POSTGRADUATE PROGRAMME IN
INFORMATION AND COMMUNICATION SYSTEMS,
SCHOOL OF PURE AND APPLIED SCIENCES

A Fully-Fledged Approach to the Winograd Schema Challenge: Tackling, Utilizing and Developing Winograd Instances

Nicos X. Isaak

A DISSERTATION SUBMITTED IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

November 2021

ISBN: 978-9963-695-76-8

VALIDATION PAGE

The present doctoral dissertation was completed in the context of the Doctoral Programme in Information and Communication Systems at the School of Pure and Applied Sciences of the Open University of Cyprus and was successfully defended by the candidate on the 22nd of October 2021.

Doctoral Candidate: Nicos X. Isaak
Doctoral Thesis Title: A Fully-Fledged Approach to the Winograd Schema Challenge: Tackling, Utilizing and Developing Winograd Instances

Examination Committee:

Chair of the examination committee: Professor Antonis Kakas, University of Cyprus

Supervisor: Associate Professor Loizos Michael, Open University of Cyprus

Committee member: Professor Kleanthes K. Grohmann, University of Cyprus

Committee member: Professor Ernest Davis, New York University

Committee member: Associate Professor Katerina Pastra, ATHENA Research Center

Professor Antonis Kakas
Faculty of Pure and Applied Sciences
University of Cyprus

Chair signature: .....

Associate Professor Loizos Michael
School of Pure and Applied Sciences
Open University of Cyprus

Supervisor signature: .....

DECLARATION OF DOCTORAL CANDIDATE

The present doctoral dissertation was submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy of the Open University of Cyprus. It is a product of original work of my own, unless otherwise stated through references, notes or any other statements.

Nicos X. Isaak
November 2021

.....
Signature

ABSTRACT (Greek)

Το Winograd Schema Challenge (WSC) είναι μία νέα πρόκληση, ένας διαγωνισμός, όπου οι ερευνητές καλούνται να δημιουργήσουν έξυπνα συστήματα Τεχνητής Νοημοσύνης (ΤΝ). Την τελευταία δεκαετία, μέσα στην οποία έχει γίνει γνωστό στην ερευνητική κοινότητα, ερευνητές καλούνται να δημιουργήσουν συστήματα που να μπορούν να επιλύουν προβλήματα μέσω χρήσης κανόνων κοινής λογικής όπως και οι άνθρωποι.

Το **WSC** αναφέρεται στη δημιουργία συστημάτων ΤΝ που είναι ικανά να επιλύουν προβλήματα αναφοράς αντωνυμιών σε προτάσεις. Συγκεκριμένα, υπάρχουν ζεύγη από σχήματα (schemas), το κάθε ένα από τα οποία αποτελείται από μία πρόταση, μία ερώτηση, η οποία αναφέρεται σε μια αντωνυμία της πρότασης και δύο πιθανές απαντήσεις (ως απαντήσεις χρησιμοποιούνται ουσιαστικά στο ίδιο γένος και αριθμό, γεγονός που κάνει ακόμη πιο δύσκολη τη δημιουργία ενός τέτοιου συστήματος). Επίσης, σε κάθε πρόταση υπάρχει μια ειδική λέξη ή φράση που όταν αλλάξει, αλλάζει και η απάντηση της ερώτησης. Ο απώτερος σκοπός είναι η κατανόηση της ανθρώπινης συμπεριφοράς, του τρόπου δηλαδή που χρησιμοποιούν τους κανόνες κοινής λογικής, για να επιλύσουν τέτοια προβλήματα. Αν και είναι κάτι το οποίο για τους ανθρώπους είναι έμφυτο, η δημιουργία τέτοιων συστημάτων είναι δύσκολη και σχεδόν ακατόρθωτη.

Η διατριβή αυτή επικεντρώνεται στη σχεδίαση και ανάπτυξη συστημάτων που αφορούν το **WSC**. Επειδή κατά τα τελευταία χρόνια υπάρχει μια τάση στην ερευνητική κοινότητα να εστιάζει σε μη διαφανείς στατιστικές λύσεις (μη-συμβολική ΤΝ) και επειδή οι άνθρωποι δεν στηρίζονται σε μοτίβα λέξεων για να επιλύουν τέτοια προβλήματα, έχουμε δημιουργήσει ένα σύστημα λογισμικού το οποίο, μέσα από την εξαγωγή γνώσης κοινής λογικής από την Αγγλική **Wikipedia**, επιλύει προβλήματα του **WSC**, τα οποία έχουν δημιουργηθεί από ειδικούς του χώρου. Το συγκριτικό πλεονέκτημα του διαφανούς αυτού συστήματος παρουσιάζεται μέσα από πειράματα που έγιναν σε υφιστάμενα σύνολα δεδομένων. Σύμφωνα με τα αποτελέσματά μας φαίνεται ότι, αν θέλουμε γενικότερα να πετύχουμε τη δημιουργία συστημάτων που χρησιμοποιούν κανόνες κοινής λογικής, θα πρέπει η ερευνητική κοινότητα να εστιάσει και στην ξεχασμένη περιοχή της κλασικής/συμβολικής ΤΝ.

Έχουν δημιουργηθεί επιπρόσθετα συστήματα τα οποία στηρίζονται στις δύο τάσεις της ΤΝ (συμβολική και μη-συμβολική) και απαντούν σε συγκεκριμένα ερευνητικά ερωτήματα

όπως: α) Πώς μπορούμε να προωθήσουμε το WSC, έτσι ώστε να γνωστοποιηθεί σε όσο το δυνατό περισσότερους ερευνητές διαφόρων ερευνητικών υποβάθρων; β) Πώς μπορούμε να δημιουργήσουμε συστήματα που να λειτουργούν ως μετρικές αυτόματης αξιολόγησης της δυσκολίας επίλυσης προβλημάτων WSC από ανθρώπους; γ) Πώς μπορούμε να δημιουργήσουμε συστήματα για την αυτόματη/ημιαυτόματη παραγωγή νέων σχημάτων WSC;

Στην προσπάθειά μας αυτή, προτείναμε τη χρήση προβλημάτων WSC ως μία νέα μορφή CAPTCHA. Παραδοσιακά, οι περισσότερες υπηρεσίες CAPTCHA εξυπηρετούν δύο στόχους: την αποτροπή κακόβουλων επιθέσεων από αυτοματοποιημένα προγράμματα και την γνωστοποίηση προκλήσεων-διαγωνισμών με σκοπό την επίλυσή τους. Στην περίπτωσή μας ο απώτερος στόχος είναι η γνωστοποίηση του προβλήματος στην ερευνητική κοινότητα, ευελπιστώντας στη μακροπρόθεσμη επίλυσή του. Μέσα από τη συμμετοχή μεγάλου αριθμού συμμετεχόντων με συγκεκριμένο πείραμα αξιολογήθηκε η χρήση, η καταλληλόλητα αλλά και η ευχρηστία του WSC σε σχέση με κύριες μορφές CAPTCHA.

Λαμβάνοντας υπόψη ότι οι μελλοντικοί διαγωνισμοί θα πρέπει να οργανώνονται με βάση τον τρόπο επίλυσής τους από τους ανθρώπους, αλλά και ότι η χρήση του WSC ως μιας νέας μορφής CAPTCHA θα πρέπει να μπορεί να παρουσιάζει σχήματα WSC διαφόρων δυσκολιών, έχουμε επίσης αναπτύξει συστήματα λογισμικού, τα οποία λειτουργούν ως μετρικές αυτόματης αξιολόγησης της δυσκολίας επίλυσης προβλημάτων WSC. Η συγκεκριμένη ανάπτυξη συστημάτων συνοδεύεται με πειραματικά αποτελέσματα που αξιολογούν την ποιότητα των προβλέψεών τους σε σχέση με τη δυσκολία που έχουν οι άνθρωποι στα συγκεκριμένα προβλήματα.

Τέλος, γνωρίζοντας ότι υπάρχει περιορισμένος αριθμός σχημάτων WSC (λόγω δυσκολίας δημιουργίας τους) και ότι η επίλυση αλλά και η χρήση του WSC ως μιας νέας μορφής CAPTCHA συνεπάγεται μεγάλες απαιτήσεις σε νέα σχήματα, σε αυτή τη διατριβή έχουμε επίσης προχωρήσει στην ανάπτυξη συστημάτων λογισμικού για την αυτόματη και ημιαυτόματη παραγωγή νέων σχημάτων WSC. Τα συστήματα αναπτύχθηκαν μέσω τεχνικών πληθανάθεσης, επεξεργασίας φυσικής γλώσσας αλλά και χρήσης νευρωνικών δικτύων. Η αξιολόγηση της ποιότητας των παραγόμενων προβλημάτων έγινε με πειράματα σύγκρισής τους με υφιστάμενα προβλήματα που έχουν κατασκευαστεί από ειδικούς στο WSC.

Η διατριβή ολοκληρώνεται παρουσιάζοντας συγκεκριμένα αποτελέσματα και προτείνοντας πιθανές μελλοντικές ερευνητικές κατευθύνσεις, συνοδευόμενες με εισηγήσεις σχετικές με τους ελλείποντες συνδέσμους που απαιτούνται για τη μελλοντική πρόοδο στον ευρύτερο χώρο του προβλήματος.

ABSTRACT

The Winograd Schema Challenge (WSC), a new novel litmus test for machine intelligence, has been proposed to advance the field of Artificial Intelligence (AI). In the last decade, the challenge has received considerable interest as a step towards building machines with commonsense reasoning, humanity's long-willed target since the late fifties.

The WSC refers to resolving pronouns in carefully structured sentences, where the information needed to resolve them is not grammatically present. The challenge consists of pairs of halves (schemas), where each half comprises a sentence, a question referring to an unresolved pronoun, and two possible pronoun targets (answers). It is believed that tackling the challenge will advance the field of AI, helping at the same time the research community to understand human behavior, which relates to the unfolding of the human mechanisms used when answering such questions. In this regard, each WSC instance should tell us something about human behavior, which needs to be explained. Although humans have no difficulties tackling it, such systems' development seems challenging and troublesome.

This dissertation focuses on methods and tools covering multiple aspects of the WSC. Given the AI's tendency to focus on behavior in a purely statistical sense, which can lead to the development of non-transparent systems (sub-symbolic AI), and that human language is not based on word patterns, we start by presenting how we developed a commonsense reasoning system to tackle the WSC. In terms of experimentation, we compare the developed system with well-known coreference resolvers. The compelling advantage of this transparent solution is presented through experiments performed on existing WSC schemas developed by experts in the field. The findings indicate that systems based on classical/symbolic AI must be a part of the solution toward the endowment of machines with commonsense reasoning.

Additional systems based on both classical AI and machine learning were developed to answer research questions such as: a) How can we promote the WSC to various academic disciplines so that they could work on the problem of actually trying to solve the WSC? b) How can we design systems that automatically differentiate between Winograd instances according to their perceived human hardness? c) How can we build systems that automatically build or considerably help humans develop schemas from scratch?

In this regard, we show how we utilized the WSC as a novel form of a completely automated public Turing test to tell computers and humans apart (CAPTCHA). We expect that the adoption and use of a WSC-based CAPTCHA will bring forward the WSC to various academic disciplines to work on the problem of actually trying to solve it, and perhaps, in the process, help build machines able to reason with commonsense knowledge. Experiments we undertook show that a WSC-based CAPTCHA is generally faster and easier to solve than, and equally entertaining as, the most typical existing CAPTCHA tasks.

Based on the fact that this is a challenging task for machines and that future Winograd challenges should be organized according to how humans tackle them, this dissertation also shows how we designed multiple approaches that can automatically differentiate between Winograd instances according to their perceived hardness for humans. According to our results, the automated approaches' performance correlates positively with the performance of humans, suggesting that these kinds of systems could be used as a metric of hardness for WSC instances.

Finally, given that the schema availability is limited and that the schema development process is challenging and troublesome, this dissertation shows how we managed to provide the research community with the necessary tools for designing Winograd schemas from scratch. The undertaken experiments show the benefits of utilizing our developed systems, which, among others, can considerably help humans in the schema development task.

The dissertation concludes with the thesis findings, discussing the implications of this research, accompanied by our thoughts on the missing links required for future progress in the field.

ACKNOWLEDGEMENTS

First and foremost, I would like to express my sincere gratitude to my advisor, Associate Professor Loizos Michael, for his assistance at every stage of the research project and his unwavering support and belief in me. His immense knowledge and ample experience have encouraged me in my studies. His guidance was really influential in shaping my academic research field. Loizos, thank you for your patient support and all of the opportunities I was given to further my research.

I would also like to thank the internal members of my doctoral committee, Professor Antonis Kakas, and Professor Kleanthes K. Grohmann, for their guidance, insightful comments, and encouragement during the preparation stage of my thesis.

I would like to particularly thank the two external examination committee members, Professor Ernest Davis and Associate Professor Katerina Pastra, who provided crucial and constructive feedback on this work. The ideas shared and feedback were valuable in the completion of this thesis. Special thanks to Professor Ernest Davis, whose work on the Winograd Schema Challenge was among those that inspired this dissertation.

Thanks to the Open University of Cyprus for providing me the opportunity to pursue a Ph.D. in Artificial Intelligence. I also want to thank my colleagues at the Computational Cognition Lab for their support during these years.

I am forever indebted to my parents and family for giving me the opportunities and experiences that have made me who I am. Especially, I must express my very profound gratitude to my wife Maria for providing me with unfailing support and continuous encouragement throughout the years of my study and through the process of researching and writing this thesis.

Part of the work reported in this thesis was supported by funding from the EU's Horizon 2020 Research and Innovation Programme under grant agreements no. 739578 and no. 823783, and from the Government of the Republic of Cyprus through the Directorate General for European Programmes, Coordination, and Development.

I would like to dedicate this thesis to Maria.

Contents

List of Figures

List of Tables

1

Introduction

Most people in the AI field trace its foundation to the late fifties (1956), where John McCarthy, Marvin Minsky, Claude Shannon, and Nathaniel Rochester organized a workshop at Dartmouth College (Mitchell, 2019). The proposal (McCarthy et al., 2006), based on which the term Artificial Intelligence (AI) was coined, listed several topics for discussion that have continued to define the AI field to the present day (e.g., abstraction, neural nets, natural language processing) (Mitchell, 2019).

Claude Shannon proposed studying the application of information theory concepts to computing and the synthesis of brain models by emphasizing representing it as a mathematical structure. Marvin Minsky proposed research, initially described in his dissertation, about training machines using input and output channels (training data) through trial and error. Nathaniel Rochester wanted to study the originality in machine performance, the process of invention, and machines with randomness. Finally, John McCarthy proposed examining the relation of language to intelligence because the human mind uses the former to handle complicated phenomena.

Although everyone was very enthusiastic about their own research and high-level optimizations indicated that AI was close in reach, sixty years later, the problem of endowing machines with a deeper understanding of the world remains a challenge (Marcus and Davis, 2019; Mitchell, 2019; Wooldridge, 2020). Of course, the Dartmouth College workshop produced significant outcomes, as it is the place where the AI goals were formed/forged, and that further led to the development of well-known AI labs and projects (e.g., the MIT AI Lab, the Stanford AI project) (Mitchell, 2019). Since 1956 we have been through winters and springs, immense media hype, and failures, where various approaches have been proposed (e.g., classical AI, machine learning). In the end, the AI community was criticized for promising too much and delivering too little (Marcus and Davis, 2019; Wooldridge, 2020).

Questions, such as whether machines could "think", began to arise in 1951 (Wooldridge, 2020). In this regard, Alan Turing, one of the fathers of AI, inspired by the Victorian-era parlor game called the Imitation Game, described what we now call the Turing Test. The basic idea was for human interrogators to tell if someone they interact with, in the form of textual questions and answers, is either a person or a computer program. Since then, various computer programs have claimed to have pass it, though this was done with clever tricks, through which they try to confuse the interrogators into believing that they are interacting with a human being (Wooldridge, 2020). According to Levesque et al. (2012), the problem with the Turing test relates to the free-form conversation nature of the challenge. Even though free-form conversations are the best way to tell how or what someone thinks of something, they are also susceptible to deception and trickery.

This led researchers to establish other similar challenges that seem to be less subject to abuse. To that end, numerous challenges have been proposed towards encouraging the development of systems that will automate, substitute, or enhance basic human abilities and increase the extent to which humans can relate and interact with them.

One of these challenges is the Winograd Schema Challenge (WSC) (Levesque et al., 2012), a carefully crafted pronoun resolution task that seems to be able to capture basic human abilities (Levesque, 2014). The WSC refers to resolving pronouns in carefully structured sentences, where the information needed to resolve them is not grammatically present. The challenge consists of pairs of halves (schemas), where each half comprises a sentence, a question referring to an unresolved pronoun in the sentence, and two possible pronoun targets (answers). In this regard, the WSC can be considered as a novel litmus test for machine intelligence. It is believed that the tackle of the challenge will advance the field of AI, helping at the same time the research community to understand human behavior, which relates to the unfolding of the human mechanisms used when answering such questions. Although humans have no difficulties in tackling it, it seems that the development of such systems is challenging and troublesome (Bender, 2015; Morgenstern et al., 2016). According to Adger (2019), humans see invisible structure, not linear order, which is necessary to understand how pronouns can refer to something. On the other hand, it seems that current developed AI systems do not have that day-to-day commonsense reasoning that humans do (Marcus, 2018; Marcus and Davis, 2019; Mitchell, 2019).

According to Marcus and Davis (2019), commonsense knowledge is the knowledge that is commonly held among people or, simply put, the kind of knowledge we expect ordinary people to possess. It is the knowledge we all humans have that we take for granted. Some of it is innate, and some is acquired throughout our lives without even being aware of it (Mitchell, 2019). The problem is that, even though we know it is something we possess as

humans, nobody seems to know how to build machines with that ability (Marcus and Davis, 2019).

1.1 Motivation

According to Levesque (2014), each WSC instance should tell us something about human behavior, that needs to be explained. With this in mind, we argue that non-transparent solutions are not the best way to tackle Winograd instances. On the other hand, at the time of writing, most approaches rely on statistical-pattern solutions to tackle specific datasets of the WSC. According to Morgenstern (2021), doing well on a test often does not mean excelling at the skills the test was designed to measure. We believe that this high performance is partly because the vast majority of AI systems are trained for specific datasets and objectives, which lead to models that are effective at finding task-specific correlations but lack explainable commonsense reasoning abilities (Marcus, 2018; Marcus and Davis, 2019; Mitchell, 2019).

AI researchers are a competitive bunch, meaning that it comes as no surprise they like to organize online challenges (e.g., Glue, SuperGlue) to drive the field forward (Mitchell, 2019). On the other hand, this is done by developing systems focusing on specific subsets of the WSC, trying to bootstrap it from scratch —e.g., via multiple submissions and retests. Do not get us wrong. Machine learning techniques, like deep learning, are valuable tools for AI that we extensively use in this thesis. However, these kinds of techniques rely on correlation rather than understanding. According to Wooldridge (2020), anyone with data and fast parallel computer hardware could successfully use deep learning to tackle many challenges. The problem is that "showing their work" is not something machine learning systems can easily do (Marcus and Davis, 2019; Mitchell, 2019), meaning that they are opaque and, on occasions, brittle. In this regard, they could be perfect in one situation and utterly wrong in another without showing what led them to make specific decisions.

In this thesis, we make the following claims: First of all, even though Levesque (2014) argued that we need to turn to our roots of knowledge representation and reasoning without treating English text as a monolithic source of information, the design of such systems remains an open research issue and any evidence for this has been mainly anecdotal. Secondly, as this is a particularly new challenge in the field (Levesque et al., 2012), scant attention has been given to finding ways to promote it in various academic disciplines to stimulate research to work on the problem of actually trying to solve it. Thirdly, although it is widely accepted that future challenges should be organized according to how humans tackle them (Bender, 2015), no study has focused on developing systems that could output the perceived human hardness index of Winograd instances. Finally, while the development of Winograd schemas

is difficult and troublesome, requiring motivation and inspiration (Morgenstern et al., 2016; Morgenstern and Ortiz, 2015), minimal evidence exists regarding the development of systems that could automatically develop or considerably help humans in the development task.

Hence, the main research questions examined in this thesis are the following:

- How can we design a system that tackles the WSC based on knowledge representation and reasoning?
- How can we promote the WSC to various academic disciplines so that they could work on the problem of actually trying to solve the WSC?
- How can we design systems that automatically differentiate between Winograd instances, according to their perceived human hardness?
- How can we build systems that automatically build or considerably help humans develop quality schemas from scratch?

1.2 Thesis Contribution

Aiming to advance the AI field, the research community is concerned with the endowment of machines with commonsense reasoning. We believe that a fully-fledged approach to the WSC will bring us closer to the complete tackle of the challenge and the long-term AI goal of endowing machines with commonsense reasoning abilities. Additionally, given that the AI field will benefit from bringing together many different tools that refer to different periods of the AI history (Marcus and Davis, 2019), in this thesis, both classical AI and machine learning are used, which might also help bring together a new generation of AI researchers who appreciate both approaches.

Concerning the first question (How can we design a system that tackles the WSC based on knowledge representation and reasoning?), we start by presenting how we developed a system, which, through the acquisition of commonsense knowledge from the English Wikipedia, tackles the WSC. Given that there is a tendency in AI to focus on behavior in a statistical-sense, and that human language is more than patterns or sequences of words (Adger, 2019), we developed a system that uses commonsense reasoning to tackle the WSC. The developed system shows how day-to-day commonsense reasoning can be operationalized through a densely connected collection of inferential knowledge to tackle the WSC. In this regard, our system could be used as a part of an extensive system to provide an insight into how learning and reasoning through knowledge acquisition can fruitfully interact for pronoun resolution. We expect that our developed system will help researchers focus on knowledge

representation and reasoning and maybe combine good old-fashioned AI (GOFAI) with modern non-symbolic AI to get closer to the long-term AI goal that's been bothering us since the 1950s.

Concerning the second question (How can we promote the WSC to various academic disciplines so that they could work on the problem of actually trying to solve the WSC?), a novel form of CAPTCHA was developed. According to Marcus and Davis (2019), to advance the AI field, AI researchers must be drawn not only from the computer science field but from a wide range of other disciplines, from psychology to linguistics to neuroscience. Based on the fact that the WSC is a challenging task for machines and that CAPTCHAs have also helped promote AI research in various challenge tasks, we will show how we utilized the WSC as a novel form of CAPTCHA to distinguish humans from bots. Specifically, we discuss the nature of this WSC-based CAPTCHA, highlight the shortcomings of typical existing approaches, and provide motivation for a detailed WSC-based CAPTCHA design. We expect that the adoption and use of WSC-based CAPTCHAs will encourage researchers of various disciplines to work on the problem of actually trying to solve the WSC, and perhaps, in the process, help build machines able to reason with commonsense knowledge.

Concerning the third research question (How can we design systems that automatically differentiate between Winograd instances, according to their perceived human hardness?), we will show how we utilized recent research (Bender, 2015) to develop mechanisms that differentiate between Winograd instances. Given that not all schemas can be tackled with the same ease, we will present how we designed and built multiple approaches that can automatically differentiate between Winograd instances according to their perceived hardness for humans as a step towards designing future challenges and as a security mechanism for the WSC-based CAPTCHA service. To that end, we will show how a particular existing system developed for the WSC can form the basis for deriving a data-driven metric of hardness for WSC schemas. Additionally, given that sub-symbolic systems like deep learning are extremely good at correlation tasks (Bengio et al., 2017; François, 2017), we will show how we developed a machine learning system (random forest and deep learning) that outputs the hardness index of any Winograd half faster than any previously used method.

Finally, concerning the last question (How can we build systems that automatically build or considerably help humans develop quality schemas from scratch?), we will show how we managed to provide the research community with the necessary tools for designing Winograd schemas that meet the challenge requirements. Given that the development of schemas is hard and troublesome even for humans, the developed systems come into play as schema replenishment mechanisms and assistants for the schema design process. To that end, we will show how we built a crowdsourced collaboration system that guides humans in the schema

development task. Furthermore, we will show how we built a system that blends NLP and deep learning to automatically develop Winograd instances to considerably help humans in the schema development task.

1.2.1 Research Outcome

The outcome of this work was presented in several international peer-reviewed conference proceedings, workshops, and journals. Additionally, systems developed as a part of this work participated in international competitions all over the world. Specifically, our developed system, Wikisense, tied in first place on the first WSC competition at IJCAI 2016. For another, a modified version of our system participated and took third place on the first Taboo Challenge[1], which took part as a side event of IJCAI 2017 (Rovatsos et al., 2018). Parts of this thesis (e.g., ideas, figures, experimental results) have appeared previously in the following publications:

1. Isaak, N., Michael, L.: Tackling the Winograd Schema Challenge Through Machine Logical Inferences. In: Pearce, D., Pinto, H.S. (eds.) STAIRS. Frontiers in Artificial Intelligence and Applications, vol. 284, pp. 75–86. IOS Press (2016), http://dblp.uni-trier.de/db/conf/stairs/stairs2016.html#IsaakM16

2. Isaak, N. and Michael, L. (2017). How the Availability of Training Material Affects Performance in the Winograd Schema Challenge. In Proceedings of the (IJCAI 2017) 3rd Workshop on Cognitive Knowledge Acquisition and Applications (Cognitum 2017).

3. Isaak, N. and Michael, L. (2017). Tackling the Taboo Challenge with Machine Logical Inferences. In Proceedings of the (IJCAI 2017) ESSENCE Taboo City Challenge Workshop (Taboo 2017).

4. Isaak, N., Michael, L.: Using the Winograd Schema Challenge as a CAPTCHA. In: Lee, D., Steen, A., Walsh, T. (eds.) GCAI-2018. 4th Global Conference on Artificial Intelligence. EPiC Series in Computing, vol. 55, pp. 93–106. EasyChair (2018). https://doi.org/10.29007/rnk8, https://easychair.org/publications/paper/pV9V

5. Isaak, N., Michael, L.: A Data-Driven Metric of Hardness for WSC Sentences. In: Lee, D., Steen, A., Walsh, T. (eds.) GCAI-2018. 4th Global Conference on Artificial

[1] https://www.essence-network.com/wp-content/plugins/really-static/static/challenge/challenge-rules/

Intelligence. EPiC Series in Computing, vol. 55, pp. 107–120. EasyChair (2018). https://doi.org/10.29007/398z, https://easychair.org/publications/paper/nRrp

6. Isaak, N. and Michael, L. (2019). WinoFlexi: A Crowdsourcing Platform for the Development of Winograd Schemas. In Liu, J. and Bailey, J., editors, AI 2019: Advances in Artificial Intelligence, pages 289–302, Cham. Springer International Publishing.

7. Isaak, N. and Michael, L. (2020). WinoReg: A New Faster and More Accurate Metric of Hardness for Winograd Schemas. In Danoy, G., Pang, J., and Sutcliffe, G., editors, GCAI 2020. 6th Global Conference on Artificial Intelligence (GCAI 2020), volume 72 of EPiC Series in Computing, pages 46–58. EasyChair.

8. Isaak., N. and Michael., L. (2020). Winventor: A Machine-driven Approach for the Development of Winograd Schemas. In Proceedings of the 12th International Conference on Agents and Artificial Intelligence - Volume 2: ICAART, pages 26–35. INSTICC, SciTePress.

9. Isaak, N. and Michael, L. (2021). Experience and Prediction: A Metric of Hardness for a Novel Litmus Test. Journal of Logic and Computation. exab005.

10. Isaak, N. and Michael, L. (2021a). Blending NLP and Machine Learning for the Development of Winograd Schemas. In Rocha, A. P., Steels, L., and van den Herik, J., editors, Agents and Artificial Intelligence, pages 188–214, Cham. Springer International Publishing.

Additionally to those mentioned above, three WSC datasets were created as an outcome of this thesis[2]:

1. A dataset consisting of 29 Greek Winograd schemas that meet all the WSC criteria developed by a Greek literature teacher.

2. A dataset consisting of 135 Winograd schemas (270 halves) developed by crowdworkers.

3. A dataset consisting of hundreds of automatically developed Winograd instances (schemas/halves).

[2]http://www.nicosisaak.info

1.3 Thesis Structure

The remainder of the thesis is structured as follows. Chapter 2 shows various central aspects of the WSC, what the challenge is about and why world knowledge and the ability to reason seem necessary to tackle it. Then, the related work section is presented. Given that the WSC has been a topic of interest for nearly a decade, multiple datasets along with systems were introduced for tackling the challenge.

Following, Chapter 3 presents Wikisense, a system that focuses on commonsense knowledge to tackle the WSC. Wikisense, which retrieves its knowledge via a supervised learning approach, shows how learning and reasoning through knowledge acquisition can fruitfully interact for the pronoun resolution. It shows, among others, how the acquisition and the extraction of general inference rules could help us tackle the WSC. To that end, we show how Wikisense utilizes the Websense engine (Isaak, 2011; Michael, 2013), an engine able to respond to user queries provided in natural language text, with inferences that are implied by the given queries according to the collective human knowledge. The proposed framework of how Wikisense utilizes the Websense engine is described and applied in the experimental section.

Chapter 4 presents how we utilized the WSC to develop a new type of CAPTCHA in order to bring more AI research into the field. Although CAPTCHAs were established as a standard technology to confidently distinguish humans from bots, beyond their typical use, they have also helped promote AI research in various challenge tasks. Based on current reports in the literature, the WSC remains a challenging task for bots and is, therefore, a candidate to serve as a novel form of CAPTCHA. Although there are a finite number of words in a language, there is an infinite number of sentences that can be built (Adger, 2019), meaning that there is a long tail of completely unpredictable Winograd instances that could be built to serve as CAPTCHAs. In this regard, Chapter 4 shows how we investigated whether this a priori appropriateness of the WSC as a form of CAPTCHA could be justified in terms of its acceptability by the human users in relation to existing CAPTCHA tasks.

Future challenges should be organized according to how humans can tackle them (Bender, 2015). Additionally, systems able to differentiate between Winograd instances could be used to ensure that the CAPTCHA service would display harder schemas to solve in the case of possible fraudulent actions. In this regard, Chapter 5 shows how we developed systems that can be used as a metric of hardness for WSC instances. The first system utilizes Wikisense to show how the performance of a particular automated approach varies with the availability of training material. To that end, we compare the results of the automatic approach with two studies, one from the literature and one that we designed and undertook. The second system adds to previous research by presenting a new system, which, based on machine learning,

outputs the hardness of any Winograd half faster and more accurately than previously used methods. At the same time, along with our developed system, we show how we extend previous work by presenting the results of a large-scale experiment we undertook.

Chapter 6 is related to the availability of Winograd schemas, which is not sufficient. Since the development of new schemas by individuals is in itself a rather challenging task, and as the WSC has been proposed as a basis for a novel form of CAPTCHA, such uses of the task necessitate the availability of an extensive and presumably continuously replenished collection of available Winograd schemas. Towards addressing this issue, we present our developed systems. The first, WinoFlexi, is a flexible online platform system that considerably helps humans in the schema development task. The second system, Winventor, blends NLP with deep learning and attempts to fully automate the schema development process to considerably help humans in the schema development task.

Finally, Chapter 7 summarizes the thesis findings, where we discuss the implications of this research accompanied by suggestions for future research directions. Given that endowing machines with a deeper understanding of the world remains a challenge, Chapter 7 closes with the discussion section, where we present our final thoughts on the missing links required for future progress in the field.

2

The Winograd Schema Challenge: Background and Related Work

2.1 Background on the WSC

The Winograd Schema Challenge (WSC) is a novel litmus test for machine intelligence and a variant of the well-known Recognizing-Textual-Entailment challenge (RTE) (Dagan et al., 2005) that is able to capture basic human abilities (Levesque et al., 2012). The challenge was named after Terry Winograd because of a well-known example that was taken from his doctoral thesis, justified in terms of machine translation, and modified accordingly to meet the challenge difficulties (see 2.1) (Davis, 2016).

The WSC is one of the most challenging tests for machines currently available and is under the affiliated organization of Commonsense-Reasoning[1] (Marcus and Davis, 2019). According to Levesque (2014), the WSC has been proposed as an alternative to the well-known Turing Test. Given that it is a challenge that tests for reading comprehension, it seems less subject to abuse by verbal tricks and canned responses (Levesque, 2014).

The challenge is about resolving ambiguities because the information needed is not grammatically present. Passing the challenge requires resolving pronouns in certain sentences where shallow parsing techniques do not seem to be directly applicable, and the use of world knowledge and the ability to reason seem necessary (Levesque, 2014). Simply put, it is a challenge that requires participants to answer binary questions referring to a definite pronoun in the sentence without having to depend on a precise notion of entailment (Levesque et al., 2012). Each Winograd schema comprises two halves, with each half consisting of a sentence, a definite pronoun, or a question, two possible pronoun targets (answers), and the correct

[1] http://commonsensereasoning.org/winograd.html

pronoun target (see 2.1). While given just one of the halves, the aim is to resolve the definite pronoun, through the question, to one of its two pronoun targets. To avoid trivializing the task, the pronoun targets are of the same gender, and both are either singular or plural. Moreover, the two halves differ in a special word or phrase that critically determines the correct pronoun target. Schemas that do not *strictly* follow these rules are called "schemas in the broad sense".

Given a Winograd schema to answer, people can anticipate, and reason about, causes and effects (Sap et al., 2019), and by realizing who did what to whom, when, where, and why (Gary Marcus, 2019; Marcus, 2018), they can easily tackle it. For instance, if someone gives us a Winograd half, "Sentence: The city councilmen refused the demonstrators a permit because they feared violence." "Question: Who feared violence?", "Pronoun targets: The city councilmen, The demonstrators", and asks us to resolve the definite pronoun "they" or just answer the question, we can easily infer that the correct answer is "The city councilmen". This example shows how we, humans, through commonsense reasoning, can easily tackle Winograd instances. We understand this because of our knowledge about the city councilmen, demonstrators, and politics (Winograd, 1972). For example, having the commonsense knowledge that city councilmen are often afraid of having other people participate in demonstrations leads us to conclude that they [the councilmen] are the ones who "feared violence".

According to Adger (2019), we humans use special abilities when we learn languages, as this is a part of our nature (starting with Chomsky (1959) review of Skinner). In this regard, we can unconsciously sense the abstract structure of sentences when we hear them, which is necessary to understand how pronouns refer to things. Not surprisingly, human adults tackle the challenge with an average score of 92%, which sets the bar high (Bender, 2015). However, it seems that the development of such systems is challenging and troublesome (Bender, 2015; Morgenstern et al., 2016).

Although most AI developed systems seem to focus on the technology of AI that gets all the attention, we should be focusing on the science of AI, which might help us understand how we can do something, or as stated by Levesque, *how is it possible for something physical (like people, for instance) to actually do X?*. In this regard, we should think of people's behavior in resolving ambiguities as something that needs to be explained, where even a single half can show us something important (Levesque, 2014).

Furthermore, well-designed schemas can test for different kinds of expertise, or problem-solving skills, or for an ability to visualize (Levesque, 2014). For instance, the following schema concerns certain material regarding COVID-19 vaccines (the special word is given in bold): *Sentence: Vaccines like Pfizer are better than Sputnik-V. They are based on the mRNA/adenovirus technology. Question: Which vaccines are based on the* ***mRNA/adenovirus***

technology? Pronoun targets: Pfizer Vaccines, Sputnik-V Vaccines. To answer such a schema correctly, one needs to know that the Pfizer vaccines are based on the mRNA technique, and the Sputnik-V vaccines use adenovirus-based technology.

Another one that tests problem-solving skills is the following: *Sentence: The sack of eggs at Morrisons supermarket was placed above the sack of wet eggs, so it had to be moved **first/second**. Question: What had to be moved **first/second**? Pronoun targets: the sack of eggs, the sack of wet eggs*. To answer such a schema, we only need to know that "when X is above Y, then X has to be moved first". So, one needs to have this commonsense knowledge and know what item is above and below. In this example, the contextual/circumstantial knowledge of wet eggs does not make any difference in the resolution of the pronoun. Hence, in the first half, the answer to the question "What had to be moved first?" is the "sack of eggs", and in the second "What had to be moved second?" is the "sack of wet eggs". However, with a slightly different example: *Sentence: The sack of eggs and the sack of wet eggs at Morrisons supermarket were placed on top of each other, so it had to be moved **first/second**. Question: What had to be moved **first/second**? Pronoun targets: the sack of eggs, the sack of wet eggs*, the contextual/circumstantial knowledge about wet eggs makes the difference in the resolution of the pronoun. In this regard, to answer such a schema, we need to know that because of COVID-19 food safety measures, a sack of wet eggs means sacks of preservative liquid which cause the eggs to float —It seems that this is a new way to sell lots of hard-boiled eggs destined for salad bars. Then, to have the commonsense knowledge that the sack of eggs had to be placed above the sack of wet eggs for safety reasons —if we placed the sack of wet eggs above the sack of eggs, we would probably break the eggs. Hence, in the first half, the answer to the question "What had to be moved first" is the "sack of eggs", and in the second "What had to be moved second" is the "sack of wet eggs".

According to Levesque et al. (2012); Levesque (2014), when constructing Winograd schemas, we should avoid making them either hard (e.g., ambiguous to answer) or too obvious to resolve (e.g., non-Google-proof answers). For instance, an obvious way to resolve a schema using the Google search engine is the following: *Sentence: The rabbit easily passed by the turtle because it was going so **fast/slow**. Question: What was going so **fast/slow**? Pronoun targets: the rabbit, the turtle.* At the time of writing, if we combine the first half's question and search Google with the keywords:"fast rabbit" and "fast turtle", we can easily resolve the definite pronoun to the rabbit ("fast rabbit": 975,000 and "fast turtle": 305,000). In the same regard, in the second half, we can resolve the definite pronoun *it* to the turtle ("slow rabbit": 344,000 and "slow turtle": 574,000). It seems that in schemas that are too obvious to resolve, one can implement a simple system that ignores the sentence structure to search on engines like Google, which pair of words co-occur more frequently.

An example of a schema whose answers are not obvious enough largely depends on how much a person knows or on the shared common ground between interlocutors, not on commonsense knowledge. As stated by Levesque et al. (2012), we have schemas that are a "near-miss", meaning too ambiguous to answer —e.g., one of the special words or one of the answers makes it too hard to resolve the schema or one of the two halves correctly. For instance, the schema *Sentence: John was **happy/jealous** when Nicos said that he tackled the WSC with promising results. Question: Who tackled the WSC? Pronoun targets: John, Nicos* is too ambiguous to resolve when used with the special word "happy". This happens because the answer of the first half can be either John or Nicos, as it relies on the shared common ground between interlocutors regarding their knowledge of the two pronoun targets. In this regard, someone reading only the first half without knowing the specific people and their relationship cannot answer it with certainty.

Challenges like the WSC aim to tackle the goal of endowing machines with human commonsense reasoning abilities. By extension, it is believed that a system that contains the commonsense knowledge to resolve Winograd schemas correctly should be capable of supporting a wide range of AI applications (Levesque et al., 2012).

Given that the WSC is a challenge that has been proposed as the means to understand human behavior (Levesque, 2014), it seems that non-transparent approaches —approaches that cannot explain their reasoning— do not seem applicable (Levesque, 2014). For instance, Winograd (1972) argued that no syntactic or semantic rules could capture this kind of ability without world knowledge. However, as with the Turing test, many researchers claim to pass it, though this is done with opaque statistical solutions that focus on discovering patterns of words in WSC schemas. It seems systems can discover tricks or systematic bias in words to tackle schemas without showing commonsense reasoning abilities as humans do.

For this thesis's scope, we started working on the WSC at the time it was introduced (2012). Since then, many have implemented well-established solutions and benchmark datasets, mainly based on machine learning techniques. Below, we will start by presenting the developed datasets and continue with the various approaches built to tackle the challenge.

2.2 Available Datasets

The WSC has been a topic of interest for nearly a decade. Since its introduction, multiple datasets have been developed, aiming to assist the development of systems to tackle the challenge. To the best of our knowledge, the only one that meets the challenge restrictions is the WSC_273 dataset, introduced back at the same period with the challenge itself.

A Winograd schema		
1st half	Sentence	The city councilmen refused the demonstrators a permit because they feared violence.
	Question	Who feared violence?
	Pronoun Targets	The city councilmen, The demonstrators
	Correct Answer	The city councilmen
2nd half	Sentence	The city councilmen refused the demonstrators a permit because they advocated violence.
	Question	Who advocated violence?
	Pronoun Targets	The city councilmen, The demonstrators
	Correct Answer	The demonstrators

Table 2.1 A Winograd schema example. The schema consists of two halves, and the objective is to resolve the definite pronoun through the question in each half.

2.2.1 WSC_273: The Original Dataset of Winograd Schemas

WSC_273 is the dataset introduced back in 2012 with the WSC. It consists of schemas manually constructed by experts in the field (Levesque et al., 2012). In this regard, all of the developed schemas strictly follow the WSC rules (see 2.1). Although it consisted of 100 schemas back in time, at the time of writing, it consists of 150 schemas. Given that schemas were added throughout various periods, several authors usually referred to them as WSC_, where _ represents the number of halves they used in their research.

2.2.2 DPR: The Definite-Pronoun-Resolution Dataset

This dataset was introduced by Rahman and Ng (2012). The dataset consists of schemas manually developed by thirty undergraduate university students under the "broad-sense" flag. Although some constraints on the Winograd schemas have been relaxed, it remains a challenging dataset (see Chapter 5). The DPR dataset consists of 943 schemas, where each half consists of a sentence, a definite pronoun, two possible pronoun targets, and the correct pronoun target. An example of such schema is the following:

1. First half: Sentence: The geology department petitioned the school board for money because they needed funding; Definite-pronoun: they; Pronoun-targets: The geology department, the school board; Correct-answer: The geology department.

2. Second half: Sentence: The geology department petitioned the school board for money because they could grant funding; Definite-pronoun: they Pronoun-targets: The geology department, the school board; Correct-answer: the school board.

2.2.3 PDP: The Pronoun-Disambiguation-Problem Dataset

The PDP dataset was developed to be used as a preliminary test before the actual round of the first WSC, which took place in 2016 (Morgenstern et al., 2016), though no team did

well enough to enter the WSC round. PDPs were manually collected, vetted, and sometimes modified from classic and popular literature, where the whole process ended with a new dataset consisting of 122 PDPs (Kocijan et al., 2020). Specifically, 62 examples of PDPs were used before the WSC as a testing set, and 60 PDPs were included in the challenge, which was administered as a side event at IJCAI 2016 (Davis et al., 2017) [all of the examples were previously evaluated by 21 human participants (Davis et al., 2016)]. Given that the PDPs were planned to be used as a preliminary test, they do not meet the WSC criteria, meaning that in each PDP, there is no an associated special word, two possible answers, or even a companion half. An example of such a PDP is the following: *Phrase: Sergeant Holmes asked the girls to describe the intruder. Nancy not only provided the policeman with an excellent description of the heavyset thirty-year-old prowler, but drew a rough sketch of his face. Snippet: rough sketch of his face. Possible-answers: (a) Sergeant Holmes (b) the intruder. Correct-Answer: (b) the intruder.*

2.2.4 LANG: Schemas in Other Languages

These are schemas translated from English or newly developed in other languages.

French, Portuguese and Chinese Schemas

Schemas of the original collection of Winograd schemas (WSC_) were translated into French, Portuguese, and Mandarin Chinese. However, there were reports that some changes needed to be made to the content to avoid unintended cues, such as grammatical gender (Kocijan et al., 2020).

Amsili and Seminck (2017) translated 144 Winograd schemas that were available at the time of writing their paper (WSC288). However, the process they undertook was able to adapt/translate only 107 Winograd schemas —e.g., for a number of schemas they could not find a direct translation into French.

Melo et al. (2019), via the help of three native Portuguese speakers, manually translated 285 original Winograd halves (WSC285) into Brazilian Portuguese[2]. As with Amsili and Seminck (2017), the translation process was developed following the rules of the challenge (Levesque et al., 2012), though eight sentences were rejected because they could not find a suitable translation to Portuguese.

Bernard and Han (2020) introduced Mandarinograd, a collection of 154 Winograd schemas translated into simplified Mandarin Chinese, adapted from the original dataset

[2]https://github.com/gabimelo/portuguese_wsc

(WSC_). When a direct translation was not possible, the authors tried to produce a thematically related example.

2.2.5 WNLI: The Winograd-Natural-Language-Inference Dataset

This is a textual entailment version of the WSC, which is part of the GLUE benchmark[3] (Wang et al., 2018). The authors converted each examined schema half into a sentence pair classification by replacing the ambiguous pronoun, in each half, with each possible referent. According to Wang et al. (2018), each half was manually constructed to thwart simple statistical methods to determine if the half's original sentence entails the sentence with the pronoun substituted. The dataset consists of 70 validation examples, 145 test examples, and a total of 634 training examples, which do not come in pairs as many of them do not contain a special word (Kocijan et al., 2020). An example that shows how the original half (see 2.1) was modified is the following: *Sentence: The city councilmen refused the demonstrators a permit because they advocated violence. Hypothesis: The demonstrators advocated violence. True/False: True*.

2.2.6 WNLI+: The Winograd-Natural-Language-Inference-Plus Dataset

WNLI+ refers to the SuperGlue benchmark (Wang et al., 2019a), named after GLUE with a new set of more challenging Natural Language Understanding (NLU) tasks. Regarding the WSC, although WNLI+ uses the same examples as the WNLI, it is used as a pronoun resolution problem. Specifically, each example consists of a sentence with a marked pronoun and noun. Their positions in the sentence are specified, where the task is to determine if the pronoun refers to the specified noun. Like with WNLI, below is an example that shows how the original half (see 2.1) was modified: *Sentence: The city councilmen refused the demonstrators a permit because they advocated violence., target: span2_index: 9, span1_index: 4, span1_text: The demonstrators, span2_text: they, idx: 16, label: true*.

