Improve Similarity Measurement for Episode Memory Model

Juzheng Zhang
Supervisor: Prof. Jianmin Zheng
Co-Supervisor: Prof. Nadia Magnenat-Thalmann
School of Computer Engineering,
Institute for Media Innovation

14/02/2017
Backgrounds

- Declarative
  - Episodic: Past experience
  - Semantic: General Knowledge
- Non-Declarative
  - Procedural: Know-hows
Applications

Episodic Memory Models

Companion Interactions

Agent Performance

Virtual Human & Social Robot

Agent
Elements of Episodic Memory

Episodes

$EP_1 \quad EP_2 \quad \ldots$

Episodes are sequence of events

Events

$EV_1 \quad EV_2 \quad \ldots \quad EV_n$

Events are collections of attributes

Attributes

$Sub \quad DirObj \quad IndirObj \quad Time \quad Loc \quad \ldots$

Attributes are descriptions to what has happened
Problem

Goal

Accurate Retrieval

Content Similarity

Order Similarity

Focus

Elements

- Weights
- Relations
- Order

Episodes

$EP_1 \rightarrow EP_2 \rightarrow \ldots \rightarrow EP_n$

Events

$EV_1 \leftrightarrow EV_2 \leftrightarrow \ldots \leftrightarrow EV_n$

Attributes

Sub DirObj IndirObj Time Loc …
Related Work

- EM-ART [Wang and Tan 2015]

\[
y_j^{(new)} = y_j^{(old)} (1 - \tau)
\]

\[
w_j^{k(new)} = (1 - \beta^k)w_j^{k(old)} + \beta^k (x^k \land w_j^{k(old)})
\]

Related Work

- **Weights**
  - Based on frequency
  - Ignore the specialness of attributes
  - e.g. One attribute appears in all event clusters, while the other attribute appears only in this event cluster.

- **Elements**
  - Well-considered

- **Relations**
  - Not considered

- **Order**
  - Give more weights to recent events
  - It put more focus on the positions of events than the order of events
  - e.g. Episode 1: a->b->c->d->e->f->g
  - Episode 2: h->i->j->k->c
  - Given cue: a->b->c, the episode 2 will be selected.
Proposed Similarity Measure

- **Weights**: Frequency + Specialness
- **Relations**: Hierarchical Relational Space
- **Order**: More balanced and robust to position
Similarity with Weights

• Idea: Both the frequency and the specialness do matter
• Apply TF-IDF Indexing
  – The weights are the extent that an attribute can represent an episode.

**Term frequency**: how many times this attribute appear in this episode

**Inverse document frequency**: how many episodes contain this attribute

Not considered by EM-ART

TF-IDF: Term frequency- Inverse Document Frequency
Similarity with Relations

Build Event Space

1. Embed Attributes and Events
   - LSA

2. Cluster Events
   - Kmeans

Build Episode Space

3. Embed Event clusters and Episodes
   - LSA

4. Cluster Episodes
   - Kmeans

Event Space

- Attributes
- Event clusters
- Predicate = Pursue
- health = Excellent
- Predicate = Escape
- Distance = Far
- Distance = MidRange
- dirObj = Rifle
- dirObj = RocketLauncher
- dirObj = SniperRifle
- dirObj = MinGun
- dirObj = ShockAmmo
- Predicate = Kill
- dirObj = LinkGun
- IndirObj = BioRifle

Episode Space

- Event Cluster
- Episode Cluster

Dimension 1 vs Dimension 2 vs Dimension 3
Event-Order Similarity

Cue Events

A  B  C

Sliding Window

Episode

\[ \text{sim} = \frac{1}{m} \cdot \text{editDistance}(\text{cue, ep_seq}) \]
Event-Order Similarity

**Cue Events**

**Episode**

\[ \text{sim} = 1 - \text{editDistance}(\text{cue, ep_seq}) / m \]
Event-Order Similarity

Cue Events

A  B  C

Episode

\[ \text{sim} = \frac{1 - \text{editDistance}(\text{cue, ep_seq})}{m} \]
Event-Order Similarity

Cue Events

A  B  C

Sliding Window

Episode

\[ \text{sim} = 1 - \text{editDistance}(\text{cue, ep_seq}) / m \]
Combination of Factors

- Elements + Weights
- Relations
- Order

1. \[ S_c = \frac{1}{R_w} + \frac{1}{R_r} \]

2. \[ S = S_c \times S_o \]
Comparison
Comparison

Episode Retrieval

Retrieval

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20

Our_Noise
Our_Partial
EMART_Noise
EMART_Partial
Comparison

Agent Performance

Baseline

Win Rate
Thank YOU
Examples

- Elements have different weights in particular contexts
Examples

• Some elements always appear together
Examples

- Some elements are similar to each other