DeepSQA: Understanding Sensor Data via Question Answering

Tianwei Xing, Luis Garcia, Federico Cerutti, Lance Kaplan, Alun Preece, Mani Srivastava

About the authors

Tianwei Xing

- Senior AI Researcher & Engineer @ Samsung Research America

Luis Garcia

- Research computer scientist @ USC (B.S & M.S. @Miami) (Ph.D. Rutgers)

Federico Cerutti

- Research Fellow @ University of Brescia (in Italy)

Lance Kaplan

- Army researcher in ARL’s Networked Sensing and Fusion Branch

Alun Preece

- Professor of Intelligent Systems @ Cardiff University (in UK)

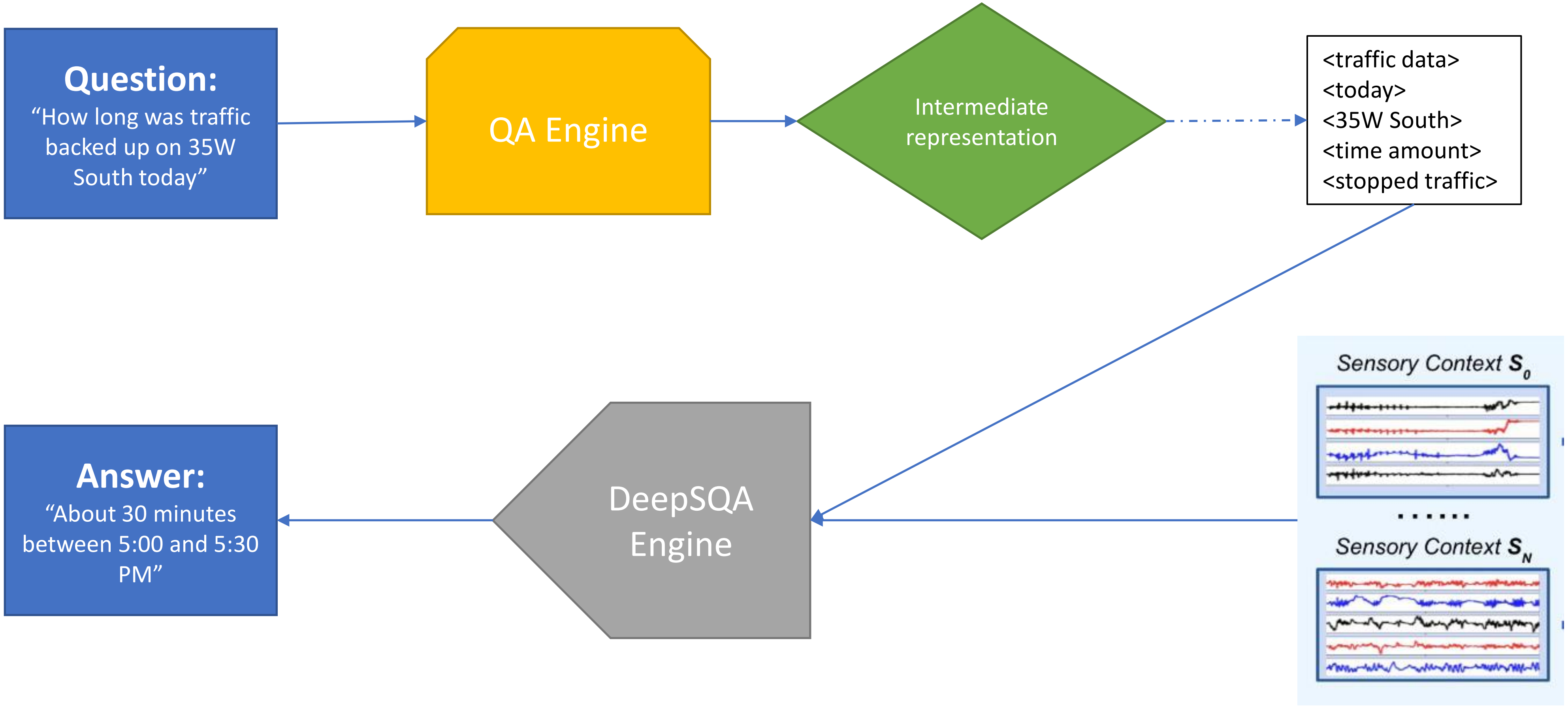
Mani Srivastava

- Professor of ECE @ UCLA

What is SQA?

- Sensory Question Answering
- The ability to ask natural language questions of a system and get a natural language answer in response
  - Based on sensory data collected

High Level View



Question:
“How long was traffic backed up on 35W South today”

QA Engine

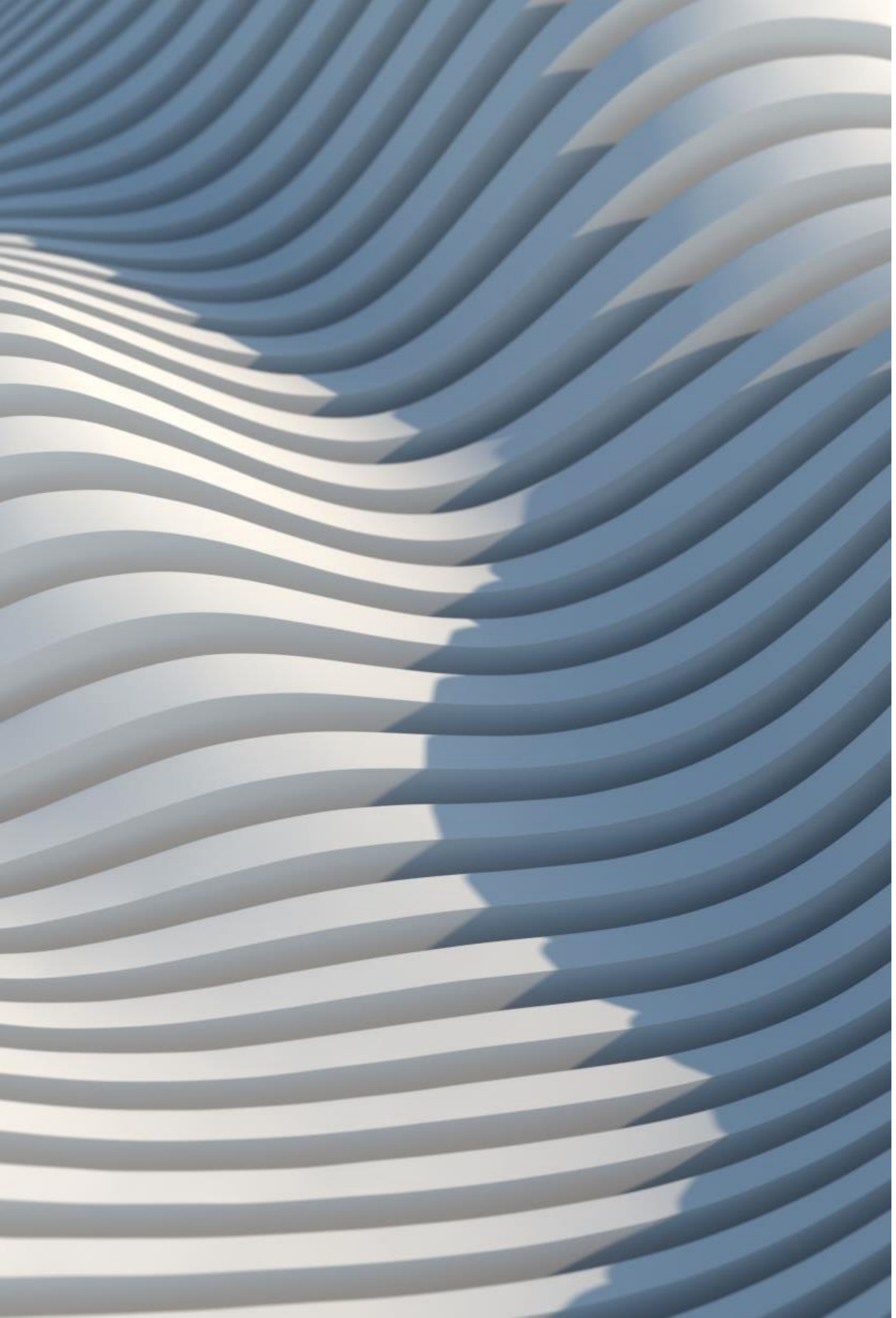
Intermediate representation

<traffic data>
<today>
<35W South>
<time amount>
<stopped traffic>

Sensory Context $\boldsymbol{S_0}$
......
Sensory Context $\boldsymbol{S_N}$

DeepSQA Engine

Answer:
“About 30 minutes between 5:00 and 5:30 PM”



