MA-INF 4222 - Lab Natural Language Processing
SS 2018

Key-Value Retrieval Networks for Task-Oriented Dialogue

Debayan Banerjee - 3034800

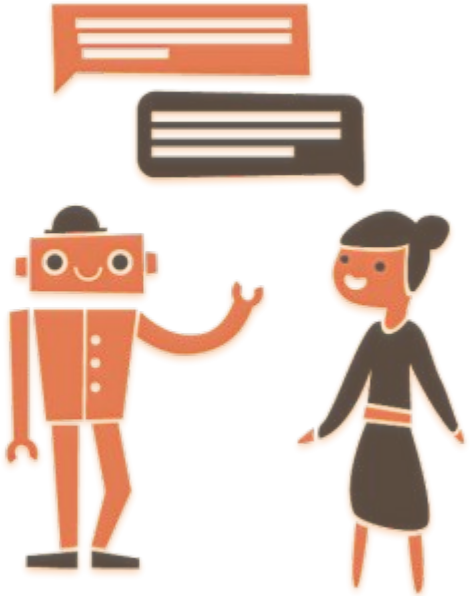
Pooja Bhatia- 3143760

Jing-Long Wu- 3045999

University of Bonn

Computer Science Master Program

Topic: Dialogue and Interactive Systems



- 2000’s: Websites
- 2010’s: Apps
- 2020’s: Dialogue Systems / Bots

Key-Value Retrieval Networks for Task-Oriented Dialogue

Details:

Paper published by: Stanford NLP Group and Ford Research and Innovation Center
Latest version published on: 14 July 2017

Objective:

To address the problem: Neural task-oriented dialogue systems often struggle to smoothly interface with a knowledge base.

Proposed Solution:

Paper proposes a new neural dialogue agent that is able to effectively sustain grounded, multi-domain discourse through a novel **Key-Value Retrieval Mechanism**.

Key-Value Retrieval Networks for Task-Oriented Dialogue

Experiment:

The study architecture is trained on data from following domains: Calendar Scheduling, Weather Information , point-of-interest (POI) navigation.

Outperforms competitive rule based system and other existing neural dialogue architectures on the provided domains according to both automatic and human evaluation metrics.

Link: https://arxiv.org/pdf/1705.05414.pdf

Sample Dialogue of POI

POI	Category	Address	Traffic Info	Distance
Dish Parking	parking garage	550 Alester Ave	no traffic	3 miles
Cafe Venetia	coffee or tea place	329 El Camino Real	road block nearby	4 miles
Mandarin Roots	chinese restaurant	271 Springer St	car collision nearby	3 miles
Dominos	pizza restaurant	776 Arastradero Rd	moderate traffic	6 miles
Travelers Lodge	rest stop	333 Arbol Dr	moderate traffic	4 miles

DRIVER: What is the nearest parking garage within 4 miles that has no traffic.

CAR: *the nearest is dish parking at 550 alester ave, 3 miles away*

DRIVER: Does it have any traffic?

CAR: *there is no traffic on the route to dish parking*

DRIVER: Thanks

CAR: *you're welcome!*

Sample Dialogue of Calendar Scheduling

Event	Time	Date	Party	Agenda	Location
dinner	5pm	the 4th	brother	-	-
meeting	1pm	the 18th	vice president	discuss company picnic	room 100
conference	11am	the 1st	HR	go over budget	room 50
meeting	4pm	the 9th	HR	onboard new members	room 100
lab apps.	6pm	the 14th the	Jon	-	-
dinner	6pm	1st	Jon		-

DRIVER: When is my dinner going to be and who will be attending

CAR: *your next dinner is on the 1st at 6 pm with your jon*

DRIVER: Great, thanks.

CAR: *you're welcome!*

Sample Dialogue of Weather Information

	monday	tuesday	wednesday	thursday	friday	...
los angeles	hail, 60-70f	foggy. 30-40f	cloudy. 20-30f	stormy 30-40f	drizzle. 30-50f	
durham	rain. 80-90f	cloudy, 50-70f	misty, 40-60f	dry. 80-100f	rain. 50-70f	...
seattle	warm, 60-70f	misty, 40-60f	foggy, 40-50f	foggy. 40-50f	clear skies 20-30f	...

DRIVER: how is the weather

CAR: *what city would you like the weather for?*

DRIVER: is it foggy in seattle?

