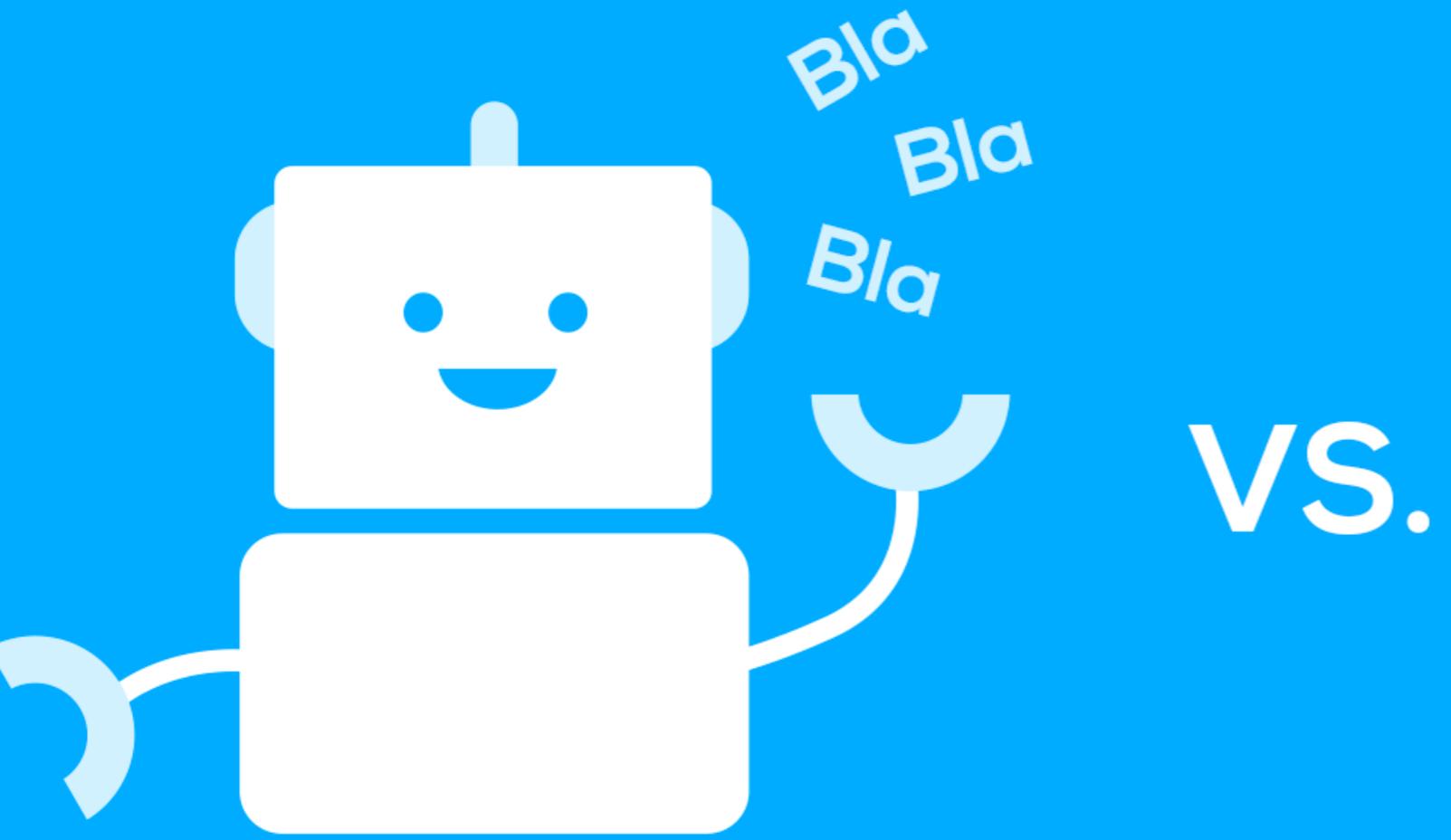


Key Issues in Building Dialog System and Chatbot

Head First Theory and Practice

Yanran Li
The Hong Kong Polytechnic University



vs.



Mainstream Approaches

Two Paradigms

Two Paradigms

M: Hello, welcome to the Cambridge dialogue system, What kind of food would you like?
H: Yeah I want to find a restaurant that serves European food.
M: Hotel du Vin and Bistro is a nice place. It serves European food.
H: Uh what is the address and phone number?
M: Hotel du Vin and Bistro is a nice place. Their address is XXX and their phone is YYY.
H: Thank you goodbye.
M: Thank you for using this system.

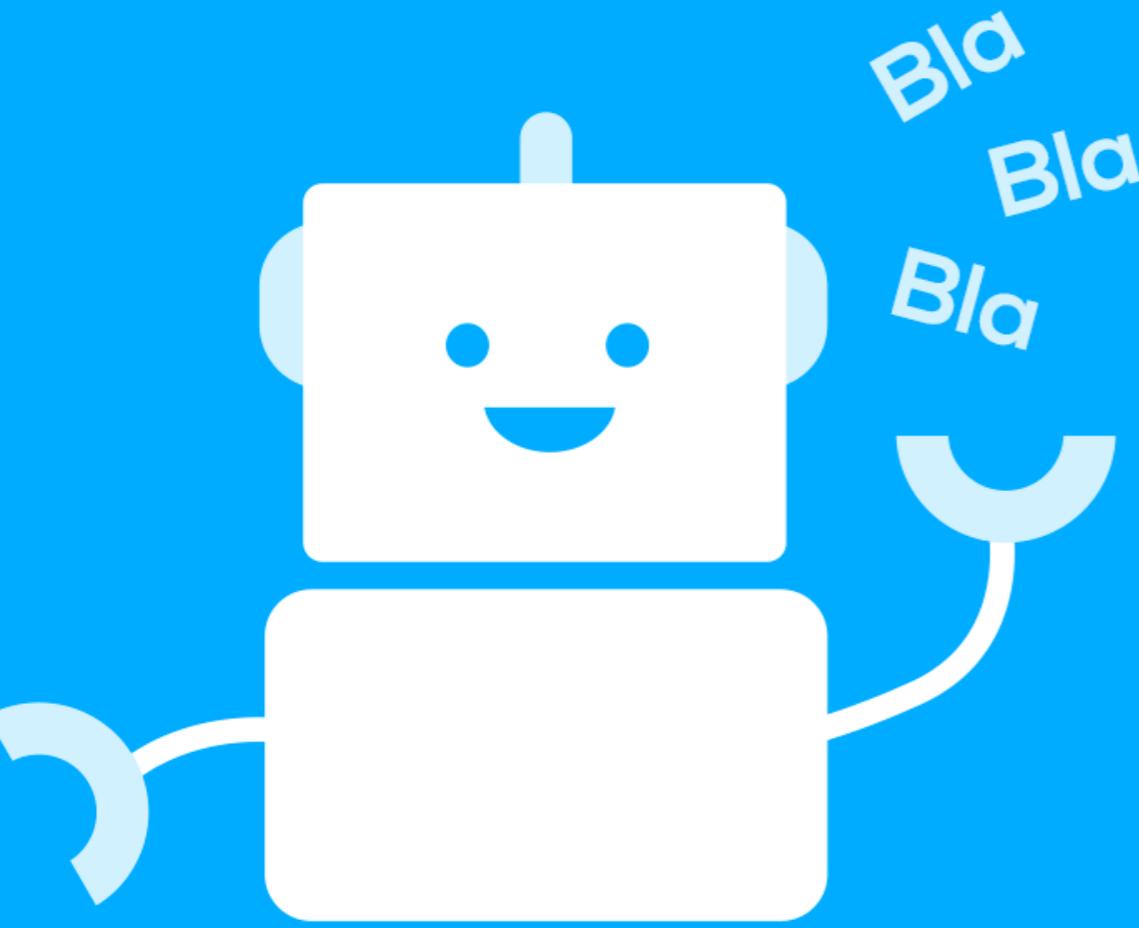
Task-oriented Domain-specific Dialog System

A: How old are you ?
B: I'm 16. Why are you asking ?
A: I thought you were 12.
B: What made you think so ?
A: I don't know what you are talking about.
B: You don't know what you are saying.

Chit-chat Open Domain Conversational Agent

Two Paradigms

- Task-oriented Dialog System
 - Task (*Goal)-oriented, complete a task, e.g. book a hotel
 - Require precise understanding, hard to collect data
 - Modular, highly handcrafted (API), restricted ability, easy to evaluate
- Chit-chat Conversational Agent
 - Chit-chat (non-*goal), auxiliary topics
 - Vast amount of data (but probably not helpful), e.g. movie lines, Weibo posts
 - End-to-end, highly data-driven, hard to evaluate



vs.

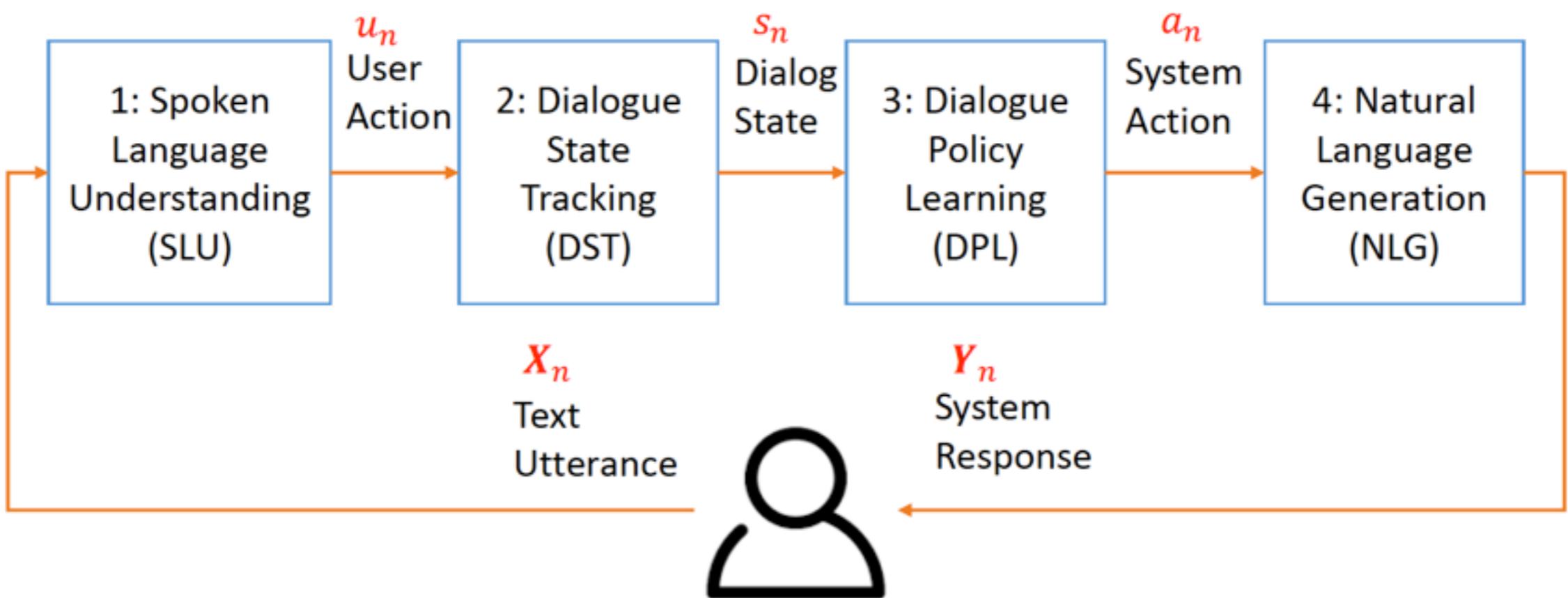


Task-oriented

Dialog System

Task-oriented Dialog System

- Modular: often 4 modules

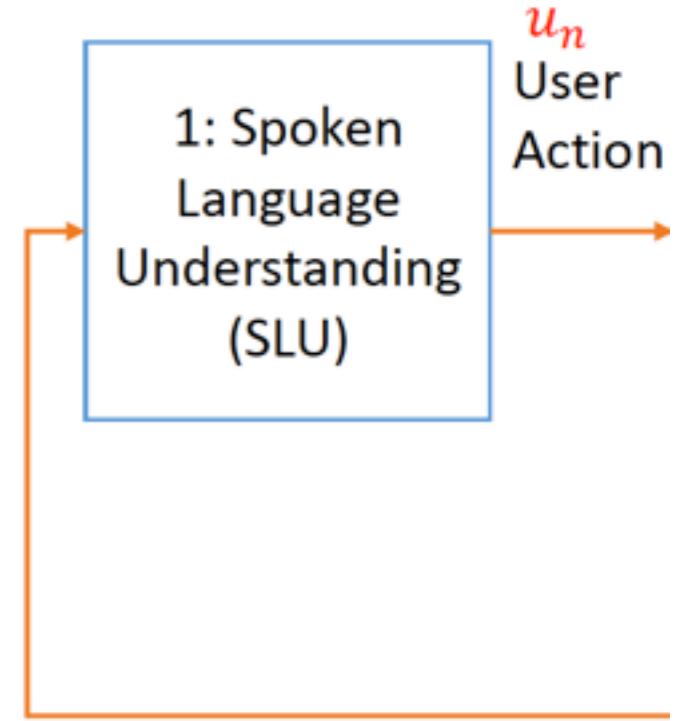


Task-oriented Dialog System

- Modular: often 4 modules
- Sub-modules, sub-tasks:
 - Spoken Language Understanding (SLU)
 - Dialogue State Tracking (DST)
 - Dialogue Policy Learning (DPL)
 - Natural Language Generation (NLG)

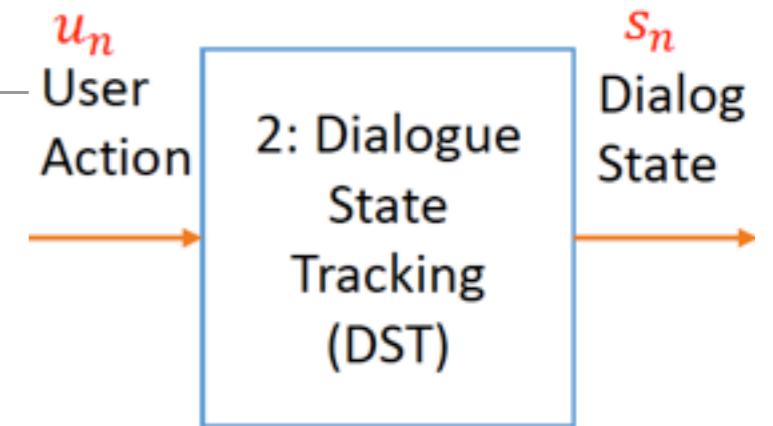
Task-oriented Dialog System

- Modular: often 4 modules
- Sub-modules, sub-tasks:
 - Spoken Language Understanding (SLU)
 - SLU turns natural language into user intention and slot-values, and it takes input and outputs structured user action
 - Dialogue State Tracking (DST)
 - Dialogue Policy Learning (DPL)
 - Natural Language Generation (NLG)



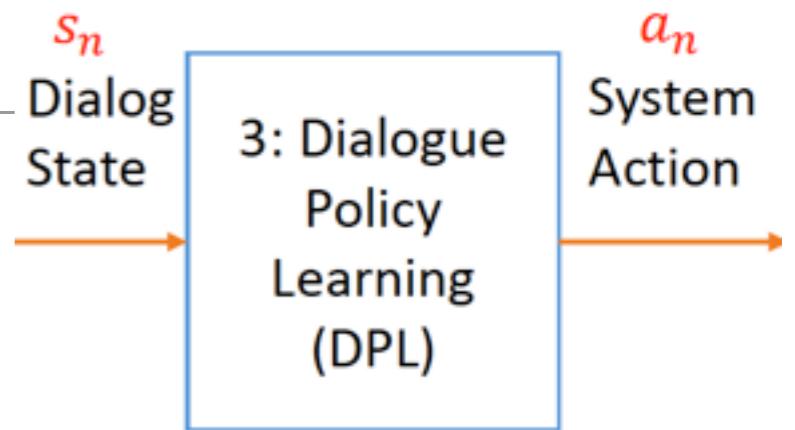
Task-oriented Dialog System

- Modular: often 4 modules
- Sub-modules, sub-tasks:
 - Spoken Language Understanding (SLU)
 - Dialogue State Tracking (DST)
 - DST tracks the current dialogue state, and outputs dialogue state
 - Dialogue Policy Learning (DPL)
 - Natural Language Generation (NLG)



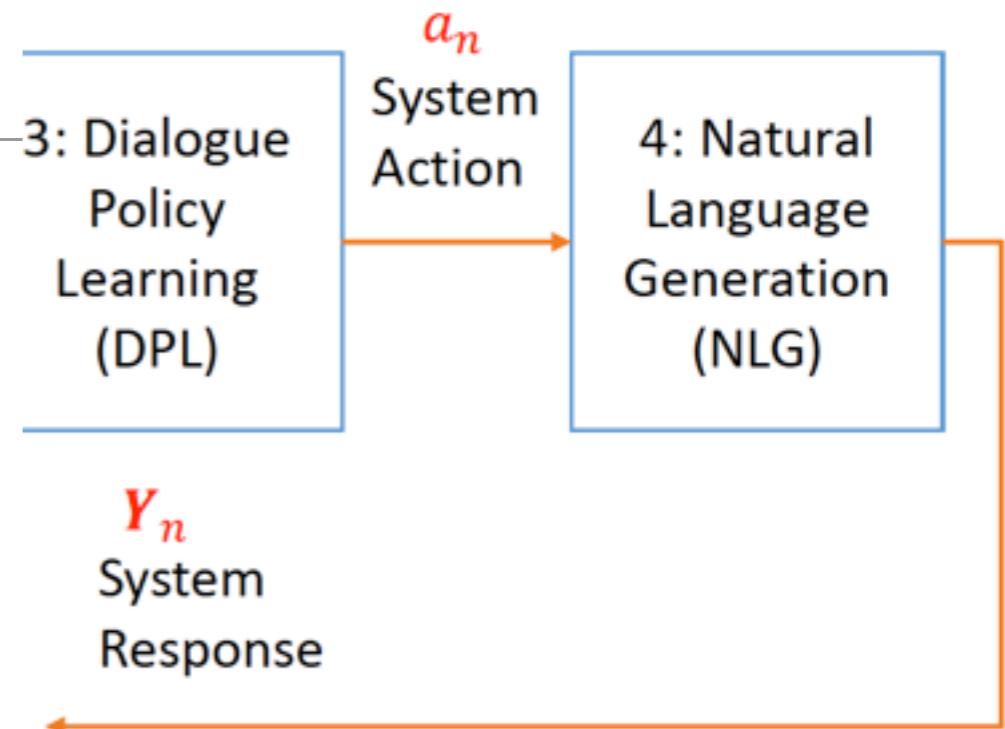
Task-oriented Dialog System

- Modular: often 4 modules
- Sub-modules, sub-tasks:
 - Spoken Language Understanding (SLU)
 - Dialogue State Tracking (DST)
 - Dialogue Policy Learning (DPL)
 - Policy decides which system action to take based on the dialogue state, and it takes dialogue state as input and outputs system action
 - Natural Language Generation (NLG)



Task-oriented Dialog System

- Modular: often 4 modules
- Sub-modules, sub-tasks:
 - Spoken Language Understanding (SLU)
 - Dialogue State Tracking (DST)
 - Dialogue Policy Learning (DPL)
 - Natural Language Generation (NLG)
 - NLG turns a system action into natural language, and it takes the system action as input and outputs the system response



Task-oriented Neural Dialog System [7]

Can I have Korean

Little Seoul serves great Korean .

