

Transaction Bot using LSTM

Vineet Kumar, PhD

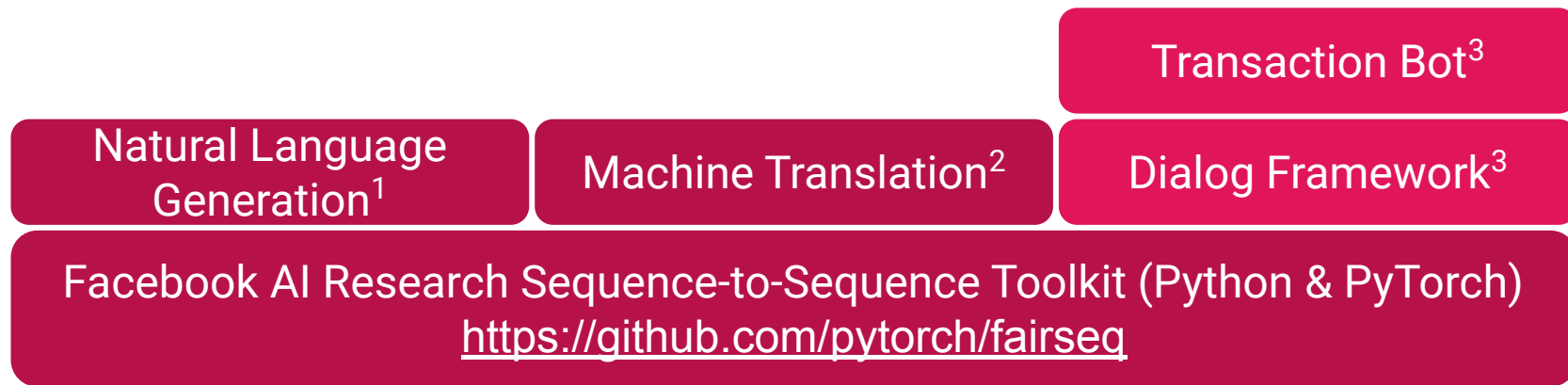
Contact: Email me through my Meetup Profile (private message, not to the group)

- Slides:** https://vineetk1.github.io/LSTM-based_Transaction_Bot.pdf
- Implementation:** <https://github.com/vineetk1/fairseq/tree/dialog>
- Datasets:** <https://fb-public.app.box.com/s/chnq60iivzv5uckpvj2n2vijlyepze6w>

Agenda

1. Open Source Implementation
2. Problem
3. A Solution (high-level view)
4. Word Embeddings
5. Encoder
6. Decoder
7. Attention
8. Beam Search
9. Results (Dataset: One dialog that passed; One dialog that failed)
10. Improvements
11. Relevant Publications

Open Source Implementation



Implementation of Transaction Bot: <https://github.com/vineetk1/fairseq/tree/dialog>

1. Predict the next word in a sequence of words
2. Given a source sequence (e.g. English), predict a target sequence (e.g. French)
3. Given a source sequence plus all previous sequences, predict a target sequence

Problem

- Build Bot using AI (NOT rule-based, slot-filling, etc.)
 - Makes reservation for table at restaurant
- Modified version of Dialog State Tracking Challenge 2 (DSTC 2)
 - 1000 dialogs for training, 1000 for validation and 1000 for testing
 - Vocabulary: 952 tokens
 - Dialog: Client and Customer Service Agent

Restaurant Reservation System

Client

Customer Service Agent

Hi!

Hello, what can I help you with today?

I'd like to book a table for six people in an expensive price range with British food.

I'm on it!

Where should it be?

In London.

Ok, let me look into some options for you.

```
api_call(British, London, Six, Expensive)
```

Actually i would prefer for four.

Sure. Is there anything else to update?

No.

Ok let me look into some options for you.

```
api_call(British, London, Four, Expensive)
```

```
* The_Place R_phone The_Place_phone
* The_Place R_cuisine british
* The_Place R_address The_Place_address
* The_Place R_location london
* The_Place R_availability four
* The_Place R_price expensive
* The_Place R_rating 7
* The_Fancy_Pub R_phone The_Fancy_Pub_phone
* The_Fancy_Pub R_cuisine british
* The_Fancy_Pub R_address The_Fancy_Pub_address
* The_Fancy_Pub R_location london
* The_Fancy_Pub R_availability four
* The_Fancy_Pub R_price expensive
* The_Fancy_Pub R_rating 8
```

What do you think of this option: The_Fancy_Pub

No, I don't like that.

Sure, let me find an other option for you.

What do you think of this option: The_Place

Let's do it!

Great let me do the reservation.

Can you provide me the address?

Here it is: The_Place_address.

You rock!

Is there anything else I can help you with?

No thanks.

You're welcome.

