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

Word Count: 1,511

## Winograd Schema Challenge

The Winograd Schema Challenge (WSC) was devised as a test to improve artificial intelligence (AI) and reduce the gap between the ability of humans and machines to interpret sentences (Isaak N. and Michael L. 2020). It is specifically focused on the ‘pronoun disambiguation problem’ which involves the interpretation of *anaphors* (words or phrases that refer back to an aforementioned word or phrase) (Levesque H.J. et al 2011, Neufeld E. and Finnestad S. 2020). It is viewed as a worthwhile alternative to the Turing Test and shares parallels with this test in that some of its prerequisites are that humans can pass it easily and passing the test resembles having the ability to ‘think’; however, the scientific research community is divided when it comes to emphasising one over the other (Turing A. M. 1950, Winograd T. 1987, Levesque H.J. et al 2011, Neufeld E. and Finnestad S. 2020).

One of the main arguments as to why it can be considered an improvement on the Turing Test is that it doesn’t rely on conversation, which can easily be adapted, and, for a machine to participate in, requires a high level of deception and a fabrication of character (Levesque H.J. et al 2011). It is considered a quicker way of determining a computer’s level of human-like intelligence because a human would immediately be able to answer any Winograd question whereas it would be a considerable challenge for a computer (Levesque H.J. et al 2011, Bailey D. et al 2018).

The challenge itself was originally constructed by Hector J. Levesque in 2011 and derives its name from the original example given by Terry Winograd in 1972 which states (Levesque H.J. et al 2011, Winograd T. 1987):

*“The city councilmen refused the demonstrators a permit because they [feared/advocated] violence. Who [feared/advocated] violence?”* (Winograd T. 1987, p.33)

A more basic sentence might say something such as “The cat cried because it was unhappy.” With simple rules of syntax logic, a machine can interpret that the “it” refers to the cat, because the cat is the subject of the sentence. However, this becomes much more complex when two subjects are used, with two alternate solutions, such as in the example given above. Answering this question demonstrates an advanced level of computer intelligence because *“no set of syntactic or semantic rules could interpret this pronoun reference without using knowledge of the world.”* (Winograd T. 1987, p.33) Therefore, more complex reasoning systems are required to solve the Winograd problems.

This example helps illustrate the form of the test’s input: a sentence, with two subjects, an ambiguous pronoun later in the sentence which could refer to either of the subjects, and a keyword that determines the answer. Each schema also has an alternate solution of the keyword that, when changed, alters the answer (Levesque H.J. et al 2011). Above, the two subjects are the ‘city councilmen’ and the ‘demonstrators’, the pronoun is ‘they’, and the two alternate keywords are ‘feared/advocated.’ The two noun phrases in a Winograd schema are always of the same semantic class and gender, and the question always asks which subject the pronoun is referring to (Levesque H.J. et al 2011, Bailey D. et al 2015). A Winograd schema cannot be solved by a quick search via a search engine (Levesque H.J. et al 2011). Furthermore, the sentence should be grammatically correct and easily solvable by a native speaker of the language it is presented in (Levesque H.J. et al 2011).

The WSC is a significant challenge because, as of yet, there is no definitive solution for automating accurate interpretations of these types of sentences (Neufeld E. and Finnestad S. 2020). The Knowledge Representation and Reasoning (KRR) approach works by building up a knowledge base of facts and rules from which the solution is deduced (Richard-Bollans A. et al 2018). This has been shown to be fairly successful in solving Winograd schemas, with graph-based representation solutions being some of the frontrunners within this approach (Sharma A. 2019).

Implications of developing solutions for the WSC include the broadening of formalisations for commonsense knowledge, which could assist both AI engineering and AI research, helping to develop Virtual Personal Assistants, for example (Morgenstern L. and Ortiz-Jr C L. 2015). It has been considered an effective way to objectively track progress of research in the field of commonsense reasoning (Morgenstern L. and Ortiz-Jr C L. 2015).

However, the WSC remains a considerable challenge for KRR for a number of reasons. Firstly, in part due to the infinite possibilities of human interpretation, the knowledge bases designed for a Winograd schema are not unique, and there is no easy way of knowing which would be optimal (Richard-Bollans A. et al 2018). Furthermore, KRR relies on commonsense reasoning, for which our current level of understanding is far from complete (Davis E. and Marcus G. 2015, Richard-Bollans A. et al 2018). Additionally, it is difficult to determine the level of abstraction needed to enable the computer to understand the problem (Davis E. and Marcus G. 2015). Scenarios which seem simple to a human may in fact require hugely complex logical deductions for a computer to understand them (Davis E. and Marcus G. 2015).