CAR: *it will not be foggy in seattle today*

DRIVER: thanks

CAR: *you're welcome!*

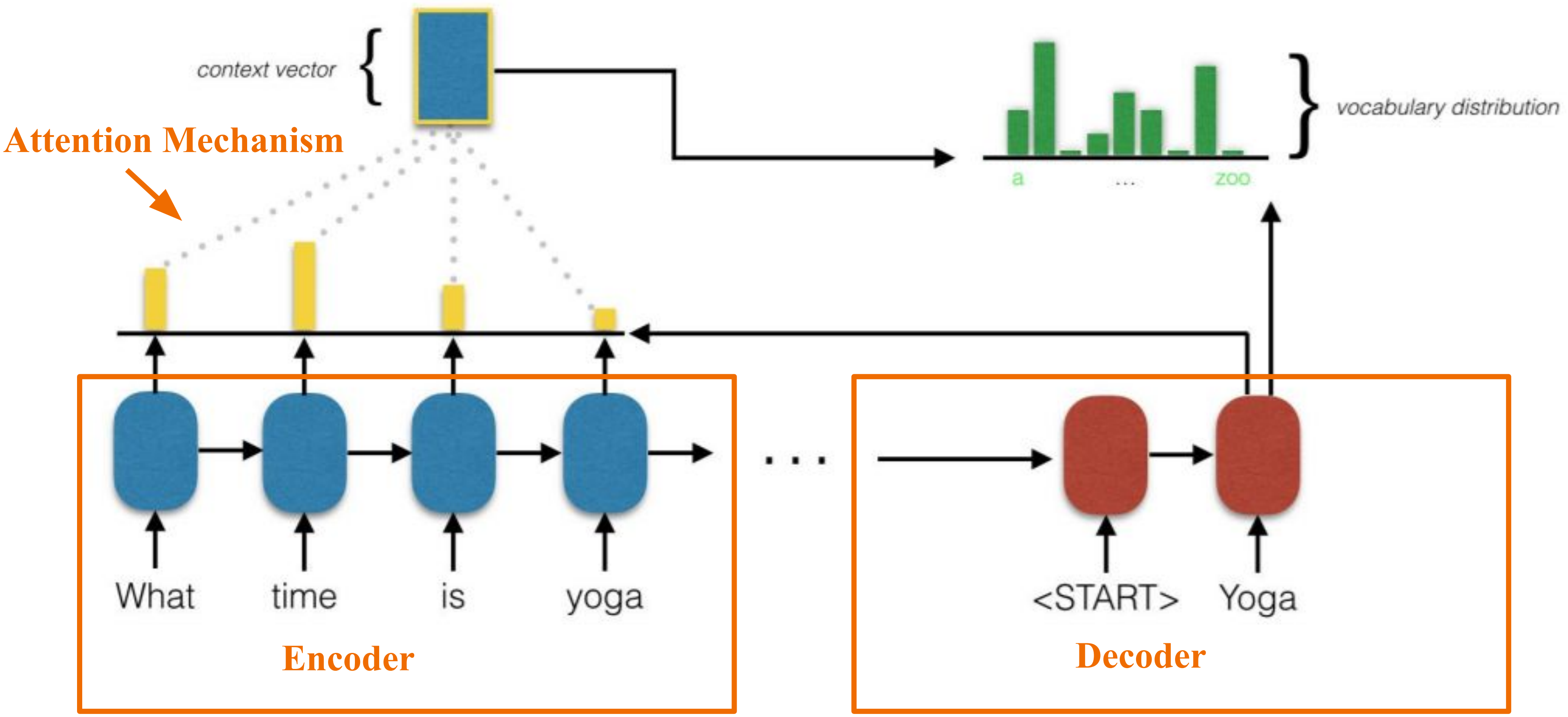
Steps: Convert Row-Based KB Entry into Key-Value Representation