2.2.7 WGEN: The WinoGender Dataset

The WinoGender dataset, inspired by the WSC, was introduced by Rudinger et al. (2018) to reveal cases where coreference resolution systems mistakenly chose their answer based on gender bias. At the time of writing, WinoGender consists of 120 manually written templates. Compared to the WSC, in each example, all of the possible pronouns can be included to create valid sentences (Kocijan et al., 2020). In this regard, the gender of the pronoun does

[3]https://gluebenchmark.com/

not affect the expected answer. In each sentence, one of the two candidates refers to an occupation presumably believed to have a highly imbalanced ratio between the two genders. Below, a Winogender example for a nurse and a patient: *The nurse notified the patient that: 1) her shift would be ending in an hour. 2) his shift would be ending in an hour. 3) their shift would be ending in an hour.*

2.2.8 WBIAS: The WinoBias Dataset

Like WinoGender, the WinoBias dataset (Zhao et al., 2018) was developed as a benchmark to identify gender bias in coreference resolution problems. These are Winograd-schema-style examples that contain entities corresponding to people referred to by their occupation (e.g., doctor, carpenter). For their experiments, Zhao et al. (2018) developed 3,160 sentences, which were split equally into development and testing sets. As in the WGEN dataset, in each example, the gender of the pronoun is irrelevant for the coreference resolution. To build their schemas, they used two different templates. The first, more challenging template follows the structure [entity1] [interacts with] [entity2] [conjunction] [pronoun] [circumstances] and the second, [entity1] [interacts with] [entity2] and then [interacts with] [pronoun] for [circumstances]. Note that, as with the WinoGender dataset, the WSC criteria are not met, meaning there are no special words. An example of the first template is the following: *half1: [The developer] argued with the designer because [she] did not like the design. half2: The developer argued with [the designer] because [his] idea cannot be implemented.*

2.2.9 WGRAN: The WinoGrande Dataset

The WinoGrande_all dataset is a large-scale dataset of 44 thousand Winograd-like examples collected via crowdsourcing (Sakaguchi et al., 2020). Though the WSC inspired the development of the dataset, the WinoGrande dataset was adjusted to improve the scale and the hardness of the dataset —for instance, it is formatted as a fill-in-the-blank problem. Like WinoGender and WinoBias, it was developed to discover if recent advances in the field have been based on spurious biases found in the other datasets. Sakaguchi et al. (2020) also filtered the WinoGrande_all to build an unbiased dataset of examples, called WinoGrande_debiased, which consists of 12,282 examples. To that end, they used an ensemble of linear classifiers, trained on random subsets of the data, and discarded the examples that were resolved by more than 75% of the classifiers. Here is an example taken from the WinGrande dataset: *Example1> sentence: Ian volunteered to eat Dennis's menudo after already having a bowl because _ despised eating intestine., option1: Ian, option2: Dennis, answer: 2. Example2>*

sentence: Ian volunteered to eat Dennis's menudo after already having a bowl because _ enjoyed eating intestine., option1: Ian, option2: Dennis, answer: 1.

2.2.10 KnowRef: The KnowRef Dataset

Emami et al. (2019), introduced KNOWREF, a new benchmark for coreference resolution, which consists of 8,724 pronoun disambiguation problems extracted from sources like Wikipedia and Reddit. Through various techniques, which incorporated human annotation, they developed Winograd-like examples, though they came without questions. Like in the original Winograd schemas, they removed gender and number cues to make their dataset harder to resolve away from biases that might make the resolving easier. Each example contains a target pronoun that must be correctly resolved to one of two possible antecedents. For instance, *[Paul] helped [Lionel] hide when [he] was pursued by the authorities*.

2.2.11 MaskedWiki: The MaskedWiki Dataset

This is an extensive collection of sentences developed by Kocijan et al. (2019b) from the English Wikipedia. The dataset, which was developed to fine-tune their Language Model (LM) to tackle the WSC, does not seem to fulfill all the requirements of the WSC. The dataset consists of 2.4 Million examples constructed by masking repeated occurrences of nouns. In this regard, it contains sentences with (at least) two occurrences of the same noun where the second occurrence is masked. For instance, *Sentence: He was ordained in 1843 and was awarded a Bachelor of Divinity in 1850 followed by a Doctor of [MASK] in 1872. Targets: Divinity, Doctor. MASK: Divinity*.

2.2.12 WIKICREM: The Wikipedia-CoREferences-Masked Dataset

This is a dataset of 2.4 million examples, generated by Kocijan et al. (2019a) in the same way as MASKEDWIKI, albeit it masks only personal names. In this regard, the dataset contains sentences that contain at least two occurrences of the same personal name, where the second occurrence is masked. For instance, *Sentence: Gina arrives and she is furious with Denise for not protecting Jody from Kingsley, as [MASK] was meant to be the parent. Targets: Gina, Denise, MASK: Denise*.

2.2.13 GAP: The Gendered-Ambiguous-Pronouns Dataset

Webster et al. (2018) introduced the GAP dataset for the GAP challenge. According to the authors existing systems do not capture ambiguous pronouns in sentences meaning that they

are mainly based on gender bias to tackle coreference resolution problems. Although they cannot be considered Winograd schemas, Winograd schemas are related to their work as they contain ambiguous pronouns. Experiments showed that gender bias in existing corpora favors masculine entities. To address issues related to gender bias, they developed GAP, which consists of about 8,908 ambiguous pronoun-name pairs derived from Wikipedia. The dataset contains a development and testing set of 4,000 examples and 908 examples for parameter tuning. An example of such a sentence is the following: *Sentence: In May, [Fujisawa] joined [Mari Motohashi]'s rink as the team's skip, moving back from Karuizawa to Kitami where she had spent her junior days. Ambiguous pronoun: she. Potential coreferent-names: Fujisawa, Mari Motohashi. Correct: Fujisawa.*

2.3 Related Work

As stated in the literature (Kocijan et al., 2020), since the WSC was introduced, various types of systems have been used to tackle it, such as feature-based, neural-based, and language model approaches (see Table 2.2). For the purpose of this thesis, feature-based approaches that incorporate some form of commonsense knowledge will be considered knowledge-based approaches. In contrast, feature-based approaches that incorporate machine learning will be considered machine learning techniques as the neural-based and the language model approaches.

Given that the WSC is a task that requires high-level language-like descriptions and logical reasoning, knowledge-based approaches are mainly based on the acquisition of some form of commonsense reasoning. On the other hand, neural approaches are primarily trained on unstructured or pre-trained data. Specifically, language model approaches are neural approaches that use large pre-trained language models, sometimes fine-tuned on other datasets to maximize their performance.

2.3.1 Knowledge-based Approaches

Sharma et al. (2015) system is based on Answer Set Programming (ASP) (Baral, 2003; Gelfond and Lifschitz, 1988), where they use a general-purpose parser (K-parser) to parse and answer Winograd schemas. K-parser is used to retrieve background knowledge directly from Google search queries (e.g., ".*not.*lift.*because.*weak.*"), forcing the search engine to return specific sentences that are semantically and structurally similar to the given WSC half. According to their results, the ASP-based technique rejects a large amount of WSC

	WSC	DPR	PDP	WNLI	WGEN	WGRAN	KnowRef
Knowledge-based Approaches							
Sharma et al. (2015)	70%						
Sharma (2019)	84%						
Sharma (2019) - KParser	42%						
Hong and Bennett (2020) - (Ensembled)						80%	
Hong and Bennett (2020) - (Knowledge-Based)						37.5%	
Feature-based Approaches							
Fahndrich et al. (2018)			74%				
Budukh (2013)	73%						
Peng et al. (2015)		76%					
Emami et al. (2018)	57%						
Machine Learning Approaches							
Rahman and Ng (2012)		73.05%					
Liu et al. (2017)	70%						
Zhang and Song (2018)	60%						
Wang et al. (2019b)	62.4%		78.3%				
Opitz and Frank (2018)	56%	63%					
Trinh and Le (2018)	63.7%		70%				
Radford et al. (2019)	70.7%						
Prakash et al. (2019)	71.06%						
Prakash et al. (2019)	70.17%						
Klein and Nabi (2019)	60.3%		68.3%				
Kocijan et al. (2019b)	72.5%			74.7%			
Raffel et al. (2019)				94.5%			
Kocijan et al. (2019a)	71.8%	84.8%	86.7%	74.7%	82.1%		
Ye et al. (2019)	75.5%			83.6%			
He et al. (2019)	75.1%		90%	89%			
Ruan et al. (2019)	71.1%						
Liu et al. (2019)				91.3%			
Sakaguchi et al. (2020)	90.1%	93.1%	87.5%	85.6%		79.1%	85%
Lin et al. (2020)						77%	
Brown et al. (2020) - (zero-shot)	88.3%					70.2%	
Brown et al. (2020) - (one-shot)	89.7%					73.2%	
Brown et al. (2020) - (few-shot)	88.6%					77.7%	

Table 2.2 Results of several approaches on the various datasets of the Winograd challenge. Please note that some of the methods used are applied to subsets of the datasets.

sentences. Specifically, it can only test two types of WSC halves, causal and direct causal (38% of the WSC282 dataset), where it correctly resolves 70% of them.

In an additional work Sharma (2019), via additional knowledge, built on top of graph-subgraph isomorphism encoded using ASP, they were able to tackle 240 out of 285 examples (WSC285), albeit the input and background knowledge had to be manually provided by a human —both hand-written graph representation and background-knowledge of each example was provided. On the other hand, the automatic extraction of knowledge using the K-Parser was able to solve only 120 examples (Kocijan et al., 2020).

According to Bailey et al. (2015), the WSC cannot be solved without human-like reasoning. They examined two types of relations for the purpose of establishing discourse coherence sufficient to tackle Winograd schemas. To accomplish this task, they introduced a theoretical framework of rules that could justify the solutions to a small subset of schemas.

Schüller (2014) relates the WSC to relevance theory that implies humans easily prefer one of the two answers in each examined half. Schuller's system combines the examined sentence dependency graph (via Stanford Parser) with a manually created background knowledge graph, where it extracts the answer based on relevance theory and via ASP. Experiments ran on four schemas from the original WSC200 dataset, where it was shown that certain parameter combinations could lead to correct disambiguation of schemas.

Hong and Bennett (2020) tackled domain-specific Winograd schemas by applying various techniques. One of them was a high-level knowledge-based reasoning method based on Sharma (2019) research, and an ensemble that combines knowledge-based reasoning with machine learning techniques to mitigate each method's weaknesses. With the term domain-specific, the authors refer to schemas that consist of sentences that relate to the usual sense of *thanking*. In total, the thanking domain consisted of 171 extracted sentences from the WinoGrande dataset based on relationships of "owing" and "being owed". Experiments performed on the extracted dataset showed that, on average, their ensembled method achieved 80% accuracy (32/40), and the knowledge-based reasoning method 37.5% accuracy (15/40).

2.3.2 Feature-based Approaches

Fähndrich et al. (2018), using sources like WordNet, Wikipedia, or domain ontologies, built semantic graphs to tackle schemas from the PDP dataset. The graphs are merged, where the resulting graph contains the facts about the words used in each examined half with additional semantic (PropBank) and syntactic information (Stanford CoreNLP). Next, according to a manually designed set of rules and through marker placing, which, according to the authors, models how semantic memory is used for reasoning in humans, they resolve the definite

pronoun to the pronoun-target with the greatest number of markers after a number of steps. Experiments show that this method achieves 74% accuracy on the PDP dataset.

Budukh (2013) developed a system consisting of four answering modules to tackle the WSC260. However, experiments showed that only 34% of the dataset could be tested. The system rejected a large number of halves as it failed to resolve the pronouns that refer to people. Finally, the resulting system, which uses world knowledge with an aggregation mechanism (ConceptNet, Web Queries, Narrative chains, sentiment analysis), achieves an average score of 73%.

Peng et al. (2015) achieved 76% accuracy on DPR using integer linear programming. They acquire statistics in an unsupervised way from multiple knowledge resources (Gigaword corpus, Wikipedia Wikifier, Web Queries, and polarity information) through the training of a coreference model by learning a pairwise mention scoring function. They separate the halves into three types and try to solve the first two. Their system accepts the sentence, the target pronoun, and the two pronoun targets as input, and it does not exploit the fact that each half comes in pairs in training or testing. Although their system achieves a high prediction score (76%), it fails to answer 27% of halves that belong to the third type.

Emami et al. (2018), developed a Web knowledge-hunting system, which was able to tackle the WSC275 with 57% accuracy. Their developed model, which is based on on-the-fly knowledge-hunting than reasoning, operates in four stages. When given a half, it develops a set of queries to capture the predicates and sends them to a search engine to retrieve relevant snippets, which are then parsed and filtered to figure out the correct pronoun target.

2.3.3 Machine Learning Approaches

Rahman and Ng (2012) introduced the first system that tackled the WSC, along with the DPR dataset. Their system uses machine learning to combine features derived from various knowledge sources (Web Queries, FrameNet, Opinion-Finder, narrative chains, semantic compatibility). Their system tries to find the best pronoun target in each half through a Ranking-based approach, based on Joachims' SVM-light package (Joachims, 2002). Although this technique achieves 73.05% accuracy, it fails when halves are equally alike (Sharma et al., 2015).

Neural Approaches

Liu et al. (2017) introduced a neural network approach trained in cause-effect relationships, such as pairs of words from text corpora. The model was trained to learn the association relationships between any two discrete events to predict whether the second part of the half

is the consequence of the first one (Kocijan et al., 2020). The neural network approach achieved 70% accuracy on a manually selected subset of 70 Winograd schemas, taken from the WSC273 dataset.

Zhang and Song (2018) use a distributed representation approach, where via unsupervised training, they tackle a subset of a manually selected set of 92 schemas (WSC273), with an average score of 60%. The model tries to transfer the meaning of an examined verb, in each half, to the definite pronoun (e.g., "fear/advocate" in Table 2.1), through the employment of word embeddings. Using the spaCy dependency parser, they parse the training data found from Wikipedia to develop dependency-based word embeddings, following the Word2Vec embeddings (Mikolov et al., 2013).

Wang et al. (2019b) proposed two unsupervised models, based on Deep Structured Similarity Model (DSSM) framework, to tackle the PDP dataset with 78.3% accuracy and the WSC273 dataset with 62.4% accuracy. Their approach uses WSC and PDP, as a pairwise ranking problem. In this regard, the aim is to generate a score that shows the correct pronoun target for each half. For instance, in the first example, in Table 2.1, "councilmen, they" gets a higher score than the incorrect one ("demonstrators", "they"). According to the authors, with their bi-directional LSTM model, they aim to capture the semantic meaning of the definite pronoun and the pronoun target based on the sentences where they occur. This is encoded into contextual representations by deep neural networks where they compute their coreference scores (Wang et al., 2019b). For training purposes, to create positive and negative samples, they leverage linguistic patterns from raw text based on the assumption that the definite pronoun refers to one of the preceding nouns.

Opitz and Frank (2018), via the training of bi-directional LSTM models (Siamese model), approach the WSC as a sequence ranking task. According to the authors, their work was the first to focus on both the DPR and the WSC dataset by presenting an end-to-end WSC system that does not rely on linguistic annotation. The Bi-LSTM-based models are trained to rank the sentence with the correct pronoun target higher than the sentence with the incorrect pronoun target. According to the authors, placing the problem as a sequence preference ranking task has two major advantages: i) it contextualizes each of these candidates to the definite pronoun's local context; ii) the two alternative sentences can define a model that determines which one can be considered more plausible. Cross-dataset experiments performed, that is, training on the DPR and testing on the WSC, showed that it is not trivial to generalize when presented with a different, smaller WSC data set —they achieved an accuracy of 63% on DPR and 56% on the WSC273 dataset.

Language Model Approaches

Trinh and Le (2018), utilizing an ensemble of LSTM language models, pre-trained on a large corpus of unlabeled data (LM-1-Billion, CommonCrawl6, SQuAD, and Gutenberg Books), tackled the WSC273 with an accuracy of 63.7% and the PDP dataset with an accuracy of 70%. They developed two sentences from each examined schema via replacing the definite pronoun with each candidate. Their ensemble model had to choose the one that results in a more probable English sentence (the one with the highest probability). In this regard, the ensemble of language models, which encodes human knowledge found in various corpora, can assign a higher probability to the sentence based on what they previously learned from their training data (Trinh and Le, 2018).

Radford et al. (2019) demonstrated that with their language model (GPT-2), a 1.5B parameter Transformer that follows the details of the OpenAI GPT model (Radford et al., 2018), they were able to tackle the WSC273 with 70.7% accuracy. To that end, they scraped web pages that have been curated/filtered by humans (e.g., on Reddit). The resulting dataset, called WebText, contained the text subset of these 45 million links, which led to the development of a dataset of 8 million documents for a total of 40 GB of text. Their approach demonstrates that language models can perform downstream tasks in a zero-shot setting, in different domains and datasets, without the need for direct supervision.

Prakash et al. (2019) enhance previously used language models by augmenting them with knowledge hunting to tackle the WSC273 and the WS283 datasets. Knowledge hunting (Sharma et al., 2015) refers to using sentences with a simple structure that contains evidence for coreference resolution of existing Winograd schemas. Given that, sometimes the needed knowledge to resolve the definite pronoun in some schemas is embedded in the pre-trained language models, by predicting phrases that occur most of the time than other ones, like in Trinh and Le (2018), they combine the knowledge hunting and neural language models to tackle the WSC. To combine the intermediate results of the two methods and predict the best pronoun target's confidence score, they use a Probabilistic Soft Logic (PSL) module (Kimmig et al., 2012). The knowledge extraction module extracts texts that are similar to the examined schema. For instance, from the example in Table (2.1), they extract the verb phrases which are connected with the discourse connective "* refuse * because * fear *". Next, like in Sharma et al. (2015), they search the web to extract text snippets from search engines. To find the most similar sentences, they use Parikh et al. (2016) natural language inference model. In this regard, the sentences contain similar verb phrases and discourse connectives. Next, they compute each entity's semantic roles in the examined schema and similar sentences to align the definite pronoun with the best candidate. Finally, via the PSL framework, they combine the two modules, the knowledge hunting and the language model,

to generate the confidence scores for each of the pronoun targets. In their experiments, they compare two pre-trained language models, the Trinh and Le (2018), and the BERT (Bidirectional Encoder Representations from Transformers) LM (Devlin et al., 2019). The best results were obtained by combining the knowledge hunting with the BERT LM, where they achieved 71.06% accuracy on WSC273 and 70.17% accuracy on the WSC285 dataset.

Klein and Nabi (2019) used a simple re-implementation of BERT (Devlin et al., 2019) to tackle the WSC. According to the authors, although BERT language models can learn context-aware word-embeddings, tackling the challenge and other commonsense reasoning tasks is not trivial. They show that BERT attention maps can be used to resolve coreference resolution problems. During training, BERT learns two prediction tasks: i) to predict masked tokens, given the context, and ii) given a sentence to predict the next one or show if two sentences are consecutive. Their proposed approach takes as input the BERT attention-tensor to output a score, one for each pronoun target, which indicates the strength of association. According to their results, their system tackles the PDP with 68.3% accuracy and the WSC273 dataset with 60.3% accuracy.

Kocijan et al. (2019b) tackled the WSC by fine-tuning large pre-trained language models, such as BERT. Experiments showed that by fine-tuning on MaskedWiki and DPR, their model achieved 72.5% accuracy on WSC273 and 74.7% accuracy on the WNLI dataset. In their work, they focus on using the BERT language model on masked token prediction. They use this method on sentences with a similar structure to the Winograd schemas to figure out which pronoun target is the best replacement for the masked definite pronoun. Specifically, in each examined sentence, they mask the definite pronoun, where the language model has to predict the best pronoun target in place of the masked pronoun.

Raffel et al. (2019) introduced a framework (Text-to-Text Transfer Transformer) to convert text-based language problems into text-to-text problems to apply the same model to different tasks. In this regard, they were able to tackle the WNLI dataset with an average score of 94.5% (T5-11B with 11 billion parameters), which was achieved due to more extensive pre-training. Via Common Crawl, which was used to scrape text from the web, they introduced a new dataset called "Colossal Clean Crawled Corpus" (750 GB), which only retained the lines that ended in a terminal punctuation mark. For fine-tuning purposes, they treat all the GLUE and SuperGLUE benchmarks as single tasks by concatenating all the constituent data sets. Regarding the WNLI dataset, to convert the WSC examples into text-to-text examples, in each schema, they mask the definite pronoun and ask the model to predict the correct pronoun target.

Kocijan et al. (2019a) developed WICIKREM to address the lack of large training sets, which could be used with language models. Experiments they performed show that fine-

tuning the BERT language model with WIKICREM consistently improves the model. Further experiments show that models trained on WICIKREM show increased performance on gender diagnostic datasets, like WGEN. BERT WIKICREM_ALL, which is additionally trained on WIKICREM and obtained by fine-tuning BERT on all the available data from the target datasets at once, achieves 84.8% accuracy on DPR, 74.7% on WNLI, and 86.7% on PDP. BERT WIKICREM_DPR, which is trained on WIKICREM and fine-tuned on DPR (10% of the DPR train set are used as the validation set), achieves 82.1% on WGEN and 71.8% on WSC273.

Ye et al. (2019) proposed a pre-training approach for the BERT language model to incorporate knowledge from sources, like ConceptNet (Speer and Havasi, 2012). They automatically create a multi-choice dataset (KG) with their proposed "align, mask, and select" (AMS) method to construct sentences with labeled concepts. Next, they replace BERT's original pre-trained tasks used: i) to predict masked tokens, and ii) to predict the next one with the KG task when given a sentence. To build the dataset (16,324,846 QA samples), they: 1) filter triples from ConceptNet, which are based on relations such as "RelatedTo" and "IsA" (resulted in 606,564 triples); 2) align each ConceptNet triple (concept1, relation, concept2) with the English Wikipedia to extract the sentences containing the two triple concepts; 3) mask one of the concepts in one sentence with a special token (to be treated as the question); 4) set the selected concept as the correct answer to the question; 5) use randomly chosen words from ConceptNet as distractors. Pre-training models using their proposed approach followed by fine-tuning in the same way as in Kocijan et al. (2019b) achieved 75.5% accuracy on WSC273 and 83.6% accuracy on the WNLI dataset.

He et al. (2019) proposed a hybrid neural network approach (HNN), which combines Kocijan et al. (2019b) masked language model with Wang et al. (2019b) deep semantic similarity model, both sharing a BERT-based contextual encoder. According to He et al. (2019), the models on which the hybrid approach is based use different methods when predicting outputs, and thus they can capture different views of the data. In this regard, the hybrid approach measures both the semantic wholeness of sentences, after replacing the definite pronoun with the pronoun targets, and the semantic relatedness of the definite pronoun and the two pronoun targets. Experiments show that the hybrid model achieves 75.1% accuracy on WSC285, 89% accuracy on WNLI, and 90% accuracy on the PDP dataset.

Ruan et al. (2019) use pre-trained language models with fine-tuning to tackle the WSC. According to the authors, this is critical for achieving the necessary performance, possibly showing that with larger fine-tuned datasets, one can achieve better performances. In this regard, they show that higher performance can be achieved by jointly modeling sentence structures. In their experiments, they approached the WSC as a "Next Sentence Prediction"

problem. After replacing the definite pronoun with each of the pronoun targets, they split the resulting sentence into two parts. Using BERT, they rate these two sentences using pre-trained next-sentence-prediction —to see if the second part follows the first part. By fine-tuning the pre-trained BERT on the DPR dataset (1882 sentences), their best model, BERT-large (348 million parameters), tackled the WSC273 dataset with 71.1% accuracy.

Liu et al. (2019) introduced RoBERTa, an optimized BERT pre-training approach, to measure the impact of various hyper-parameters and training data sizes. They measure RoBERTa's performance on various tasks such as GLUE, RACE and SQuAD. RoBERTa is trained on more extensive data (160GB of uncompressed text), that is, longer sentences in a more extended period, and does not include the BERT's next-sentence-prediction feature. According to the authors, the results show that the pre-training of the masked language model under the options above is competitive with all other published methods. Regarding the WNLI dataset, RoBERTa, which is fine-tuned using the margin ranking loss from Kocijan et al. (2019b), achieves an average score of 91.3%.

Sakaguchi et al. (2020) performed experiments to test if recent results of neural language-model approaches (e.g., BERT) were based on spurious biases that exist in various training datasets. In this regard, they have developed the WinoGrande dataset. Recall that along with this dataset, they have developed WinoGrande_debiased, a dataset consisting of debiased schemas. Based on the fine-tuned RoBERTa language model (Liu et al., 2019), they gained contextualized embeddings for each instance, and used them to discard the top instances that were correctly resolved by more than 75% of classifiers trained on the embeddings. Results show that when training on the WinoGrande dataset, RoBERTa, achieves 79.1% accuracy on WinoGrande, 85% on KNOWREF, 85.6% on WNLI, 87.5% on PDP, 90.1% on WSC273, and 93.1% accuracy on the DPR dataset. Although this shows the WinoGrande dataset's strength when used as a resource for transfer learning, according to the authors, it also raises the concern that they are likely to overestimate the true capabilities of their systems. According to Lin et al. (2020), despite careful controls, the WinoGrande dataset might contain incidental biases that these sophisticated models can exploit.

Lin et al. (2020) tackled the WinoGrande dataset with an average score of 77%, via a T5 sequence-to-sequence model (Raffel et al., 2019). Given that encoder-decoder models can tackle text generation tasks, for fine-tuning purposes, during the training phase, and for each WinoGrande example, two further examples are produced. Each example contains the hypothesis and the premise, referring to either the first or the second candidate. The correct statement is labeled with the entailment label while the other with the contradiction label. Finally, at inference (test) time, each example is decomposed and fed to T5 to predict if the target token refers to an "entailment" or a "contradiction" statement.

Brown et al. (2020) used the GPT-3 model, a language model with over 175 billion parameters, to tackle various NLP challenges under three kinds of training, namely, zero-shot, one-shot, and few-shot learning. Under zero-shot, GPT-3 was given only the challenge description with zero task examples, and it tackled the WSC273 with an average score of 88.3% and the WinoGrande dataset with 70.2%. Under one-shot, along with the task description, the model was given only one example of the challenge, and it tackled the WSC273 with an average score of 89.7% and the WinoGrande dataset with 73.2%. With the few-shot learning, where the model sees a few examples of each challenge, it achieved 88.6% on the WSC273 and 77.7% on the WinoGrande dataset.

3

Tackling the WSC with Logical Inferences

3.1 Introduction

One of the most important challenges in AI is understanding how to create systems that acquire and manipulate commonsense knowledge (Valiant, 2006). The goal is to build machines that autonomously acquire relevant background knowledge and use it afterward to solve different kinds of problems (Michael, 2009). With the creation of cognitive systems, humanity aims to replace or substitute basic human abilities so that humans can relate and interact with them.

This chapter describes how we developed a new method that tackles the Winograd Schema Challenge (WSC) based on commonsense reasoning. According to Levesque (2014), the WSC should serve as the means to understand human behavior towards developing machines with commonsense reasoning. As he said, probably anything that correctly answers a series of these questions is thinking in the full-bodied sense we usually reserve for people. Put simply, each WSC instance should tell us something about human behavior that needs to be explained (Levesque, 2014). In this regard, here, we examine the task of resolving cases of definite pronouns, for which traditional linguistic constraints on coreference as well as commonly-used resolution heuristics are not proper, or the procedure they follow is very similar to a statistical approach, without invoking commonsense reasoning similar to what humans do.

At the time of writing, more than 95% of the developed approaches try to solve the WSC based on pattern-of-word or neural network solutions (see Chapter 2). Subsets of the original dataset (WSC_) can be tackled with an average score of 66%, where the lowest

score is 42%, and the highest 90%. This high performance is partly because most developed approaches are trained for specific datasets and objectives. Although this leads to models that are effective at finding task-specific correlations, at the same time, these kinds of systems lack explainable and straightforward commonsense reasoning, meaning they are opaque and brittle (Marcus, 2018; Marcus and Davis, 2019; Mitchell, 2019). Additionally, heuristics or methods non-crucial to the WSC purpose, namely the choice of word embeddings or the use of language models, can easily affect the results (Kocijan et al., 2020). For instance, language models that predict probabilities of the next or the previous word in a sentence may have their limits (Brown et al., 2020), meaning that more text does not always yield better results. In this line of research, recent experiments have shown that state-of-the-art language models struggle when trying to solve challenges that directly relate to abductive reasoning, meaning they lack reasoning abilities that are trivial for humans (Bhagavatula et al., 2019).

The motive behind the WSC is the simulation of human-like reasoning in machines to test machine's ability to answer commonsense questions regarding sentence comprehension. In this sense, a machine presents that type of behavior when it reaches a conclusion from a situation similar to how humans do it, which is not trivial whatsoever. According to Mitchell (2019), there is this paradox in AI initially reported by Marvin Minsky, based on which "easy things are hard to develop". Building intelligent machines with commonsense reasoning is a task that has been bothering the AI community for a long time (Mitchell, 2019). Words like "intelligent" are characterized as suitcase words, which means something broader and more challenging to acquire, similar to thinking and cognition (Minsky, 2007).

We believe that the real meaning in reading comprehension is in reading between the lines (Michael, 2009), that is, intelligently building machines that understand implied messages, connect concepts and relations (Harwell, 2018), and, similar to what the WSC is all about, to be able to resolve ambiguities in sentences. For learning to be meaningful to solve problems like the WSC, machines need to deal with missing information and employ new learning techniques to act as expert assistants that collaborate with humans.

Past experimental work, based on unaxiomatized knowledge acquisition from text, was promising for extracting knowledge from text automatically (Michael and Valiant, 2008). It aimed to make it possible for systems to acquire knowledge on a large scale by learning and then use it robustly for reasoning. It is widely believed that logical inferences are necessary in order to build natural language representations, as well as to reason about information encoded in representations. In this regard, here, we present a technique that focuses on commonsense knowledge, which can be retrieved and learned via a supervised learning approach, called auto-didactic (Michael, 2010). It includes, among others, the acquisition and the extraction of general inference rules that could help us solve different RTE problems,

like the WSC. In this sense, progress on the WSC could be made possible by allowing machines to learn and reason as humans do.

This work differs from previous work mainly in three aspects: i) It tackles the WSC through commonsense reasoning. ii) It uses only one source for knowledge acquisition (the English Wikipedia). iii) To tackle Winograd schemas, it utilizes logical inferences drawn from the Websense engine (Isaak, 2011; Michael, 2013). Below, we start by presenting the Websense engine's structure, followed by the proposed methodology on the WSC. The sections below explain each of these tasks, along with the tools and techniques we have developed.

3.2 The Websense Engine

According to Valiant (2006), progress in the AI field could be achieved, by having machines acquire commonsense knowledge, through learning, from natural text. This could be achieved by emphasizing on the acquisition of knowledge in terms of computer-readable rules, the efficiency of the acquisition task, and the robustness of the acquired knowledge. The autonomous understanding of a text, called Machine Reading (MR), could be used as a new full-fledged approach of AI to extract knowledge from text (Etzioni et al., 2006). MR aims to combine multiple textual entailment steps (Dagan et al., 2005), based on a given text, to form a coherent set of premises (inferences). Further experiments, in a subsequent work (Michael and Valiant, 2008) to that of Valiant (2006), provided evidence of the feasibility of that approach on a massive scale.

According to Michael (2009); Mitchell (2005), the Web offers a plentiful source of human knowledge encoded in text, from which machines can obtain its factual content to use as a structured knowledge base. The Websense engine has been developed in the sense that even an individual piece of knowledge might not be stated explicitly on a single Web page but be implicitly encoded across the Web through several sectors like expert knowledge, cultural biases, misconceptions, fictional statements, and deliberate lies (Isaak, 2011; Michael, 2013).

Websense is a novel engine that brings together the goal of endowing machines with commonsense knowledge and the goal of understanding text from the Web through machine reading (MR). Specifically, Websense can respond to user queries provided in natural language text, with inferences that are implied by the given queries according to the collective human knowledge found across the Web (see Table 3.1). Put simply, the Websense engine can output logical inferences, comparable to what humans do, so that they can relate and interact with it as if it were a piece of advice coming from a knowledgeable assistant. According to Michael (2013), progress on the WSC could possibly be made by allowing these types of

systems to draw inferences similar to those drawn by humans. For instance, if Websense is trained for the term "spyware", given the text "The virus infected something", the engine will respond with the logical inference that "The virus infected a computer" —had we trained it for "covid-19", it would probably change its response to "The virus infected a human being". On another note, if we train it for articles, given the text "members sharing something", the engine will respond with logical inferences resembling "members sharing articles and files".

Query: `Virus infected something.`
Scene-Constructor transformation: $\texttt{virus}(t_1) \wedge \texttt{something}(t_4) \wedge \texttt{infected}(t_1, t_4)$
Logical Inferences: $\texttt{computer}(t_4)$
Response: `The Virus infected a computer.`

Table 3.1 User interaction with the Websense engine, which was trained under the "spyware" keyword.

Given that recent advances in the field mostly rely on engineering feats —statistical approaches (e.g., n-grams, neural networks) that lack a reasonable process for acquiring robust knowledge to use for reasoning—, Websense is based on an enhanced version of good old-fashioned symbolic AI (GOFAI). Specifically, instead of externally providing the necessary knowledge in the form of written rules, Websense's logic-based knowledge is autonomously extracted by reading the Web. In this sense, the engine provides both the means and the opportunity to exploit this vast knowledge source for the benefit and progress of the AI field (Michael, 2013).

According to Marcus and Davis (2019), we humans know all kinds of things, and every sentence we encounter requires us to make inferences about how those things interrelate with what we read. The goal was to use the Websense engine's logical inference mechanism to test it on the WSC. The Websense engine acts as humans do while reading, writing, thinking, or answering questions, which is very helpful. It resembles an expert assistant able to provide us with useful inference rules that might help us in the pronoun resolution task.

3.2.1 The Websense Engine's Architecture

In its current version, the Websense engine works in two distinct modes, called the learning mode and the interactive mode. The engine is written using advanced techniques to accelerate the process. In the learning mode, without human supervision, it can learn anything from the Web for a single subject (e.g., *spyware*). In the interactive mode, it can accept any user query for input and return the commonsense conclusions produced through another component, called the Reasoner (see logical inferences in Table 3.1). In short, for each examined subject, Websense builds a relational knowledge base, with each rule providing a *websense definition*

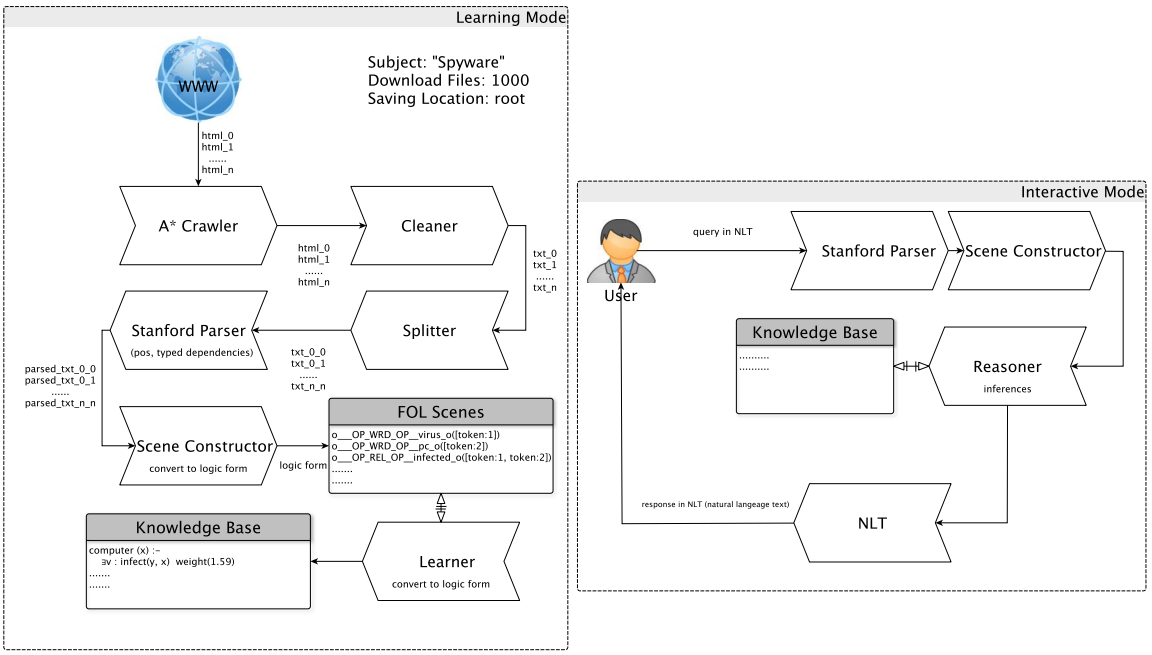


Figure 3.1 Websense's architecture. In the first mode, it crawls the WWW to learn and build its knowledge base (relational rules). In the second mode, it accepts any user query in natural language text (NLT) and generates inferences implied by the given query.

of the predicate at the head of the rule, accompanied by variables (facts) that appear in a line of the body of each rule. At the same time, the engine offers guarantees on the soundness of its inferences (Michael, 2013).

Web Crawling

The Crawler component performs the crawling procedure, which locates the best results in HTML format based on the A* algorithm (see the top-left part of Figure 3.1). In its current version, the Crawler is externally provided with specific keywords (one at a time) and downloads only pages containing those keywords. Based on the A* algorithm, the HTML pages are reordered in descending order, meaning that pages with highly relevant content are visited first. The goal here is to locate and download valuable content by allowing the engine to determine the number of links of each visited page internally.

The Crawler browses all the web pages linked from the visited web pages and adds them with the number of keyword occurrences in a queue. Next, after the queue is reordered in descending order, the Crawler continues by visiting the first web page of the queue — after browsing a specific URL, the page is removed from the queue. The whole procedure continues until a specified number of web pages is reached. To avoid visiting web pages

with the same content, a cyclic redundancy check (crc-32) technique is used. In the end, the crawling procedure allows Websense to build its knowledge around a specific subject, enabling it, at the same time, to draw relevant inferences —e.g., for the subject "spyware" the engine might conclude that "hackers act like thieves".

Cleaning and Parsing

After the specified number of web pages is reached, the Crawler downloads the requested HTML files in a specified location. Next, a component called the Cleaner performs the HTML-cleaning procedure. The Cleaner parses each downloaded file and removes all HTML tags. Afterward, clean pages are split into sentences for speed improvement through a component called the Splitter. To that end, anything except English sentences is removed. In its current version, the splitting procedure does not consider that sentences might belong to the same paragraph.

Afterward, each sentence is parsed through the Stanford Parser (De Marneffe et al., 2006) to convert its meaning into certain existentially quantified conjunctions over predicates (Michael, 2013). Stanford parser can show the typed dependencies included between the words in each sentence (see table 3.3). Specifically, Stanford typed dependencies are designed to provide a simple description of the grammatical relationships in a sentence that can easily be understood and effectively used by people without linguistic expertise to extract textual relations (De Marneffe and Manning, 2008). For instance, for the sentence, *The virus infected the pc*, Stanford Parser returns the following typed-dependencies *det (virus-2, The-1), nsubj (infected-3, virus-2), det (pc-5, the-4), dobj (infected-3, pc-5)*, which map the words with their grammatical relations onto a directed graph representation —the words are nodes, and the grammatical relations are edge labels (De Marneffe and Manning, 2008). Specifically, the returned typed dependencies of our example sentence have the following meaning:

- det (virus-2, The-1): This dependency shows the relation between the head of a noun phrase and its determiner.
- nsubj (infected-3, virus-2): This is the nominal subject dependency, showing a noun phrase that is the syntactic subject of a clause. Most of the time, the governor of a nominal subject (nsubj) relation is a verb.
- dobj (infected-3, pc-5): This refers to the direct object (noun phrase) of a verb phrase, which is the predicate of that clause.

Scene Building

Following, each sentence's typed dependencies are being parsed through another component, called the Scene-Constructor, which produces first-order semantic scenes (see table 3.1). In short, from each parsed output, Websense proceeds to extract facts about certain entities that each examined sentence contains.

These kinds of scenes relate to first-order logic (Michael and Valiant, 2008), which, according to Wooldridge (2020), is the *lingua franca* of mathematics and reasoning. According to Michael and Valiant (2008), these kinds of scenes are useful in acquiring knowledge from text. For instance, the nsubj (x-3, y-1) and dobj (x-3, z-3) typed dependencies can be combined to create semantic scenes related to the nominal-subject and direct-object of the examined sentence. Take for example the following typed-dependencies, nsubj (infect-3, virus-1) and dobj (infect-3, system-3), based on which the Scene-Constructor outputs the following scenes: $\texttt{virus}(t_1) \wedge \texttt{something}(t_4) \wedge \texttt{infected}(t_1, t_4)$. In its current version, the Scene-Constructor can build thirteen such relations.

Learning Mode

Next, the Scene-Constructor's scenes are given as input to another component, called the Learner (Michael and Valiant, 2008), to produce the required knowledge around the subjects we are interested in. Wholes of such scenes correspond to the inputs available to Websense's learning mode, where they end up being the formulas in the bodies of the rules found in the Learner's knowledge base (see the bottom-left part of Figure 3.1).

We start by marking the Learner's lexicon with the subjects we are interested in. Then, the Learner proceeds with the parsing of the Scene-Constructor's scenes to build its knowledge. The produced knowledge file contains prolog-like rules and facts ("Head : Body"), where "a head is true if its body is true". The body of each head —subject identified in the lexicon— contains facts or rules in Disjunctive Normal Form (DNF). On a second front, DNF formulas lead to the development of knowledge bases whose rules are semantically closer to the type of knowledge examined in many works in the area of knowledge representation and reasoning (Michael, 2013). Additionally, anything —rule or fact— that has a weight>=1 is considered significant. The basic idea is that each formula in the body is associated with a number, where the larger the number, the more preferred the formula is. Additionally, the employed learning algorithm provides a priori guarantees on the appropriateness of the responses (Michael, 2009). Put simply, the degree of a belief is justified empirically on how often it is found to be true (Valiant, 2006). For instance, the knowledge file might contain the next rule, telling us that a computer can be infected by a virus: $\texttt{computer(v)}\text{:-}\ \exists \texttt{v : infect(x, v)}$

$\wedge$ `virus(x) weight(1.59).`

Reasoning Mode

In the final step, the Reasoner component accepts user queries in natural language to generate logical inferences implied by the given queries, which are important facts identified by the given knowledge file (see the bottom-right part of the Interactive mode in Figure 3.1).

Specifically, the Reasoner component draws inferences by applying a relational knowledge base on a set of input semantic scenes (predicates). In this regard, each rule in Websense's knowledge base is applied only once on the set of input semantic scenes determined by the user query. The rule heads that are found to be true comprise the engine's inferences for that input (Michael, 2013). For instance, if Websense is given the query "The virus infected something", via Reasoner, it will conclude and respond in natural language that the virus infected a computer (see logical inferences and response in Table 3.1). Therefore, the given scene, $\texttt{virus}(t_1) \wedge \texttt{something}(t_4) \wedge \texttt{infected}(t_1, t_4)$, would have triggered the rule to infer computer (t_4).

According to Marcus and Davis (2019), when this kind of formal logic works well, meaning that it represents knowledge with sufficient clarity along with the ability to reason, it might be a substantial shortcut toward endowing machines with commonsense reasoning abilities.

Sentence-Generating Mode

Finally, to produce human-friendly inferences, Websense makes use of the Natural Language Text (NLT) component (see the bottom-left part of the interactive mode in Figure 3.1). This is a component that develops simple English sentences by combining the Reasoner's inferred scenes with the scenes from the user query. For instance, for the user query scenes $\texttt{virus}(t_1) \wedge \texttt{something}(t_4) \wedge \texttt{infected}(t_1, t_4)$ and the Reasoner's logical inference $\texttt{computer}(t_4)$, the NLT component returns the sentence, "The virus infected computer" (see Table 3.1).

3.3 Wikisense

In its current version, Websense can accept simple English sentences to generate implied inferences according to the collective human knowledge found across the WWW. In short, the engine mimics humans when reading newspapers or talking to each other, however in

a simple form. For instance, when we read the sentence "The cat caught a mouse", we might draw inferences about a clever, fast, and hungry cat. This work aims to use the Websense engine's components to attempt to answer Winograd instances as humans do. For that purpose, we modified and used an upgraded version of the Websense Engine (called Wikisense) that acquires richer knowledge faster by applying principled learning techniques (see Figure 3.2).

3.3.1 Learning-Framework for the WSC

Let us take the first half from Table 3.2, which will be referred to as the *catch-clever* example. Wikisense's purpose is to take the input sentence, the question, and the two possible pronoun targets, to return the correct pronoun target. For instance, if we give Wikisense the *catch-clever* sentence and ask, "Who is clever?", we want it to respond with the correct pronoun target, which is the *cat*.

First half	Sentence	The cat caught the mouse because it was clever.
	Question:	Who is clever?
	Pronoun Targets	the cat, the mouse
	Correct Answer:	the cat
Second half	Sentence	The cat caught the mouse because it was careless.
	Question:	Who is careless?
	Pronoun Targets	the cat, the mouse
	Correct Answer	the mouse

Table 3.2 A Winograd schema: The *catch* example.

That being said, initial experiments that involved testing the engine on Winograd halves could not help us resolve definite pronouns. For instance, for the catch-clever example (see first half in Table 3.2), there were not enough relations between the sentence words that could help us resolve the definite pronoun. No matter the amount of effort we applied, when we let the engine find anything for the words *catch, mouse, cat, clever* (one at a time), the engine could not generate sufficient results to reach conclusions —e.g., The mouse is being caught by something else, The cat is catching something else. In the end, it was shown that the Learner's knowledge was too generic to help us conclude the correct pronoun target (the cat).