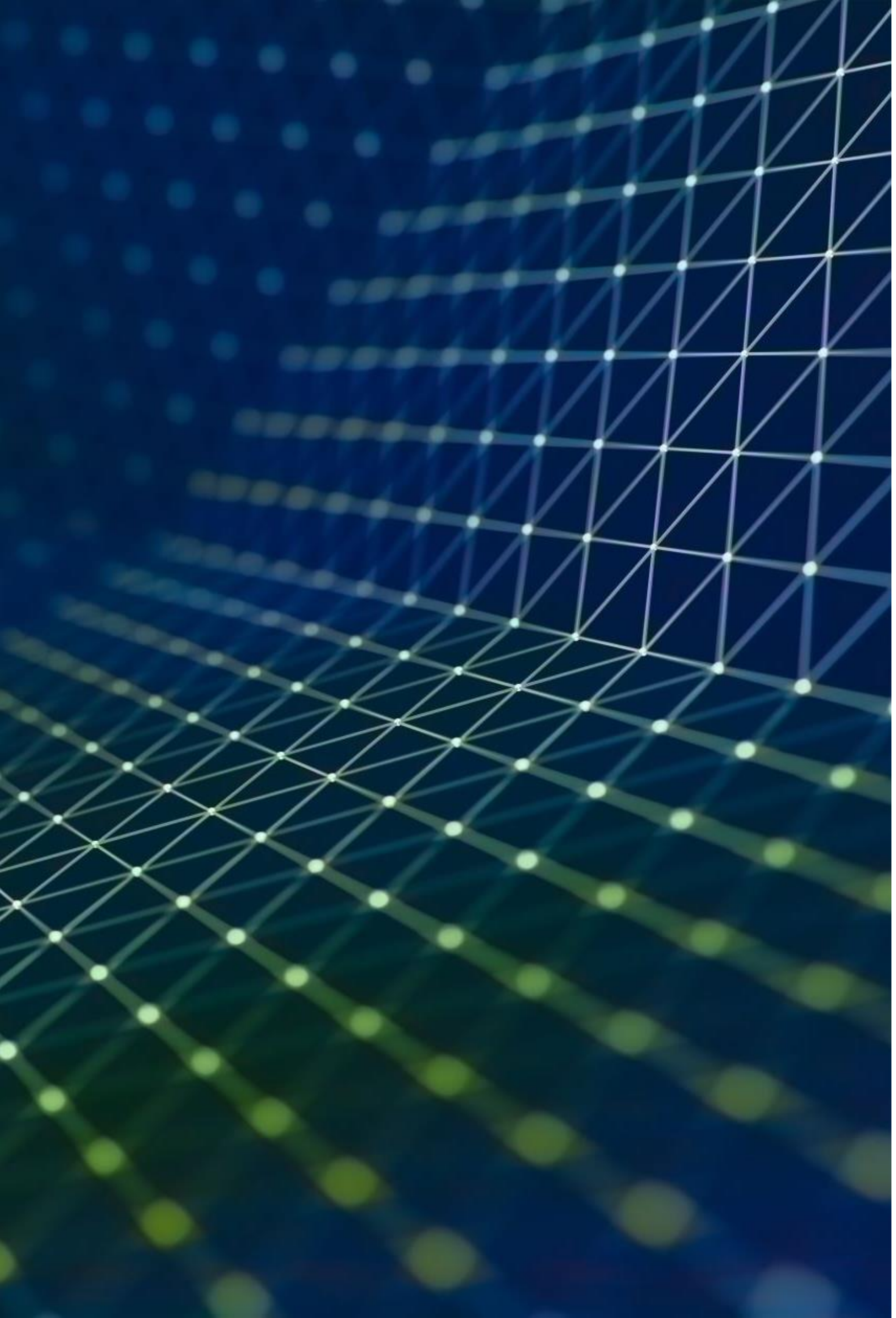
Problem

Massive increase in IoT devices (specifically wearable) have created an influx of sensory data

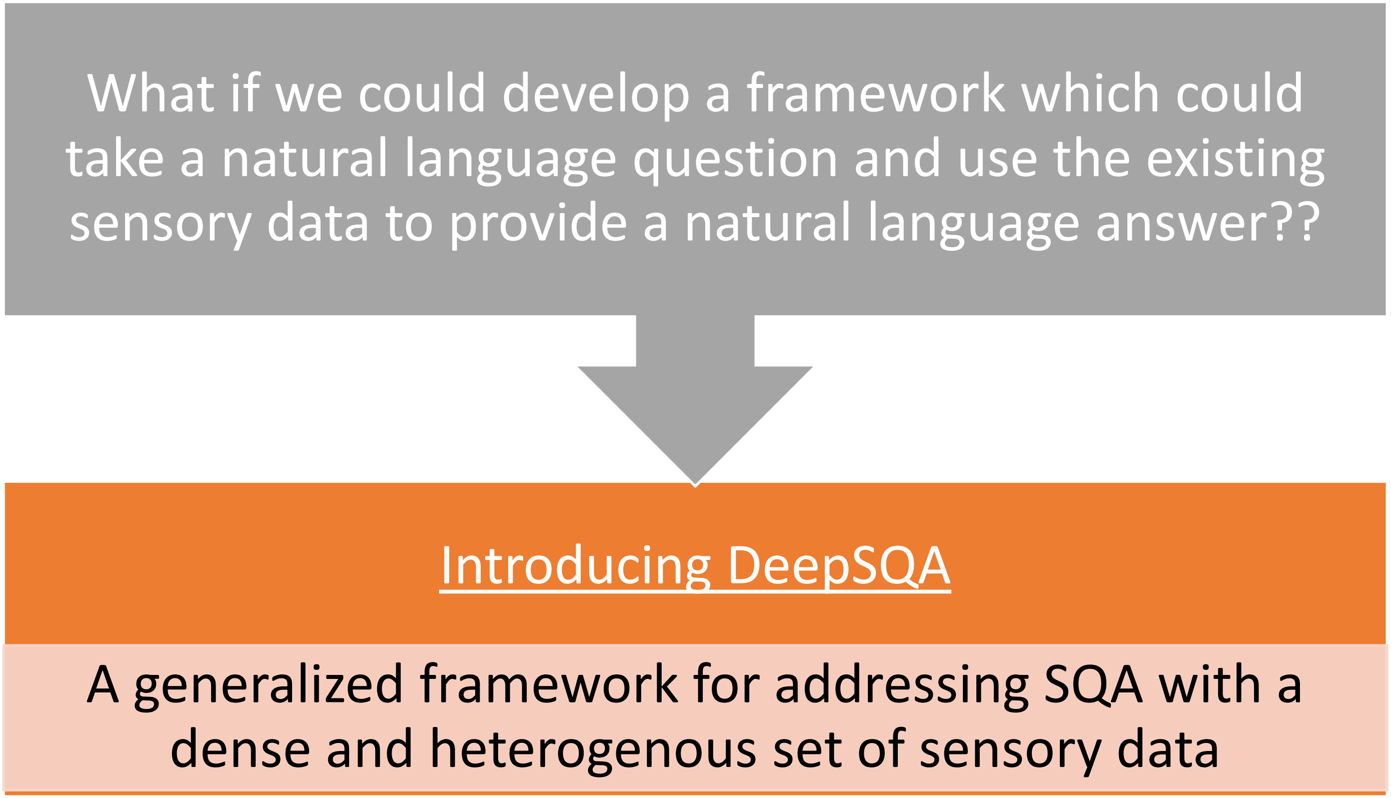
Deep Learning algorithms have done a fantastic job of translating sensory data into useful information to humans

Pre-Trained Deep Learning models cannot adapt to increasingly vast and diverse question set

If there is a new question to ask the data, must train a new deep learning model to answer this one question



DeepSQA



What if we could develop a framework which could take a natural language question and use the existing sensory data to provide a natural language answer??