Task-oriented Neural Dialog System

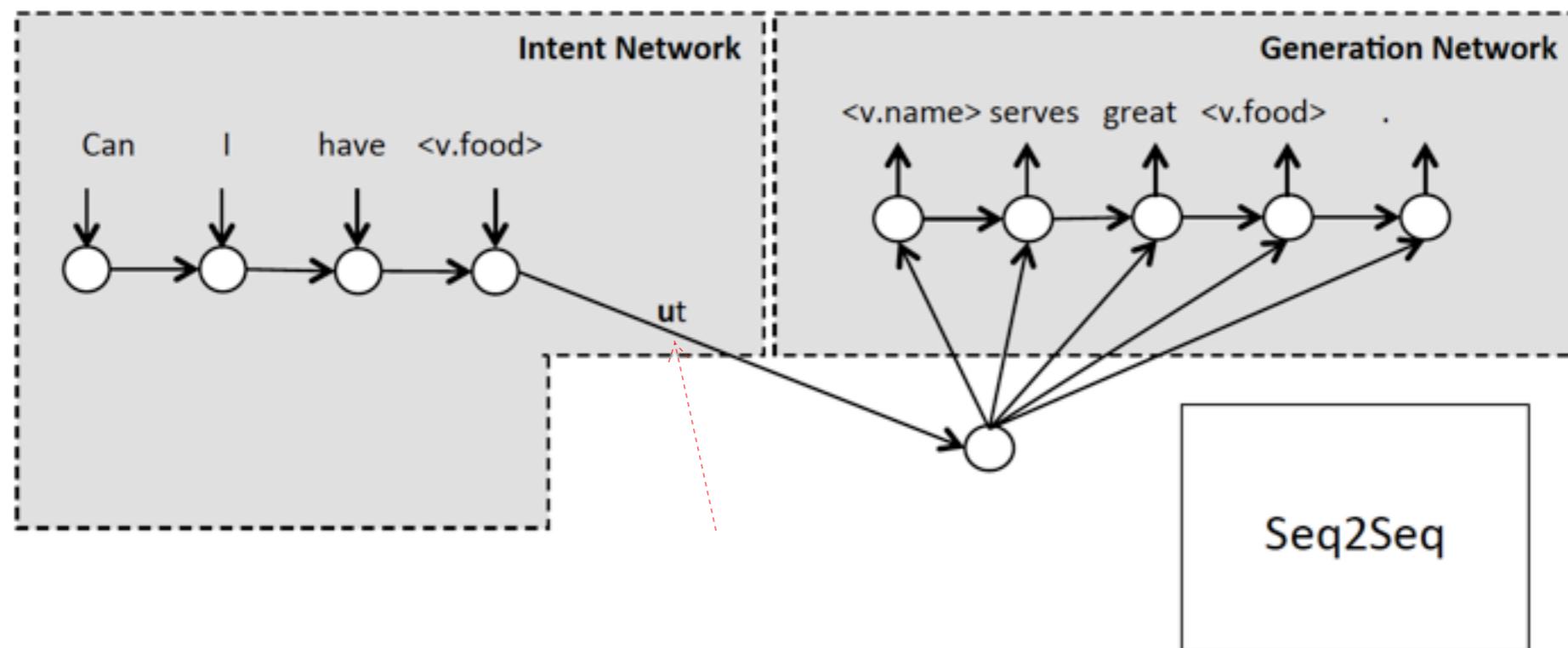
Can I have <v.food>

<v.name> serves great <v.food> .