Task 1

Issuing API calls

Task 2

Updating API calls

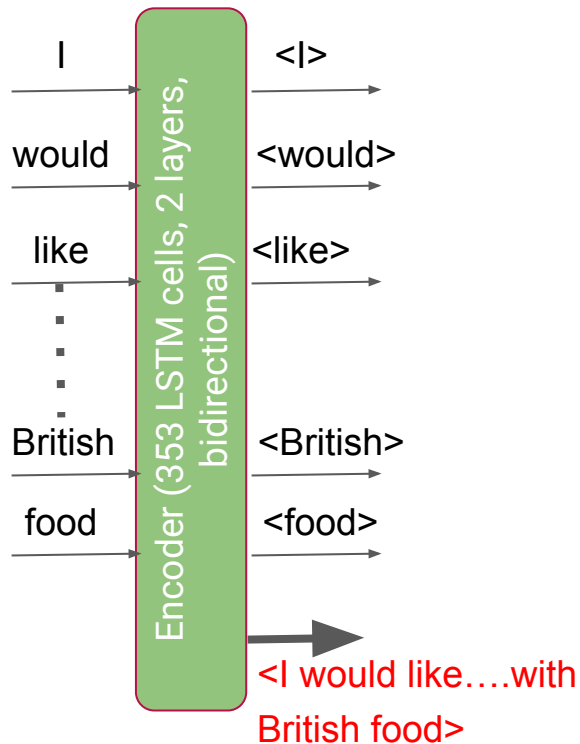
Task 3

Displaying options

Task 4

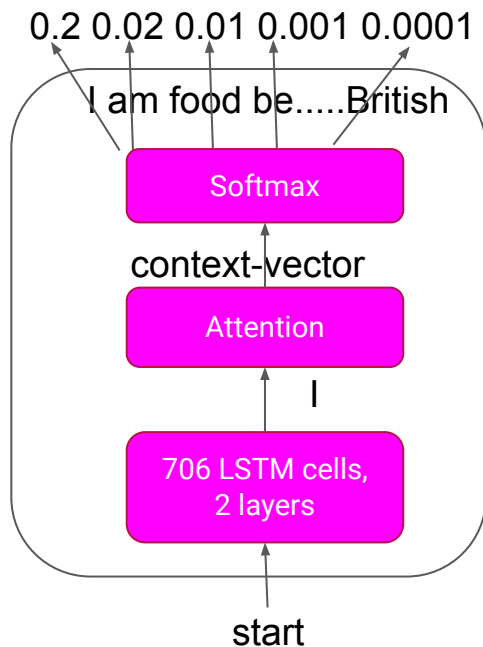
Providing extra-information

A Solution (high-level view)



Encoder

I am on it ! Where should it be ?



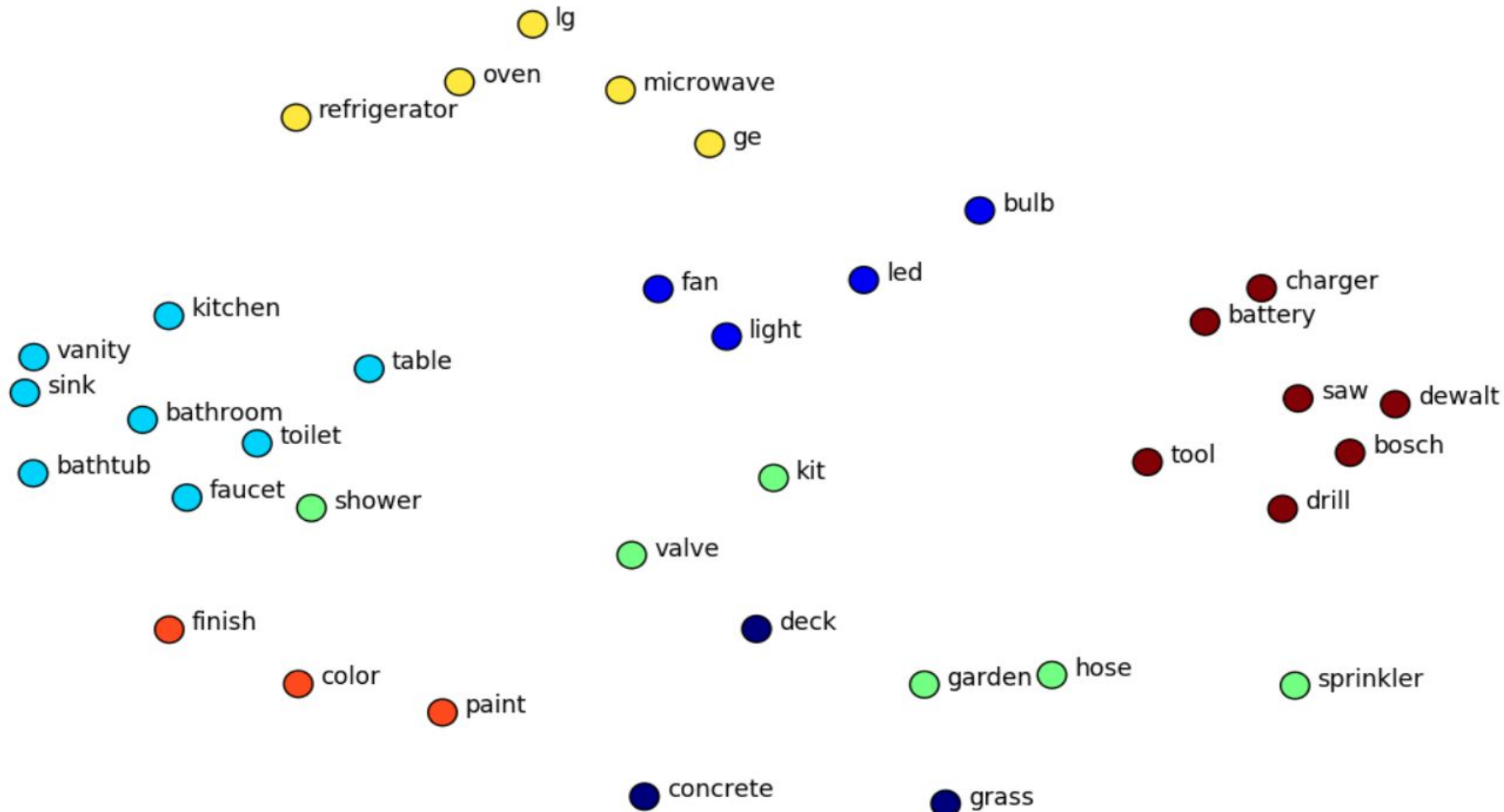
Decoder with Attention

1. Forward propagation
2. Calculate Loss (e.g. using "label smoothed cross entropy" function)
3. Backward propagation for gradients
4. Update parameters using Optimizer (e.g. nag)

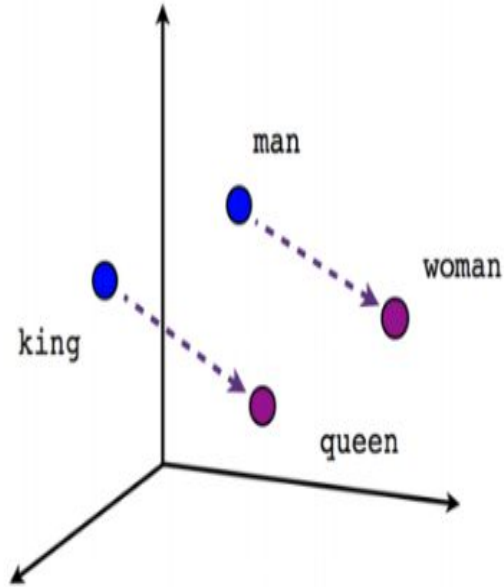
Word Embeddings; word->vector

- Convert words to vectors
- Vectors: 50-d to 1000-d and more
- Download pre-trained vectors; use as look-up table
- Algorithms to generate vectors: word2vec, glove
 - Unsupervised learning
 - Distributional Similarity