Introducing DeepSQA

A generalized framework for addressing SQA with a dense and heterogenous set of sensory data

Authors Contribution

- **DeepSQA**:
  - Framework for combining QA engine and sensory data to answer natural language questions with natural language responses
- **SQAgen**:
  - A software framework for generating SQA friendly data sets from labeled sensory data
- **OppQA**:
  - An open source SQA data set that they created that has
    - 1000 sensory contexts
    - 91,000 questions
  - Based on temporal and spatial relationship between sensors

Related Work

Advancement of using machine learning to translate SPECIFIC sensory data into SPECIFIC responses

- *CON* → Requires re-training to answer new questions

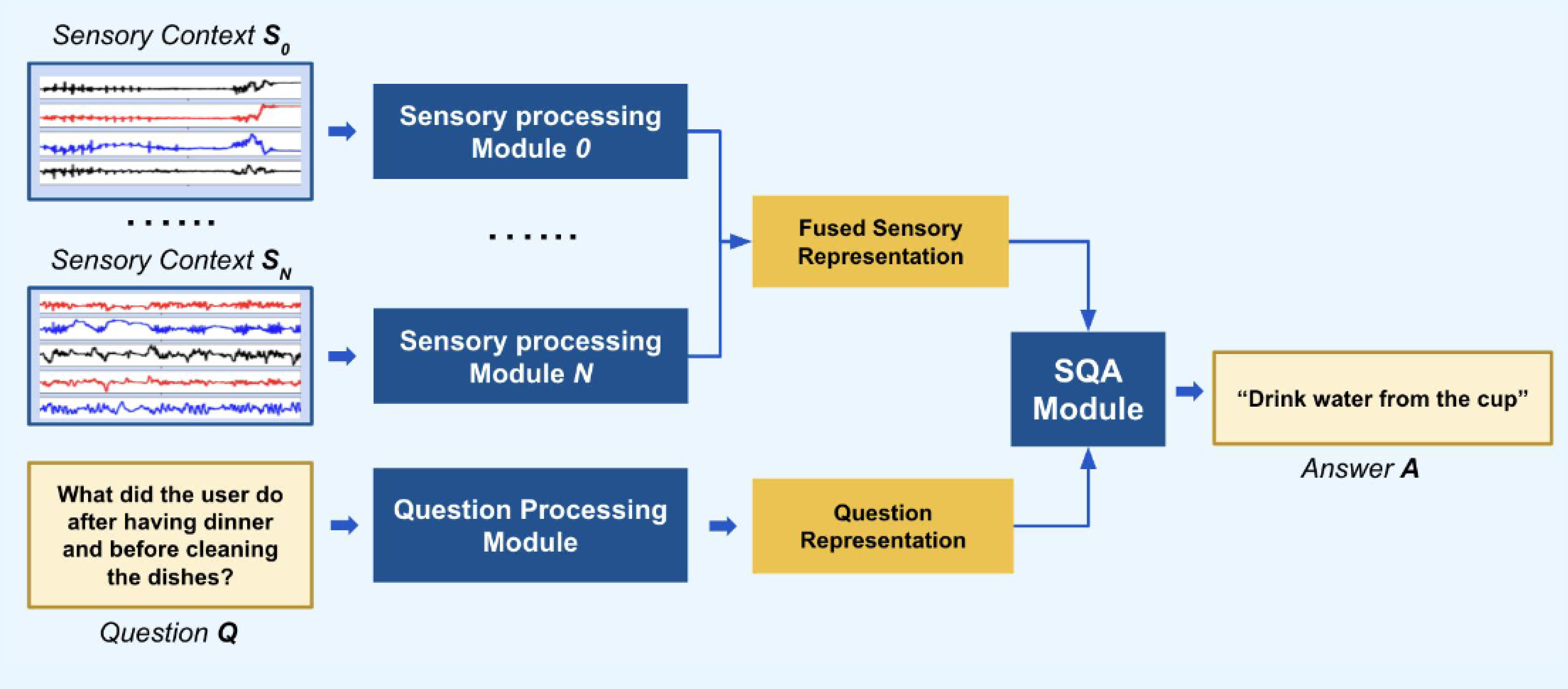
Advancment in different QA engines

- Authors use off the shelf QA engines and test DeepSQA on them

Lots of research human understandable data (audio, visual)

- Authors focus DeepSQA on data not easily interpreted by humans:
  - E.g. inertial sensors

How Does Deep SQA Work?



Sensory Context S_0
......
Sensory Context S_N
Sensory processing Module *0*
......
Sensory processing Module *N*
Fused Sensory Representation
SQA Module
"Drink water from the cup"
Answer **A**
What did the user do after having dinner and before cleaning the dishes?
Question **Q**
Question Processing Module
Question Representation

What is a LSTM?

Long short-term memory Neural Network

Characterized by numerous feedback connections allowing for memory that a typical feed forward DNN does not allow for

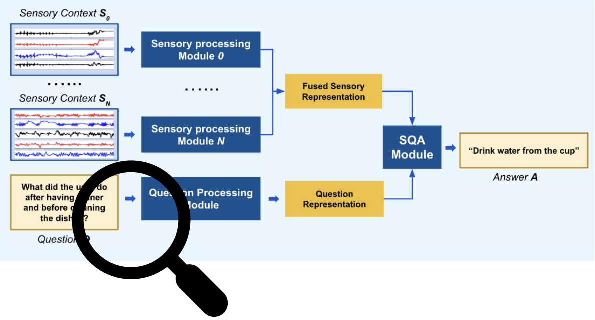
Particularly adapted to learning ORDER DEPENDNECE

Bi-Directional?

This is the concept that the input data through the forward and backwards direction (thus increasing the information available to the DNN at any given time)

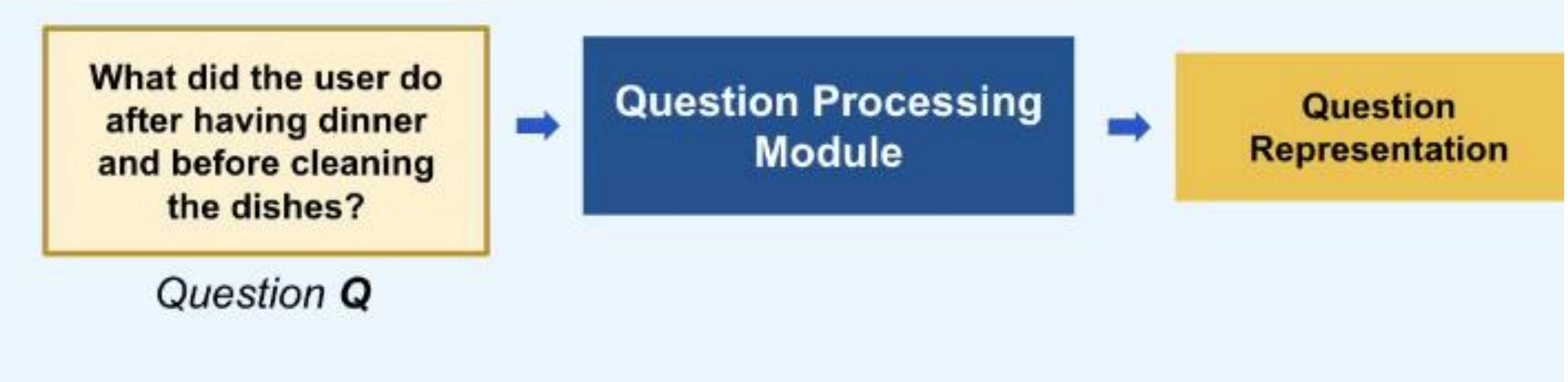
Convolutional? (ConvLSTM)

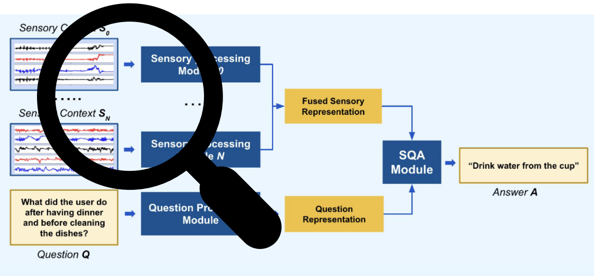
Introduces convolutional layers which provide a number of filters to the data prior to reaching the fully-connected dense network


Sensory Context S_0
Sensory processing Module 0
......
......
Sensory Context S_N
Sensory processing Module N
Fused Sensory Representation
SQA Module
"Drink water from the cup"
Answer A
What did the u do after having ner and before ning the dish ?
Ques on Processing odule
Question Representation
Questio

NL → Question Representation

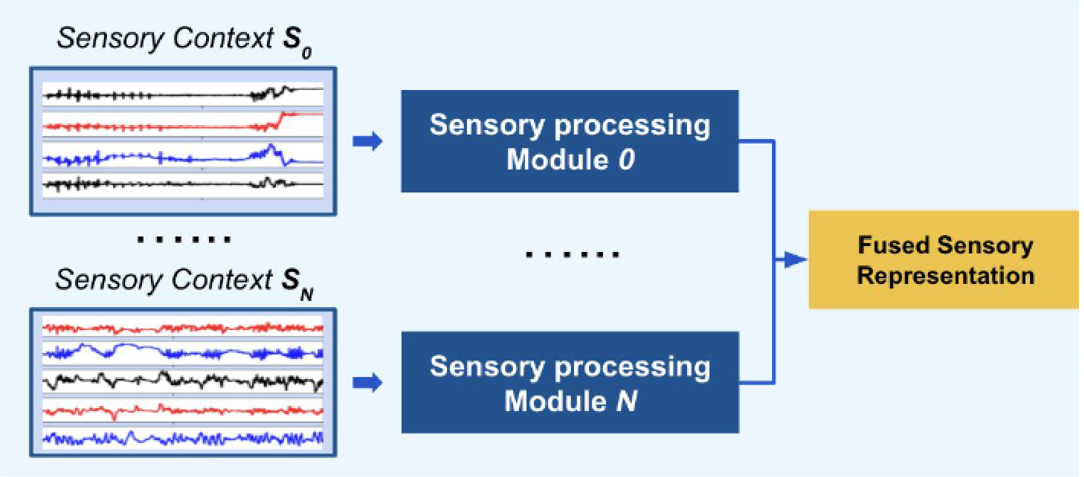
- Question 'q' w/ M words
- Translated to M # of embedding vectors via embedding matrix
- M <embedding vectors> sent through bi-directional LSTM
    - Outputs a "Question Representation" which is a formation of the NL into something the next DNN can read


