Module 2

Neural Models for Machine Translation and Conversation Generation

Module 2 Overview

- Overview of conventional statistical MT
- Neural machine translation
- Neural conversation generation

Statistical machine translation (SMT)

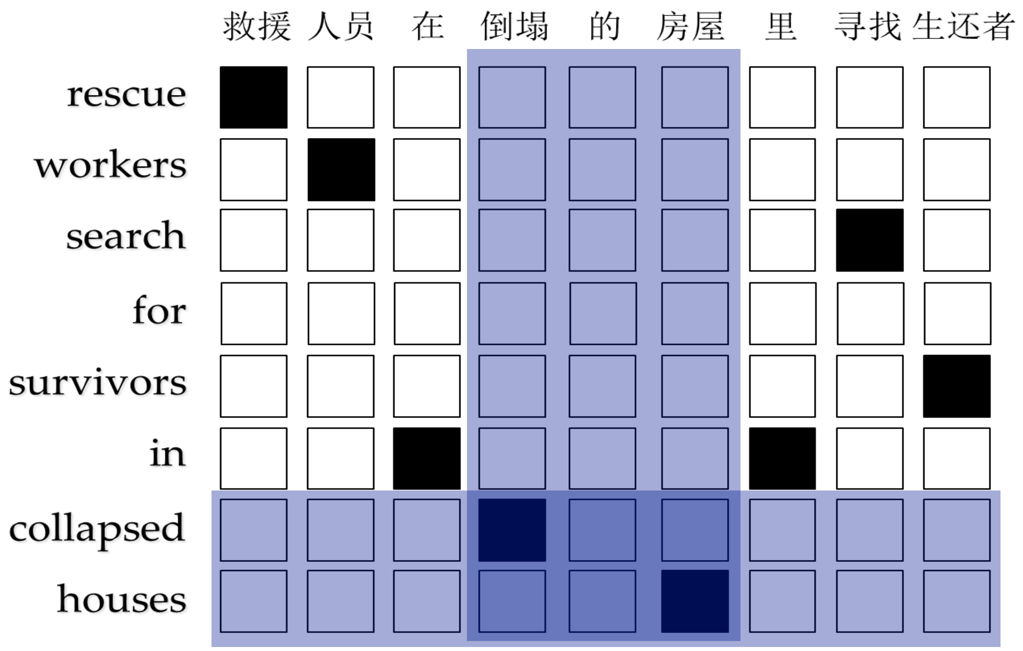
***S*:** 救援 人员 在 倒塌的 房屋 里 寻找 生还者
***T*:** Rescue workers search for survivors in collapsed houses

- Statistical decision: $T^* = \underset{T}{\operatorname{argmax}} P(T|S)$
- Source-channel model: $T^* = \underset{T}{\operatorname{argmax}} P(S|T)P(T)$
- Translation models: $P(S|T)$ and $P(T|S)$
- Language model: $P(T)$
- Log-linear model: $P(T|S) = \frac{1}{Z(S,T)} \exp \sum_i \lambda_i h_i(S,T)$
- Evaluation metric: BLEU score (higher is better)

Phrase-based SMT

救援人员在倒塌的房屋里寻找生还者 *Chinese*

Phrase translation modeling



(s, t)

(救援, rescue)
(人员, workers)
(在, in)
(倒塌, collapsed)
(房屋, house)
(里, in)
(寻找, search)
(生还者, survivors)
(救援 人员, rescue workers)
(在 倒塌, in collapsed)
(倒塌 的, collapsed)
(的 房屋, house)
(寻找, search for)
(寻找 生还者, search for survivors)
(生还者, for survivors)
(倒塌 的 房屋, collapsed house)

MLE: $P(\boldsymbol{t}|\boldsymbol{s}) = \frac{N(\boldsymbol{s},\boldsymbol{t})}{\sum_{t'} N(\boldsymbol{s},\boldsymbol{t}')}$

