Module: COMP5450M – Knowledge Representation and Reasoning

Students: Sian Carey, Emma Briggs and Sarah Smith

KRR Challenges and Potential

2020

Table of Contents

[Introduction 1](#_Toc57037429)

# Introduction

# Examples

## Example 1

**Emma’s Problem**

**Problem: From a collection of Winograd schemas, number 137 [].**

*“I tried to paint a picture of an orchard, with lemons in the lemon trees, but they came out looking more like [light bulbs / telephone poles]. What looked like [light bulbs / telephone poles]?”*

Here, the problem has two options: the objects “light bulbs” and “telephone poles”, which relate to two possible subjects: “lemons” and “lemon trees” (as painted on the picture of an orchard). Should “light bulbs” be the answer, the correct solution is “lemons”, which resemble light bulbs in shape, size, and colour, and should “telephone poles” be used, the solution is “lemon trees”, which share the same characteristics as telephone poles in appearance. This is obvious to a human, who can look at these objects side by side and intuit that they resemble each other, but to a machine, the deduction cannot be made straight away as each of the possible answers could refer to either subject, and the answer is not available via use of a search engine, one of the criteria for being a Winograd schema [].

To solve this problem using KRR methods, it first made sense to define a set of axioms which relate to common knowledge. From there, introducing inference rules contributes towards solving the problem in its entirety.

It can be considered common knowledge that lemons are (mostly) yellow, light bulbs – when switched on – are yellow, and that both trees and telephone poles are a dark colour. Therefore assigning these colours to objects can be considered basic assumptions. Further, both lemons and light bulbs are oval in shape, and both telephone poles and trees are cylindrical. Therefore, these characteristics can also be included as axioms. Finally, we consider the size of these objects. Size commonly has three dimensions: height, depth, and width []. Both lemons and light bulbs are short, thick and wide, and both trees and telephone poles are tall, and thin in both width and depth, and so we append these characteristics to our list of axioms accordingly. In addition, lemons being the plural of lemon, light bulbs being the plural of light bulb, and so on, are instantly deducible facts, and these are also added to the list of axioms.

Our list of axioms then becomes:

IsColour(Lemon, Yellow)

IsColour(Light\_bulb, Yellow)

IsColour(Tree, Dark)

IsColour(Telephone\_pole, Dark)

IsShape(Lemon, Oval)

IsShape(Light\_bulb, Oval)

IsShape(Telephone\_pole, Cylinder)

IsShape(Tree, Cylinder)

IsLength(Lemon, Short)

IsDepth(Lemon, Thick)

IsWidth(Lemon, Wide)

IsLength(Light\_bulb, Short)

IsDepth(Light\_bulb, Thick)

IsWidth(Light\_bulb, Wide)

IsLength(Tree, Tall)

IsDepth(Tree, Thin)

IsWidth(Tree, Thin)

IsLength(Telephone\_pole, Tall)

IsDepth(Telephone\_pole, Thin)

IsWidth(Telephone\_pole, Thin)

IsPlural(Trees, Tree)

IsPlural(Ovals, Oval)

IsPlural(Cylinders, Cylinder)

IsPlural(Light\_bulbs, Light\_bulb)

IsPlural(Telephone\_poles, Telephone\_pole)

IsPlural(Lemons, Lemon)

IsPlural(Lemon\_trees, Lemon\_tree)

Now it comes to add inference rules. It holds true that if we paint an object, we can see what has been painted, and if we can see something, it always looks like at least one other thing (even if that thing is the object itself, or a blob).

So we add:

all x ( Paint(x) -> See(x) )

all x ( See(x) -> exists y LooksLike(x,y) )

Now we need the fact that lemon trees are from the tree family, and that plants of the same type resemble each other (approximately).

IsPlantType(Lemon\_tree, Tree)

all x all y ( IsPlantType(x, y) -> LooksLike(x, y) )

Then we add transitivity and reflexivity for LooksLike:

all x all y all z ( ( LooksLike(x,y) & LooksLike(y,z) ) -> LooksLike(x,z) )

all x all y ( LooksLike(x,y) <-> LooksLike(y,x) )

Next we need to add the fact that two objects of the same length, depth, and width both have some (approximate) size which is the same:

all x all y ( ( exists l exists w exists d ( IsLength(x,l) & IsLength(y,l) & IsWidth(x,w) & IsWidth(y, w) & IsDepth(x,d) & IsDepth(x,d) ) ) -> exists v ( IsSize(x, v) & IsSize(y, v) ) )

Now we need to include the fact that the lemons and lemon trees were painted (instantly deduced from the sentence statement):

Paint(Orchard)

Paint(Orchard) -> ( Paint(Lemons) & Paint(Lemon\_trees) )

Now we need to state that if an object looks like another object, the plural of that object looks like the plural of the other object (if a lemon looks like a light bulb, it follows that a bunch of lemons could look like a bunch of light bulbs).

all x all y all u all v ( ( LooksLike(x, u) & IsPlural(y, x) & IsPlural(v, u) ) -> LooksLike(y, v) ).

Furthermore, objects can only have one shape or size (although they can have multiple colours):

all x all w all y ( ( IsShape(x, w) & -(y=w) ) -> ( -IsShape(x,y) ) ).

all x all v all y ( ( IsSize(x, v) & -(y=v) ) -> ( -IsSize(x,y) ) ).