What did the user do after having dinner and before cleaning the dishes?
Question Q
Question Processing Module
Question Representation

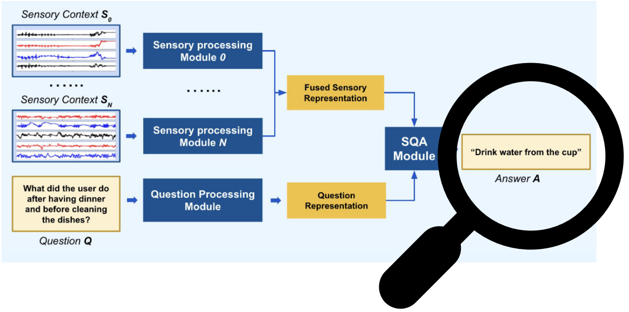


Raw Data → FSR

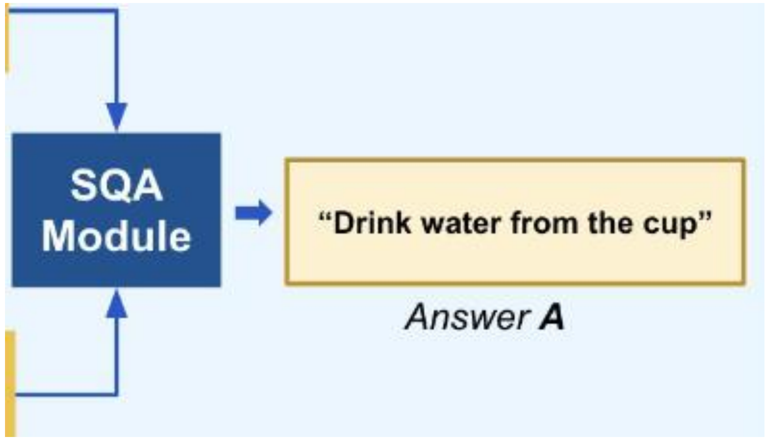
- FSR → Fused Sensory Data Representation
- Helps to:
  - Extract relevant sensory features
  - Reasoning about their long-term temporal dependence
- “Sensory Contexts” created based on their time dependence which can be fused between sensors


Sensory Context S_0
Sensory processing Module 0
Sensory Context S_N
Sensory processing Module N
Fused Sensory Representation

Question + FSR → Answer


Sensory Context S_0
......
Sensory Context S_N
Sensory processing Module *0*
......
Sensory processing Module *N*
Fused Sensory Representation
What did the user do after having dinner and before cleaning the dishes?
Question **Q**
Question Processing Module
Question Representation
SQA Module
"Drink water from the cup"
Answer **A**

- DeepSQA proposes a DNN approach to assimilating FSR and Question data into an answer
- DNN have shown to perform better then human-extracted feature algorithms
- Now we must train the DNN…
  - But wait… PROBLEM!

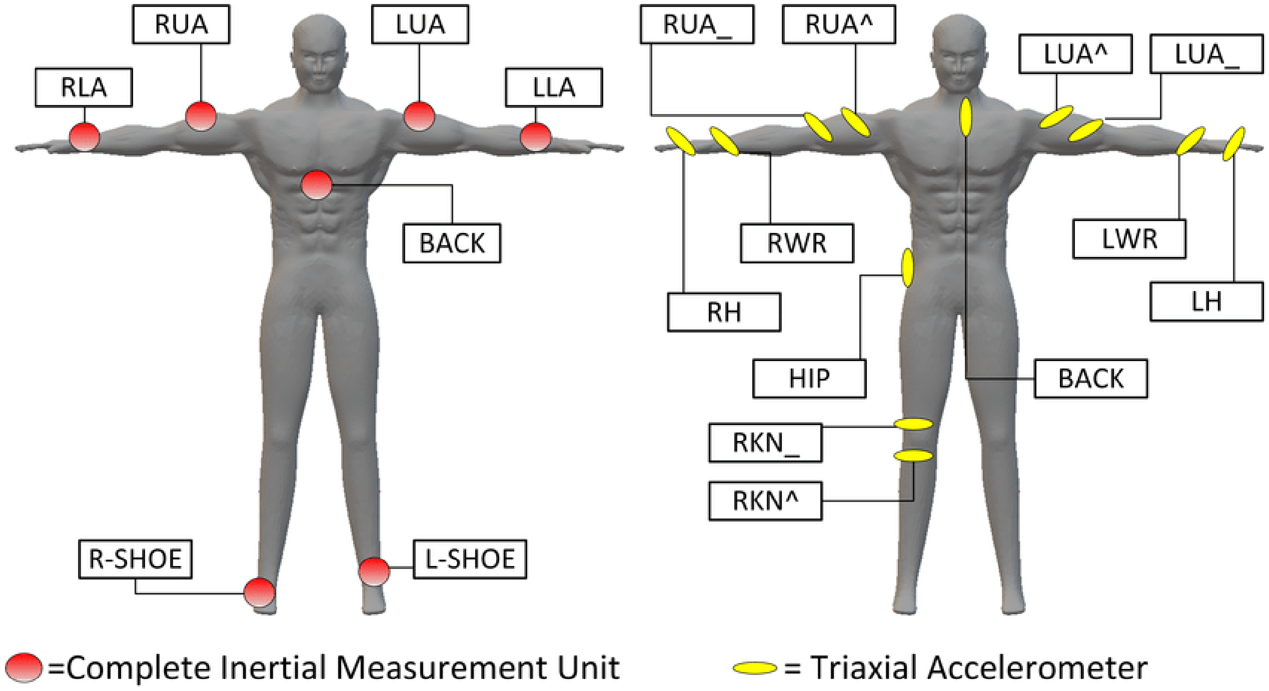


Problem With DNN Training

- Unlike image and audio training, there is not a huge public data set of labeled sensory data
- Maybe we can use crowd sourcing to train a bunch of data
  - WRONG!
  - Humans have an extremely hard time looking at a plot of inertial data and saying… “Oh that person is doing jumping jacks”
- What do we doooooo ☹
  - **SQA-Gen to the rescue!**

OPPURTUNITY Dataset

- Publicly available dataset that was used to benchmark human activity recognition (HAR) tasks
- Provide 7 x 3D accelerometers placed on each human
  - 4 humans studied in 6 separate intervals totaling 8 hours
- Rich annotations provided:
  - **High-level activity** (e.g. “Relaxing”, “coffee time”, “cleanup”)
  - **Locomotion** (e.g. “sit”, “stand”, “walk”, “lie”)
  - **Low-level activity** (e.g. “opening & closing door”, “drinking tea”, etc.