Simple, but suffers the data sparseness problem

Neural machine translation

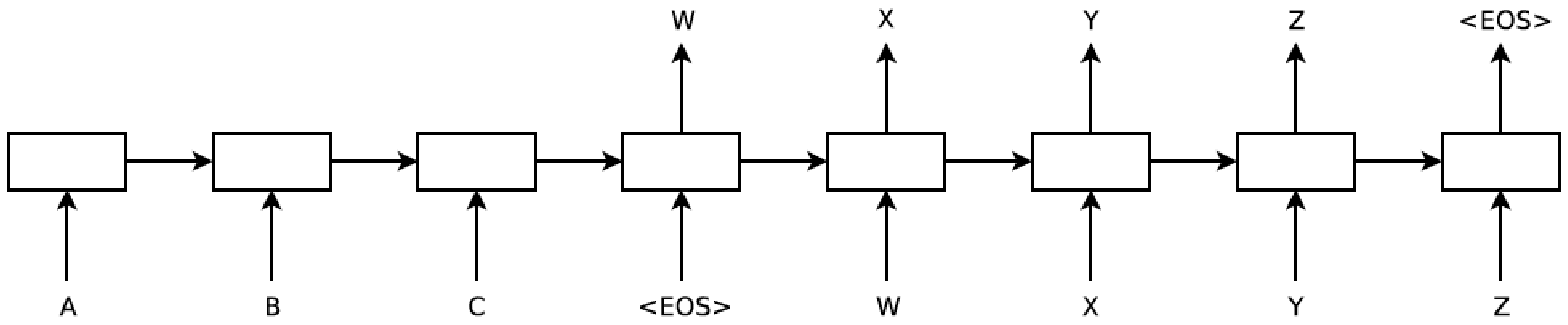
[Sutskever+ 14; Cho+ 14; Bahdanau+ 15]

- Build a single, large NN that reads a sentence and outputs a translation
  - Unlike phrase-based system that consists of many component models

- Encoder-decoder based approach
  - An encoder RNN reads and encodes a source sentence into a fixed-length vector
  - A decoder RNN outputs a variable-length translation from the encoded vector
  - Encoder-decoder RNNs are jointly learned on bitext, optimize target likelihood

Encoder-decoder model of [Sutskever+ 2014]

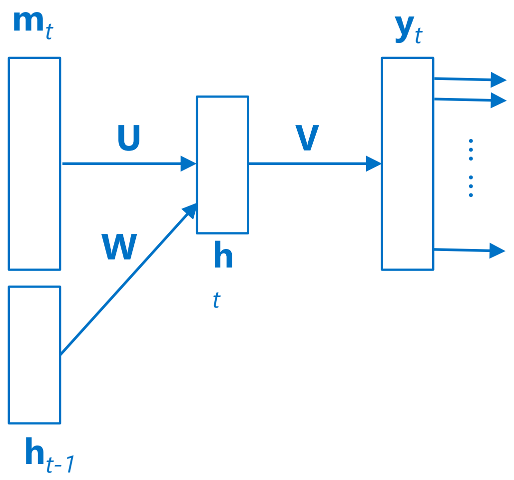
- "A B C" is source sentence; "W X Y Z" is target sentence

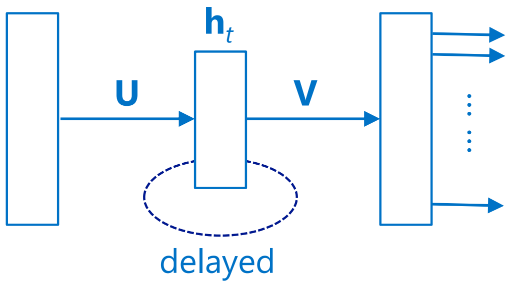


- Treat MT as general sequence-to-sequence transduction
  - Read source; accumulate hidden state; generate target
  - <EOS> token stops the recurrent process
  - In practice, read source sentence in reverse leads to better MT results
- Train on bitext; optimize target likelihood using SGD

