



# NATURAL LANGUAGE QUESTION ANSWER SYSTEMS

Anna Bethke, Andy Keller

Deep Learning Data Scientists, AIPG

# INTRODUCTIONS



Andy Keller

Deep Learning Data Scientist  
AIPG



Anna Bethke

Head of AI4Good  
AIPG

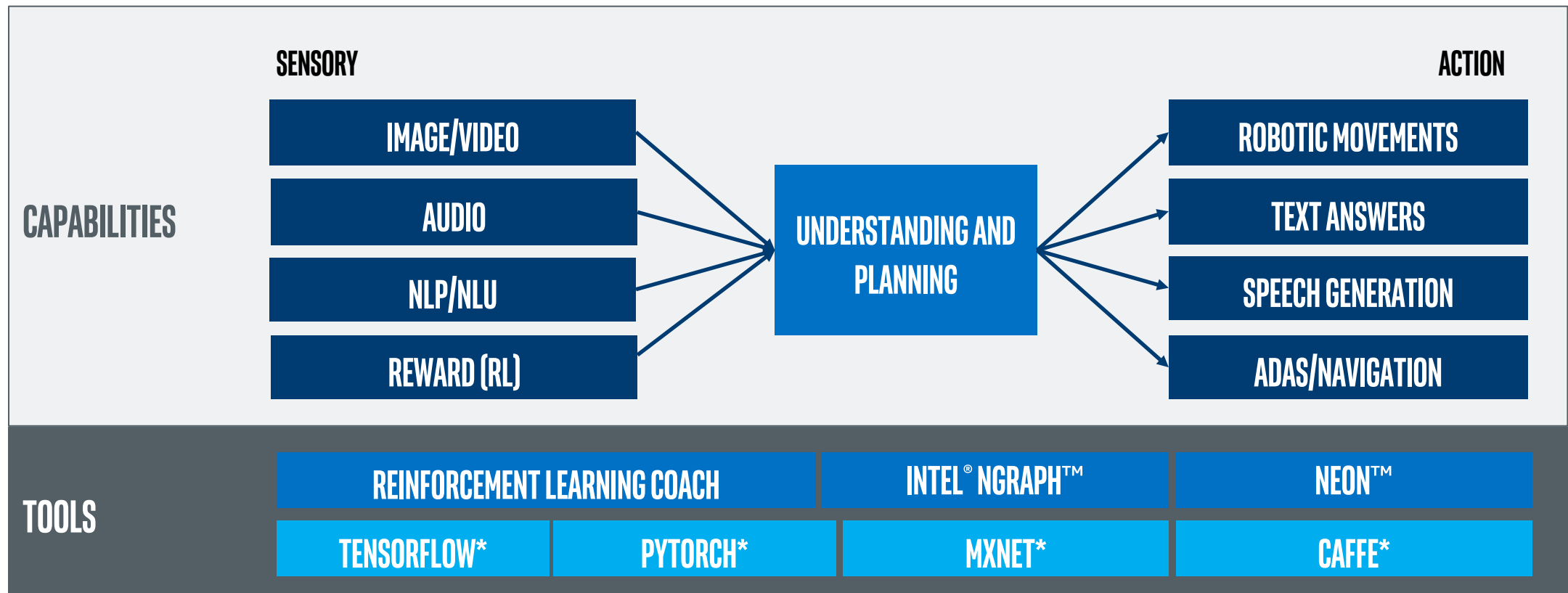
- UCSD: Deep neural networks and memory augmented neural networks for NLP

- MIT: Human computer interaction and aerospace engineering

# OUTLINE

- AI-Lab introduction
- Dialog management and question answer systems
- nGraph Library overview
- Memory-network model building blocks
- Code walk through and demonstration
- Next steps

# INTEL® AI LAB

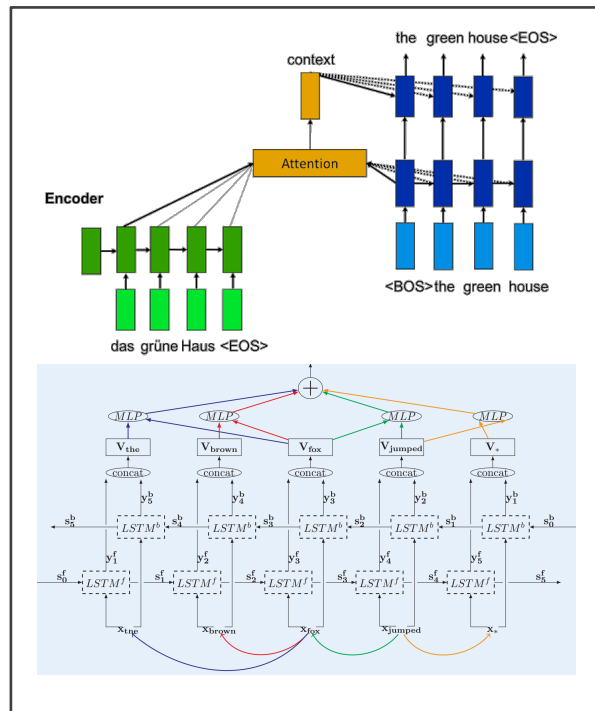


\*Other names and brands may be claimed as the property of others.