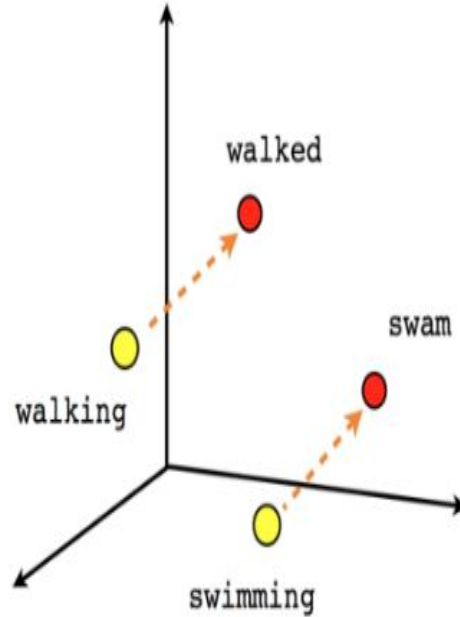
Distributional Similarity -- words that frequently occur together in text are similar



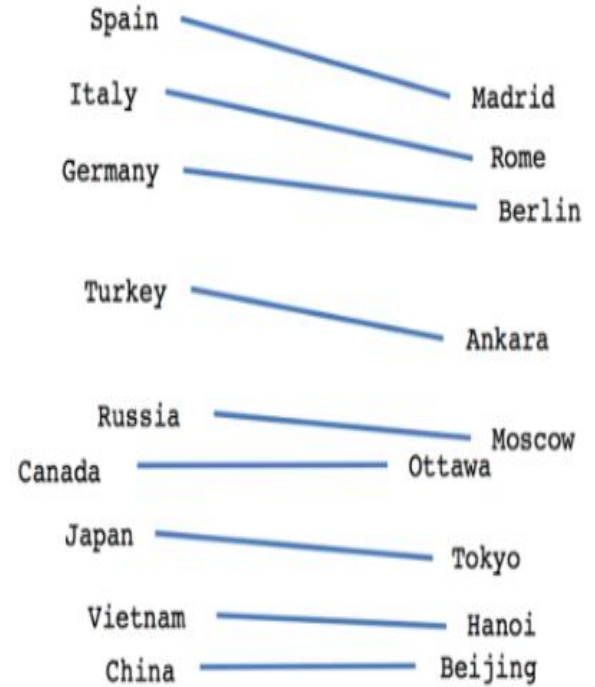