Event	Time	Date	Party	Agenda
dinner	8:00 PM	the 13th	Ana	-

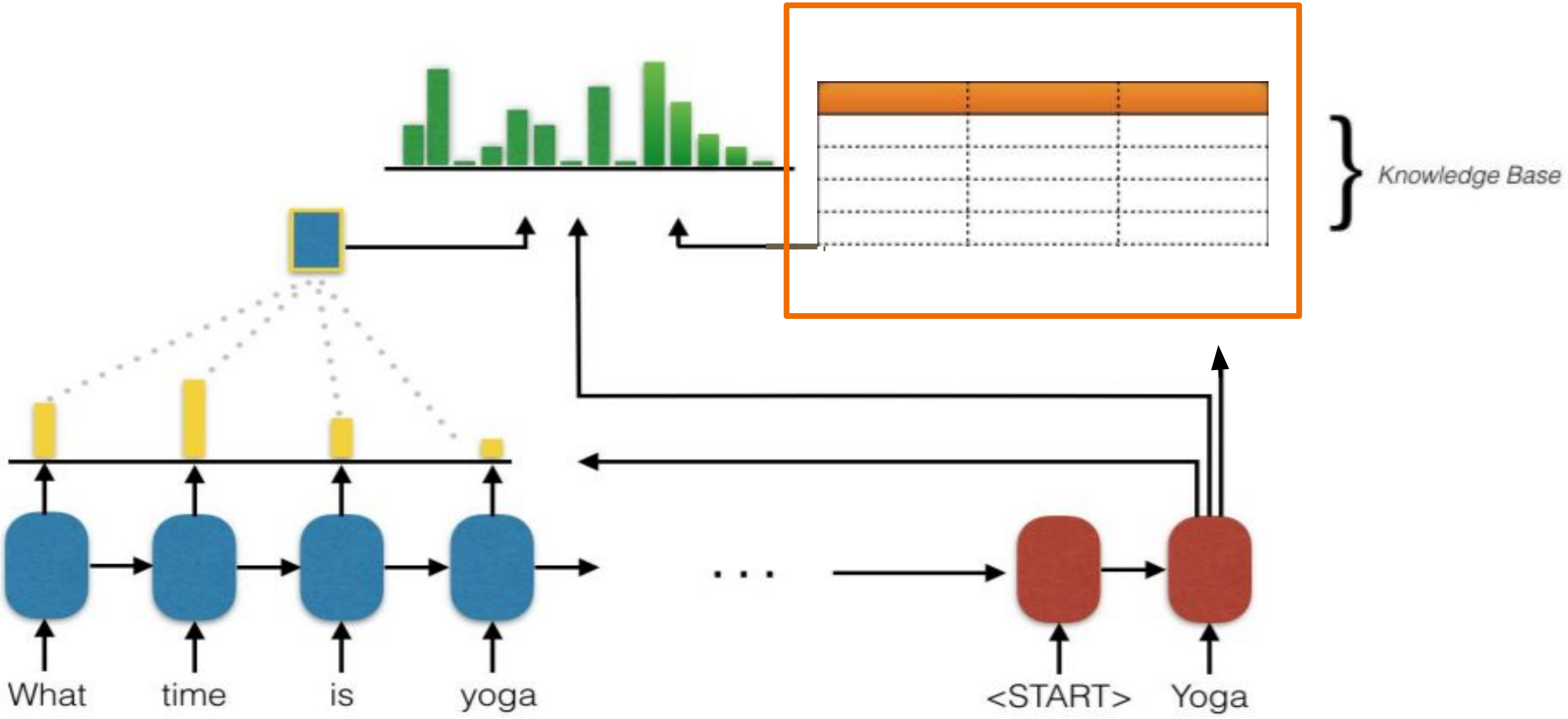
Subject	Relation	Object
dinner	time	8pm
dinner	date	the 13th
dinner	party	Ana
dinner	agenda	-

