Computational Episodic Memory for Intelligent Agents

Juzheng Zhang, Jianmin Zheng, Nadia Magenet Thalmann

School of Computer Engineering

21/03/2013
Background

- Tulving E. (1972)
  - “Episodic memory is a system that receives and stores information about temporally dated episodes or events, and temporal-spatial relations among them.”
- Relations with Semantic Memory
  - Distinction: Knowing (fact) vs. Remembering (experience)
  - Connection:
Introduction

- **Data Structure**
  - The episodes are constituted by sequence of events, and each event contains several attributes.
Related Works

• Current Trends
  – AI ---> Improve agent performance by learning
  – HCI/HRI ---> Enhance believability of communication

• Main methods
  – Database
  – Sparse Distributed Memory [D'Mello et.al 05]
  – Bayesian Network [Kadlec & Brom 13]
  – Neural Network [Subagdja et.al 12]
  – Connectionist Model [Lim et.al 11]
Issues for Previous works

1. Few works consider the semantic meaning in episodic memory encoding and retrieval
   – Benefits
     • Synonymy & Polysemy  Example
     • Enhance the encoding of information  Example

2. Few works build metric space to easily figure out the relations among episodes (get similarity)
Our Work

• Goal
  – Find the general relations among episodes, events and attributes,
  – Arrange episodes based on these relations.
  – Realize functionalities such as encoding, storage, retrieval, reconstruction, forgetting…
Our Framework

- **Cue**
- **New Episode**
- **Training Episodes**
- **External Semantic Knowledge**

**Memory Space**

- **Episodes**
  - $EP_1$, $EP_2$, ...

- **Event Layer**
  - $EV_1$, $EV_2$, ..., $EV_n$

- **Attributes**
  - Sub, DirObj, IndirObj, Time, Loc, ...

**STM (WM)**

- **Reconstruction**
- **Storage**
- **Retrieval**
- **Forgetting**

**LTM**

- **Retrieval**
- **Forgetting**

**Clustering**
Training & Encoding

• Method
  – Build event space (ev & attr) ⟷ SVD
  – Cluster events ⟷ AP
  – Build episode space (ep & ev) ⟷ SVD
  – Cluster episodes ⟷ AP

|       | Ev1 | Ev2 | Ev3 | Ev4 | Ev5 | ...
|-------|-----|-----|-----|-----|-----|-----
| Sub=Jack | 2   | 2   | 0   | 0   | 2   |
| Pre=give  | 4   | 0   | 4   | 0   | 0   |
| Loc=school | 1   | 1   | 0   | 1   | 0   |
| dirObj=book | 1   | 0   | 1   | 0   | 1   |
| indirObj=Tom | 1   | 1   | 0   | 0   | 0   |
| Sub=Tom    | 0   | 0   | 2   | 2   | 0   |
| ...        |     |     |     |     |     |     |

Entry

• Give important attributes larger weights
• Important attributes are easy to be remembered correctly
• Depends on the purpose
Training & Encoding

- **Method**
  - Build event space (ev & attr) → SVD
  - Cluster events → AP
  - Build episode space (ep & ev) → SVD
  - Cluster episodes → AP

<table>
<thead>
<tr>
<th></th>
<th>Ev1</th>
<th>Ev2</th>
<th>Ev3</th>
<th>Ev4</th>
<th>Ev5</th>
<th>..</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sub=Jack</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>Pre=give</td>
<td>4</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Loc=school</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>dirObj=book</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>indirObj=Tom</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Sub=Tom</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

SVD

\[ TSD' \]

Dimension reduction

\[ T_1 S_1 D_1' \]

coordinates

\[ Attr = T_1 S_1 \]

\[ Ev = D_1 S_1 \]
Event Space
Training & Encoding

• Method
  – Build event space (ev & attr) → SVD
  – Cluster events → AP
  – Build episode space (ep & ev) → SVD
  – Cluster episodes → AP

|        | Ev1 | Ev2 | Ev3 | Ev4 | Ev5 | ...
|--------|-----|-----|-----|-----|-----|------
| Sub=Jack | 2   | 2   | 0   | 0   | 2   | ..
| Pre=give | 4   | 0   | 4   | 0   | 0   | ..
| Loc=school | 1   | 1   | 0   | 1   | 0   | ..
| dirObj=book | 1   | 0   | 1   | 0   | 1   | ..
| indirObj=Tom | 1   | 1   | 0   | 0   | 0   | ..
| Sub=Tom | 0   | 0   | 2   | 2   | 0   | ..