Some features/properties of vectors



Male-Female



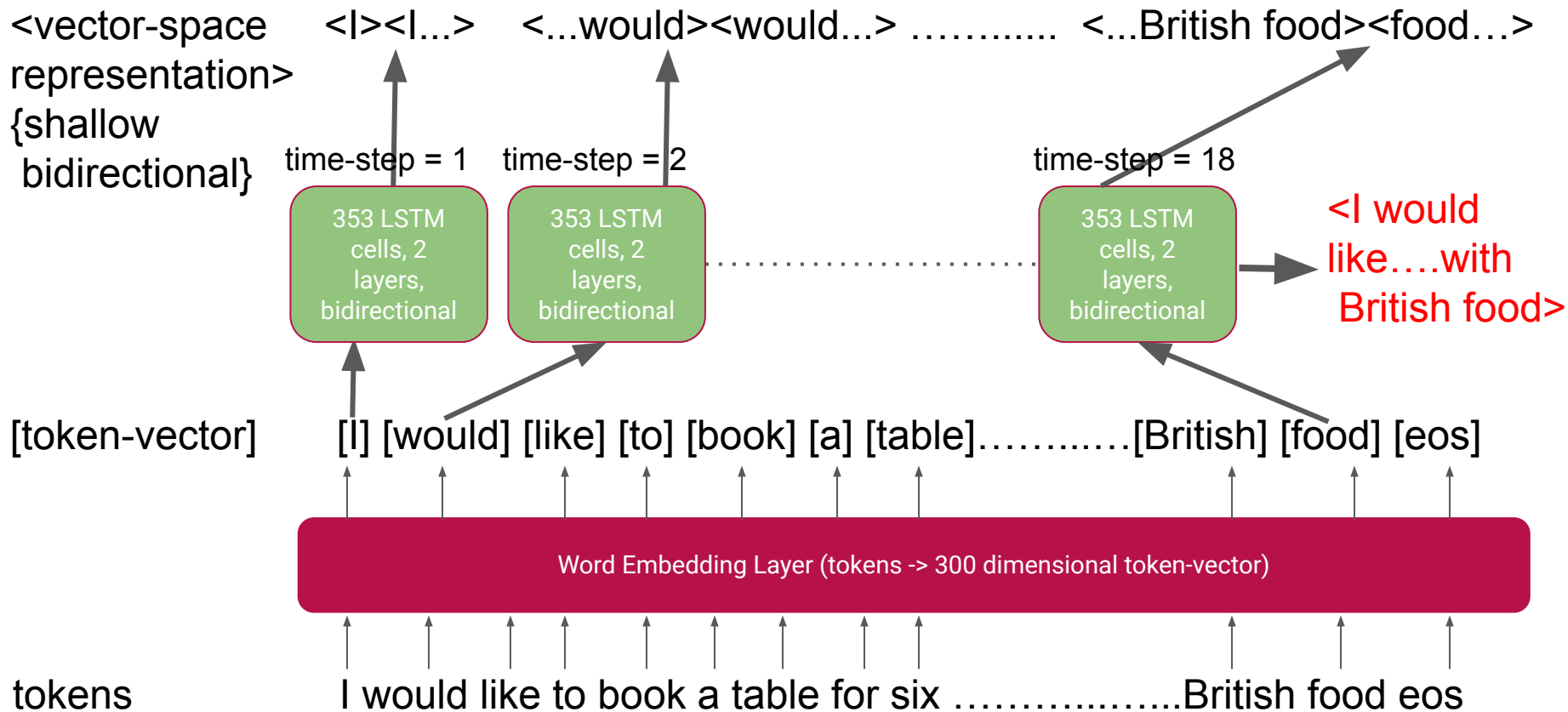
Verb tense



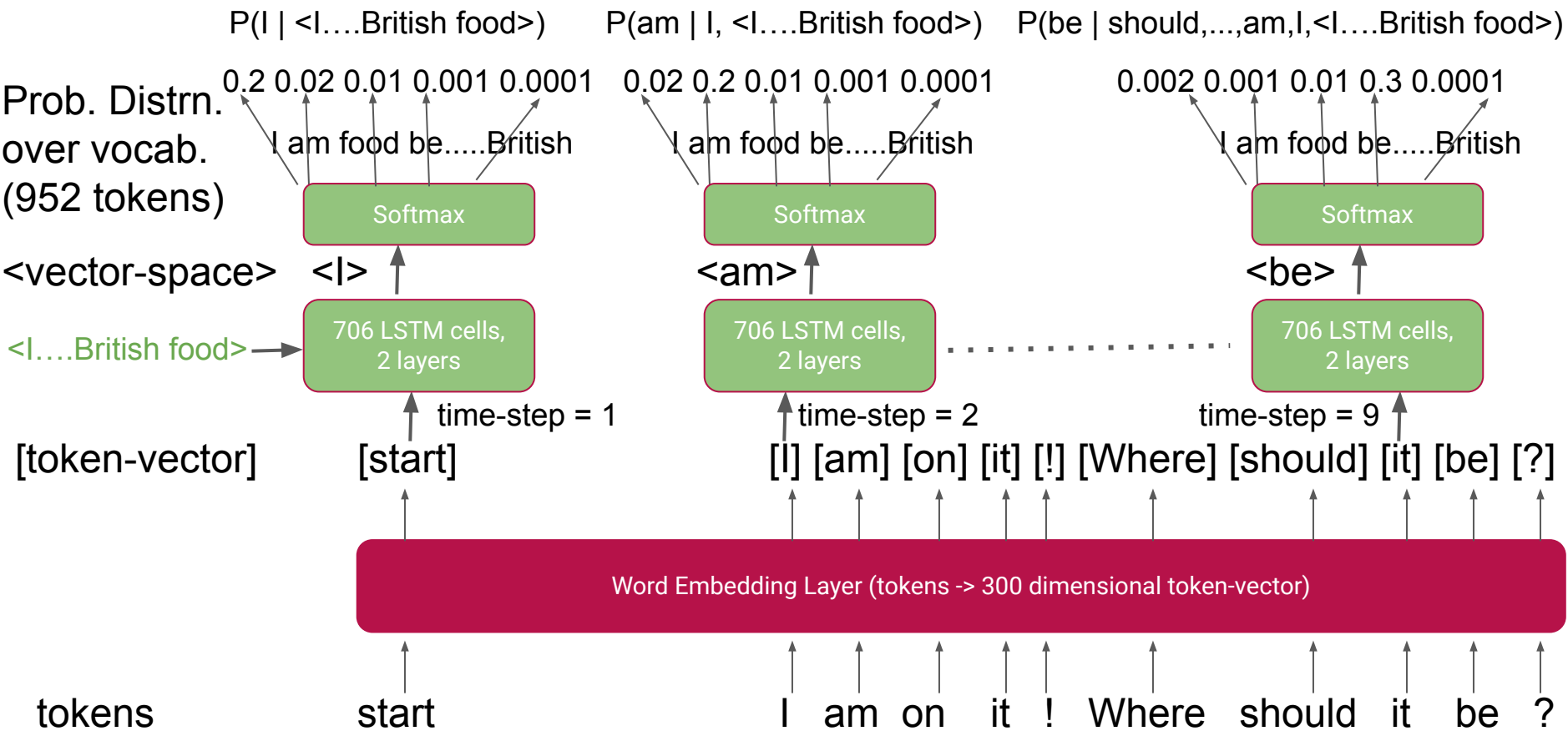
Country-Capital

Encoder (Training)