To strengthen Wikisense's knowledge and reasoning abilities, we focused on human commonsense reasoning ability, although this was not an easy task as "easy things for humans are hard for machines" (Mitchell, 2019). When someone has to resolve pronouns, they mainly focus on sentence verbs, nouns, and adjectives, and subsequently, through these relations, try to find the correct answer (Adger, 2019; Michael and Valiant, 2008). According to Marcus

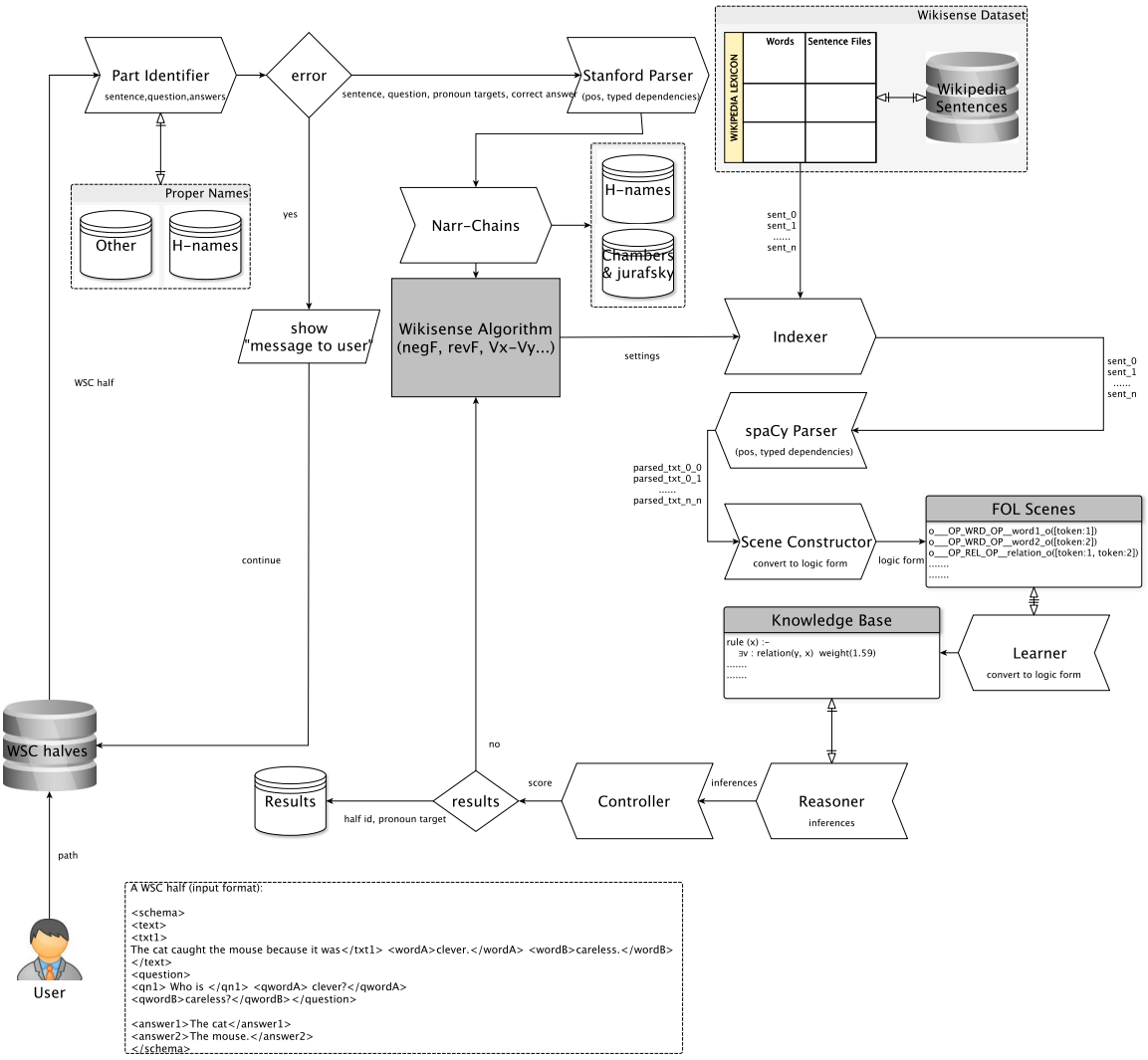


Figure 3.2 Wikisense's architecture. For any given Winograd half and based on various settings, the engine searches the English Wikipedia to learn and build its knowledge base (relational rules) in order to tackle it.

and Davis (2019), when we read or hear something, our brain parses and deconstructs the sentence into its constituent nouns and verbs and what they mean. Furthermore, hierarchical structure, and not linear order, is necessary to understand how pronouns refer to things, as languages mostly use the structure as one aspect of what limits the links between pronouns and names (Adger, 2019).

Below, we continue to describe the design and modifications of the individual components of Wikisense and discuss certain choices made.

User-Controllable Interface

The engine interface in Figure 3.2 illustrates a simple example given by a user, the parsing and filtering of that example, the engines' inferences drawn from it, the engine's response, and the storing of the results. Having in mind the original WSC dataset (WSC_), Wikisense, entirely programmed from scratch in the Python[1] programming language, was designed to allow access to a broad range of schemas that strictly follow the challenge guides and rules —compared to Wikisense, the Websense engine was programmed in vb.net. In this case, in its current version, Wikisense accepts Winograd schemas, where each half consists of a sentence, a question, two pronoun targets, and the correct answer. In the first step, the user must define an XML or a text file with the Winograd schemas at hand (this is depicted in the bottom-left part of Figure 3.2). Then, the user starts the process, where for each half, Wikisense either generates the correct/wrong answer or an error flag —in case it is unable to identify the parts of a schema.

Part Identifier

One side problem of the WSC is identifying the schema parts correctly. Recall that most scholars use well-structured schema-like examples instead of having their systems struggle to relate the question, the sentence, and the two pronoun targets (see Chapter 2).

Our current design uses the Part Identifier component to identify each half's parts by combining predicates (scenes) of the sentence and question with the two pronoun targets (this is depicted in the top-left part of Figure 3.2). To that end, this component uses tools such as the Stanford Parser, proper-noun, and historical-noun lists to classify each half as a noun or a proper-noun problem. Additionally, it utilizes various tools (e.g., lemmatizers, semantic analyzers) to identify pronoun targets that might appear with different names or are not explicitly mentioned in the sentence. For instance, in the schema (Lenat, 2008) *Sentence: Joe saw his brother skiing on TV last night but the fool didn't* ***recognize him/have his coat***

[1]https://www.python.org

on. *Question: Who is the fool? Pronoun targets: Joe, Joe's brother*, we use those tools to identify the relations where the pronoun target participates in (Joe's brother). Finally, if the Part Identifier cannot identify the parts of a Winograd instance, it notifies us accordingly via a notification message and continues to the next one.

Wikipedia Indexing

Initially, we changed the engine's Crawler (this is depicted in the top-left part of Figure 3.2). Because of Crawler's idleness, we created an offline version to work with the English Wikipedia (called Indexer), which is of higher quality both in language and information. The Web-knowledge acquisition through the A* based Crawler, along with a cyclic redundancy check technique, was time-consuming. For instance, for a nine-word sentence, the search process could last from 30 minutes to 24 hours, depending on the size of each HTML page. Moreover, given that Wikipedia is developed on the WWW, is open source and free for anyone to write, Wikipedia is indirectly related to the Websense Engine development (an individual piece of knowledge might not be stated explicitly in a single Web page, but be implicitly encoded across the Web).

In this regard, we downloaded the English Wikipedia, which resulted in a clean-text of 18GB's from Common-Crawl[2]. To drive the whole learning procedure to a specific learning domain, excluding generalities and likelihood of confusion and to reduce time complexity, we removed the Cleaner and Splitter components from the whole process. We only used them once to build a static version of the Wikisense dataset (this is depicted in the top-right part of Figure 3.2). For that purpose, the 18GB's of Wikipedia resulted in a *database* of nearly 88 million English sentences, consisting of the lexicon part and the sentence part —The lexicon part contains everything except the stop-words of the sentence part. Specifically, the lexicon part contains files of lemmatized indexed words that lead to sentences consisting of those words. In this regard, the Indexer component always returns sentences that contain the words we are interested in. For example, if our current Indexer's keyword is *mouse*cat/catch*, it means that all sentences that include the words *mouse, cat, and catch* will be further transmitted to the next step.

In our current design, the Indexer is externally guided via user-controllable settings. Specifically, we updated the Indexer to search via a combination of subjects (e.g., noun, verb, adjective) instead of a single one. Furthermore, the Indexer was enhanced with *synonym* and *antonym* capabilities —nouns, verbs, and adjectives— to be able to search and learn more widely. Using synonyms and antonyms as a direct keyword *plugin* could also help retrieve more Wikipedia sentences. For instance, in the case of synonyms, instead of searching only

[2]http://commoncrawl.org/the-data/get-started/

for the word *cat*, we could also search for the words *moggy, feline, mouser, kitty*. Work in the past used different hybrid approaches to combine shallow analysis with synonyms to attempt to solve different RTE problems (Bos and Markert, 2005). Additionally, we equipped Indexer with extra verb capabilities related to each verb's root. For example, instead of searching for *mouse*cat/catch*, Indexer can acquire Wikipedia pages based on a variety of keywords, such as *mouse*cat/catch, caught, catching, catches*. Also, pluralization capabilities were added for nouns, verbs, and adjectives that enhanced the keyword list with more items/words.

Narrative-Chain Replacer

Work in the literature found it challenging to answer schemas with proper names (Budukh, 2013). In this regard, if at least one of the two possible answers is a proper name, the Chambers and Jurafsky (2008) narrative-chains are used to replace them. These are ordered sets of events (verbs) centered around a common protagonist. Therefore, for any given proper name, the Narrative-Chain Replacer component browses a list consisting of thousands of proper names downloaded from the WWW (see top-middle part of Figure 3.2). This does not apply to pronoun targets that refer to historical names, as there is sufficient information available in Wikisense's dataset.

Scene Building

Recall that the Scene-Constructor component accepts as input the typed-dependencies that are produced from the Stanford Parser (see Table 3.3) and generates first-order semantic scenes. For example, if we provide as input the *catch-clever* sentence dependencies, through a pre-running semantic option (e.g., S_DobjNsubj) it generates the scene $\texttt{cat}(t_2) \wedge \texttt{mouse}(t_5) \wedge \texttt{catch}(t_2, t_5)$, which tells us that a cat catches a mouse.

Stanford Parser: `nsubj(caught-3, cat-2) dobj(caught-3, mouse-5)`
SpaCy: `verb: catch catch_subject: cat, catch_object: mouse`

Table 3.3 Stanford and spaCy parser output for the catch-clever sentence.

This relation has been created because of a direct connection between the entities connected through the *dobj* and *nsubj* dependencies, based on which we can create *subject-verb-object* relations. However, more semantic rules like the above had to be built in order to use the Scene-Constructor on the pronoun resolution problem. Given that virtually every sentence we utter is novel (Adger, 2019), the more semantic rules we have, the more cause-and-effect structure of the world we will be able to capture. To that end, new relations have been created by studying the Stanford-Parser's typed-dependencies manual (De Marneffe and

Manning, 2008). Finally, the Scene-Constructor component concluded with multiple useful but time-consuming relations.

Stanford parser is a well-tested parser widely used, but the speed is not one of its advantages. This is why the parsing procedure was updated to be used in combination with a faster one. Given that Wikisense was needed to parse thousands of sentences and not to be restricted by the parser's speed, the engine uses the Stanford parser in combination with a new parser on the NLP field, called spaCy (see table 3.3). SpaCy features a high-performance tokenizer, part-of-speech tagger, named entity recognizer, and syntactic dependency parser, which offers one of the fastest syntactic parsings in the world[3]. In this regard, we updated the Scene-Constructor to work also with spaCy. Through SpaCy, Scene-Constructor finds the sentence subjects (e.g., "nsubj", "nsubjpass", "csubj", "csubjpass", "agent", etc.) and objects (e.g., "dobj", "dative", "attr", etc.) to create semantic scenes.

Because of spaCy's speed and Sanford-Parser's legacy and acceptance, both parsers are in use. First, Stanford Parser parses each WSC sentence to locate the pronoun-target positions in the sentence and correlate the question with the WSC sentence. If Stanford Parser cannot provide useful outputs, then we also use spaCy. Afterward, as it is a faster parser, we use only spaCy to parse Wikipedia's sentences located by the Indexer.

3.3.2 A Simplified Running Example

A big issue concerned Wikisense's controlling procedure until the final step of the pronoun resolution. In this regard, the whole process was enhanced to guide the whole procedure to the pronoun resolution. For better understanding, below, we describe Wikisense's controlling procedure through a simplified example concerning the *catch-clever* half (see Figure 3.2).

At first, the engine loads the examined half, and through the Part Identifier identifies the sentence, the question, the two pronoun targets (answers), and the correct pronoun answer. Next, it parses both the sentence and the question, and in correlation with the two possible answers that have to be located in the sentence, creates the Indexer's search keywords. If at least one of the two possible answers is a proper name, the Narrative-Chain Replacer is called to replace them.

Regarding the keywords, the engine keeps only verbs, nouns, and adjectives from the WSC sentence and the question. Specifically, it splits the keyword procedure into two parts, based on the sentence and the question, and produces the necessary keywords for the Wikipedia search that correlate the two pronoun targets between them and the question (e.g., see table 3.4).

[3]https://spaCy.io

The keyword-generation process starts with the keyword that directly relates the two pronoun targets based on the parsing results (see 1Q in Table 3.4) —e.g., the nominal subject and direct object relation, built by the Scene-constructor component (e.g., `mouse*cat/catch`). In the event that the parsing procedure cannot generate results, an alternative heuristic procedure, which creates the keyword using the positions of the two pronoun targets, is called. If the first keyword (1Q) cannot be created, the current WSC half is abandoned. Similarly, if the two pronoun targets cannot be located in the sentence, the examined half is also abandoned. For instance, the two pronoun targets might be stated in a sentence slightly differently from the given answers, though we maximally eliminate these kinds of problems using the Part Identifier component —for instance[4], in the phrase, *I saw Jim yelling at some guy in a military uniform with a huge red beard. I don't know who he was, but he looked very unhappy*, the two answers given are *Jim*, and *the guy in uniform*.

If the examined half is not abandoned, then the first keyword (1Q) can be directly used to output semantic scenes to the knowledge needed to solve the pronoun resolution (e.g., $\texttt{cat}(t_2) \wedge \texttt{mouse}(t_5) \wedge \texttt{catch}(t_2, t_5)$). All semantic-like parsers extract only some of the semantic relations encoded in a given text, though the rest requires further work. The other keywords are created in the following order (see table 3.4): *(1.)* Between verbs that are included in the first keyword (1Q) and the verbs, adjectives, nouns from the question (e.g., 2Q). *(2.)* Between the two pronoun targets and the verbs, adjectives, nouns from the question (e.g., 3Q).

(1Q) cat*mouse/catch, (2Q) catch/clever, (3Qa) cat/clever, (3Qb)mouse/clever

Table 3.4 Indexer's keyword queries for the catch sentence.

After creating the keywords, Wikisense summons the Indexer component with the new keyword (e.g., 2Q, `catch/clever`) and waits until it generates the requested amount of sentences. The default is one thousand Wikipedia sentences for each examined half, though it can be easily modified via the Wikisense settings. Indexer seeks to retrieve the specified number of sentences through various settings, starting with sentences that contain the requested words.

In case the requested amount cannot be retrieved, it continues to search via other settings such as enhancing the keyword items with their synonym and antonym values or verb-roots. Finally, if the requested amount cannot be retrieved, the procedure advances to the next step with the current retrieved number. If the system cannot retrieve sentences, the current half is abandoned, and the procedure continues with the next half.

[4] see example 95 from https://cs.nyu.edu/~davise/papers/WinogradSchemas/WSCollection.xml

Rule 1:
cat(x)
∃v : catch(x, v) weight(1.03031)
∃v : catch(x, v) ∧ mouse(v) weight(1.000000)
Rule 2:
mouse(x)
∃v : catch(v, x) weight(1.110000)
Rule 3:
clever(x)
∃v : catch(x, v) weight(1.044202)

Table 3.5 The Learner's knowledge for our simplified example. The first rule means that a cat catches a mouse. The second rule that a mouse is being caught by somebody else, and the third rule that the clever catches somebody.

When the requested amount of sentences is found, Wikisense locates sentences obtained to replace those words with the original keywords. This is done to have homogeneity among sentences to further help the commonsense reasoning process of Learner and Reasoner. Then the whole procedure continues with the other components that follow:

- Every sentence is parsed via spaCy, wherein correlation with the Scene-Constructor, the semantic scenes are developed to be given as input to Learner —semantic scenes of the type subject, verb, object that are created through the sentence/question's subjects and objects.
- If essential scenes are included, then a knowledge file is produced, which will be used by Reasoner to resolve the definite pronoun to the correct pronoun target.
- If a knowledge file is not produced, the whole procedure runs again with another searching keyword.

In our *catch-clever* example (see Table 3.5), after adding the keyword's `cat*mouse/catch` semantic-scenes and processing the keyword `catch/clever`, a useful knowledge file is produced. If we carefully observe this knowledge file, we can conclude the following: *Between the cat and the mouse, the clever cat is catching the mouse*. We consider this characteristic of our engine as advantageous because the ability of a system to show its work is one of the most important benefits of symbolic AI (Wooldridge, 2020).

The produced knowledge file (e.g., see Table 3.5) contains rules, where all variables in the head of each rule are assumed to be universally quantified over that rule. In our example, the Learner's knowledge consists of three rules. The first rule has two scenes in its body. The first scene informs us that if an entity_x catches an entity_v then the entity_x has the property

cat. In the same regard, the second scene tells us that if an entity_x catches an entity_v that has the property mouse, then the first entity (entity_x) has the property cat. The second rule, consisting of a single scene, shows that if an entity_v catches an entity_x, then the entity_x has the property mouse —according to the generated weights, this scene is more important than the first scene of the first rule. Finally, the last rule depicts that if an entity_x catches an entity_v, then the entity_x has the property *clever*.

The final step is the definite pronoun resolution, where Wikisense summons the Reasoner component to extract the correct pronoun target from the produced knowledge file. Recall that the Reasoner draws inferences by applying a relational knowledge base on a set of input predicates. In this regard, it was modified to automatically take a query from Wikisense with a preformed scene, via which it indirectly asks if Reasoner can conclude anything else from this query. Specifically, it asks if Reasoner knows more about the first keyword's semantic scene (1Q).

For instance, in our *catch-clever* example, we are getting an inference that the *cat* is also *clever*, meaning that the correct pronoun target is the cat. Given the query $\texttt{cat}(v_1) \wedge \texttt{mouse}(v_2) \wedge \texttt{catch}(v_1, v_2)$ Reasoner concludes that: $\texttt{clever}(v_1)$.

Generally speaking, Wikisense overviews the learning procedure's generated amount of knowledge. If the problem is solved, it returns the correct pronoun target. Otherwise, it keeps into the knowledge-base only the first sentence's part semantic scenes (e.g., `mouse*cat/catch`) and proceeds to the next Indexer call. If a keyword is not available, the engine returns a message that the pronoun cannot be resolved. Below, we explain Wikisense's knowledge acquisition process in detail, which was discovered after several experiments.

3.3.3 Knowledge Acquisition Algorithm

Wiksense's knowledge acquisition algorithm is depicted as a black box in the top-center part of Figure 3.2. Recall that, in every step, Wikisense acquires sentences via Indexer from the English Wikipedia, builds the necessary scenes, and feeds the Learner. At the end of each step, it asks Reasoner to return the correct pronoun target through the question. If no conclusions can be provided, it proceeds to the next step.

To discover the best knowledge acquisition algorithm (e.g., the best keyword sequence and keyword settings), we examined the DPR dataset from Rahman and Ng (2012). We labeled the examined WSC halves *supporting* and not training because we used them only to *guide* the Wikisense's learning procedure. Given that the DPR dataset does not include questions but only definite pronouns, through another built component (called Questionnaire), we automatically added the necessary questions (to match the original WSC_ dataset). Recall that each WSC half consists of a sentence, a question, a definite pronoun, and the two

pronoun targets. In this regard, for each examined half, the Questionnaire component parses the sentence to locate the position of the definite pronoun. Next, it removes all the sentence words from the start to the position of the definite pronoun. Finally, it replaces the definite pronoun with English question words, accordingly (e.g., it to What, they to Who). For instance, for our catch-clever example sentence, "The cat caught the mouse because it was clever", the Questionnaire component would generate the "What was clever?" question.

Next, we updated the Reasoner component to return the *value-confidence* for each half's processed keyword. This is an integer value that shows the Reasoner confidence for the generated results. For instance, we need to know how many rules in the knowledge file can specify that the subject of the word *catch* is also *clever* (see table 3.5). This is done to build a principle mechanism to combine multiple inferences in a single Wikisense inference. The magnitude of the values of each body rule in the knowledge file does not matter in this case: all that is important is for the magnitude to be above a certain threshold (e.g., $>= 1.0$). Ultimately, these kinds of rules capture discrete chunks of knowledge, based on which we know when we can safely conclude from premises (Wooldridge, 2020). In this regard, we were calling Wikisense, forcing it to return a feature vector that we were going to examine in a later step. The vector consists of fourteen values that show verb relations, verb-synonym relations, verb-antonym relations, and noun relations. Below, we explain how Wikisense built the feature vector for a total of 1697 supporting WSC instances.

1. Through the two parsers, Wikisense determines if negation is addressed to any of the two possible pronoun targets (labeled as negF). As stated in the literature, negation is important because it changes the direction of the rules (Peng et al., 2015; Rahman and Ng, 2012).

2. It determines whether the two nouns appear in the sentence in reverse order, contrary to how they appear in each half's answers (labeled as revF).

3. It runs the first keyword that connects the sentence with the question (e.g., `catch/clever`) and stores the *value-confidence* without synonyms or antonyms enabled (labeled as Vx-Vy). *Vx* is the confidence for the first pronoun target, and *Vy* is the confidence for the second pronoun target (e.g., *Vx* shows that the subject of the verb catch is also clever).

4. Wikisense runs the same keyword as previously (e.g., `catch/clever`) and stores the *value-confidence* directly through synonyms (labeled as Sx-Sy). For instance, if catchSyn1, catchSyn2 are the synonyms of the word "catch" and cleverSyn1, cleverSyn2 are the synonyms of the word "clever", Indexer runs for all the correlations between the

two word synonyms (e.g., `catch/cleverSyn1`, `catch/cleverSyn2`, etc.). For each correlation, the engine determines the *value-confidence* of the first noun (*Sx*) and the second noun (*Sy*) and adds them to a score-counter (Sxcounter for noun1, Sycounter for noun2). In the end, it exports the counters to the feature vector (e.g., *Sy* shows that the object of the verb is also clever).

5. The same process is repeated for the antonyms of the same keyword, and the resulting *value-confidence* is stored (labeled as Ax-Ay).

6. The procedure continues with the noun keywords (e.g., `cat/clever`, `mouse/clever`), where the *value-confidence* without synonyms and antonyms enabled is stored (labeled as Nx-Ny). On the one hand, *Nx* shows that the first noun (e.g., *cat*) is clever, and on the other hand, *Ny* shows that the second noun (e.g., *mouse*) is clever.

7. Through a heuristic approach, Wikisense determines the times that the first noun (e.g., *cat*) appears right before (labeled as NBx) or right after (labeled as NBy) the question word part (e.g., *clever*). In this regard, it applies semantic scenes in the knowledge file without the usage of spaCy (in case spaCy might not generate useful results). Also, it determines the times that the first keyword's verb word (e.g., *catch*) appears before (labeled as VBx) or after (labeled as VBy) the question's word (e.g., *clever*).

To summarize, for each half, we are store the following values: *NegF-RevF-Vx-Vy-Sx-Sy-Ax-Ay-Nx-Ny-NBx-Nby-VBx-VBy*. For instance, the *catch-clever* half's (feature-vector) is *False-False-0-0-1-0-0-0-0-1-0-9-2*. The feature vector procedure ran for almost a week, where initial results showed that the order of steps that had to be followed consisted of the same sequence as above (see Algorithm 1).

In each examined half, Wikisense tries to resolve the definite pronoun through each distinct step (called full-iteration-cycle). If it has enough knowledge, each step provides the correct pronoun target and proceeds to the next half. In case no other step is available, it generates a message saying that the current pronoun cannot be resolved and proceeds to the next half.

Our initial experiment results showed that more emphasis is placed on verb-like keyword relations (e.g., `catch/clever`) than on noun-keyword relations. For instance, if we reverse these steps, then Wiksense's performance decreases, showing our parsers' usefulness (e.g., the Vx-Vy step is more important than the last two steps). The *conf=30%* shows that if Wikisense's inference decision for e,g., the *Vx* is *stronger* than *Vy* by at least 30%, then the pronoun target is the subject of the verb, and if the opposite exists, then the pronoun target is the object of the verb. We determined this percentage by testing with the supporting data,

and we observed that it is similar to our decisions. For instance, if we are indecisive about two options and cannot quickly determine which one is the pronoun target, we might return the option with more weight according to our experience.

Algorithm 1 Wikisense's Knowledge Acquisition Algorithm.

```
 1: function RESOLVEPRONOUN (sent, negF, revF, question, answers)
 2:     conf=30%, pairs=[(Vx, Vy), (Sx, Sy), (Ax, Ay), (Nx, Ny), (NBx, NBy), (VBx, VBy)]
 3:     for pair in pairs do
 4:         correctIndex=CALCVALUES (pair, negF, revF, conf, sent, question)
 5:         if correctIndex!=-1 then return answers [correctIndex]
 6:     end for
 7:     return -1
 8: end function
 9: function CALCVALUES (pair, negF, revF, conf, sent, question)
10:     x, y= RUNANDESTIMATE (pair, sent, question)
11:     if negF==True then x,y=y, x
12:     if revF==True then x,y=y, x
13:     if x > y and (x-y)/x >= conf then
14:         return 0
15:     else if y > x and (y-x)/y >= conf then
16:         return 1
17:     else
18:         return -1
19:     end if
20: end function
```

3.4 Experimental Evaluation

In this section, we present results obtained by applying the methodology described in this chapter (see Algorithm 1). The Winograd halves we use in our experiments are derived from the WSC286 dataset, which is intended to be used by participants to tackle the WSC. The dataset consists of 286 halves (143 schemas), and we do not exploit the fact that every two halves belong to the same schema.

Wikisense processed 286 halves, and for each half, it had to return the correct pronoun target, always replying with a noun name or with the strings *don't know* or *unaccomplished*. *Don't know* means that Wikisense could not determine the correct pronoun target, while *unaccomplished* that it could not proceed to knowledge acquisition because the first keyword could not be created (see 1Q in table 3.4).

	Correct	C_A	Wrong	W_A	Unresolved	U_A
Stanford CoreNLP	107	140.5	112	145.5	67	0
Wikisense	170	183.5	89	102.5	27	0

Table 3.6 Results of Stanford CoreNLP and Wikisense (where _A shows the Adjusted scores).

3.4.1 Baselines and Results

Stanford CoreNLP: According to Stanford-NLP-Group[5], Stanford CoreNLP (Manning et al., 2014) is considered the one-stop-shop for NLP, like for the coreference resolution of pronouns. We parse each examined half via Stanford CoreNLP and use the provided results as our baseline. As shown in Table 3.6, Stanford CoreNLP correctly resolves 107 pronouns, incorrectly resolves 112, and does not decide on the remaining 67.

Wikisense: According to our results (see Table 3.6), Wikisense correctly resolves 170 pronouns and incorrectly resolves 89. For the remaining 27, it answers with *don't know* for 12 halves and with *unaccomplished* for 15 halves (see Table 3.6). Hence, the whole procedure ran for 259 WSC halves, and our system achieved a 65.6% prediction score.

Since Wikisense was built for the WSC, it accepts as input the examined half with the two possible pronoun targets, meaning that it restricts its answer only between two possible pronoun targets. Since this is not the case with Stanford CoreNLP, to assure a fair comparison between the two systems, we must ensure that Stanford CoreNLP also resolves the definite pronoun to one of the two pronoun targets. We can see this through the "Adjusted Score" (_A) columns of Table 3.6.

Comparison of the Adjusted scores (_A) shows that our system outperforms the Stanford CoreNLP by 43 points, showing Wikisense's usefulness on the coreference resolution problem. In this regard, possible integration of the two systems can improve the Stanford CoreNLP's usage on the coreference resolution. For instance, in the following schema, which Wikisense resolved correctly, Stanford CoreNLP wrongly resolved the definite pronoun (it) to *table* in both halves: *(1.) The table won't fit through the doorway because it is too wide. What is too wide? answers: The table, The doorway (2.) The table won't fit through the doorway because it is too narrow. What is too narrow? answers: The table, The doorway.*

3.4.2 Support-Vector-Machine Approach

To test Wikisense's usefulness on the WSC and compare it to a machine learning approach, we attempted to resolve the examined schemas (WSC286) through a support-vector-machine (SVM) model. The SVM model was built based on the feature vector values collected earlier

[5]https://stanfordnlp.github.io/CoreNLP/

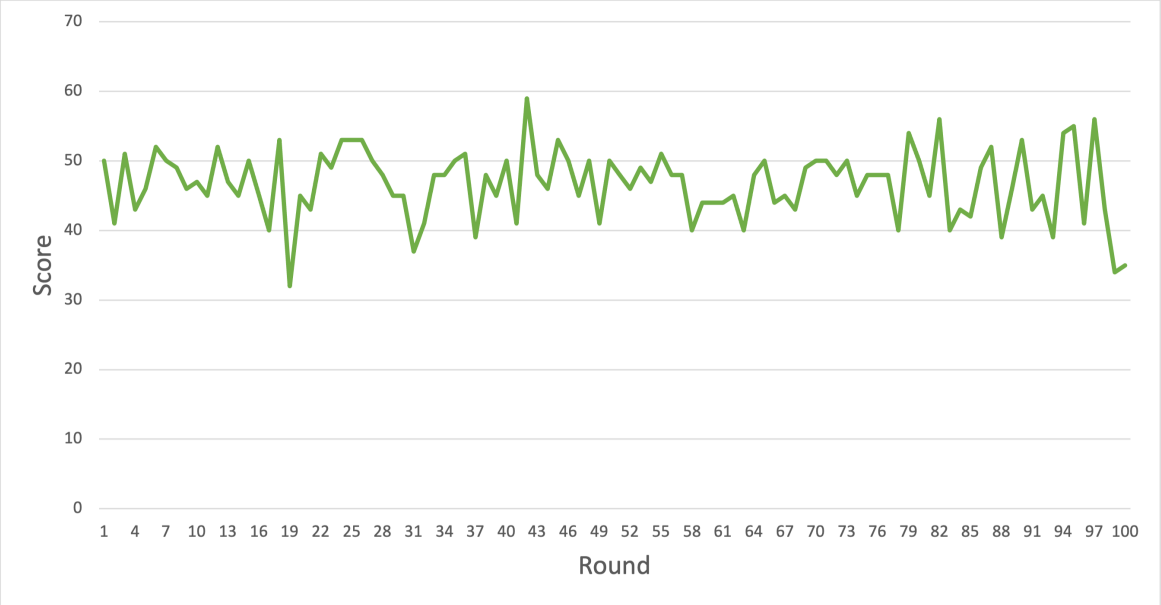


Figure 3.3 Support Vector Machine results. We randomly selected the 70-30 ratio from the WSC286 dataset and ran the procedure 100 rounds.

(e.g., see 3.3). As in Rahman and Ng (2012), we used a ratio of 70-30, where 70% of halves were used for training and 30% for testing. We randomly selected the 70-30 ratio from the WSC dataset and ran the procedure 100 times. The median score of prediction was 47% (minimum 32% and maximum 59%), showing that our initial Wikisense strategy can return better results.

3.4.3 Error Analysis

The whole point of the WSC is to design schemas challenging for resolving ambiguous pronoun references. Of course, not all halves are of the same difficulty level, which is why Wikisense incorrectly resolved some of them. We can see this from various other works that test their systems on corpus subsets (Budukh, 2013; Sharma et al., 2015).

Wikisense failed to answer 27 halves (9%). According to our analysis, the engine could not create the first sentence's part keyword (1Q) in 15 halves. It also failed to locate an important word from the question to create the necessary Indexer's searching-keywords in four halves. Moreover, Indexer did not manage to find many, or rich in context Wikipedia sentences for 8 WSC halves.

Some WSC halves were incorrectly answered because of the similar background knowledge found by the Indexer. For instance, for the schema *30.) The firemen arrived after the police because they were coming from so far away 31.) The firemen arrived before the police*

because they were coming from so far away, through the question *Who came from far away?* Wikisense returned the same pronoun target (*the firemen*). We can see that the differences between the two sentences are negligible.

Furthermore, we have seen Wikisense making the same decision for some schemas (pair of halves) and returning the same pronoun target. By removing these schemas, the score rises to 70% (131 correct, 56 wrong), meaning that we can easily build on Wikisense to enhance its decision mechanism for better pronoun resolution.

3.5 Chapter Summary

This chapter has demonstrated the usability of a technique applied to the WSC by acquiring commonsense knowledge from the English Wikipedia. Designing and implementing systems that emphasize commonsense knowledge might benefit different NLP tasks as a concrete step forward in endowing machines with the ability to reason. In this regard, our study provides an insight into how learning and reasoning through knowledge acquisition can fruitfully interact for pronoun resolution. There is still much room for improvement, but our approach works well with respect to the WSC.

These days, statistical approaches, like deep learning, although they perform really well, have some apparent shortcomings in the sense that they are opaque and, on occasions, brittle. Their obtaining results are not transparent, and at the same time, they do not seem to generalize well when their testing data are too different from their training ones (Marcus and Davis, 2019; Mitchell, 2019).

On the other hand, Wikisense first considers the WSC half at hand and then retrieves the relevant training material on which it is trained. Although there are probably better solutions, we consider this characteristic advantageous because Wikisense can be trained on the fly depending on various keyword settings. At the same time, Wikisense "shows its work", meaning its solutions are transparent, in the sense that, to understand why it took a specific decision, we can look at its beliefs and reasoning. In this regard, Wikisense illustrates how day-to-day commonsense reasoning can be operationalized through a connected collection of inferential knowledge.

Related-Publication

1. Isaak, N., Michael, L.: Tackling the Winograd Schema Challenge Through Machine Logical Inferences. In: Pearce, D., Pinto, H.S. (eds.) STAIRS. Frontiers in Artificial Intelligence and Applications, vol. 284, pp. 75–86. IOS Press (2016), http://dblp.uni-trier.de/db/conf/stairs/stairs2016.html#IsaakM16

4

Using the Winograd Schema Challenge as a CAPTCHA

4.1 Introduction

We approached the WSC when it was first introduced when the research community was faced with the challenge itself and the many difficulties it was introduced with (Levesque et al., 2012; Morgenstern et al., 2016; Morgenstern and Ortiz, 2015). In the first and only WSC, which took place as a side event of the 25th International Joint Conference on Artificial Intelligence (IJCAI-16), no more than six systems participated (from four groups), achieving no more than 48% on the PDP dataset (Davis et al., 2017). According to Davis et al. (2017), no team performed well enough to qualify for the second round of the actual WSC test because of the challenge difficulties. Furthermore, at the time of writing, experiments showed that humans could easily tackle the WSC with an average score of 92% (Bender, 2015).

In the previous chapter, we argued that designing and implementing transparent systems that emphasize commonsense knowledge might benefit different NLP tasks as a concrete step forward in endowing machines with the ability to *understand* any given text. Narrow solutions that focus on behavior in a purely statistical sense are not a solution to the problem, as they do not understand what they are dealing with —something that appears intelligent does not mean that it is (Marcus and Davis, 2019). According to Marcus and Davis (2019), to bring the AI field forward, we need a new generation of researchers who appreciate both classical AI and machine learning. Furthermore, as this is a considerably new challenge in the field, scant attention has been given to finding ways to promote it in various academic disciplines in order to bring more research to work on the problem of actually trying to solve it. To advance the field of AI, researchers must be drawn not only from the computer

science field but also from a wide range of other disciplines, from psychology to linguistics to neuroscience (Marcus and Davis, 2019).

This chapter presents how we utilized the WSC to develop a new type of CAPTCHA (Completely Automated Public Turing test to tell Computers and Humans Apart) as a challenge promotion mechanism that might bring more research from a wide range of various disciplines. Beyond the typical use for security reasons, CAPTCHAs (Von Ahn et al., 2003) have established themselves as a standard technology to distinguish humans from bots and promote AI research in different domains. Consequently, it was natural to consider what other challenge tasks for AI could serve a role in CAPTCHAs, like the WSC, which is a challenging task for bots and is, therefore, a candidate to serve as a form of CAPTCHA. Given the current state of affairs, the WSC is pragmatically more secure in discriminating humans from machines when compared to existing CAPTCHA approaches, such as Image recognition tasks which can be tackled with high accuracy scores. Additionally, recent research in the field investigated how far they could push the difficulty level of Winograd schemas, proposing at the same time various mechanisms to build challenging ones (Cozman and Munhoz, 2020). Moreover, given that there is an infinite number of English sentences that can be formed (Adger, 2019), there is an infinite number of Winograd schemas that can be developed too. In this sense, we can always find ways to build novel forms of challenging schemas.

We want to point out that, like with other kinds of CAPTCHAs, the purpose of the WSC-CAPTCHA development is, through its usage, to encourage researchers to work on the problem of actually trying to solve it, and perhaps, in the process, help towards the building of machines able to reason with commonsense knowledge. Hence, in this chapter, we investigate whether this a priori appropriateness of the WSC as a form of CAPTCHA can be justified in terms of its acceptability by the human users in relation to existing CAPTCHA tasks. An empirical study we undertook, which involved a total of 329 students, aged between 11 and 15, showed that the WSC is generally faster and easier to solve than, and equally entertaining with, the most typical existing CAPTCHA tasks.

4.2 CAPTCHAs

CAPTCHAs are programs that generate and grade challenge-response tests, mainly for security reasons, that most humans can reliably pass, but current computer programs cannot pass (Von Ahn et al., 2003). According to Belk et al. (2013), such tests have been used to prevent automated bots from performing illicit and fraudulent actions, including the degradation of the quality of a provided service. The robustness of a CAPTCHA is its

strength in resisting adversarial attacks, which has attracted considerable attention in the research community (Fidas et al., 2011). On the flip side of things, CAPTCHAs have served as challenges for the AI community to develop automated tools that can reliably pass such tests (Von Ahn et al., 2003).

Perhaps, due to their primary characteristic of being solvable by humans, but beyond the capabilities of current computer programs, CAPTCHAs must also be usable, robust, and friendly to humans. On the other hand, with the progress made by the AI community to develop automated tools that pass CAPTCHAs, some forms of CAPTCHAs can no longer be considered as having those characteristics. For instance, as OCR systems improve, so does their ability to pass text-based CAPTCHAs, resulting in further distorting their presented text, making it less friendly to humans and decreasing the test's usability. The same applies to other types of CAPTCHAs that relate to image recognition. For instance, since 2012, the advances of ConvNets (Deep Convolutional Networks) on the ImageNet computer-vision competition (Krizhevsky et al., 2012) have gradually diminished the security of image-based CAPTCHAs.

At the end of the day, various types of CAPTCHAs end up being more difficult to be passed by humans than by machines (Belk et al., 2013), obviating their need for existence. For example, Google's reCAPTCHA-V1 has been solvable with an accuracy of 99.8% since 2014, prompting Google to abandon this type of CAPTCHA in March of 2018. Similarly, UnCAPTCHA, an AI-based automated system, can break Google's audio-based reCAPTCHA challenges with an accuracy of 85% (Seals, 2017). Other visual-based CAPTCHA schemes were broken with a near 100% success rate by different novel attacks (Yan and El Ahmad, 2007). Similarly, a study designed a novel low-cost attack that leverages deep learning technologies for the semantic annotation of images, automatically solved 70.78% of Google's image reCAPTCHA-V2 challenges (Sivakorn et al., 2016).

The current state of affairs might point to the need for a new type of CAPTCHA, for which AI techniques have not yet been developed to defeat it. In turn, the introduction of a WSC-based CAPTCHA will serve the dual role of presenting the AI community with a new challenge task.

To lay a foundation for WSC-based CAPTCHAs, below, we compare how human performance, usability, and time needed for solving a WSC-based CAPTCHA relate to how humans perform on other types of CAPTCHAs. We start by presenting our methodology, followed by our analysis and discussion of our findings. The sections below explain each of these tasks, along with the tools and techniques we have developed.

4.3 Methodology

To investigate whether this a priori appropriateness of the WSC as a form of CAPTCHA, a request was sent to a secondary education school in Cyprus to recruit participants. To that end, the necessary permissions were obtained from the school's principal and the Cyprus Pedagogical Institute[1], which is responsible for research in public schools in Cyprus.

4.3.1 Recruitment Process

The recruitment process sought to recruit a representative sample of participants that were not familiar with the WSC, based on the fact that this is the first such study undertaken in Cypriot schools. According to Cyprus' department of Secondary General Education[2], participants were familiar with the use of computers, having taken two hours of computer lessons per week as part of their education, and having been active in using the Internet, social networks, and blogs. Moreover, according to the school's computer-science teachers, the participants had been exposed to CAPTCHA challenges at least once in the past. The survey was run in the school's computer science labs (each holding up to 16 students), during a 40-minute period, and under supervision by a school teacher.

Although alternative recruitment processes could have also been adopted, the approach that we have followed was chosen for the following reasons:

- Availability: at the time of writing, one of the authors was a teacher at a secondary school, where after acquiring the necessary permissions, we were able to involve school students in the empirical study.
- Sample-size: Being a teacher eliminated the laboratory studies' well-known limitation, directly related to a small and not necessarily representative sample pool (Gadiraju et al., 2017). Having access to hundreds of participants helps to generalize the experimental findings to larger populations.
- Monitoring: Compared to a crowdsourcing solution, in the *in-class* study that we undertook, we were able to monitor participants closely and measure their response times more accurately.
- Complementarity: Studies with adults have already been reported in the literature, and the present study sought to complement those studies by examining a distinct population and developing a new corpus that might be useful to the research community.

[1] http://www.pi.ac.cy/pi/index.php
[2] http://www.moec.gov.cy/dme/en/index.html

	Grade A	Grade B	Grade C
males	62	52	54
females	33	56	55
10-11	5	-	-
12-13	89	101	8
14-15	1	7	101

Table 4.1 Demographic of participants.

- Developmental: Adults could be argued to have reached a plateau in their WSC (and close to 100% accuracy, raising issues with the statistical analysis), so that age differences would not play an important role in the reported accuracy. For teenagers, we expected that age differences would yield different results worth analyzing.

4.3.2 Participants

The study took part between November 2017 and December 2017, where a total of 329 students volunteered and participated. Participants were teenagers, residents of Cyprus who speak Greek fluently. All of them were students at a single 3-grade gymnasium school, between 11 and 15 years old (see Table 4.1). Participants reported that they were not aware of having any kind of vision problem that hampered their effort to identify colors, shapes, or patterns. Out of 329 students who attempted the task, 17 did not finish the task, while nine students did not volunteer to participate in the survey. Participants were offered a candy costing €0.30 as compensation for their time. Also, they were promised that at the end of the study, the group of students with the best overall results would enter a lottery for a gift of high value, though no value was specified. Although the study was anonymous, each group of participants was identified with a unique group number for the lottery.

4.3.3 Survey Design

The study was designed to record user performance, perceptions, and reaction times on the various types of CAPTCHAs. Participants were asked to complete an online survey designed on Lime-Survey, consisting of questions recording their demographic information and five parts that included different types of CAPTCHAs. At the end of the survey, for each type of CAPTCHA, via a five-point numeric *Likert* scale, participants had to select the level of difficulty from “1: very easy” up to “5: very difficult”. Additionally, they were asked to select the type of CAPTCHA that they considered the most entertaining, using a pull-down selection widget.

Regarding the various types of CAPTCHAs that we could have used (Hasan, 2016), we have chosen a representative sample of text, image, and math-based CAPTCHAs. According to the school's computer-science teachers, in the specific period of time, students were most familiar with distorted-text, 3D-text CAPTCHAs, and Google's image-reCAPTCHA (v2) challenge. The math-based CAPTCHA was included as it requires users to use their cognitive abilities (Hernandez-Castro and Ribagorda, 2010), which relates to the need for cognitive processing that is also present when solving the WSC. Along with the WSC-based CAPTCHAs, we ended up with five different types of CAPTCHAs, where each, given the 40-minute survey period, consisted of twenty instances of the specified type.

Text-based CAPTCHAs

This is the simplest type of CAPTCHA that has been invented and implemented in various services. Basically, via modified/distorted letters and/or digits, Text-based CAPTCHAs try to prevent bots from resolving them to gain access to Web services. Although nowadays bots can easily tackle them, they are still commonly used (Hasan, 2016).

For our study, two text-based CAPTCHA mechanisms were developed using available open-source software. We started with the cool-php-captcha, which is a *distorted-text* CAPTCHA that generates *friendly* CAPTCHAs[3] based on cool colors. We downloaded the PHP files locally to our computer and generated twenty text-based CAPTCHA instances (see Figure 4.1), along with their corresponding correct answers. Finally, we uploaded the instances on LimeSurvey's distorted-text section.

Participants were expected to type in their answer, which was then compared against the correct answer. Each distorted-text word contained an average of 7 characters, generated using the following parameters: random_word_generation=True, Yperiod=12, Yamplitude=14, Xperiod=11, Xamplitude=5, maxRotation=8, image_scale=2, blur=false.

We used text-based CAPTCHAs with Latin characters as most Greek-speaking students are known to use Greeklish daily (Themistocleous, 2009), which are Greek-written words using Latin/English characters. Even Google provides Greeklish support in Gboard, which helps people transform Greeklish into Greek. To verify the familiarity of our participants with Latin characters, we ran a simple pre-study test two weeks before our survey. All school students were asked to select "Latin" or "Greek" to the question *Which character set do you find easier to use on a QWERTY keyboard? Latin, or Greek?*. According to our results, 94% of the students selected "Latin" as their preferred input language in their day-to-day life.