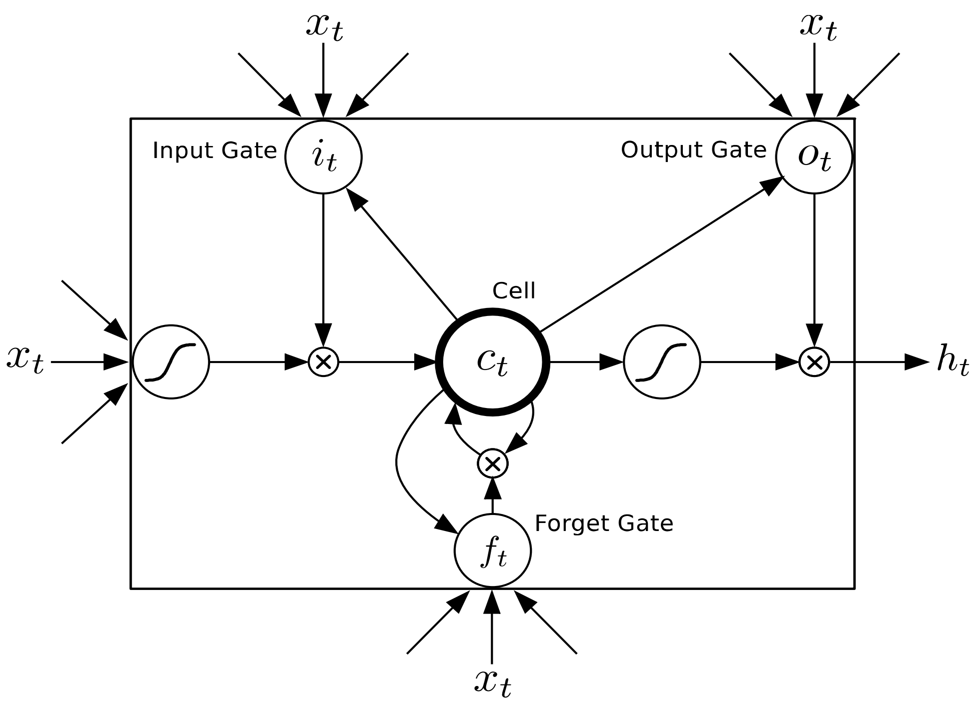
Potentials and difficulties of RNN

- In theory, RNN can "store" in $\boldsymbol{h}$ all information about past inputs
- But in practice, standard RNN cannot capture very long distance dependency
  - Vanishing/exploding gradient problem in backpropagation
  - Not robust to noise
- Solution: long short-term memory (LSTM)





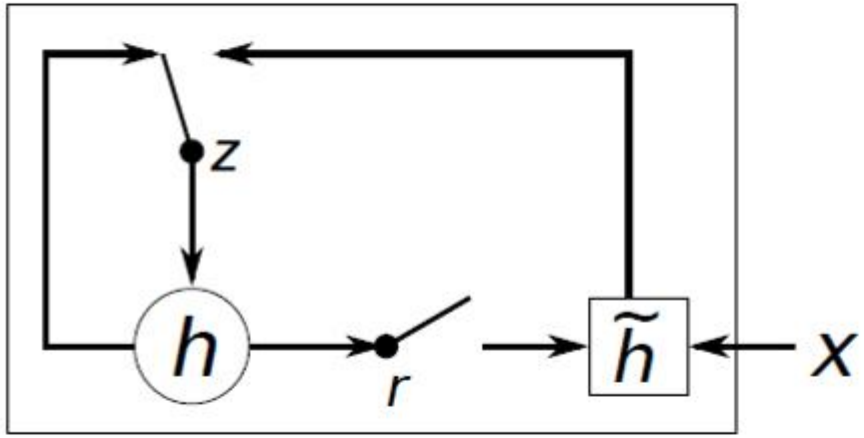
A long short-term memory cell
[Hochreiter & Schmidhuber 97; Graves+ 13]



$$i_t = \sigma\left(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i\right)$$
$$f_t = \sigma\left(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f\right)$$
$$c_t = f_t c_{t-1} + i_t \tanh\left(W_{xc}x_t + W_{hc}h_{t-1} + b_c\right)$$
$$o_t = \sigma\left(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o\right)$$
$$h_t = o_t \tanh(c_t)$$

Information flow in an LSTM unit of the RNN, with both diagrammatic and mathematical descriptions. W's are weight matrices, not shown but can easily be inferred in the diagram (Graves et al., 2013).

A 2-gate memory cell [Cho+ 14]



An illustration of the proposed hidden activation function. The update gate z selects whether the hidden state is to be updated with a new hidden state $\tilde{h}$. The reset gate r decides whether the previous hidden state is ignored. See

$$r_j = \sigma\left(\left[\mathbf{W}_r\mathbf{x}\right]_j + \left[\mathbf{U}_r\mathbf{h}_{\langle t-1\rangle}\right]_j\right)$$

$$z_j = \sigma\left(\left[\mathbf{W}_z\mathbf{x}\right]_j + \left[\mathbf{U}_z\mathbf{h}_{\langle t-1\rangle}\right]_j\right)$$

$$\tilde{h}_j^{\langle t\rangle} = \phi\left(\left[\mathbf{W}\mathbf{x}\right]_j + \left[\mathbf{U}\left(\mathbf{r}\odot\mathbf{h}_{\langle t-1\rangle}\right)\right]_j\right)$$

$$h_j^{\langle t\rangle} = z_j h_j^{\langle t-1\rangle} + (1 - z_j)\tilde{h}_j^{\langle t\rangle}$$