Affinity Propagation
• Automatically select the number of clusters
• The results are independent of the initial points

\[ TSD' \]
\[ T_1 S_1 D_1' \]
Dimension reduction
coordinates
\[ Attr = T_1 S_1 \]
\[ Ev = D_1 S_1 \]
Training & Encoding

**Method**
- Build event space (ev & attr) → SVD
- Cluster events → AP
- Build episode space (ep & ev) → SVD
- Cluster episodes → AP

<table>
<thead>
<tr>
<th></th>
<th>Ep1</th>
<th>Ep2</th>
<th>Ep3</th>
<th>Ep4</th>
<th>Ep5</th>
</tr>
</thead>
<tbody>
<tr>
<td>EC1</td>
<td>2</td>
<td>7</td>
<td>0</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>EC2</td>
<td>4</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>EC3</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td></td>
</tr>
<tr>
<td>EC(n)</td>
<td>8</td>
<td>3</td>
<td>1</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>EC(n+1)</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- First and last few events have larger entries
- Middle events have smaller entries
- Based on serial-position effect

Entry
### Training & Encoding

#### Method

- Build event space \((ev \ & \ attr) \rightarrow SVD\)
- Cluster events \(\rightarrow AP\)
- Build episode space \((ep \ & \ ev) \rightarrow SVD\)
- Cluster episodes \(\rightarrow AP\)

<table>
<thead>
<tr>
<th></th>
<th>Ep1</th>
<th>Ep2</th>
<th>Ep3</th>
<th>Ep4</th>
<th>Ep5</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>EC1</td>
<td>2</td>
<td>7</td>
<td>0</td>
<td>5</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>EC2</td>
<td>4</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>EC3</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td></td>
</tr>
<tr>
<td>EC(n)</td>
<td>8</td>
<td>3</td>
<td>1</td>
<td>3</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>EC(n+1)</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Affinity Propagation**
- Automatically select the number of clusters
- The results are independent of the initial points

\[
ESP' \downarrow \quad \text{Dimension reduction}
\]

\[
E_1 S_1 P_1'
\downarrow \quad \text{coordinates}
\]

\[
Ev = E_1 S_1
\]

\[
Ep = P_1 S_1
\]
Training & Encoding

• **Method**
  – Build event space (ev & attr) → SVD
  – Cluster events → AP
  – Build episode space (ep & ev) → SVD
  – Cluster episodes → AP

• **Benefits**
  – Spaces are constructed by main features
  – Relations are easily computed (dot product)
  – Similar episodes are arranged together
Clustering & Storage

<table>
<thead>
<tr>
<th>Time</th>
<th>Pointers for Ep-Coords</th>
</tr>
</thead>
<tbody>
<tr>
<td>05/21</td>
<td>0x00128</td>
</tr>
<tr>
<td>05/20</td>
<td>0x01217</td>
</tr>
<tr>
<td>05/19</td>
<td>0x01731</td>
</tr>
</tbody>
</table>

Time List

Time-based Retrieval

Cue-based Retrieval

Exemplar 1  Exemplar 2  ...  Exemplar n

Memory Stack
Reconstruction & Forgetting

- **Reconstruction**
  - Episodes and events can be reconstructed from their coordinates

- **Forgetting**
  - Full forgetting: remove less used episodes
  - Partial forgetting: remove unimportant details (reduce dimension)
Implementation

• **Platform**
  – UT2004 + Pogamut 3
  – DeathMatch1v1

• **Goal**
  – Improve the performance of the agent

• **EM Functionalities**
  – Weapon selection (weapon + distance → effect)
  – Items pick up (item → location)
Implementation

• Platform
  – Nadine

• Data
  – Generated by probabilistic pattern (To be improved)

• Goal
  – Enhance believability

• EM Functionalities
  – Encoding, Storage, Retrieval

• Drawbacks
  – Unnatural setting and languages
Thank You!