Key	Value
dinner_time	8pm
dinner_date	the 13th
dinner_party	Ana
dinner_agenta	-

Sequence to Sequence Encoder-Decoder

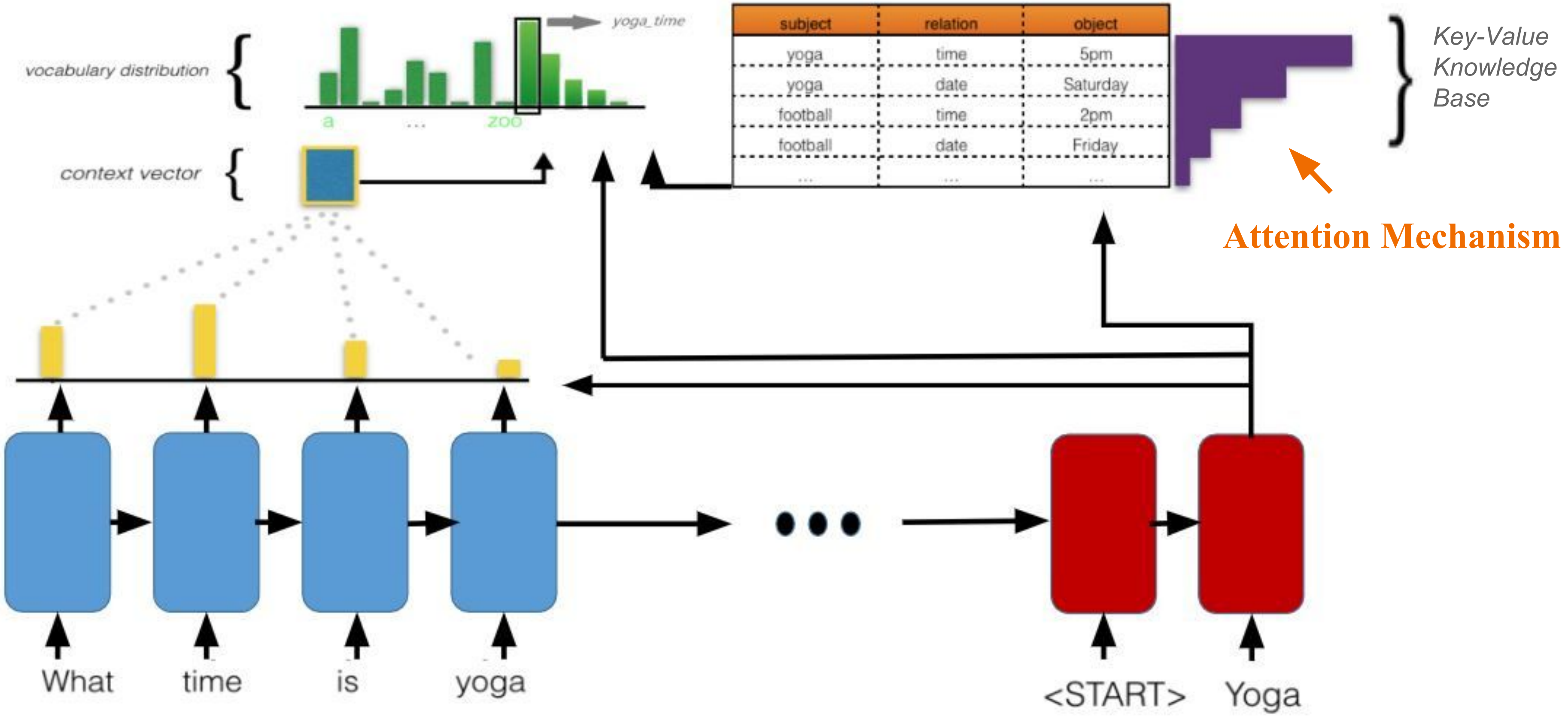


Incorporating the KB

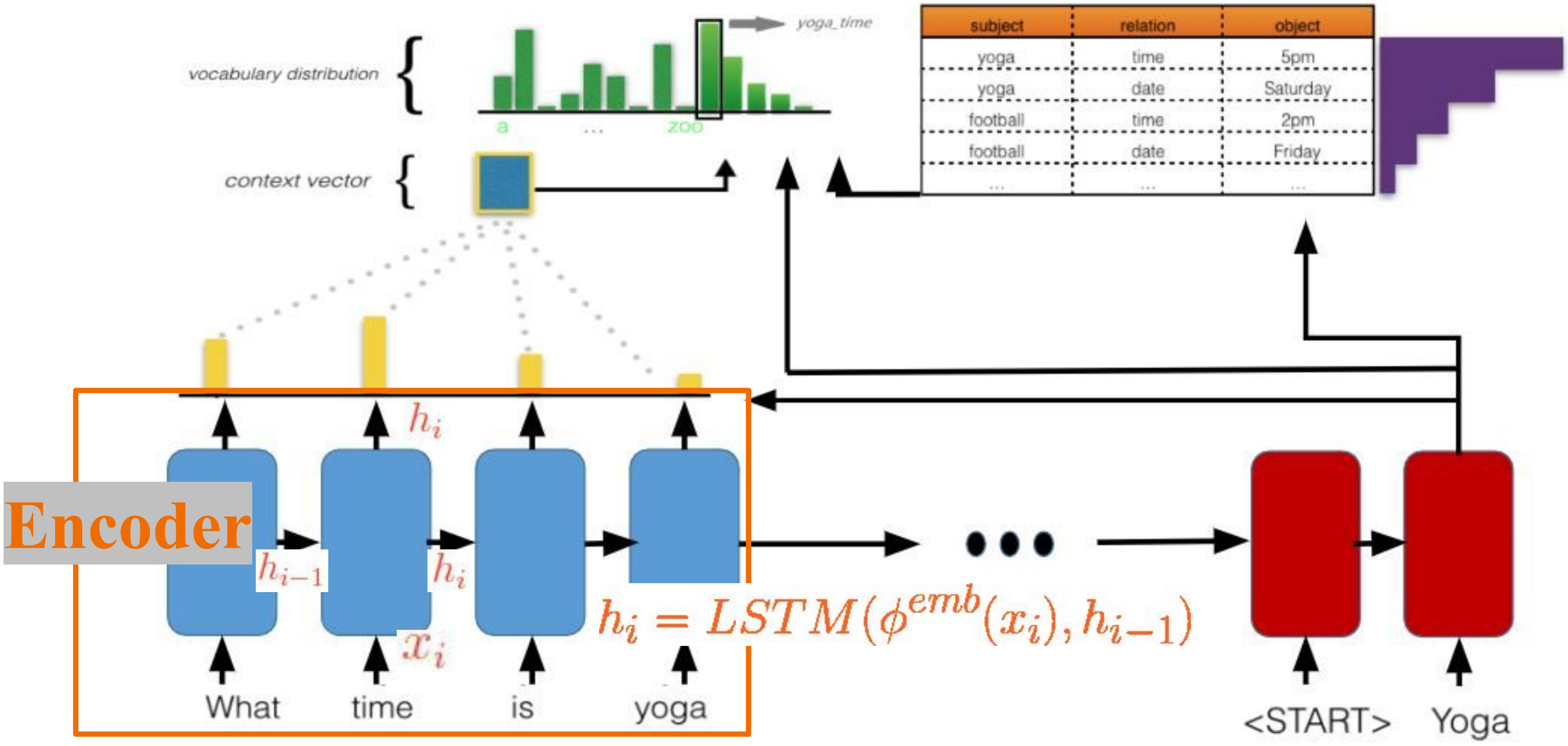


Encoder **Decoder**