[3]https://github.com/josecl/cool-php-captcha

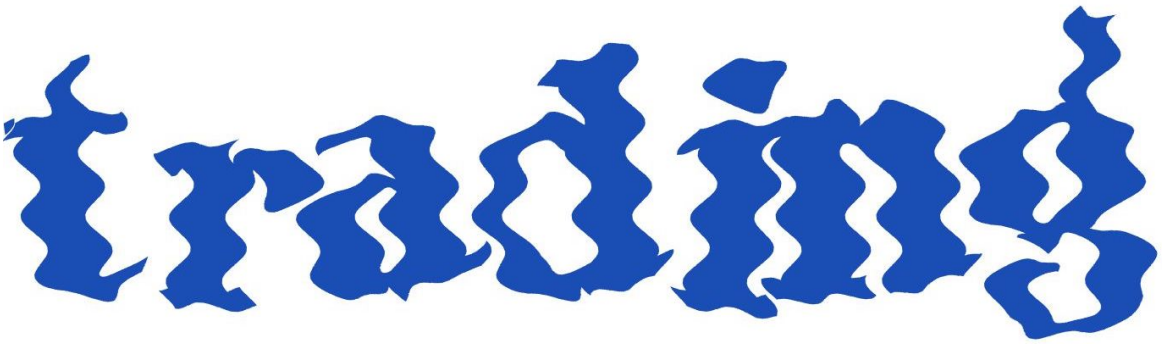


Figure 4.1 A distorted-text CAPTCHA example.

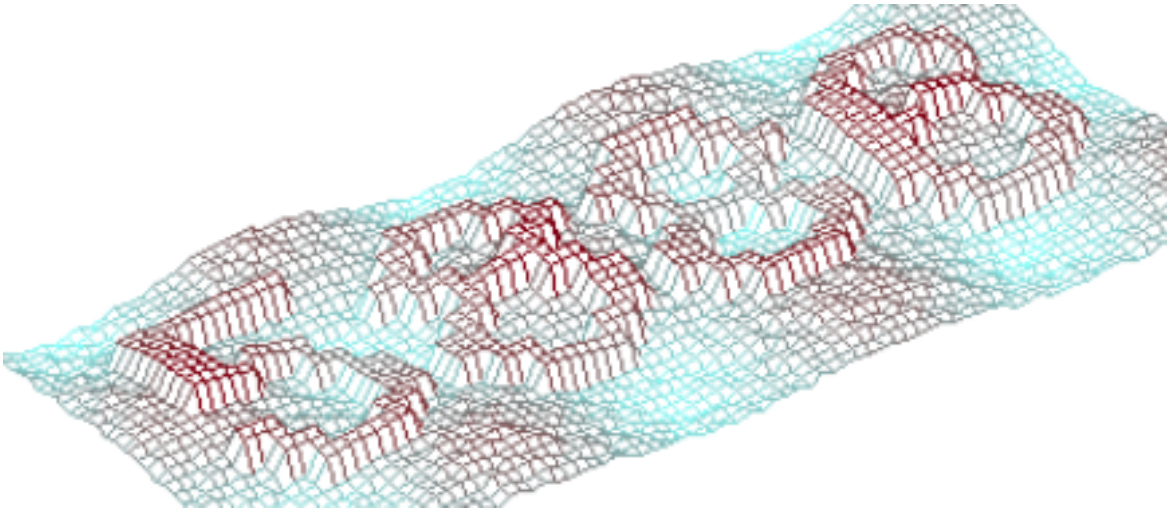


Figure 4.2 A 3D-text CAPTCHA example.

Our second text-based CAPTCHA was a PHP 3D-text CAPTCHA[4], which randomly generates 4-digit 3D CAPTCHAs (see Figure 4.2). Like with the distorted-text CAPTCHA, we requested and saved locally twenty instances of the 3D-text CAPTCHA and uploaded them on the LimeSurvey's 3D-text section. Each 3D-text word contained four integers, generated using the following parameters: startX=random(0,35), startY=random(0,80), angle_of_camera_moved_up=35.

[4]https://github.com/qmegas/captcha-3D

Image-based CAPTCHAs

Image–based CAPTCHAs require selecting images from a set of such that have a certain characteristic (Hasan, 2016). For instance, to prevent bots from resolving them, these kinds of CAPTCHAs require selecting images that belong to a specific category, as the selection of images that show street signs. In this regard, an image CAPTCHA mechanism[5] was used to implement the Google reCAPTCHA-V2 service (see Figure 4.3). Even though Google advertises this reCAPTCHA-V2 service as invisible to humans and visible only to bots (Verger, 2017), students often have to solve this type of CAPTCHA to access their email accounts.

As with text-based CAPTCHAs, we requested and saved locally twenty instances from the CAPTCHA service along with their corresponding correct answers and uploaded them on LimeSurvey's Image-based section. In order to apply real-life situations via JavaScript, we programmed each image to work via click-and-select. In this regard, participants were expected to select the right images by clicking on them, and their choices were then compared against the correct answer.

After we made sure the twenty randomly generated Image-based CAPTCHAs would not share any images, we ended up with the following instances:

- Three CAPTCHA image sets (2 x 4) based on storefronts.
- Seven CAPTCHA image sets (3 x 3) based on cars, mountains or hills, bridges, and apartment buildings.
- Ten CAPTCHA image sets (4 x 4) based on vehicles and street signs.

Math-based CAPTCHAs

Math-based CAPTCHAs ask users to solve a mathematical equation in order to differentiate them between bots. The difficulty level of the mathematical equation varies across implementations (Hernandez-Castro and Ribagorda, 2010). In this regard, this type of CAPTCHAs can be either considered too easy or too complicated to solve.

Among the several available implementations, and considering the age and educational level of our participants, we opted for choosing an implementation that produced relatively easy tests[6] that use simple arithmetic operations (see Figure 4.4). As with other types of CAPTCHAs, we downloaded the necessary PHP files, generated and saved locally twenty

[5]https://demo.codeforgeek.com/google-captcha/
[6]https://www.hscripts.com/scripts/php/math-captcha.php

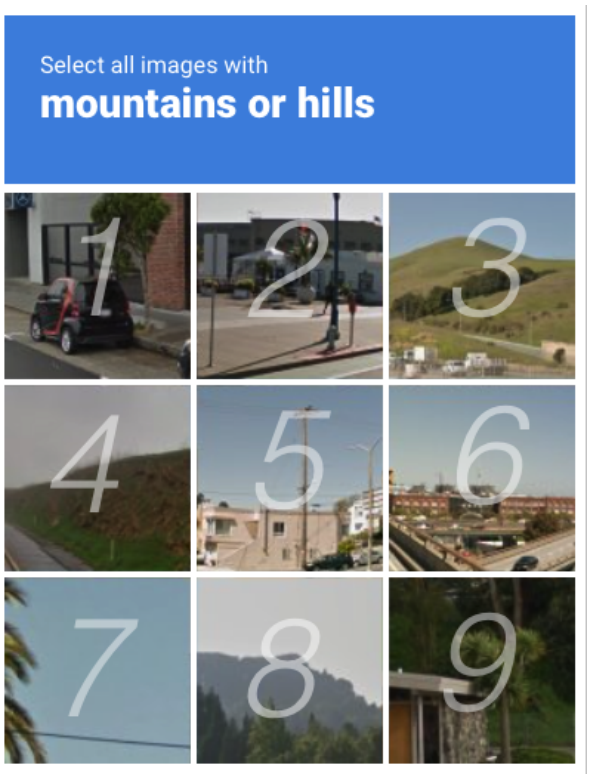


Figure 4.3 An image-based CAPTCHA example.

math CAPTCHA instances, along with their corresponding correct answers, and uploaded them on the LimeSurvey's math-based CAPTCHA section.

A WSC-based CAPTCHA

This is the first-ever designed WSC-based CAPTCHA, so we had to develop our CAPTCHA service from scratch. A client (e.g., Web developer) who wants to utilize our WSC-based CAPTCHA service has to register via an email account to receive an access key (see registration-process in Figure 4.5). Next, a call to the CAPTCHA API with the correct access key returns a randomly chosen half (sentence, question or a definite pronoun, and the two possible answers). Finally, a client submits a selected answer and receives a response on whether it is correct or not (see protection-mechanism in Figure 4.6).

Through an optional argument defined in the API call, a client can request Winograd halves in different languages. Furthermore, they can request halves that follow either the PDR dataset (Rahman and Ng, 2012) or the original dataset (WSC_) pattern (Levesque et al., 2012), that is, halves with definite pronouns or questions. The served WSC-based CAPTCHA instances are currently based on WSC halves developed by the authors, research collaborators, or taken from various published corpora (Amsili and Seminck, 2017; Levesque

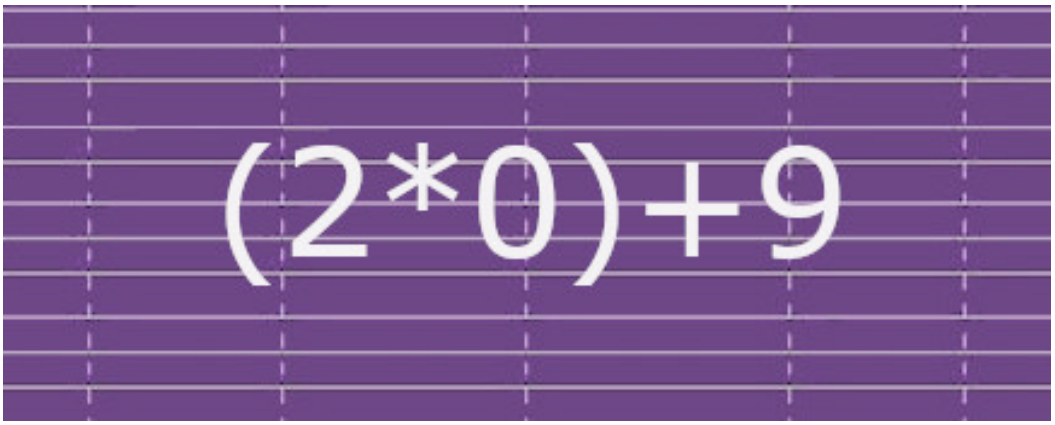


Figure 4.4 A math-based CAPTCHA example.

et al., 2012; Rahman and Ng, 2012). Additionally, we have designed our tools and platform to help with the gathering and evaluation of multilingual CAPTCHAs[7]. At the time of writing, our service consists of 3000 schemas, and our registered users can add Greek, French, and English schemas. The schemas are manually checked by us for consistency with the WSC rules and then added to the service's database.

As an example of interaction with the service, when requesting an English WSC-based CAPTCHA, the service might be called with http://cognition-srv1.ouc.ac.cy/wsc/loadWscByIdnew.php?key=trial&lang=en, and return the following: *ID: 1537 Sentence: George scored against Thomas in the shootout, so he won the game. Pronoun: he Answers: George, Thomas*. To check a proposed answer for that instance, the service might be called with http://cognition-srv1.ouc.ac.cy/wsc/receive_Id_AnswerToRespondGet.php?sentenceid=1537&answer=George., and return the following: *Result: correct*.

To cater for the study's Greek-speaking participants, the WSC halves used in the study were developed by a Greek Literature teacher (Ph.D.) and added to the service's database (see Figure 4.7). Beyond ensuring that the instances followed the rules of the WSC in terms of having co-referents of the same gender and number, the teacher was asked to develop Greek Winograd schemas that are comparably challenging as those found in the literature. According to the Greek Literature teacher, the developed schemas were judged to be of medium difficulty, considering that each sentence included two verbs.

For the purpose of this study, we requested and saved locally twenty instances from the WSC-based CAPTCHA service, along with their corresponding correct answers, and uploaded them on the LimeSurvey's WSC section. At the time of the experiment, participants were expected to select the correct answer by clicking on a radio button.

[7]http://cognition.ouc.ac.cy/ws_builder

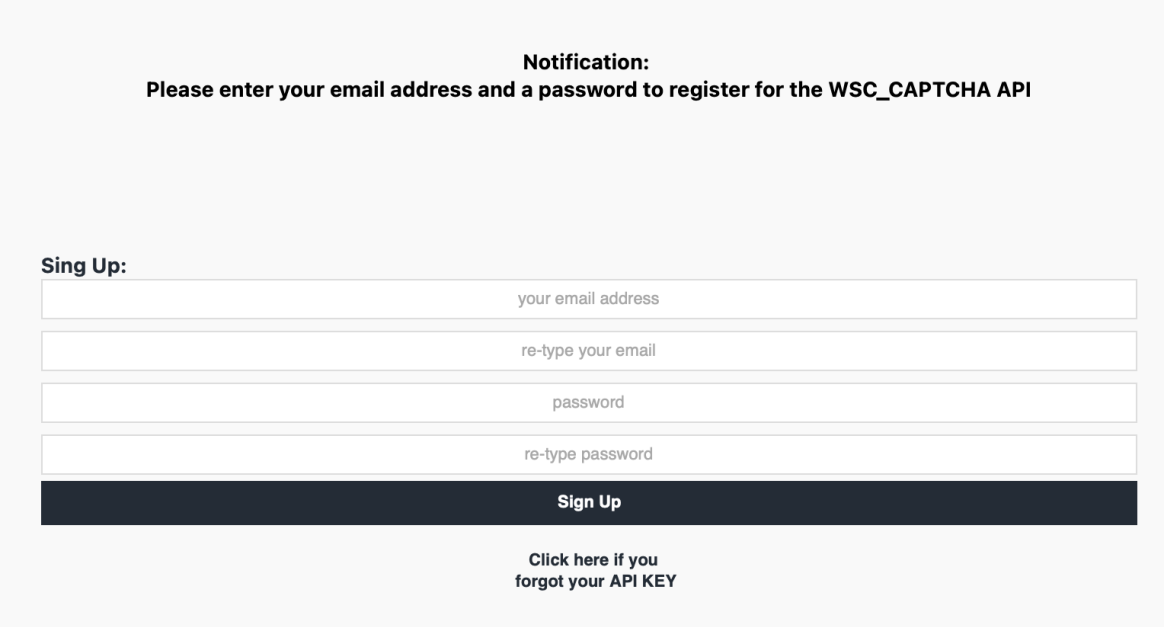


Figure 4.5 Registration form for the WSC-based CAPTCHA service.

4.3.4 Materials

For the purpose of this experiment, we used LimeSurvey[8], an online survey tool installed on Cognition Lab's server[9]. Participants took the experiment in School's computer science labs equipped with seventeen identical 17" desktop computers that run on i5 Intel CPUs. Each lab consists of three rows of desktop computers that are based on single-seat basic tables. Participants took the experiment on the Google Chrome browser window.

4.3.5 Procedure

Participants were instructed to answer each question quickly but without sacrificing accuracy. Also, they were told that surfing the WWW was not allowed, nor talking to each other or asking questions to the teachers. Although all participants faced the same CAPTCHA instances, the five types of CAPTCHAs were presented to the participants in different order to counterbalance any ordering effects.

The participants participated in the survey by visiting the school's web page and following the provided links. Each student was able to see one CAPTCHA on each page, with directions written in Greek, and their progress was displayed at the top of the page. Once an instance was completed, it was not possible to revisit and edit the answer.

[8]https://www.limesurvey.org
[9]https://cognition.ouc.ac.cy/surveys/index.php

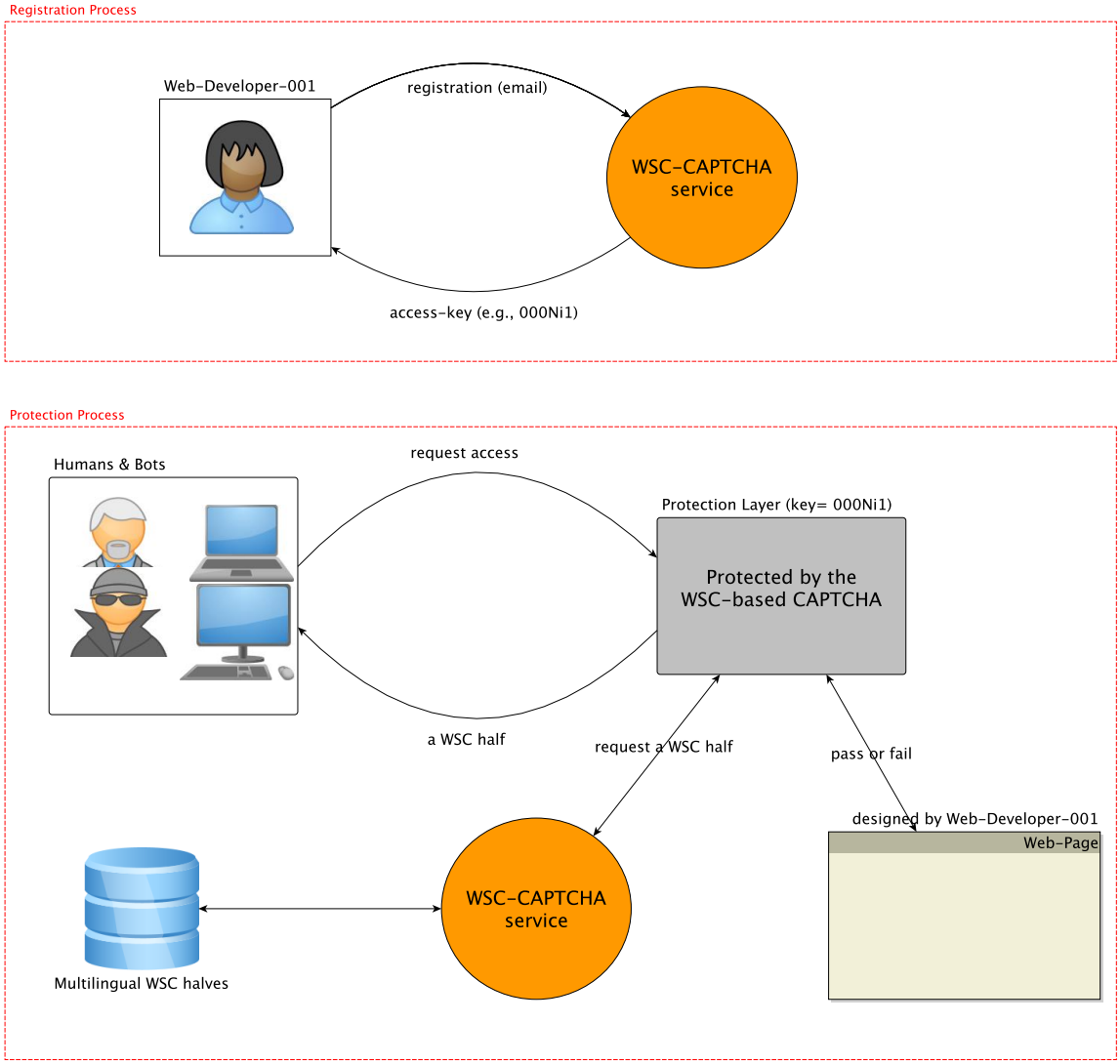


Figure 4.6 The WSC-based CAPTCHA protection mechanism.

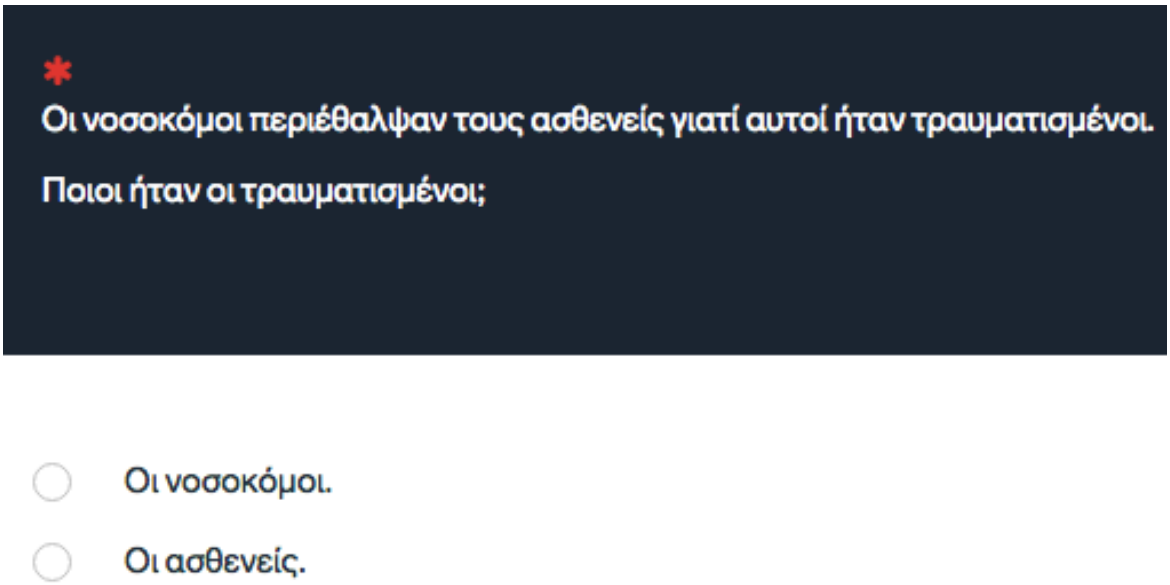


Figure 4.7 A Greek Winograd half, used in the study. English-translation: Sentence: The nurses treated the patients because they were wounded. Question: Who were wounded? Answers: The nurses, The patients.

4.3.6 Hypotheses

For the purpose of this study, we formulated four null hypotheses:

1. Participants cannot achieve higher accuracy on the WSC-based CAPTCHA than on other types of CAPTCHAs.
2. There is no significant difference regarding the time needed to solve different types of CAPTCHAs.
3. Participants do not consider the five types of CAPTCHAs as challenging to pass.
4. There is no general preference of participants towards a certain type of CAPTCHA.

4.4 Results and Discussion

The survey was administered to 329 participants, of whom 312 (94%) were able to fully complete it. According to our results, participants scored the highest mean accuracy of 82% ($\sigma = 0.12$) on the WSC-based CAPTCHAs (see Figure 4.8) and the lowest mean accuracy of 48% on the image-based CAPTCHAs ($\sigma = 0.36$). On the text-based CAPTCHAs, participants scored a mean accuracy of 69% ($\sigma = 0.26$) on the 3D-text CAPTCHAs, and

71% ($\sigma = 0.24$) on the distorted-text CAPTCHAs. On math-based CAPTCHAs, where some cognitive processing similar to the WSC was needed, they scored a mean accuracy of 74% ($\sigma = 0.15$), 6% below the score on the WSC-based CAPTCHAs.

Regarding the survey results, the difference in scores was shown to be statistically significant. Using an ANOVA analysis, the first null hypothesis was rejected with F=5.52 > Fcrit=2.46 (p=0.00048), meaning that participants achieved higher accuracy on the WSC-based CAPTCHA than on other types of CAPTCHAs. Thus, for WSC-based CAPTCHAs, even teenagers can achieve a significantly high degree of accuracy, whereas machines are still not able to reliably solve the task.

On the contrary, participants scored a mean accuracy of 71% on distorted-text CAPTCHAs, while machines tackle them almost to an accuracy of 100% (Yan and El Ahmad, 2007), meaning that the general strength of machines in solving this type of CAPTCHAs is an area of increasing concern (Hernandez-Castro and Ribagorda, 2010). Regarding the 3D-text CAPTCHAs, which do not offer more security than the traditional 2D-text CAPTCHAs (Nguyen et al., 2014), participants scored a mean accuracy of 69%, which is very close to the 71% of the distorted-text CAPTCHAs. Some weaknesses of text-based CAPTCHAs can be seen from recent achievements in the field. For instance, Google engineers have defeated distorted-text CAPTCHA thanks to a Street View algorithm by 99.8% (Technoblog.org, 2017; Tung, 2017). Furthermore, unCAPTCHA can break the audio version of this challenge by 85% (Seals, 2017).

On image-based CAPTCHAs, while there are systems that can solve them to an accuracy of 70.78% (Sivakorn et al., 2016), our participants did not manage to achieve a score higher than 48%. Finally, while participants scored a mean accuracy of 74% on math-based CAPTCHAs, there are numerous articles that show how they can be handled as textual challenges that can be easily parsed and solved, using, for instance, the DeCaptcher[10] service. According to Anvesh Sinha (2016), this type of CAPTCHA can be easily tackled using a low-cost attack.

4.4.1 Timing

To the best of our knowledge, this is one of the first reports of timing comparisons between different types of CAPTCHA. In this regard, our timing results could be used by other researchers as part of their future works. A cursory glance at Figure 4.9, which depicts the response time distribution on the different types of CAPTCHAs, shows that participants

[10]https://de-captcher.com

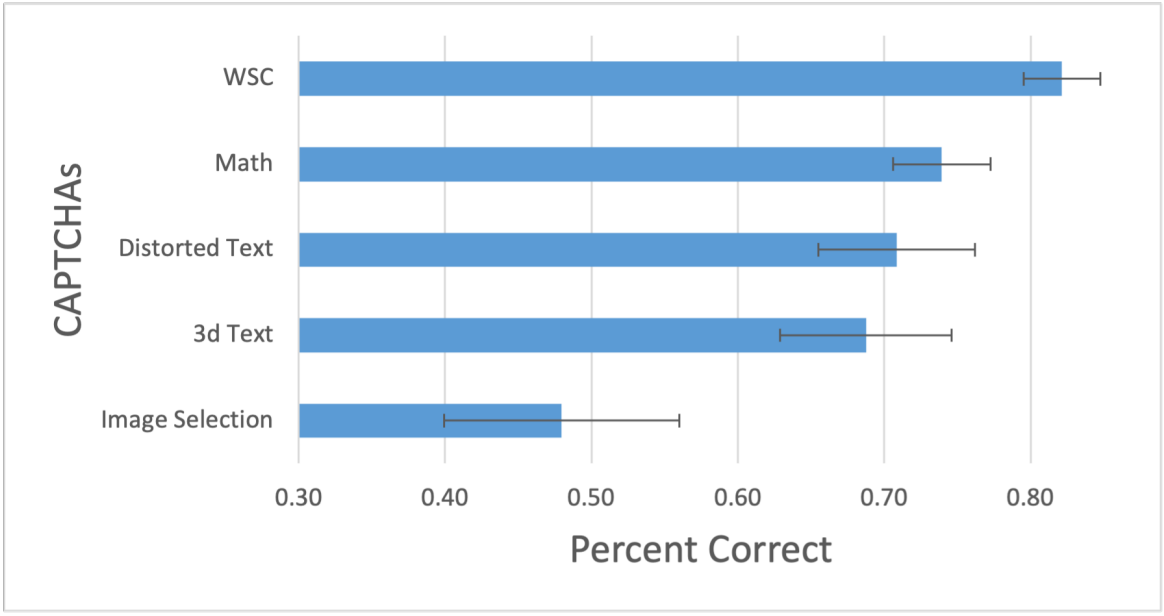


Figure 4.8 Distribution of scores (with standard errors) on solving different types of CAPTCHAs.

scored the lowest timing mean on the WSC-based CAPTCHA (13.41 seconds) and the largest on the distorted-text CAPTCHAs (15.58 seconds).

Although there was a slight difference in median values across timings between the WSC-based CAPTCHA and the distorted-text CAPTCHA (only 2.17 seconds), we consider the timing differences to be a finding warranting further examination, as it suggests that the second null hypothesis might also be rejected. According to our results, not only a WSC-based CAPTCHA can be solved faster than a distorted-text CAPTCHA, but also more accurately.

One could argue that participants' ability to tackle faster the WSC-based CAPTCHA is a direct consequence of the binary nature of the responses in the WSC-based CAPTCHAs, compared to the distorted-text CAPTCHAs where the answer needs to be typed in. On the other hand, we argue that the WSC-based CAPTCHAs require significant time to read the sentence and additional time to reason about the answer, which relates to each individual's commonsense and reasoning abilities. Moreover, regarding the WSC, where recent studies with adults have reported an accuracy score of at least 92%, it might show the challenge difficulties for our teenager participants. Furthermore, we know that current bots require multiple minutes to solve a WSC sentence; e.g., Wikisense needs, on average, three minutes for each WSC half. Oppositely, bots break existing CAPTCHAs very quickly; e.g., unCAPTCHA breaks 450 reCAPTCHA audio challenges in under 6 seconds (Bock et al., 2017; Seals, 2017).

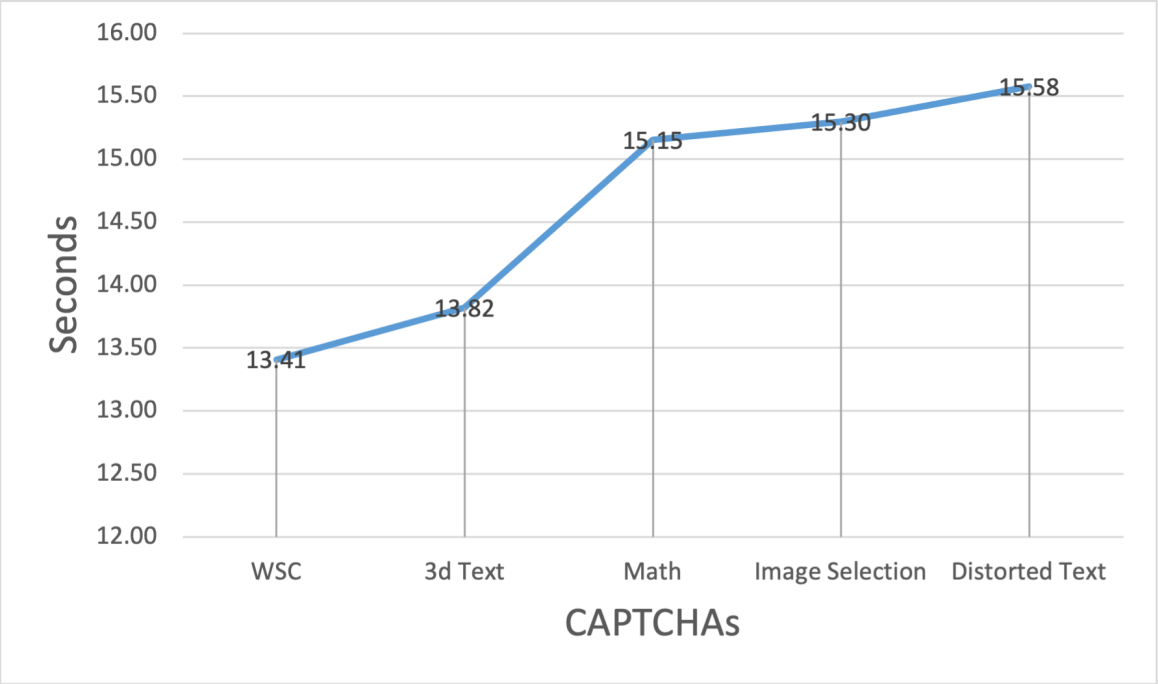


Figure 4.9 Distribution of response times on solving different types of CAPTCHAs.

4.4.2 Grade and Age Factor

Ninety-five of our participants were students of the first high-school grade (aged 11-12), 108 were students of the second high-school grade (aged around 13), and 109 were students of the third high-school grade (aged 14-15).

Results provide preliminary evidence of a positive correlation between grade and accuracy across all types of CAPTCHAs. Specifically, the higher the grade, the better the results, meaning that students of more advanced grades and greater age achieved higher scores —the only exception was the second and third grades that achieved the same score on the WSC-based CAPTCHA (see Figure 4.10).

The most significant difference in scores was between the first and second grades, where participants of the second grade scored 2% more on WSC-based CAPTCHAs, 3% more on 3D-text CAPTCHAs, 4% more on image-based CAPTCHAs, 7% more on math-based CAPTCHAs, and 8% more on distorted-text CAPTCHAs. Moreover, regarding the timing differences, it was shown that the average timings of students of more advanced grades were smaller across all CAPTCHA types (see Figure 4.11).

According to Bender (2015), differences in value judgments rely on the knowledge that people accumulate through their experiences in the real world. In this regard, based on our results, it can be inferred that humans can answer commonsense questions like those in the

WSC from the early high school ages. On the other, this does not seem to happen with the other types of CAPTCHAs, where competence in the tasks seems to increase with age, even within the high school age span.

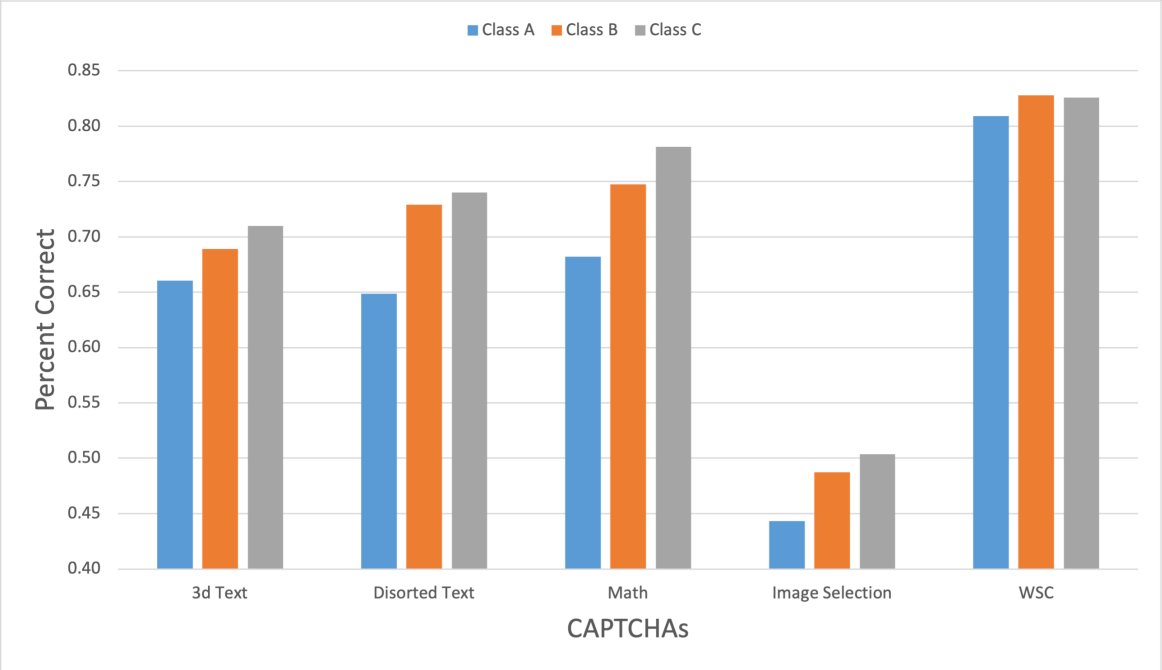


Figure 4.10 Distribution of scores on solving different types of CAPTCHAs, based on participants' classes.

4.4.3 Gender Factor

We have also undertaken an analysis of our results based on participants' gender. The main goal was to determine if the gender factor plays a significant role in participants' ability to tackle the various types of CAPTCHAs. Recall that every participant who completed the experiment also submitted their gender (see Table 4.1).

According to our results, significant correlations were obtained between gender and accuracy across all types of CAPTCHAs (see Figure 4.12). Specifically, on the text-based CAPTCHAs, the mean female score was 71% on the 3D-text CAPTCHAs and 75% on the distorted-text CAPTCHAs, which are 4% and 8% more than the male scores, respectively. Similarly, on math-based CAPTCHAs, the mean female score was 77%, 6% more than the mean male score, and on image-based CAPTCHAs, the mean female score was 51%, which is 5% more than the mean male score. Finally, on the WSC-based CAPTCHAs, the mean female score was 85%, which is 6% more than the mean male score. With respect to the

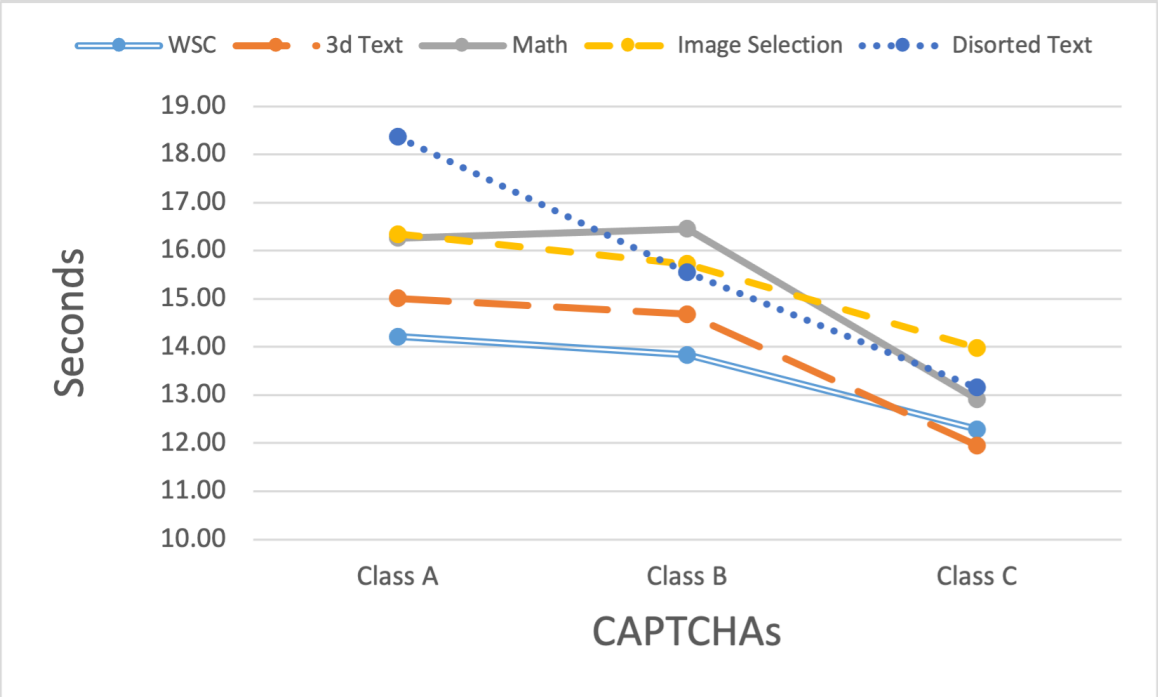


Figure 4.11 Distribution of response times on solving different types of CAPTCHAs based on participants' classes.

WSC, our findings reveal a higher rate of achievement from females. The difference of 6%, considering the challenge difficulties, might show significant differences in value judgments or unfamiliar concepts in the day-to-day life of the two genders.

4.4.4 Participant's Subjective-Judgments

Recall that for each type of CAPTCHA, a five-point numeric *Likert* scale was used to rank the level of difficulty from "1: very easy" up to "5: very difficult". Figure 4.13 is a graphic summary of the participants' subjective evaluation of the *difficulty* of different types of CAPTCHAs. On the five-point numeric *Likert* scale, 54% of the participants rated WSC-based CAPTCHAs as very easy, 27% as easy, and only 3% rated it as very difficult. Overall, it seems that the majority of the participants (81%) consider the WSC-based CAPTCHA as an easy type of CAPTCHA. As depicted in Figure 4.13, the math-based CAPTCHA was judged as being difficult by 19% of the participants, followed by the 3D-text CAPTCHA with 13%, the distorted-text CAPTCHA with 13%, and the image-based CAPTCHA with 8%, equal to the 8% of the WSC-based CAPTCHA.

Taken altogether, the data presented here provide evidence that the participants consider the WSC-based CAPTCHA as an easy CAPTCHA and the other types of CAPTCHA as

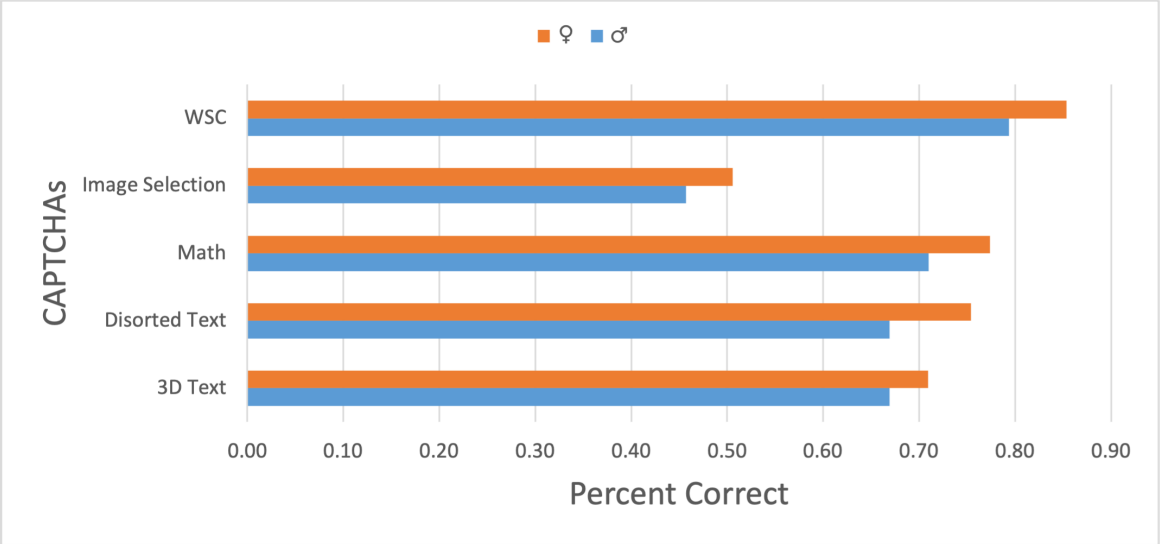


Figure 4.12 Distribution of scores on solving different types of CAPTCHAs, based on participants' gender.

harder. Based on the results, we can reject the third null hypothesis, meaning that participants consider current CAPTCHAs as hard ones that seem to hamper usability and productivity.

At the end of the survey, participants were asked to select the most entertaining CAPTCHA type (see Figure 4.14). In this regard, 22% of the participants selected the distorted-text CAPTCHA as the most entertaining type, 20% the 3D-text CAPTCHA, 19% the WSC-based CAPTCHA, along with the image-based CAPTCHA, and only 12% selected the math-based CAPTCHA as the most entertaining type. Eight percent of the participants did not select any of the CAPTCHA types as being entertaining.

It might seem counter-intuitive that both the 3D-text CAPTCHA and the distorted-text CAPTCHA were ranked as difficult by 13% of the participants, but at the same time, they were selected as the most entertaining types of CAPTCHAs. A possible reason for this discrepancy might be that they were entertaining because they were puzzling. After all, it is gratifying when people work out games and puzzles. Another reason is that some participants might have been drawn to select the distorted-text and 3D-text CAPTCHAs as most entertaining because of the use of color. Maybe modified WSC-based CAPTCHAs that utilize color to highlight the pronoun and its two possible co-referents might be viewed as more entertaining. Furthermore, even though the WSC-based CAPTCHA was not first in the entertainment ranking, we believe that the difference from the first in ranking might be inconsequential, especially given that unfamiliarity with the WSC-based CAPTCHA might have negatively impacted the participants' judgment. Overall, in terms of the WSC-based

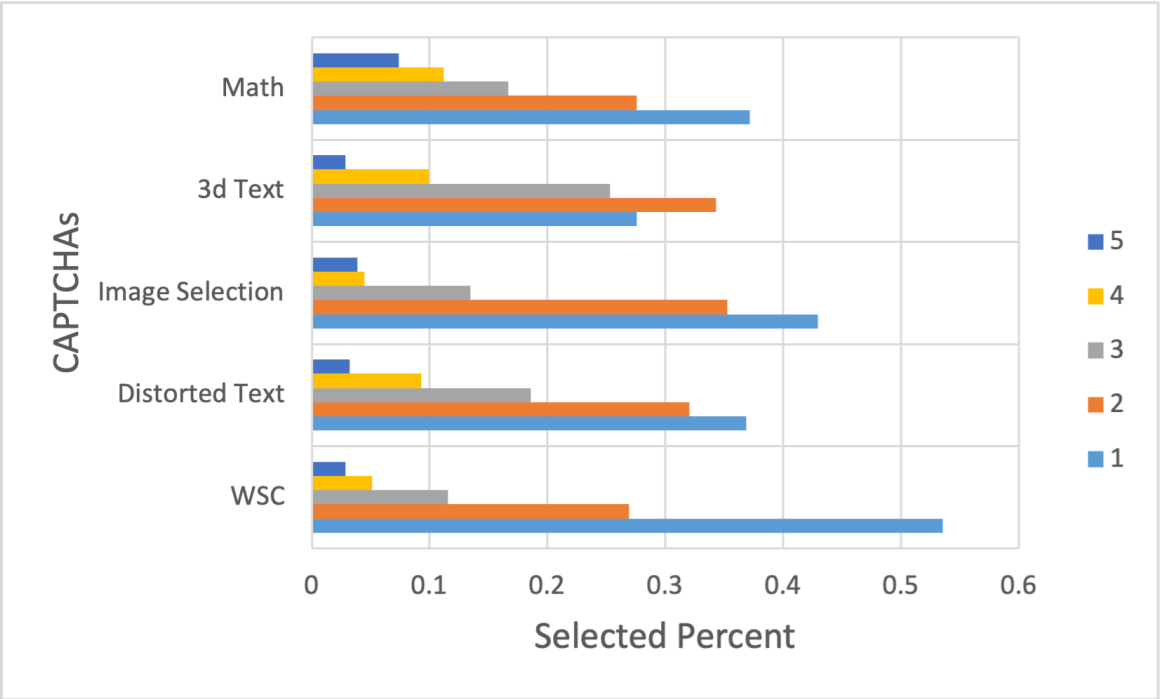


Figure 4.13 Distribution of participant preferences via a Likert scale that scores the difficulty of different types of CAPTCHAs.

CAPTCHA, its combined low score for difficulty and high score for entertainment suggests that it might find wide acceptability among users.

The general picture emerging from the analysis is that participants' opinions in terms of CAPTCHA difficulty and entertainment scores show that different CAPTCHA types are evaluated differently. Users might have preferences among different types of CAPTCHAs, which rejects the fourth null hypothesis.

4.4.5 Participant Observation-Analysis

Teachers who were responsible for monitoring the study have forwarded some remarks from the students. In the process of reviewing these remarks, we noticed several concerns or upshots worth mentioning.