# INTEL® AI LAB NATURAL LANGUAGE PROCESSING (NLP)

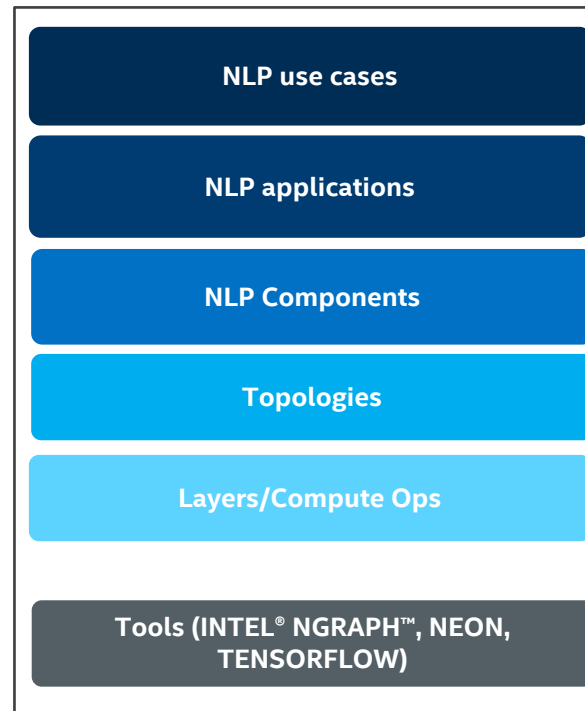
## NLP Research

Develop SOTA and innovative models, Optimize for pioneering AI hardware



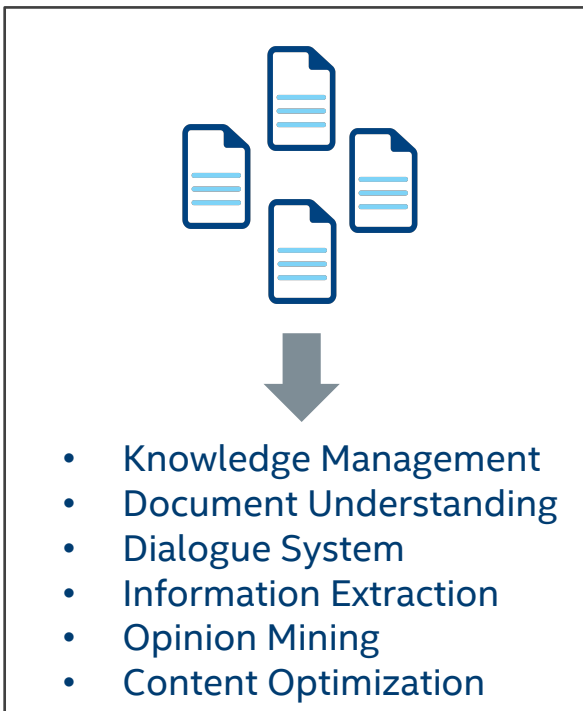
## NLP Library

Building an open and flexible NLP library using Intel technologies

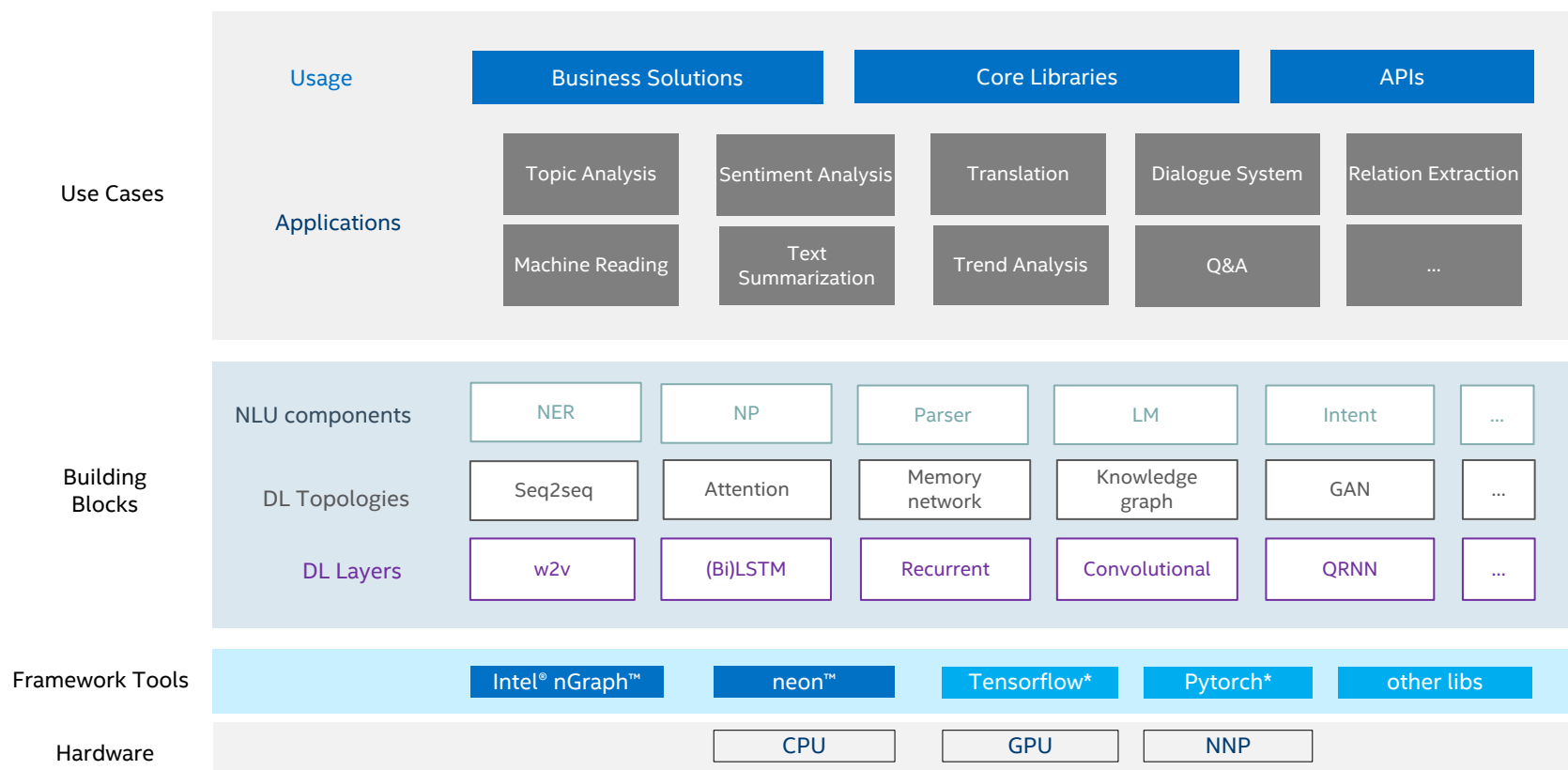


## Data Science

Applied NLP to domain use cases and build business applications with Intel customers



# INTEL® AI LAB NLP ARCHITECT – AN OPEN AND FLEXIBLE STACK



<https://github.com/NervanaSystems/nlp-architect>

# HANDS-ON LAB SETUP

- Code: [https://github.com/NervanaSystems/ai-lab-nlp/tree/master/core\\_models/memn2n\\_dialogue](https://github.com/NervanaSystems/ai-lab-nlp/tree/master/core_models/memn2n_dialogue)
- Trained model and code download: <https://tinyurl.com/y7hocn8s>
- **Setup:**
  - cd ai\_lab\_nlp
  - virtualenv .venv
  - . .venv/bin/activate
  - python setup.py install
  - cd ai\_lab\_nlp/core\_models/memn2n\_dialogue/
  - pip install -r requirements.txt
- **Training:**
  - python train\_model.py --task 5 --weights\_save\_path memn2n\_weights.npz --data\_dir . --test
- **Inference:**
  - python interactive.py --task 5 --model\_file memn2n\_weights.npz

```
-----|| Attention
hi 1_TIME <USER>|| 0.0
hello what can i help you with today 2_TIME <BOT>|| 0.0
can you make a restaurant reservation in madrid 2_TIME <USER>|| 0.0
i'm on it 3_TIME <BOT>|| 0.0
<SILENCE> 3_TIME <USER>|| 0.0
any preference on a type of cuisine 4_TIME <BOT>|| 0.0
french please 4_TIME <USER>|| 0.0
how many people would be in your party 5_TIME <BOT>|| 0.0
for four please 5_TIME <USER>|| 0.0
which price range are looking for 6_TIME <BOT>|| 0.0
in a cheap price range please 6_TIME <USER>|| 0.0
ok let me look into some options for you 7_TIME <BOT>|| 0.0
<SILENCE> 7_TIME <USER>|| 0.0
api_call french madrid four cheap 8_TIME <BOT>|| 0.0
resto_madrid_cheap_french_7stars R_number four 8_TIME <USER>|| 0.0
resto_madrid_cheap_french_8stars R_cuisine french 9_TIME <USER>|| 0.0
resto_madrid_cheap_french_8stars R_location madrid 10_TIME <USER>|| 0.0
resto_madrid_cheap_french_8stars R_price cheap 11_TIME <USER>|| 0.0
resto_madrid_cheap_french_8stars R_rating 8 12_TIME <USER>|| 0.0
resto_madrid_cheap_french_8stars R_phone resto_madrid_cheap_french_8stars_phone 13_TIME <USER>|| 0.0
resto_madrid_cheap_french_8stars R_address resto_madrid_cheap_french_8stars_address 14_TIME <USER>|| 0.0
resto_madrid_cheap_french_8stars R_number four 15_TIME <USER>|| 0.0
resto_madrid_cheap_french_5stars R_cuisine french 16_TIME <USER>|| 0.0
resto_madrid_cheap_french_5stars R_location madrid 17_TIME <USER>|| 0.0
resto_madrid_cheap_french_5stars R_price cheap 18_TIME <USER>|| 0.0
resto_madrid_cheap_french_5stars R_rating 5 19_TIME <USER>|| 0.0
resto_madrid_cheap_french_5stars R_phone resto_madrid_cheap_french_5stars_phone 20_TIME <USER>|| 0.0
resto_madrid_cheap_french_5stars R_address resto_madrid_cheap_french_5stars_address 21_TIME <USER>|| 0.0
resto_madrid_cheap_french_5stars R_number four 22_TIME <USER>|| 0.0
resto_madrid_cheap_french_7stars R_cuisine french 23_TIME <USER>|| 0.0
resto_madrid_cheap_french_7stars R_location madrid 24_TIME <USER>|| 0.0
resto_madrid_cheap_french_7stars R_price cheap 25_TIME <USER>|| 0.0
resto_madrid_cheap_french_7stars R_rating 7 26_TIME <USER>|| 0.0
resto_madrid_cheap_french_7stars R_phone resto_madrid_cheap_french_7stars_phone 27_TIME <USER>|| 0.0
resto_madrid_cheap_french_7stars R_address resto_madrid_cheap_french_7stars_address 28_TIME <USER>|| 0.0
<SILENCE> 29_TIME <USER>|| 0.0
what do you think of this option: resto_madrid_cheap_french_7stars 31_TIME <BOT>|| 0.0
let's do it 31_TIME <USER>|| 0.0
great let me do the reservation 32_TIME <BOT>|| 0.0
do you have its address 32_TIME <USER>|| 0.0
here it is resto_madrid_cheap_french_7stars_address 33_TIME <BOT>|| 0.0
thanks 33_TIME <USER>|| 0.0
is there anything i can help you with 35_TIME <BOT>|| 1.0
no 35_TIME <USER>|| 0.0
-----||
```



# QUESTION ANSWER SYSTEMS

The background of the slide features a blue gradient that transitions from a lighter shade on the left to a darker shade on the right. Overlaid on this gradient is a complex network of thin, light blue lines connecting numerous small, dark blue circular nodes. These nodes and lines are distributed across the lower half of the image, creating a sense of a digital or neural network. The overall aesthetic is clean, modern, and technological.

# WHAT IS A QUESTION & ANSWERING SYSTEM?

“Most, if not all, tasks in natural language processing can be cast as a question answering problem” – Kumar et al. <sup>1</sup>

- Open Domain Dialogue
  - Conversational agents
- Goal Oriented Dialogue
  - Information retrieval
  - Reading comprehension
  - Personal assistant / concierge

<sup>1</sup> [arXiv:1506.07285](https://arxiv.org/abs/1506.07285)

# OPEN DOMAIN: CONVERSATIONAL AGENTS

- Traditionally trained as a language model on raw dialogues
  - Datasets: Movie Subtitles, Chat-room logs (Ubuntu IT help)
- Recent approaches use encoder-decoder models to predict the next word conditioned on the last response or history of conversation.
  - Very limited contextual memory
  - Virtually no domain specificity

**Human:** “How much is two plus two?”

**Machine:** “four.”

**Human:** “How much is ten minus two?”

**Machine:** “seventy-two.”

**Human:** “What is your job?”

**Machine:** “I’m a lawyer.”

**Human:** “What do you do?”

**Machine:** “I’m a doctor.”

# INFORMATION RETRIEVAL

- Trained on question/answer pairs with associated knowledge base or textual knowledge source
  - Typically single-turn dialog, works best with highly structured knowledge base
  - Domain is closed to simple informational questions

## **KB entries for Blade Runner:**

Blade Runner directed\_by Ridley Scott

Blade Runner release\_year 1982

Blade Runner starring Harrison Ford

Blade Runner distributed\_by Warner Bros.

**Human:** “Ridley Scott directed which films?”

**Machine:** “Blade Runner”

**Human:** “What year was the movie Blade Runner released?”

**Machine:** “1982”

# READING COMPREHENSION

- Given a story/document as context, structured answer/question pairs
  - Trained using (story, question, answer) triplets.
  - More relaxed than information retrieval tasks, questions are typically more challenging and focused on the provided short story

## **Story:**

The BBC producer allegedly struck by Jeremy Clarkson will not press charges against the “Top Gear” host, his lawyer said Friday. .... the British broadcaster found he had subjected producer Oisin Tymon “to an unprovoked physical and verbal attack.”

## **Question:**

Which producer will not press charges against Jeremy Clarkson?