## Choices of Plausible Alternatives (COPA)

**ADD REFERENCES**

Another popular approach to commonsense causal reasoning is the Choices of Plausible Alternatives (COPA) dataset. The 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. Each question would feature a text fragment and a hypothesis with the goal of determining whether or not the hypothesis is true based on its entailment in the text fragment. The following example demonstrates positive entailment:

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

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

Although RTE was a great evaluation tool for inferential capability, it was found to be much less useful for evaluating commonsense inference. This was due to the inherent distinction between entailment and implication. 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. In summary, 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 based on modified RTE. The dataset contains 1000 multiple choice questions with each question being composited of: A statement (premise) and two choices (plausible alternatives). Both alternatives could have a causal relation to the premise and thus could be a plausible cause or effect of the given premise. For example:

### Scenario 1 – (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.*

### Scenario 2 – (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.*

The examples presented above demonstrate the two formats available in the COPA dataset, with the alternatives demonstrating either: The plausible effects of the premise (forward causal reasoning) or the plausible causes of the premise (backward casual reasoning). In both examples either of the presented alternatives is deemed to be plausible. However, the correct answer is the alternative that is deemed to be the most plausible.

To the human reader it is intuitively evident that in the first scenario, the first alternative is the most plausible *result/effect* of the premise, and that in the second scenario the second alternative is the most plausible *cause* of the premise. This intuition or “commonsense” requires the individual to implicitly know that misdiagnosing a patient would likely make the patient distrustful of the doctor and potentially file a lawsuit if there was malpractice or that socially many people “meet for coffee” as a way to catch up likely more often than visiting a café at a new location. The reader does not need to be explicitly told this information in order to make the correct inference.

The COPA dataset provides a way to test a system’s or AI’s ability to make commonsense causal reasoning judgements. The questions pose a forced choice in its alternatives with a definitive right answer which makes it easier to determine a system’s performance. Interestingly, the dataset was compiled in such a way that when tested on human raters the raters were nearly in 100% agreement on the answers of all of the question. Furthermore, the dataset is split into a development set and test set to prevent overfitting with the system being able to achieve an accuracy of exactly 50% just by guessing.

In order to make a correct inference, the system would likely need to have a causality bridge (knowledge that connects the premise to the correct alternative). This is due to “commonsense” often being assumed knowledge and not explicitly expressed in day-to-day activities when determining the plausibility of the cause or effect of an event. This continues to pose a substantial challenge for AI causal reasoning inference systems.

# Examples

## Example 1

**Problem 1: From a collection of Winograd schemas, number 137 (Davis E. et al 2011).**

*“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]?” (*Davis E. et al 2011)

Here, the problem has two options: the objects “light bulbs” and “telephone poles”, which correspond to the two solutions – the subjects “lemons” and “lemon trees” – respectively. These solutions are instinctive to a human, who can intuit that light bulbs more closely resemble lemons and telephone poles more closely resemble lemon trees. For a machine, the syntax makes it difficult. The answer is not immediately available via search engine, one of the criteria for representing a Winograd schema (Levesque H. J. et al 2011).

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. Therefore, building a knowledge base using first-order logic seemed like a suitable strategy (Grosan C. and Abraham A. 2011).

**Facts**

These facts are basic assumptions which relate to common knowledge; for example, the shape, general size, and colour of ubiquitous objects such as trees and lemons. This also includes semantic rules such as the fact that ‘lemons’ is the plural of ‘lemon’ etc.

Examples for the lemon object:

IsColour(Lemon, Yellow)

IsShape(Lemon, Oval)

IsLength(Lemon, Short)

IsDepth(Lemon, Thick)

IsWidth(Lemon, Wide)

IsPlural(Lemons, Lemon)

Similar facts are constructed for other basic objects such as trees, light bulbs and telephone poles.

**Inferences**

First we need to represent the objects having been painted (instantly deduced from the problem statement):

Paint(Orchard)

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

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 splash of colour).

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 infer that two objects of the same length, depth, and width both have some (approximate) size which is equivalent:

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 state that if an object resembles another, semantically, they do so in plural form as well:

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 vice versa.

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) ).

Word count: ~600 words.

## Example 2

**Problem 2: From a collection of Winograd schemas, number 148 (Davis E. et al 2011).**

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

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 parts 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. It can be assumed, that if all people involved are playing a game of tag, then one person chasing another means the first person is trying to touch the second and hence is “it”. If the first person is running away from the second, then this conclusion switches around and the second person is “it”.

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”.

Logical Formula

In order to represent the two sentences in logical formula it is first required to define the constants, predicates and relations we need to build up the sentences. There are only two constants required for this, Ethan and Luke. There are also only two predicates required, we will take these as T(x) to represent ‘x is playing tag’ and IT(x) to represent ‘x is “it”’. Finally, we also require two relations. R(x,y) represents ‘x runs from y’ and C(x,y) represents ‘x chases y’. Using these we can then form the following two sentences.