SQA-Gen: SQA data set generation tool

1. **Sensory Context Generation:**
   - Sliding 60 second intervals split the data (w/ 20 second stride)
2. **Scene Representation:**
   - Scene information from Opportunity dataset allows to formulate a scene with words describing activity
3. **Question Generation:**
   - Based on 16 different families (e.g. “action”, “time”, “counting” etc.) using 110 text templates
4. **Question Type and Answering Balance:**
   - They ensured to re-run and balance all question types and length for good training

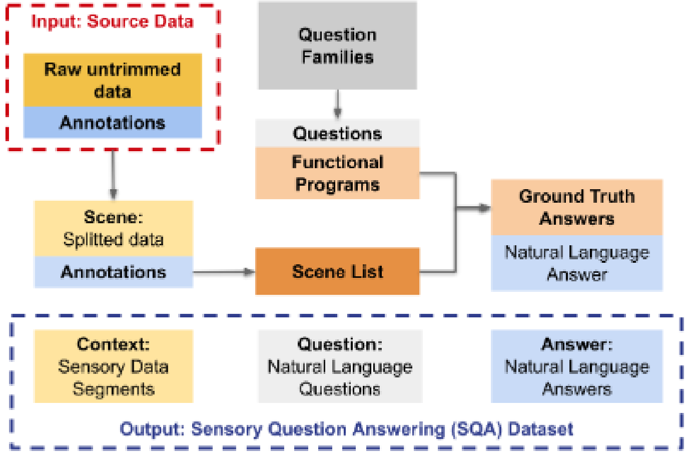
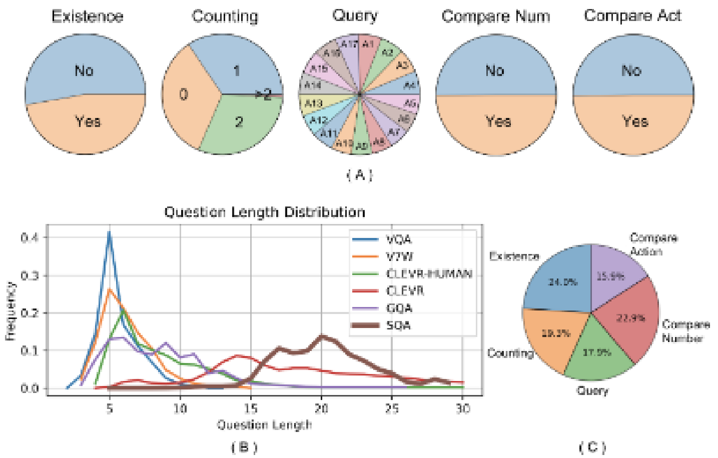


Figure 3: Data generation Pipeline



Example Question:

Natural Language:

- "What did the user do before opening the fridge and after closing the drawer?“

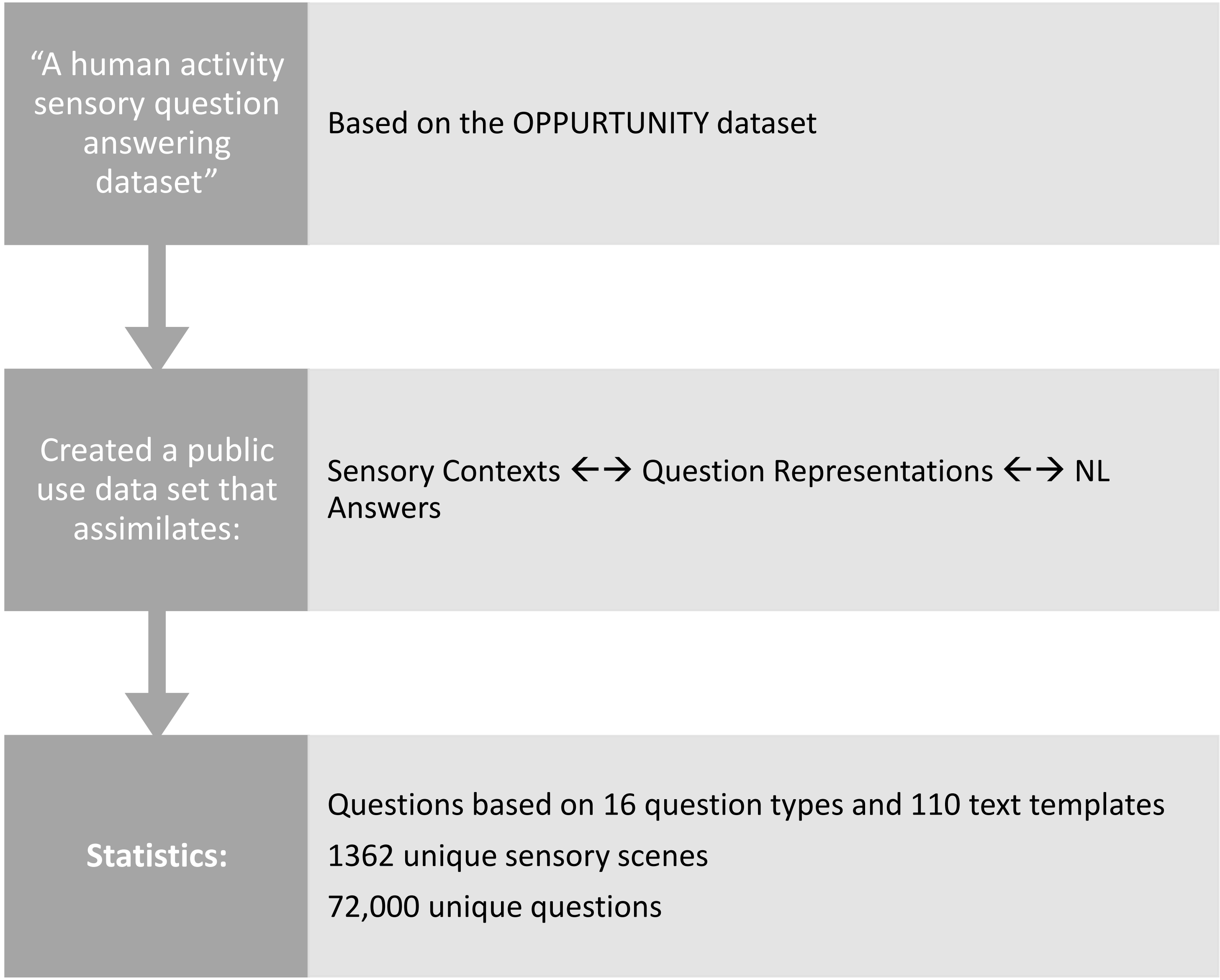
Semantic Breakdown (text template):

- "What did the user do [Relation] [Activity] [Combinator][Relation] [Activity]“

Functional Representation:

- "query_action_type(**AND**(relate(before, open the fridge), relate(after, close the drawer)))"

The birth of OppQA



"A human activity sensory question answering dataset"

Based on the OPPURTUNITY dataset

Created a public use data set that assimilates:

Sensory Contexts ←→ Question Representations ←→ NL Answers

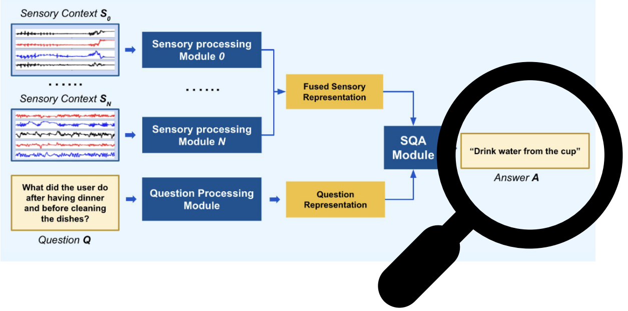
Statistics:

Questions based on 16 question types and 110 text templates

1362 unique sensory scenes

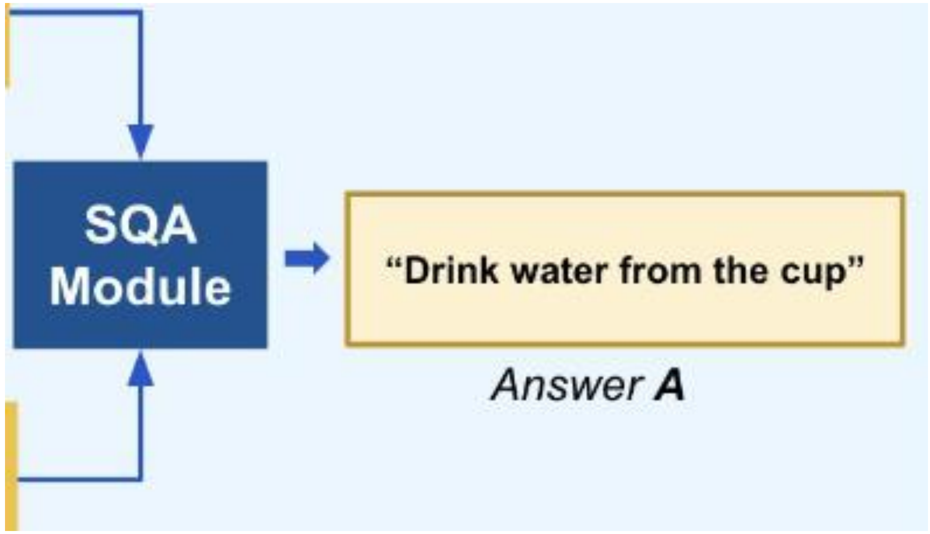
72,000 unique questions

Question + FSR → Answer



Sensory Context S_0
Sensory processing Module 0
......
Sensory Context S_N
Sensory processing Module N
......
Fused Sensory Representation
SQA Module
"Drink water from the cup"
Answer A
What did the user do after having dinner and before cleaning the dishes?
Question Q
Question Processing Module
Question Representation