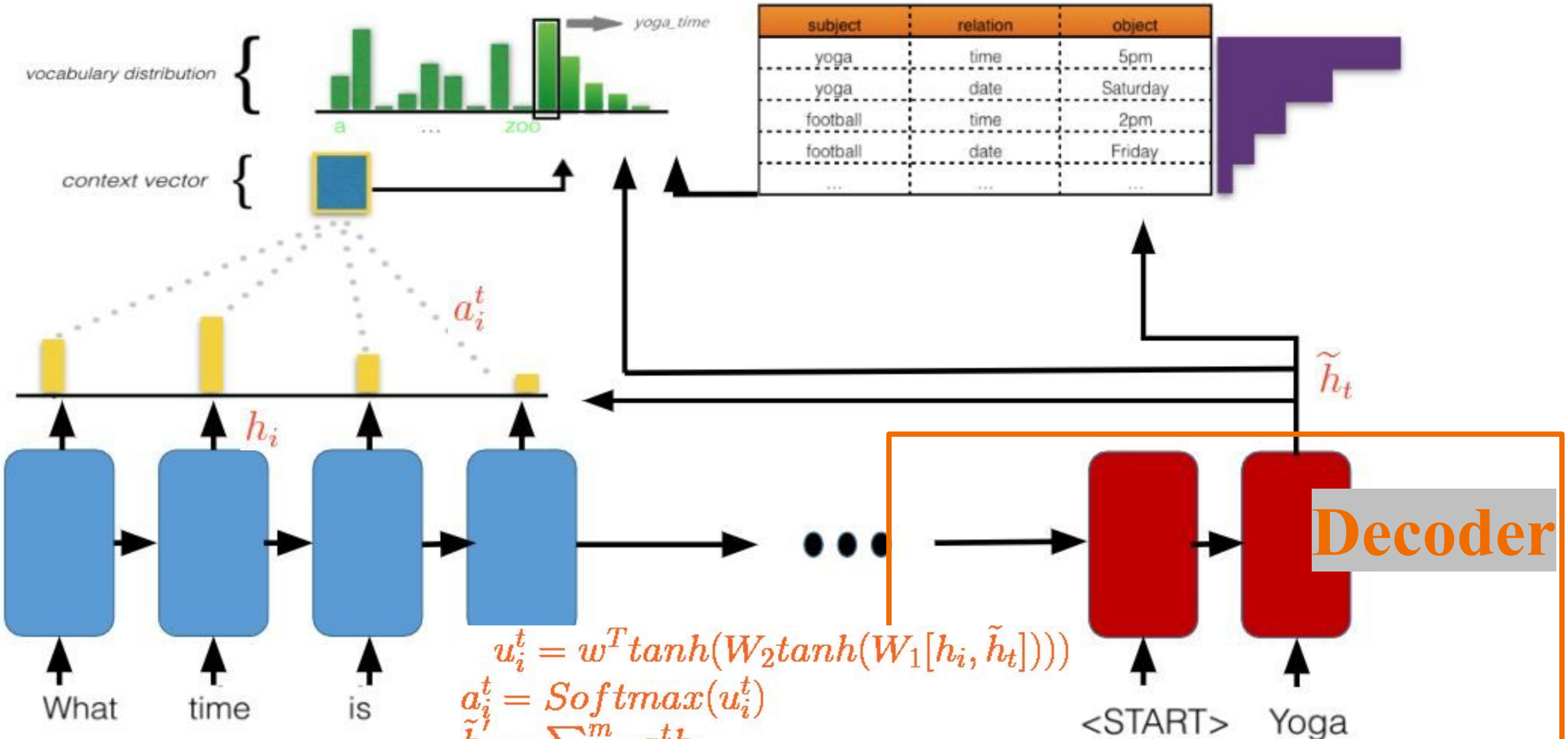
Key-Value Retrieval Network



Key-Value Retrieval Network



Key-Value Retrieval Network



subject	relation	object
yoga	time	5pm
yoga	date	Saturday
football	time	2pm
football	date	Friday
...	...	...