Finally, we deduce that if two objects x and y are the same shape, size and colour, then they look like each other. And if two objects look like each other, they must have the same shape, size, and colour. This is an ‘if and only if’ statement because the objects only look alike if these conditions hold true.

all x all y ( ( exists z exists w exists v (IsShape(x,z) & IsShape(y,z) & IsColour(x,v) & IsColour(y, v) & IsSize(x, w) & IsSize(y, w) ) ) <-> LooksLike(x, y) ).

## Example 2

**Sian’s Example**

During a game of tag, Ethan [chased/ran from] Luke because he was "it". Who was "it"?  
**Answer:**Ethan/Luke

Part 1

* Statement of problem *(25)*

In this section the problem being studied is number 148 of the Winograd Schemas, namely

During a game of tag, Ethan [chased/ran from] Luke because he was “It”.

Who was “it”?

Answer: Ethan/Luke

* What knowledge/reasoning principles are needed to know the answer *(175)*

In order to answer this the program will require additional knowledge and reasoning principles, primarily about the game of tag. We will first consider the knowledge that the program will need.

A large part of the knowledge are the rules of tag. First, tag is a game that multiple people play, and in tag you have exactly one person who is labelled as “it”. If the person who is labelled as “it” touches another person then that label transfers, so the original person no longer has the label “it” and the touched person does have the label “it”. Furthermore, if someone has the label “it” then they will try and give it away.

The other important part of knowledge that must be taught are the meanings of ‘chased’ and ‘ran from’. If one person chases another then the first person is trying to touch the second. If one person runs from another then the first person is trying to not be touched by the second.

There are also reasoning principles that must be taught. Maybe? To do with only touching someone if you’re it? Only run away from people who are it?

* How these deduce the answer of each sentence (in English) *(100)*

If the sentence has the option ‘chased’ in it, then the program should give the answer Ethan. This is because Ethan chased Luke, so Ethan is trying to touch Luke and therefore Ethan is trying to give his “it” away. Hence Ethan is currently “it”.

Alternatively, it the sentence has the option ‘ran from’ in it, then the program should give the answer Luke, as Ethan is trying to not get touched by Luke therefore Luke is “it”.

Part 2 *(200)*

* A logical formula representing each version of the sentence
* Logical formulas for axioms/general facts required for each sentence

T(x) – x is playing tag

IT(x) – x is “it”

Tou(x,y) – x is trying to touch y

E – Ethan

L – Luke

R(x,y) – x runs from y

C(x,y) – x chases y

(T( E ) & T(L) & C(E,L)) -> IT( E )

(T( E ) & T(L) & R(E,L)) -> IT(L)

Exists x (T(x) & IT(x)) – there is always someone playing tag who it “it”

All x All y ((T(x) & T(y) & C(x,y)) -> Tou(x,y)) – If two people play tag and one chases the other then the chaser is trying to touch the chasee

All x All y ((T(x) & T(y) & R(x,y)) -> Tou(y,x)) – If two people play tag and one is running from the other the second person is trying to touch the first person

All x All y ((T(x) & T(y) & Tou(x,y)) -> IT(x)) – If one person is trying to touch another the first person is “it”

## Example 3

**Sarah’s Section**

The Choices of Plausible Alternatives (COPA) dataset was developed by Andrew Gordon as an evaluation tool for “open-domain commonsense causal reasoning”. The dataset was heavily inspired from the Recognizing Textural Entailments (RTE) challenges. A RTE question consists of a text fragment and a hypothesis. The goal of this question is to determine if the truth of the hypothesis is entailed in the text fragment. For example:

*Text fragment: “Cavern Club sessions paid the Beatles £15 evenings and £5 lunchtime.*

*Hypothesis: The Beatles performed at Cavern Club at lunchtime.*

This is an example of positive entailment. Although RTE is a great evaluation tool for inferential capability it is not as useful for evaluating commonsense inference. Through the use of the RTE challenges the distinction between entailment and implication was made. Entailments are inferences that are necessarily true due to the meaning in the text. Whereas implications are inferences that are expected to be true due to likely causes, effects of the text or are default assumptions. The entailment between two text fragments is either strongly positive or negative whereas implications are judged by the degree of plausibility.

In order to test for commonsense casual implication, the COPA date set was developed (modified RTE). The dataset contains over 1000 questions with each question being composited of three parts: The premise and two plausible alternatives. The plausible alternatives demonstrate either the cause or the effect of the given premise. For example:

(Backward causal reasoning)

*Premise: The women met for coffee. What was the CAUSE of this?*

*Alternative 1: The cafe reopened in a new location.*

*Alternative 2: They wanted to catch up with each other.*

(Forward causal reasoning)

*Premise: The physician misdiagnosed the patient. What happened as a RESULT?*

*Alternative 1: The patient filed a malpractice lawsuit against the physician.*

*Alternative 2: The patient disclosed confidential information to the physician.*

In both examples either alternative is plausible, but the correct answer is the alternative that is the most plausible. Explain why this a good test e.g. how have they ensured that the system is reasoning not just guessing or using different methods (causality bridges)

*Premise: The man fell unconscious. What was the CAUSE of this?*

*Alternative 1: The assailant struck the man in the head.*

*Alternative 2: The assailant took the man's wallet.*

A causal bridge may be that injuries to the head cause unconsciousness.

# Conclusion

# References