Although questions were not allowed during the test, an interesting side finding was that students, mainly from the two first grades, asked questions about the meaning of words in WSC halves. For instance, some students were not familiar with the word *shallow* (as in "shallow water"), or the word *magpie* as being a specific type of bird. On another occasion, in a specific WSC half, which had the lowest accuracy score, with 40% of the students answering incorrectly, students were unable to determine whether the word *unstable* was

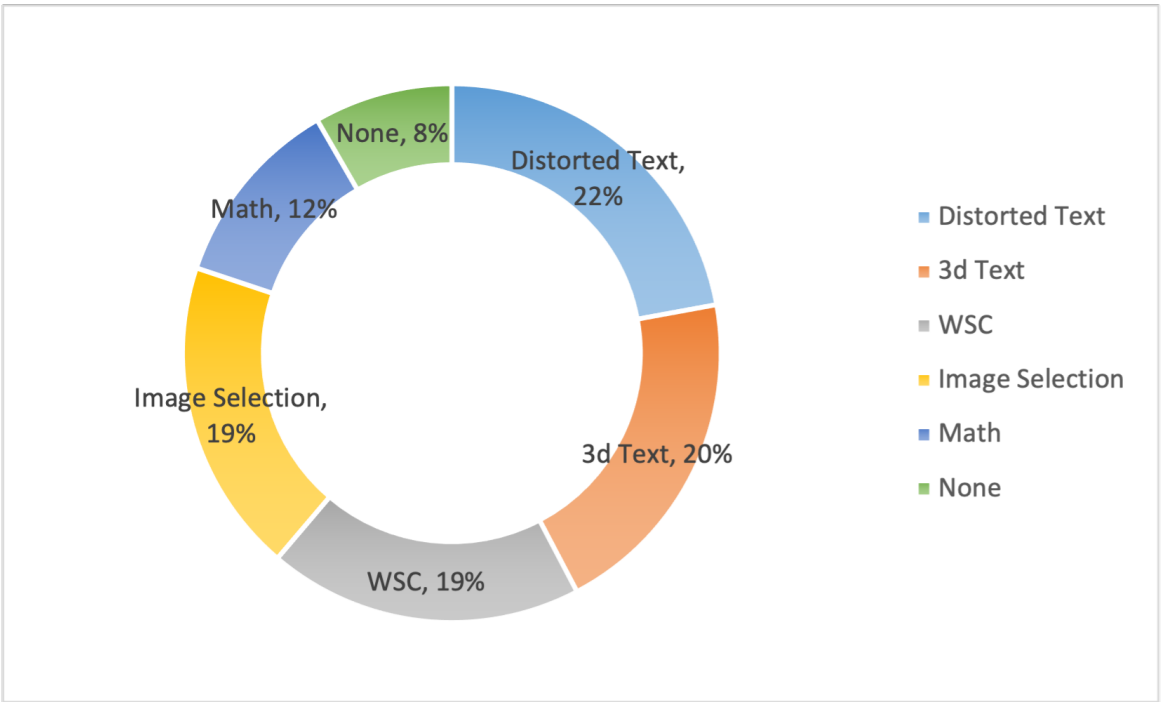


Figure 4.14 Distribution of participant preferences based on the most entertaining type of CAPTCHA.

meant to characterize a girl or a chair, with both entities being feminine in Greek. It seems that students scored better on halves that were more directly related to their own experiences, with the WSC half mentioning, for example, a student and an exam being answered correctly by 98% of the participants.

Teachers also noticed that text-based CAPTCHAs, which were judged as difficult by 13% of the participants, forced students to rotate the screens to determine the displayed text. Furthermore, students complained about ambiguities in the image-based CAPTCHAs, not knowing, for instance, if choosing an image that showed the wheel of a car was acceptable as an image of a car, and whether choosing an image that showed the pole of a street sign was acceptable as an image of a street sign. It might be a stretch to tackle this type of CAPTCHA task, which, indeed, seems to be a big problem in Google's new CAPTCHA service (Technoblog.org, 2017). Regarding the math-based CAPTCHA, teachers commented on some students who were asking to use a calculator to calculate the results of the simple arithmetic operations.

According to our results, the two groups of students that asked the most questions were the groups that achieved the lowest mean accuracy score. Perhaps unsurprisingly, the same two groups of students come from the same class, which is considered by school teachers to

be the class with the lowest-performing students of the school, and the one with the most students that take extra support lessons in school.

4.5 WSC-based CAPTCHA benefits and security

Beyond the conclusions resulting from our study on the appropriateness of the WSC as a novel form of CAPTCHA, it is instructive to consider two other aspects of WSC-based CAPTCHAs.

4.5.1 Accessibility Benefits

According to Elson et al. (2007); Moreno et al. (2014), certain types of CAPTCHAs provoke and raise accessibility barriers, especially to users with disabilities. For instance, vision-impaired users have difficulties with CAPTCHAs that include images or text. Acknowledging that not all users can recognize, solve, and access a CAPTCHA and that certain types of CAPTCHAs are inherently not adjustable to address these concerns has led researchers to find other ways to control spam by bots (Elson et al., 2007). In this regard, future CAPTCHA implementations should incorporate several methods to interact with users following the Web-Content Accessibility-Guidelines (WCAG 2.1)[11].

In this work, we put forward that the WSC-based CAPTCHA can offer a way out of this situation, as it can easily be adapted to adopt the Web Content-Accessibility-Guidelines (Yan and El Ahmad, 2008) to be perceivable, operable, understandable, and robust, and provide a solution that is accessible to people with disabilities. Although based on text, the WSC-based CAPTCHA is not predicated on the difficulty of people being able to *read* the text, as is the case in the text-based CAPTCHAs. For instance, day-to-day text-based or voice-based digital assistants (e.g., chatbots, Siri, ALEXA) can provide state-of-the-art text-to-speech assistance to people with disabilities. Thus, one can easily envision extensions where the WSC sentence, question, and the user's answer are communicated verbally. Not only can this extension cater to vision-impaired users, but it can also cater to users who might be unable to use a keyboard either because of mobility issues or because of lack of an input device. On the other hand, people unable to speak can easily choose between the two possible answers with a simple mouse click, hand gesture, or a keypress.

Moreover, based on our earlier observation that the use of color in a CAPTCHA might have a positive impact on its usability or accessibility (Yan and El Ahmad, 2008), one can

[11] https://www.w3.org/TR/WCAG21/

consider designing WSC-based CAPTCHAs that can use images to represent the possible answers or use colors to highlight the important parts of the WSC half.

4.5.2 Security Strengthening

Although the WSC-based CAPTCHA usage was justified in terms of its acceptability by the human users, one could argue against using the WSC-based CAPTCHA because its error rate for discriminating humans from machines is not sufficiently low. For instance, even if humans could achieve an accuracy of 100%, machines can, at the very least, achieve 50% accuracy by chance.

This reasoning is based on the fact that WSC relies on closed-ended questions and that these questions have only two possible answers, which thwarts its security. The reasoning goes that other CAPTCHAs are more appropriate since they either use open-ended questions (e.g., text-based CAPTCHAs) or close-ended questions with several possible answers (e.g., image-based CAPTCHAs) and ensure a lower discrimination error rate.

Although the point about other existing types of CAPTCHAs being *in principle* better at discriminating humans from machines than the WSC-based CAPTCHAs is well-taken, the argument above remains mostly a philosophical one. It seems that the discriminatory power of existing types of CAPTCHAs is, nowadays, worse than the WSC-based CAPTCHAs for the simple reason that machines can solve those CAPTCHAs with an accuracy much higher than 50%, often comparable to that of humans.

Nevertheless, to shield the WSC-based CAPTCHA Achilles' heel, one could consider certain extensions to increase its security level at the expense, potentially, of its ease of use. To aid in the design of a more secure WSC-based CAPTCHA, one could apply the following enhancements:

1. Turn the half's question into an open-ended one by asking the user (human or bot) to identify and type in the answer among possibly multiple co-referents in the half's sentence.

2. Combine distorted-text CAPTCHA techniques to obscure the possible co-referents in the half's sentence. Ask the user/bot to identify and type the pronoun target in a specified text-box.

3. Require the successful resolution of multiple WSC halves, in a row, within a single WSC-based CAPTCHA.

4. Combine mouse movement techniques, as used in Google's reCAPTCHA service, to see if a human or bot moves the mouse to select the correct answer.

5. Combine image-based CAPTCHA techniques by presenting the potential answers of a WSC-based CAPTCHA instance as images. Like in the image-based CAPTCHA, humans or bots should select the right images by clicking on them.

6. The WSC-based CAPTCHA service could ban and block IP addresses that might repeatedly try random answers to pass the challenge.

7. Turn the WSC half into an image-based puzzle, where the user/bot has to connect the correct pronoun target to the correct position. The two pronoun targets would be the only ones to match the position of the definite pronoun.

4.6 Chapter Summary

In this chapter, we have argued that the Winograd Schema Challenge (WSC) can form the basis of a new type of CAPTCHA. We have presented this WSC-based CAPTCHA's nature, highlighting the shortcomings of typical existing approaches, providing at the same time motivation for a detailed WSC-based CAPTCHA design. Although CAPTCHAs' designing is a tedious task, we expect this work to be a good starting point for future WSC-based CAPTCHAs designers.

Beyond offering a type of CAPTCHA that, given the current state of affairs, seems to be more secure in distinguishing humans from machines when compared to existing approaches, this study has shown that this is achieved without essential compromises in usability. The a priori appropriateness of the WSC as a form of CAPTCHA was justified in terms of its acceptability by the human users in relation to existing CAPTCHA tasks. In this regard, it was shown that the WSC is generally faster and easier to solve than, and equally entertaining with, the most typical existing CAPTCHA tasks. To address some of the weaknesses of closed-ended questions, these types of CAPTCHAs might have, various security improvements that enhance the CAPTCHA service's protection mechanisms were introduced.

We expect that the adoption and use of WSC-based CAPTCHAs will encourage more AI researchers to work on the problem of actually trying to solve the WSC, and perhaps, in the process, help towards the endowment of machines with commonsense knowledge, able to reason as humans do.

1. Isaak, N., Michael, L.: Using the Winograd Schema Challenge as a CAPTCHA. In: Lee, D., Steen, A., Walsh, T. (eds.) GCAI-2018. 4th Global Conference on Artificial Intelligence. EPiC Series in Computing, vol. 55, pp. 93–106. EasyChair (2018). https://doi.org/10.29007/rnk8, https://easychair.org/publications/paper/pV9V

5

Metrics of Hardness to Differentiate Between Winograd Instances

5.1 Introduction

As we have seen in previous chapters, well-constructed Winograd schemas are easy for humans and hard for machines because they require the use of commonsense knowledge to answer them. Specifically, in every Winograd instance, we need to have the background knowledge that is not revealed in the words of the sentence to clarify what is going on (Levesque, 2014). Furthermore, it seems that not all Winograd instances are equally easy or hard for humans (Bender, 2015), and the task of predicting their hardness index is an interesting question. To the best of our knowledge, what we know about the perceived human hardness index on the WSC is based on Bender's work (Bender, 2015). Bender, through an experiment, identified that human adults tackle the WSC with a mean accuracy of 92%. According to Bender (2015), future challenges should be validated and organized according to how humans can tackle them. In this regard, certain people are unfamiliar with certain concepts in WSC halves, and their performance correlates with this familiarity.

The solution to this was to develop two systems that are predictive of the perceived human hardness when tackling Winograd instances, meaning that they could be used to group instances according to their perceived hardness indexes. Systems able to differentiate between Winograd instances could also be used to ensure that WSC-based CAPTCHA services would display more challenging instances to solve in the case of possible fraudulent actions. To the best of our knowledge, this is the first work to report the results of this approach's feasibility.

The first system is based on Wikisense, which was introduced in Chapter 3. Recall that Wikisense has demonstrated the plausibility of using commonsense knowledge automatically

acquired from *raw text* in English Wikipedia. In this regard, we will start by presenting the results of a large-scale experiment that shows how the performance of that particular automated approach varies with the availability of training material. We compare the Wikisense-based approach results with two studies, one from the literature investigating how adult native speakers tackle the WSC and one that we design and undertake to investigate how non-native teenager speakers tackle the WSC. According to our results, the automated approach's performance correlates positively with the performance of humans, suggesting that the performance of the particular automated approach could be used as a metric of hardness for WSC instances.

The second system has been developed in the strong sense that although machine learning techniques (e.g., deep learning) function without actually understanding the text they are processing, they are extremely good on correlation tasks (Marcus and Davis, 2019; Mitchell, 2019). It seems that systems can discover patterns of words or systematic bias in words to tackle various challenges without showing commonsense and reasoning abilities. Our empirical study shows that our new system, which is based on random forest classifier and deep learning (LSTM-based), is considerably faster and more accurate than any other previously used method. At the same time, along with our developed system, we extend Bender's work (Bender, 2015) by presenting the results of a large-scale experiment that shows how human performance varies across Winograd instances.

We want to point out that our developed systems do not purport to replicate the cognitive mechanisms used by humans when solving the WSC but only to offer a phenomenological account of this perceived hardness. We are not claiming, however, that this metric can be used to anticipate how hard it is for *machines* to resolve certain WSC instances, nor, by extension, that it can be used to select material for WSC competitions in order to test the progress of machines on the WSC.

Below we start by presenting our developed systems, followed by our analysis and discussion of our findings. The sections below explain each of these tasks, along with the tools and techniques we have developed.

5.2 The Wikisense-based Approach

5.2.1 Introduction

One potentially promising approach to handling the WSC builds natural language representations and supports the necessary reasoning with the available information by acquiring knowledge in general inference rules (see Chapter 3). Here, we present the results of a

large-scale experiment to see how this kind of approach can be used as a data-driven metric of hardness for WSC instances (halves). To that end, we reuse the Wikisense-based approach and compare its results with two studies: i) one from the literature (Bender, 2015) that investigates how adult native speakers tackle the WSC and ii) one that we design and undertake to investigate how non-native teenagers tackle the WSC. To the best of our knowledge, no study has focused on how the amount of training material for a learning-based approach to the WSC can be used as a data-driven metric of hardness for WSC instances, and any evidence for this has been mainly anecdotal.

In a study involving more than 400 adults, Bender (2015) showed that certain people are unfamiliar with certain concepts in WSC instances, and their performance ends up being correlated with this familiarity. In this sense, the use of our proposed metric, trained on appropriately selected training data, can be used to provide an a priori level of objective hardness of WSC instances so that the challenge can be personalized to the strengths and weaknesses of particular groups of *human* participants. Showing a positive correlation of the system's performance with the performance of humans would suffice to offer evidence that the system can be used to automatically differentiate between WSC instances based on their perceived hardness for humans.

The system considered in this work (called the Wikisense-based approach) effectively improves its behavior as it gets more training data. Since the WSC is claimed to require commonsense knowledge to be solved by humans, this might suggest that WSC instances that are harder for humans are the ones that require more training, and hence more effort to identify the right knowledge; or, put differently, harder instances are the ones that require the use of commonsense knowledge that is less common to found in written text, because it is usually implied. Our experiment supports this hypothesis by showing that the system's performance is correlated with human performance. In particular, adults asked to solve the WSC are shown to perform better than teenagers. Since age is generally correlated with more experiences, and thus the acquisition of knowledge that might be less common, this is in line with the above hypothesis.

5.2.2 How the Availability of Training Material Affects Performance in the WSC

The Wikisense-based approach focuses on Wikisense (see Chapter 3), which, unlike certain other WSC systems (see Chapter 2), has a particular online flavor. Wikisense first considers the WSC half at hand and then retrieves relevant material from the English Wikipedia to build its knowledge, or simply put, training material on which it is trained. It is, therefore,

straightforward to adapt the amount of training material that will be made available to the system and consider the effects of data availability on its performance.

Recall that, Wikisense creates multiple keyword-queries based on any given WSC half. For every query, in turn, Wikisense retrieves a number of sentences from the English Wikipedia that match the query, as specified by the system's parameter. Using those sentences as training material, Wikisense determines if it can conclude that one of the two answers of the WSC half can be inferred. If not, it attempts to use the subsequent query and repeats the process.

Wikisense utilizes dependency parsers to turn raw text into semantic relations. These relations act, in turn, as the features of learning examples from which inference rules are induced, following the approach of Michael and Valiant (2008). In case sufficiently confident rules are identified, those rules are used to draw inferences about the WSC half.

Below, we describe the knowledge enchantments that we performed to discover how training material affects the performance in the WSC and discuss certain choices made.

To evaluate how the size of the training corpus affects Wikisense's performance, we selected the first 100 WSC halves from the original WSC_286 dataset[1], and used the *Wikisense* system with 12 training set sizes ($1 \cdot 10^1$, $2 \cdot 10^1$, $5 \cdot 10^1$, $1 \cdot 10^2$, $2 \cdot 10^2$, $5 \cdot 10^2$, $1 \cdot 10^3$, $2 \cdot 10^3$, $5 \cdot 10^3$, $1 \cdot 10^4$, $2 \cdot 10^4$, $5 \cdot 10^4$). For each set size, we ran *Wikisense* 100 times (rounds) for statistically significant results. Each set determines the training sentences number, and at each time, we use randomly selected sentences; one training sentence can be of any length and can also be used multiple times. For every round, in every set, we store if that specific round successfully or incorrectly resolved each WSC half or if it left it unanswered (unresolved; see Figure 5.1). Below, we present the results that were obtained after several months of testing. For testing purposes and to reduce the time complexity factor, we ran Wikisense with only the first option (*Vx-Vy*), meaning that we did not use Wikisense's full potential as it would have taken us years of experiments to train it with twelve different training sets (see Chapter 3).

Hypotheses

For the purpose of this study, we formulated the following null hypothesis:

1. The size of the training set does not affect Wikisense's performance regarding the tackle of the WSC.

[1]http://www.cs.nyu.edu/faculty/davise/papers/OldSchemas.xml

	round 1	round 2	round 3	Correct	Wrong	Unanswered
s001	wrong	correct	correct	2	1	0
s002	correct	unanswered	unanswered	1	0	2
s003	unanswered	unanswered	unanswered	0	0	3
s004	unanswered	unanswered	unanswered	0	0	3

Correct	1	1	1
Wrong	1	0	0
Unanswered	2	3	3

Figure 5.1 A snapshot of Wikisense's results trained with the smallest training-set ($S = 1 \cdot 10^1$).

Materials

For the experiments, we have used a variety of hardware and software, and the whole running procedure took several months. Among other software, we have used the Wikisense system, a Python library for plots (matplotlib)[2], Stanford and spaCy parser, and a spreadsheet software to design and administer the experiments. In the end, we have run our experiments on five different systems:

- Apple MacBook Pro with Intel Core i7 2,4 GHz, 16 GB 1333 MHz DDR3, SSD.
- Apple Mac Pro 3.46GHz 12 Core Xeon Processor 5.1, 32GB RAM DDR3, SSD, HDD.
- Apple iMac with Intel i5 2.5 GHz, 20 GB DDR3, SSD.
- Asus Lamborghini with Intel Core i7 2.20 GHz, 16 GB 1333 MHz DDR3, SSD, HDD.
- Lenovo Thinkstation with two Quadro Xeon Processors 2.27 GHz, 20 GB 1333 MHz DDR3, SSD, HDD.

Results

Figure 5.2 compares the *correct*, *wrong*, and *unanswered* pronoun resolution; the horizontal axis shows the training set sizes, while the vertical axis depicts the *unanswered*, the *correct*, and the *wrong* resolution means. A cursory glance at Figure 5.2 reveals that as the size of the training set increases, the number of unanswered WSC halves decreases, while the numbers of both the correctly answered and incorrectly answered WSC halves increase, with the latter seemingly increasing at a lower rate. The null hypothesis that the size of the training set

[2]https://matplotlib.org

does not affect the performance of the Wikisense system in terms of the correct answers it produces can, therefore, be rejected using an ANOVA analysis that gives F = 20.860 > Fcrit = 3.2849, showing that the means of the three populations (correct, wrong, unanswered) are not equal. As shown in Figure 5.2, the number of unanswered WSC halves monotonically reduces as the training set size increases. The only exception to this monotonicity is when $S = 1 \cdot 10^4$, where we observe an increase of 0.36%, benefiting the number of correctly answered WSC halves. This is the point in the graph where the distance between the correctly answered and the incorrectly answered WSC halves is the largest (8%). Comparing the performance of the system when $S = 1 \cdot 10^3$, the default value used by Wikisense for the tackle of the WSC, to the system's performance when $S = 5 \cdot 10^4$, we can see a measurable increase of 5%, which suggests that the performance of Wikisense can be further improved with the simple adjustment of the training set.

Figure 5.3 shows the system's performance for each of the 100 rounds that were run for the two extreme values of *S*, demonstrating a consistent (not simply on average, but on each individual round) ability of the system to answer correctly more often than incorrectly when *S* is larger. As we can see, there is a significant difference of 22% between the largest set and the smallest set, on benefit of the correct pronoun resolution; we can clearly see the correct pronoun resolution line, depicted on higher values, in the largest set. As shown in Figure 5.3, the lines of the smallest set are mixed, contrary to the lines of the largest set, showing a bigger gap between the two lines in the largest set; with higher values for the *correct* pronoun resolution. Thus, not only do larger training sets lead to fewer unanswered WSC halves, but among those that are answered, the percentage of the correctly answered ones tends to become larger than the percentage of the incorrectly answered ones.

Overall, larger training sets seem to lead the Wikisense system to answer more WSC halves, and among those answered, to answer correctly more often. Given the knowledge-based workings of the Wikisense system, this could be taken as an indication that richer and more useful knowledge is acquired from larger training sets. Our results indicate only that more information helps *improve* performance, not that it suffices to achieve *human-level* performance. On the other hand, we will offer evidence in the sequel that even without achieving human-level performance, one can still use the Wikisense system to determine how hard a WSC half might be for humans.

Half-level Analysis

To better understand how the availability of training material affects the performance and why larger training set sizes offer better pronoun resolution, we proceed to a half-level analysis.

Wikisense failed to answer 92 WSC halves with the smallest Wikipedia set (WSC halves that remained unanswered $> 50\%$) and 46 with the biggest. It might seem counter-intuitive that the unresolved number is almost 50%. However, with the biggest set, we are able to resolve 14 more halves than with the default one ($1 \cdot 10^3$), which is a huge step regarding the challenge difficulties (Ackerman, 2016; Morgenstern et al., 2016).

Means of each WSC half, by each set, are presented in Figure 5.4, which shows the *correct*, *wrong*, and *unanswered* pronoun resolution. The horizontal axis shows the training set sizes, while the vertical axis highlights each tested WSC half. Each *correct* WSC half resolution is depicted with green color, for each set, while each half *wrong* resolution is depicted with red color; the blue color shows the *unanswered* resolution. Our sentence level analysis is consistent with previous results showing that the *unanswered* color (blue) gradually reduces, in each bigger set, while the *correct* color (green) increases.

Figures 5.5, 5.6, and 5.7 show the *unanswered*, the *correct*, and the *wrong* pronoun resolution through RGB colors; the WSC halves are reordered by the performance in the largest training set. For instance, a cursory glance at Figure 5.5 shows that the bigger training sets have the darkest color, meaning that the *unanswered* WSC halves number reduces as the set size grows. Also, there is a significant difference in the green color density across the horizontal axis of Figure 5.6; it shows that the *correct* pronoun resolution for bigger sets is better than in the smaller sets. Also, we can see how the *wrong* pronoun resolution changes, on the WSC half level, in Figure 5.7.

Regarding the WSC halves that remained unanswered in all rounds of all training sets, Wikisense could not create a keyword in some of them, while on others, it could not find enough or useful training sentences to resolve pronouns. For instance, for a keyword like "punish bully", instead of returning valuable training data, it returned sentences about a "Wooly Bully" song that did not relate to our initial keyword. Taken altogether, it seems that a large number of unanswered sentences relate to various factors like the quantity and the quality of our training data and to Wikisense's shortcoming in acquiring proper training sentences for each given generated keyword.

5.2.3 Human Performance on the WSC

Showing a positive correlation of the performance of the system with the performance of humans would suffice to offer evidence that the system can be used to automatically differentiate between WSC halves based on their perceived hardness for humans. In this regard, here, we present evidence from two studies in support of the claim that the performance of the Wikisense system *varies* across WSC halves in a manner analogous to the performance of humans. The first study comes from the literature and concerns adult native speakers,

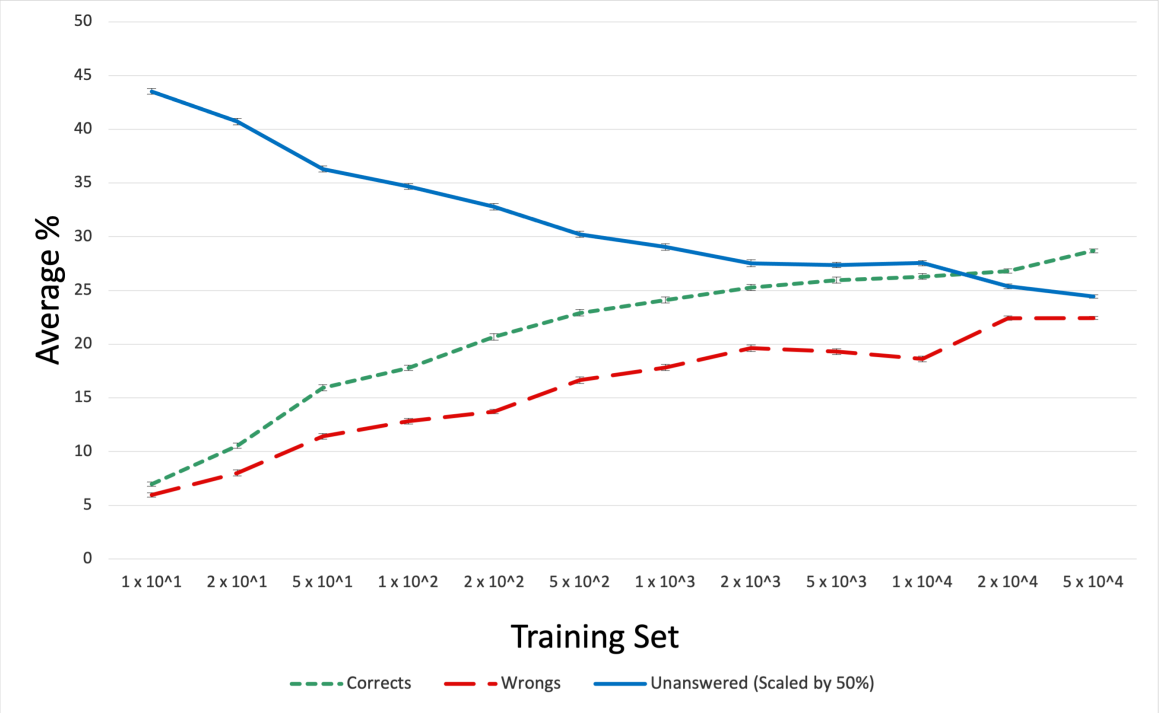


Figure 5.2 Performance evaluation along with standard-errors on the entire corpus across different values of S.

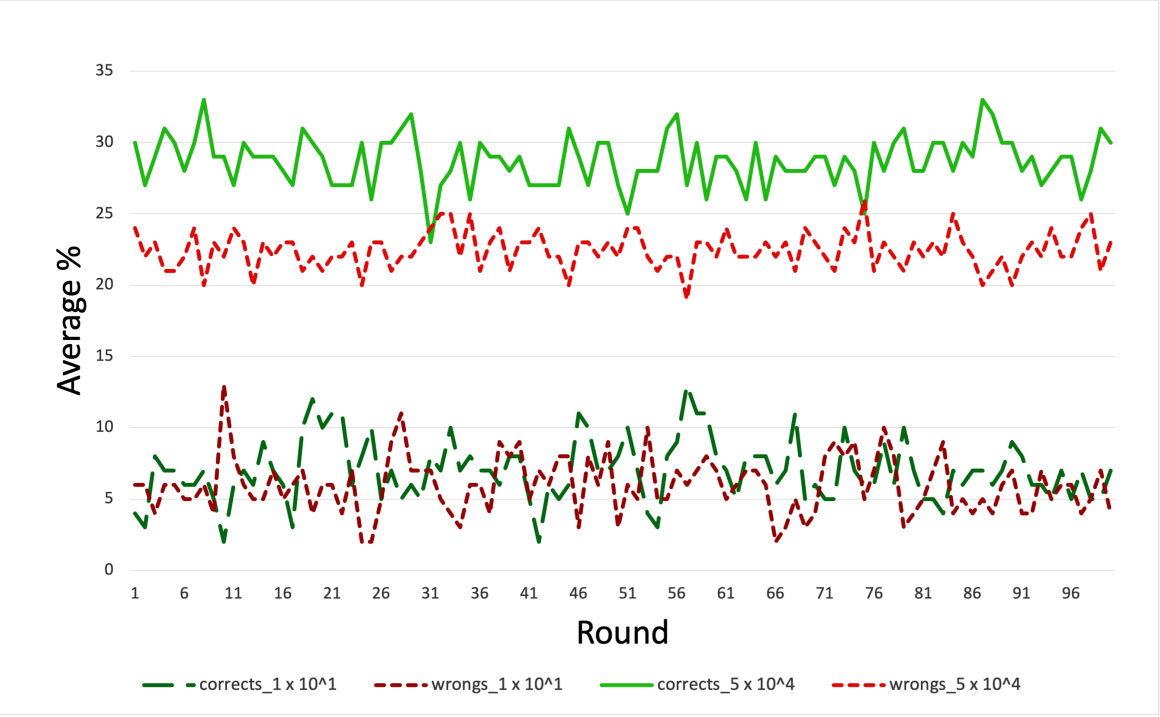


Figure 5.3 Percentages of the correctly and incorrectly answered WSC halves in each round. The plot shows these percentages for the two extreme values of S.

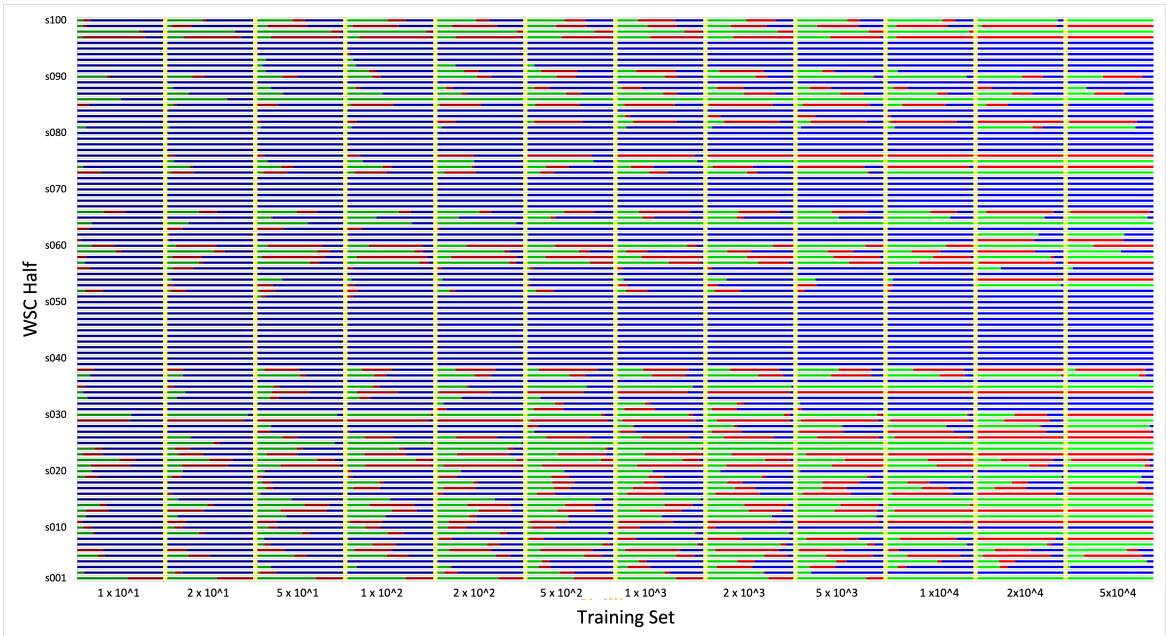


Figure 5.4 The coloring of each horizontal bar indicates the percentage of rounds in which each WSC half on the Y axis was correctly answered (green color), incorrectly answered (red color), or remained unanswered (blue color), for each value of *S* on the X axis.

while the second study was designed as part of this work and concerns teenager non-native speakers.

Adult Native Speaker Performance

According to Bender (2015), certain people are unfamiliar with certain concepts in WSC halves, and their performance ends up being correlated with this familiarity. In terms of the performance of humans on the WSC, Bender (2015), through an experiment he undertook, identified that human adults tackle the WSC with a mean accuracy of 92% —91% if we consider only the first 100 WSC halves (see Figure 5.8). To the best of our knowledge, this is the only set available to provide us with the necessary training and testing data. Regarding the required time needed to tackle a WSC half, it was found that adults need, on average, 15 seconds to answer a given schema. Bender used schemas developed by experts from the WSC_ dataset, which, at the time of writing, consisted of 143 schemas (WSC286). The experiment ran on Amazon's Mechanical Turk, where 407 adult speakers, who speak English fluently, participated. Results, which showed that adult speakers are, on average, able to correctly resolve 92% of the Winograd schemas, set the bar very high compared to what systems can achieve (Kocijan et al., 2020). On the other hand, in the experiments, it was shown that there are halves that are harder to resolve than others; for instance, there are

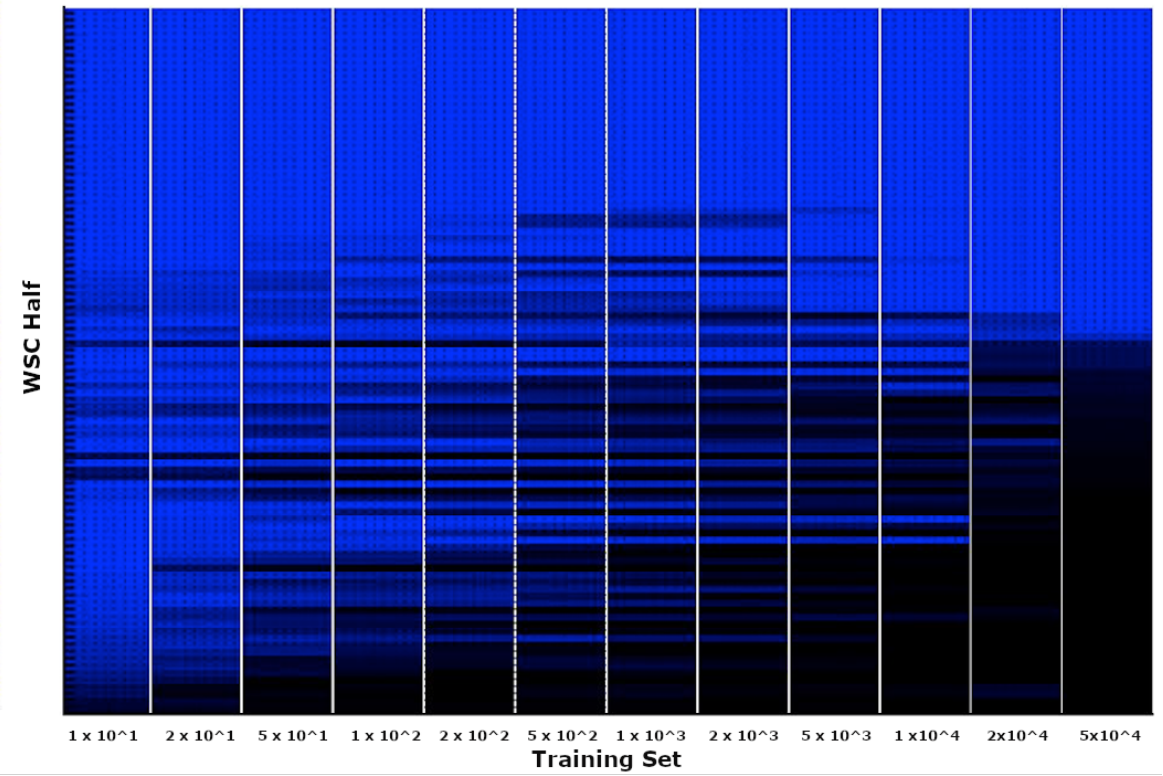


Figure 5.5 Color intensity shows how often (among 100 rounds) each WSC half on the Y axis has been answered (correctly or incorrectly), for each value of S on the X axis. The WSC halves on the Y axis have been reordered based on the percentage with which they have been answered when $S = 5 \cdot 10^4$.

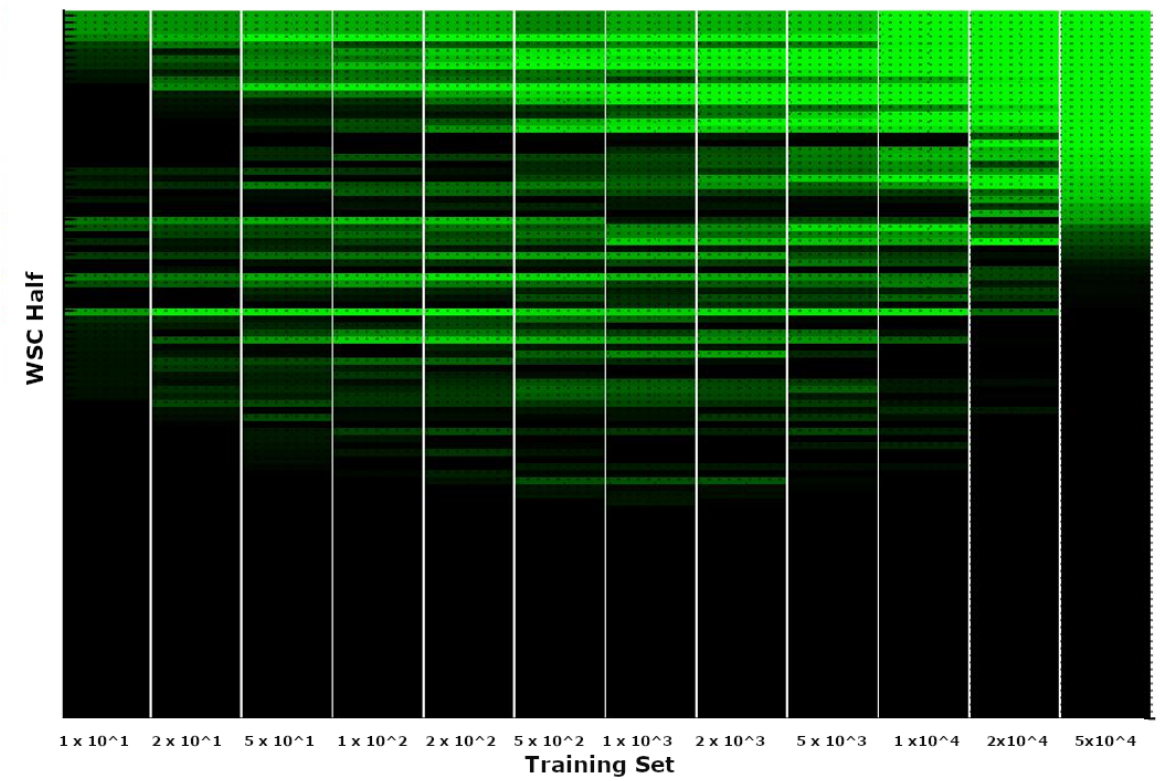


Figure 5.6 Color intensity shows how often (among 100 rounds) each WSC half on the Y axis has been correctly answered, for each value of S on the X axis. The WSC halves on the Y axis have been reordered based on the percentage with which they have been answered when $S = 5 \cdot 10^4$.

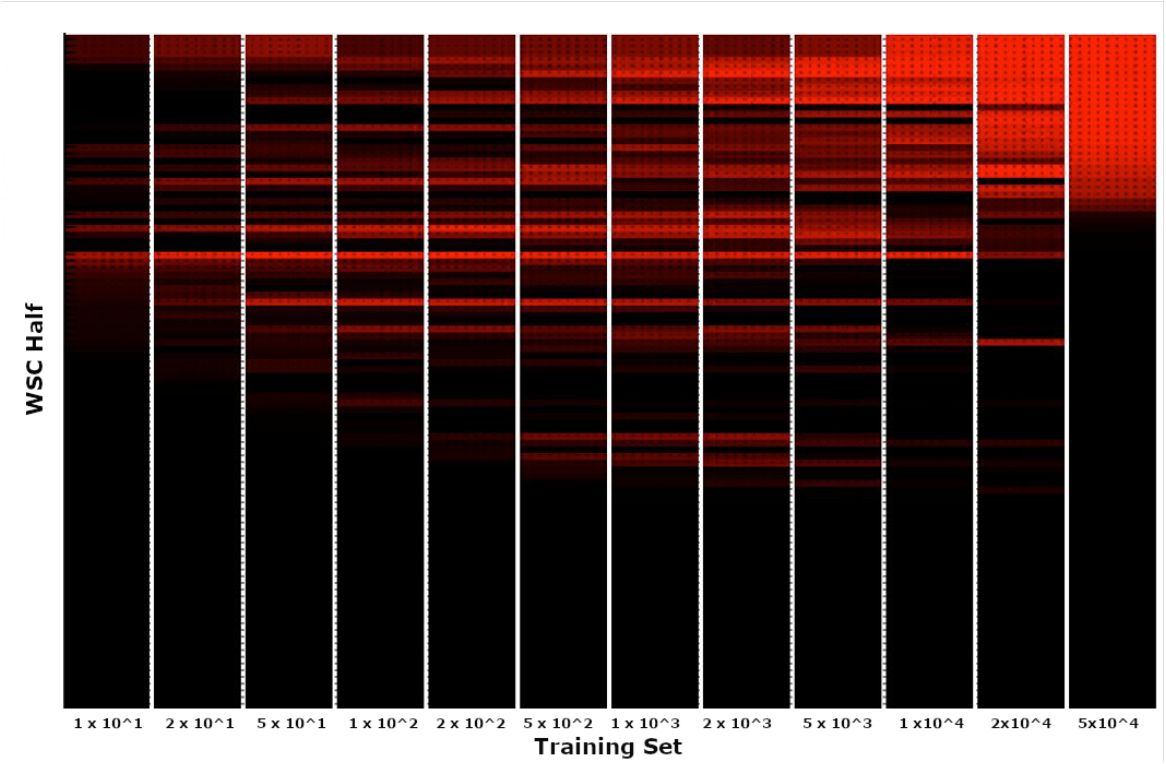


Figure 5.7 Color intensity shows how often (among 100 rounds) each WSC half on the Y axis has been wrongfully answered, for each value of S on the X axis. The WSC halves on the Y axis have been reordered based on the percentage with which they have been answered when $S = 5 \cdot 10^4$.

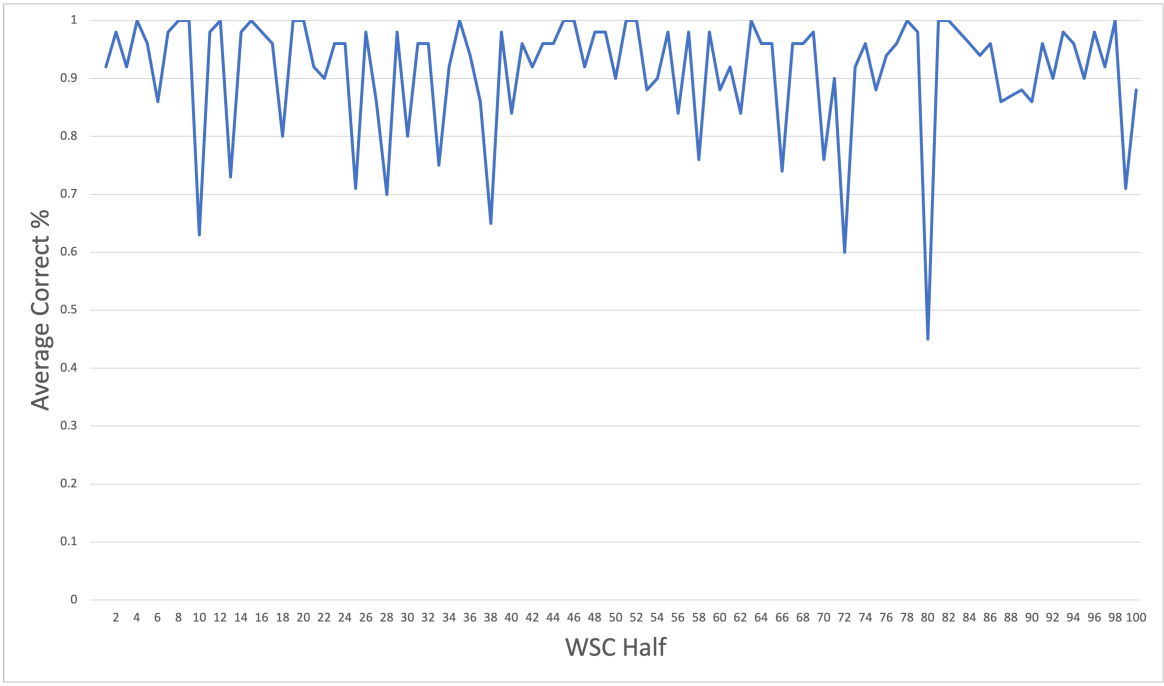


Figure 5.8 Bender adult accuracy scores on the 100 WSC halves used in our experiments.

halves on which humans scored a mean of 45%. A detailed analysis of human performance on each WSC instance is available online[3].

Teenager Non-Native Speaker Performance

Given that Bender's results refer to adult speakers of English, we undertook an analogous study to determine how teenager speakers tackle the WSC. To the best of our knowledge, no study has focused on how teenagers can tackle the WSC, and any evidence for this has been mainly anecdotal. The study, which was carried out in a lab setting, took part in December 2017, where 126 English-speaking students volunteered and participated. Participants were teenagers, residents of Cyprus who speak English fluently. All of them were students at a single 3-grade gymnasium school, and they were between 11 and 15 years old (see Table 5.1).

In terms of their knowledge of the English language, 37 reported that it was "good", 66 that it was "very good", and 23 that they speak English fluently (out of which nine mentioned that English is their mother tongue). All participants had experience with the WSC, as they had previously participated in another study that involved the WSC (although that study was in Greek).

[3]https://github.com/benderdave/wsc-exp.git

	Grade A	Grade B	Grade C
males	16	31	9
females	13	40	17
10-11	-	-	-
12-13	29	68	1
14-15	-	3	25

Table 5.1 Demographic of participants.

We split the 100 WSC sentences that were used in the evaluation of the *Wikisense* system into four equal sets, ensuring that no set included both halves from the same schema. Participants were asked to answer the questions of the WSC halves in one of the sets. The participation was anonymous, and it lasted about ten minutes during school break-time between lessons. The study was undertaken in the school's computer-science labs under supervision by a teacher. Each WSC half was displayed on a screen, followed by the question. Two choices were displayed side-by-side, with a comment box below each question. Access to translation services was not allowed, and each participant was instructed to write any remarks (on whether a question was confusing or non-intuitive) in a specified comment box. Participants were offered a €0.50 chocolate bar as compensation for their time.