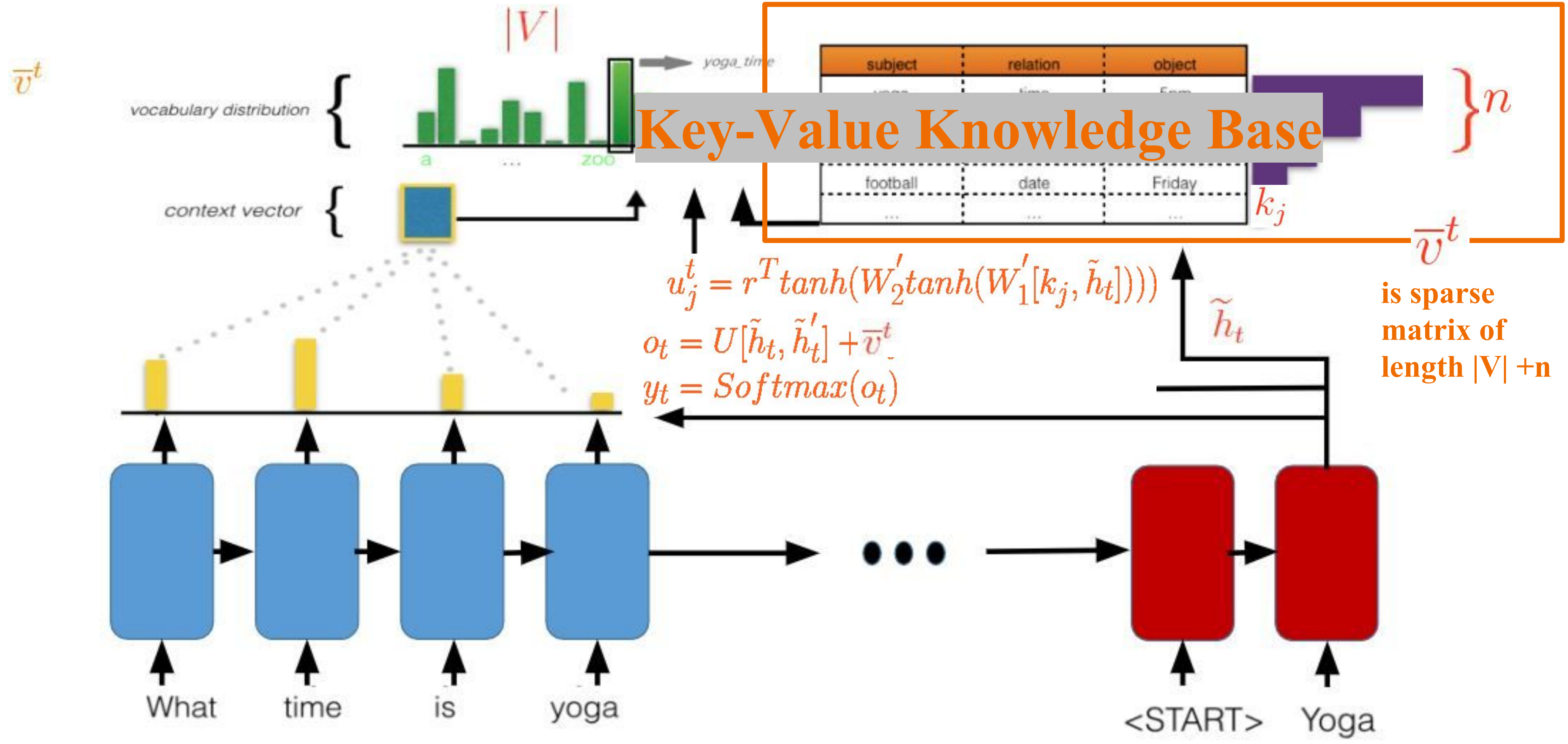
$$u_i^t = w^T tanh(W_2 tanh(W_1[h_i, \tilde{h}_t]))$$

$$a_i^t = Softmax(u_i^t)$$

$$\tilde{h}_t = \sum_{i=1}^{m} a_i^t h_i$$

$U, W1, W2, w$ are trainable parameters of the model

Key-Value Retrieval Network



Scope of Work

1. Read and understand paper
2. Read and understand Keras code (a github implementation exists)
3. Translate it to a pyTorch implementation with better documentation
4. Replicate evaluation results from paper

Code snippet with respect to the equations

$$u_i^t = w^T \tanh(W_2 \tanh(W_1[h_i, \tilde{h}_t]))$$

```
self.dense1_dialogue = nn.Sequential(nn.Linear(200,200),nn.Tanh())
self.dense2_dialogue = nn.Sequential(nn.Linear(200,200),nn.Tanh())
self.dense3_dialogue = nn.Sequential(nn.Linear(200,200),nn.Tanh())
dense1 = self.dense1_dialogue(encoder[0])
dense2 = self.dense2_dialogue(decoder[0])
dense3 = self.dense3 dialogue(torch.add(dense1, dense2))
```

Code snippet with respect to the equations

$$u_i^t = w^T \tanh(W_2 \boxed{\tanh(W_1[h_i, \tilde{h}_t])})$$

```
self.dense1_dialogue = nn.Sequential(nn.Linear(200,200),nn.Tanh())
self.dense2_dialogue = nn.Sequential(nn.Linear(200,200),nn.Tanh())
self.dense3_dialogue = nn.Sequential(nn.Linear(200,200),nn.Tanh())
dense1 = self.dense1_dialogue(encoder[0])
dense2 = self.dense2_dialogue(decoder[0])
dense3 = self.dense3_dialogue(torch.add(dense1, dense2))
```

Code snippet with respect to the equations

$$u_i^t = w^T \boxed{\tanh(W_2 \tanh(W_1[h_i, \tilde{h}_t]))})$$

```
self.dense1_dialogue = nn.Sequential(nn.Linear(200,200),nn.Tanh())
self.dense2_dialogue = nn.Sequential(nn.Linear(200,200),nn.Tanh())
self.dense3_dialogue = nn.Sequential(nn.Linear(200,200),nn.Tanh())
dense1 = self.dense1_dialogue(encoder[0])
dense2 = self.dense2_dialogue(decoder[0])
dense3 = self.dense3_dialogue(torch.add(dense1, dense2))
```