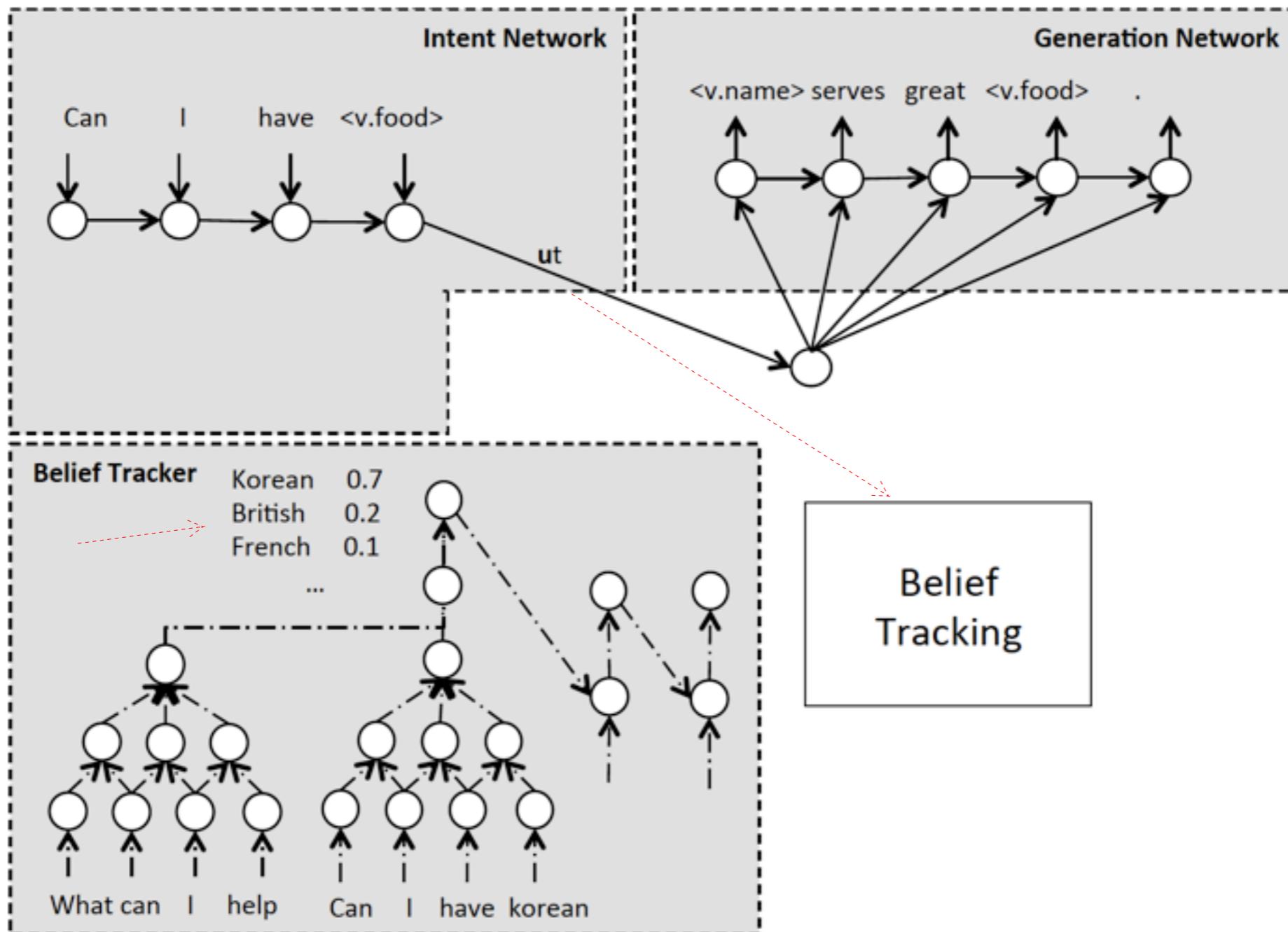
Delexicalisation



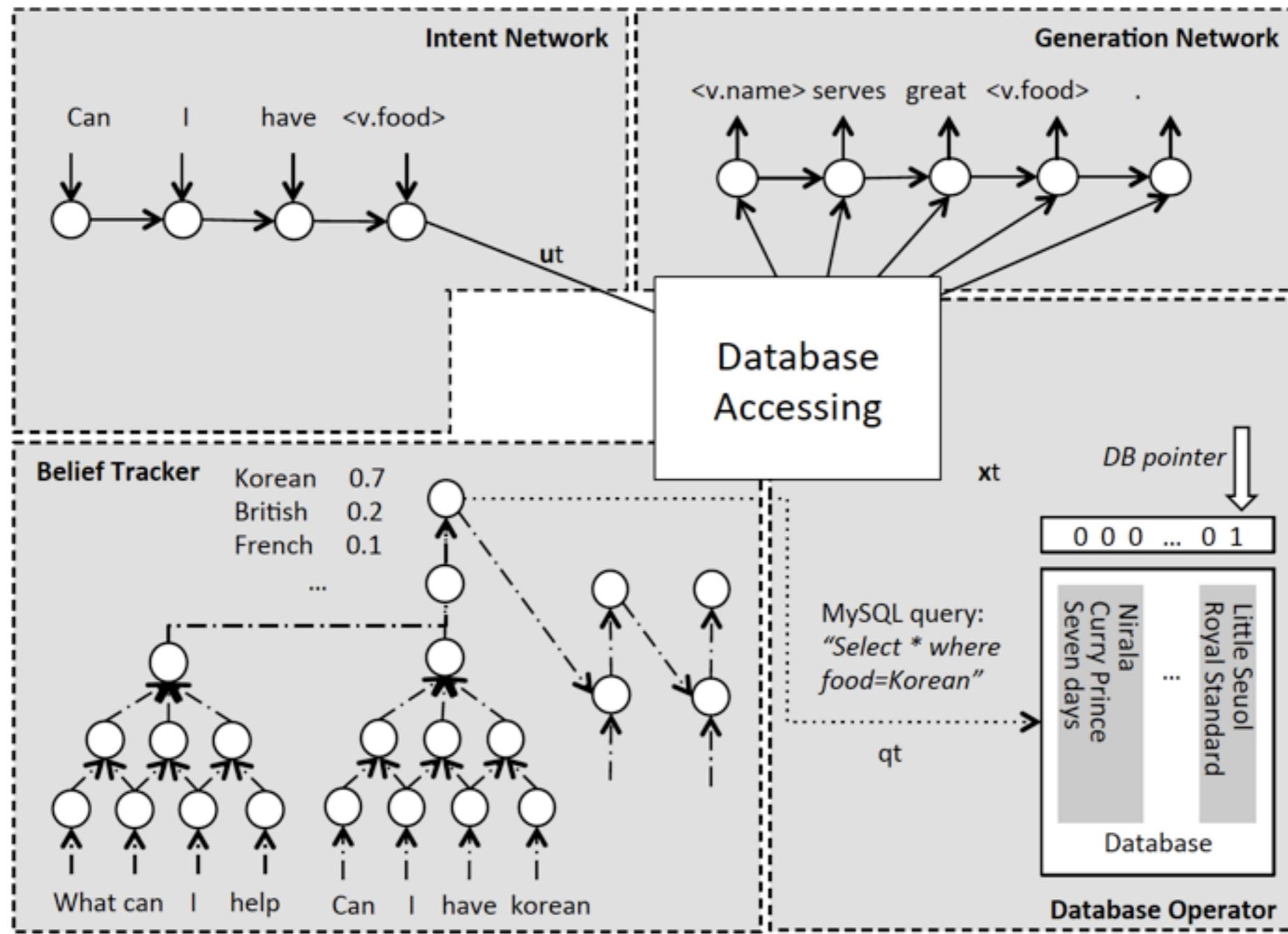
Task-oriented Neural Dialog System



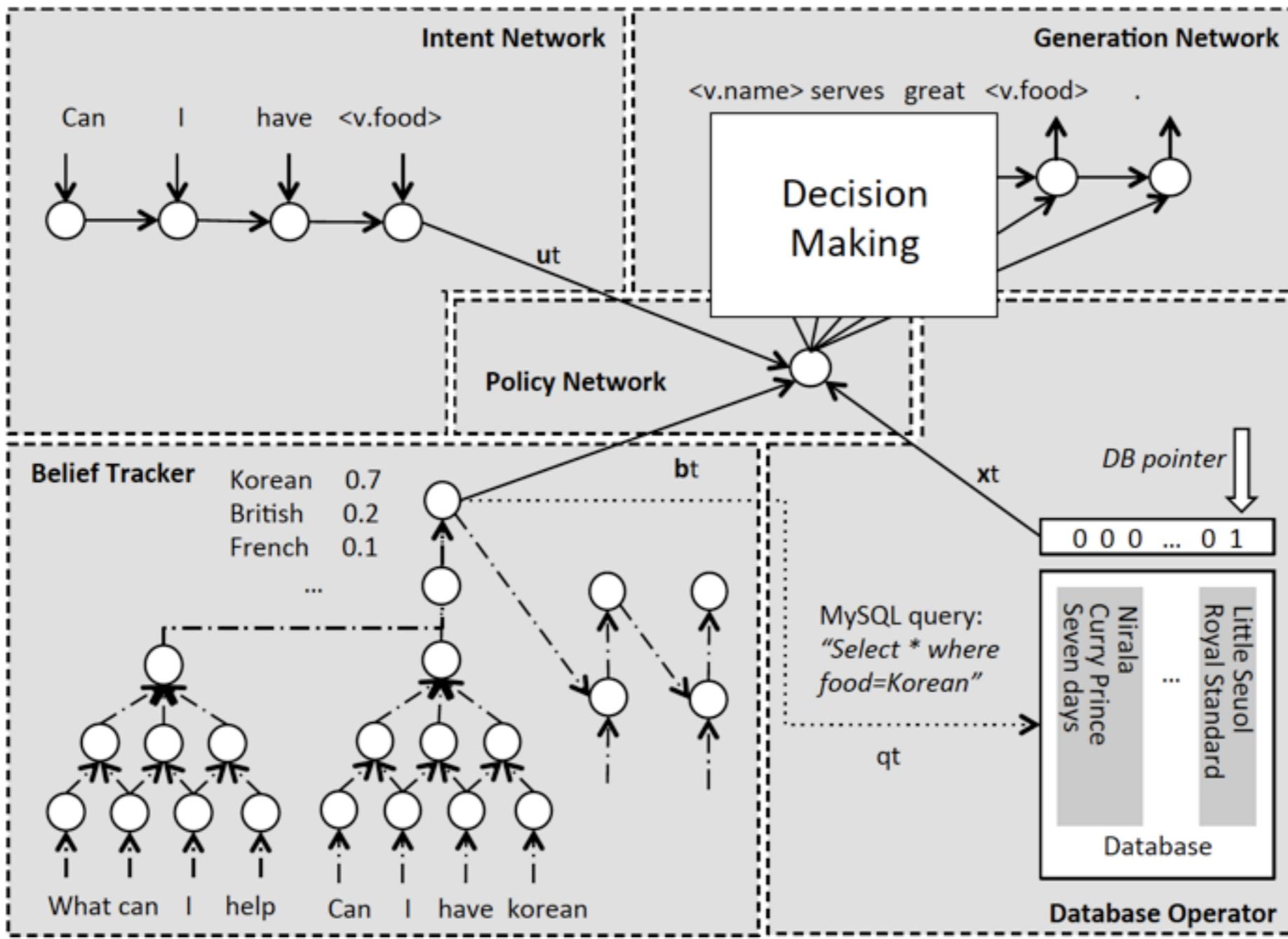
Task-oriented Neural Dialog System



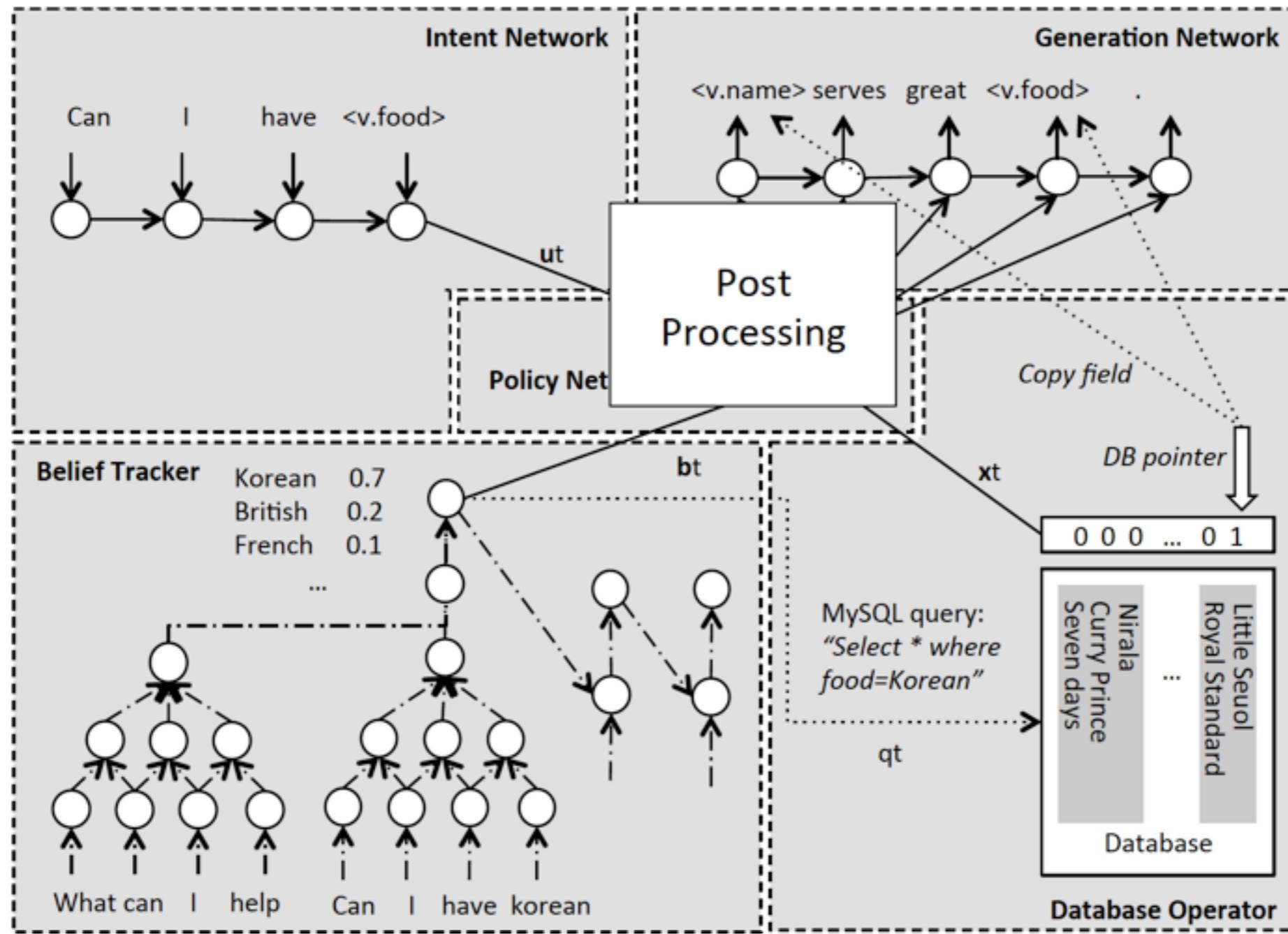
Task-oriented Neural Dialog System

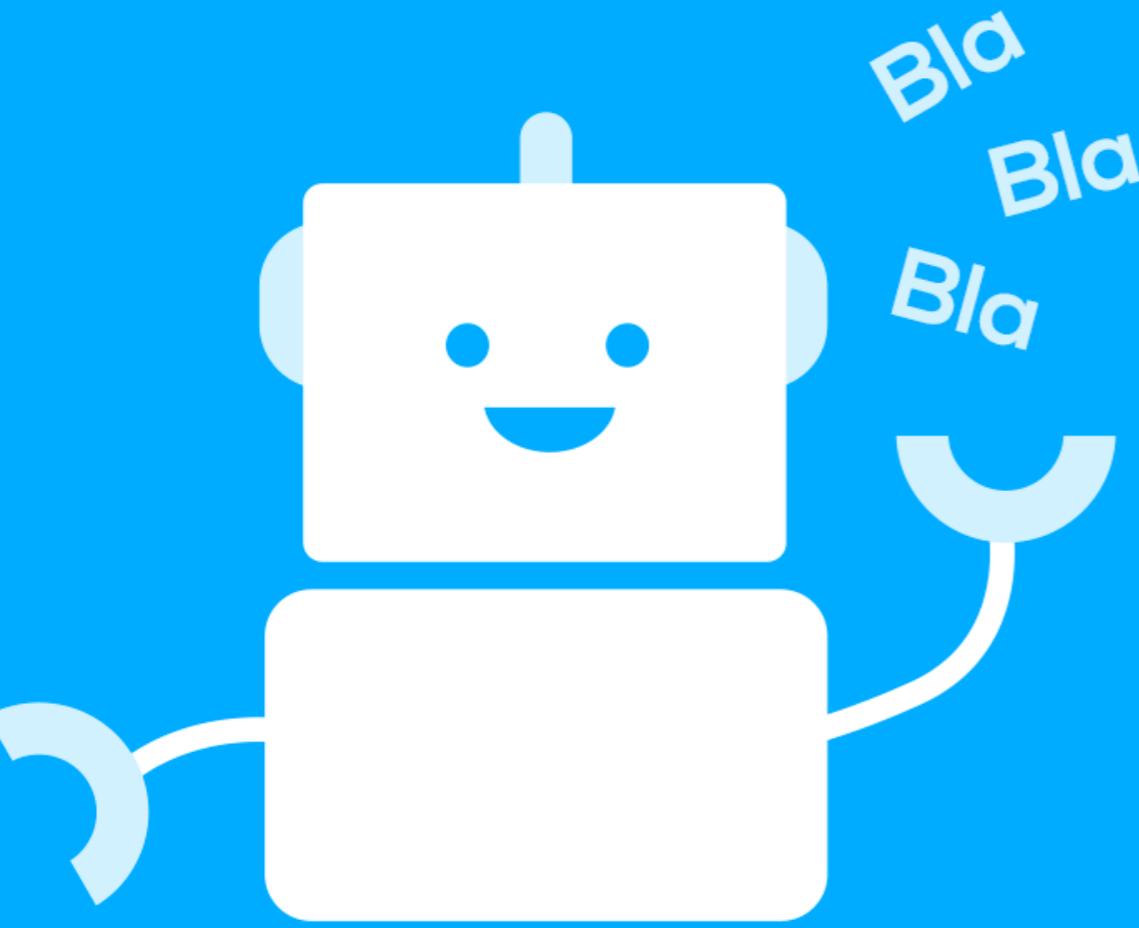


Task-oriented Neural Dialog System



Task-oriented Neural Dialog System





vs.



Chit-chat

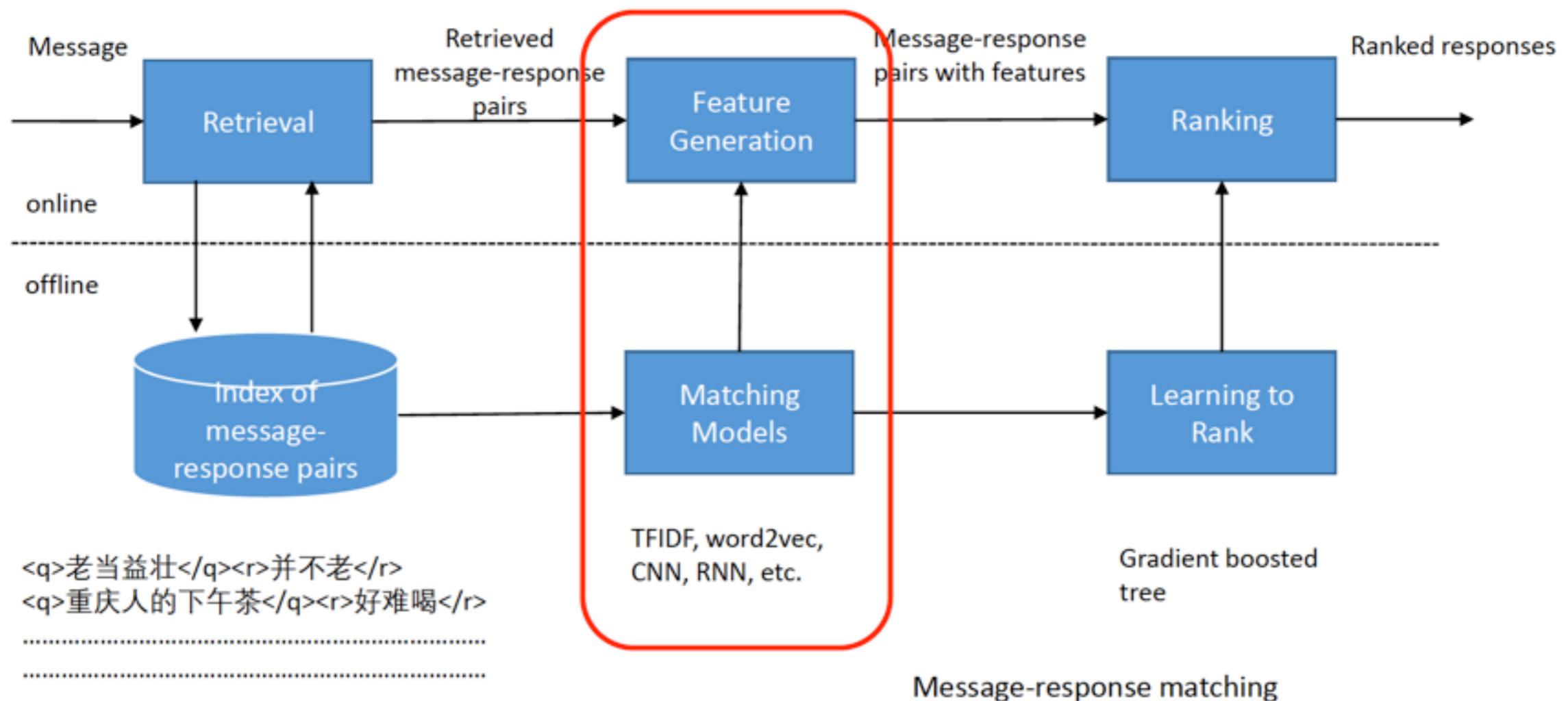
Conversational Agents

Chit-chat Conversational Agent

- Retrieval-based
- Generation-based

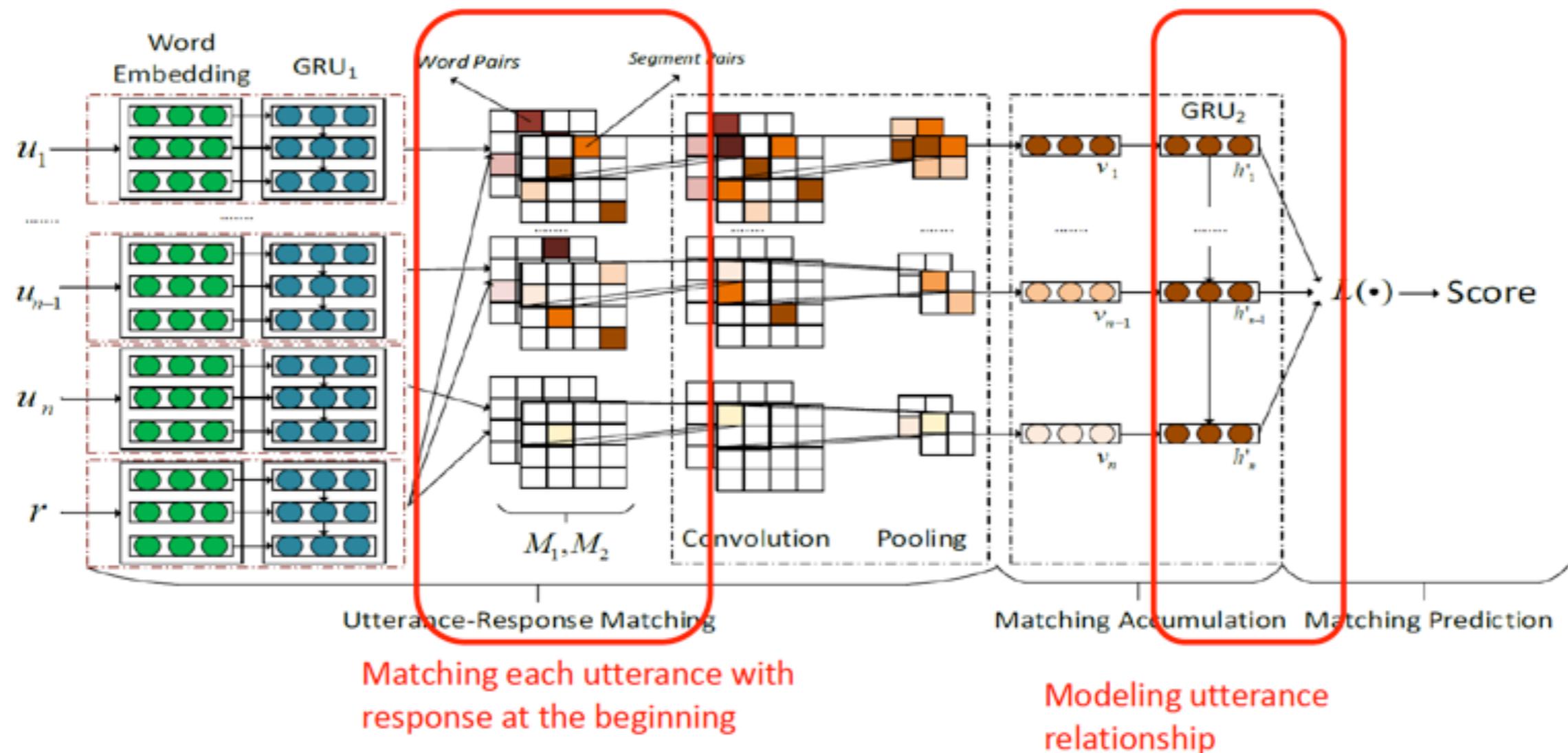
Chit-chat Conversational Agent

- Retrieval-based [9,10]



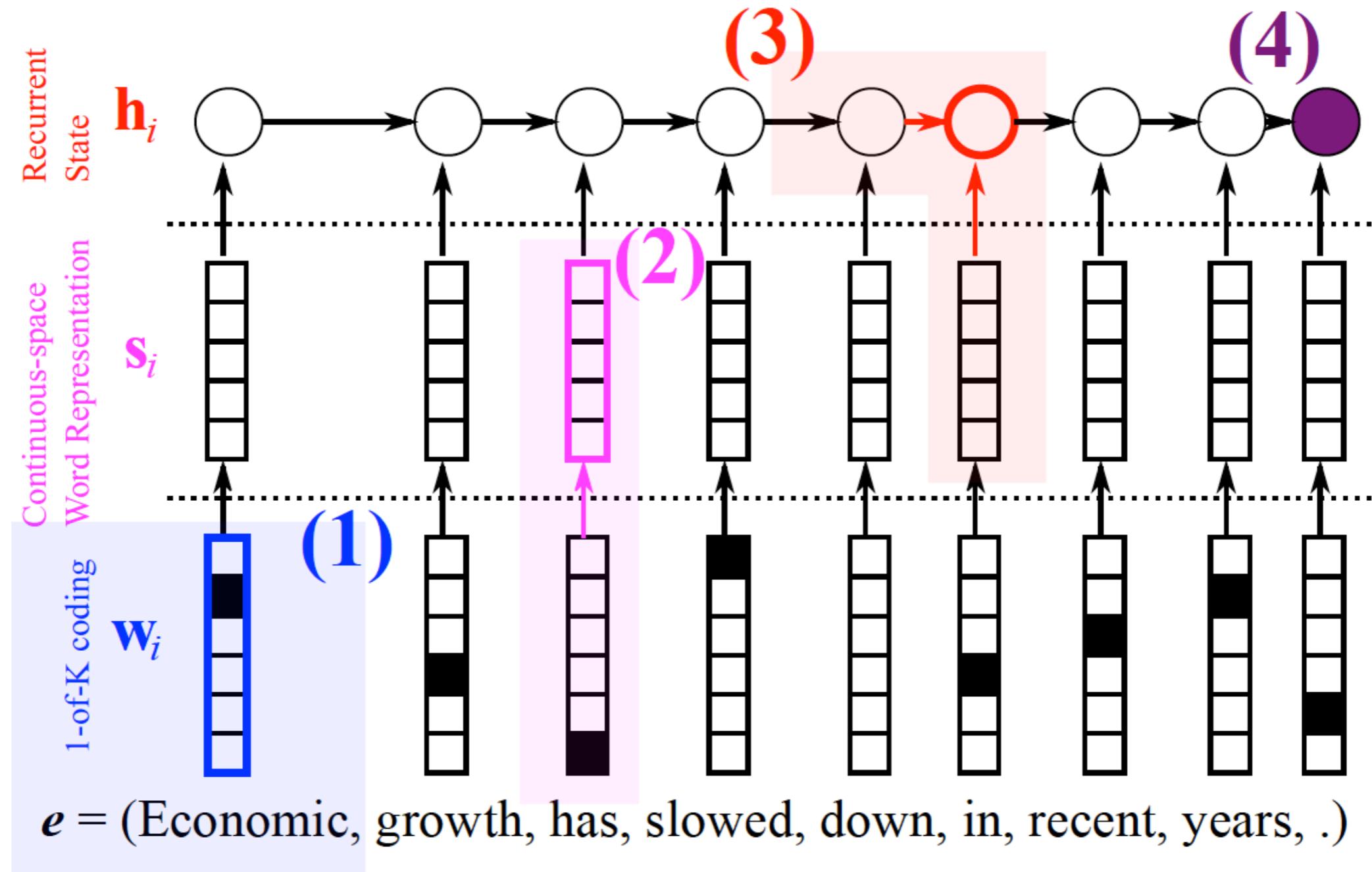
Chit-chat Conversational Agent

- Retrieval-based [11]