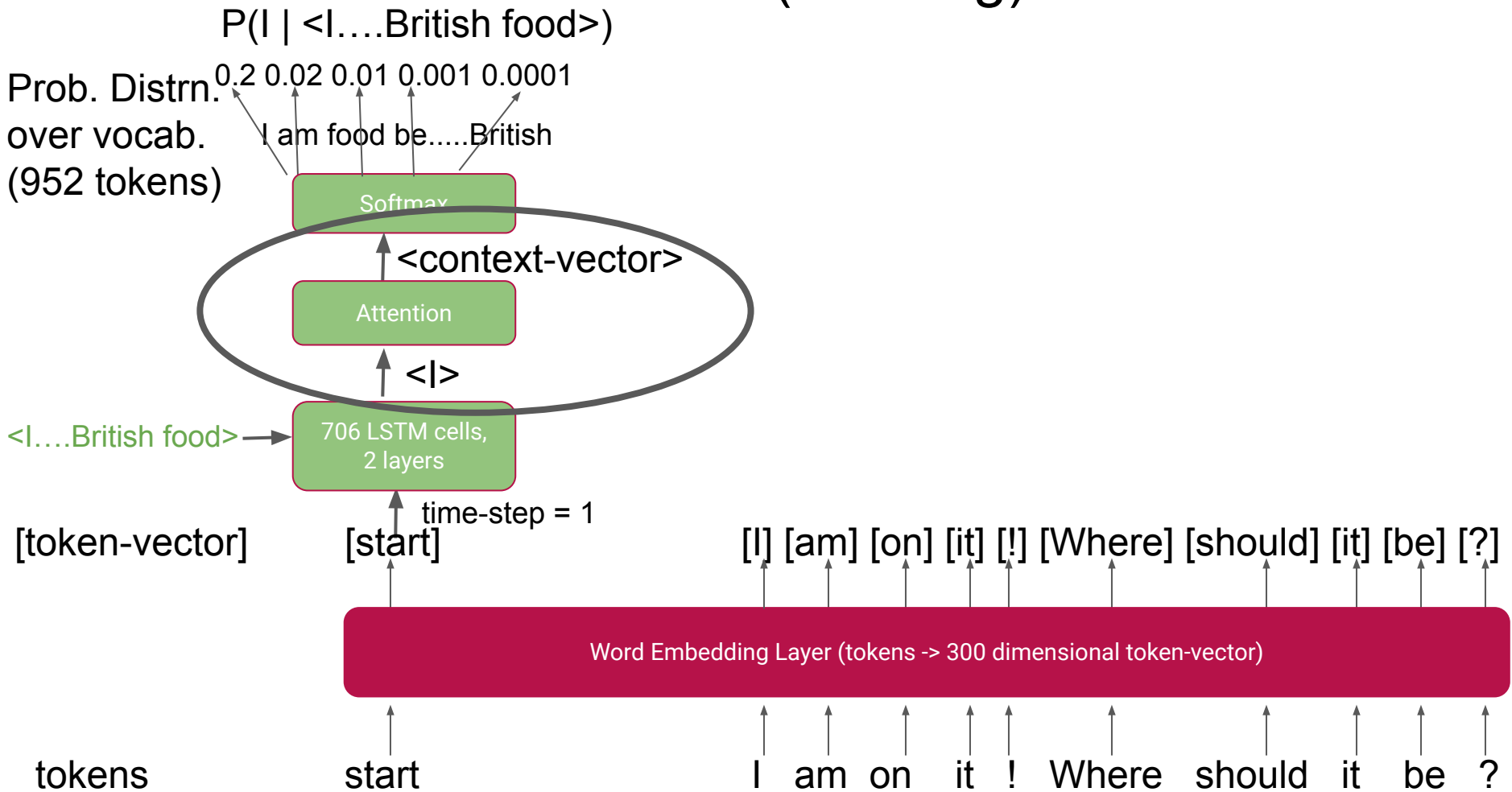
Dialog State
<vector-space
representation>
{shallow
bidirectional}



Decoder (Training)



Attention (Training)



Luong's Attention (Training)

Context Vector $\tanh(W(\langle \text{attention vector} \rangle \langle I \rangle))$

Attn. Vector $0.5(\langle I \rangle \langle I \dots \rangle) + 0.2(\langle \dots \text{would} \rangle \langle \text{would} \dots \rangle) + \dots + 0.04(\langle \dots \text{British} \dots \rangle)$

Normalized Attn. Score 0.5 0.2 0.04

Attn. Score 500 212 30

Dot Product $(\langle I \rangle \langle I \dots \rangle) \cdot \langle I \rangle$ $(\langle \dots \text{would} \rangle \langle \text{would} \dots \rangle) \cdot \langle I \rangle$ $(\langle \dots \text{British} \dots \rangle) \cdot \langle I \rangle$

Dialog State $\langle I \rangle \langle I \dots \rangle$ $\langle \dots \text{would} \rangle \langle \text{would} \dots \rangle$ $\langle \dots \text{British food} \rangle \langle \text{food} \dots \rangle$

Vinyal's Attention (Training)

Context Vector $\tanh(W(\langle \text{attention vector} \rangle \langle I \rangle))$

Attn. Vector $0.5(\langle I \text{ would} \rangle \cdot \langle I \rangle) + 0.2(\langle I \text{ would like} \rangle \cdot \langle I \rangle) + \dots + 0.04(\langle \text{..British food} \rangle \cdot \langle I \rangle)$

Normalized Attn. Score

0.5	0.2	0.04
-----	-----	-------	------

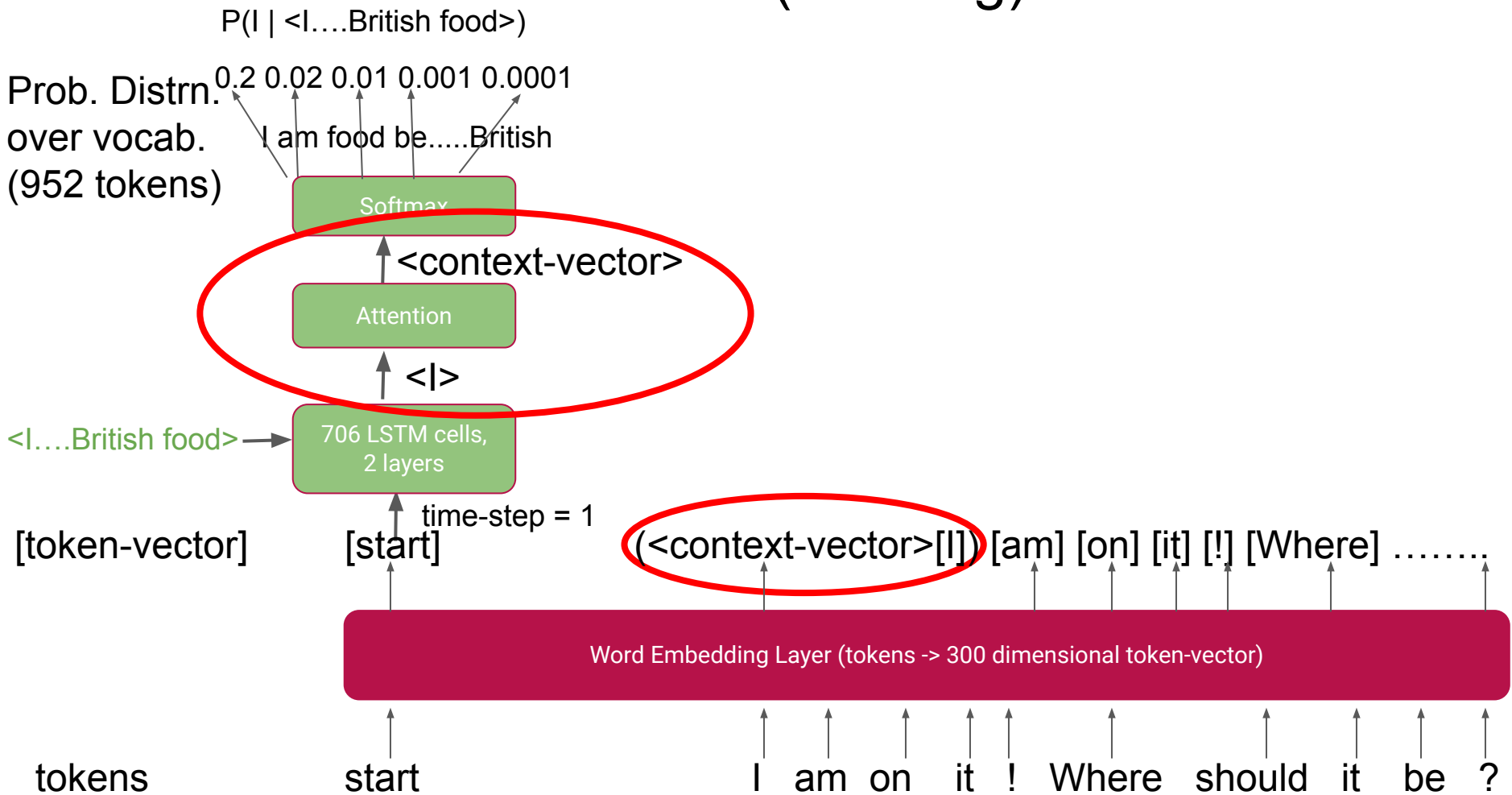
Vinyal's Attn. Score $v^T (\tanh(W(\langle I \rangle \langle I \dots \rangle) \langle I \rangle))$ where vector v is trainable parameter