(T(Ethan) & T(Luke) & C(Ethan,Luke)) -> IT(Ethan)

(T(Ethan) & T(Luke) & R(Ethan,Luke)) -> IT(Luke)

We will now represent some of the facts that are required to know the answer in logical formula. In order to do this we need to define one more relation, Tou(x,y) will represent ‘x is trying to touch y’. Now we can define the following sentences.

If two people play tag and one chases the other then the chaser is trying to touch the person being chased.

All x All y ((T(x) & T(y) & C(x,y)) -> Tou(x,y))

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) & R(x,y)) -> Tou(y,x))

If one person is trying to touch another the first person is “it”.

All x All y ((T(x) & T(y) & Tou(x,y)) -> IT(x))

## Example 3

### COPA Problem: Number 1

*Premise: My body cast a shadow over the grass. What was the CAUSE of this?*

*Alternative 1: The sun was rising.*

*Alternative 2: The grass was cut.*

The presented COPA question above, is an example of a backward casual reasoning question where the aim is to determine what *caused* the premise. In this case we want to deduce what caused the body of someone to cast a shadow over the grass. It is intuitively clear to the reader that the correct answer is meant to be the first alternative “the sun was rising”. However, this is not immediately clear without some background “commonsense” knowledge. It is assumed that the reader knows that when an object (the body) obscures a light source (the sun) then the object will cast a shadow on the surface (the grass) opposite to the light source. It also assumes that the reader is aware that a rising sun will cast a shadow and that cut grass does not create a shadow.

### Logical Formula

In order to tackle this question, we first need to extract the key words and concepts needed to answer the question. Firstly, we assume that the light travels in a straight line and reflects back when obstructed by a dense surface or object. We also assume that any obstructing object has a surface behind it. If the obstructing object is smaller than the surface behind it then a “shadow” (lack of light), in the shape of the object, is cast onto the surface. Secondly, we need to know that the sun is a light source and because it is rising it is casting light into the domain. We also need to know that the body is a dense object and likely smaller than the grass surface. We can also assume that due to gravity the body is standing on the surface and that there is grass behind the body due to the assumed size of the surface domain. It can therefore be inferred that as the second alternative choice “the grass was cut” does not have a light source that this could not be the cause of the shadow.

We want to define that the sun is a light source, that the body is an object and that the grass is the surface.

Rising(sun) – (this will become LightSource(sun))

Object(body)

Surface(grass)

all x (Rising(x) -> LightSource(x))

We assume that the light source must be “rising” in order to be a light source (i.e., only the sun is a light source). If the sun is rising, then the sun is a light source.

Exists x (LightSource(x) & All y (LightSource(y) -> x = y))

We assume that there is only one light source (the sun).

Exists z Exists y (Object(y) & Surface(z) -> BehindOf(z,y))

We ensure that there is an object that has a surface behind it (to cast a shadow onto).

Exists x Exists y (LightSource(x) & Object(y) -> Obscures(y,x))

There is an object that obscures a light source.

All x All y All z (LightSource(x) & Object(y) & Surface(z) & BehindOf(z,y) & Obscures(y,x) -> CastShadow(y,z))

If there is a light source, an object, a surface, the surface is behind the object, and the object is in front of the light source then the object will cast a shadow on the surface.

All x All y All z (LightSource(x) & Object(y) & Surface(z) & BehindOf(z,y) & -Obscures(y,x) ->

-CastShadow(y,z))

Similarly, if there is a light source, an object, a surface, the surface is behind the object, but the object is **not** in front of the light source then the object will **not** cast a shadow on the surface.

All x All y All z (LightSource(x) & Object(y) & Surface(z) & -BehindOf(z,y) & Obscures(y,x) ->

-CastShadow(y,z))

Again, if there is a light source, an object, a surface, the object is in front of the light source, but the surface is **not** behind the object then the object will **not** cast a shadow on the surface.

# Conclusion

Using the first-order logic approach with facts and inference rules (using example 1 above to illustrate) and running these rules through an automated system for solving theorems - such as Prover9 - would successfully enable the computer to come to the conclusion that the solution ‘light bulbs’ refers to ‘lemons’ and ‘telephone poles’ refers to ‘lemon trees’, thus solving the pronoun disambiguation problem (McCune W. 2005). One of the reasons this approach works well is that it allows the ability to successfully capture relations between objects and express information about a large domain in a compact way (Grosan C. and Abraham A. 2011, Richard-Bollans A. et al 2018).

