). A primer on neural network models for natural language processing. *Journal of Artificial Intelligence Research*, 57, 345-420.