## **Answer:**

Oisin Tymon

# PERSONAL ASSISTANT / CONCIERGE

- Making reservations, automated customer support
- Typically trained with a dialogue simulator
  - Multiple turn dialogue where most important goal is ultimate outcome

**Human:** “I’d like to book dinner for 6 tonight at an Italian restaurant.”

**Machine:** “Ok, what price range would you like?”

**Human:** “Expensive”

**Machine:** `api_call(Italian, 6, expensive)`

**Machine:** What do you think of ‘Enoteca Adriano’?



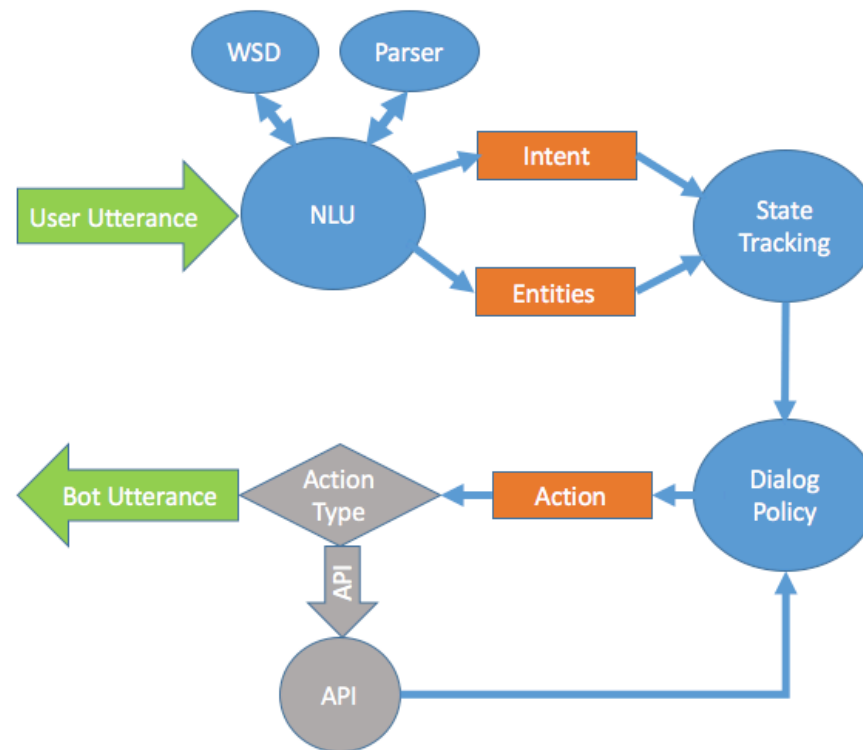
# **DIALOGUE MANAGEMENT SYSTEMS**

The background features a blue gradient that transitions from a lighter shade on the left to a darker shade on the right. Overlaid on this gradient is a complex network of thin, light blue lines connecting numerous small, dark blue circular nodes. These nodes and lines are distributed across the lower half of the image, creating a sense of a digital or neural network. The overall aesthetic is clean, modern, and technological.

# INTENT-BASED DIALOG AGENTS

Most commercial dialogue agents:

- Multiple separately trained components joined with logic
- Expensive to gather labeled data





# END-TO-END DIALOGUE AGENTS

Remove rigid dialogue structure and train a single model with input / output pairs – allow the model to maintain it's own latent dialogue state

## Advantages:

- Data requirements are significantly less expensive
- Has the potential to learn new tasks via raw conversational data
- Can handle long open-ended conversations
- Can combine knowledge from the full conversation to solve complex tasks and implicitly handle edge-cases

# MEMORY NETWORKS

Memory Networks have four generalized component networks:

- I: (input feature map) map input to the internal feature space
- G: (generalization) update memories given new input
- O: (output) produce new output in feature space using memories
- R: (response) map output O to natural language response

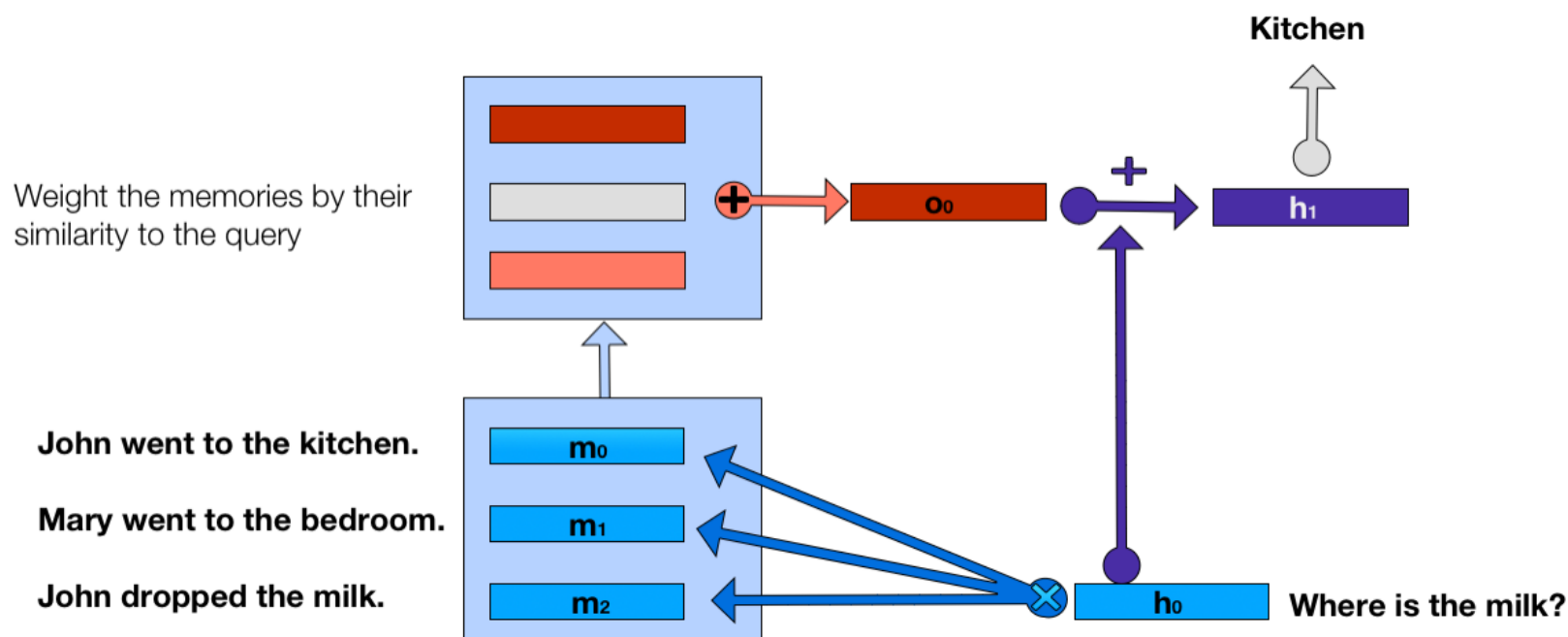
Weights can be shared between components

<sup>1</sup> [arXiv:1410.3915](https://arxiv.org/abs/1410.3915)

# END-TO-END MEMORY NETWORKS

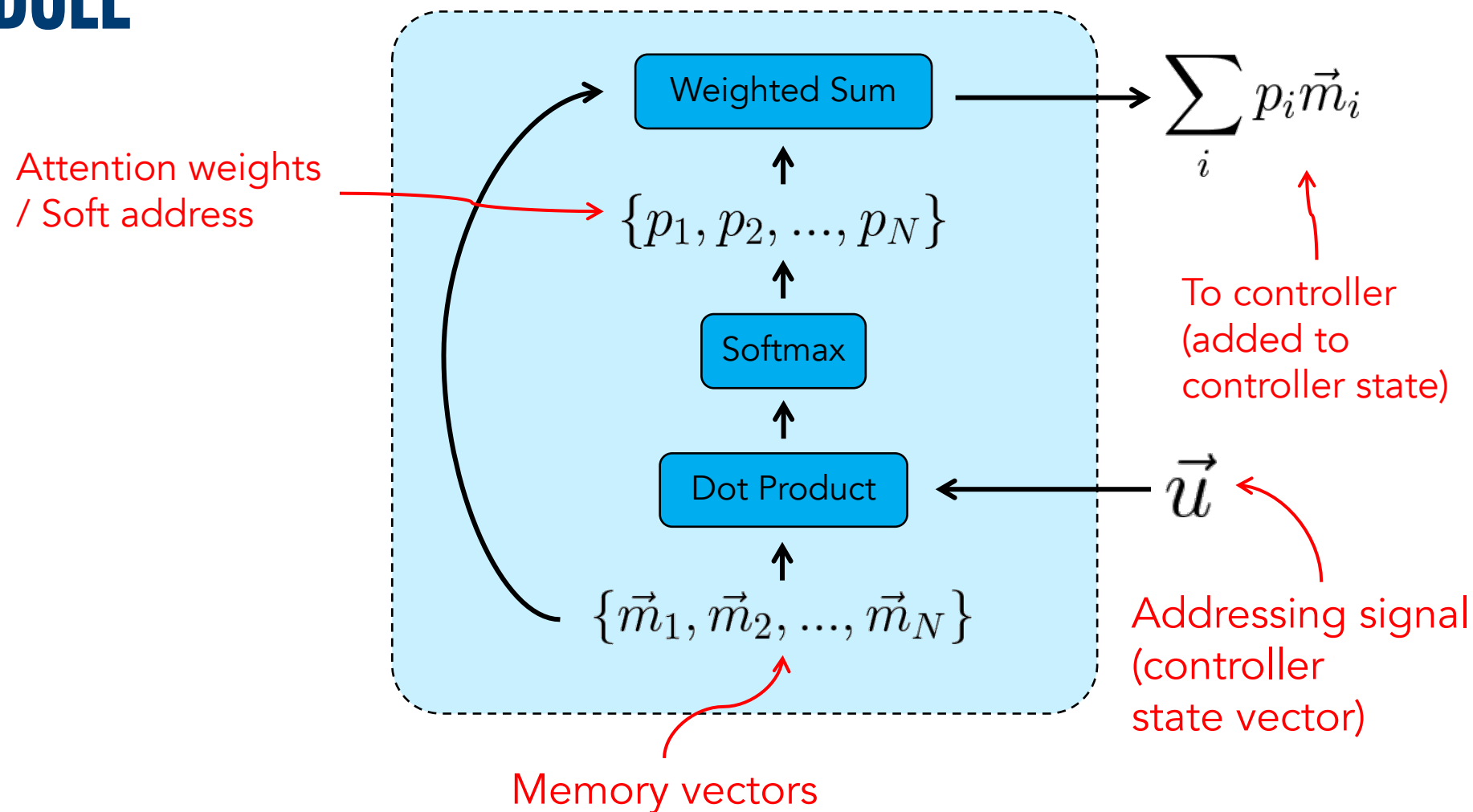
## Overall Idea:

- Allow network to query the memories and receive the most relevant memories as additional input before generating an output.