Chit-chat Conversational Agent

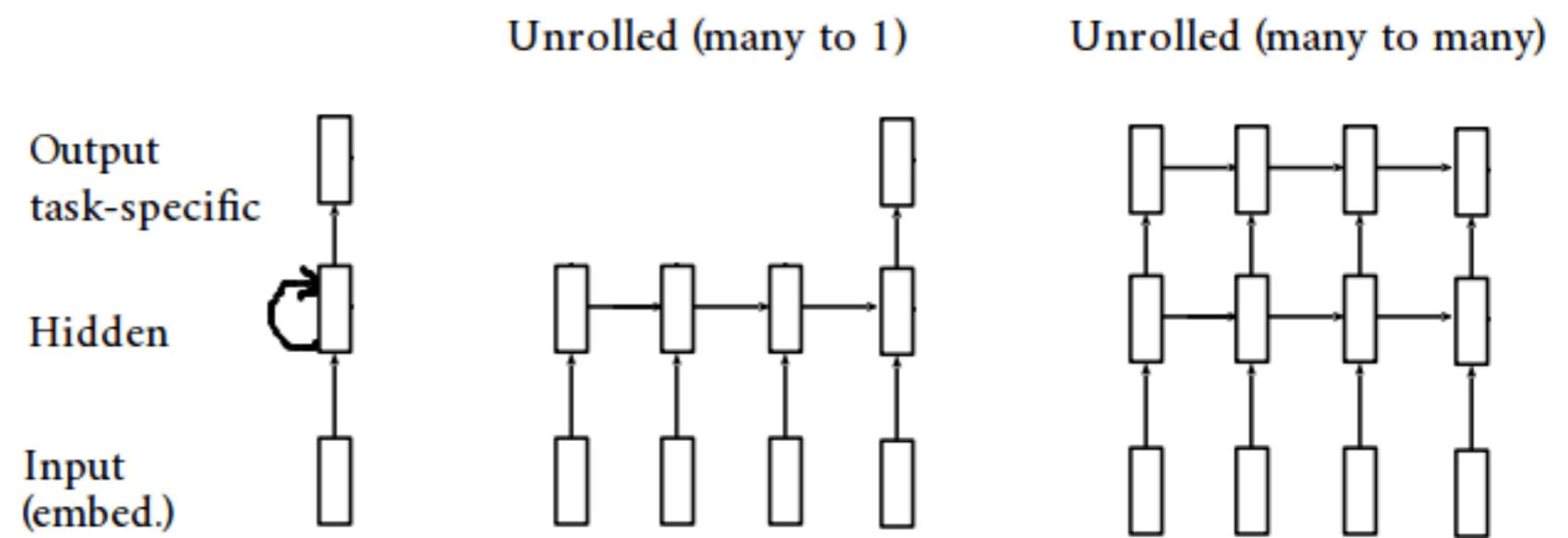
- Generation-based



Recurrent Neural Networks (RNN)

- ♦ Sentence as sequence of words

- ♦ Recursively



RNN Language Model (RNNLM) [2]

- ♦ Non Markovian assumption

cat, dog, is, ...

- ♦ RNNLM

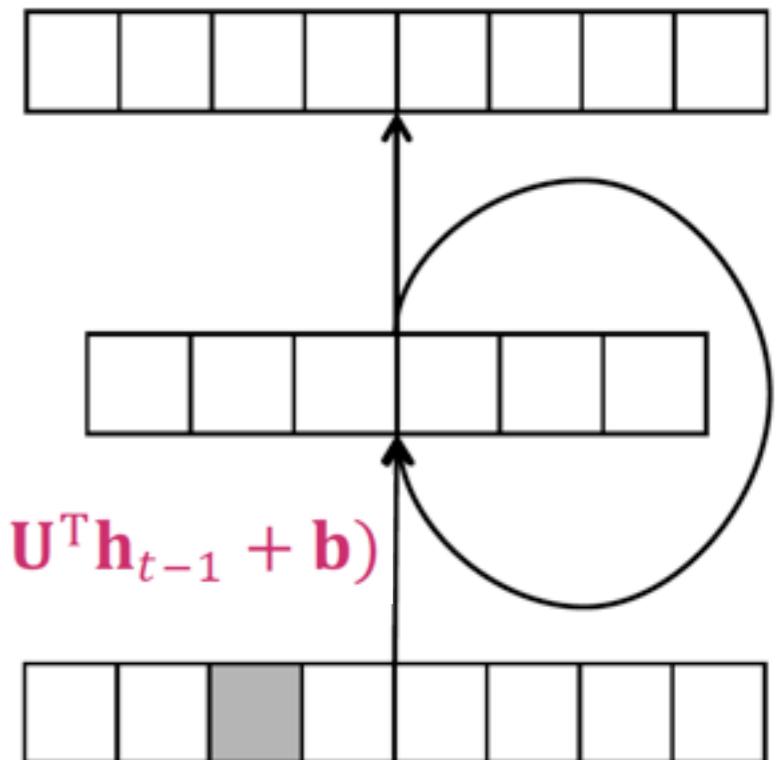
- ♦ 1-of-V encoding for each word x_t

- ♦ Recurrent transition function $\mathbf{h}_t = \tanh(\mathbf{W}^T \mathbf{x}_t + \mathbf{U}^T \mathbf{h}_{t-1} + \mathbf{b})$

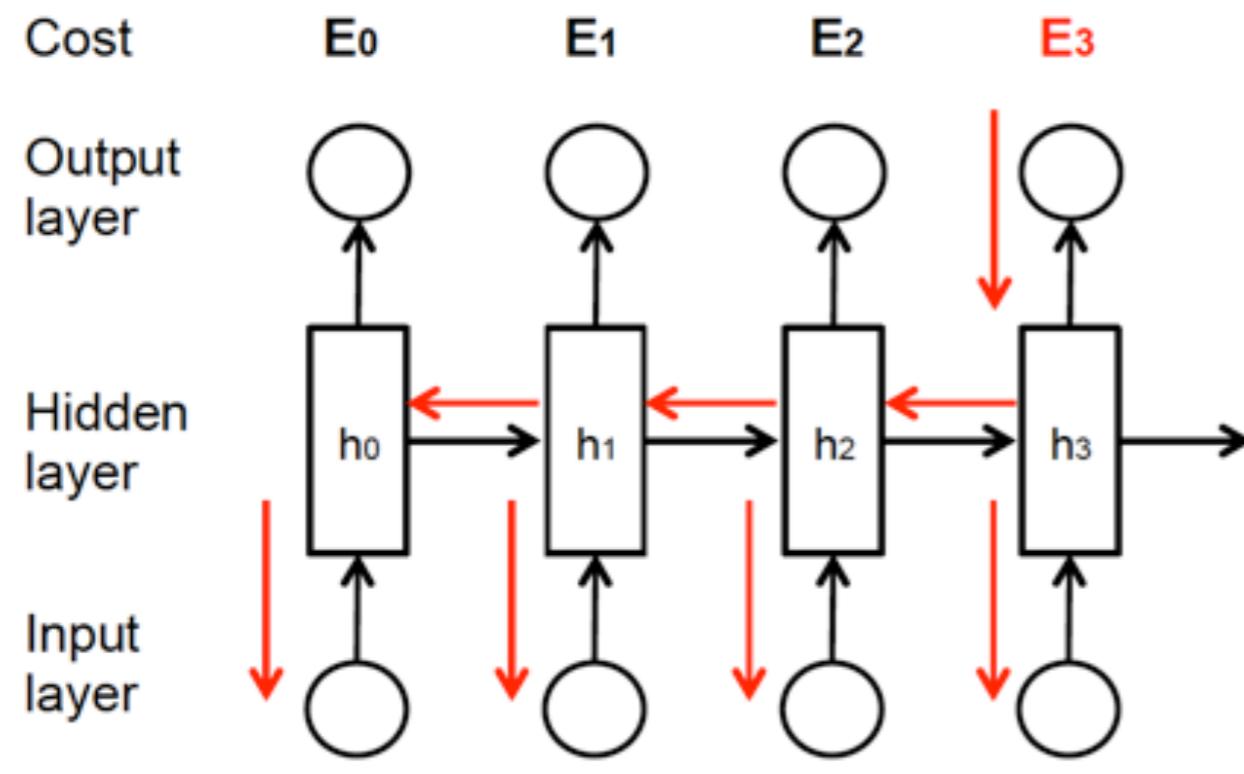
- ♦ Softmax output $\mathbf{p}_t = \text{softmax}(\mathbf{V}^T \mathbf{h}_t + \mathbf{c})$

- ♦ Read, update, predict!

- ♦ Can model dependency of arbitrary length



RNN Optimization & Problem



$$\mathbf{h}_t = \tanh(\mathbf{W}^T \mathbf{x}_t + \mathbf{U}^T \mathbf{h}_{t-1} + \mathbf{b})$$

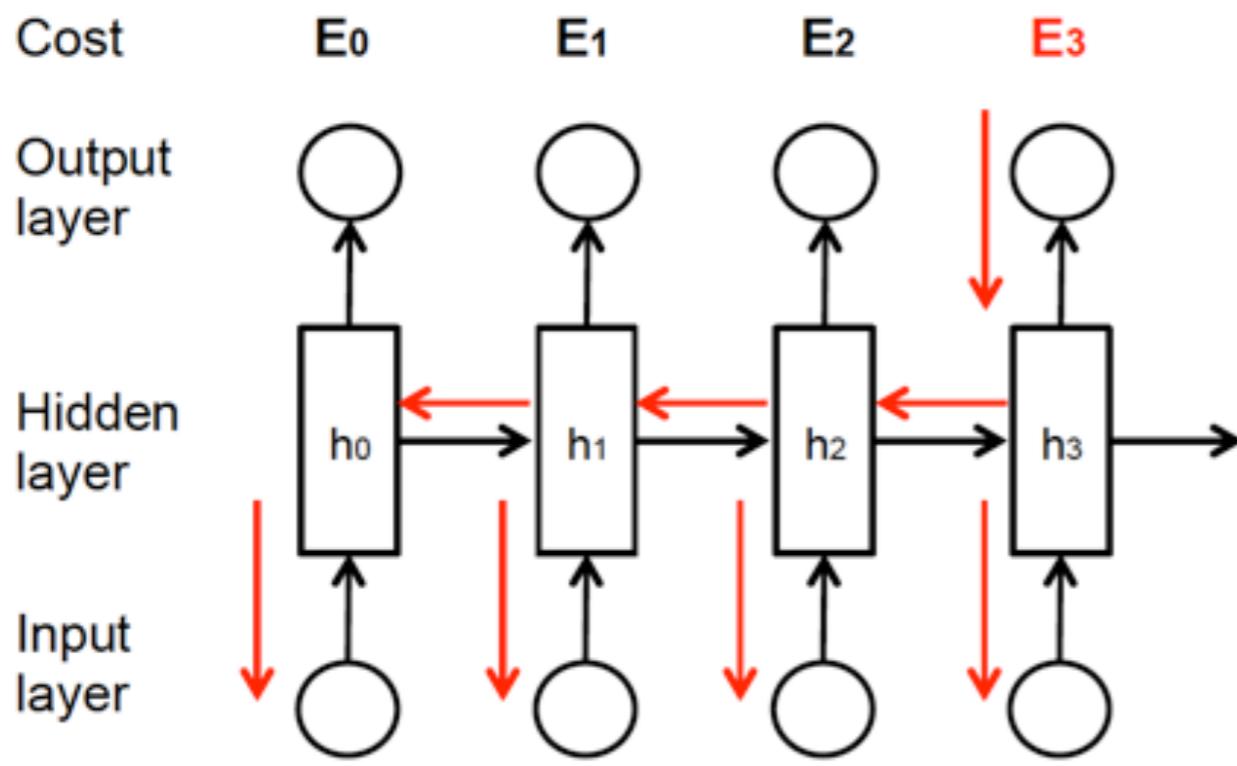
$$\mathbf{p}_t = \text{softmax}(\mathbf{V}^T \mathbf{h}_t + \mathbf{c})$$

$$E_3 = -\mathbf{y}_3^T \log_{10} \mathbf{p}_3$$

$$\frac{\partial E_3}{\partial \mathbf{W}} = \sum_{k=0}^3 \frac{\partial E_3}{\partial \mathbf{p}_3} \frac{\partial \mathbf{p}_3}{\partial \mathbf{h}_3} \frac{\partial \mathbf{h}_3}{\partial \mathbf{h}_k} \frac{\partial \mathbf{h}_k}{\partial \mathbf{W}}$$

$$= \sum_{k=0}^3 \frac{\partial E_3}{\partial \mathbf{p}_3} \frac{\partial \mathbf{p}_3}{\partial \mathbf{h}_3} \left(\prod_{j=k+1}^3 \frac{\partial \mathbf{h}_j}{\partial \mathbf{h}_{j-1}} \right) \frac{\partial \mathbf{h}_k}{\partial \mathbf{W}}$$

RNN Optimization & Problem



$$\mathbf{h}_t = \tanh(\mathbf{W}^T \mathbf{x}_t + \mathbf{U}^T \mathbf{h}_{t-1} + \mathbf{b})$$

$$\mathbf{p}_t = \text{softmax}(\mathbf{V}^T \mathbf{h}_t + \mathbf{c})$$

$$E_3 = -\mathbf{y}_3^T \log_{10} \mathbf{p}_3$$

$$\begin{aligned} \frac{\partial E_3}{\partial \mathbf{W}} &= \sum_{k=0}^3 \frac{\partial E_3}{\partial \mathbf{p}_3} \frac{\partial \mathbf{p}_3}{\partial \mathbf{h}_3} \frac{\partial \mathbf{h}_3}{\partial \mathbf{h}_k} \frac{\partial \mathbf{h}_k}{\partial \mathbf{W}} \\ &= \sum_{k=0}^3 \frac{\partial E_3}{\partial \mathbf{p}_3} \frac{\partial \mathbf{p}_3}{\partial \mathbf{h}_3} \left(\prod_{j=k+1}^3 \frac{\partial \mathbf{h}_j}{\partial \mathbf{h}_{j-1}} \right) \frac{\partial \mathbf{h}_k}{\partial \mathbf{W}} \end{aligned}$$

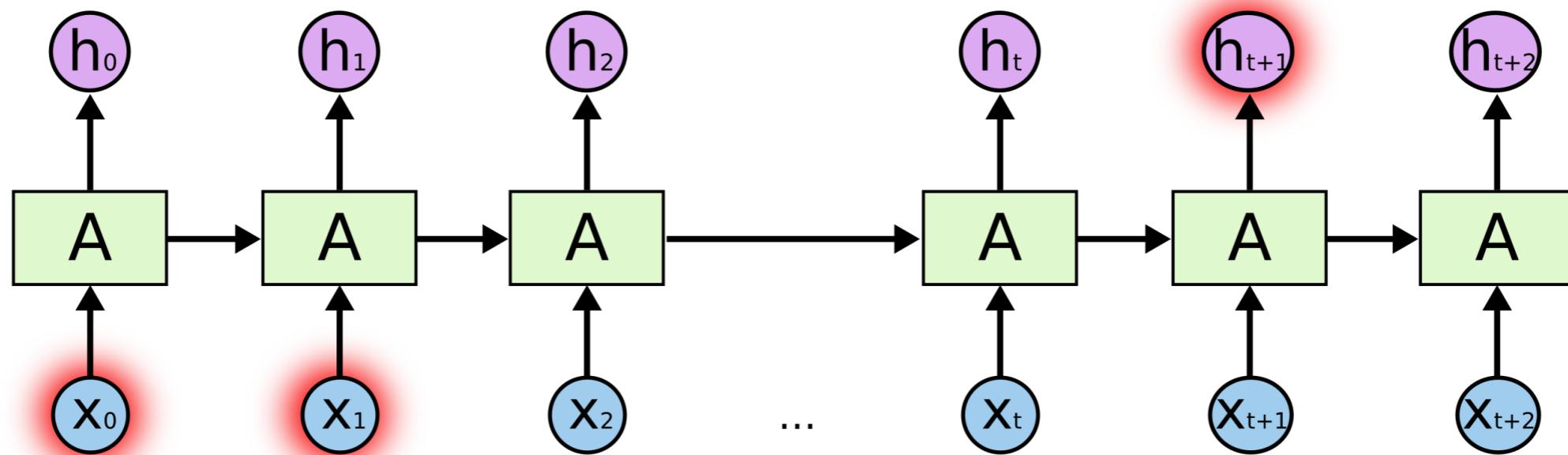
$$\frac{\partial \mathbf{h}_j}{\partial \mathbf{h}_{j-1}} = \mathbf{U}^T \cdot \text{diag}(\tanh'(\mathbf{m}_j)) \quad \xleftarrow{\text{Jacobian Matrix}}$$

$$\mathbf{m}_j = \mathbf{W}^T \mathbf{x}_j + \mathbf{U}^T \mathbf{h}_{j-1} + \mathbf{b}$$

$\|\mathbf{U}\| \cdot \|\text{diag}(\tanh'(\mathbf{m}_j))\| < 1$ **Vanishing Gradient Problem!** [1]

Long-Short Term Memory (LSTM)

- ◆ RNN has difficulty in long dependancies [3]