- Now that we have OppQA to train the SQA Module we can implement, train and evaluate
- SQA Module developed 3 different ways to test multiple state of the ML techniques



Baseline Models Compared Against

Prior:

- Answers only yes or no questions based on question answer training set

PriorQ:

- Predicts the most common answer for questions

LSTM (question only):

- Question data processed using bi-lateral LSTM and answers predicted from that

ConvLSTM (data only):

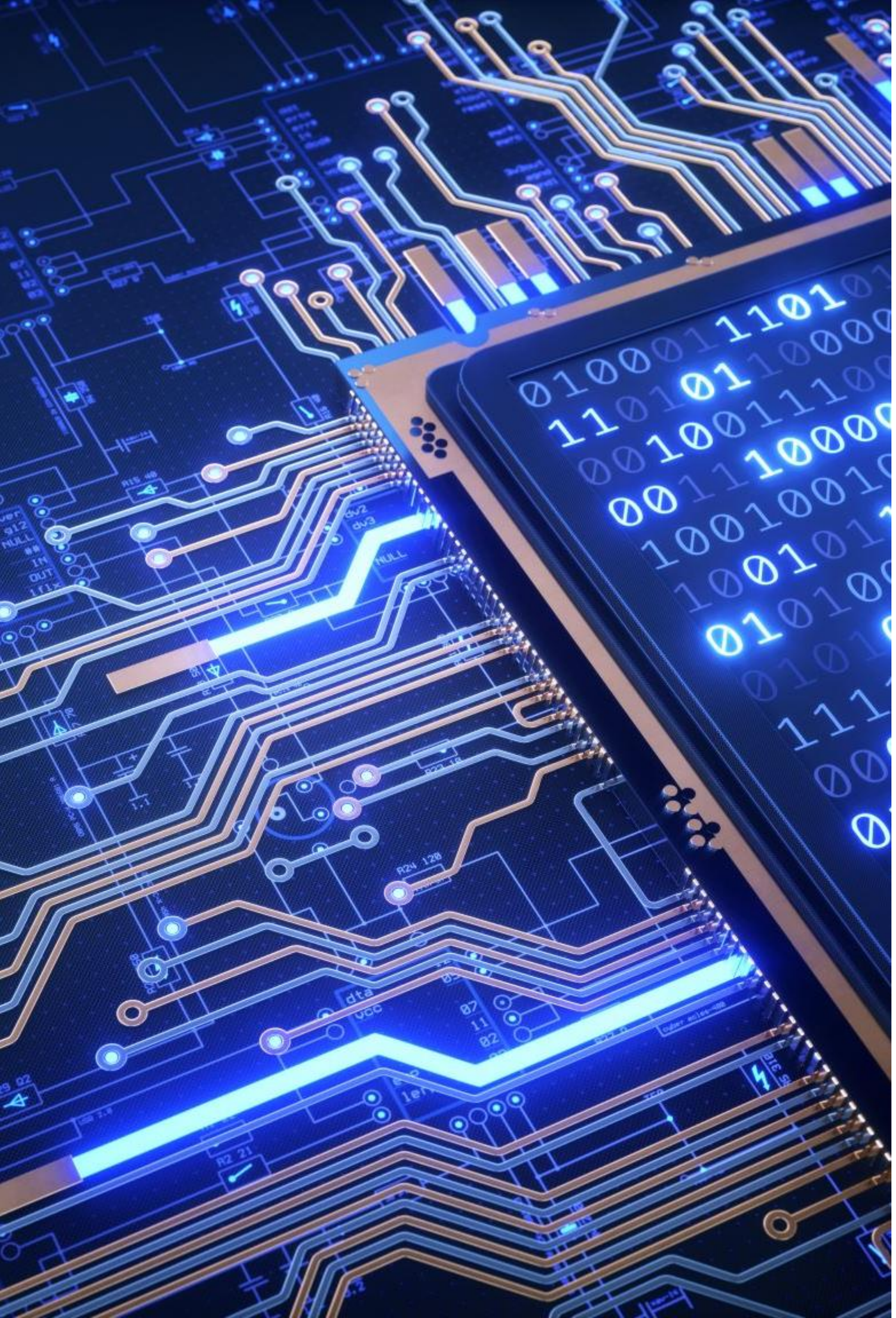
- Sensory context data fed through ConvLSTM and answers predicted from that

Neural Symbolic:

- Use hardcoded logic to combine activities and locomotion with predictions

DeepSQA – 3 Flavors

- Each version:
  - Sends **question** through Bi-Lateral LSTM
  - Sends **data** through convolutional LSTM
- Each version fuses the data differently
- All three versions use a simple 2-layer MLP to relate fused questions/data to answers



Fusing Mechanism:
Simple element wise multiplication

(1) DeepSQA-ConvLSTM

1. Combines the baseline models of LSTM on questions and ConvLSTM on sensory data
2. Both 128 dimensions thus can use element wise multiplication to fuse questions and sensory contexts
3. Fused data sent through a 2-layer MLP to get answer distribution
   - MLP = multi-layer perceptron or basic NN

Fusing Mechanism:
Stacked Attention

(2) DeepSQA-SA

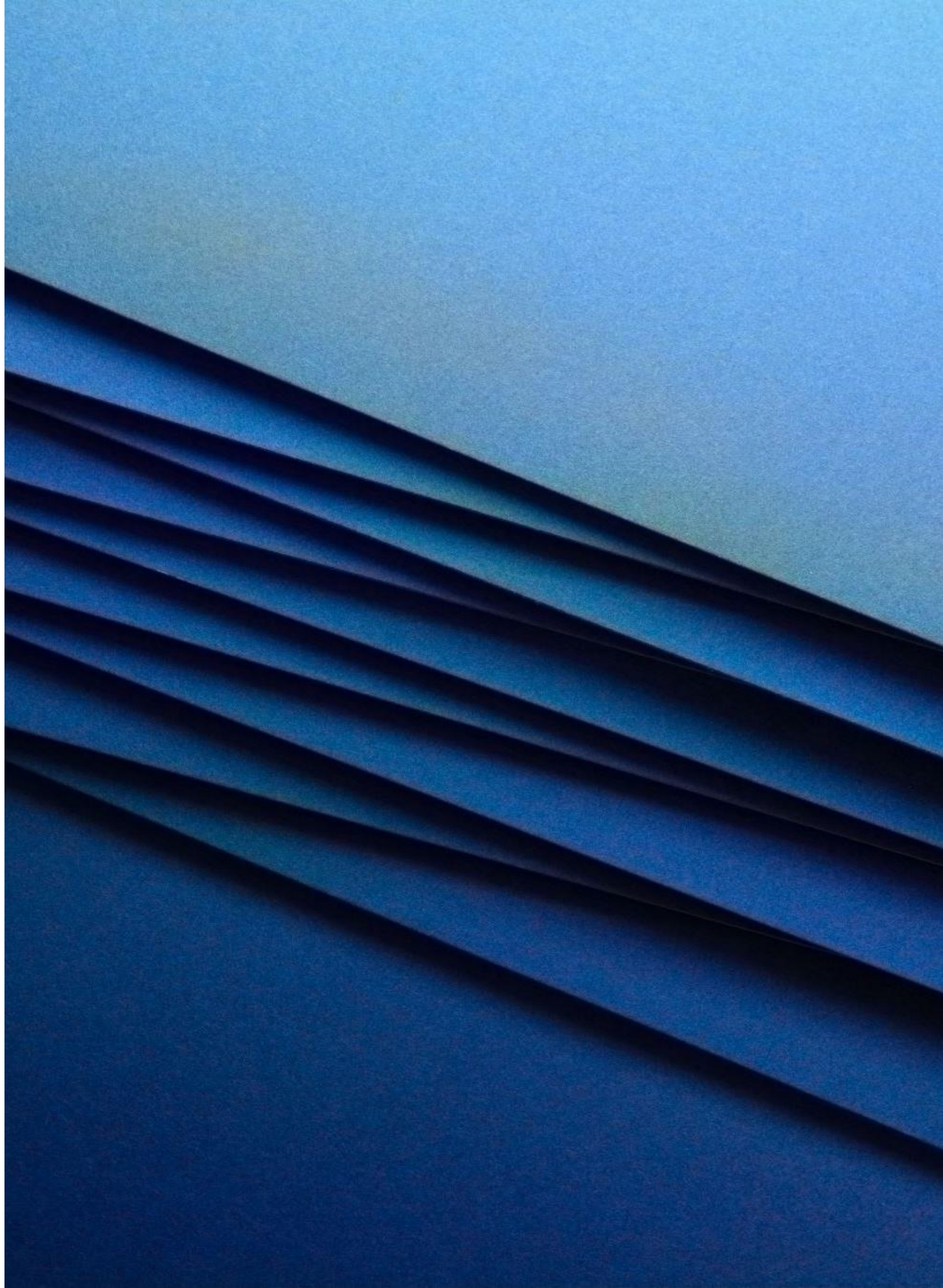
Questions processed through LSTM and sensory contexts processed through ConvLSTM

