Derin Gezgin | Russell Kosovsky | Jay Nash
Fall 2024 | COM316: Artificial Intelligence

Homework 11: Frames - Thematic-Role Frames - Version Space Learning

Problem One

Redo the sentences from the previous HW using thematic-role frames as described in class.

Suzie told Robbie to put the wedge on the red block.

(**action1**
    (verb speak)
    (agent Suzie)
    (recipient Robbie)
    (desired-result **action2**))

(action2
    (verb move-object)
    (agent Robbie)
    (object wedge)
    (destination on-red-block))

Robbie was unhappy because of the poor grade he received on a test.

(state-change1
    (state unhappy)
    (agent Robbie)
    (cause **action3**))

(action3
    (verb receive)
    (agent Robbie)
    (object poor-grade)
    (mean test))

Joe moved the car to the garage.

(action4
    (verb move-object)
    (agent Joe)
    (object car)
    (destination garage))

Suzie hit Robbie, which made him sad.

(**action5**
    (verb strike)
    (agent Suzie)
    (recipient Robbie)
    (result **state-change2**))

(state-change2
    (state sad)
    (agent Robbie)
    (cause **action5**))

Derin Gezgin | Russell Kosovsky | Jay Nash
Fall 2024 | COM316: Artificial Intelligence
Homework 11: Frames - Thematic-Role Frames - Version Space Learning

Problem Two

Here is a new thematic-role frame issue we need to deal with. Let's say that we told someone: *"Joe went to McDonalds. He left feeling satisfied with his meal."* If we ask them: *"Did he eat?"* They would say yes (or at least that the statement implies that they ate). If we asked them: *"How did he get his food?"* They could probably answer this also without a problem.

Now, what if we went through the same sequence with a computer? It probably couldn't answer correctly, even if we used our spiffy thematic-role frames to represent the questions.

Suggest and explain a method to deal with this.

1. Representation of Sequences:

We can have a boilerplate representation of a typical sequence of events in specific contexts. For example, a restaurant representation would have typical steps that happen when someone visits a restaurant. For example, a restaurant representation would have the following components:

- **Entry**: Person enters the restaurant.
- **Ordering**: Person orders food.
- **Service**: Food is prepared and served.
- **Consumption**: Person eats the food.
- **Payment**: Person pays for the meal. (could also be after Ordering)
- **Exit**: Person leaves the restaurant.

2. Linking our Representation Format to Frames:

When the computer reads "Joe went to McDonalds," it would recognize McDonald's as a restaurant which would fire the "restaurant representation", and the computer will initiate the sequence of events associated with dining. The second sentence in our input will also contribute to this sequence of events and enhance understanding of the computer.

3. Making Inferences from the Combined Structure:

Did Joe eat?

Yes, because our restaurant representation includes eating as a required event, and Joe felt satisfied with his meal.

How did Joe get his food?

From our combined representation, we can conclude that Joe got his food by ordering it at the restaurant and paying for it before/after his order.

With this form of representation, we can have default boilerplates for common representatinos (visiting a place, eating, shopping, etc.). We can also link places or specific attributes to boilerplate list of actions which will automatically added to our representation list. We can also use the thematic roleframes to enhance this representation. This allows computer to understand implied actions and make more general reasoning. At the same time, our program can be aware of context and trigger specific actions depending on location, or different factors.

Derin Gezgin | Russell Kosovsky | Jay Nash
Fall 2024 | COM316: Artificial Intelligence
Homework 11: Frames - Thematic-Role Frames - Version Space Learning

Problem Three

Show the steps that would take place while using version-space learning on this problem. Here are the input examples (yes under passable indicates that the block is passable):

Number	Width	Height	Color	Wheels	Passable
1	wide	low	green	No wheels	yes
2	narrow	low	red	wheels	no
3	wide	low	yellow	No wheels	yes
4	narrow	high	green	No wheels	no
5	wide	high	blue	No wheels	no
6	wide	low	blue	wheels	no
7	narrow	low	green	No wheels	no

1. Initialization:

- **Specific Hypotheses (S):** Our specific hypotheses will start as we do not accept anything as passable.
- **General Hypotheses (G):** Our general hypotheses will start as we accept everything as passable.
  S = {}
  G = {? ? ? ?}

2. How to Process the Examples:

- **Positive Example:**
    - Generalize S minimally to include the new example, ensuring each new specific hypothesis is a specialization of some hypothesis in G. Prune from G any hypotheses inconsistent with the new example.
- **Negative Example:**
    - Specialize G minimally to exclude the new example, ensuring each new general hypothesis is a generalization of some hypothesis in S. Prune from S any hypotheses that incorrectly classify the negative example as positive.