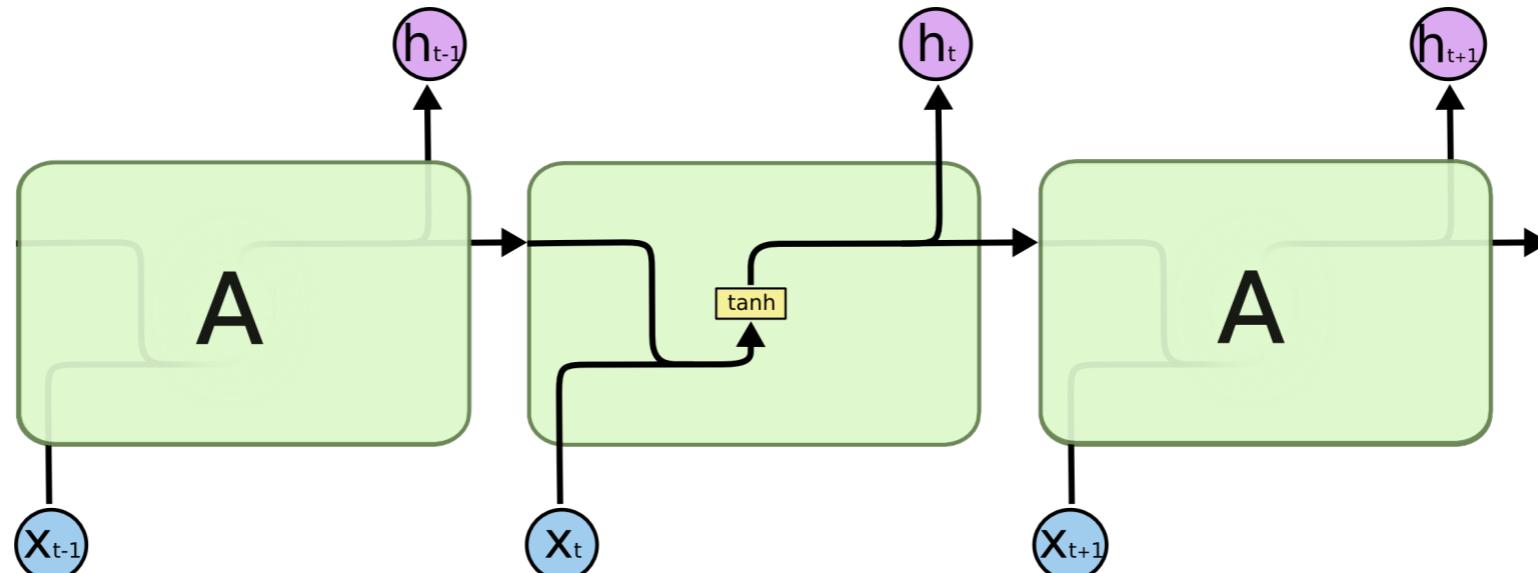


Long-Short Term Memory (LSTM)

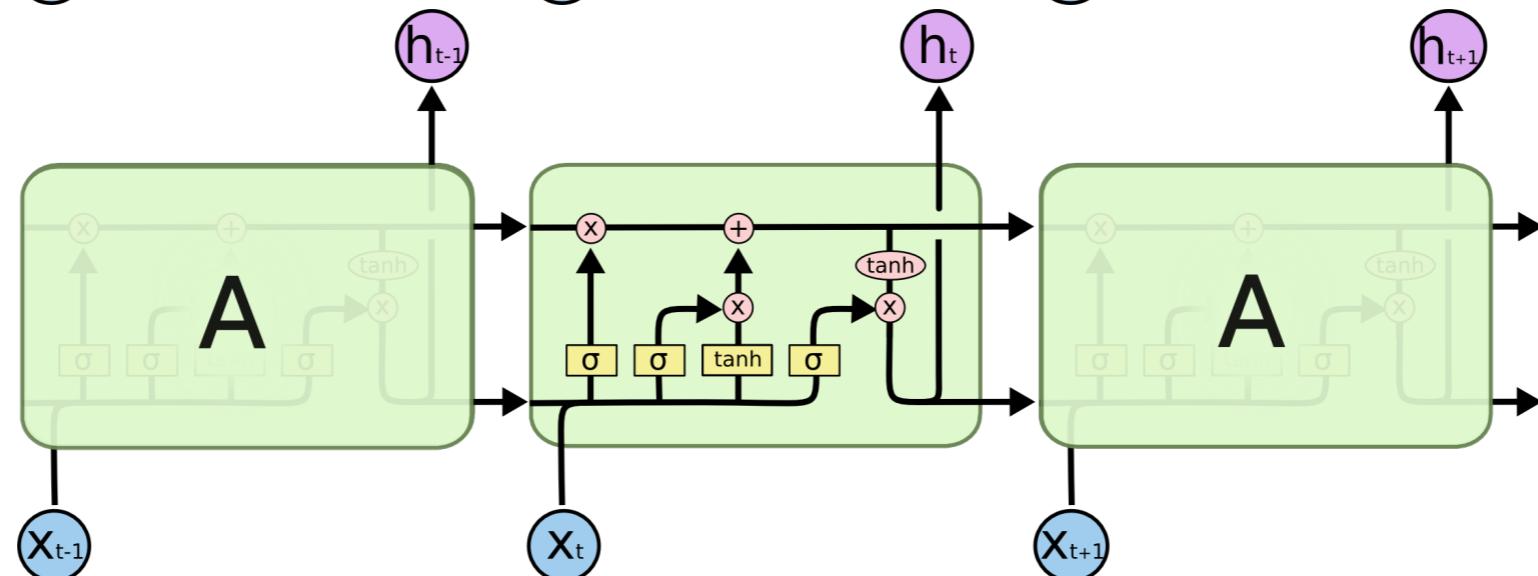
- ♦ Four interacting layers [3]



- ♦ RNN



- ♦ LSTM



Long-Short Term Memory (LSTM)

- ◆ Three sigmoid **gates** and One **cell**

$$\mathbf{i}_t = \sigma(\mathbf{W}_{wi}\mathbf{x}_t + \mathbf{W}_{hi}\mathbf{h}_{t-1})$$

$$\mathbf{f}_t = \sigma(\mathbf{W}_{wf}\mathbf{x}_t + \mathbf{W}_{hf}\mathbf{h}_{t-1})$$

$$\mathbf{o}_t = \sigma(\mathbf{W}_{wo}\mathbf{x}_t + \mathbf{W}_{ho}\mathbf{h}_{t-1})$$

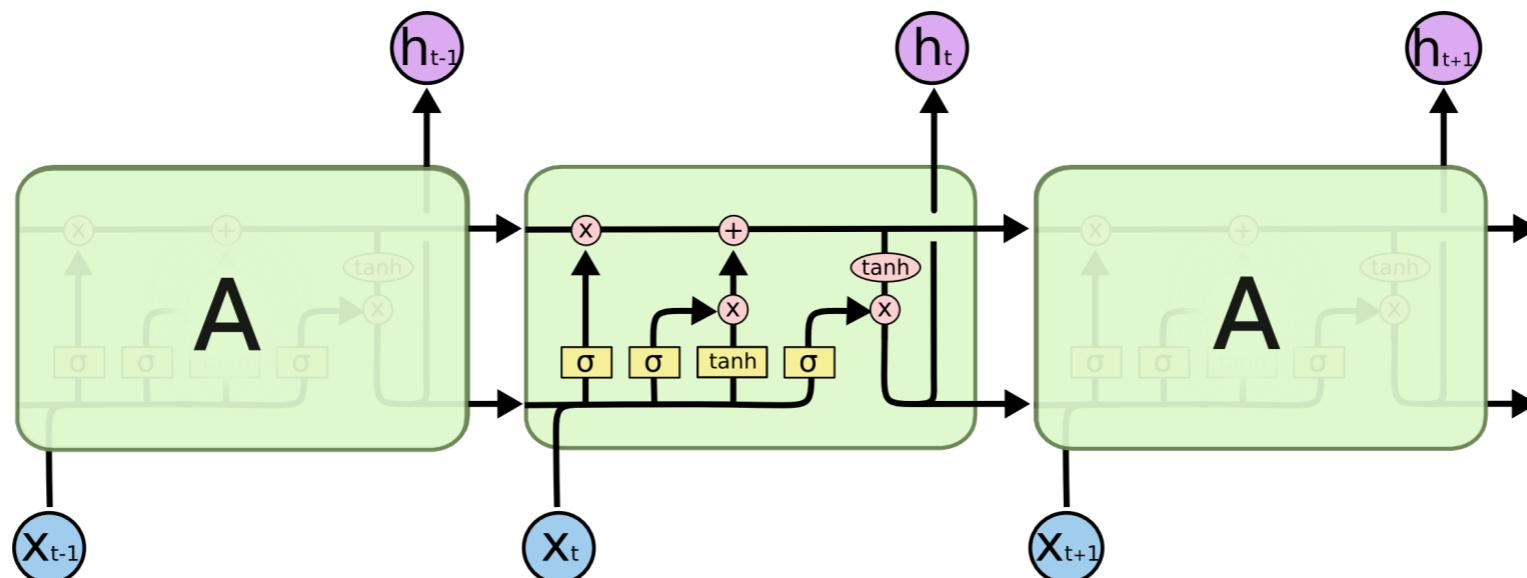
$$\hat{\mathbf{c}}_t = \tanh(\mathbf{W}_{wc}\mathbf{x}_t + \mathbf{W}_{hc}\mathbf{h}_{t-1})$$



$$\mathbf{c}_t = \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \hat{\mathbf{c}}_t$$

$$\mathbf{h}_t = \mathbf{o}_t \odot \tanh(\mathbf{c}_t)$$

- ◆ LSTM



Sigmoid gates - forget

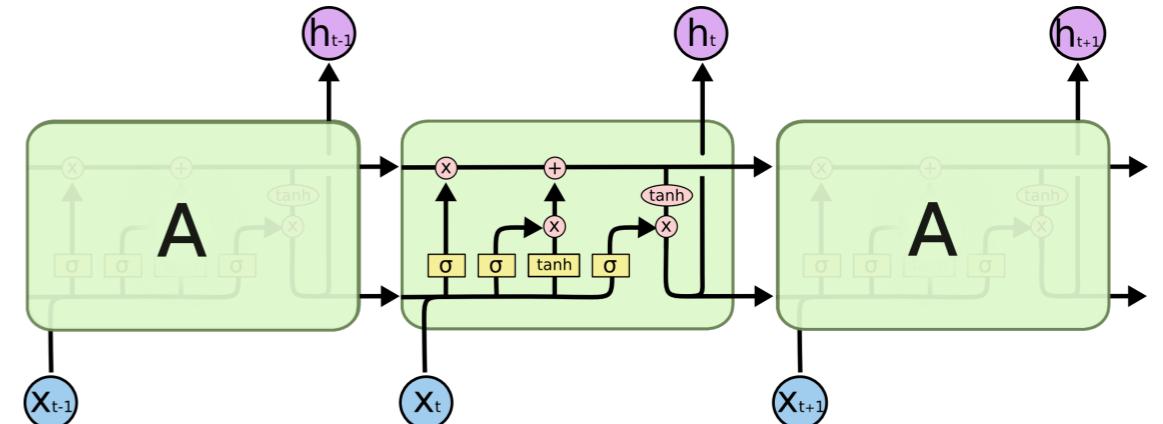
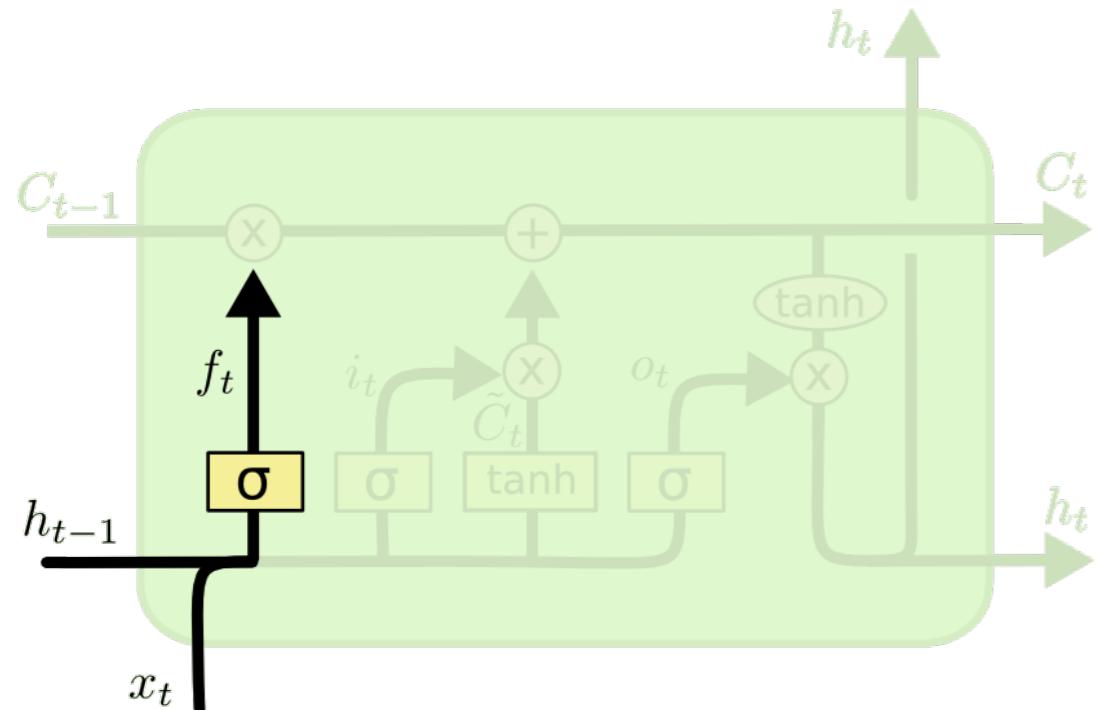
- Three sigmoid **gates** and One **cell**

$$\mathbf{i}_t = \sigma(\mathbf{W}_{wi}\mathbf{x}_t + \mathbf{W}_{hi}\mathbf{h}_{t-1})$$

$$\mathbf{f}_t = \sigma(\mathbf{W}_{wf}\mathbf{x}_t + \mathbf{W}_{hf}\mathbf{h}_{t-1})$$

$$\mathbf{o}_t = \sigma(\mathbf{W}_{wo}\mathbf{x}_t + \mathbf{W}_{ho}\mathbf{h}_{t-1})$$

$$\hat{\mathbf{c}}_t = \tanh(\mathbf{W}_{wc}\mathbf{x}_t + \mathbf{W}_{hc}\mathbf{h}_{t-1})$$



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

Sigmoid gates - new information

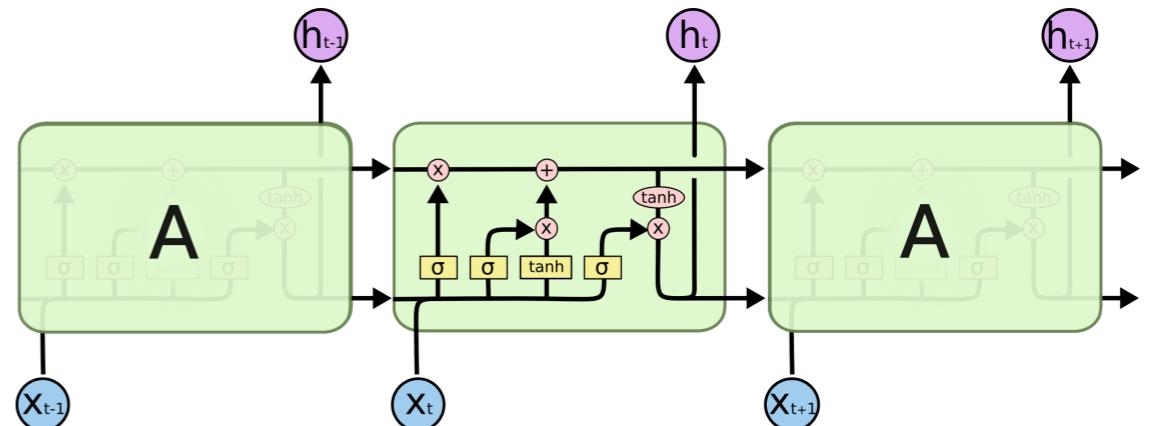
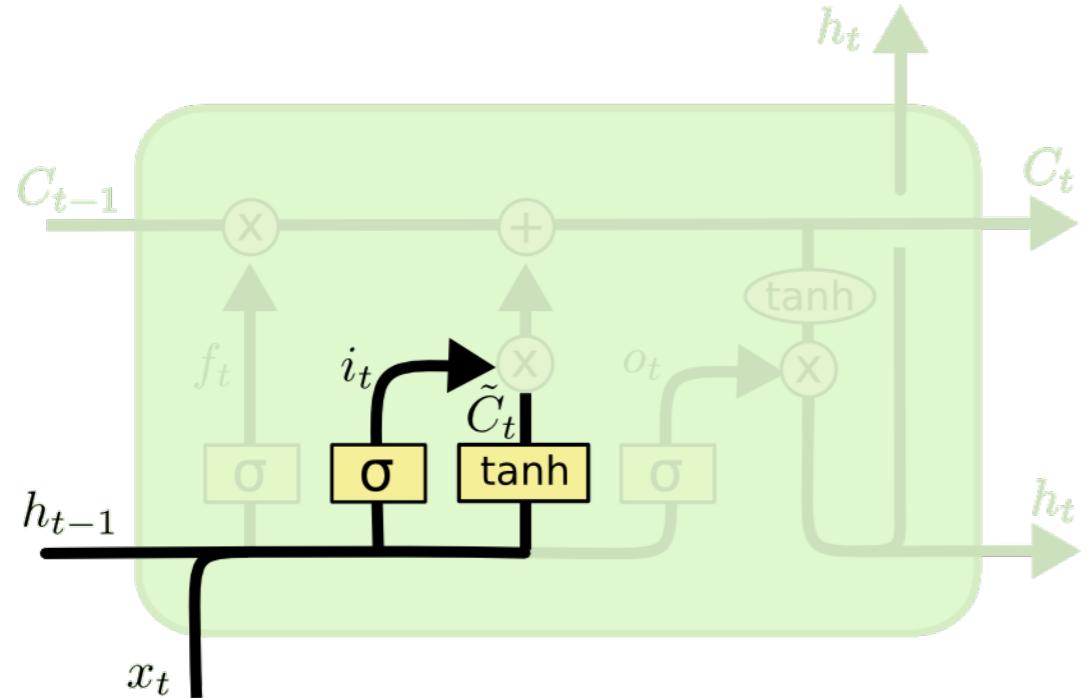
- Three sigmoid **gates** and One **cell**

$$\mathbf{i}_t = \sigma(\mathbf{W}_{wi}\mathbf{x}_t + \mathbf{W}_{hi}\mathbf{h}_{t-1})$$

$$\mathbf{f}_t = \sigma(\mathbf{W}_{wf}\mathbf{x}_t + \mathbf{W}_{hf}\mathbf{h}_{t-1})$$

$$\mathbf{o}_t = \sigma(\mathbf{W}_{wo}\mathbf{x}_t + \mathbf{W}_{ho}\mathbf{h}_{t-1})$$

$$\hat{\mathbf{c}}_t = \tanh(\mathbf{W}_{wc}\mathbf{x}_t + \mathbf{W}_{hc}\mathbf{h}_{t-1})$$