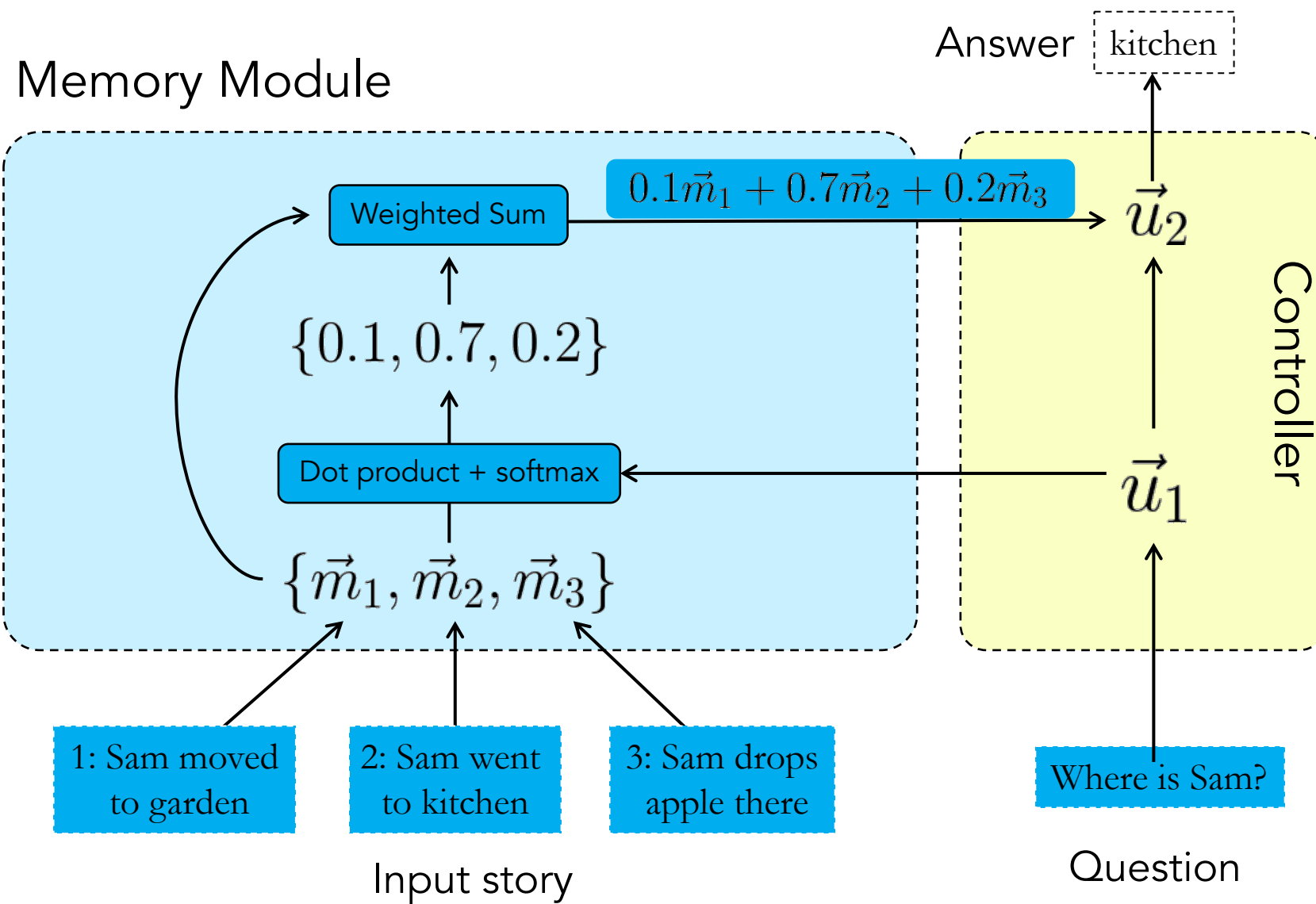
<sup>1</sup> [arXiv:1503.08895](https://arxiv.org/abs/1503.08895)

# MEMORY MODULE



Source: <http://www.thespermwhale.com/jaseweston/icml2016/>

# EXAMPLE



Source: <http://www.thespermwhale.com/jaseweston/icml2016/>

# END-TO-END MEMORY NETWORKS (1-HOP)

## Input:

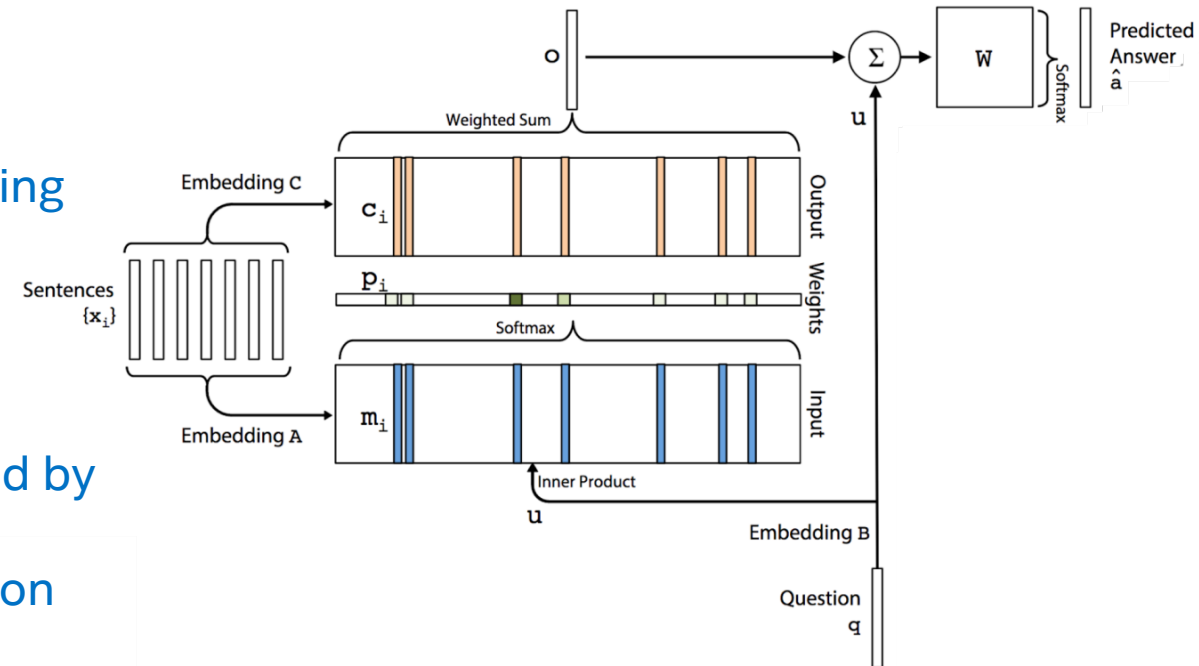
- Embed question as Bag-of-Words
- Embed each memory as Bag-of-Words

## Query:

- Dot question embedding with each memory embedding
- Softmax over these dot products to get similarity of question with each memory

## Output:

- Take weighted sum of memory embeddings (weighted by softmax output)
- Add weighted sum of memory embeddings to Question embedding



## Respond:

- Apply final linear layer w/ softmax & predict output word

# END-TO-END MEMORY NETWORKS (1-HOP)

$$\text{Memory Slots} = \{x_0, x_1, x_2, \dots, x_{nsent}\} \quad x_i = \{x_{i1}, x_{i2}, \dots, x_{in}\}$$