Based on the study results (see Figure 5.9), teenagers scored a mean accuracy of 60.77% ($\sigma = 0.16$). The nine teenagers with English as their mother tongue scored a mean accuracy of 54.83%, indicating that the teenager group's lower performance compared to the adult group might not be a result of the teenagers being non-native speakers but a result of their age. Additionally, beyond the performance difference, teenagers and adults were found to have a positive correlation coefficient (Pearson) of 0.43. Although this does not indicate moderately correlated groups, given both the challenge difficulties and the difference in age between them, we considered it important in the sense that some halves that were easier or harder to answer by one group were also considered the same by the other group.

5.2.4 Experiments: Measuring the Hardness of WSC Halves

Here, using the data from the two studies above, we examine whether the performance of the Wikisense system can be predictive of the hardness of the WSC halves for humans. As a baseline, we compare the predictive ability of the system against that of other coreference resolution systems.

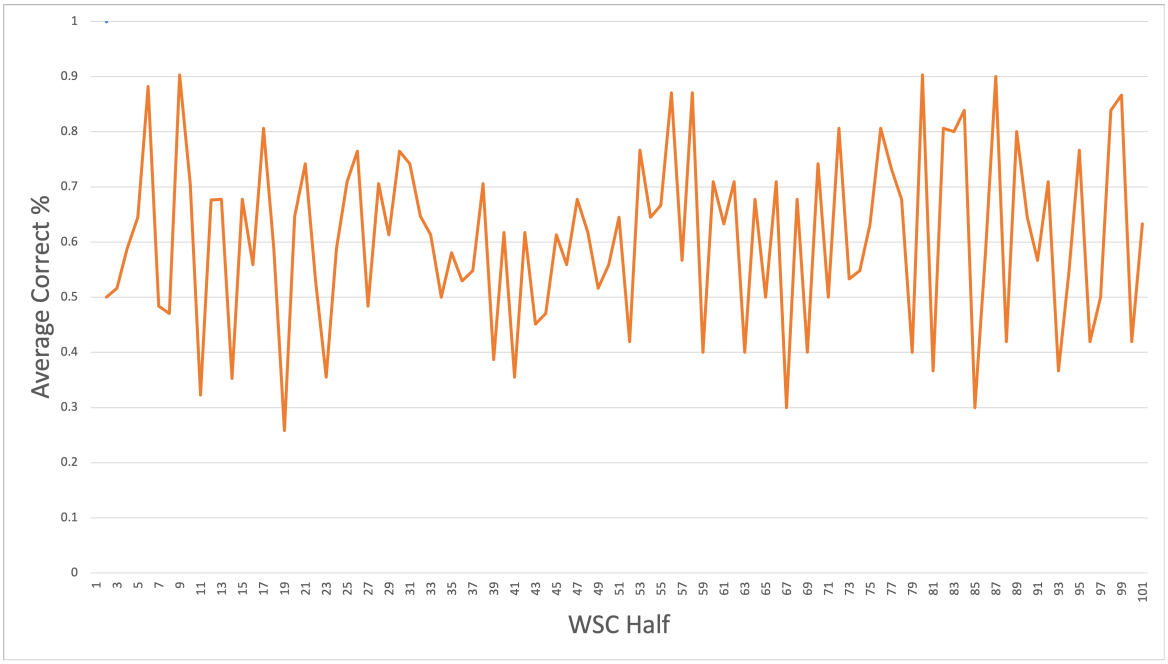


Figure 5.9 Teenager accuracy scores on the 100 WSC halves used in our experiments.

A Boolean Hardness Metric

We start with the simple approach of characterizing a WSC half as either "easy" or "hard", depending on whether it can be resolved correctly or incorrectly by an automated way, based on the Wikisense system.

For our Wikisense-based approach, we proceed as follows: For a given WSC half and a given value of a training-set S, we run the Wikisense system for 100 rounds and record the most frequent result returned by the system. Thus, we can determine if, most of the time, the system responded with the first answer, with the second answer, or abstained from responding. We repeat the process for all twelve possible values of S, as described in the preceding paragraphs. If the majority of these twelve repetitions yield the same answer, then we take that to be the approach's answer. We, then, check to see if the answer is correct or not, characterizing, respectively, the WSC half as "easy" or "hard"; some WSC halves remain uncharacterized.

To compare this boolean-hardness metric against what can be derived from other systems, we consider three coreference resolution systems from the literature. For each system, a WSC half is characterized as "easy" or "hard" (or remains uncharacterized) depending on whether the system can correctly or incorrectly resolve the WSC half (or does not produce an answer).

Based on WSC halves' characterizations by each of the four considered approaches, we group the WSC halves into an "easy" and a "hard" group and compare humans' performance on these two groups to see whether their performance varies. The results are summarized in Table 5.2, which shows that the boolean hardness metric derived from the *Wikisense*-based approach can discriminate better between what humans find "easy" and "hard" in the WSC. In particular:

1. Wikisense-Based Approach: It can be inferred from Table 5.2 that, the WSC halves characterized as "easy" and "hard" can be resolved by adults with a mean accuracy of 93% ($\sigma = 0.08$) and 87% ($\sigma = 0.12$), respectively, compared to their overall mean accuracy of 91%. Analogously, the WSC halves characterized as "easy" and "hard" can be resolved by teenagers with a mean accuracy of 66% ($\sigma = 0.16$) and 57% ($\sigma = 0.17$), respectively, compared to their overall mean accuracy of 60.77%.

2. Stanford CoreNLP (Manning et al., 2014)[4]: The WSC halves characterized as "easy" and "hard" can be resolved by adults with a mean accuracy of 90% ($\sigma = 0.12$) and 93% ($\sigma = 0.08$), respectively, showing a negative correlation with the human performance (see Table 5.2). The same phenomenon appears with teenagers, where the WSC halves characterized as "easy" and "hard" can be resolved with a mean accuracy of 60% ($\sigma = 0.14$) and 62% ($\sigma = 0.17$), respectively.

3. Illinois Co-reference Resolver (Bengtson and Roth, 2008; Peng et al., 2015)[5]: The WSC halves characterized as "easy" and "hard" can be resolved by adults with a mean accuracy of 93% ($\sigma = 0.07$) and 91% ($\sigma = 0.10$), respectively, showing a smaller discriminatory power than our proposed approach (see Table 5.2). This is even more evident with teenagers, where the WSC halves characterized as "easy" and "hard" can be resolved with a mean accuracy of 62% ($\sigma = 0.16$) and 61% ($\sigma = 0.14$), respectively.

4. Knowledge Parser (K-Parser) (Sharma et al., 2015)[6]: This system was built for the WSC, yet its performance seems to be non-predictive of human performance (see Table 5.2). The WSC halves characterized as "easy" and "hard" can be resolved by adults with a mean accuracy of 89% ($\sigma = 0.13$) and 93% ($\sigma = 0.08$), respectively. Analogously, the WSC halves characterized as "easy" and "hard" can be resolved by teenagers with a mean accuracy of 57% ($\sigma = 0.14$) to 62% ($\sigma = 0.16$), respectively, showing an important gap in the wrong direction.

[4]http://nlp.stanford.edu:8080/corenlp/process
[5]https://cogcomp.org/page/demo_view/Coref
[6]www.kparser.org

	adults		teenagers	
	"easy"	"hard"	"easy"	"hard"
Stanford CoreNLP	0.90	0.93	0.60	0.62
Illinois Coref.	0.93	0.91	0.62	0.61
K-Parser	0.89	0.93	0.57	0.62
Wikisense-based	0.93	0.87	0.66	0.57

Table 5.2 Predictive behavior of human performance from simple boolean hardness metrics derived from automated systems.

The results ultimately show that the performance of the *Wikisense*-based approach *varies* across WSC halves in a manner that resembles the variability of the human performance more closely than what other systems can achieve.

A Real-Valued Hardness Metric

As afforded by the Wikisense system's access to training material and aiming to derive a more fine-grained hardness metric, we then consider a particular way of deriving a real-valued hardness index for each WSC half, called the Wikisense-based approach (see Figure 5.10).

As within the boolean hardness metric, given a WSC half and a value of S, we run Wikisense for 100 rounds and record the most frequent result returned by the system (see the top-left part of Figure 5.10). Thus, we can determine if, most of the time, the system responded with the first answer, with the second answer, or abstained from responding. For each case where the response is one of the two answers, we check and mark the answer as correct or incorrect. We repeat the process for all twelve possible values of S and end up with a set of twelve labels (see the bottom and top-middle part of Figure 5.10). Intuitively, if all of these labels are "unanswered", we do not have enough information to give a hardness index to the examined half. This particular approach ends up giving a hardness index to 57 out of the 100 WSC halves under consideration, and our subsequent discussion refers to only these 57 instances.

Now, consider the case where at least one label is "correct", and therefore, the system has identified, at least once, knowledge that is relevant to, and *appropriate* for, the particular WSC half. The more "correct" labels one has, the easier it would seem that this WSC half is. Considering that out of the cases with an "unanswered" label, one could randomly guess the correct answer half of the time, we can adjust the number of "correct" labels to include half of the "unanswered" labels. Normalizing this value by dividing by twelve, we end up with a number in the interval $[0, 1]$ that is higher for easier WSC halves. Taking one minus this value gives us the hardness index of the WSC half. The whole procedure is handled by the

Controller component of the Wikisense-based approach (see the bottom-right part of Figure 5.10).

If none of the labels is "correct", and since we compute a hardness index only if there is at least one label that is not "unanswered", it must be the case that there exists at least one "incorrect" label. Therefore, the system has identified, at least once, a knowledge that is relevant to, but *inappropriate* for, the particular WSC half. One could argue that the more "incorrect" labels one has, the harder this WSC half should be. However, given the simple approach that the Wikisense system follows in retrieving relevant training data, one could also make another argument. Since there are *no* "correct" labels, the more "incorrect" labels one has should simply be taken as an indication of the availability of more relevant knowledge, ignoring the fact that it led to the wrong answer. The availability of knowledge suggests, then, an easier WSC half. Considering that out of the cases with an "unanswered" label, one could randomly guess the incorrect answer half of the time, we can adjust the number of "incorrect" labels also to include half of the "unanswered" labels. Normalizing this value by dividing by twelve, we end up with a number in the interval $[0, 1]$ that is higher for easier WSC halves. Taking one minus this value gives us the hardness index of the WSC half.

In terms of the human performance data, we treat the human hardness index of a WSC half to be the percentage of people from a certain group that resolved the half incorrectly. Our computed hardness index and the human hardness index for the adult and the teenager groups in our discussed studies have correlation coefficients of 0.38 and 0.37, respectively. Both results offer evidence that our proposed computed hardness index might indicate how humans perceive the hardness of the WSC, and that this indication might not be significantly affected across different human groups.

Figure 5.11 shows in more detail how the computed hardness index and the human hardness index vary across WSC halves, suggesting that certain WSC halves that are more easy or hard for humans are accordingly labeled as such by the computed hardness index. The figure also shows that despite the teenager group performing almost consistently worse than the adult group, their performance across WSC halves seems to vary analogously. Furthermore, on this specific subset of schemas, the two groups had a positive correlation of 0.49, 6% higher than their initial correlation of 0.43, suggesting that human participants can more closely answer schemas on which relevant knowledge was found on Wikipedia.

The Wikisense-based system, a python-written package that takes as input a WSC half and outputs its hardness index, is available online[7] to download. The system can adjust the conditions under which it chooses to produce a hardness index or abstain from producing

[7]http://cognition.ouc.ac.cy/ws_hardness

one. For instance, if the parameters are appropriately adjusted to compute a hardness index for only 10% of the tested WSC halves, the correlation coefficient against the teenager group becomes 0.70.

5.2.5 Qualitative Analysis

Based on the teenage participants' remarks in our study, we present a qualitative analysis that relates those remarks to the performance of the Wikisense-based approach.

Unanswered WSC halves

Twenty-seven WSC halves remained *unanswered* in all rounds across all training sets. For instance, the half *I couldn't put the pot on the shelf because it was too tall. Question: What was too tall?* was accompanied by a remark that it was very confusing; the mean adult accuracy was 45%, and the mean teenager accuracy was 37%. Another example is the half *Frank was upset with Tom because the toaster he had bought from him didn't work. Question: Who had bought the toaster?*, which was accompanied by a remark that it was very difficult; the mean adult accuracy was 75%, and the mean teenager accuracy was 50%. Likewise, on the half *Pete envies Martin although he is very successful. Question: Who is very successful?* the mean adult accuracy was 84%, and the mean teenager accuracy was 35%. Additionally, the half *The lawyer asked the witness a question, but he was reluctant to repeat it. Question: Who was reluctant to repeat the question?* was accompanied by a remark on not understanding the meaning of "reluctant"; the mean adult accuracy was 63%, and the mean teenager accuracy was 32%.

Unformulated Queries

Wikisense was not able to formulate a query to retrieve training data in four of the previously mentioned twenty-seven halves. In some other cases, despite formulating a query (e.g., lie/cautious), the system could not retrieve enough training data to create the necessary knowledge. For instance, the half *The cat was lying by the mouse hole waiting for the mouse, but it was too cautious. What was too cautious?* was accompanied by a remark on not understanding its meaning; the mean adult accuracy was 90%, and the mean teenager accuracy was 42%. Another example is the half *In the middle of the outdoor concert, the rain started falling, but it continued until 10. Question: What continued until 10?*, which was accompanied by the "an interesting half" remark; the mean adult accuracy was 60%, and the mean teenager accuracy was 53%.

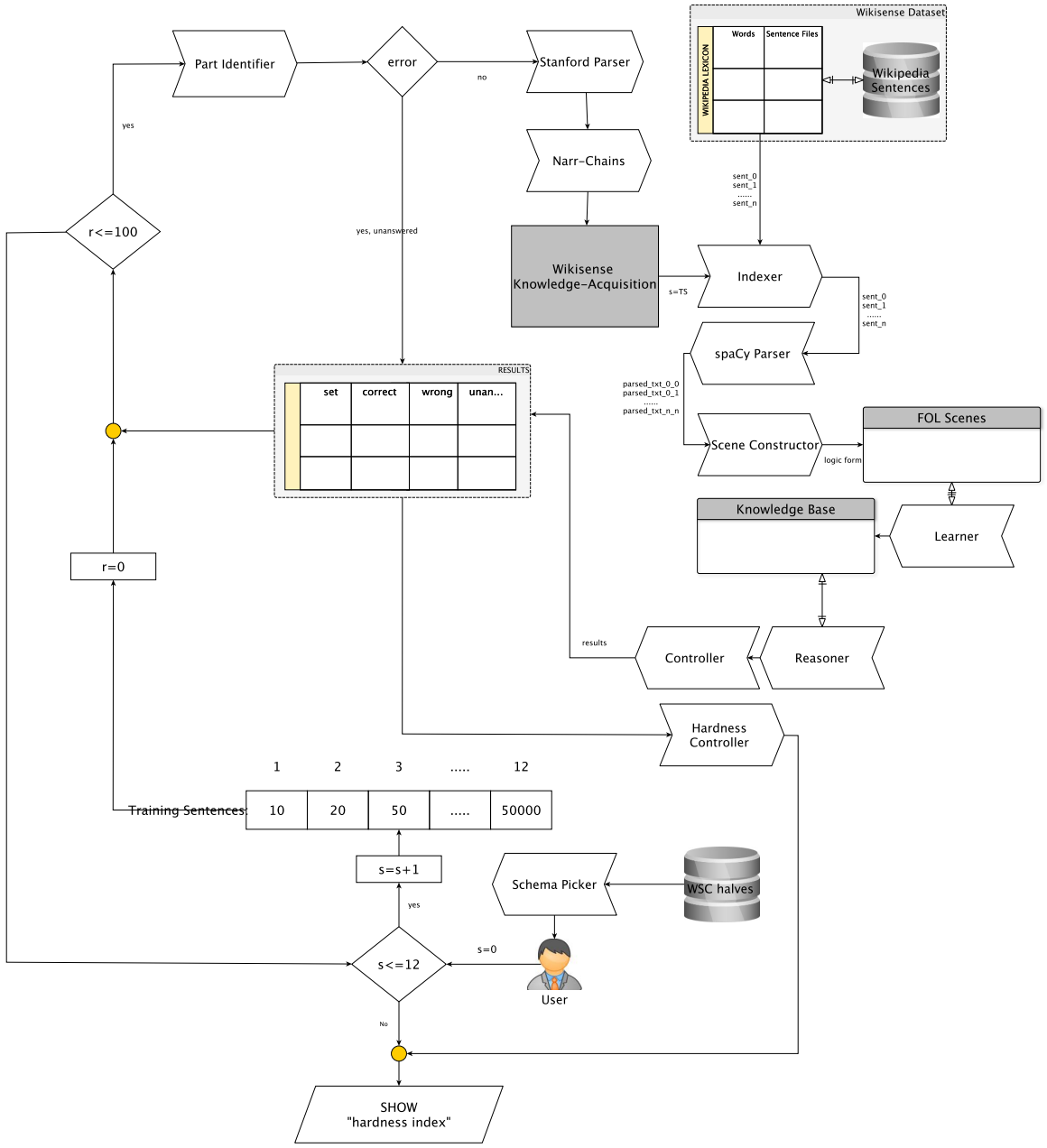


Figure 5.10 The Wikisense-based approach. A system able to differentiate between Winograd halves according to their perceived hardness for humans.

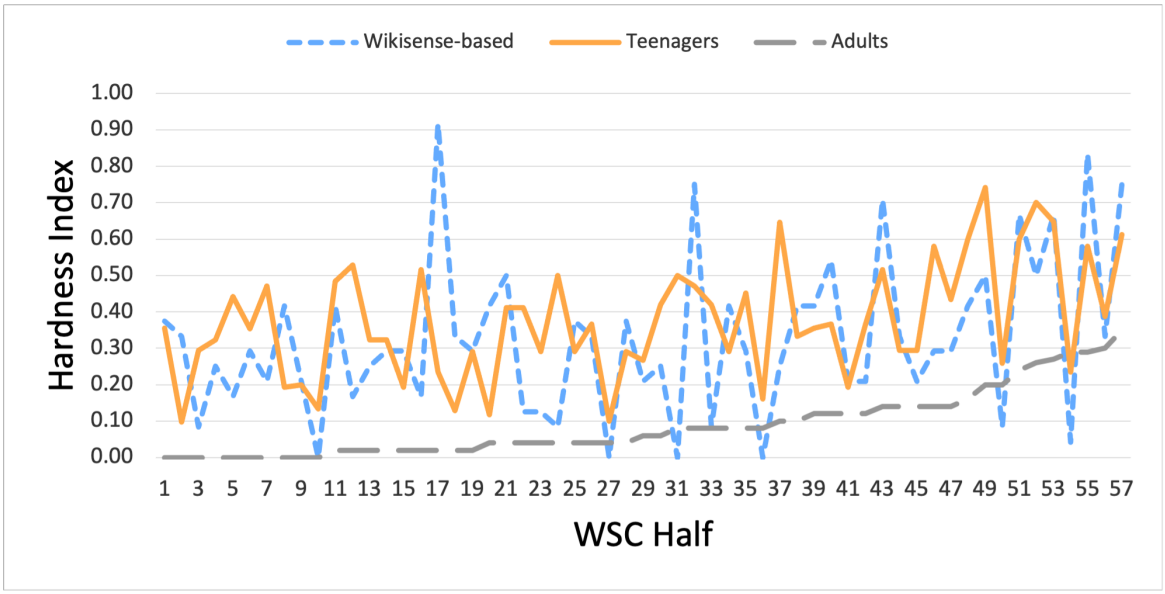


Figure 5.11 Variability of our developed Wikisense-based hardness index across the 57 WSC halves on which it was computed, in relation to the variability of the human hardness index for adults and teenagers. The results are sorted by the accuracy of adults, where smaller hardness indexes show easier halves to resolve.

Correctly-Resolved WSC halves

Three WSC halves were correctly resolved across all training sets: i) *The city councilmen refused the demonstrators a permit because they feared violence. Question: Who feared violence?*, ii) *Bob paid for Charlie's college education, but now Charlie acts as though it never happened. He is very ungrateful. Question: Who is ungrateful?*, iii) *Anne gave birth to a daughter last month. She is a very charming baby. Question: Who is a charming baby?*

Wikisense resolved the first half through the query refuse/fear. It might be considered as an easy half because the subject of the verb "refuse" is the one who fears that something is going to happen. In this regard, the query directly leads to the correct pronoun target. However, fifty percent of the teenagers did not manage to resolve the pronoun correctly, compared to only 8% of adults.

Two teenagers commented that they found it very difficult, with one specifying that they did not know the meaning of the word "councilmen". No remarks were received on the second and third halves. In the second half, the mean teenager accuracy was 90%, and the mean adult accuracy was 96%. Finally, in the third half, the mean teenager accuracy was 87% and the mean adult accuracy was 100%.

There were halves that the system could resolve correctly only when the size of the training set was sufficiently large. For example, the half *Jim yelled at Kevin because he was*

so upset. Question: Who was upset? was correctly resolved only with the two largest training set sizes. Three teenagers commented that the half was difficult; the mean teenager accuracy was 53%, and the mean adult accuracy was 100%. As another example, the half *Paul tried to call George on the phone, but he wasn't successful. Question: Who was not successful?* was correctly resolved from the fourth training set size onwards; the mean teenager accuracy was 43%, and the mean adult accuracy was 98%. Finally, the half *There is a gap in the wall. You can see the garden behind it. Question: You can see the garden behind what?* only 40% of the teenagers managed to resolve it, compared to 85% of the adults.

Incorrectly-Resolved WSC halves

We observed that some queries led the system to wrong conclusions. For instance, the half *Anne gave birth to a daughter last month. She is a very charming woman. Question: Who is a charming woman?* was wrongfully resolved across all training set sizes. The query give/charming ended-up producing more training data supporting the inference *daughter*, as there seem to be more training sentences for charming children than for charming adults. On the other hand, humans do not typically refer to a female newborn as a woman; the mean teenager accuracy was 84%, and the mean adult accuracy was 92%.

The half *Alice tried frantically to stop her daughter from chatting at the party, leaving us to wonder why she was behaving so strangely. Question: Who was behaving strangely?* was accompanied by the remark that it was odd; the mean teenager accuracy was 40%, and the mean adult accuracy was 71%.

There were also WSC halves that, although they were correctly resolved with smaller training sets, were incorrectly resolved with larger training sets. For instance, the half *Tom threw his schoolbag down to Ray after he reached the bottom of the stairs. Question: Who reached the bottom of the stairs?* was correctly resolved only until the ninth training set. Teenagers correctly resolved the sentence 35% of the time, while adults 90% of the time.

Confusing WSC halves

For certain WSC halves, there was no obvious relation between the system's performance and the training set size. For instance, in the WSC half *Frank felt vindicated when his longtime rival Bill revealed that he was the winner of the competition. Question: Who was the winner of the competition?* the performance of the system across the twelve training set sizes started with not producing an answer and flipped back and forth between producing the right and the wrong answers as the training set sizes increased. It seems that such halves

might be confusing even for humans; the mean teenager accuracy was 35%, and the mean adult accuracy was 73%. A teenager characterized it as a complicated example.

As an additional example, the mean accuracy on the half *The sack of potatoes had been placed below the bag of flour, so it had to be moved first. Question: What had to be moved first?* was 35% for teenagers and 69% for adults, whereas the mean accuracy on the sentence *My meeting started at 4:00 and I needed to catch the train at 4:30, so there wasn't much time. Luckily, it was delayed so it worked out. Question: What was delayed?* was 30% for teenagers and 74% for adults, with two teenagers remarking that it was very confusing.

Extreme Cases

Certain WSC halves are characterized as extreme cases related to the system's performance and the training set size. While in the beginning, with small training set sizes, they were either correctly or wrongfully resolved, with the largest training size the complete quite the opposite happened. An extreme case is the following half "Sentence: Anna did a lot worse than her good friend Lucy on the test because she had studied so hard. Question: Who studied hard? Answers: Anna, Lucy", which was correctly answered early on with small training set sizes but incorrectly answered with the largest training set size. Another example is the half "Sentence: Tom threw his schoolbag down to Ray after he reached the bottom of the stairs. Question: Who reached the bottom of the stairs? Answers: Tom, Ray", which was correctly answered early on with small training set sizes but incorrectly answered with the last three training set sizes. Regarding the human factor, the mean teenager accuracy was 74% on the first and 35% on the second half, while the mean adult accuracy was 80% on the first and 90% on the second, respectively. As mentioned before, it seems that the acquired training sentences can easily flip back and forth between producing the right and the wrong answers. However, one could argue that since the largest training set is bigger than all the other training sets together by 11120 sentences, this should not be a surprise. Like we discussed before, it seems that Wikisense results are directly dependable on factors like the quantity and the quality of our training data and in our engine's shortcoming in acquiring proper training sentences for specific keywords.

We also analyzed the relationship between the keyword parts, based on their part of speech, in halves that remained unanswered and in halves that were correctly resolved in all training sets (> 50%). The results yielded some interesting findings. The halves that were correctly answered had a verb in the left part of the keyword in 95% of the cases. The right part of the keyword was a noun or an adjective in 66% of the cases and 34% a verb. On the other hand, the unanswered halves had a verb in the left part of the keyword 67% of the time and 33% a preposition. The right part of the keyword was in 63% a noun or an adjective and

37% a verb. Another interesting side finding was that if we eliminate the unanswered halves in those that remained unanswered by 100%, in all sets, the right part of the keyword is in 41%, a verb. These findings would suggest that, if the left keyword part ——which indirectly connects the two possible pronoun targets—— is a verb, and the right keyword part, ——which connects the question with the sentence—— is an adjective or a noun, then we have better possibilities in acquiring better knowledge from our training data.

Schema Issues

In analyzing the behavior of Wikisense on pairs of halves (schemas), we have observed that it was never the case that both halves of the same schema were correctly resolved. We speculate that this happens because the simple form of queries that we have used in the context of this work effectively missed the minor differences between pairs of halves, giving rise to the same query for both sentences. This directly points to an opportunity to further improve the system's performance by creating more nuanced queries. If this improvement ends up yielding a worse metric of hardness, this might be an indication that humans might also, to some extent, ignore parts of a WSC half that might be critical in its correct resolution.

Another observation worth reporting is that the mean accuracy of teenagers, when tested on the first half of a schema versus their mean accuracy when tested on the second half of the same schema, has a gap of 20% ($\sigma = 0.15$), suggesting that there might be schemas that do not include halves of the same hardness. However, given that the two halves were randomly displayed, the teenagers might be biased toward assuming the correct referent based on their position, whether positioned first or second in the sentence.

Associative Sentences

Trichelair et al. (2018) have identified that 37[8] of the sentences in the WSC_273 library can be solved using simple statistics over patterns. Specifically, this subset of the original WSC sentences is labeled as associative sentences, meaning sentences in which one candidate antecedent associates strongly with the clause containing the pronoun, while the other candidate antecedent exhibits no such association strength. Based on our analysis, eight associative sentences were found in our testing set, where, in two of them, the Wikisense-based approach was unable to resolve the definite pronoun in all training sets (unanswered). For those specific halves, the human adult score for the first one was 96%, and the teenager score was 77% —*It was a summer afternoon, and the dog was sitting in the middle of the*

[8]https://github.com/ptrichel/How-Reasonable-are-Common-Sense-Reasoning-Tasks/blob/master/WSC_associative_label.json

lawn. After a while, it got up and moved to a spot under the tree, because it was cooler. Question: What was cooler? However, for the second half, the human adult score was 45%, and the teenager score was 37%, indicating not an easy half to answer —*I couldn't put the pot on the shelf because it was too tall. Question: What was too tall?* For the rest six halves, the Wikisense-based approach was able to tackle them with an average of 60% ($\sigma = 0.11$) while at the same time adults can tackle them with an average of 93% ($\sigma = 0.07$) and the teenagers with an average of 68% ($\sigma = 0.15$). The results align with how the Wikisense-based approach acquires its knowledge, meaning that our system, maybe the teenagers too, does not seem to exploit the factor that some Winograd instances can be solved using simple statistics over patterns.

Sentence Structure

It seems that the structure and semantics of WSC schemas that are based on simple sentences are not that easily detectable. On the other hand, one could argue that complex sentences have a syntactic structure that might be easier for humans to answer. In an attempt to analyze the linguistic characteristics of those WSC instances, we observed the following: Regarding the teenager group, we have identified that their lowest accuracy was on simple sentences though this was a subset of only two halves that were part of the same schema —"Sentence: I spread the cloth on the table in order to protect/display it. Question: To protect/display what? Answers: The table, the cloth". Specifically, we detected an average teenager accuracy of 54%. Next, we observed that their average score on compound-based halves was 60%, on complex-based halves 61%, and finally on compound/complex-based halves 65%.

Regarding the adult group, we observed that their average score on compound/complex-based halves was 90%, on compound-based halves 91%, on complex-based halves 92%, and finally on simple-based halves 96%. Results show that adults did not find the simple-based halves as challenging as teenagers but found the compound/complex-based halves harder than other types of halves.

Regarding the Wikisense-based approach, we observed that it could not tackle the simple-based halves as they remained unanswered in all training set sizes. We have found that its average score on compound-based halves was 75%, on complex-based halves 77%, and finally on compound/complex-based halves 87%.

Taken altogether, on a hardness scale from one to four, where the lower, the more challenging to resolve (simple, compound, compound/complex, complex), it seems that teenagers and adults "ranked" the compound-based halves and the complex-based halves at the same positions —second and third, respectively. However, the results should be taken

with a grain of salt as our testing halves were not equivalently developed based on the given sentence types.

5.3 The WinoReg Approach

5.3.1 Introduction

In our previous section, we have seen the Wikisense-based hardness-metric, which could be used in future challenges or in the WSC CAPTCHA service to differentiate between Winograd halves, albeit with limitations regarding the number of schemas it could be applied on and the time needed for the whole process, which was found to be time-consuming. According to our results, the resulting model was able to offer the hardness index on only 57% of our tested halves, which is in direct relation with the keyword implementation of Wikisense that is based on the semantic analysis of the given halves. Additionally, because of its dependency on training during query-answering, it was found that the Wikisense-based approach needs, on average, eight hours to output the hardness index of a given half, which is disproportional to the potential use of the Wikisense-based approach to differentiate between Winograd halves according to their perceived hardness for humans.

To find a faster and more accurate way to output the hardness index of any Winograd half, in this section, we consider a new-novel system, called WinoReg (from Winograd Regression), which is based on machine learning (ML). Although machine learning is not like human learning (Wooldridge, 2020), it is excellent at making predictions about data, which in our case seems probable. Through experience and prediction, WinoReg learns how to compute the hardness of a given half based on two different approaches, i) the Random-Forest approach, which directly relates to feature engineering, and ii) the LSTM-based approach that does not relate to feature engineering but requires access to the hardness index of more Winograd halves.

Within both approaches, WinoReg proceeds by first training the regression model and then using the learned model for faster computation during its deployment. Regarding the feature engineering of the Random-Forest approach, these features come from several works in the literature that have developed WSC-related systems, which we have re-implemented as needed (Budukh, 2013; Peng et al., 2015; Rahman and Ng, 2012; Sharma et al., 2015). Regarding the LSTM-based model and its need for more training data, we extended Bender's work (Bender, 2015) with a study that we designed and undertook, which involved 306 crowdsourced workers and 943 halves.

5.3.2 WinoReg's High-level Architecture

Figure 5.12 depicts WinoReg's high-level architecture. WinoReg works in two operational modes: the random-forest and the LSTM-based mode. In both modes, it outputs the hardness of any WSC half through regression analysis. Specifically, it examines the relationship between the halves and the perceived human hardness indexes (Bender, 2015).

After the training, WinoReg uses the learned model for faster computation during its deployment. Within the Random-Forest approach, WinoReg analyzes each half to output a required number of features. Next, all features are given as an input to the learned model to output the hardness of a given half. On the other hand, within the LSTM-based approach, WinoReg does not require estimating the values of features, meaning that any given half can be given directly to the model to acquire its hardness index. In both cases, WinoReg can load a half from a schema database to output its hardness index, which, like within the Wikisense-based approach, is a value in the range of 0-1. Compared to the Wikisense-based approach, no half is discarded.

Next, we will show how WinoReg works based on the approaches above. Specifically, in the first part, we will discuss how the engine estimates the values of features to build the random forest model, and, in the second part, we will show how deep learning comes into play.

5.3.3 WinoReg_RF: A Random-Forest Approach

Within this approach, WinoReg is based on training a regression model with the use of Decision Trees —called WinoReg_RF. We use, in particular, the random forest algorithm, which was introduced in 2001 (Breiman, 2001). The random forest algorithm, which involves constructing an ensemble of Decision Trees, each trained on random subsets of the data, showed significant improvements in the accuracy of different problems (Breiman, 2001). A recent research line showed that it is one of the best algorithms that maintain high imputation performance on linear regression across a range of performance metrics (Suresh et al., 2019). Like any other machine learning algorithm, the random forest algorithm focuses on forming rules with reasonable accuracy, which could be used to predict future data (Probst et al., 2019). In this regard, we aim to train a model using the random forest algorithm to estimate the perceived human hardness index of Winograd halves (see Figure 5.13).

According to François (2017), within machine learning, we need to transform our data to find the appropriate representations to make it more manageable to the task at hand. Given that we want to estimate the hardness index of any half, which indirectly relates to selecting the correct answer, our WinoReg_RF approach expects features related to the half parts,

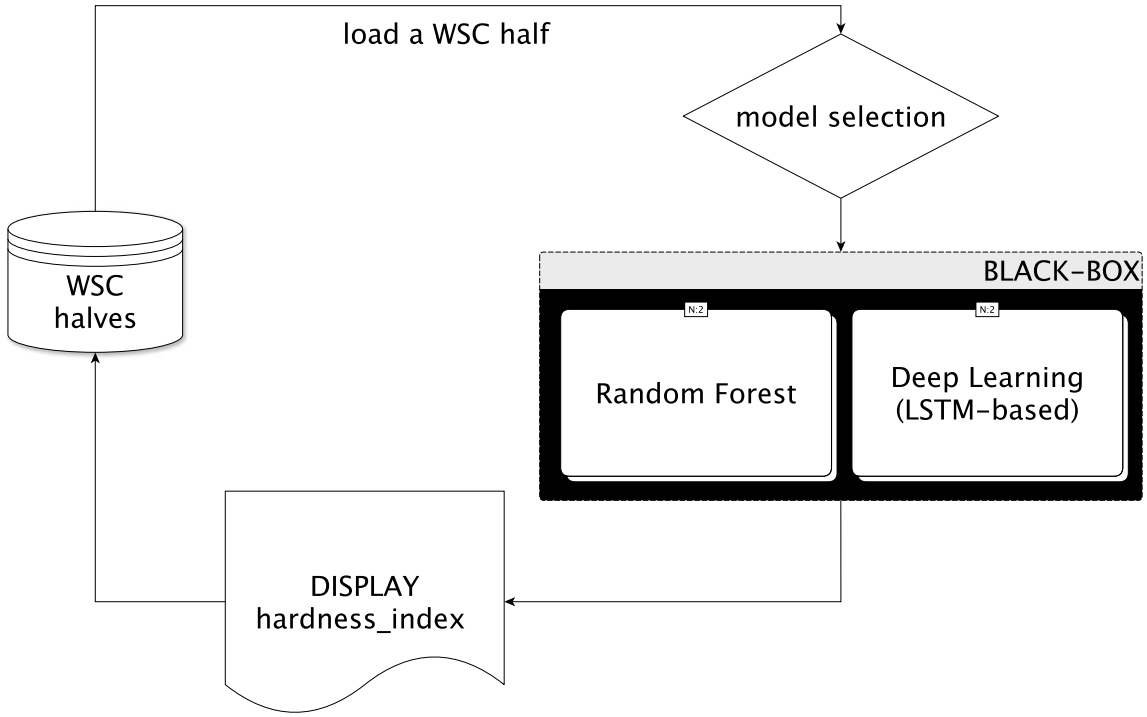


Figure 5.12 WinoReg's Architecture. The black-box shows that the system can output the perceived human hardness index based on two distinct modes, the Random-Forest and the LSTM-based mode.

namely, the sentence, the question, and the two pronoun targets (candidates). Compared to the Wikisense-based approach, WinoReg does not make use of the correct answer of each half. To train WinoReg_RF, we engineer fifty features from twelve components (see Figure 5.13). Most features are based on non-open-source systems previously built to tackle the WSC (Budukh, 2013; Peng et al., 2015; Rahman and Ng, 2012).

WinoReg_RF utilizes the *spaCy*[9] dependency parser to turn raw text into semantic relations. These relations act, in turn, as the basic feed for the feature development process. We use *spaCy*, like in the previous chapters, to output various relations between the sentence, the question, and the two pronoun targets to use them in our feature engineering.

According to Sharma et al. (2015), the semantic relations are considered reasonable if they can express the structure of the text and can differentiate, at the same time, between the events and their participants. In this regard, via spaCy, we can output relations that show how the pronoun targets relate to the definite pronoun and the events in which they participate.

[9]https://spacy.io
spaCy's statistical model: en_core_web_sm

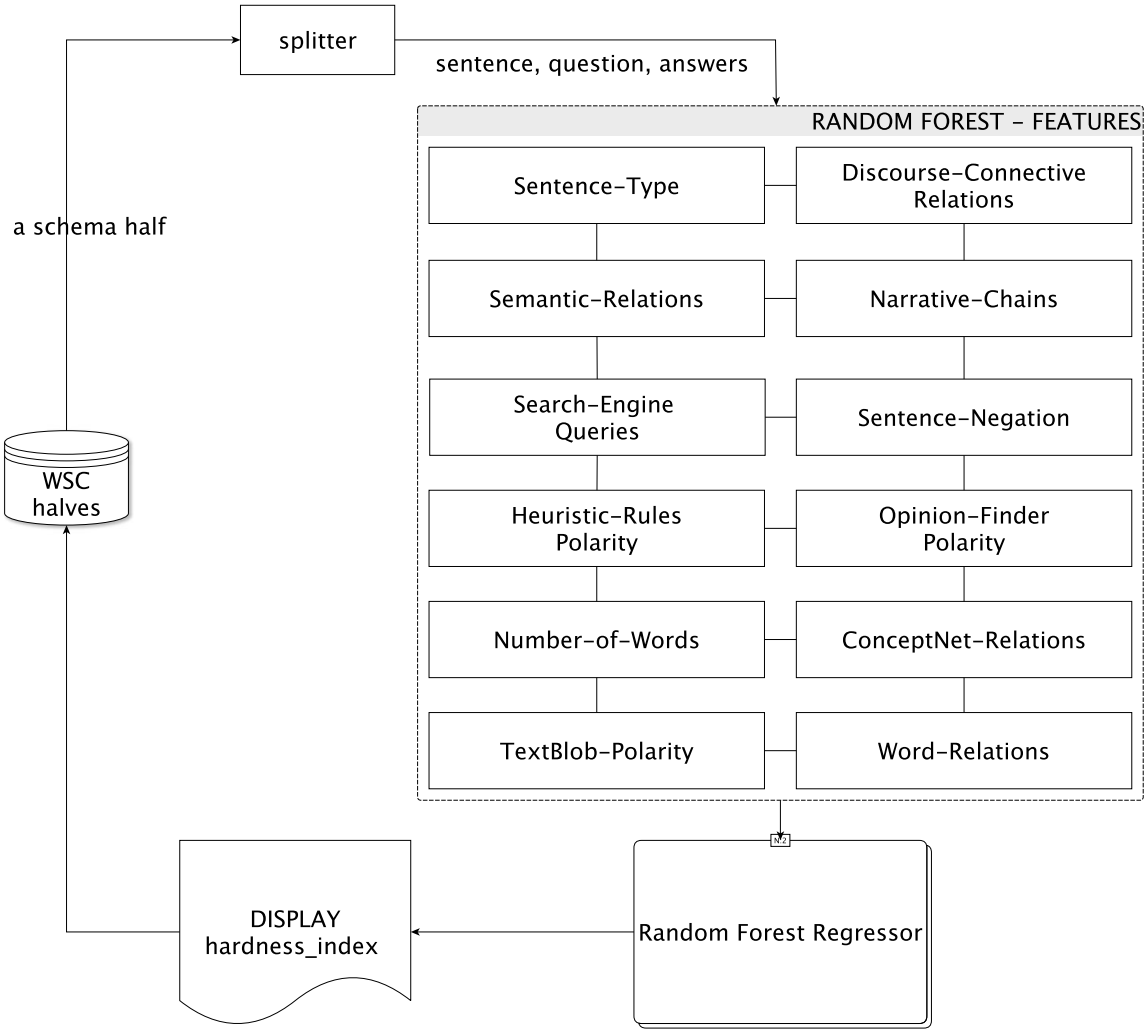


Figure 5.13 WinoReg's Architecture based on random forest: Given a Winograd half, WinoReg outputs the perceived human hardness index.

For instance, consider the *catch* example we saw in Chapter 3: *Sentence: The cat caught the mouse because it was clever. Question: Who is clever? Answers: The cat, The mouse*. Via spaCy, we can output three semantic relations, which tell us that "a cat caught a mouse", and "something/someone is clever":

[cat-noun, caught-verb, mouse-noun]

[it-pronoun, was-aux-verb, clever-adj]

[was-aux_verb, caught-verb]

A more thorough analysis of the feature development process is given in the following paragraphs, where we describe each component in detail.

Sentence-Type

Humans use abstract syntax to organize and build sentences in new and creative ways, but for an AI system to understand the meaning of simple sentences is a complicated process (Adger, 2019). In this regard, each half sentence's structure plays a vital role in its quality and difficulty, where sentences with complex structures seem harder to resolve. To that end, we use a tool that we designed to output each examined half's sentence-type (called Sentence-Structure Identifier).

Given an English sentence, the Sentence-Structure Identifier outputs its pattern/type, which can be either a simple, a compound, a complex, or a compound-complex sentence (this is depicted in Figure 5.14). Simple sentences have only one independent clause (SV; where S=Subject and V=Verb), while compound sentences can have two or more independent clauses (e.g., "SV and SV"). On the other hand, complex sentences can have one independent clause plus one or more dependent clauses (e.g., "SV because SV"), and compound-complex sentences can have two or more independent clauses plus one or more dependent clauses (e.g., "SV and SV because SV"). The connector in each complex sentence shows how the dependent clause relates to the independent clause. Based on the typical connectors found in Winograd schemas, we consider the following groupings of connectors for our analysis: *i)* Cause/Effect: because, since, so that; *ii)* Comparison/Contrast: although, even though, though; *iii)* Place/Manner: where, how, however; *iv)* Possibility/Conditions: if, whether, unless; *v)* Relation: that, which, who; *vi)* Time: after, as, before.

Within this component, we engineer two features, namely ST and SP, containing the string values of the sentence-type and pattern.

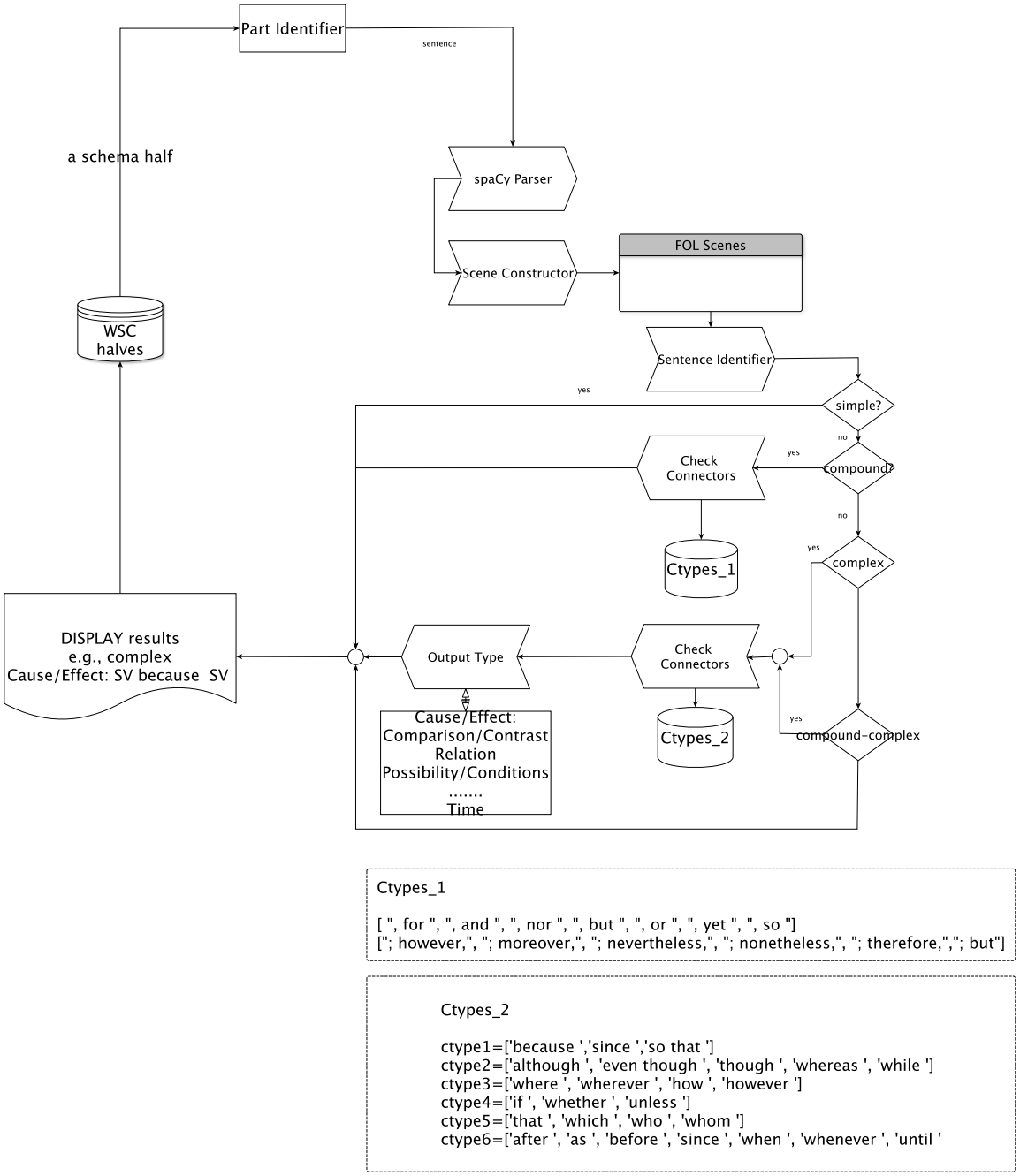


Figure 5.14 The Sentence-Structure Identifier component: Given a Winograd half, it outputs the sentence's pattern/type which can be either a simple, a compound, a complex, or a compound-complex sentence.