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

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

Sigmoid gates - Cell Value

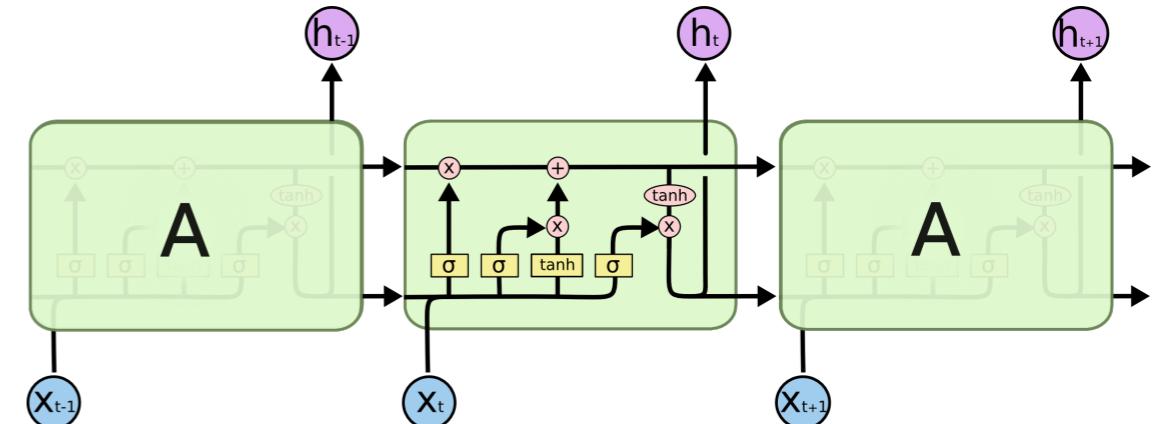
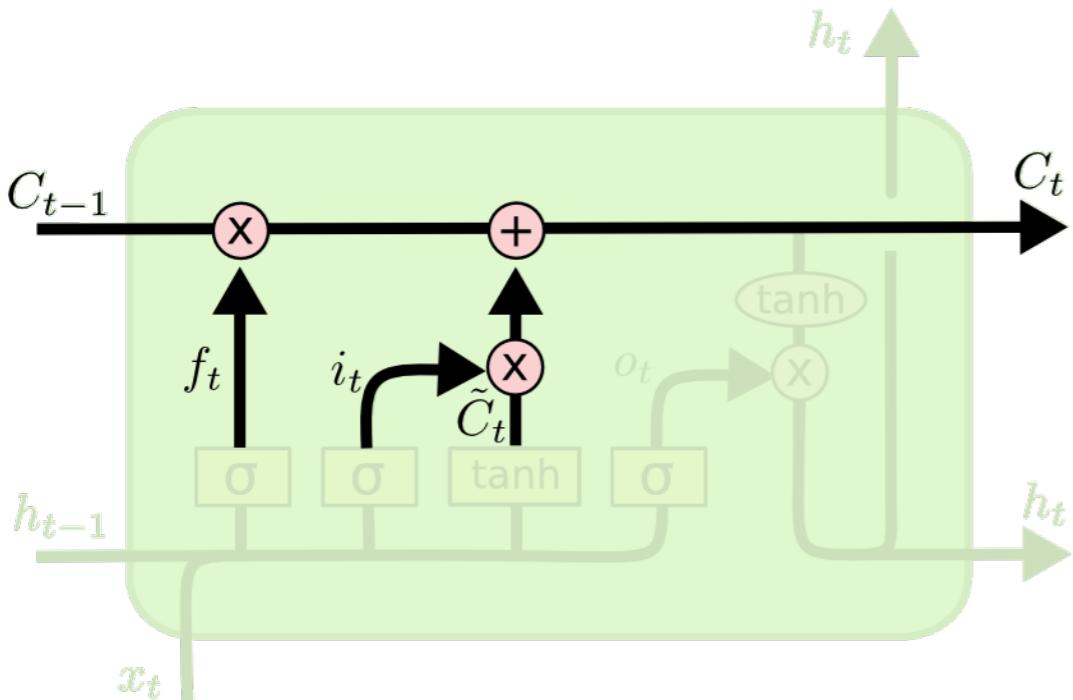
- ♦ Three sigmoid **gates** and One **cell**

$$\mathbf{i}_t = \sigma(\mathbf{W}_{wi}\mathbf{x}_t + \mathbf{W}_{hi}\mathbf{h}_{t-1})$$

$$\mathbf{f}_t = \sigma(\mathbf{W}_{wf}\mathbf{x}_t + \mathbf{W}_{hf}\mathbf{h}_{t-1})$$

$$\mathbf{o}_t = \sigma(\mathbf{W}_{wo}\mathbf{x}_t + \mathbf{W}_{ho}\mathbf{h}_{t-1})$$

$$\hat{\mathbf{c}}_t = \tanh(\mathbf{W}_{wc}\mathbf{x}_t + \mathbf{W}_{hc}\mathbf{h}_{t-1})$$



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

Sigmoid gates - Output

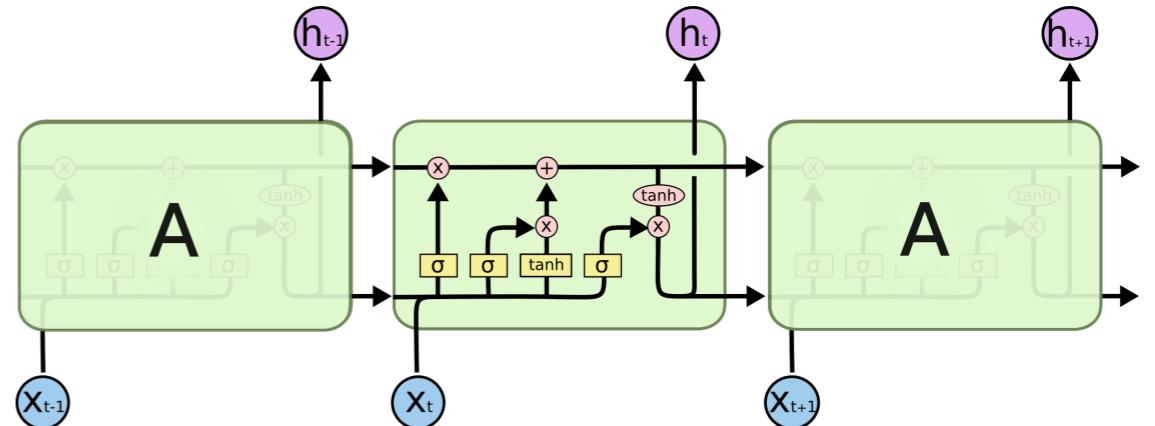
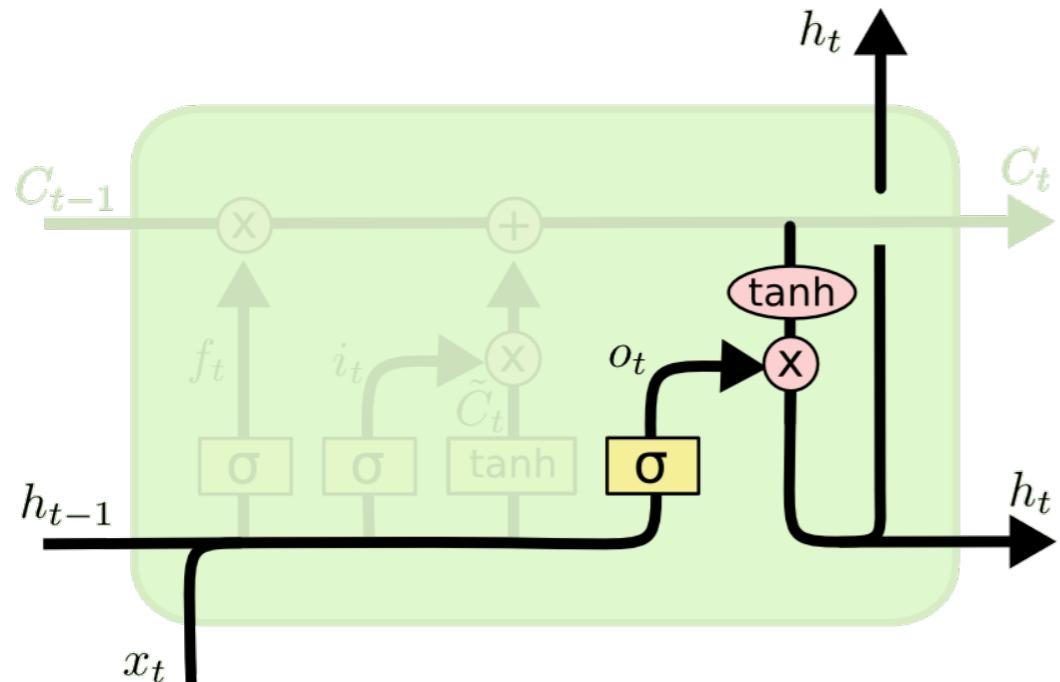
- Three sigmoid **gates** and One **cell**

$$\mathbf{i}_t = \sigma(\mathbf{W}_{wi}\mathbf{x}_t + \mathbf{W}_{hi}\mathbf{h}_{t-1})$$

$$\mathbf{f}_t = \sigma(\mathbf{W}_{wf}\mathbf{x}_t + \mathbf{W}_{hf}\mathbf{h}_{t-1})$$

$$\mathbf{o}_t = \sigma(\mathbf{W}_{wo}\mathbf{x}_t + \mathbf{W}_{ho}\mathbf{h}_{t-1})$$

$$\hat{\mathbf{c}}_t = \tanh(\mathbf{W}_{wc}\mathbf{x}_t + \mathbf{W}_{hc}\mathbf{h}_{t-1})$$



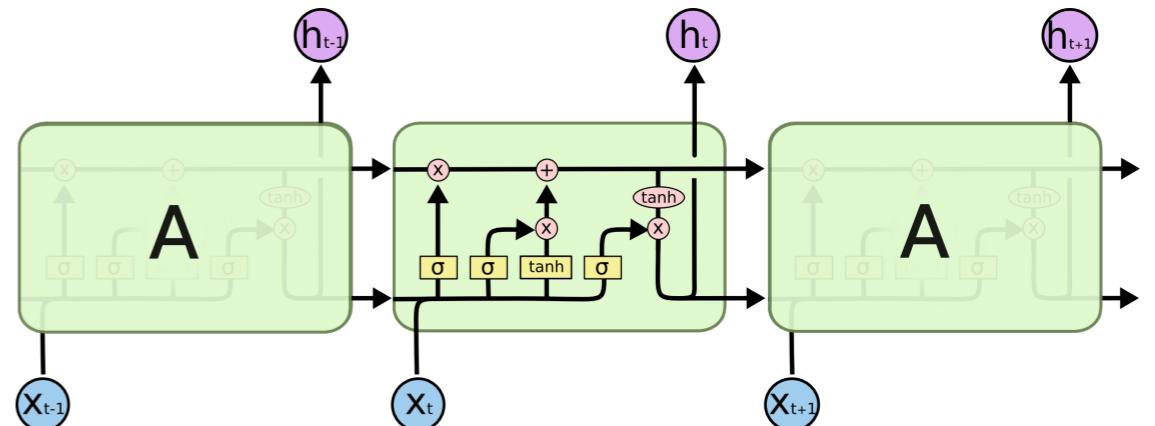
$$o_t = \sigma(W_o [h_{t-1}, x_t] + b_o)$$

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

LSTM theoretically prevent gradient vanishing

- ◆ Consider memory cell update

$$\mathbf{C}_t = \mathbf{i}_t \odot \hat{\mathbf{C}}_t + \mathbf{f}_t \odot \mathbf{C}_{t-1}$$



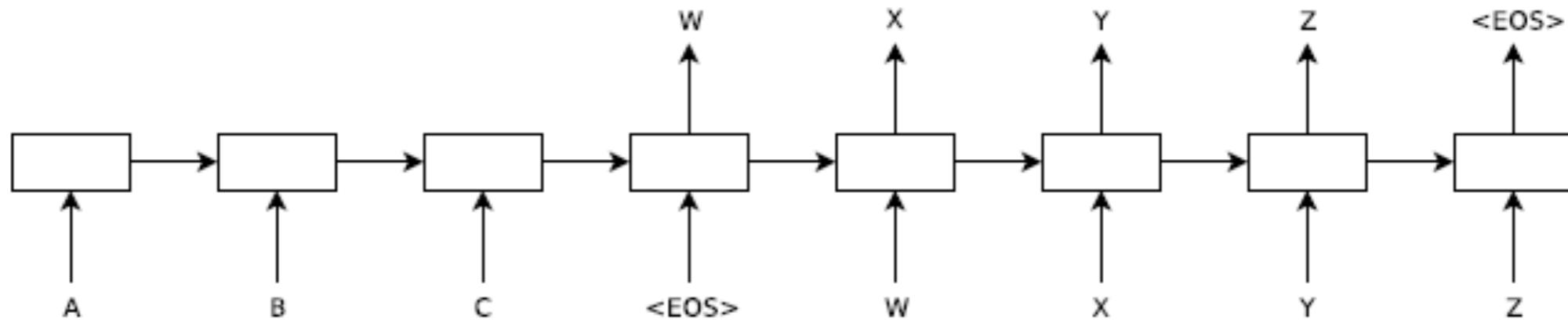
- ◆ We can back-prop the gradient by chain rule

$$\frac{\partial E_t}{\partial C_{t-1}} = \frac{\partial E_t}{\partial C_t} \frac{\partial C_t}{\partial C_{t-1}} = \frac{\partial E_t}{\partial C_t} f_t$$

- ◆ If f_t maintains a value of 1, gradient is perfectly propagated.