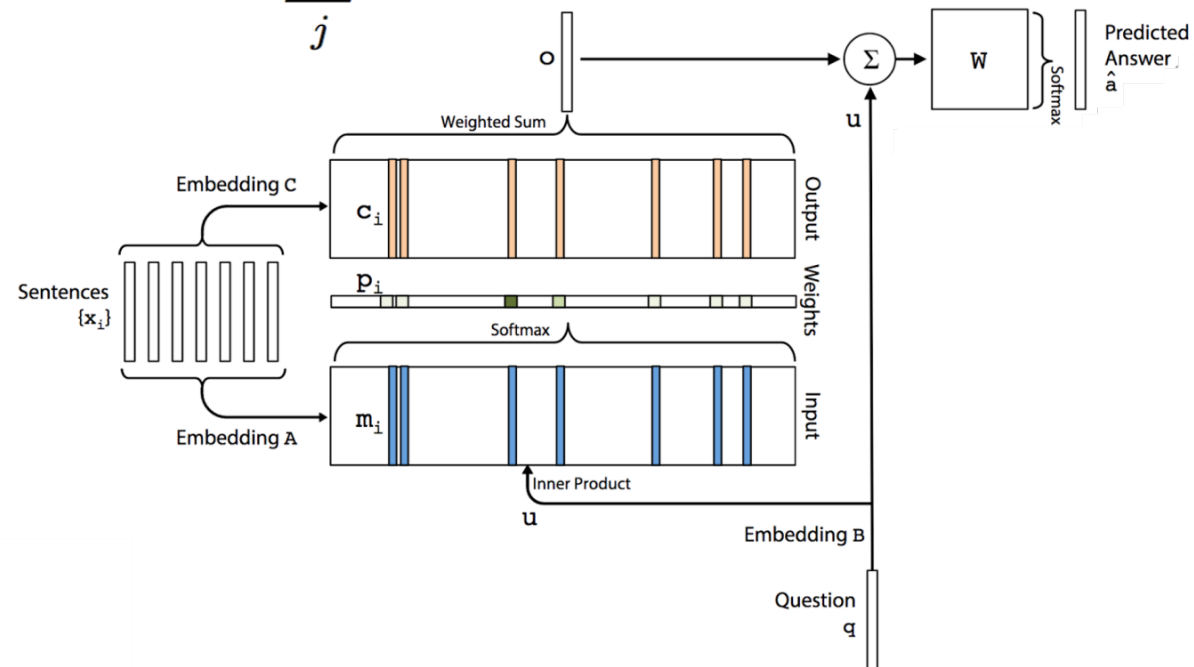
$$q = \{q_1, q_2, \dots, q_n\}$$

$$u = \sum_j^n Bq_j \quad m_i = \sum_j^n Ax_{ij} \quad c_i = \sum_j^n Cx_{ij}$$

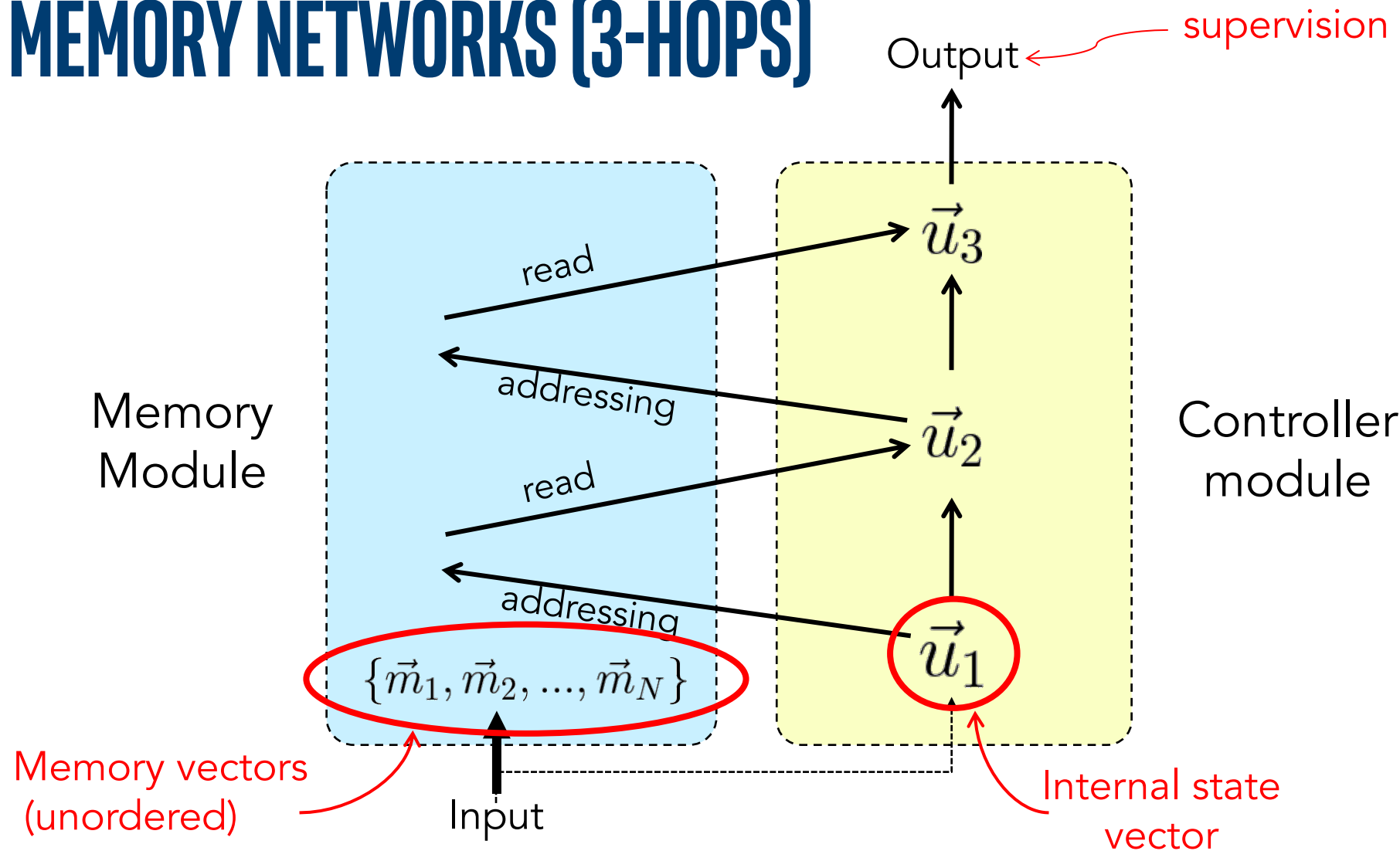
$$p_i = \text{Softmax}(u^T m_i)$$

$$o = \sum_i^{nsent} p_i c_i$$

$$\hat{a} = \text{Softmax}(W(o + u))$$



# END-TO-END MEMORY NETWORKS (3-HOPS)

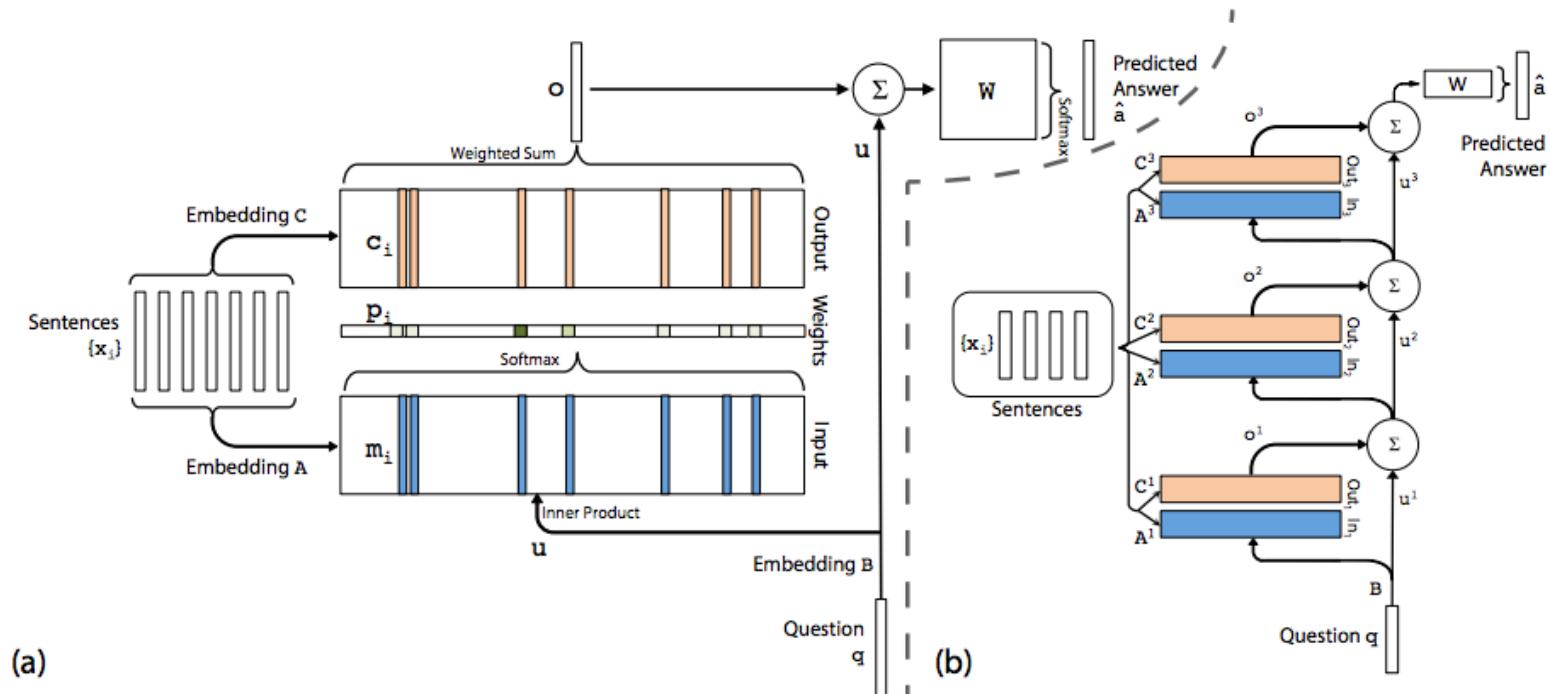


Source: <http://www.thespermwhale.com/jaseweston/icml2016/>



# END-TO-END MEMORY NETWORKS (MULTIPLE HOPS)

- Each 'hop' consists of a 'read' from memory and sum into internal model state  $u$ .
- To perform multiple hops, simply read & sum multiple times before predicting answer with internal state – improves results.



# COOL. WHAT CAN THEY DO?

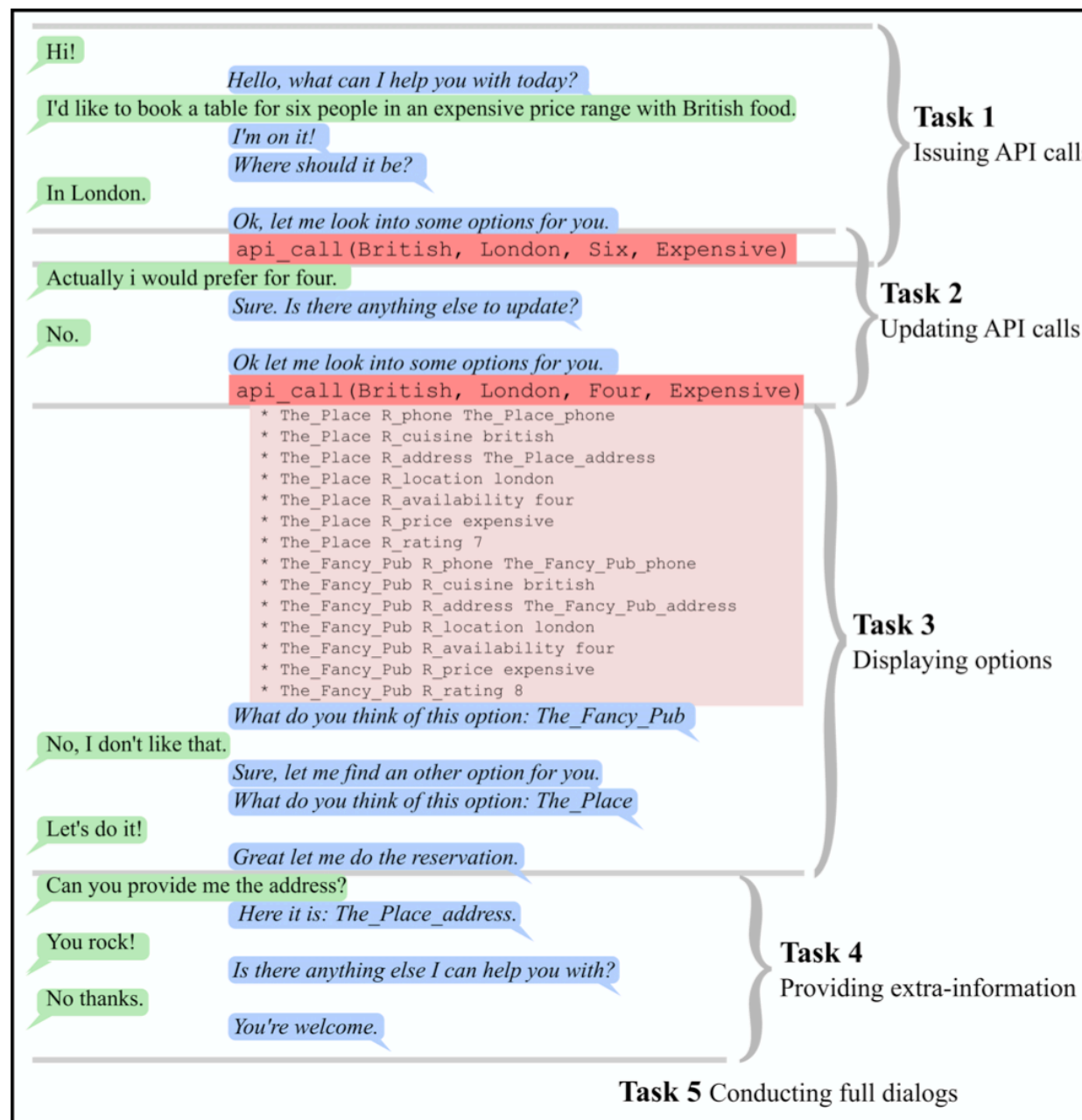
- Memory is flexible and persistent
  - Can store any information that would help answering queries.
- For dialogue:
  - Memory = History of conversation
- For information retrieval:
  - Memory = Knowledge base entries
- For reading comprehension:
  - Memory = Story / document sentences