However, even though encoding this knowledge base could prove successful for this one example, this approach does not come without limitations. Firstly, the monotonic reasoning of this approach in general risks incomplete information and so may prove insufficient (Grosan C. and Abraham A. 2011). Secondly, a computer may have to reason about potentially conflicting information within the same knowledge base. It is restricted to the domain it is given, and therefore wouldn’t perform well with uncertainty (Davis E. and Marcus G. 2015). Vagueness is another known challenge in automated language interpretation (Bennett B. 2005). Using non-monotonic KRR techniques – such as fuzzy logic or default rules - may prove more successful (Zadeh L. 1965, Besnard P. 1989). For example, in 1 above, expressing the extent of resemblance may be more accurate. Lemons do not look exactly like light bulbs, and so a greater level of precision could be useful for capturing resemblance (Bennett B. 1998). Many similar problems could be found in the other Winograd schemas due to the complex, nuanced nature of language (Richard-Bollans A. et al 2018).

The game of tag related Winograd problem, under the conditions stated above in example 2, can also be solved by KRR methods. The condition required is that in a game of tag, you only run from the person who is “it” and you only chase people if you are “it”. Whilst this is true in the rules of the game it is not necessarily true in life and often in tag the players may not know who is “it” and run from any other players. Therefore, a possible answer to the original question is that neither Ethan or Luke are “it”. However, as we are given with the problem that either Ethan or Luke are “it”, it can be assumed by both humans and the computer program that the required condition holds.

However, this ability to solve one exact problem is not useful unless it can be replicated across many different problems of the same structure. As there are many of these problems to solve (Isaak N. and Michael L. 2020) it is important that any solution is wide ranging and has the ability to work on a significant number of them. Alternative KRR methods may assist in reaching a solution better than the option given above and may do better than the other methods offered.

Furthermore, it should be noted that the KRR approach as a whole is limited. Logic rules rarely correspond well to human thought processes, and so a computer may appear to be ‘thinking’ like a human, but really be following none of the same processes (the human brain processes information in myriad ways, often building on experience and context, which are extremely difficult to automate) (Dranovsky A. 2011, Davis E. and Marcus G. 2015). Additionally, this approach is not generalisable from one rule to the next: a whole new set of rules is required for each Winograd schema, rendering this approach tedious and inefficient (Davis E. and Marcus G. 2015).

There are other solutions to this problem, such as statistical (Chakraborty and Sundararajan, 2007) or through Machine Learning (ML) (Zhang et al., 2018, Chen Y. et al 2019). It has been shown that ML may be able to achieve a higher level of accuracy by taking advantage of a large amount of online data for solving problems requiring commonsense knowledge; however, such techniques do not incorporate inference methodically like KRR techniques do and often have very computationally expensive training processes (Ng V. 2017, Richard-Bollans A. et al 2018). Perhaps a combination of multiple methods could be even more successful (Rahman A. and Ng V. 2012, Richard-Bollans A. et al 2018).

ADD COPA CONCLUSION

# References

Bailey, D., Harrison, A., Lierler, Y., Lifschitz, V. and Michael, J. 2015. The Winograd Schema Challenge and Reasoning About Correlation. In: *Working Notes of the Symposium on Logical Formalizations of Commonsense Reasoning.*

Davis, E. and Marcus, G. 2015. Commonsense reasoning and commonsense knowledge in artificial intelligence. Communications of the ACM. 58(9), pp.92-103.

Isaak, N. and Michael, L. 2020. Winventor: A Machine-driven Approach for the Development of Winograd Schemas.

Levesque, H.J., Davis, E. and Morgenstern, L. 2011. The Winograd Schema Challenge. Proceedings of the Thirteenth International Conference on Principles of Knowledge Representation and Reasoning. 13, pp.552-561.

Neufeld, E. and Finnestad, S. 2020. In defense of the Turing test. Ai & Society. 35(4), pp.819-827.

Richard-Bollans, A., Alvarez, L.G. and Cohn, A.G. 2018. The Role of Pragmatics in Solving the Winograd Schema Challenge. Proceedings of the Thirteenth International Symposium on Commonsense Reasoning (Commonsense 2017).

Sharma, A. 2019. Using Answer Set Programming for Commonsense Reasoning in the Winograd Schema Challenge. Theory and Practice of Logic Programming. 19(5-6), pp.1021-1037.

Winograd, T. 1972. Understanding natural language. Cognitive Psychology. 3(1), pp.1-191.

Chakraborty, S. and Sundararajan, K.K. 2007. Winograd's algorithm statistically revisited: It pays to weigh than to count! *Applied Mathematics and Computation.* **190**(1), pp.15-20.

Zhang, H.M., Song, Y.Q. and Acm. 2018. *A Distributed Solution for Winograd Schema Challenge.*

Ernest Davis, Leora Morgenstern, Charles Ortiz. 2011. *The Winograd Schema Challenge: Collection of Winograd Schemas.* [Online]. Available from: <https://cs.nyu.edu/faculty/davise/papers/WinogradSchemas/WSCollection.html>