Sentence-Negation

Negation plays an important role in capturing the semantics of text, as it is used to reverse the polarity of parts of a statement (Blanco and Moldovan, 2011; Rahman and Ng, 2012). Moreover, humans master negation in their early stages of life (Adger, 2019). To encompass these kinds of rules, we analyze each half to estimate if the two pronoun targets and the definite pronoun are governed by negation. This is done via the sentence and question triples of each examined half (see the *catch* example). For the sentence-negation component, we create two binary features (*STN* for the two pronoun targets, *QTN* for the definite pronoun) that contain the value of 1 if negation exists and the value of 0 if it does not.

Narrative-Chains

According to Budukh (2013); Rahman and Ng (2012), narrative chains provide us with the story containing an event-based description of the participation of a common central actor called the protagonist. Basically, narrative chains are a sequence of events, in a story, with the role of the protagonist or the actor denoted as -*s*: subject or -*o*: object. To build our features, we start with spaCy to output the subject and the object and continue with Chambers and Jurafsky's (size 12) narrative chains (Chambers and Jurafsky, 2008), which are ordered sets of 12 events (verbs) centered around a common protagonist.

Specifically, for any given half, we determine the events the two pronoun targets and the definite pronoun participate in along with their protagonist role (subject or object). For instance, in the following half: *Sentence: The city councilmen refused the demonstrators a permit because they advocated violence. Question: Who advocated violence? Answers: The city councilmen, The demonstrators*, via Wikisense mechanisms, we output two triples: i) refused (x-subject, y-object), ii) advocate (they-subject, violence-object), in which we want to determine the protagonist of the *refuse-?* event, that participates in the advocate event as a subject (the definite pronoun —they— indicates the subject position).

Next, from Chambers and Jurafsky's narrative chains, and for each such pair, we extract all the chains that contain both elements (refuse and advocate). In our example, the Chambers and Jurafsky's narrative chain contains *refuse-o and advocate-s*, meaning that the protagonist in this chain is the object of a refuse event and the subject of an advocate event (the demonstrators). If WinoReg_RF cannot find narrative chains containing both elements, it runs the same procedure again, but with a similarity mechanism enabled. In previous works, the algorithm gives out no decision if there is no narrative chain matching between each event participated by each pronoun target with each event participated by the definite pronoun [(refuse, advocate-s)].

In the end, for the narrative chains component, we create a feature (NCH) that equals 1 if the answer is the first pronoun target, 2 if it is the second pronoun target, and -1 if we cannot output triples or find any narrative chains.

Number-of-Words

As with the sentence-type component, it seems that each half's sentence length directly relates to the resolution of the definite pronoun, where WSC halves with longer sentences tend to be harder to resolve. According to Marcus and Davis (2019), when we humans read a sentence in less than a second, we immediately parse and reconstruct it into its constituent noun and verbs to understand its meaning. In this regard, longer sentences increase the possibility to contain more nouns and verbs than shorter ones. Of course, in the end, we can sense a structure in every sentence, which helps us to understand its meaning (Adger, 2019). Following our findings, we engineer a feature that directly relates to the length of each sentence in terms of words (called SL).

Semantic-Relations

This component directly relates to the semantic relations of a given half's sentence. According to Rahman and Ng (2012), by using queries to search Web engines, we might encounter precision and recall problems related to the way these Web engines utilize their resources to return results —e.g., sentences might be returned based on the position of the words of the given query. Specifically, when a pronoun target and a verb appear next to each other, it does not mean that a subject-verb relation exists between them (a problem of precision). On the other hand, these queries fail to obtain subject-verb relations where a pronoun target and verb are not close to each other (a problem of recall). To eliminate these kinds of problems, we search Wikisense's Wikipedia-corpus (see Chapter 3) to see how many times each pronoun target appears as a subject or object. If the definite pronoun appears as a subject in a triple relation, we search to find which pronoun target appears as a subject most of the time; Otherwise, if the pronoun appears as an object, we search to find which pronoun target appears most of the time as an object. From the semantic-relation component, we create a single feature SEM, which equals 1 if the definite pronoun has the same role as the first pronoun target, 2 if it has the same role as the second pronoun target, and -1 if we cannot determine their roles.

Word-Relations

Word-relations component relates to candidate-independent, and candidate-dependent relations, where, according to previous works (Rahman and Ng, 2012), they seem to play an important role in the tackle of the WSC. The only catch is that they can only be applied in sentences that contain a connective (Cn) word (e.g., because). In this regard, for the candidate-independent features, we create two features (WN, WP), where WN refers to the number of words in each sentence (except the two candidates and the Cn), and WP refers to the number of word pairs. Those are pairs of words appearing before Cn, with each word appearing after Cn, excluding adjective-noun pairs, noun-adjective pairs, and the two candidates. For instance, for the sentence "The city councilmen refused the demonstrators a permit because they feared violence", the WP feature equals 24 and contains pairs like "city-feared", "city-violence", and "councilmen-feared".

For the candidate-dependent features, we engineer three features, namely HN, VF, and AF. Specifically, HN contains the number of the two candidates' headwords that were returned by the dependency parser; if we cannot determine the two candidates in the half's sentence, then the HN feature is set to 0. Subsequently, the VF feature contains the number of the verbs, and JF the number of the adjectives that modify the two candidates.

Search-Engine Queries

Work from the literature has shown that search-engine queries can provide us with world knowledge, which is useful for the tackle of the challenge (Peng et al., 2015; Rahman and Ng, 2012; Sharma et al., 2015).

Consider the *catch* example we saw earlier (Chapter 3, 3.2): *Sentence: The cat caught the mouse because it was clever. Question: Who is clever? Answers: The cat, The mouse*. In this example, we can acquire world knowledge to learn that someone clever can easily catch other things, which leads us to resolve the definite pronoun to the *cat*. In this regard, as other works have shown, we follow a similar approach to build features that are based on search queries. For every half, we build six queries, namely QR1: A1VQ, QR2: A2VQ, QR3: A1VQW, QR4: A2VQW, QR5: JA1, QR6: JA2; A1 and A2 are the two pronoun targets, VQ the question verb that governs the definite pronoun, W the sequence of words following VQ in the question, and J the question adjective that follows a verb-to-be. For instance, for the *catch* example, we generate and search the Google search engine with the following queries: (QR1) "cat was"; (QR2) "mouse was"; (QR3) "cat was clever"; (QR4) "mouse was clever"; (QR5) "clever cat"; and (QR6) "clever mouse". Next, as in Rahman and Ng (2012), using

the number of hits that the search engine returned, we built eight binary features, namely, GL1i1, GL1i2, GL2i1, GL2i2, GL3i1, GL3i2, GL4i1, GL4i2.

The first two features are computed from QR1 and QR2 (GL1i1, GL1i2), the next two from QR3 and QR4 (GL2i1, GL2i2), and the third from QR5 and QR6 (GL3i1, GL3i2). Finally, the last two features are computed based on the results returned from all queries (GL4i1, GL4i2). For instance, if the absolute value of |QR1, QR2|, is bigger than the threshold of 20% (*th*) in favor of the first pronoun target, then GL1i1 equals 1 and GL1i2 equals 0. Otherwise, if the opposite exists, then GL1i1 is set to 0 and GL1i2 to 1. To estimate the other features, we follow a similar approach; more details about the procedure can be found in the paper where it was initially introduced (Rahman and Ng, 2012).

Recent experiments with the GPT3 language-model (Brown et al., 2020) have shown potential contamination in their training set while tackling the WSC or other similar tasks. Put simply, this relates with text found in the WWW, which contains WSC schemas or similar discussions that might help relevant models or specific search engines find cues they were not supposed to find. Although this is a challenging task that needs to be examined further when designing benchmarks and when training models (Brown et al., 2020), both the way the search queries are constructed and the defined threshold of 20% help avoid problems that relate to potential contamination.

To avoid problems with proper-names (persons) where we cannot retrieve search query hints, we use FrameNet (Baker et al., 1998). As stated in other works (Budukh, 2013; Rahman and Ng, 2012), it is unlikely that search engines will return meaningful counts for persons. In this regard, in halves where the pronoun targets are proper names, we search FrameNet to find and substitute them with their roles. Specifically, for every triple relation, we search FrameNet for NP.EXT and NP.OBJ relations, where, NP.EXT shows the subjects and NP.OBJ the objects of the corresponding event (for instance, in the *catch*, if instead of a cat and mouse we had persons, then we would search FrameNet for the event catch). In case of a successful search from FrameNet, we replace the persons with their FrameNet roles. Consequently, we form six queries to search the Google search engine, and, using the number of hits that the search engine returned, we generate eight features: GLF1i1, GLF1i2, GLF2i1, GLF2i2, GLF3i1, GLF3i2, GLF4i1, GLF4i2.

ConceptNet-Relations

ConceptNet is a freely available semantic commonsense toolkit (Liu and Singh, 2004). Put simply, ConceptNet is an approach to collecting commonsense knowledge from crowdsourcing and has been around the field since 1999. Its knowledge-base is a semantic network, where nodes are the concepts and edges the relations among them. It is like a parser that

describes and expresses general human knowledge from sentences that were automatically acquired from the Open-mind Common-Sense project (Liu and Singh, 2004; Singh et al., 2002; Speer et al., 2017).

ConceptNet contains concepts about common basic knowledge about various facts, connected with other facts, using different kinds of relations (e.g., *relatedTo, AtLocation, IsA, PartOf*) (Budukh, 2013). WinoReg_RF makes use of ConceptNet to find possible relations between the two pronoun targets and the word —verb, adjective— that governs the definite pronoun; this is done by a ConceptNet function that returns a value in the range of 0-1, where the higher the value, the higher the relatedness is. In this regard, we engineer a feature (called CN) that equals 1 if the first pronoun target's relatedness value is greater than the second pronoun target's value. If the opposite exists, then the value of CN equals 2, and if we cannot find any difference, it equals -1. Additionally, like before, we consider FrameNet (Baker et al., 1998) for issues with proper names and create the CNF feature, where its values are being computed in the same way as the CN values.

Discourse-Connective Relations

According to Rahman and Ng (2012), causal relations, which are signaled by discourse connectives, show the world knowledge between events. Consider the following half: *Sentence: The lion eat the zebra because it was hungry. Question: Who is hungry? Answers: The lion, The zebra*. In the half's sentence, "The lion eat the zebra because it was hungry", there is a causal relation, which is given by the discourse connective "because", between the events "eat" and "hungry"; this *causal* relation helps us resolve the definite pronoun "it" to the lion.

Put simply, for each half, we search Wikisense's Wikipedia corpus for a triple of the form (V, Cn, X) and count its frequencies of occurrence. Cn is a discourse connective, V is a verb in the clause that governs the two pronoun targets, and X is a stemmed verb or an adjective that governs the definite pronoun. Each triple has to be validated through the following procedure: i) we search the Wikipedia corpus to find its frequencies of occurrence; ii) if the number of occurrences is at least 100, then we proceed to the next step (Rahman and Ng, 2012); iii) if X is a verb, then we resolve the definite pronoun to the pronoun target that shares the same role; otherwise, if the sentence does not involve comparison and X is an adjective, we resolve the definite pronoun to the pronoun target that serves as the subject of V. Finally, to encode this heuristic decision, we create a binary feature (CNT). CNT equals 1 if the definite pronoun is resolved to the first pronoun target and 2 if it is resolved to the second pronoun target. Otherwise, in case we cannot resolve the definite pronoun, CNT equals -1.

Event-Polarity via Heuristic Rules

As stated by Budukh (2013); Peng et al. (2015); Rahman and Ng (2012) there are halves where we can resolve the definite pronoun by comparing the two pronoun targets according to their polarity values. This process refers to word polarity, which has been widely studied in the NLP field (Hassan and Radev, 2010) and can be summarized in three steps: i) find the polarity of the definite pronoun; ii) determine the polarity of the two pronoun targets; iii) select the pronoun target that has the same polarity as the definite pronoun. The polarity of the definite pronoun equals the polarity value of the verb for which the pronoun serves as the subject, or, in case the verb does not exist, the polarity of the adjective that modifies it. To find the polarity values, we use the Wilson et al. (2005b) subjectivity-lexicon, a lexicon that assigns various events their polarity as negative, positive, or neutral.

Let us use the following half to explain the word-polarity procedure: *Sentence: The city councilmen refused the demonstrators a permit because they advocated violence. Question: Who advocated violence? Answers: The city councilmen, The demonstrators*. According to the half's semantic relations, we know the following:

city-councilmen is the subject of the event *refuse*
demonstrators is the object of the event *refuse*
they is the subject of the event *advocate*.

From the Wilson et al. (2005b) subjectivity-lexicon, we acquire the polarity of the *refuse* event, which is negative. In this regard, the polarity of the deep subject (*city councilmen*) becomes negative, and the polarity of the object becomes positive (*demonstrators*). Additionally, we know that the polarity of the event *advocate* in the subjectivity-lexicon is positive, meaning the polarity of the definite pronoun *they*, which participates in the subject of the event *advocate*, becomes positive. Consequently, we can conclude that the polarity of both the definite pronoun and the *demonstrators* is the same, which leads us to resolve the definite pronoun —they— to *demonstrators*.

Based on the event-polarity procedure, we engineer six binary features, namely, RP1i1, RP1i2, RP2i1, RP2i2, RP3i1, RP3i2, which are initially set to zero. The first two features, RP1i1 and RP1i2, refer to the correct pronoun target, where, in our example, are set to RP1i1=0 and RP1i2=1 (since the correct pronoun target —demonstrators— is the second one). The two other features (RP2i1 and RP2i2) are the concatenation of the polarity values, determined for both the definite pronoun and the two pronoun targets; in our example, RP2i1=negative-positive, and RP2i2=positive-positive.

To estimate RP3i1 and RP3i2, we simply take the previous features of RP2i1 and RP1i2 and append, if exists, the polarity reversing connective, such as *although*, which is a connective that flips the polarity (Rahman and Ng, 2012). Specifically, if a polarity reversing

connective exists, we simply take RP2i1 and RP2I2 and append the connective. For instance, RP3i1 = RP2i1 + connective, RP3i2 = RP2i2 + connective. Furthermore, we enhanced the polarity features by creating an additional feature (RPTL) that shows the best pronoun target. To that end, we take the first two binary features (RP1i1, RP1i2) and generate a new one (RPTL). If RP1i1 $>$ RP1i2, then the value of RPTL equals 1, and, otherwise, if the opposite exists, the value of RPTL equals 2. If we cannot determine RP1i1 and RP1i2, then RPTL is set to -1.

Event-Polarity via OpinionFinder

OpinionFinder is a machine-based sentiment-analyzer (Wilson et al., 2005a) that we use to resolve the definite pronouns in our examined halves. OpinionFinder automatically assigns various events their polarity as negative, positive, or neutral. Put simply, instead of using the heuristic rules, via OpinionFinder, we automatically acquire the polarity values of both the two pronoun targets and the definite pronoun (Peng et al., 2015; Rahman and Ng, 2012). To that end, we compute the OpinionFinder polarity features in the same way we did within the heuristic-rules component and create seven features (OP1i1, OP1i2, OP2i1, OP2i2, OP3i1, OP3i2, OPTL).

Event-Polarity via TextBlob

Given that our previous polarity features are based on similar approaches, namely, the Wilson et al. (2005b) subjectivity-lexicon, and the Wilson et al. (2005a) OpinionFinder, here, we use another, simpler polarity mechanism —called TextBlob-Polarity[10]. TextBlob (Loria, 2018) is an NLP library that can process textual data and output, among others, the events' polarity values. Specifically, with TextBlob's sentiment analysis, we return the verb's polarity that governs the two pronoun targets and the verb's polarity that governs the definite pronoun. Finally, we create two features (TBSPOL, TBQPOL) that can be either neutral, positive, or negative.

5.3.4 WinoReg_DL: A Deep-Learning Approach

Within this approach, we train WinoReg using deep learning (see Figure 5.15), which is another increasingly popular method inspired by the biological brain (Bengio et al., 2017; François, 2017; LeCun et al., 2015). As stated in the literature, deep learning can be seen as an extension of shallow neural network models, which have been around for many decades

[10]https://textblob.readthedocs.io/en/dev/

(Schmidhuber, 2015), albeit the term deep learning with the current resurgence started in 2006 (Bengio et al., 2017; Socher et al., 2012).

According to the literature (François, 2017; Schmidhuber, 2015), deep-learning has won numerous pattern and image recognition contests and achieved promising results on different NLP tasks. In this regard, techniques incorporating deep learning have been steadily gaining popularity (Bengio et al., 2017). However, it seems that this is not something that alone can lead the way to progress towards Artificial General Intelligence (AGI) (Marcus and Davis, 2019; Mitchell, 2019). It appears that deep learning is an excellent tool that is really good at specific tasks (e.g., statistics or patterns or words), missing at the same time what linguists call compositionality. Although patterns of words matter, this is just a tiny part of what our minds bring to this task (Adger, 2019). For instance, specific animals (e.g., rats, monkeys) are better at learning certain patterns in syllables than humans (Adger, 2019).

With deep learning algorithms, machines could learn good representations of data to help NLP tasks enormously. We can say that humans develop representations to enable learning and reasoning to achieve multiple tasks at hand, like tackling the WSC, which indirectly relates to the schema hardness. Given that deep learning systems are very effective at learning correlations, we train WinoReg within a Deep-Learning approach that can estimate the perceived human hardness indexes of Winograd halves (see Figure 5.15).

Specifically, WinoReg's deep-learning architecture is based on LSTM networks (see Figure 5.15), an updated version of RNNs that are capable of learning long-term dependencies (Hochreiter and Schmidhuber, 1997). LSTM networks may also be interpreted as something similar to computer memory (Sundermeyer et al., 2012). As stated in the literature, LSTM neural networks perform well in the field of language modeling (LM) (Sundermeyer et al., 2012), which can be used to solve various NLP tasks (Kocijan et al., 2019b). A language model is an essential model that captures how meaningful sentences can be constructed from individual words, which, in our case, seems to relate to the hardness of schemas. In the absence of features, with LSTM networks, WinoReg_DL can learn the joint probability function of sequences of words in a given sentence (Bengio et al., 2003), and at the same time, take into account all of the predecessor words (Sundermeyer et al., 2015, 2012) to output the perceived human hardness index of any given WSC half.

Within this approach, WinoReg_DL splits each examined half to select the sentence, as this is the only input-value needed for our LSTM-based approach. Next, it parses the examined sentence via spaCy dependency parser to remove the stop-words since they often occur in abundance. Then, for every word in the sentence, it returns its lemmatization as a way to determine possible relations between common words. The final step is to feed the parsed sentence into the model to retrieve its hardness index.

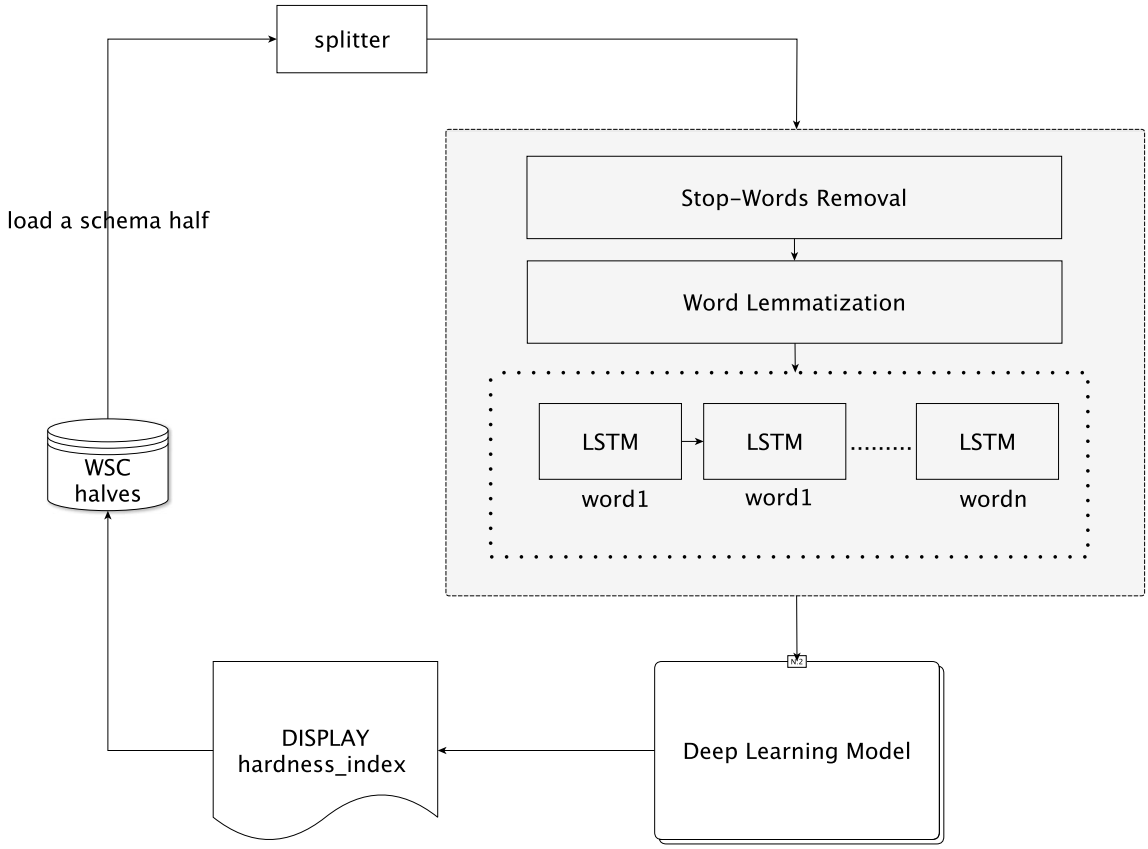


Figure 5.15 WinoReg's Architecture based on Deep-Learning (LSTM): Given a Winograd half WinoReg_DL outputs the perceived human hardness index.

Data Enhancement via Crowdsourcing

Although it is debatable (Gary Marcus, 2019), it is widely accepted that deep learning killed feature engineering, which is time-consuming (Socher et al., 2012). According to LeCun et al. (2015), most of the time, conventional machine learning techniques require considerable domain expertise for feature engineering. On the other hand, with deep learning, the amount of skill required for feature engineering reduces as the amount of training data increases (Bengio et al., 2017).

According to the literature, most approaches today that incorporate deep learning succeed because we can provide them with the necessary resources, as it is widely accepted that to generalize better, you have to do training on more data (Bengio et al., 2017). Additionally, if we can provide deep learning with a sufficient amount of data (LeCun et al., 2015), it will also reduce the generalization error / over-fitting (Bengio et al., 2017).

In our case, the availability of training data is limited since we only have access to 143 schemas (286 halves). To increase Bender's training data, we run an experiment on the **MicroWorkers (MW)** platform[11], which offers a reliable solution for various fields and research purposes (Peer et al., 2015) —Amazon Turk, which was used in Bender's experiments (Bender, 2015), was not available in our region.

As we have seen in Chapter 2, although multiple datasets exist, the only one close to the original WSC_ dataset is the DPR dataset (Rahman and Ng, 2012). The main difference between the two datasets is the absence of questions in the DPR dataset. To match Bender's experiment (Bender, 2015), we manually developed and added the necessary questions into all schemas. For the sake of simplicity, in our questionnaire, we use only the first half of each schema. Below, we will explain how we designed and ran our experiment along with our results.

Materials: For the questionnaire design, we used LimeSurvey software from our lab server[12]. All materials used in the experiment, including the halves used, are available online[13].

Design: We built the questionnaire and posted the link on the MicroWorkers platform (see Figure 5.16). A large body of work has shown MicroWorkers (MW) to be a reliable and cost-effective source for various fields and research purposes (Hirth et al., 2011; Peer et al., 2015). Platforms like MW offer a framework that enables employers to submit individually

[11]https://www.microworkers.com
[12]http://limesurvey.org
[13]http://www.nicosisaak.info

designed tasks to the crowd. MW has almost 1.5 million subscribed workers and offers more than 40 million tasks. The platform offers many features which can influence the completion time and the results. Moreover, it provides campaign creators with predefined groups of workers from different regions that are organized according to their skills (e.g., best-rated countries, writers, workers with certain language qualification tasks).

A total of 943 halves were included, where each half was displayed on a single screen. Each half's sentence was displayed at the top, followed by the question and the two possible answers displayed alongside (see Figure 5.17). Additionally, there was a comment section for participants to offer any comments they might have. All of the participants were informed that they could not change a submitted answer once the survey started. Compared to Bender's work (Bender, 2015), our workers were not given immediate feedback (correct or incorrect) after each trial, nor, by extension, access to their updated score.

Our questionnaire consisted of ten sections that ran independently. Each section included 100 unique halves except for the tenth, which included the last 43 halves of the dataset. Each participant was allocated only one position, meaning that they could participate in only one section.

Before taking the survey, each participant had to read a consent form to agree to participate. Next, they had to select their age, English language literacy level and pass a training phase to get familiarized with the task; immediate feedback (correct/incorrect) was given to the participants in the training phase. Further instructions were given as a warning not to sacrifice accuracy for speed.

To avoid problems related to cheating, we also included several test questions randomly displayed among the other schemas. As dealing with cheating in crowdsourcing platforms is a major challenge, test questions were used to verify if a given worker indeed holds a particular skill (Christoforaki and Ipeirotis, 2014; Hirth et al., 2011). Via an adaptive interjection of test questions at any time in any given place, we aimed to select the answers of really motivated participants. In this regard, in the end, we selected only the answers of participants who scored at least 70% on the test questions. Note that all participants were a priori informed about the test question mechanism.

The testing phase consisted of ten WSC halves that were explicitly designed to select the best participants' answers in the end. According to Bender (2015), many schemas suffer from ambiguity, meaning that it is difficult even for humans to answer them; this is related to the fact that the design of schemas is tricky and troublesome (Morgenstern et al., 2016). In this regard, the testing questions were designed to show the correct pronoun-antecedent without ambiguities. For instance, *Sentence: Jane sings better than Susan because she is*

a professional. Question: Who is a professional? Answers: Jane, Susan. Correct Answer: Jane.

Participants: The questionnaire started in April 2020 and ran for two months (from 21/4/20 23:30 to 01/06/20 23:00). According to our results, 306 adults aged 18 to 65+ (see Figure 5.18) from English-speaking countries attempted and finished the task. In terms of their knowledge of the English language, 8% reported that it was "very good", 19% that it was "good", and 2% that it was not good. Out of 429 participants who initially attempted the task, 115 did not finish —the participants selected at least one answer but left before they completed the task. Furthermore, eight participants did not pass the testing phase. The total cost of our campaign was $322. In the end, every half was answered by at least 30 participants.

Results: Based on the results, participants scored a mean accuracy of 91% ($\sigma = 0.14$), taking an average of 17.9 ($\sigma = 1.09$) seconds to answer every WSC half. It can be inferred from Figure 5.19 that our experimental results are in line with Bender's results (Bender, 2015), meaning that the human adults can tackle the WSC with a mean of 91-92%. The evidence we found supports Bender's results, meaning that this could serve as a baseline for human adult performance on the WSC.

Furthermore, our results show that the two datasets do not differ significantly. Specifically, in Bender's work, it was noted that the majority of the DPR dataset (Rahman and Ng, 2012) seems to be easy Winograd schemas. On the other hand, our results do not seem to confirm their observation. It seems that the hardness indexes of the two datasets are similar, meaning that human adults make the same effort to solve them, albeit the majority of the recent work in the literature believed that the DPR dataset is easier than the original WSC_ dataset (Kocijan et al., 2020). Our results are in line with a recent work showing that humans almost need the same effort to tackle schemas from the two challenges (96.5% for the original WSC dataset and 95.2% for the DPR dataset) (Sakaguchi et al., 2020).

On the other hand, this does not seem to be the case for the machines, as it seems that they can tackle the DPR dataset with bigger success with a mean of 85% compared to the WSC267 dataset, where they scored a mean of 71% (Sakaguchi et al., 2020). Maybe this is because the DPR challenge instead of questions includes only the definite pronouns, making the resolution to the machines easier to resolve. On the other hand, the WSC_ dataset is more closely to a question answering challenge that might require considerable effort for machines to tackle.

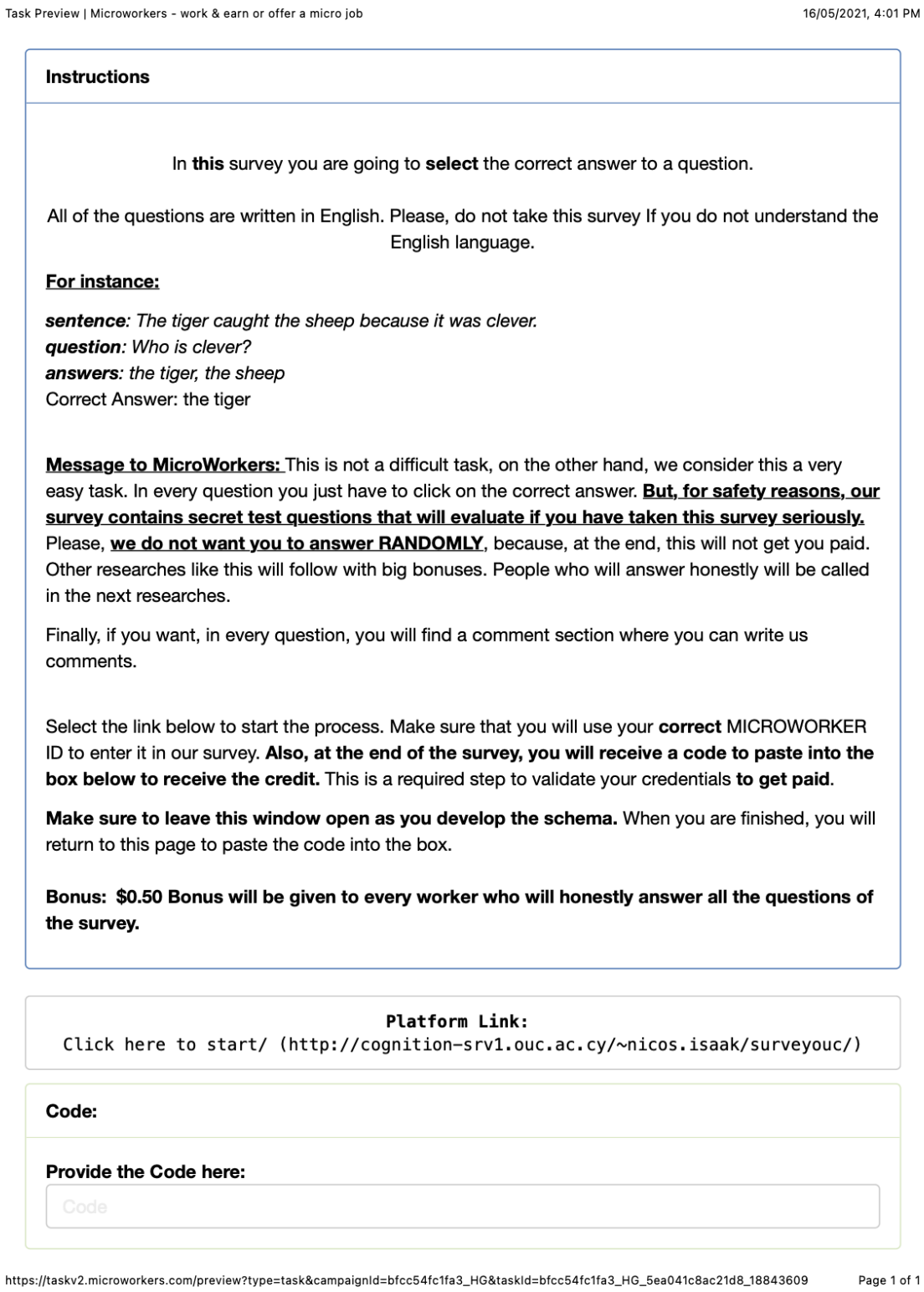


Figure 5.16 The ad we placed on the MicroWorkers platform to attract workers to answer WSC halves of the DPR dataset.

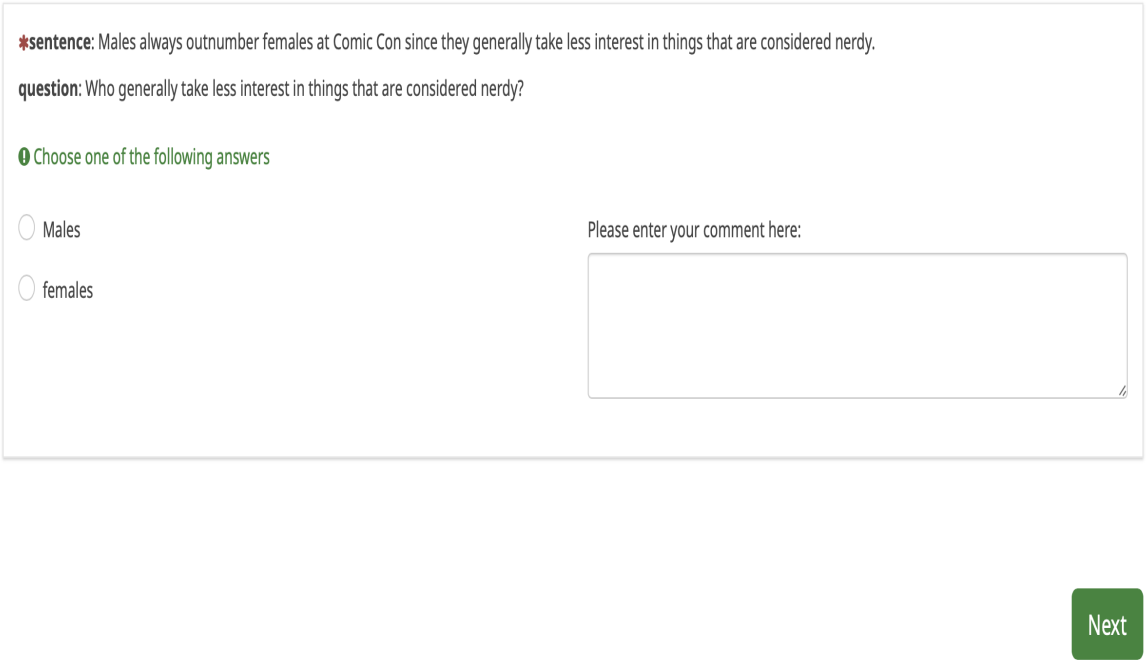


Figure 5.17 Screenshot of the experiment window.

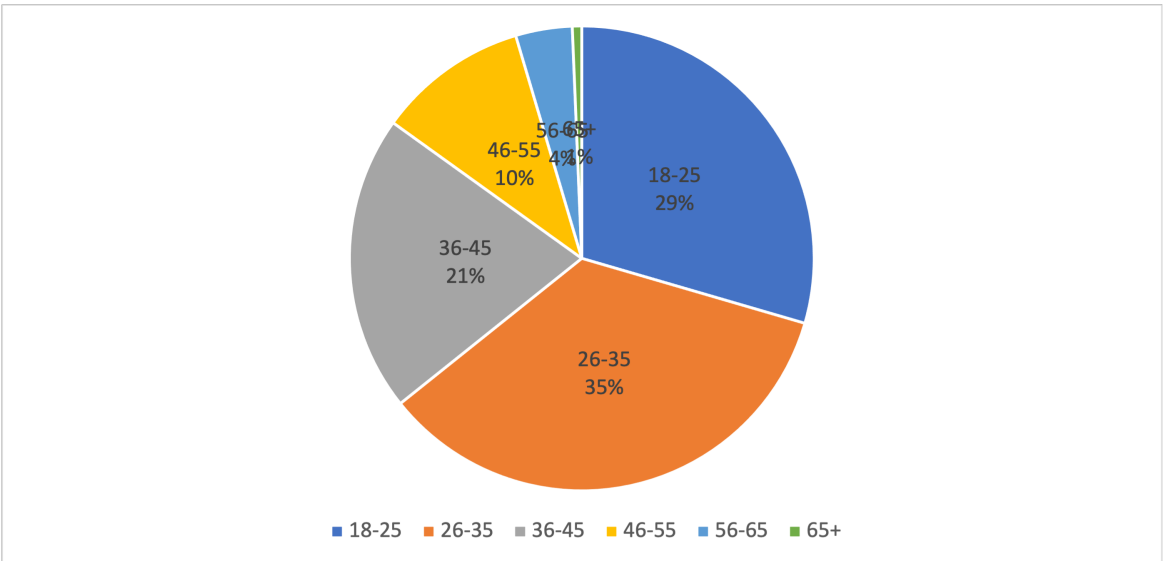


Figure 5.18 Distribution of reported ages.

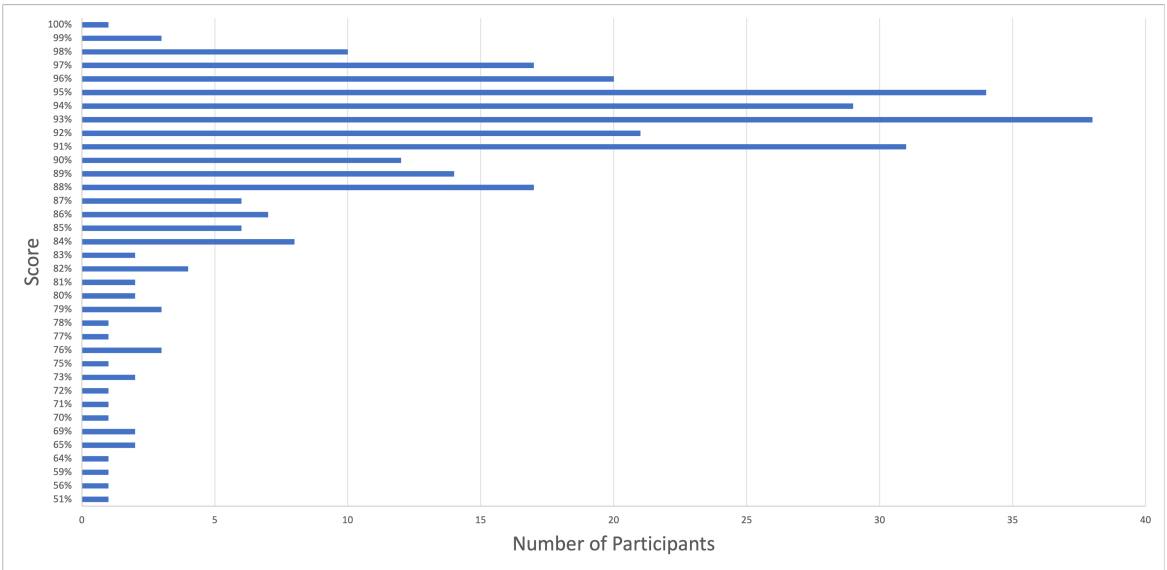


Figure 5.19 Questionnaire results: Distribution of scores grouped by the number of participants.

Fine-tuned Dataset: To get more data for our deep learning approach, we took the 943 schema halves of the DPR dataset (Rahman and Ng, 2012), used in our experiment, and added them to Bender's dataset along with their hardness indexes. To avoid having unbalanced data between the two datasets —943 schema halves of the DPR dataset over 286 schema halves of the original WSC dataset—, through oversampling, we increased the number of observations of the original dataset. These were copies of the existing halves, excluding the 100 halves used for testing purposes. The whole process resulted in 1872 halves, which were used for training and testing purposes.

5.3.5 Measuring the Hardness of WSC Halves (Experimental Evaluation)

In this section, we present our results by applying the methodology described in the previous paragraphs. We undertook several experiments to investigate if both WinoReg_RF and WinoReg_DL could be used to automatically differentiate between Winograd halves based on their perceived hardness for humans. We start by presenting WinoReg results based on the Random-Forest approach and continue with the LSTM-based approach.

Materials

Both experiments ran on a laptop computer (MacBook Pro 2018) with a 2.2 GHz 6-Core Intel Core i7 CPU, 16GB RAM, Radeon Pro 555X GPU with 4GB of GDDR5.

WinoReg_RF: The Random-Forest Approach

Here, by using the data from Bender's study (Bender, 2015), we examine whether the performance of WinoReg_RF can be predictive of the hardness of the WSC instances for humans. The results are reported on the testing set (100 halves of the original WSC286 dataset), expressed in terms of accuracy and correlation coefficient. For comparison purposes, the testing set is identical to the one used within the Wikisense-based approach. According to Bender's results, the human adult bar on the testing set is 91%.

System	Correlation Coefficient	Accuracy
Fixed Baseline	-1	90.87
Linear-Regression Baseline	0.16	90.65
Wikisense-based	0.22	77
WinoReg_RF	0.33	91.64

Table 5.3 Results of the Fixed Baseline, the Linear-Regression Baseline, the Wikisense-based approach, and WinoReg_RF, which was trained based on the Random-Forest approach —*accuracy* is calculated based on the mean absolute percentage error (accuracy = 100 - np.mean(mape)).

The fixed Baseline: For comparison purposes, we trained our Random Forest algorithm with only one feature, the human adult bar of 91%. Next, like within the Wikisense-based approach, we tested it on the first 100 WSC halves. Not surprisingly, although our results show an accuracy of 90.87%, this is done with a negative correlation coefficient of -1, meaning that two variables, WinoReg and adult's results, move in opposite directions (see Table 5.3).

The Linear-Regression Baseline: We also undertook a simple linear regression on the features extracted on the first 100 WSC halves. According to our results, the linear regression model achieved an accuracy of 90.65%, with a correlation coefficient of 16% (see Table 5.3).

Wikisense-based Approach: Recall that the Wikisense-based approach can return results only for 57% of the examined halves with a correlation coefficient of 38%. Given that the human adult bar on our testing set is 91%, for the unresolved halves, we can assume that the

Wikisense-based approach achieves an accuracy of 77% on all of the remaining halves, with a correlation coefficient of 22% (see Figure 5.20).

WinoReg_RF: The general picture emerging from the analysis is that WinoReg_RF can achieve an accuracy of 91.64%, significantly outperforming the Wikisense-based approach by 14.64% in accuracy and 11% in correlation coefficient (see Figure 5.21). For a better comparison between WinoReg_RF and the Wikisense-based approach, we compared the two methods only on the 57 halves the Wikisense-based approach could resolve. In this regard, the correlation coefficient of WinoReg_RF and humans rises to 47%, which is nine percentage points bigger than what the Wikisense-based approach was able to achieve (38%).

Taken altogether, the data presented here provide evidence that the performance of WinoReg, which is based on the random forest algorithm, *varies* across WSC halves in a manner that resembles the variability of the human performance more closely than what previous systems could achieve. This can be seen in both Figure 5.20 and Figure 5.21 that depict how the computed hardness index and the human hardness index vary across WSC halves. The results suggest that certain WSC halves that are easier or harder for humans are accordingly labeled as such by WinoReg_RF.

Speed Analysis: Given that the hardness index plays an essential role in organizing future Winograd challenges and the quality of the developed schemas, it is crucial to access the hardness index without delays. In this regard, we performed a speed analysis to show how fast WinoReg_RF can provide us with the hardness index of Winograd halves. Compared to the Wikisense-based approach, which requires on average eight hours for every WSC half, it was found that WinoReg_RF can return the hardness index of a WSC half, on average, in 1.6 minutes. This is the time needed to estimate the required features fed to the Random-Forest model. The results ultimately show that WinoReg_RF can deliver the hardness index of schemas 300 times faster than the Wikisense-based approach.

Feature Analysis: Here, we present the results obtained from analyzing the features used to train our Random Forest model. Consequently, as shown in Figure 5.22, where each element on the Y-axis presents the performance of WinoReg trained on all types of features except for the one shown, the correlation coefficient drops whichever feature is removed. In this regard, the results provide evidence of the importance of all feature types. The results show that the Number-of-Words, the Discourse-Connective-Relations, the Sentence-Type, the TextBlob-Polarity, and the Word-Relations are the essential features. Our results are in line with previous studies, where it was shown that features like sentence length,

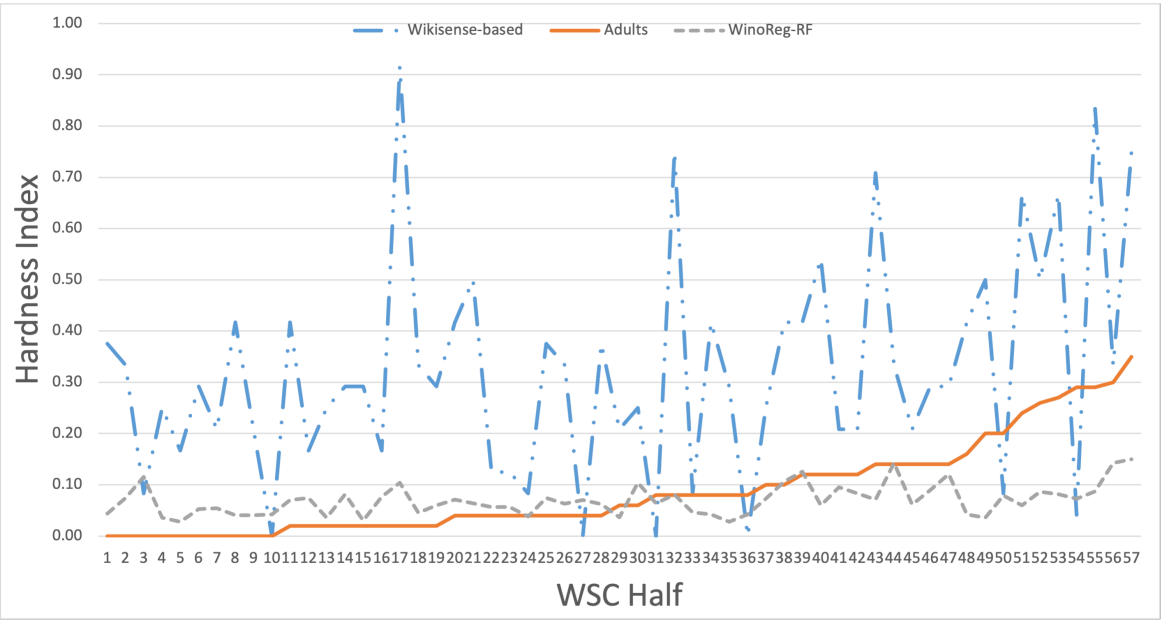


Figure 5.20 Variability of WinoReg_RF and Wikisense-based hardness-index across the 57 WSC halves on which the Wikisense-based approach originally was computed in relation to the variability of the human hardness-index. The results are sorted by the accuracy of adults, where smaller hardness indexes show easier halves to resolve.

sentence pattern, and word relations play an essential role in both the quality of the schemas and the tackle of the challenge (Rahman and Ng, 2012). Regarding the TextBlob-Polarity, our results show that it is better in capturing the polarity context than the other polarity features, which was unexpected as it is not commonly used in the literature. Regarding the OpinionFinder, previous works have stated that this might happen because it was trained on a completely different training set (Rahman and Ng, 2012). Contrary to our expectations, and unlike what other studies have mentioned (Rahman and Ng, 2012), Search-Engine-based features are not among the most valuable features. We believe that this might have happened because of changes in the Google search algorithm, which might have led to different results. Additionally, contrary to other works (Budukh, 2013), it seems that ConceptNet-Relations is not among the most useful features. Maybe its similarity factor cannot easily capture the semantics of each sentence. Lastly, it seems that the Negation-Feature is among the features that offer the least, which might be attributed to the fact that our dependency parser could determine if negation exists in only 41% of the tested WSC halves.