Story (13: compound coherence)	Support	Hop 1	Hop 2	Hop 3
Sandra and Daniel travelled to the bathroom.		0.13	0.00	0.00
Afterwards they went back to the office.		0.01	0.00	0.00
Daniel and Mary travelled to the hallway.		0.01	0.00	0.00
Following that they went back to the office.		0.06	0.04	0.00
Mary and Sandra moved to the hallway.	yes	0.59	0.02	0.00
Then they went to the kitchen.	yes	0.02	0.94	1.00
Where is Sandra? Answer: kitchen Prediction: kitchen				

# BABI GOAL ORIENTED DIALOGUE DATASET

- Synthetic dataset used as a baseline for goal oriented dialogue systems
- Designed as a conversation between a customer and a restaurant booking agent (the bot)
- Divided into 4 separate tasks to for partial accuracy analysis. 5<sup>th</sup> task is full dialogue.

[arXiv:1605.07683](https://arxiv.org/abs/1605.07683)



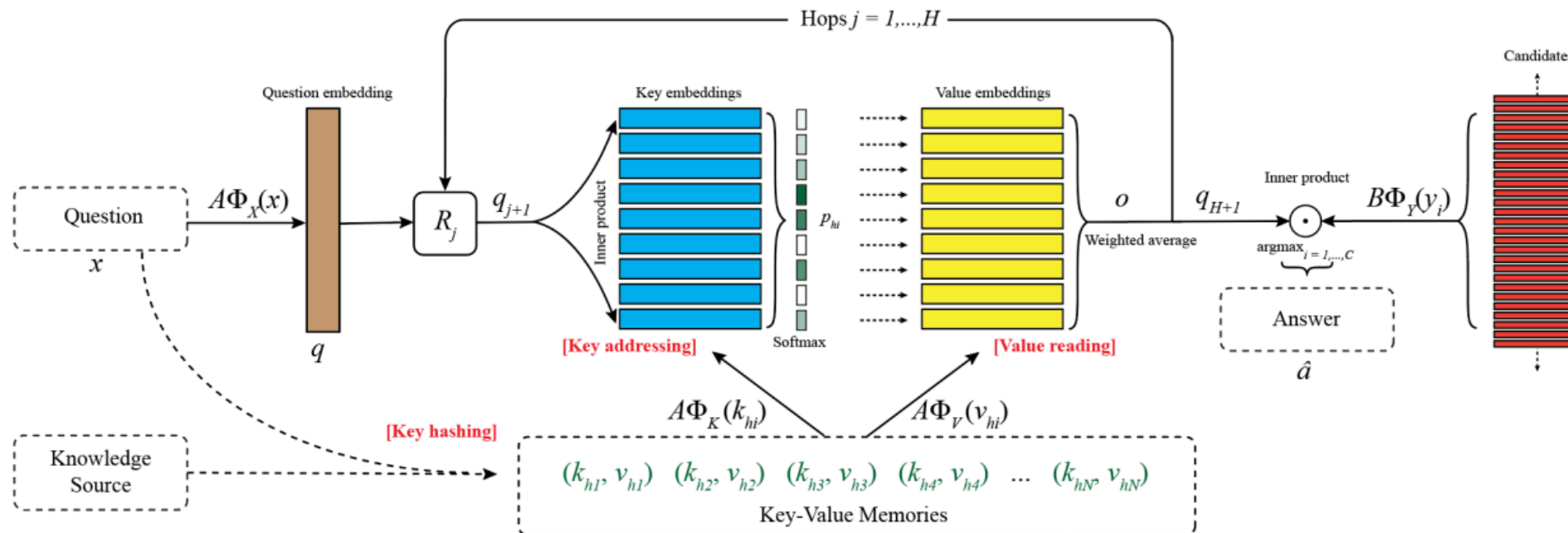
# GOAL ORIENTED DIALOGUE PERFORMANCE

Table 2: **Test results across all tasks and methods.** For tasks T1-T5 results are given in the standard setup and the out-of-vocabulary (OOV) setup, where words (e.g. restaurant names) may not have been seen during training. Task T6 is the Dialog state tracking 2 task with real dialogs, and only has one setup. Best performing methods (or methods within 0.1% of best performing) are given in bold for the per-response accuracy metric, with the per-dialog accuracy given in parenthesis.

Task	Rule-based Systems	TF-IDF Match	Nearest Neighbor	Supervised Embeddings	MemNNs	MemNNs +match
T1: Issuing API calls	100 (100)	5.6 (0)	55.1 (0)	<b>100</b> (100)	<b>99.9</b> (99.6)	<b>100</b> (100)
T2: Updating API calls	100 (100)	3.4 (0)	68.3 (0)	68.4 (0)	<b>100</b> (100)	98.3 (83.9)
T3: Displaying options	100 (100)	8.0 (0)	58.8 (0)	64.9 (0)	<b>74.9</b> (2.0)	<b>74.9</b> (0)
T4: Providing information	100 (100)	9.5 (0)	28.6 (0)	57.2 (0)	59.5 (3.0)	<b>100</b> (100)
T5: Full dialogs	100 (100)	4.6 (0)	57.1 (0)	75.4 (0)	<b>96.1</b> (49.4)	93.4 (19.7)
T1(OOV): Issuing API calls	100 (100)	5.8 (0)	44.1 (0)	60.0 (0)	72.3 (0)	<b>96.5</b> (82.7)
T2(OOV): Updating API calls	100 (100)	3.5 (0)	68.3 (0)	68.3 (0)	78.9 (0)	<b>94.5</b> (48.4)
T3(OOV): Displaying options	100 (100)	8.3 (0)	58.8 (0)	65.0 (0)	74.4 (0)	<b>75.2</b> (0)
T4(OOV): Providing inform.	100 (100)	9.8 (0)	28.6 (0)	57.0 (0)	57.6 (0)	<b>100</b> (100)
T5(OOV): Full dialogs	100 (100)	4.6 (0)	48.4 (0)	58.2 (0)	65.5 (0)	<b>77.7</b> (0)
T6: Dialog state tracking 2	33.3 (0)	1.6 (0)	21.9 (0)	22.6 (0)	<b>41.1</b> (0)	<b>41.0</b> (0)

# KEY-VALUE MEMORY NETWORKS

- Extension of end-to-end Memory Networks with dictionary style memory lookup
- Arbitrary mapping between lookup-keys and values allows network designer to incorporate domain knowledge



[arXiv:1606.03126](https://arxiv.org/abs/1606.03126)



# WIKIMOVIES DATASET

- Possible knowledge representations:
  - Raw Wikipedia document
  - Knowledge base of entities and relations
    - Either human annotated or created through Information Extraction techniques
- 100,000 Q&A pairs with 13 question types corresponding to relations in knowledge graph

Method	KB	IE	Doc
(Bordes <i>et al.</i> , 2014) QA system	93.5	56.5	N/A
Supervised Embeddings	54.4	54.4	54.4
Memory Network	78.5	63.4	69.9
Key-Value Memory Network	<b>93.9</b>	<b>68.3</b>	<b>76.2</b>

**Table 2:** Test results (% hits@1) on WIKIMOVIES, comparing human-annotated KB (KB), information extraction-based KB (IE), and directly reading Wikipedia documents (Doc).

## Doc: Wikipedia Article for Blade Runner (partially shown)

Blade Runner is a 1982 American neo-noir dystopian science fiction film directed by Ridley Scott and starring Harrison Ford, Rutger Hauer, Sean Young, and Edward James Olmos. The screenplay, written by Hampton Fancher and David Peoples, is a modified film adaptation of the 1968 novel “Do Androids Dream of Electric Sheep?” by Philip K. Dick. The film depicts a dystopian Los Angeles in November 2019 in which genetically engineered replicants, which are visually indistinguishable from adult humans, are manufactured by the powerful Tyrell Corporation as well as by other “mega-corporations” around the world. Their use on Earth is banned and replicants are exclusively used for dangerous, menial, or leisure work on off-world colonies. Replicants who defy the ban and return to Earth are hunted down and “retired” by special police operatives known as “Blade Runners”. ...

## KB entries for Blade Runner (subset)

Blade Runner *directed\_by* Ridley Scott  
 Blade Runner *written\_by* Philip K. Dick, Hampton Fancher  
 Blade Runner *starred\_actors* Harrison Ford, Sean Young, ...  
 Blade Runner *release\_year* 1982  
 Blade Runner *has\_tags* dystopian, noir, police, androids, ...

## IE entries for Blade Runner (subset)

Blade Runner, Ridley Scott *directed* dystopian, science fiction, film  
 Hampton Fancher *written* Blade Runner  
 Blade Runner *starred* Harrison Ford, Rutger Hauer, Sean Young...  
 Blade Runner *labelled* 1982 neo noir  
 special police, Blade *retired* Blade Runner  
 Blade Runner, special police *known* Blade

## Questions for Blade Runner (subset)

Ridley Scott *directed* which films?  
 What year was the movie Blade Runner released?  
 Who is the writer of the film Blade Runner?  
 Which films can be described by dystopian?  
 Which movies was Philip K. Dick the writer of?  
 Can you describe movie Blade Runner in a few words?