Code snippet with respect to the equations

$$u_i^t = \boxed{w^T \tanh(W_2 \tanh(W_1[h_i, \tilde{h}_t]))})$$

```
self.dense1_dialogue = nn.Sequential(nn.Linear(200,200),nn.Tanh())
self.dense2_dialogue = nn.Sequential(nn.Linear(200,200),nn.Tanh())
self.dense3_dialogue = nn.Sequential(nn.Linear(200,200),nn.Tanh())
dense1 = self.dense1_dialogue(encoder[0])
dense2 = self.dense2_dialogue(decoder[0])
dense3 = self.dense3_dialogue(torch.add(dense1, dense2))
```

Code snippet with respect to the equations

$$a_i^t = \text{Softmax}(u_i^t)$$

```
attention = F.softmax(dense3, dim=2)
```

$$\tilde{h}_t' = \sum_{i=1}^{m} a_i^t h_i$$

```
n_hidden = torch.mul(attention, encoder[0])
```

$$o_t = U[\tilde{h}_t, \tilde{h}_t']$$

```
self.dialogue_output = nn.Linear(400,1954)
output = self.dialogue_output(torch.cat((encoder[0],
                                         n_hidden),dim=2))
```

Code snippet with respect to the equations

$$u_j^t = \boxed{r^T \boxed{\tanh(W_2' \boxed{\tanh(W_1'[k_j, \tilde{h}_t])})}}$$

K: is the word embeddings of the keys from table

```
self.keyvalue_dense1 = nn.Sequential(nn.Linear(431,20),nn.Tanh())
self.keyvalue_dense2 = nn.Sequential(nn.Linear(200,200),nn.Tanh())
self.keyvalue_dense3 = nn.Sequential(nn.Linear(200,431),nn.Tanh())
```

```
n_dense1 = self.keyvalue_dense1(input_embed2)
n_dense2 = self.keyvalue_dense2(decoder)
n_dense3 = self.keyvalue_dense3(torch.cat((n_dense1, n_dense2), dim=1))
```

$$o_t = U[\tilde{h}_t, \tilde{h}_t'] + \bar{v}^t$$

```
n_out = torch.add(output, n_dense3)
```