Sequence to sequence learning with

- ◆ The encoder-decoder framework [4]

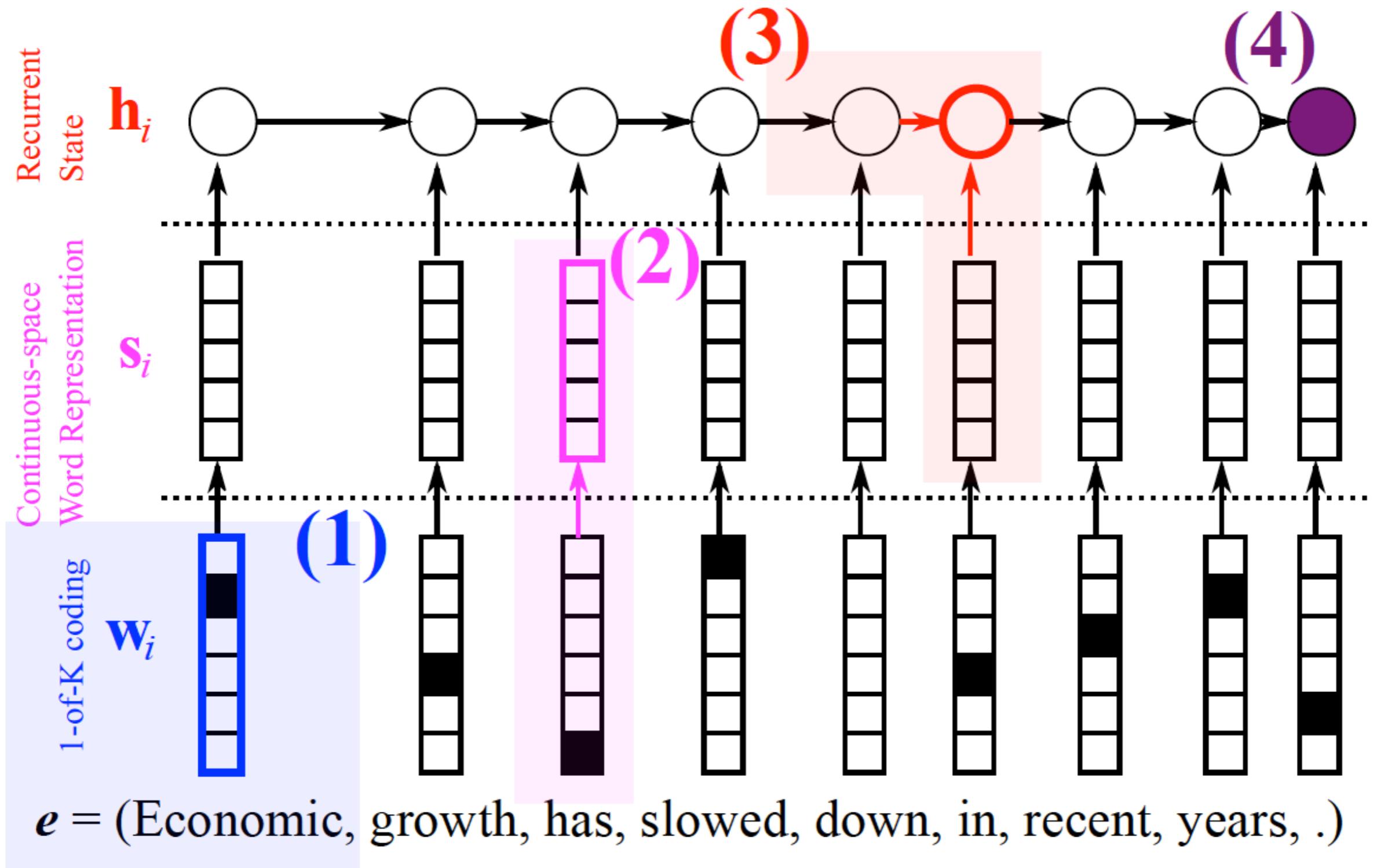


- ◆ Encoder reads sentence “A B C” in source language
- ◆ Decoder generates sentence “W X Y Z” in target language

The encoder-decoder framework

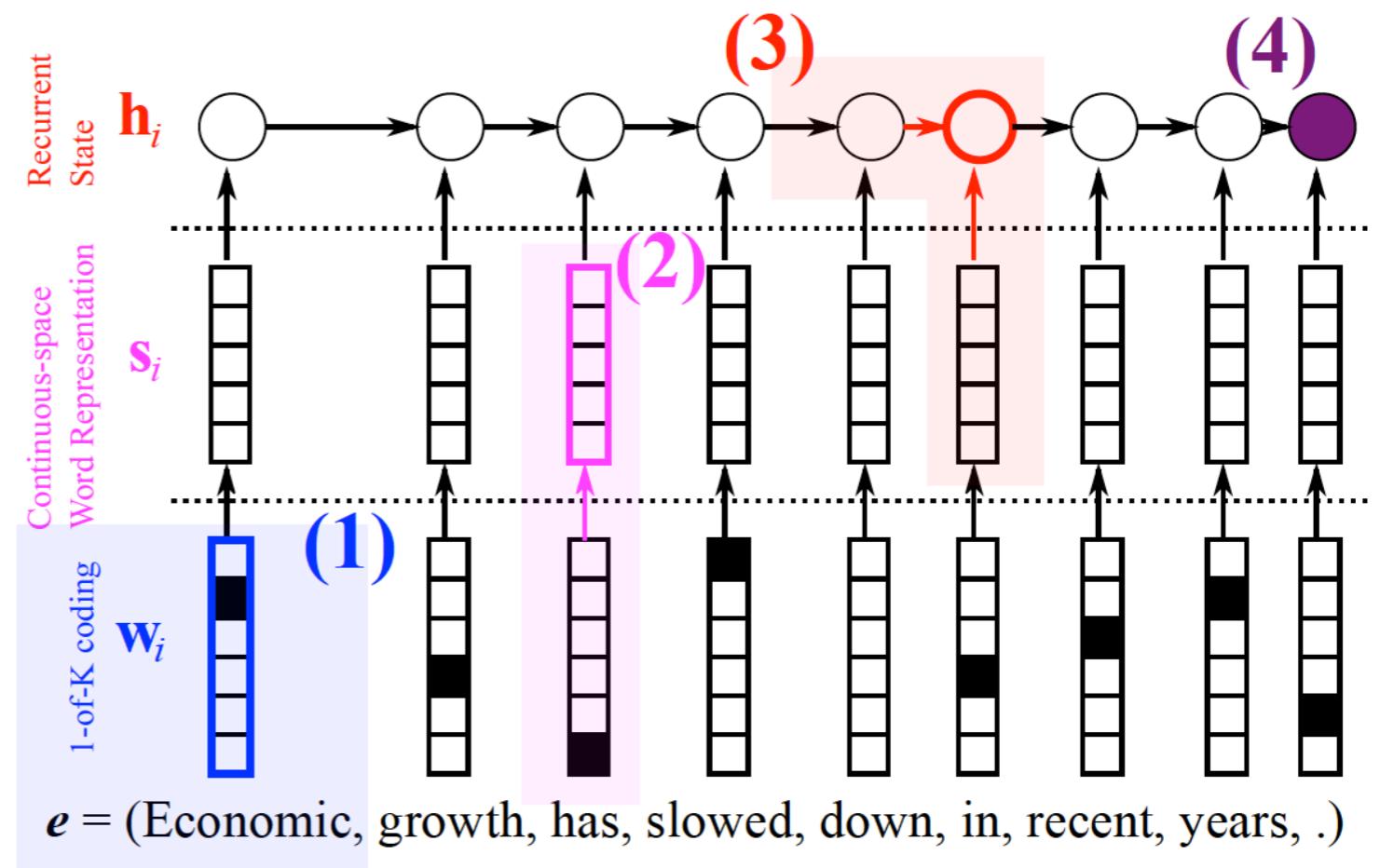
- ◆ Comprehensive
 - ◆ It unites **content understanding** and **language modelling** in a single architecture.
 - ◆ Thus, it allows the **end-to-end automatic** generation.
- ◆ Flexible
 - ◆ It allows to condition on anything, i.e. images, videos, words, etc.
- ◆ Scalable
 - ◆ Able to supplement the language model with external corpora.

The encoder-decoder framework – Encoder



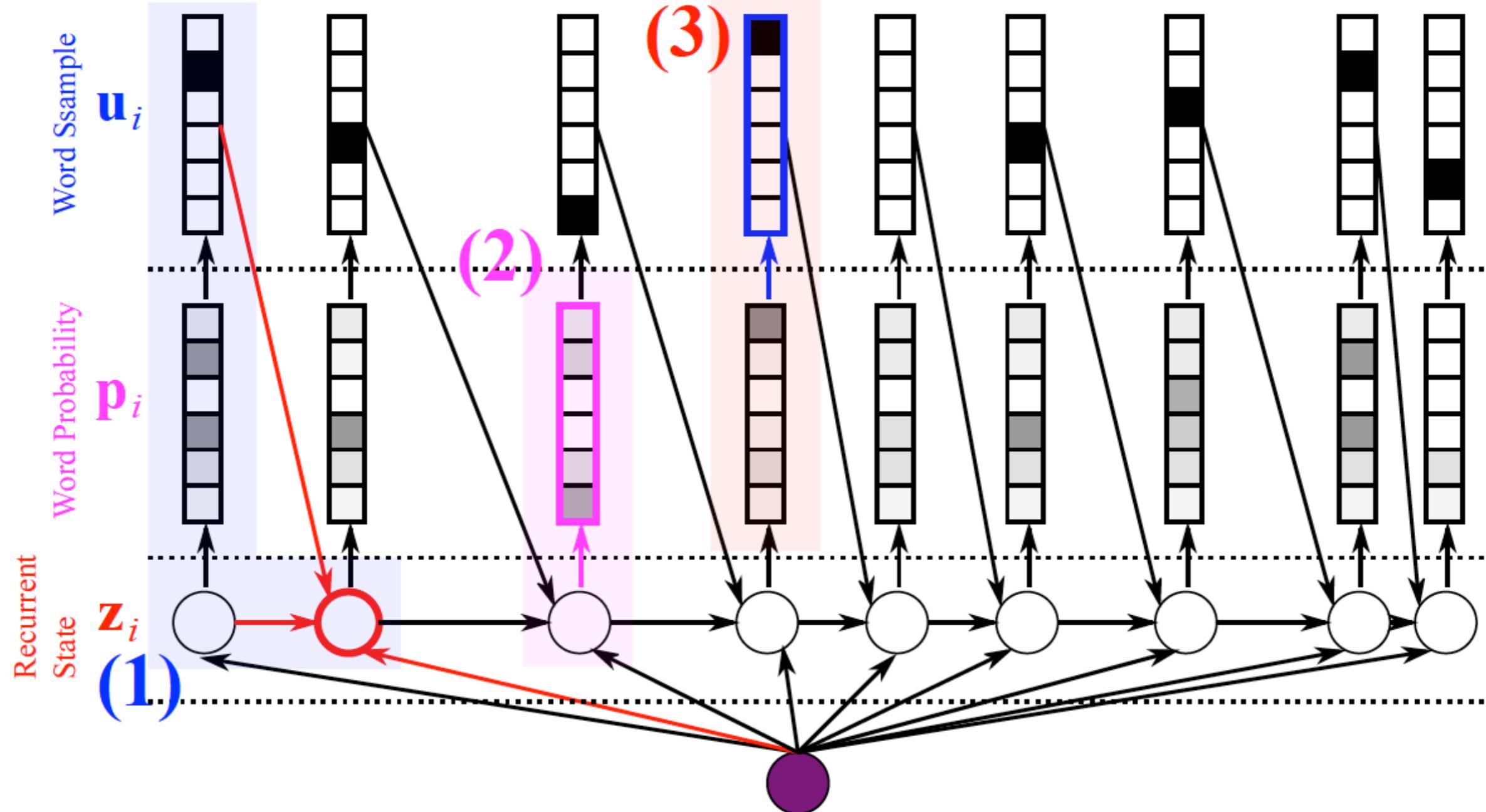
The encoder-decoder framework – Encoder

- ◆ 1-of-K coding of source words
- ◆ Continuous-space representation
- ◆ Recursively read words



The encoder-decoder framework – Decoder

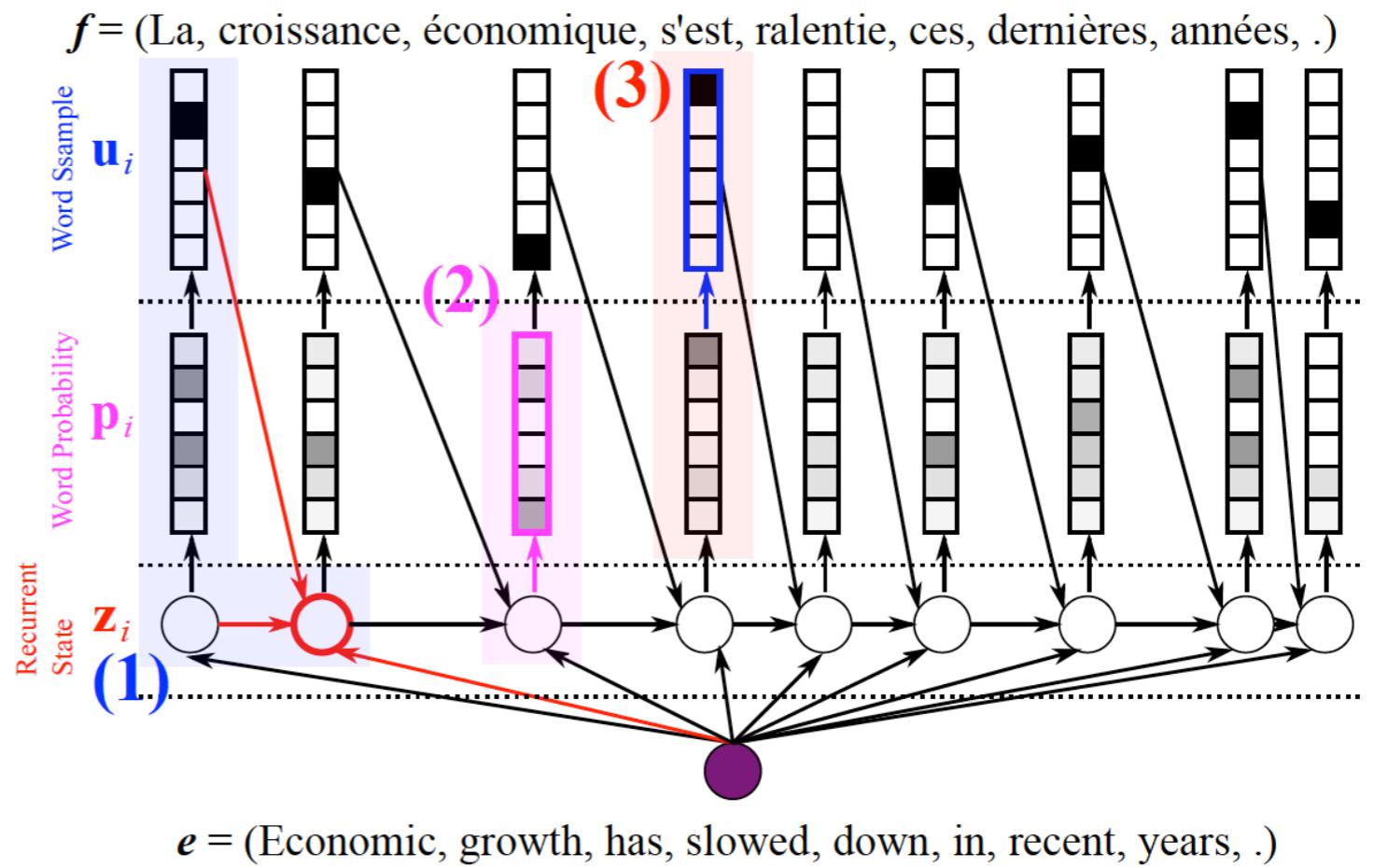
$f = (\text{La, croissance, économique, s'est, ralenti, ces, dernières, années, .})$



$e = (\text{Economic, growth, has, slowed, down, in, recent, years, .})$

The encoder-decoder framework – Decoder

- ♦ Recursively update the memory
- ♦ Compute the next word probability
- ♦ Sample a next word



A Neural Conversational Model

- ♦ The encoder-decoder [6]

