Lecture 16:
Dynamic Neural Networks for Question Answering

Christopher Manning and Richard Socher
Can all NLP tasks be seen as question answering problems?
QA Examples

I: Mary walked to the bathroom.
I: Sandra went to the garden.
I: Daniel went back to the garden.
I: Sandra took the milk there.
Q: Where is the milk?
A: garden
I: Everybody is happy.
Q: What's the sentiment?
A: positive
QA Examples

I: Jane has a baby in Dresden.
Q: What are the named entities?
A: Jane - person, Dresden - location
I: Jane has a baby in Dresden.
Q: What are the POS tags?
A: NNP VBZ DT NN IN NNP
I: I think this model is incredible.
Q: In French?
A: Je pense que ce modèle est incroyable.
Goal

A joint model for general QA
First Major Obstacle

- For NLP no single model **architecture** with consistent state of the art results across tasks

<table>
<thead>
<tr>
<th>Task</th>
<th>State of the art model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Question answering (babI)</td>
<td>Strongly Supervised MemNN (Weston et al. 2015)</td>
</tr>
<tr>
<td>Sentiment Analysis (SST)</td>
<td>Tree-LSTMs (Tai et al. 2015)</td>
</tr>
<tr>
<td>Part of speech tagging (PTB-WSJ)</td>
<td>Bi-directional LSTM-CRF (Huang et al. 2015)</td>
</tr>
</tbody>
</table>
Second Major Obstacle

- Fully joint multitask learning* is hard:
  - Usually restricted to lower layers
  - Usually helps only if tasks are related
  - Often hurts performance if tasks are not related

* meaning: same decoder/classifier and not only transfer learning
Dynamic Memory Networks
An architecture for any QA task
High level idea for harder questions

- Imagine having to read an article, memorize it, then get asked various questions → Hard!
- You can't store everything in working memory
- **Optimal**: give you the input data, give you the question, allow as many glances as possible
Dynamic Memory Network

Episodic Memory Module

Semantic Memory Module

Answer module

Input Module

Question Module

Mary got the mail there.
John moved to the bedroom.
Susan went back to the kitchen.
Mary ran to the hallway.
John put the football there.
John put down the football.
Mary went to the garden.

Where is the football?
The Modules: Input

Standard GRU. The last hidden state of each sentence is accessible.
Further Improvement: BiGRU
The Modules: Question

\[ q_t = GRU(v_t, q_{t-1}) \]
The Modules: Episodic Memory

\[ h_i^t = g_i^t GRU(s_i, h_{i-1}^t) + (1 - g_i^t) h_{i-1}^t \]

Last hidden state: \( m^t \)
The Modules: Episodic Memory

• Gates are activated if sentence relevant to the question or memory

\[ z_i^t = [s_i \circ q; s_i \circ m^{t-1}; |s_i - q|; |s_i - m^{t-1}|] \]

\[ Z_i^t = W^{(2)} \tanh \left( W^{(1)} z_i^t + b^{(1)} \right) + b^{(2)} \]

\[ g_i^t = \frac{\exp(Z_i^t)}{\sum_{k=1}^{M_i} \exp(Z_k^t)} \]

• When the end of the input is reached, the relevant facts are summarized in another GRU
The Modules: Episodic Memory

- If summary is insufficient to answer the question, repeat sequence over input
The Modules: Answer

\[ a_t = \text{GRU}([y_{t-1}, q], a_{t-1}), \quad y_t = \text{softmax}(W^{(a)} a_t) \]
Related work

- Sequence to Sequence (Sutskever et al. 2014)
- Neural Turing Machines (Graves et al. 2014)
- Teaching Machines to Read and Comprehend (Hermann et al. 2015)
- Learning to Transduce with Unbounded Memory (Grefenstette 2015)
- Structured Memory for Neural Turing Machines (Wei Zhang 2015)
- Memory Networks (Weston et al. 2015)
- End to end memory networks (Sukhbaatar et al. 2015)
Comparison to MemNets

Similarities:
• MemNets and DMNs have input, scoring, attention and response mechanisms

Differences:
• For input representations MemNets use bag of word, nonlinear or linear embeddings that explicitly encode position
• MemNets iteratively run functions for attention and response

• **DMNs show that neural sequence models can be used for input representation, attention and response mechanisms**
  → naturally captures position and temporality
• Enables broader range of applications
Learning Program Embeddings to Propagate Feedback on Student Code

