**Winograd Schema Challenge – Introduction:**

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. et al 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. et al 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 (Winograd T. 1987, Levesque H.J. et al 2011, Neufeld E. et al 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. et al 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. et al 2019).

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. et al 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. et al 2015). Scenarios which seem simple to a human may in fact require hugely complex logical deductions for a computer to understand them (Davis E. et al 2015).

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.

Word Count: 780