Human: *what is the purpose of life ?*

Machine: *to serve the greater good .*

Human: *what is the purpose of living ?*

Machine: *to live forever .*

Human: *what is the purpose of existence ?*

Machine: *to find out what happens when we get to the planet earth .*

Human: *where are you now ?*

Machine: *i 'm in the middle of nowhere .*

Human: *what is the purpose of dying ?*

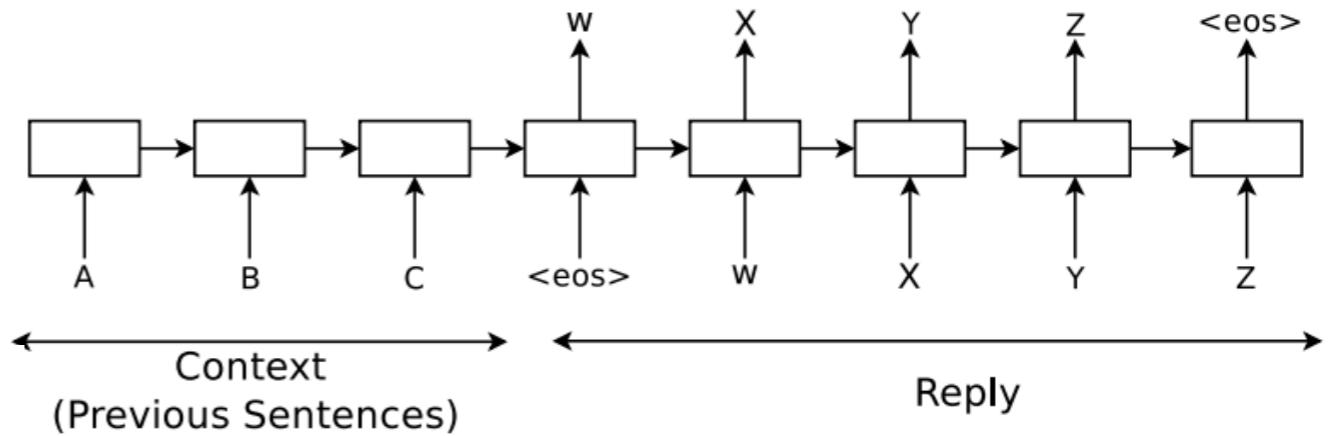
Machine: *to have a life .*

Human: *what is the purpose of being intelligent ?*

Machine: *to find out what it is .*

Human: *what is the purpose of emotions ?*

Machine: *i don 't know .*



Human: *what do you think about tesla ?*

Machine: *he 's a good conductor .*

Human: *what do you think about bill gates ?*

Machine: *he 's a good man .*

Human: *What do you think about messi ?*

Machine: *he 's a great player .*

Human: *what do you think about cleopatra ?*

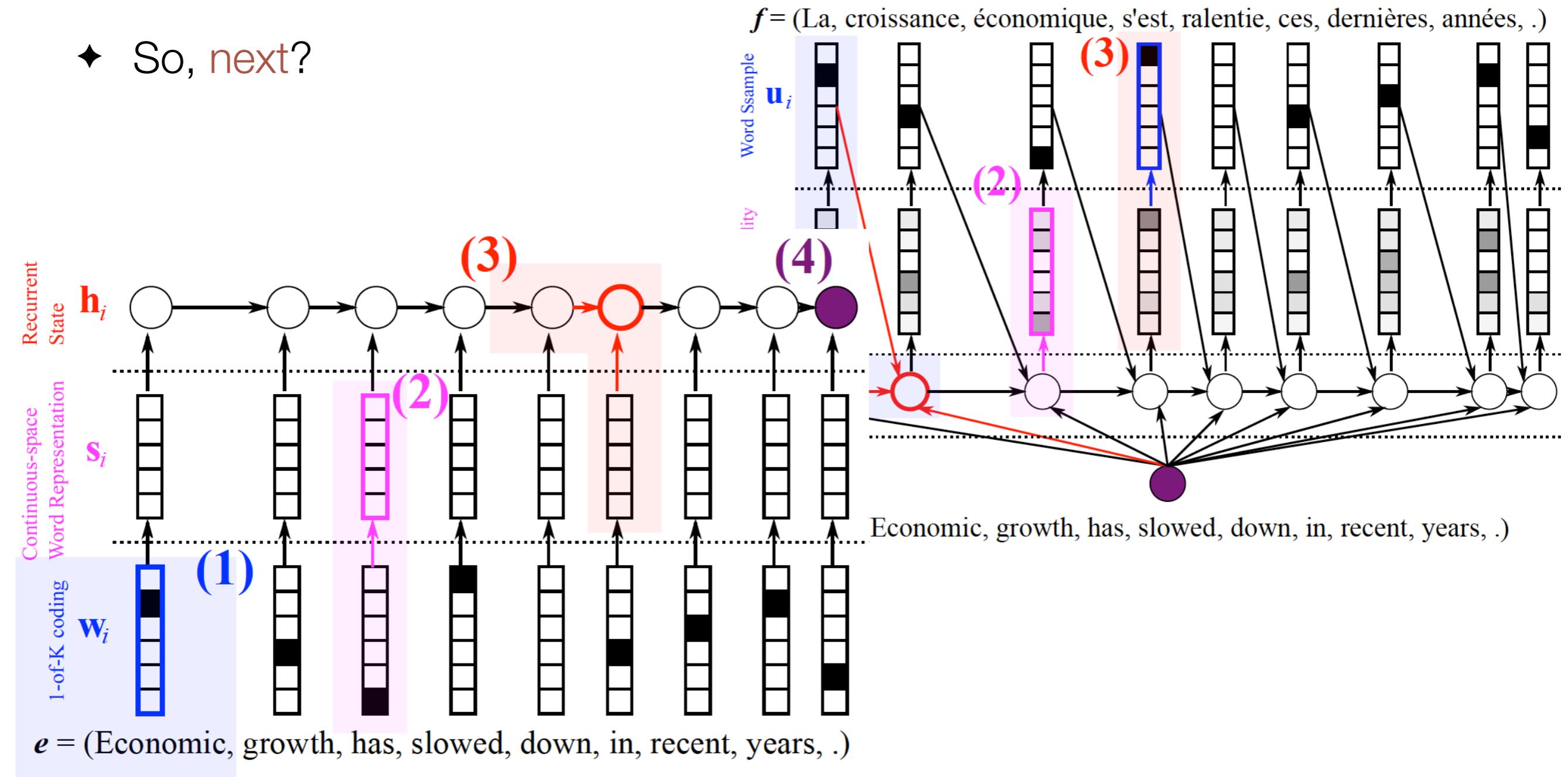
Machine: *oh , she 's very regal .*

Human: *what do you think about england during the reign of elizabeth ?*

Machine: *it was a great place .*

It looks fantastic, but...

- ◆ “You can’t cram the meaning of a whole %&!\$# sentence into a single \$&! #* vector!”
- ◆ So, next?



Difficulties for all generation tasks

- ◆ RNN/LSTM's problem
 - ◆ impotent for too long sentences (semantic drift)
 - ◆ grammatically problematic
 - ◆ highly frequent pattern domination (less variation)

Difficulties for all generation tasks

♦ RNN/LSTM’s problem^[8]

i went to the store to buy some groceries .
i store to buy some groceries .
i were to buy any groceries .
horses are to buy any groceries .
horses are to buy any animal .
horses the favorite any animal .
horses the favorite favorite animal .
horses are my favorite animal .

Table 1: Sentences produced by greedily decoding from points between two sentence encodings with a conventional autoencoder. The intermediate sentences are not plausible English.

“ i want to talk to you . ”
“*i want to be with you . ”*
“*i do n’t want to be with you . ”*
i do n’t want to be with you .
she did n’t want to be with him .

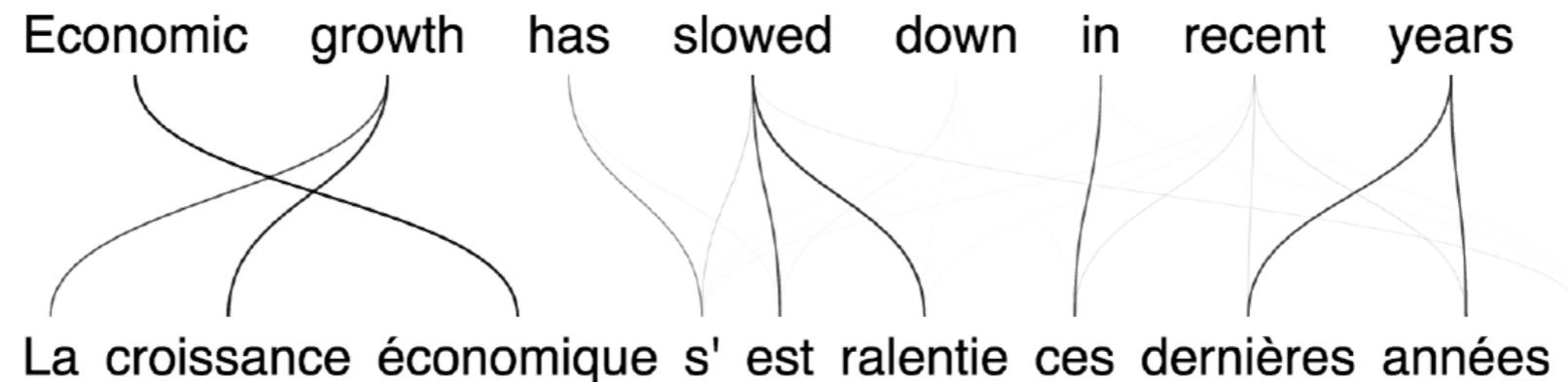
he was silent for a long moment .
he was silent for a moment .
it was quiet for a moment .
it was dark and cold .
there was a pause .
it was my turn .

Table 8: Paths between pairs of random points in VAE space: Note that intermediate sentences are grammatical, and that topic and syntactic structure are usually locally consistent.

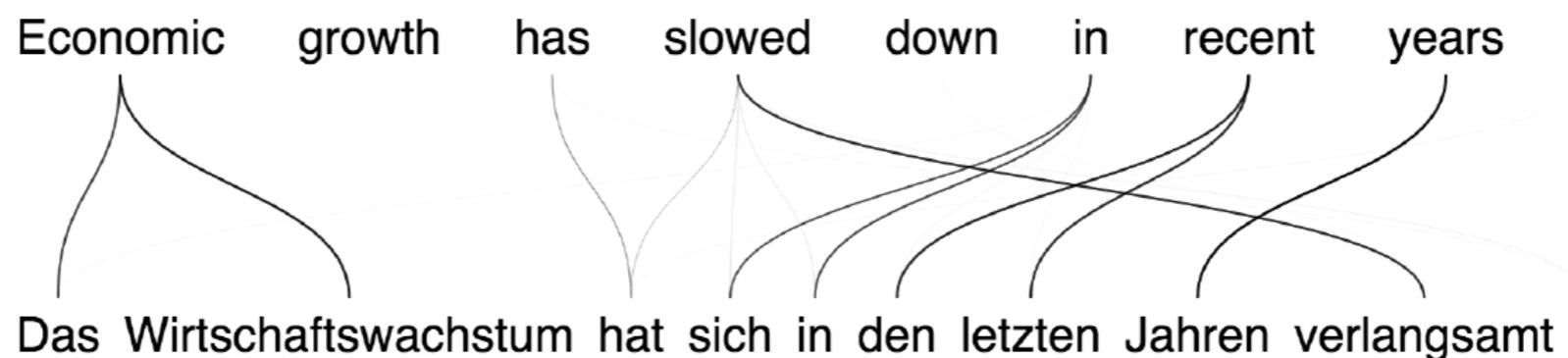
Attention-based Neural Machine Translation

- ♦ fixed representation of source sentence → soft and dynamic^[5]

English-French

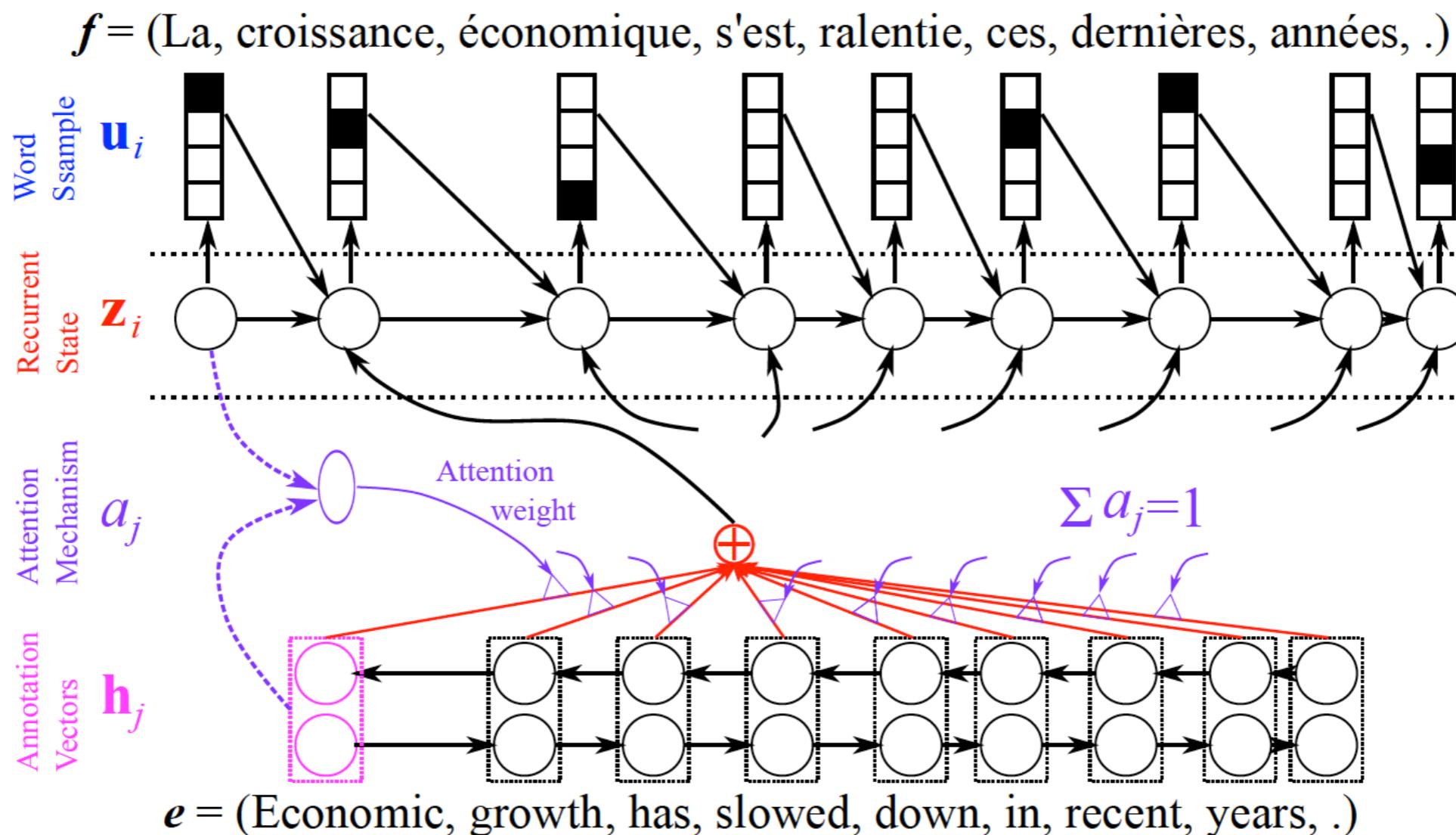


English-German



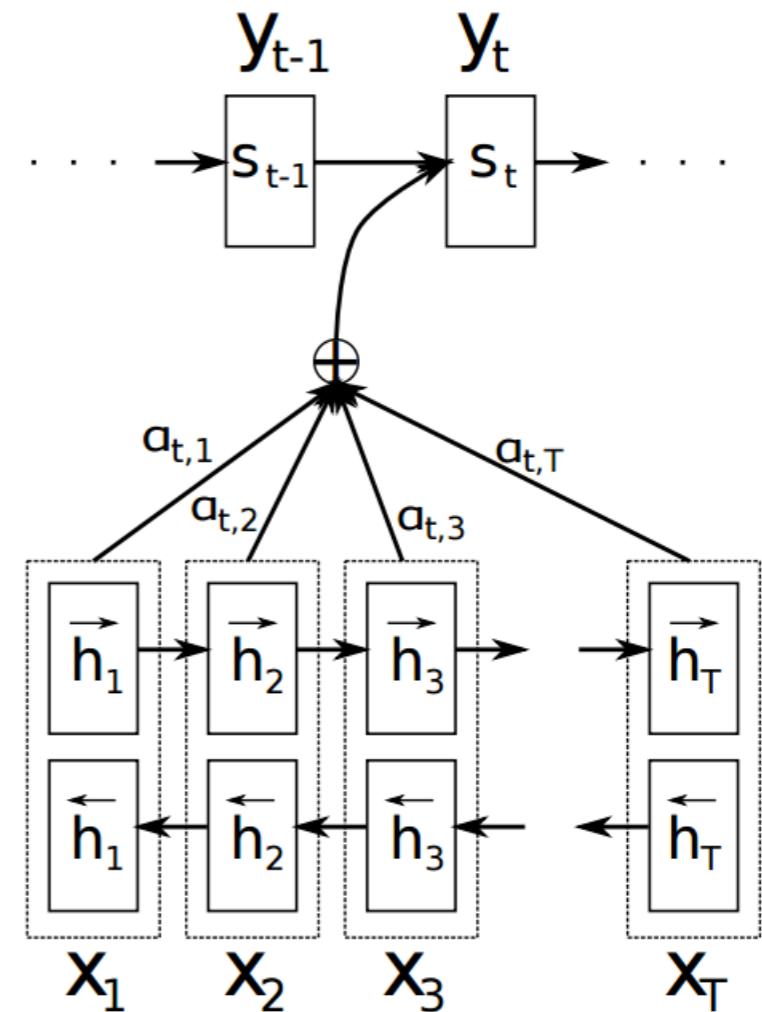
Attention Mechanism

- ♦ fixed representation of source sentence → soft and dynamic^[5]



Attention Mechanism

- ◆ At every generation step t [5]
 - ◆ Score source h_j by
$$e_{tj} = \mathbf{v}^T \tanh(\mathbf{W} \cdot s_{t-1} + \mathbf{U} \cdot h_j)$$
$$\alpha_{tj} = \text{softmax}(e_{tj})$$
 - ◆ Take an expectation over sources
$$c_t = \sum_j \alpha_{tj} h_j$$
 - ◆ Everything is differentiable.
Back-prop end-to-end!



Homework

- ◆ Tutorial Reading:
 - ◆ Deep Learning For Chatbots:
 - ◆ <http://www.wildml.com/2016/04/deep-learning-for-chatbots-part-1-introduction>
 - ◆ <http://www.wildml.com/2016/07/deep-learning-for-chatbots-2-retrieval-based-model-tensorflow/>
- ◆ Code Reading:
 - ◆ Neural Dialog System: <https://github.com/shawnwun/NNDIAL>
 - ◆ A Neural Conversational Model: <https://github.com/Conchylicultor/DeepQA>
 - ◆ Attention-based Conversational Model: https://github.com/pbhatia243/Neural_Conversation_Models



Thanks for your attention!

Q&A

References

- [1] Pascanu, Razvan, Tomas Mikolov, and Yoshua Bengio. "On the difficulty of training recurrent neural networks." International Conference on Machine Learning. 2013.
- [2] Mikolov, Tomas, et al. "Recurrent neural network based language model." Interspeech. Vol. 2. 2010.
- [3] Colah. "Understanding LSTMs". <http://colah.github.io/posts/2015-08-Understanding-LSTMs/>
- [4] Sutskever, Ilya, Oriol Vinyals, and Quoc V. Le. "Sequence to sequence learning with neural networks." Advances in neural information processing systems. 2014.
- [5] Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio. "Neural machine translation by jointly learning to align and translate." arXiv preprint arXiv:1409.0473 (2014).
- [6] Vinyals, Oriol, and Quoc Le. "A neural conversational model." arXiv preprint arXiv:1506.05869 (2015).
- [7] Wen, Tsung-Hsien, et al. "A network-based end-to-end trainable task-oriented dialogue system." arXiv preprint arXiv:1604.04562 (2016).
- [8] Bowman, Samuel R., et al. "Generating sentences from a continuous space." arXiv preprint arXiv:1511.06349 (2015).
- [9] Hu, Baotian, et al. "Convolutional neural network architectures for matching natural language sentences." Advances in neural information processing systems. 2014.
- [10] Qiu, Xipeng, and Xuanjing Huang. "Convolutional Neural Tensor Network Architecture for Community-Based Question Answering." IJCAI. 2015.
- [11] Wu, Yu, et al. "Sequential Match Network: A New Architecture for Multi-turn Response Selection in Retrieval-based Chatbots." arXiv preprint arXiv:1612.01627 (2016).