Authors: Chris Piech, Jonathan Huang, Andy Nguyen, Mike Phulsuksombati, Mehran Sahami, Leonidas Guibas.
Published at ICML 2015.
http://www.jmlr.org/proceedings/papers/v37/piech15.pdf

Research Highlight presented by Lisa Wang
Karel, the Robot

An educational visual programming language for beginners, created by Richard E. Pattis at Stanford.
Representing Computer Programs

// Example student solution
function run() {
    // move then loop
    move();
    // the condition is fixed
    while (notFinished()) {
        if (isPathClear()) {
            move();
        } else {
            turnLeft();
        }
        // redundant
        move();
    }
}

Want concise representation of computer program, capturing the intended functionality of the code, even if the program would crash.

Given a pre-condition state, what would the post-condition be after executing the program?
What you already know: Encoding Sentences

On natural language sentences:
→ E.g. train RNN/CNN/Recursive NN on a language modeling task
→ use trained network to create embeddings of sentences

Can we do the same for computer programs?
Encoding and Decoding States

At each node of the tree: small neural network to encode and decode state

Encoder:

$$f_P = \phi(W^{enc} \cdot P + b^{enc})$$

Decoder:

$$\hat{Q} = \psi(W^{dec} \cdot f_Q + b^{dec})$$

$$\quad = \psi(W^{dec} \cdot M_A \cdot f_P + b^{dec})$$

Figure 2. Diagram of the model for a program $A$ implementing a simple “step forward” behavior in a small 1-dimensional gridworld. Two of the $k$ Hoare triples that correspond with $A$ are shown. Typical worlds are larger and programs are more complex.
Objective Loss Function

$$L(\Theta) = \frac{1}{n} \sum_{i=1}^{n} \ell^{pred}(Q_i, \hat{Q}_i(P_i, A_i; \Theta)) + \frac{1}{n} \sum_{i=1}^{n} \ell^{auto}(P_i, \hat{P}_i(P_i, \Theta)) + \frac{\lambda}{2} \mathcal{R}(\Theta),$$

$\ell^{pred}$ measures how well the model is doing on predicting post-conditions

$\ell^{auto}$ quantifies quality of encoder / decoder on reconstructing provided pre-conditions
Recursive Neural Network to Generate Program Embeddings

// Example student solution
function run() {
    // move then loop
    move();
    // the condition is fixed
    while (!notFinished()) {
        if (isPathClear()) {
            move();
        } else {
            turnLeft();
        }
    }
    // redundant
    move();
}

Note: Programs already have inherent tree structure, so no additional parsing necessary!

Training task: Starting at the leaves, predict the post-condition given pre-condition and the program at that node
Summary

• Paper presented a neural network method to **encode programs** as mapping from precondition space to postcondition space, using **recursive neural nets**
• Learned representations can be used for other tasks, e.g.:
  – Cluster students by program similarity
  – Predict feedback
  – Perform knowledge tracing over multiple code submissions.
Application (ongoing research):
Knowledge Tracing over Program Submissions

- Understand a student’s knowledge over time while she is solving a programming challenge (potentially with intermediate submissions)

- Predict/suggest interventions:
  - Hint
  - Instructional video
  - Motivational video
  - Choice of next exercise

https://studio.code.org/hoc/18
Training objective: Given the sequence of program embeddings, predict future student performance.

Program embeddings

Sequence of program submissions of a single student on the same exercise

Predictions (at every timestep $t$, or only last $t$)

Recurrent hidden LSTM layer

At last time step, hidden layer contains representation for the input sequence
babl 1k, with gate supervision