**Table 1:** WIKIMOVIES: Questions, Doc, KB and IE sources.

# MEMORY REPRESENTATIONS

- Sentence-level
  - Key & Value are both identical BOW vector representations of each sentence
- Window-level
  - Key: window of  $n$  words (as BOW)
  - Value: (entity) word in center of window
- Window-level + Title
  - Key: window of  $n$  words + document title (as BOW)
  - Value: document title

Memory Representation	Doc
Sentence-level	52.4
Window-level	66.8
Window-level + Title	74.1
Window-level + Center Encoding + Title	<b>76.9</b>

**Table 3:** Development set performance (% hits@1) with different document memory representations for KV-MemNNs.

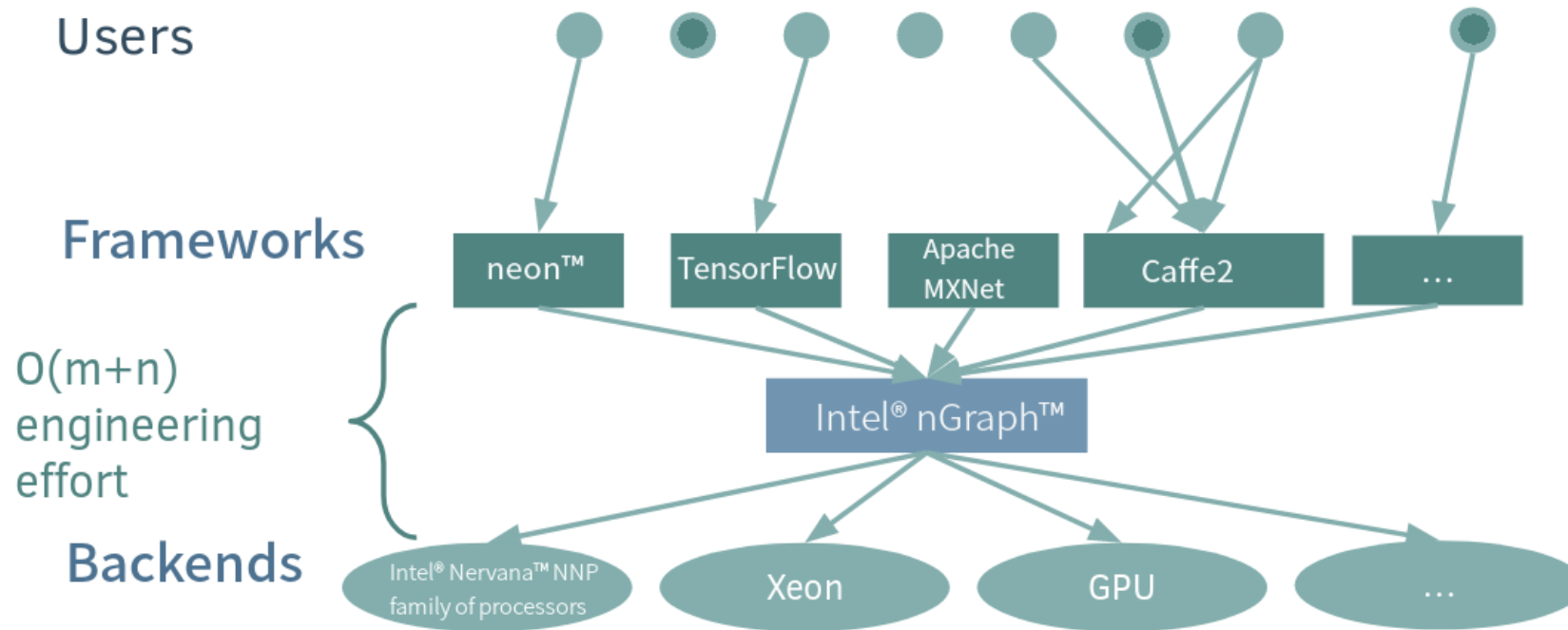
# **NGRAPH™ OVERVIEW AND MEMORY-NETWORK MODEL BUILDING BLOCKS**





# INTEL® NGRAPH™

- An open source library for developers of Deep Learning systems
- In-memory intermediate representation (IR) for deep learning



# MORE INFORMATION ON NGRAPH™ PYTHON

- Python Wrapper Github Repository
  - <https://github.com/NervanaSystems/ngraph-python>
- Documentation
  - <http://ngraph.nervanasys.com/docs/legacy/>

# TENSOR

- Attributes:
- dtype: the type of the elements
- rank: number of dimensions
- shape: an n-tuple of non-negative integers. The length of the tuple is the rank
- All weight matrices, inputs, outputs, and internal states are represented as Tensors.

```
# Initialize projection matrix between internal model states  
self.R_proj = ng.variable(axes=[self.embedding_axis, self.embedding_axis_proj],  
                           initial_value=init)
```

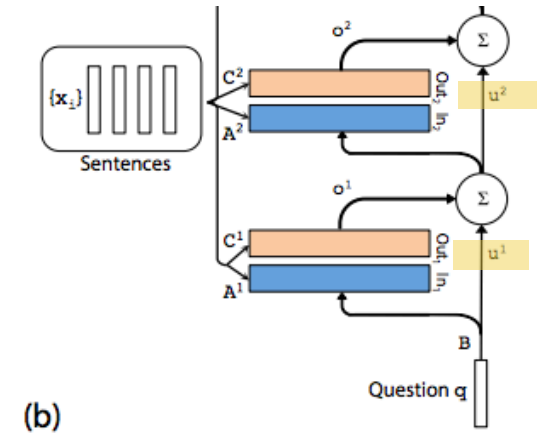
# AXES

- Labels a dimension (and size of dimension) of a tensor
- Convenient – dimensions of tensors can be specified w/o calculating lengths
- Safe – Prevents interference between distinct axes with same lengths
- Generic – Order of axes does for multidimensional tensor does not imply specific data layout or striding

```
# Make axes  
self.batch_axis = ng.make_axis(length=batch_size, name='N')  
self.sentence_rec_axis = ng.make_axis(length=max_utt_len, name='REC')  
self.memory_axis = ng.make_axis(length=memory_size, name='memory_axis')  
self.embedding_axis = ng.make_axis(length=emb_size, name='F')  
self.embedding_axis_proj = ng.make_axis(length=emb_size, name='F_proj')  
self.cand_axis = ng.make_axis(length=num_cands, name='cand_axis')  
self.cand_rec_axis = ng.make_axis(length=max_cand_len, name='REC')
```

# FULLY CONNECTED LAYERS

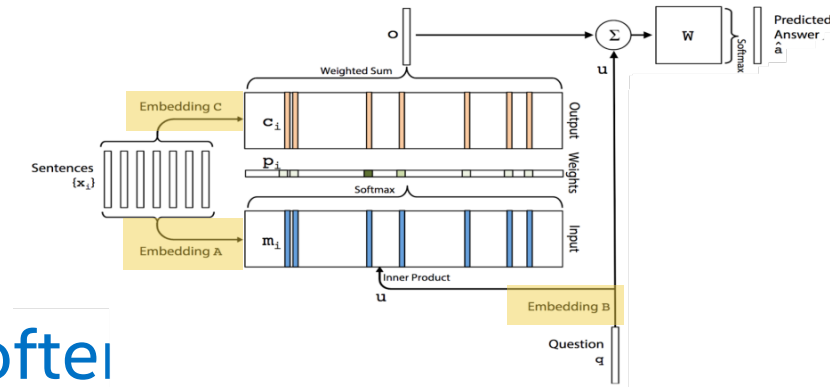
- Used as an intermediate projection in between memory hops.
- `ng.dot()` is used to perform a Matrix-vector multiplication.
- Axes are redefined to match that of the original internal state so they can be added.



```
# Add the output back into the internal state and project
u_k = ng.cast_axes(ng.dot(self.R_proj, o_k), [self.embedding_axis, self.batch_axis]) + u[-1]
```

# LOOKUP TABLE

- Mapping from one object space to another – often word embedding layer
- Decrease dimensionality
- Map from vocab-size one-hot vector to dense embedding-size vector



```
# Weight sharing of A's accross all hops input and output
self.LUT_A = ModifiedLookupTable(vocab_size, emb_size, init, update=True, pad_idx=0)
# Use lookuptable W to embed the candidate answers
self.LUT_W = ModifiedLookupTable(vocab_size, emb_size, init, update=True, pad_idx=0)

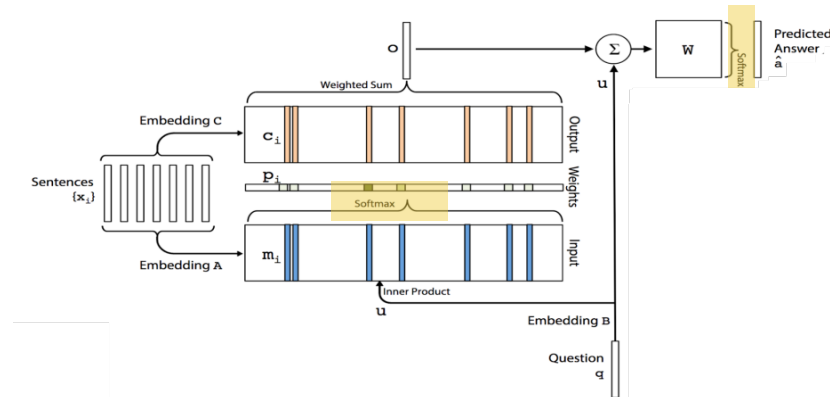
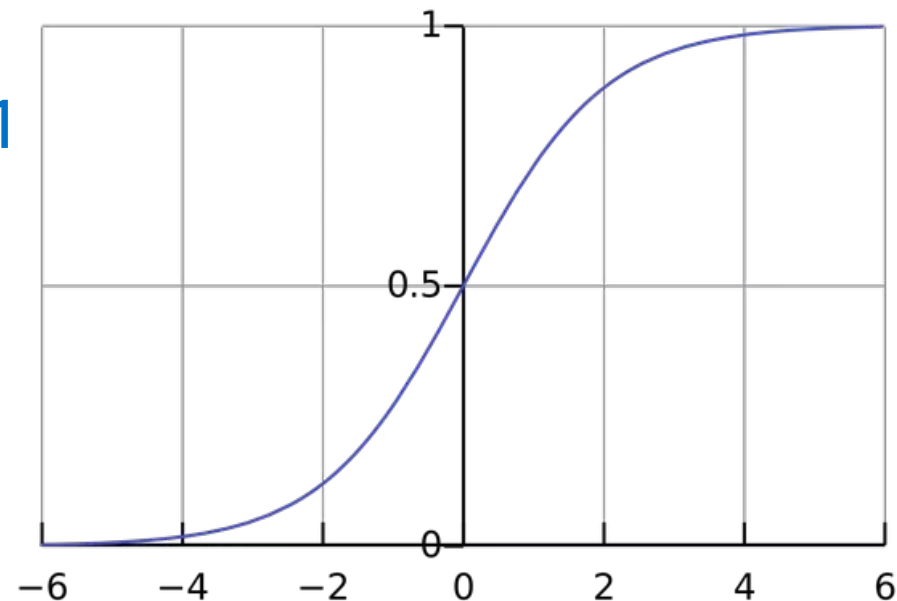
story = ng.cast_axes(inputs['memory'], [
    self.batch_axis, self.memory_axis, self.sentence_rec_axis])

# Re-use the query embedding matrix to embed the memory sentences
# m_A shape: [batch, memory_axis, sentence_axis, F]
m_emb_A = self.LUT_A(story)
m_A = ng.sum(m_emb_A, reduction_axes=[self.sentence_rec_axis])
```