Stacked Attention Mechanism employed to fuse sensory data and question:

- 2-layer CNN calculates spatial attention weights to multiple the fuses by

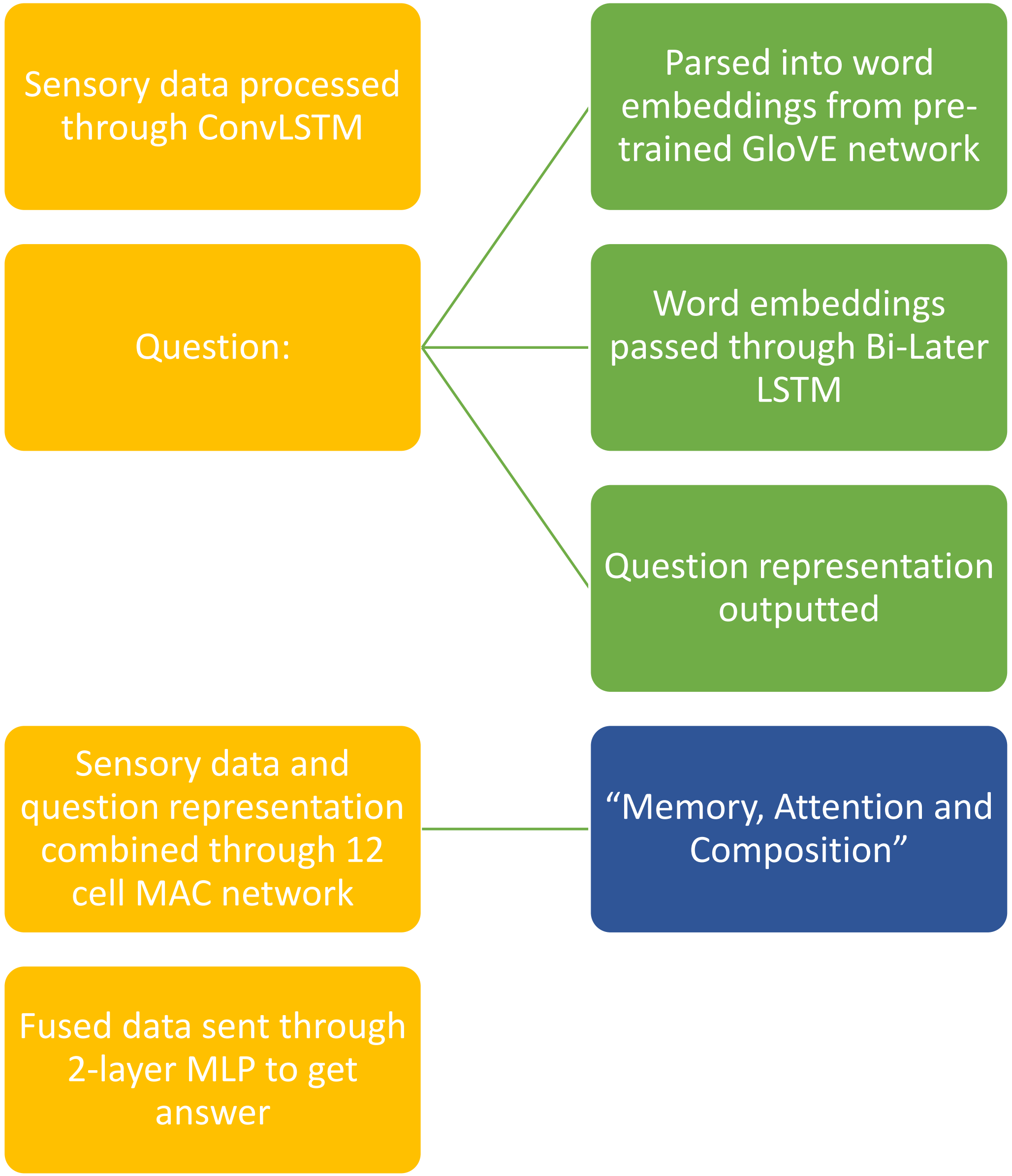
Glimpse number of 2 extracts two sets of different sensory contexts which are concatenated with question representation

Fused data sent through 2-layer MLP to get answer



(3) DeepSQA-CA

Fusing Mechanism:
Compositional Attention



Sensory data processed through ConvLSTM

Question:

Parsed into word embeddings from pre-trained GloVE network

Word embeddings passed through Bi-Later LSTM

Question representation outputted

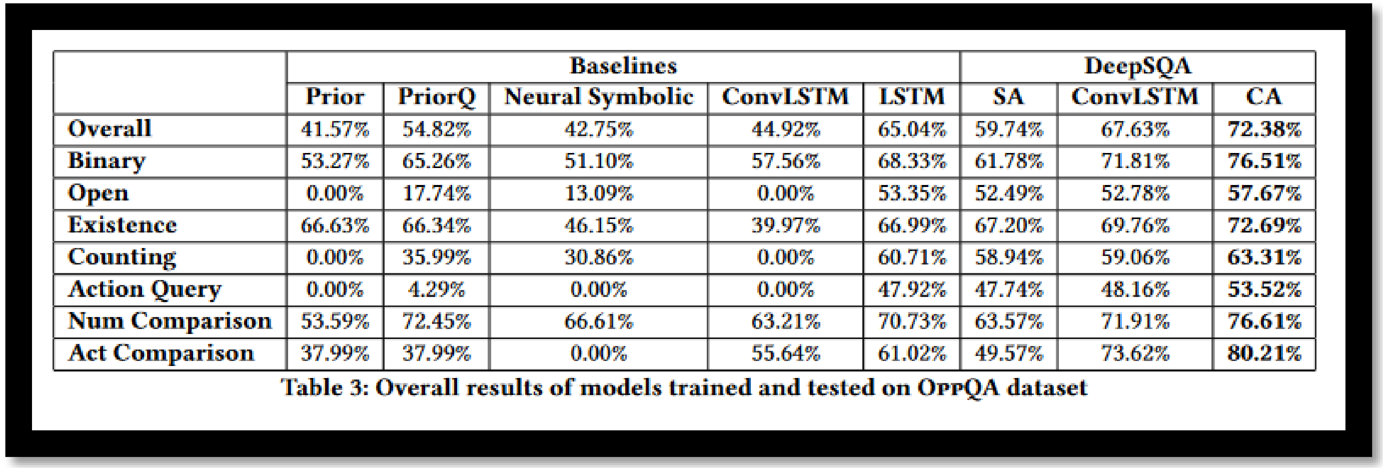
Sensory data and question representation combined through 12 cell MAC network

“Memory, Attention and Composition”

Fused data sent through 2-layer MLP to get answer

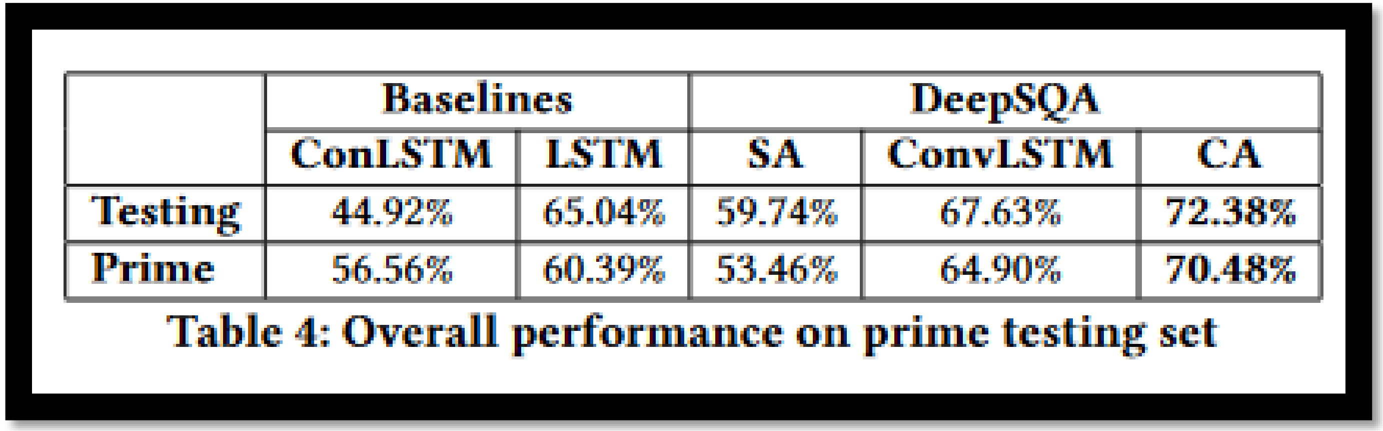
Results

- Deep SQA-ConvLSTM and CA performed the best
- LSTM on just the data performed fairly well