<table>
<thead>
<tr>
<th>Task</th>
<th>MemNN</th>
<th>DMN</th>
<th>Task</th>
<th>MemNN</th>
<th>DMN</th>
</tr>
</thead>
<tbody>
<tr>
<td>1: Single Supporting Fact</td>
<td>100</td>
<td>100</td>
<td>11: Basic Coreference</td>
<td>100</td>
<td>99.9</td>
</tr>
<tr>
<td>2: Two Supporting Facts</td>
<td>100</td>
<td>98.2</td>
<td>12: Conjunction</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>3: Three Supporting facts</td>
<td>100</td>
<td>95.2</td>
<td>13: Compound Coreference</td>
<td>100</td>
<td>99.8</td>
</tr>
<tr>
<td>4: Two Argument Relations</td>
<td>100</td>
<td>100</td>
<td>14: Time Reasoning</td>
<td>99</td>
<td>100</td>
</tr>
<tr>
<td>5: Three Argument Relations</td>
<td>98</td>
<td>99.3</td>
<td>15: Basic Deduction</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>6: Yes/No Questions</td>
<td>100</td>
<td>100</td>
<td>16: Basic Induction</td>
<td>100</td>
<td>99.4</td>
</tr>
<tr>
<td>7: Counting</td>
<td>85</td>
<td>96.9</td>
<td>17: Positional Reasoning</td>
<td>65</td>
<td>59.6</td>
</tr>
<tr>
<td>8: Lists/Sets</td>
<td>91</td>
<td>96.5</td>
<td>18: Size Reasoning</td>
<td>95</td>
<td>95.3</td>
</tr>
<tr>
<td>9: Simple Negation</td>
<td>100</td>
<td>100</td>
<td>19: Path Finding</td>
<td>36</td>
<td>34.5</td>
</tr>
<tr>
<td>10: Indefinite Knowledge</td>
<td>98</td>
<td>97.5</td>
<td>20: Agent’s Motivations</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>Mean Accuracy (%)</td>
<td>93.3</td>
<td>93.6</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Experiments: Sentiment Analysis

### Stanford Sentiment Treebank

Test accuracies:

- **MV-RNN and RNTN:** Socher et al. (2013)
- **DCNN:** Kalchbrenner et al. (2014)
- **PVec:** Le & Mikolov. (2014)
- **CNN-MC:** Kim (2014)
- **DRNN:** Irsoy & Cardie (2015)
- **CT-LSTM:** Tai et al. (2015)

<table>
<thead>
<tr>
<th>Task</th>
<th>Binary</th>
<th>Fine-grained</th>
</tr>
</thead>
<tbody>
<tr>
<td>MV-RNN</td>
<td>82.9</td>
<td>44.4</td>
</tr>
<tr>
<td>RNTN</td>
<td>85.4</td>
<td>45.7</td>
</tr>
<tr>
<td>DCNN</td>
<td>86.8</td>
<td>48.5</td>
</tr>
<tr>
<td>PVec</td>
<td>87.8</td>
<td>48.7</td>
</tr>
<tr>
<td>CNN-MC</td>
<td>88.1</td>
<td>47.4</td>
</tr>
<tr>
<td>DRNN</td>
<td>86.6</td>
<td>49.8</td>
</tr>
<tr>
<td>CT-LSTM</td>
<td>88.0</td>
<td>51.0</td>
</tr>
<tr>
<td>DMN</td>
<td>88.6</td>
<td>52.1</td>
</tr>
</tbody>
</table>
Analysis of Number of Episodes

• How many attention + memory passes are needed in the episodic memory?

<table>
<thead>
<tr>
<th>Max passes</th>
<th>task 3 three-facts</th>
<th>task 7 count</th>
<th>task 8 lists/sets</th>
<th>sentiment (fine grain)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 pass</td>
<td>0</td>
<td>48.8</td>
<td>33.6</td>
<td>50.0</td>
</tr>
<tr>
<td>1 pass</td>
<td>0</td>
<td>48.8</td>
<td>54.0</td>
<td>51.5</td>
</tr>
<tr>
<td>2 pass</td>
<td>16.7</td>
<td>49.1</td>
<td>55.6</td>
<td>52.1</td>
</tr>
<tr>
<td>3 pass</td>
<td>64.7</td>
<td>83.4</td>
<td>83.4</td>
<td>50.1</td>
</tr>
<tr>
<td>5 pass</td>
<td>95.2</td>
<td>96.9</td>
<td>96.5</td>
<td>N/A</td>
</tr>
</tbody>
</table>
Analysis of Attention for Sentiment

- Sharper attention when 2 passes are allowed.
- Examples that are wrong with just one pass
Analysis of Attention for Sentiment

1-iter DMN (pred: very positive, ans: negative)

2-iter DMN (pred: negative, ans: negative)
Analysis of Attention for Sentiment

• Examples where full sentence context from first pass changes attention to words more relevant for final prediction.
Analysis of Attention for Sentiment

• Examples where full sentence context from first pass changes attention to words more relevant for final prediction
Experiments: POS Tagging

- PTB WSJ, standard splits
- Episodic memory does not require multiple passes, single pass enough
Live Demo

Dynamic Memory Network by MetaMind

Story

Despite the glowing reviews, this movie wasn't an especially surprising or interesting experience.