# SOFTMAX

- Useful for multiclass classification
- Generates normalized probability distribution from un-normalized vector input
- All values between 0-1 and sum to 1

$$h_{\theta}(x) = \frac{1}{1 + \exp(-\theta^T x)},$$



```
# rename V to vocab_axis to match answer
a_logits = ng.cast_axes(a_logits, [self.batch_axis, self.cand_axis])
a_pred = ng.softmax(a_logits, self.cand_axis)
```

# END TO END DIALOG SYSTEM DEMO

- **Open your AWS EC2 link**
  - `ip_address.us-west-1.compute.amazonaws.com:8888/tree#running`
  - A notebook and a terminal, these can be accessed under the "Running" tab
  - Password: aidc2018
- **Training:**
  - In the jupyter notebook
- **Inference:**
  - In the terminal
  - `python interactive.py --task 5 --data_dir ../ --model_file memn2n_weights.npz`

```
-----|| Attention
hi 1_TIME <USER> || 0.0
hello what can i help you with today 2_TIME <BOT> || 0.0
can you make a restaurant reservation in madrid 2_TIME <USER> || 0.0
i'm on it 3_TIME <BOT> || 0.0
<SILENCE> 3_TIME <USER> || 0.0
any preference on a type of cuisine 4_TIME <BOT> || 0.0
french please 4_TIME <USER> || 0.0
how many people would be in your party 5_TIME <BOT> || 0.0
for four please 5_TIME <USER> || 0.0
which price range are looking for 6_TIME <BOT> || 0.0
in a cheap price range please 6_TIME <USER> || 0.0
ok let me look into some options for you 7_TIME <BOT> || 0.0
<SILENCE> 7_TIME <USER> || 0.0
api_call french madrid four cheap 8_TIME <BOT> || 0.0
resto_madrid_cheap_french_7stars R_number four 8_TIME <USER> || 0.0
resto_madrid_cheap_french_8stars R_cuisine french 9_TIME <USER> || 0.0
resto_madrid_cheap_french_8stars R_location madrid 10_TIME <USER> || 0.0
resto_madrid_cheap_french_8stars R_price cheap 11_TIME <USER> || 0.0
resto_madrid_cheap_french_8stars R_rating 8 12_TIME <USER> || 0.0
resto_madrid_cheap_french_8stars R_phone resto_madrid_cheap_french_8stars_phone 13_TIME <USER> || 0.0
resto_madrid_cheap_french_8stars R_address resto_madrid_cheap_french_8stars_address 14_TIME <USER> || 0.0
resto_madrid_cheap_french_8stars R_number four 15_TIME <USER> || 0.0
resto_madrid_cheap_french_5stars R_cuisine french 16_TIME <USER> || 0.0
resto_madrid_cheap_french_5stars R_location madrid 17_TIME <USER> || 0.0
resto_madrid_cheap_french_5stars R_price cheap 18_TIME <USER> || 0.0
resto_madrid_cheap_french_5stars R_rating 5 19_TIME <USER> || 0.0
resto_madrid_cheap_french_5stars R_phone resto_madrid_cheap_french_5stars_phone 20_TIME <USER> || 0.0
resto_madrid_cheap_french_5stars R_address resto_madrid_cheap_french_5stars_address 21_TIME <USER> || 0.0
resto_madrid_cheap_french_5stars R_number four 22_TIME <USER> || 0.0
resto_madrid_cheap_french_7stars R_cuisine french 23_TIME <USER> || 0.0
resto_madrid_cheap_french_7stars R_location madrid 24_TIME <USER> || 0.0
resto_madrid_cheap_french_7stars R_price cheap 25_TIME <USER> || 0.0
resto_madrid_cheap_french_7stars R_rating 7 26_TIME <USER> || 0.0
resto_madrid_cheap_french_7stars R_phone resto_madrid_cheap_french_7stars_phone 27_TIME <USER> || 0.0
resto_madrid_cheap_french_7stars R_address resto_madrid_cheap_french_7stars_address 28_TIME <USER> || 0.0
<SILENCE> 29_TIME <USER> || 0.0
what do you think of this option: resto_madrid_cheap_french_7stars 31_TIME <BOT> || 0.0
let's do it 31_TIME <USER> || 0.0
great let me do the reservation 32_TIME <BOT> || 0.0
do you have its address 32_TIME <USER> || 0.0
here it is resto_madrid_cheap_french_7stars_address 33_TIME <BOT> || 0.0
thanks 33_TIME <USER> || 0.0
is there anything i can help you with 35_TIME <BOT> || 1.0
no 35_TIME <USER> || 0.0
-----||
```



# NIPS 2018 – CONVAI2

- The Conversational Intelligence Challenge 2
  - Develop a conversational agent based on the Persona-Chat dataset<sup>1</sup>
- Key-Value Memory networks are an excellent baseline!
- <http://convai.io/>

Persona 1	Persona 2
I like to ski	I am an artist
My wife does not like me anymore	I have four children
I have went to Mexico 4 times this year	I recently got a cat
I hate Mexican food	I enjoy walking for exercise
I like to eat cheetos	I love watching Game of Thrones

[PERSON 1:] Hi  
[PERSON 2:] Hello ! How are you today ?  
[PERSON 1:] I am good thank you , how are you.  
[PERSON 2:] Great, thanks ! My children and I were just about to watch Game of Thrones.  
[PERSON 1:] Nice ! How old are your children?  
[PERSON 2:] I have four that range in age from 10 to 21. You?  
[PERSON 1:] I do not have children at the moment.  
[PERSON 2:] That just means you get to keep all the popcorn for yourself.  
[PERSON 1:] And Cheetos at the moment!  
[PERSON 2:] Good choice. Do you watch Game of Thrones?  
[PERSON 1:] No, I do not have much time for TV.  
[PERSON 2:] I usually spend my time painting: but, I love the show.

Example dialog from the PERSONA-CHAT dataset. Person 1 is given their own persona (top left) at the beginning of the chat, but does not know the persona of Person 2, and vice-versa. They have to get to know each other during the conversation.

<sup>1</sup> [arXiv:1801.07243](https://arxiv.org/abs/1801.07243)

# CONVAI2 DEMO

- <https://github.com/DeepPavlov/convai>
- You'll find simple integration with the ParlAI dataset interface and simple baseline in ngraph included in the provided code
- Submission deadline September 30th

Method		Profile	Fluency	Engagingness	Consistency	Persona Detection
Model						
Human		Self	4.31(1.07)	4.25(1.06)	4.36(0.92)	0.95(0.22)
<i>Generative PersonaChat Models</i>						
Seq2Seq		None	3.17(1.10)	3.18(1.41)	2.98(1.45)	0.51(0.50)
Profile Memory		Self	3.08(1.40)	3.13(1.39)	3.14(1.26)	0.72(0.45)
<i>Ranking PersonaChat Models</i>						
KV Memory		None	3.81(1.14)	3.88(0.98)	3.36(1.37)	0.59(0.49)
KV Profile Memory		Self	3.97(0.94)	3.50(1.17)	3.44(1.30)	0.81(0.39)
Twitter LM		None	3.21(1.54)	1.75(1.04)	1.95(1.22)	0.57(0.50)
OpenSubtitles 2018 LM		None	2.85(1.46)	2.13(1.07)	2.15(1.08)	0.35(0.48)
OpenSubtitles 2009 LM		None	2.25(1.37)	2.12(1.33)	1.96(1.22)	0.38(0.49)
OpenSubtitles 2009 KV Memory		None	2.14(1.20)	2.22(1.22)	2.06(1.29)	0.42(0.49)

Table 4: **Human Evaluation** of various PERSONA-CHAT models, along with a comparison to human performance, and Twitter and OpenSubtitles based models (last 4 rows), standard deviation in parenthesis.

# QUESTIONS?

- <https://github.com/NervanaSystems/nlp-architect>
- Andy ✉ [andy.a.keller@intel.com](mailto:andy.a.keller@intel.com)
- Anna ✉ [anna.bethke@intel.com](mailto:anna.bethke@intel.com) T: @data\_beth

# NOTICES AND DISCLAIMERS

- Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. Performance tests, such as SYSmark and MobileMark, are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products. For more complete information visit: <http://www.intel.com/performance>.
- Tests document performance of components on a particular test, in specific systems. Differences in hardware, software, or configuration will affect actual performance. Consult other sources of information to evaluate performance as you consider your purchase. For more complete information about performance and benchmark results, visit [www.intel.com/benchmarks](http://www.intel.com/benchmarks).
- Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Performance varies depending on system configuration. No computer system can be absolutely secure. Check with your system manufacturer or retailer or learn more at intel.com.
- The products described may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.
- Intel, the Intel logo, Xeon, Xeon Phi and Nervana are trademarks of Intel Corporation in the U.S. and/or other countries.
- \*Other names and brands may be claimed as the property of others
- © 2018 Intel Corporation. All rights reserved.