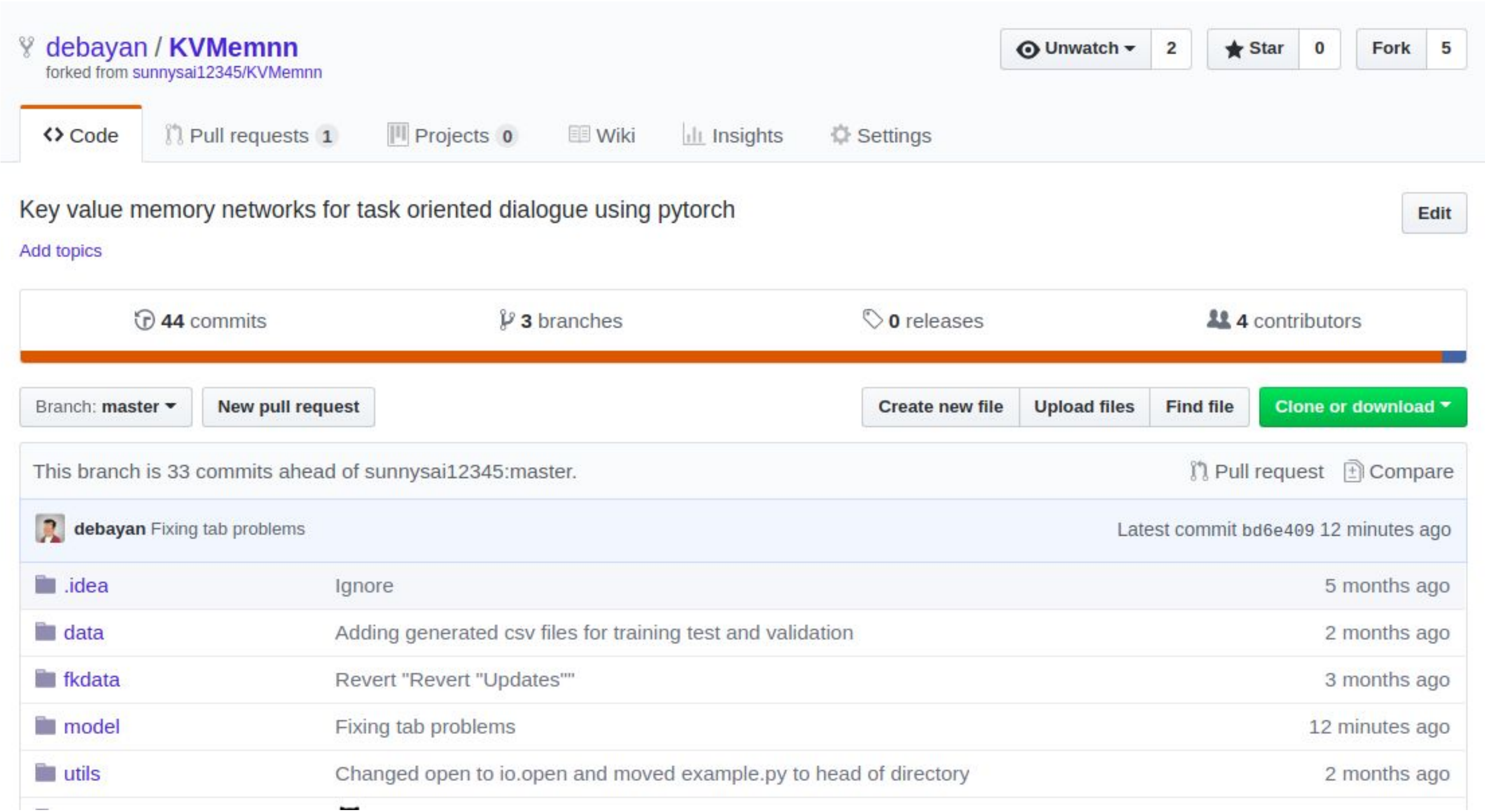
Automatic Evaluation

BLEU:
n-gram based evaluation metrics employed in evaluating machine translation systems
Also been used for evaluating dialogue systems both of the chatbot and task-oriented variety

Entity F1:
To compute, micro-average over the entire set of system dialogue responses
Measures ability to generate relevant entities from underlying KB and to also capture semantics of dialog

Model	BLEU	Ent F1	Scheduling Ent. F1	Weather Ent. F1	Navigation Ent. F1
Rule-Based	6.6	43.8	61.3	39.5	40.4
Copy Net	11.0	37.0	28.1	**50.1**	28.4
Attn. Seq2Seq	10.2	30.0	30.0	42.4	17.9
KV Retrieval Net (no enc. attn.)	10.8	40.9	59.5	35.6	36.6
KV Retrieval Net	**13.2**	**48.0**	**62.9**	47.0	**41.3**
Human Performance	13.5	60.7	64.3	61.6	55.2

Code Snippet and GitHub Screenshots



Code Snippet and GitHub Screenshots

Pytorch: Key value memory networks for task oriented dialogue

The code here implements the one described in the paper Key-Value Retrieval Networks for Task-Oriented Dialogue https://arxiv.org/pdf/1705.05414.pdf

Dataset Thanks to the authors for making the dataset public. https://nlp.stanford.edu/blog/a-new-multi-turn-multi-domain-task-oriented-dialogue-dataset/

Installation

Install the dependencies

```
numpy==1.14.5
pandas==0.23.3
pkg-resources==0.0.0
python-dateutil==2.7.3
pytz==2018.5
six==1.11.0
torch==0.4.0
```

To Run the training use

```
python run.py
```

To run the examples once training is done

Code Snippet and GitHub Screenshots

```
#All equations are refered to https://arxiv.org/pdf/1705.05414.pdf
#Implementation of equation 2
dense1 = self.dense1_dialogue(encoder[0])#apply tanh on after applying linear transformation on encoder output
dense2 = self.dense2_dialogue(decoder[0])#apply tanh on after applying linear transformation on decoder output
dense3 = self.dense3_dialogue(torch.add(dense1, dense2))#combine the output from dens1 and dense2, then applying li

attention = F.softmax(dense3, dim=2) #Apply attention function of the dense3. #Implementation of equation3
n_hidden = torch.mul(attention, encoder[0]) #Weighted summerize the result with attention output and sencode output
output = self.dialogue_output(torch.cat((encoder[0], n_hidden),dim=2)) #Linear transformation on the output of casc

# input2: Key value table
if device == torch.device("cuda"):
    input_embed2 = self.input_embed_keyvalue(input_keyvalues.cuda())
else:
    input_embed2 = self.input_embed_keyvalue(input_keyvalues)

#Implementation of equation 7
input_embed2 = reshape(input_embed2, self.batch_size, 431, self.embedding_size, self.pad_length)
```

Conclusion

Introduced the Key-Value Retrieval Network

Novel end-to-end system that can be used to incorporate KB information

Outperforms several baseline models (Rule Based, Copy-Augmented Sequence-to-Sequence Network) in both automatic and human evaluation metrics.

Introduced New Dataset of 3,031 dialogues in an in-car virtual assistant domain to help the data scarcity issue present in task-oriented dialogue research.

Tasks/Contributions

Task	Status	Contribution
Read paper and understand	Done	Debayan, Pooja, Jing-Long
Run sample keras code and understand	Done	Debayan, Pooja, Jing-Long
Write pytorch code and comment appropriately	Done	Debayan, Pooja, Jing-Long
Debug	In progress	Debayan, Pooja, Jing-Long
Evaluate code	In progress	Debayan, Pooja, Jing-Long
Presentation	Done	Pooja, Jing-Long

Thank You