Question

What is the sentiment?

Run DMN

Get new example
Modularization Allows for Different Inputs

- Episodic Memory
  - Answer: Kitchen
  - Episodic Memory
  - Answer: Palm

- Input Module
  - John moved to the garden.
  - John got the apple there.
  - John moved to the kitchen.
  - Sandra picked up the milk there.
  - John dropped the apple.
  - John moved to the office.

- Question
  - Where is the apple?

- Input Module
  - What kind of tree is in the background?
Input Module for Images
Accuracy: Visual Question Answering

VQA test-dev and test-standard:

- Antol et al. (2015)
- ACK Wu et al. (2015);
- iBOWIMG - Zhou et al. (2015);
- DPPnet - Noh et al. (2015); D-NMN - Andreas et al. (2016);
- SAN - Yang et al. (2015)

<table>
<thead>
<tr>
<th>Method</th>
<th>test-dev All</th>
<th>Y/N</th>
<th>Other</th>
<th>Num</th>
<th>test-std All</th>
</tr>
</thead>
<tbody>
<tr>
<td>VQA Image</td>
<td>28.1</td>
<td>64.0</td>
<td>3.8</td>
<td>0.4</td>
<td>-</td>
</tr>
<tr>
<td>Question</td>
<td>48.1</td>
<td>75.7</td>
<td>27.1</td>
<td>36.7</td>
<td>-</td>
</tr>
<tr>
<td>Q+I</td>
<td>52.6</td>
<td>75.6</td>
<td>37.4</td>
<td>33.7</td>
<td>-</td>
</tr>
<tr>
<td>LSTM Q+I</td>
<td>53.7</td>
<td>78.9</td>
<td>36.4</td>
<td>35.2</td>
<td>54.1</td>
</tr>
<tr>
<td>ACK</td>
<td>55.7</td>
<td>79.2</td>
<td>40.1</td>
<td>36.1</td>
<td>56.0</td>
</tr>
<tr>
<td>iBOWIMG</td>
<td>55.7</td>
<td>76.5</td>
<td>42.6</td>
<td>35.0</td>
<td>55.9</td>
</tr>
<tr>
<td>DPPnet</td>
<td>57.2</td>
<td>80.7</td>
<td>41.7</td>
<td>37.2</td>
<td>57.4</td>
</tr>
<tr>
<td>D-NMN</td>
<td>57.9</td>
<td>80.5</td>
<td>43.1</td>
<td>37.4</td>
<td>58.0</td>
</tr>
<tr>
<td>SAN</td>
<td>58.7</td>
<td>79.3</td>
<td>46.1</td>
<td>36.6</td>
<td>58.9</td>
</tr>
<tr>
<td>DMN+</td>
<td><strong>60.3</strong></td>
<td>80.5</td>
<td>48.3</td>
<td>36.8</td>
<td><strong>60.4</strong></td>
</tr>
</tbody>
</table>
Attention Visualization

What is the main color on the bus? Answer: blue
What type of trees are in the background? Answer: pine
How many pink flags are there? Answer: 2
Is this in the wild? Answer: no
Attention Visualization

Which man is dressed more flamboyantly? Answer: right
Who is on both photos? Answer: girl
What time of day was this picture taken? Answer: night
What is the boy holding? Answer: surfboard
Attention Visualization

What is this sculpture made out of?  Answer: metal

What color are the bananas?  Answer: green

What is the pattern on the cat’s fur on its tail?  Answer: stripes

Did the player hit the ball?  Answer: yes
What is the girl holding?  tennis racket
What is the girl doing?  playing tennis
Is the girl wearing a hat?  yes
What is the girl wearing?  shorts
What is the color of the ground?  brown
What color is the ball?  yellow
What color is her skirt?  white
What did the girl just hit?  tennis ball
Live Demo

Dynamic Memory Network by MetaMind

Question
What color is the building that has a clock?

Run DMN
Summary

• Most NLP tasks can be reduced to QA
• DMN accurately solves variety of QA tasks
• More advanced: Dynamic Coattention Networks