~~Luong's Attn. Score~~

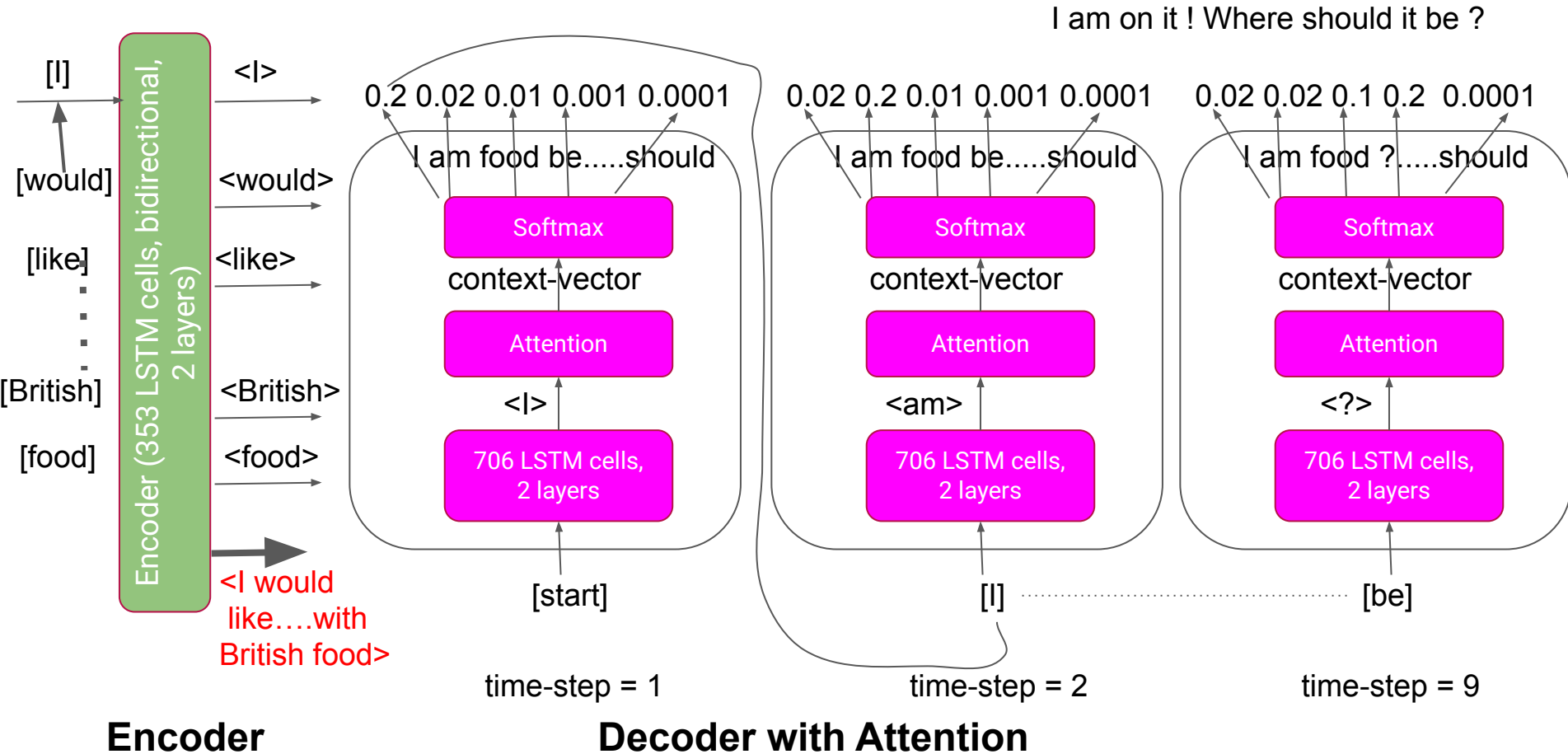
~~Dot Product~~ ~~..... ($\langle I \rangle \langle I \dots \rangle$) \cdot $\langle I \rangle$ ($\langle \dots \text{would} \rangle \langle \text{would} \dots \rangle$) \cdot $\langle I \rangle$ ($\langle \dots \text{British} \dots \rangle$) \cdot $\langle I \rangle$~~

Dialog State $\langle I \rangle \langle I \dots \rangle$ $\langle \dots \text{would} \rangle \langle \text{would} \dots \rangle$ $\langle \dots \text{British food} \rangle \langle \text{food} \dots \rangle$