	Baselines					DeepSQA		
	Prior	PriorQ	Neural Symbolic	ConvLSTM	LSTM	SA	ConvLSTM	CA
Overall	41.57%	54.82%	42.75%	44.92%	65.04%	59.74%	67.63%	**72.38%**
Binary	53.27%	65.26%	51.10%	57.56%	68.33%	61.78%	71.81%	**76.51%**
Open	0.00%	17.74%	13.09%	0.00%	53.35%	52.49%	52.78%	**57.67%**
Existence	66.63%	66.34%	46.15%	39.97%	66.99%	67.20%	69.76%	**72.69%**
Counting	0.00%	35.99%	30.86%	0.00%	60.71%	58.94%	59.06%	**63.31%**
Action Query	0.00%	4.29%	0.00%	0.00%	47.92%	47.74%	48.16%	**53.52%**
Num Comparison	53.59%	72.45%	66.61%	63.21%	70.73%	63.57%	71.91%	**76.61%**
Act Comparison	37.99%	37.99%	0.00%	55.64%	61.02%	49.57%	73.62%	**80.21%**

Table 3: Overall results of models trained and tested on OppQA dataset

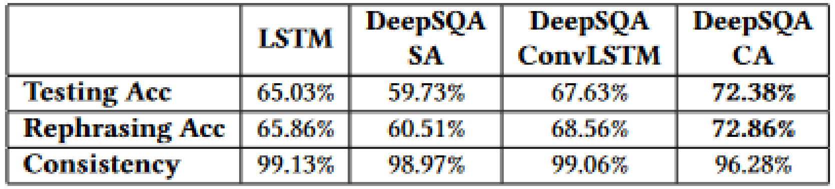


	Baselines		DeepSQA		
	ConLSTM	LSTM	SA	ConvLSTM	CA
Testing	44.92%	65.04%	59.74%	67.63%	**72.38%**
Prime	56.56%	60.39%	53.46%	64.90%	**70.48%**

Table 4: Overall performance on prime testing set

Results

- Re-Phrasing questions demonstrated robustness in the methods, especially the DeepSQA-CA
- The longer the question the less accurate all solutions are



	LSTM	DeepSQA SA	DeepSQA ConvLSTM	DeepSQA CA
Testing Acc	65.03%	59.73%	67.63%	**72.38%**
Rephrasing Acc	65.86%	60.51%	68.56%	**72.86%**
Consistency	99.13%	98.97%	99.06%	96.28%

Table 5: SQA robustness to linguistic variations

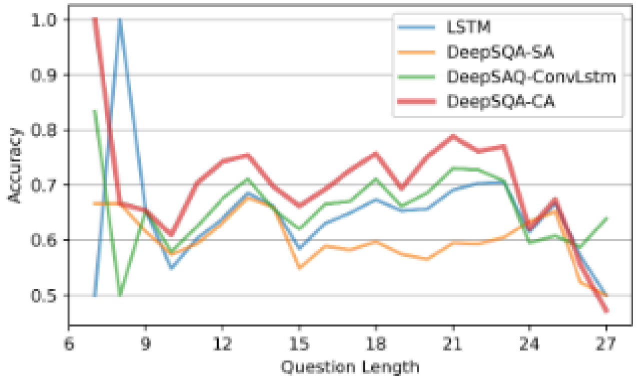
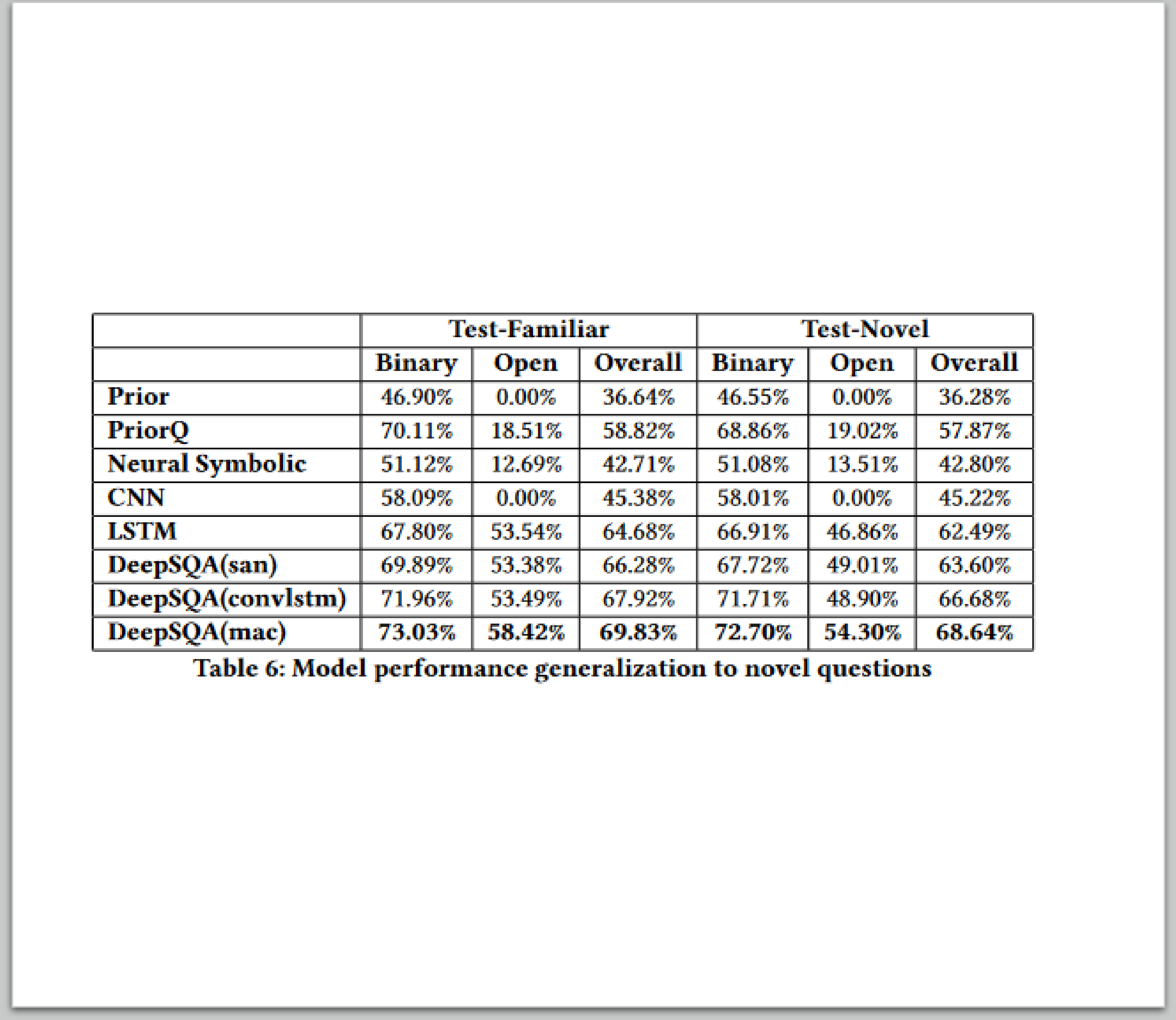


Figure 6: Models performance w.r.t. question length.

Results

- Most systems better at answering “Yes/No” questions
- Accuracy did not drop too much for introducing new questions
- DeepSQA-CA (or MAC) performs the best



	Test-Familiar			Test-Novel		
	Binary	Open	Overall	Binary	Open	Overall
Prior	46.90%	0.00%	36.64%	46.55%	0.00%	36.28%
PriorQ	70.11%	18.51%	58.82%	68.86%	19.02%	57.87%
Neural Symbolic	51.12%	12.69%	42.71%	51.08%	13.51%	42.80%
CNN	58.09%	0.00%	45.38%	58.01%	0.00%	45.22%
LSTM	67.80%	53.54%	64.68%	66.91%	46.86%	62.49%
DeepSQA(san)	69.89%	53.38%	66.28%	67.72%	49.01%	63.60%
DeepSQA(convlstm)	71.96%	53.49%	67.92%	71.71%	48.90%	66.68%
DeepSQA(mac)	**73.03%**	**58.42%**	**69.83%**	**72.70%**	**54.30%**	**68.64%**

Table 6: Model performance generalization to novel questions

Questions

- What other scenarios can you think of to use this system?
  - Other than activity characterization
- They did not discuss latency or compute overhead in this paper?
  - How long do you think answers take to achieve?