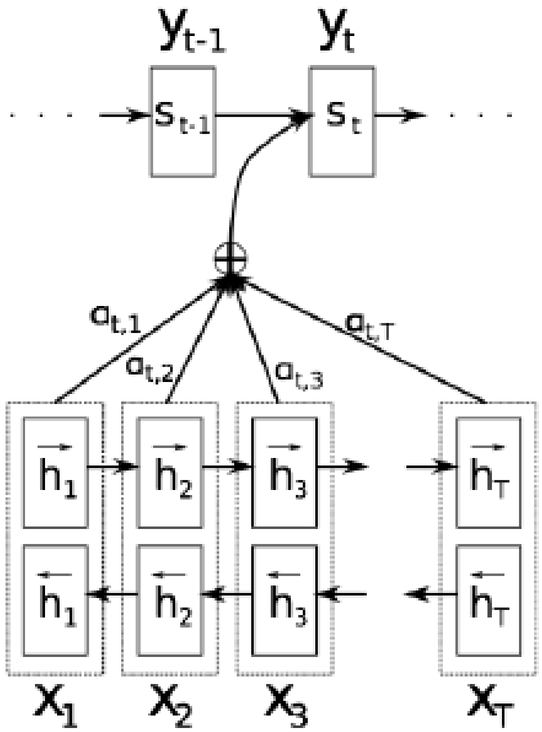
Joint learning to align and translate

- Issue with encoder-decoder model for SMT
  - Compressing a source sentence into a fixed-length vector makes it difficult for RNN to cope with long sentences.

- Attention model of [Bahdanan+ 15]
  - Encodes the input sentence into a sequence of vectors and choose a subset of these vectors adaptively while decoding

Attention model of [Bahdanan+ 15]

- Encoder:
  - bidirectional RNN to encode each word and its context
- Decoder:
  - Searches for a set of source words that are most relevant to the target word to be predicted.
  - Predicts a target word based on the context vectors associated with these source words and all the previous generated target words.
- Close to state-of-the-art performance
  - **Better at translating long sentences**



Neural Conversation

[Sordoni+ 15; Li+ NAACL16; Li+ ACL16]

Neural model learns the general shape of conversation, and the system output is situationally appropriate and coherent…

H: hi
M: how are you ?
H: not bad
M: what 's wrong ?
H: nothing really
M: wan na talk about it ? i 'm here if you wan na talk x
H: well , i 'm just tired
M: me too , i 'm here if you wan na talk
H: thanks , what are you doing ?
M: you 're welcome and i 'm watching tv and you ?
H: on netflix ?
M: yeah , i 'm watching it on netflix .
H: what are you watching ?
M: a tv show .
H: nice , did you have dinner already ?
M: no , not yet . what about you ?

A more goal-oriented example. No grounding into a real calendar, but the "shape" of the conversation is fluent and plausible…

H: would thursday afternoon work sometime ?
M: yeah , sure . just let me know when you 're free .
H: after lunch is probably the best time
M: okay , sounds good . just let me know when you 're free .
H: would 2 pm work for you ?
M: works for me .
H: well let 's say 2 pm then i 'll see you there
M: sounds good .

- Generating responses vs. retrieving responses
- Easy to incorporate contextual info via embedding
    - User profile – personalized conversation
    - knowledge bases – grounded conversation
- The engine is E2E learned from conversation experience
    - Learning a goal-oriented conversation engine via RL

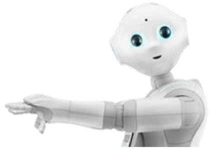
Neural Response Generation: The Blandness Problem

How was your weekend?

I don't know.

What did you do?

I don't understand what you are talking about.



This is getting boring…

Yes that's what I'm saying.

Blandness problem: cause and remedies

[Li et al., NAACL 2016]

Common ML objective (maximum likelihood)

(whatever the user says) $\xrightarrow{p(\text{target}|\text{source})}$

I don't know.

I don't understand...

That's what I'm saying

Mutual information objective:

(whatever the user says) $\xrightarrow{p(\text{target}|\text{source})}$ I don't know.

(whatever the user says) $\xleftarrow[p(\text{source}|\text{target})]{}$ I don't know.

Beyond blandness: Examples

Wow sour starbursts really do make **your mouth water**... mm drool.
Can I have one?

Of course you can! They're **delicious**!

Milan apparently **selling Zlatan** to balance the books... **Where next**, Madrid?

I think he'd be a **good signing**.

'tis a fine **brew** on a day like this! Strong though, **how many** is sensible?

Depends on how much you **drink**!

Well he was on in Bromley a while ago... **still touring**.

I've never **seen him live**.