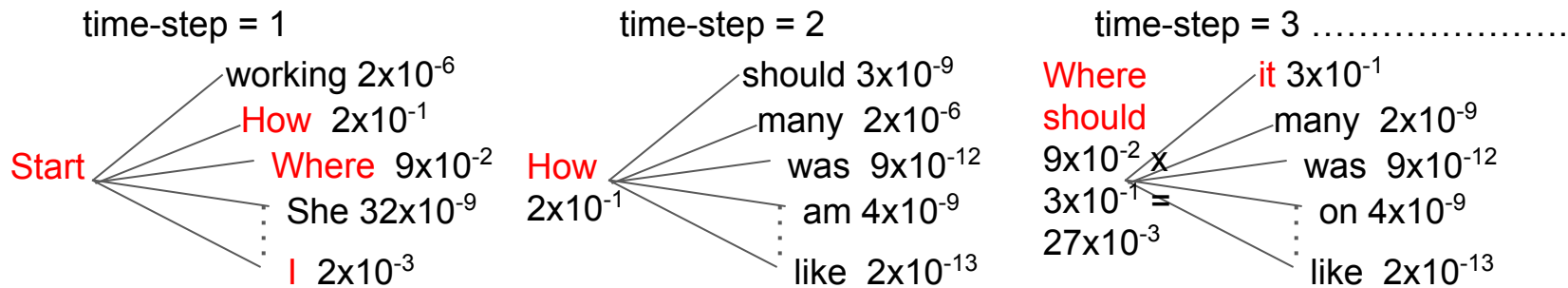
Attention (Training)



Inference



Beam Search (Inference); beam-width = 3



x = I would like .. British food

time-step = 1

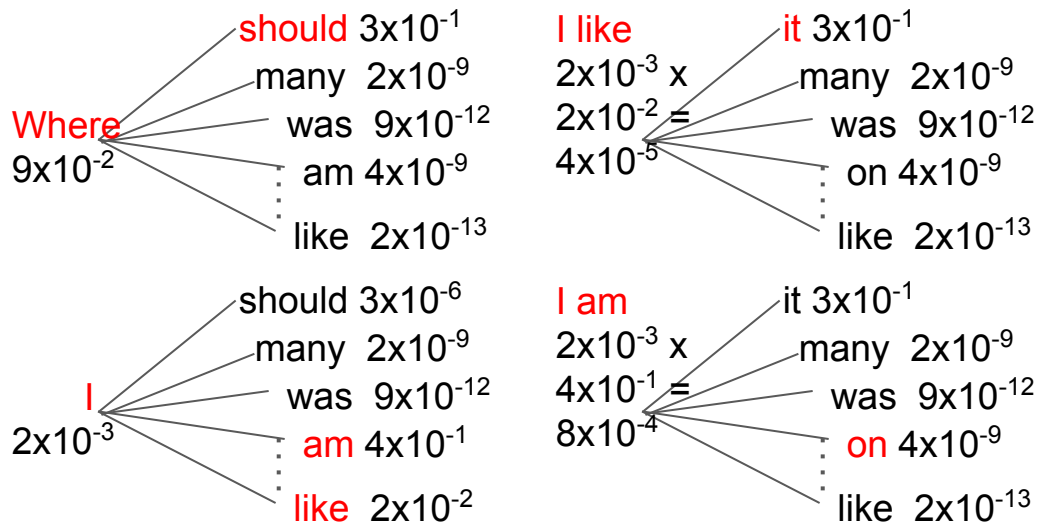
$P(y^1 | x)$

time-step = 2

$(\arg \max_y) P(y^1, y^2 | x) = P(y^1 | x) \cdot P(y^2 | x, y^1)$

time-step = 3

$(\arg \max_y) P(y^1, y^2, y^3 | x) = P(y^1 | x) \cdot P(y^2 | x, y^1) \cdot P(y^3 | x, y^1, y^2)$



Vocabulary of 952 tokens; if each sequence is 10 words long then 952^{10} combination of sequences:

How many people are in your party?

I am on it! Where should it be?

Where should it be?

I like it! Where should it be?

Dataset

- Real human-machine dialogs vs. Synthetic
 - Bots do very well on Synthetic datasets
 - Real datasets are a challenge
- Dialog State Tracking Challenge 2 (DSTC 2)
<https://fb-public.app.box.com/s/chnq60iivzv5uckpvj2n2vijlyepze6w>
- Noisy because of speech -> text
- Accuracy metric is unnecessarily stringent
 - F1 metric is better but not used

Result

- ** Translated 1117 dialogs with 11237 turns and 782520 tokens using beam=3 in 195.9s (5.70 dialogs/s, 57.36 turns/s, 3994.54 tokens/s)
- ** % number of dialogs that passed = $(18/1117 \times 100) = 1.61\%$
 - ** (# of turns in dialog: # of occurrences) = (5: 7), (6: 8), (7: 3)
- ** % number of turns that passed = $(4249/11237 \times 100) = 37.81\%$
- ** % number of dialogs that failed = $(1099/1117 \times 100) = 98.39\%$
 - ** (# of consecutive turns that passed, counting from beginning of dialog / # of turns in dialog: # of occurrences) = (2/4: 1), (1/5: 10), (3/5: 1), (1/6: 63), (2/6: 2), (3/6: 3), (1/7: 142), (2/7: 8), (3/7: 3), (4/7: 1), (5/7: 1), (1/8: 160), (2/8: 16), (3/8: 2), (1/9: 149), (2/9: 14), (1/10: 118), (2/10: 11), (3/10: 1), (1/11: 89), (2/11: 13), (3/11: 3), (5/11: 1), (1/12: 77), (2/12: 11), (1/13: 44), (2/13: 5), (3/13: 1), (1/14: 37), (2/14: 4), (3/14: 1), (1/15: 23), (2/15: 5), (3/15: 2), (1/16: 13), (2/16: 3), (3/16: 1), (1/17: 9), (2/17: 1), (1/18: 9), (2/18: 2), (1/19: 11), (1/20: 6), (2/20: 1), (3/20: 1), (1/21: 6), (1/22: 2), (2/22: 1), (1/23: 3), (1/24: 2), (1/25: 1), (1/26: 1), (1/27: 2), (1/28: 1), (1/29: 1)
- ** Generate test with beam=3: BLEU4 = 23.11, 25.5/22.9/22.1/22.1 (BP=1.000, ratio=2.070, syslen=283321, reflen=136863)

Dialog Passed

Tr1-S: <SILENCE>

Tr1-T: Hello , welcome to the Cambridge restaurant system . You can ask for restaurants by area , price range or food type . How may I help you ?