Derin Gezgin | Russell Kosovsky | Jay Nash
Fall 2024 | COM316: Artificial Intelligence
Homework 11: Frames - Thematic-Role Frames - Version Space Learning

Example 1: Passable

- *Attributes:* [wide, low, green, no wheels]
- *Action:* Generalize S to include this example.
- *Updated S:*
  {(Width: wide, Height: low, Color: green, Wheels: no wheels)}

Example 2: Not Passable

- *Attributes:* [narrow, low, red, wheels]
- *Action:* Specialize G to exclude this negative example.
- Specific of G Version to Exclude Example 2:
  G1: (Width: Wide, Height: ?, Color: ?, Wheels: ?)
  G2: (Width: ?, Height: High, Color: ?, Wheels: ?)
  G3: (Width: ?, Height: ?, Color: Green, Wheels: ?)
  G4: (Width: ?, Height: ?, Color: Yellow, Wheels: ?)
  G5: (Width: ?, Height: ?, Color: Blue, Wheels: ?)
  G6: (Width: ?, Height: ?, Color: ?, Wheels: No Wheels)
- *Updated G:* { G1, G2, G3, G4, G5, G6 }

Example 3: Passable

- *Attributes:* [wide, low, yellow, no wheels]
- *Action:* Generalize S to include this example.
- *Updated S:*
  {(Width: wide, Height: low, Color: ?, Wheels: no wheels)}
- Remove the hypothesis in G that does not cover the positive example (S). G2,G3,G5 will be removed.
- *Updated G:* { G1 G4 G6 }

Example 4: Not Passable

- *Attributes:* [narrow, high, green, no wheels]
- Action: Specialize G to exclude this negative example.
- Considering our current specialization (wide, low, ?, no wheels) and our new negative example, we can update our general hypotheses.
- Updated G values:
  G1: (Width: Wide, Height: ?, Color: ?, Wheels: ?)
  G4: (Width: Wide, Height: ?, Color: ?, Wheels: No Wheels)
  G6: (Width: ?, Height: Low, Color: ?, Wheels: No Wheels)

Example 5: Not Passable

- *Attributes:* [wide, high, blue, no wheels]
- Action: Specialize G to exclude this negative example.
- Considering our current specialization (wide, low, ?, no wheels) and our new negative example, we can update our general hypotheses.
- Updated G values:

G1: (Width: ?, Height: Low, Color: ?, Wheels: No Wheels)
G4: (Width: Wide, Height: Low, Color: ?, Wheels: ?)
G6: (Width: Wide, Height: Low, Color: ?, Wheels: No Wheels)

Example 6: Not Passable

- *Attributes:* [wide, low, blue, wheels]
- Action: Specialize G to exclude this negative example.
- Considering our current specialization (wide, low, ?, no wheels) and our new negative example, we can update our general hypotheses.
- Updated G values:

G1: (Width: ?, Height: Low, Color: ?, Wheels: No Wheels)
G6: (Width: Wide, Height: Low, Color: ?, Wheels: No Wheels)

Example 7: Not Passable

- *Attributes:* [narrow, low, green, no wheels]
- Action: Specialize G to exclude this negative example.
- Considering our current specialization (wide, low, ?, no wheels) and our new negative example, we can update our general hypotheses.
- Updated G values:

G6: (Width: Wide, Height: Low, Color: ?, Wheels: No Wheels)

As our examples finish, we can conclude that the rule for passable object is *wide, low, {color does not matter}, no wheels*