Tr1-H0-P: Hello , welcome to the Cambridge restaurant system . You can ask for restaurants by area , price range or food type . How may I help you ?

Tr2-S: cheap restaurant on the east part of town

{Price, Location}

Tr2-T: api_call R_cuisine east cheap

Tr2-H0-P: api_call R_cuisine east cheap

Tr3-S: <SILENCE>

Tr3-T: the_missing_sock is a nice place in the east of town and the prices are cheap

Tr3-H0-P: the_missing_sock is a nice place in the east of town and the prices are cheap

Tr4-S: what is the address

Tr4-T: Sure , the_missing_sock is on the_missing_sock_address

Tr4-H0-P: Sure , the_missing_sock is on the_missing_sock_address

Tr5-S: phone number

Tr5-T: The phone number of the_missing_sock is the_missing_sock_phone

Tr5-H0-P: The phone number of the_missing_sock is the_missing_sock_phone

Tr6-S: what type of food

{cuisine}

Tr6-T: the_missing_sock serves international food

Tr6-H0-P: the_missing_sock serves international food

Tr7-S: thank you good bye

Tr7-T: you are welcome

Tr7-H0-P: you are welcome

Entity Types (required):

Cuisine, Location, Price

Entity Types (other):

Post-code, Phone, Address

Dialog Failed

Tr1-S: <SILENCE>

Tr1-T: Hello , welcome to the Cambridge restaurant system . You can ask for restaurants by area , price range or food type . How may I help you ?

Tr1-H0-P: Hello , welcome to the Cambridge restaurant system . You can ask for restaurants by area , price range or food type . How may I help you ?

Tr2-S: expensive australian restaurant

{Price, Cuisine}

Tr2-T: You are looking for a australian restaurant right?

Tr2-H0-F: What part of town do you have in mind?

{Location?}

Tr3-S: yes

Tr3-T: I'm sorry but there is no restaurant serving australian food

Tr3-H0-F: What kind of food would you like?

Tr4-S: expensive french restaurant

{Price, Cuisine}

Tr4-T: You are looking for a australian restaurant right?

Tr4-H0-F: api_call french R_location expensive

Tr5-S: french

Tr5-T: There are restaurants serving french in the expensive price range .
What area would you like?

Tr5-H0-F: api_call french R_location expensive

Tr6-S: any

Tr6-T: api_call french R_location expensive

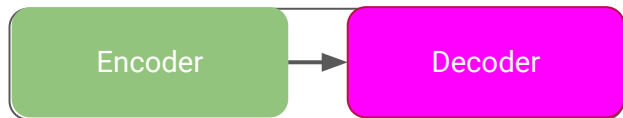
Tr6-H0-F: Would you like something in the cheap , moderate , or
expensive price range?

Entity Types (required):
Cuisine, Location, Price
Entity Types (other):
Post-code, Phone, Address

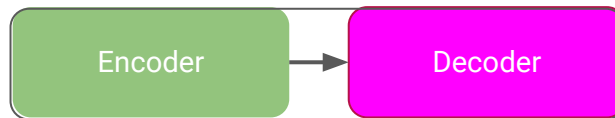
Improvements

- Entity-type (e.g. Cuisine, Location, Place) in input
- Attention-based copy mechanism
 - Pick output token from Dialog State or Decoder output
- Pre-training & Fine-tuning (e.g. BART)
- Modifications to architecture

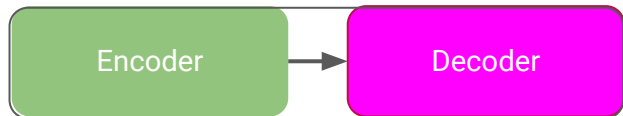
Pre-train & Fine-tune



Initialize Parameters with random weights and bias



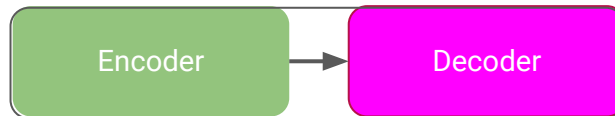
Train with Dialog dataset



Pre-train with relevant (e.g. Natural Language Generation) datasets

Pre-trained model: BART

Transfer the learning



Fine-tune with Dialog dataset

Publications

1. Bordes, A., Boureau, Y. L., & Weston, J. (2016). Learning end-to-end goal-oriented dialog. *arXiv preprint arXiv:1605.07683*. <https://arxiv.org/pdf/1605.07683.pdf>
2. Eric, M., & Manning, C. D. (2017). A copy-augmented sequence-to-sequence architecture gives good performance on task-oriented dialogue. *arXiv preprint arXiv:1701.04024*. <https://arxiv.org/pdf/1701.04024.pdf>
3. Vinyals, O., Kaiser, Ł., Koo, T., Petrov, S., Sutskever, I., & Hinton, G. (2015). Grammar as a foreign language. In *Advances in neural information processing systems* (pp. 2773-2781). <https://papers.nips.cc/paper/5635-grammar-as-a-foreign-language.pdf>
4. Luong, M. T., Pham, H., & Manning, C. D. (2015). Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*. <https://arxiv.org/pdf/1508.04025.pdf>
5. Bahdanau, D., & Cho, K. (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv: 1409.0473*. <https://arxiv.org/pdf/1409.0473.pdf>
6. Jia, R., & Liang, P. (2016). Data recombination for neural semantic parsing. *arXiv preprint arXiv:1606.03622*. <https://www.aclweb.org/anthology/P16-1002.pdf>
7. Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., ... & Zettlemoyer, L. (2019). Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*. <https://arxiv.org/pdf/1910.13461.pdf>