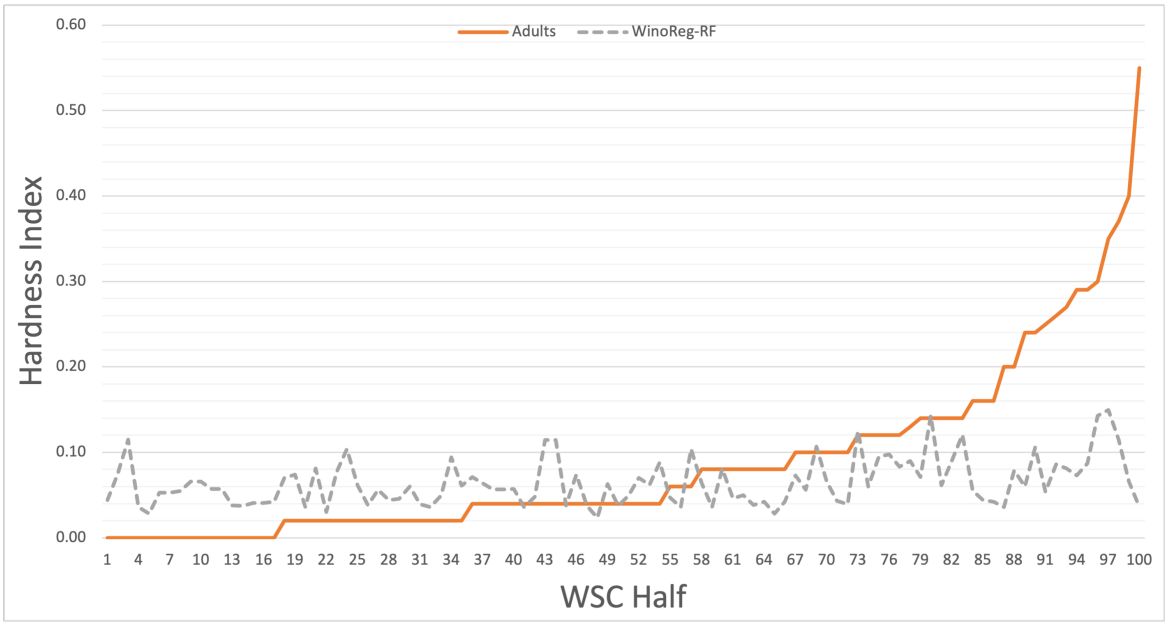


Figure 5.21 Variability of the WinoReg_RF hardness-index and the perceived human hardness-index across our testing set (100 WSC halves). WinoReg is trained based on the Random-Forest approach. The results are sorted by the accuracy of adults, where smaller hardness indexes show easier halves to resolve.

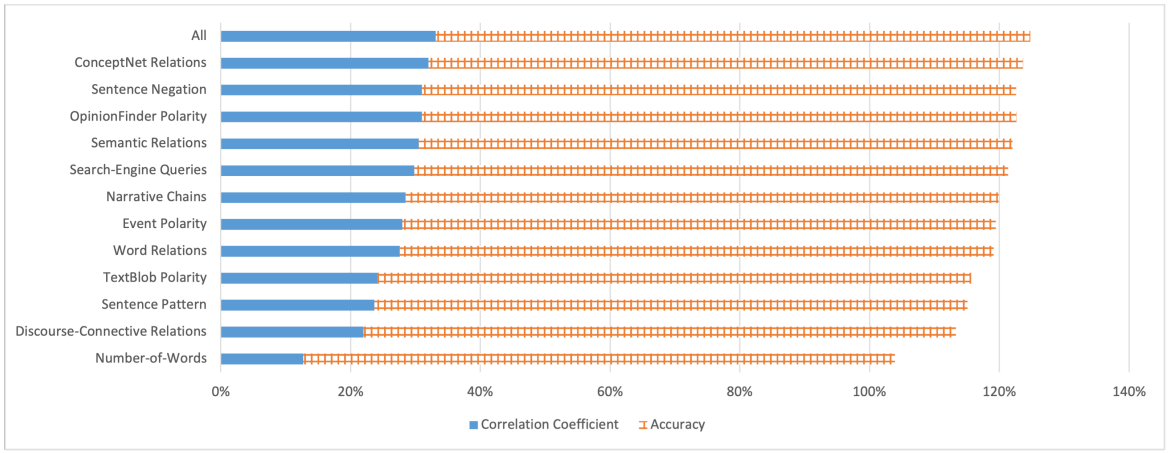


Figure 5.22 Results of WinoReg_RF's feature-decrement experiments. We can see the model's performance trained on all types of features except for the one shown in that row.

WinoReg_DL: The LSTM-based Approach

Here, we present our results by applying the LSTM-based approach described in the previous sections. Within our experiments, we examine whether this a priori appropriateness of the LSTM-based approach can be predictive of the hardness of the WSC halves for humans. The results are expressed in terms of accuracy and correlation coefficient. For comparison purposes, the testing set is identical to the one used in both the Random Forest (WinoReg_RF) and the Wikisense-based approach.

The optimal values for hyper-parameters used, which are vital to enhancing a neural network model (Le et al., 2017), were determined through trial-and-error. To build and train our model, we used the Keras functional API (François, 2017). We compiled our model with the "Adamax" optimizer and the "mean_absolute_error" loss function to compute the mean of the absolute difference between labels and predictions. In this regard, the regression loss function represents the measure of success for the task at hand to predict the hardness-index of any WSC half.

We started with the sequential model API, where model layers were created and added to it. Initially, we added an embedding layer to associate vectors with words. In this regard, we considered all the words in our dataset (input dim= 3648), with a maximum of each half's sentence of fifty words (output dim=50). Next, we added our LSTM layer, consisting of eighty-seven units (neurons), and, to prevent overfitting, we used a dropout layer (0.2) to ignore randomly selected neurons during the training process. Additionally, we used a recurrent dropout of 0.2 to mask the connections between the recurrent units. Finally, we added a single unit layer to reduce our LSTM network's shape to match our desired output —hardness score prediction.

We have also examined transfer-learning to train our model with better generalization properties, as several transfer-learning methods significantly improved a wide range of NLP tasks in the literature (Ruder et al., 2019). To improve the accuracy of our model, we have tested two well-known datasets, Glove (glove.6B.50d to glove.6B.300d) (Pennington et al., 2014) and fastText (cc.en.300.vec) (Joulin et al., 2016). We tested them by loading their pre-training vectors into our embedding layer for obtaining vector representations for words so that similar words would be based together, albeit without any success in our results. Given that pre-trained embeddings have been used successfully in various NLP tasks, the reason for this discrepancy might be the artificial structure of the Winograd instances. Specifically, due to the WSC paucity of data, the embeddings learned in other tasks cannot be used to output the hardness of artificially created sentences.

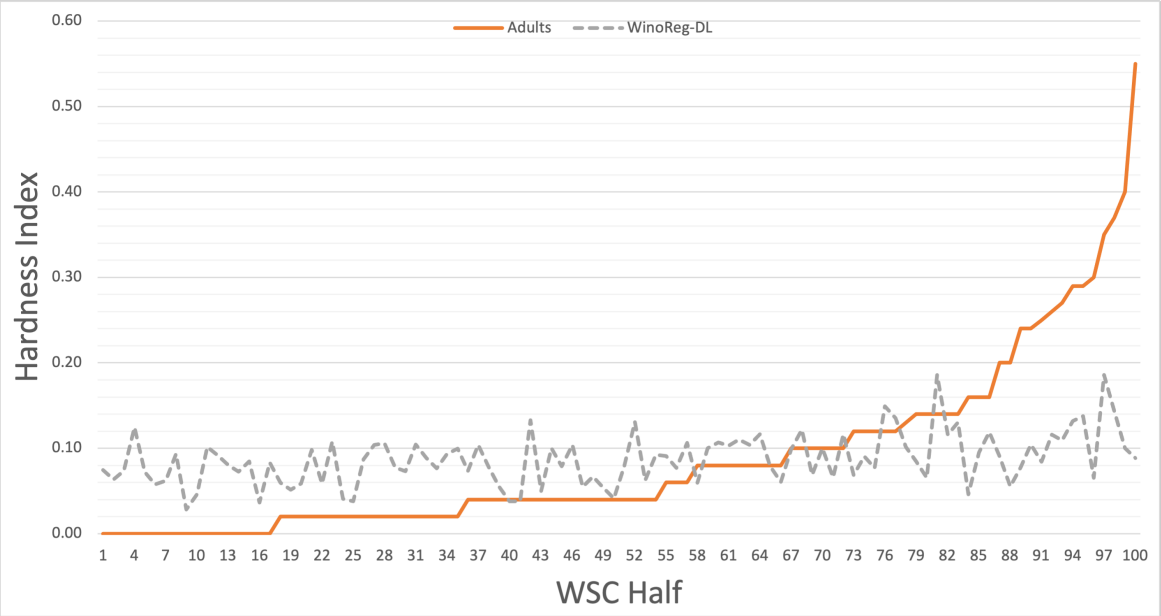


Figure 5.23 Variability of the WinoReg hardness-index and the perceived human hardness-index across our testing set (100 halves). WinoReg is trained based on the LSTM-based approach. The results are sorted by the accuracy of adults, where smaller hardness indexes show easier halves to resolve.

Finally, for our training and testing purposes, we split the dataset into a training and a validation set (validation_split=0.3; train on 1310 samples, validate on 562 samples) and tested the resulting model on the testing dataset (100 halves).

System	Correlation Coefficient	Accuracy
Wikisense-based	0.22	77
WinoReg_RF	0.33	91.64
WinoReg_DL (LSTM-based)	0.39	93.27

Table 5.4 Results of the Wikisense-based hardness, and WinoReg based on both the Random-Forest and the LSTM-based Approach.

Results: Our tests show a positive correlation between WinoReg_DL results and the perceived human hardness-indexes across the WSC halves (see Table 5.4). Specifically, within the LSTM-based approach, WinoReg_DL achieved an accuracy of 93.27% with a correlation coefficient of 39%.

Compared to the Wikisense-based approach, WinoReg_DL can achieve a higher correlation coefficient of 17%. Additionally, if we compare the two systems on the 57% of

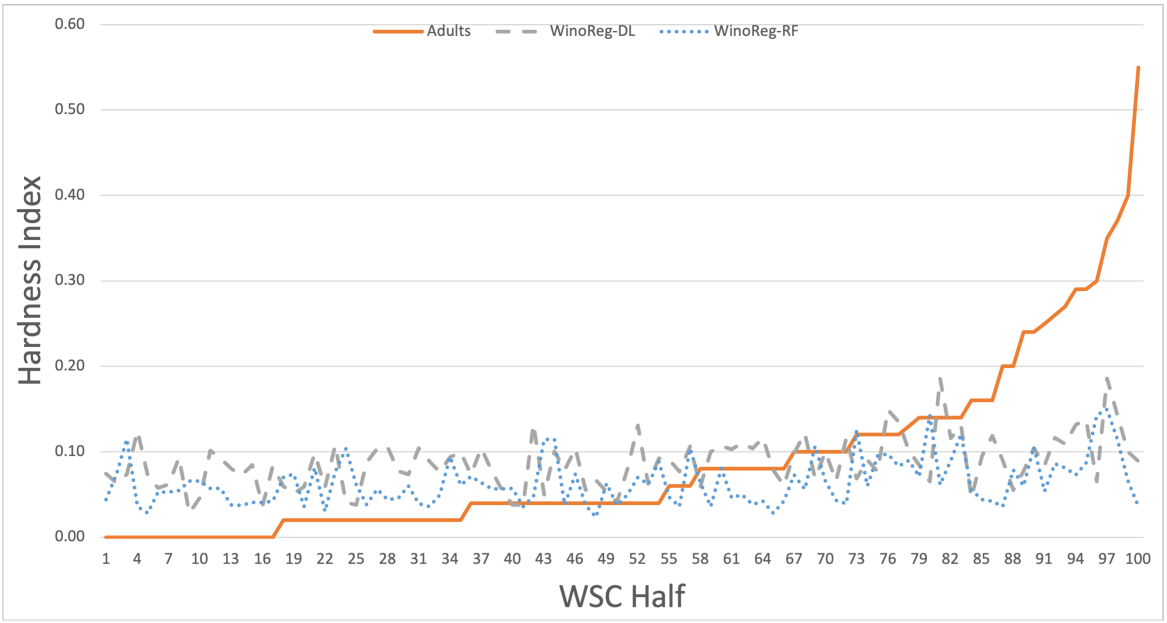


Figure 5.24 Variability of WinoReg approaches, in relation to the perceived human hardness-index across our testing set (100 schema halves). The results are sorted by the accuracy of adults, where smaller hardness indexes show easier halves to resolve.

the schemas the Wikisense-based approach was able to solve, the correlation-coefficient difference rises to 10%, in favor of WinoReg_DL (38% vs. 48%).

As shown in Table 5.4, the LSTM-based approach has an advantage over the Random-Forest approach. Our results highlighted that WinoReg results correlate better to adult results within the LSTM-based than the Random-Forest approach. Particularly, the former outperforms the latter by 2% in accuracy and 6% in correlation coefficient. Additionally, the LSTM-based approach does not require feature engineering, which offers it a compelling advantage over the Random-Forest approach.

We also performed a speed analysis to identify how fast the Deep-Learning approach can provide us with results. Our analysis revealed that within the LSTM-based approach, WinoReg could return results, on average, in 1.6 msec for any given half, which means 60 thousand times faster the Random Forest and 18 million times faster the Wikisense-based approach. This means that WinoReg_DL can return results in real-time with no further delays.

Taken all together, the general picture emerging from the analysis is that the performance of WinoReg_DL *varies* across Winograd halves in a way that resembles the variability of the human performance more closely than what other approaches could achieve (see Figure 5.23). In this regard, certain WSC instances that are easier or harder for humans are respectively identified as such by WinoReg_DL. Overall, both WinoReg approaches have

a compelling advantage over the Wikisense-based approach. Specifically, both WinoReg approaches correlate positively with a correlation coefficient of 22%, suggesting that halves that are easier or harder for the Random-Forest approach might also be labeled as such more closely by the LSTM-based approach (see Figure 5.24).

5.4 Chapter Summary

In this chapter, we investigated the possibility of building systems that can output the perceived human hardness index of Winograd instances.

We have shown how a particular existing system developed for the WSC can form the basis for deriving a data-driven metric of hardness for WSC instances. In this regard, experiments we undertook showed that the Wikisense-based approach could output the hardness indexes of Winograd instances, albeit with limitations regarding the number of instances it could be applied on. Evidence that the system's computed hardness index is correlated with the perceived human hardness was offered through two studies, one from the literature and one designed as part of this work.

In the following paragraphs, we investigated the possibility of building a system that can output the perceived human hardness index of any Winograd instance in the shortest time possible. Our results have shown that this is possible via the training of a system based on two approaches: the Random-Forest and the LSTM-based approach. Results have shown that our results correlate positively with human results. In particular, results have shown that with the Random-Forest approach, we can achieve 91.64% of accuracy with a 33% correlation coefficient, whereas with the LSTM-based approach, 93.27% of accuracy with a 39% correlation coefficient. Even though the results of the two approaches seem close, the substantial benefit of the LSTM-based approach lies in the response time of the model, which is 60 thousand times faster than the Random-Forest approach.

Researchers or challenge organizers can use our systems to group Winograd instances regarding their perceived human hardness indexes. Moreover, our systems can be used by WSC-based CAPTCHA services to ensure that more challenging Winograd instances would be generated to prevent fraudulent actions.

---

1. Isaak, N. and Michael, L. (2017). How the Availability of Training Material Affects Performance in the Winograd Schema Challenge. In Proceedings of the (IJCAI 2017) 3rd Workshop on Cognitive Knowledge Acquisition and Applications (Cognitum 2017).

2. Isaak, N., Michael, L.: A Data-Driven Metric of Hardness for WSC Sentences. In:Lee, D., Steen, A., Walsh, T. (eds.) GCAI-2018. 4th Global Conference on Artificial Intelligence. EPiC Series in Computing, vol. 55, pp. 107–120. EasyChair (2018). https://doi.org/10.29007/398z, https://easychair.org/publications/paper/nRrp

3. Isaak, N. and Michael, L. (2020). WinoReg: A New Faster and More Accurate Metric of Hardness for Winograd Schemas. In Danoy, G., Pang, J., and Sutcliffe, G., editors, GCAI 2020. 6th Global Conference on Artificial Intelligence (GCAI 2020), volume 72 of EPiC Series in Computing, pages 46–58. EasyChair.

4. Isaak, N. and Michael, L. (2021). Experience and Prediction: A Metric of Hardness for a Novel Litmus Test. Journal of Logic and Computation. exab005.

6

Designing new Winograd Instances from Scratch

6.1 Introduction

It is well-known that the WSC is a carefully crafted pronoun resolution task, meaning the development of schemas is a laborious job (Ortiz, C, Nuance Communications, personal communication, February 2018). According to Morgenstern et al. (2016), the development of schemas requires creativity and inspiration, and it is too troublesome to be done on a yearly or biennial basis to support, for instance, competitions on the WSC or the testing of systems that might have been trained on existing collections of Winograd schemas. To the best of our knowledge, the availability of Winograd schemas is not sufficient. Two well-known datasets exist that match the challenge restrictions, the original WSC_ (Levesque et al., 2012) and the DPR dataset (Rahman and Ng, 2012), which is a relaxed version of the original WSC_ dataset.

In Chapter 5, we showed that significant correlations were obtained between the Wikisense training set sizes and human performance. We showed that we could resolve more schemas with larger training sets but not enough to tackle the WSC. If we want to realize flexible commonsense reasoning, obtaining the data can also be very challenging. It seems that the availability of more schemas might contribute to the faster solution of the problem. Moreover, we expect that the adoption and use of WSC-based CAPTCHAs (see Chapter 4) will also present AI researchers with the novel challenge of automating the construction of new Winograd instances.

In either case above, an extensive collection of available Winograd schemas/halves would seem to be a prerequisite, or at least a facilitator, for further work and progress,

meaning that such uses of the task necessitate the availability of an extensive and presumably continuously-replenished, collection of available Winograd instances.

Towards addressing these limitations, we propose two approaches for the development of Winograd instances from scratch: i) via the WinoFlexi, which is a flexible online crowdsourcing system, and ii) through Winventor, which is a machine-driven approach that blends the advantages of deep learning and Natural Language Processing (NLP). Our empirical evaluation of WinoFlexi's performance suggests that it allows crowdworkers to develop Winograd schemas of good quality similar to that of most typical existing collections. On the other hand, given that this is a challenging task even for humans, Winventor develops a limited number of Winograd instances (schemas/halves). We want to point out that Winventor, as this is the first work in the field, does not purport to replace humans but to considerably help them in the schema development task.

Below we start by presenting WinoFlexi and Winventor, followed by our analysis and discussion of our findings. The sections below explain each of these tasks, along with the tools and techniques we have developed.

6.2 The WinoFlexi Approach

6.2.1 Introduction

In this section, we present WinoFlexi, a flexible online collaboration system that allows members of crowdsourcing platforms to collaborate *explicitly* for the development of Winograd schemas. Currently, more skilled labor activities are carried out online via crowdsourcing platforms. These platforms can eliminate geographic constraints and help workers pursue work that they find valuable (Christoforaki and Ipeirotis, 2014). While crowdsourcing strategies can be found throughout the centuries, no study has looked specifically at how we can build tools that could help develop Winograd schemas, and any evidence for these kinds of tools has been mainly anecdotal.

With WinoFlexi, we hope to bring together researchers and people from across disciplines concerned with the acquisition and use of language data in the context of data science and knowledge-based applications, like the WSC. WinoFlexi uses a combination of tools that enhance the schema-development process: *i)* it is more cheat-proof than existing crowdsourcing platforms, and *ii)* it uses test questions that are closer to the schema-development process that benefit non-dubious workers and ban dubious ones. Our empirical study with workers from an existing crowdsourcing platform showed that WinoFlexi could be used for the development of Winograd schemas that are comparable to the original dataset (WSC286),

developed by experts. In this regard, WinoFlexi is a system able to promote the original goals of the WSC through the development of high-quality schemas.

Below, we present a related but quite different method used to collect Winograd-like instances from the literature. The rest section focuses on the design of appropriate crowdsourcing mechanisms for our particular task and the evaluation of the developed Winograd schemas.

6.2.2 The WinoGrande Collection

Compared to our work, the closest point of comparison is the WinoGrande dataset consisting of 44k Winograd-like examples (Sakaguchi et al., 2020). Although the WinoGrande dataset was inspired by the original WSC design, it was adjusted to improve both the scale and the hardness of the dataset. In this regard, the resulting schemas do not have the same structure as the original WSC287 dataset. For instance, a WinoGrande schema is formatted as a fill-in-the-blank problem, resolving the task more straightforward without including questions and definite pronouns. In each schema, the blanks correspond to mentioning one of the pronoun names in the context —e.g., *1.) Sentence: The food of Dennis is made spicy, but Donald's is spicier because _ is from South America., option1: Dennis, option2: Donald. 2.) Sentence: The food of Dennis is made spicy, but Donald's is blander because _ is from South America., option1: Dennis, option2: Donald.*

Furthermore, in developing the WinoGrande dataset, the authors used and enhanced the creativity of crowd workers by priming them by a randomly chosen topic as a suggestive context of two domains, social and physical commonsense. Compared to the WinoGrande dataset, as we will see in the following sections, WinoFlexi's dataset consists of schemas comprising sentences, questions, and pronoun targets. Furthermore, WinoFlexi is a fully developed system that employs several mechanisms to train, assist, and enhance workers' creativity of any crowdsourcing platform in developing Winograd schemas from scratch.

6.2.3 WinoFlexi's Architecture

We continue to present our platform and its constituent modules (see Figure 6.1) and discuss how the crowd collaborates to built schemas under WinoFlexi's evaluation mechanisms. Recognizing that the schema development process is tedious and troublesome, WinoFlexi is built to act as an assistant with effective incentive mechanisms for the crowd.

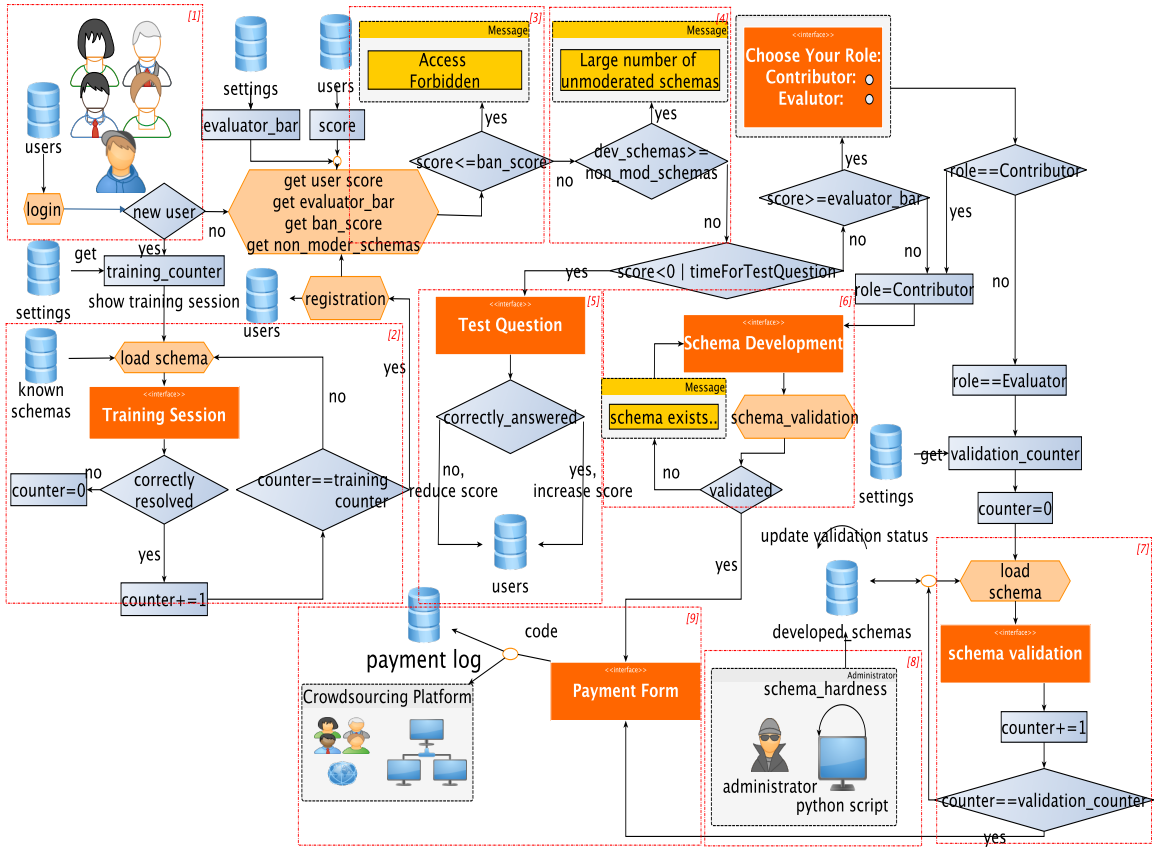


Figure 6.1 WinoFlexi's architecture for the development of Winograd schemas. The various parts of the architecture are marked in red rectangles and are discussed in section 6.2.3.

Registration Process

The first step for each worker is to apply as a Contributor to our platform, where they register their credentials[1] (see *part-1* in Figure 6.1). Workers need not be domain experts but need to have a strong command of English to ensure that schemas have no spelling, syntactic, or grammatical errors and comply with the schema development process.

Training Phase

To maximize the quality of the developed schemas, every Contributor has to complete a training task (see Figure 6.2 and *part-2* in Figure 6.1). During the training phase, workers are familiarized with the development process by being asked to correctly resolve randomly selected schemas from the original dataset (WSC300) (Levesque et al., 2012). The length of the training phase can be increased either by the system administrator or automatically by

[1]http://cognition.ouc.ac.cy/mcSchemaBuilder

WinoFlexi to ensure that the produced schemas' quality meets expectations. In particular, if the *auto-training* flag is enabled, then the length of the training phase for every new registered Contributor is determined by how much the number of invalid schemas produced so far exceeds the number of valid ones.

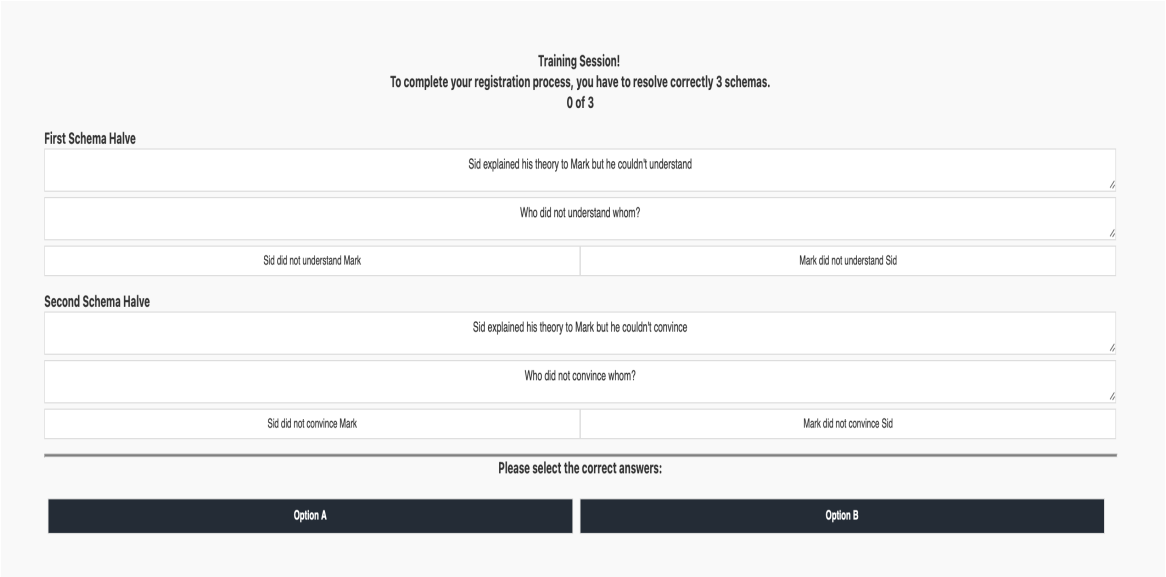


Figure 6.2 The Contributor's Training Phase.

Contributors

Contributors are workers who develop schemas from scratch (see *part-6* in Figure 6.1) using the dashboard shown in Figure 6.3. When a Contributor adds a schema (a pair of halves), WinoFlexi does some basic checks: *i)* It checks if each half comprises a sentence, a question, and two pronoun targets. *ii)* It checks if the correct pronoun target of each half has been selected. *iii)* It checks if the sentence, the question, and the two pronoun targets of each half are related. *iv)* It checks if the two halves are related. Relatedness is checked using the heuristic approach shown in Figure 6.4 applied to each of the pairs sentence-question, sentence-first_pronoun_target, sentence-second_pronoun_target.

Evaluators

Workers who validate schemas are called Evaluators (see *part-7* in Figure 6.1). Contributors are allowed to take on this dual role if they meet two requirements: first, the percentage of their valid and approved (by other Evaluators) schemas among those that they have contributed that far exceeds a certain threshold —which we have set to be 90%, corresponding to the bar for

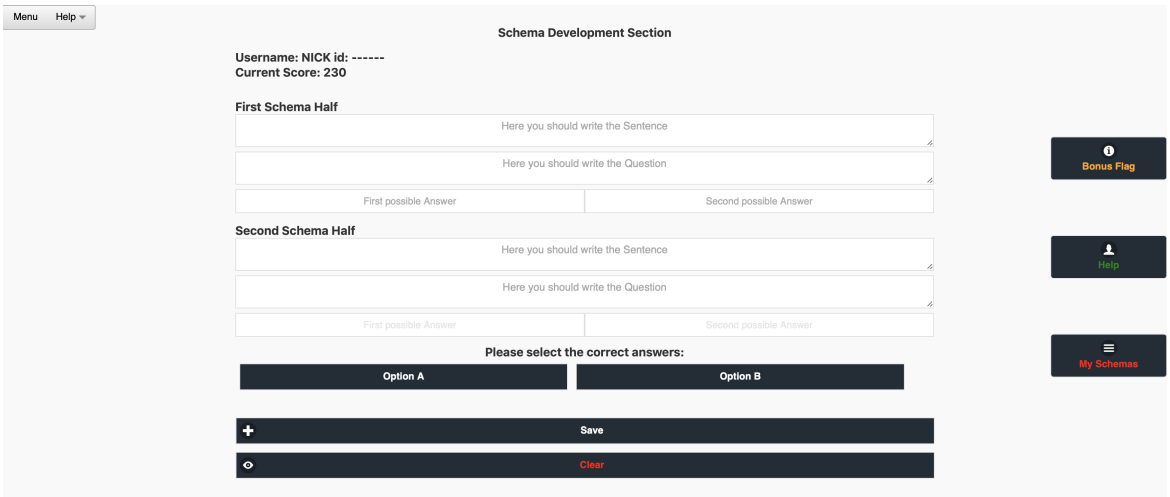


Figure 6.3 The Contributor's dashboard.

near-adult human abilities on the WSC (Bender, 2015); second, their score (which we discuss later) is above a certain another threshold. Contributors who are also Evaluators choose the role they interact with WinoFlexi at login time. At the beginning of the development process, the only Evaluator is the system administrator. The evaluation process requires answering several yes/no questions using the dashboard shown in Figure 6.5. Affirmative responses to all but the first question are necessary to characterize a schema as valid. Additionally, the Evaluators have access to a similarity tool to detect if the Contributors follow a pattern to develop similarly-looking schemas. The tool acts like a *leakage-detector* (Christoforaki and Ipeirotis, 2014) that queries the WinoFlexi-dataset and original WSC dataset to determine if a newly-contributed schema is "leaked", in that it is significantly similar to an existing schema. Specifically, this tool calculates the similarity between the Contributor's schema currently under evaluation and each schema of the datasets above. If a similarity value is found to be above a specific threshold (default is 0.60), the Contributor's schema is marked as leaked. Given that the usage of this tool is not a prerequisite step, the Evaluators are allowed to change the similarity threshold to any value in the range of 1 to 100. Each approved schema increases the Contributor's score and each "leaked" schema decreases it, affecting whether the Contributor will meet the requirements to become an Evaluator.

Quality-Assurance Measures

Here, we will present additional mechanisms that are used to ensure the quality of the developed schemas. We start with **Test-Questions** that many crowdsourcing platforms use as an assessment method, offering their certification mechanisms to verify that a given worker indeed holds a particular skill (Christoforaki and Ipeirotis, 2014; Hirth et al., 2011). Previous

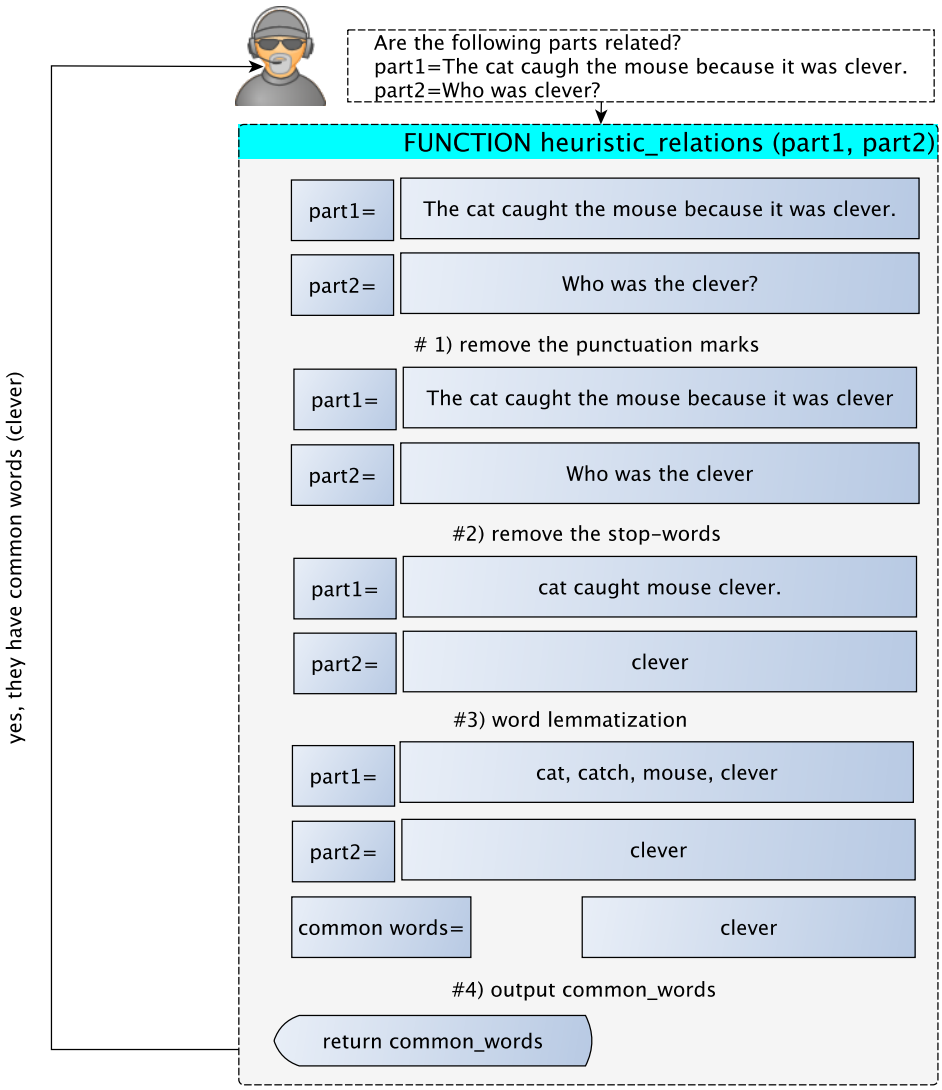


Figure 6.4 Heuristic relations to eliminated problems with schema cohesion.

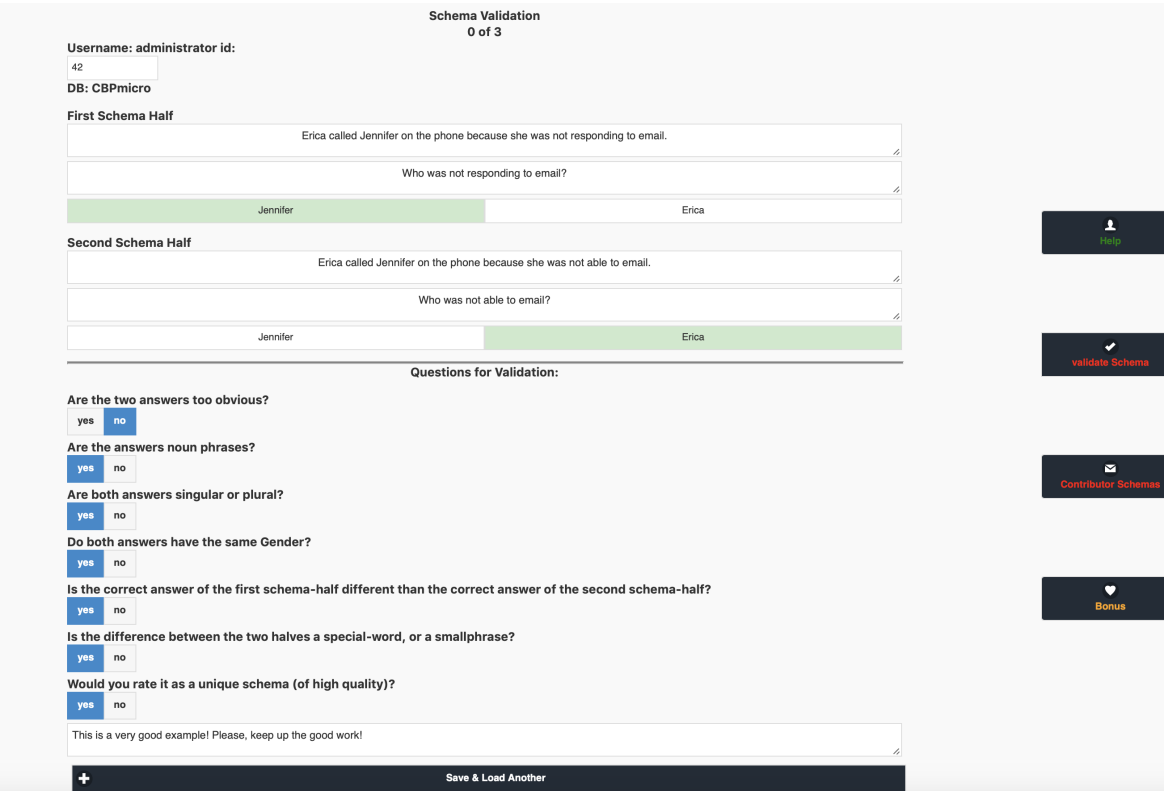


Figure 6.5 The Evaluator's dashboard.

works indicate that more interactive studies may motivate participants to read instructions more carefully, leading to better compliance (Peer et al., 2015). Our approach is based on the adaptive interjection of test questions and on rewarding the worker with a positive score for successfully resolving them (see *part-5* in Figure 6.1). WinoFlexi can be enabled to display test questions as often as necessary to both Contributors and Evaluators; this can be manually handled by the system administrator or automatically controlled by the system. By default, a test question has a 10% probability of being displayed after every login. If the *auto-testing* flag is enabled, this probability is adjusted in a manner analogous to how the length of the training phase is adjusted. Test questions are selected from the WinoFlexi-dataset (validated contributed schemas) and the original WSC dataset (WSC300), meaning that both collections include schemas that strictly follow the WSC rules. Correct/wrong answers to test questions increase/decrease a worker's score.

We continue with the **Ban-Score** mechanism. Online certification of skills is still problematic since dealing with cheating is a major challenge. To that end, the *ban-score* mechanism automatically bans workers who have a sufficiently low score (see *part-3* in Figure 6.1), with the threshold identified empirically.

Furthermore, to prevent workers from entering a large number of potentially invalid schemas, WinoFlexi uses the **UnValidated** mechanism. This mechanism limits the number of schemas each worker can develop before they undergo the validation process (see *part-4* in Figure 6.1).

Finally, WinoFlexi leverages the **Schema-Hardness** mechanism (see Chapter 5) to generate feedback to the Contributors (see *part-8* in Figure 6.1). Towards this goal, we follow a single-step approach for labeling schemas with a hardness score which indirectly shows if a schema is considered hard to answer by a machine; Winograd schemas are accordingly labeled as such by the computed hardness index. The hardness index is presented to the Contributors and the Evaluators. If the majority of a Contributor's schemas are easy, then our system prompts them to develop schemas that are harder to answer.

Payments and Rewards

Payments and rewards refer to compensations used to motivate both Contributors and Evaluators to ensure the quality of the developed schemas. We start with the **Payment-Procedure**. Most of the micro-tasks on the crowdsourcing platforms are priced individually, and workers are paid a base rate multiplied by the number of correctly completed tasks. Whatever their motives are, workers want to earn money and seek out tasks to maximize their expected earnings. To make sure that only the workers who developed schemas are going to get paid, we enhanced WinoFlexi with a payment verification plug-in, controlled by the system's

Administrator (see *part-9* in Figure 6.1). Upon each schema development (or validation), Contributors and Evaluators are prompted with a notification message and a code, automatically generated and inserted into our database. Each worker has to provide the same code on their crowdsourcing platform to receive the actual payment.

Regarding **rewards**, we know that workers recruited through crowdsourcing platforms must receive a small fixed payment for participating in the experiment and/or a bonus for high-quality results (Hirth et al., 2011). Past work has shown that the quality of work produced in a crowdsourcing working session can be influenced by the presence of financial incentives, such as bonuses. WinoFlexi adopts this philosophy and rewards Contributors based on "relative performance", namely only the worker that performs best receives rewards.

6.2.4 Experimental Design and Results

In recent years, a growing number of researchers have been using well-known crowdsourcing platforms (Peer et al., 2015). As we have seen in Chapter 5, a large body of work has shown **MicroWorkers (MW)** to be a reliable and cost-effective source for various fields and research purposes (Hirth et al., 2011; Peer et al., 2015). To attract the worker's attention, we used a simplified title (*Develop Groups of Sentences, Questions & Answers that Meet Certain Criteria*) and promoted it on the MW platform (see Figure 6.6). To attract workers, we advertised our campaign as a 5-minute task for the development of each schema. With every login, each worker was allowed to develop one schema. Workers were allowed to participate as many times as they wanted, and they were getting paid per schema. Furthermore, workers were given instructions explaining the task directing them to develop schemas without sacrificing accuracy. It was made clear that the development of invalid schemas might ban them from the system.

The halves of a Winograd schema cannot be solvable by either selectional restrictions or word associations. Although these are not obvious concepts, we avoided giving our workers such guidance to avoid putting a high cognitive load on the crowd and led them to enjoy the whole process. Instead, we decided to guide the whole process through WinoFlexis' mechanisms. One could argue that the obvious process for some people is not obvious for others as this is directly related to their personal biases and culture. For instance, in developing the WinoGrande dataset (Sakaguchi et al., 2020), even though the workers were advised to avoid creating schemes with instance-level biases like word association, results showed that the constructed dataset still had dataset-specific biases, meaning that it is not something one could entirely avoid.

We promoted WinoFlexi only under the Hired-Section of *English Speaking Countries + En*, meaning that only members of that group were able to participate. Our selected

workers have both English proficiency and admission tests passed. For our task, we offered compensation of $1.00 for each developed schema or the validation of three schemas in a row. We also advertised a bonus for quality developed schemas.

The experiments ran for one week and yielded more than 165 schemas (see Table 6.1), from 50 workers aged 18 to 65. From the developed schemas, 135 (81%) were valid, and 30 invalid. The highest score of a worker was 250 points, and the lowest was -70 (see Figure 6.7); the Contributor with the lowest score was automatically banned by WinoFlexi. The majority of the workers had a non-negative score. The top three workers had a score of at least 170, which well-exceeded the second condition for qualifying as an Evaluator. The total cost of our campaign was $258.00. The Contributors were paid $165.00 for the schema development process, with an additional $63.00 given as bonuses. On the other hand, $30.00 was paid to Evaluators for the schema evaluation process.

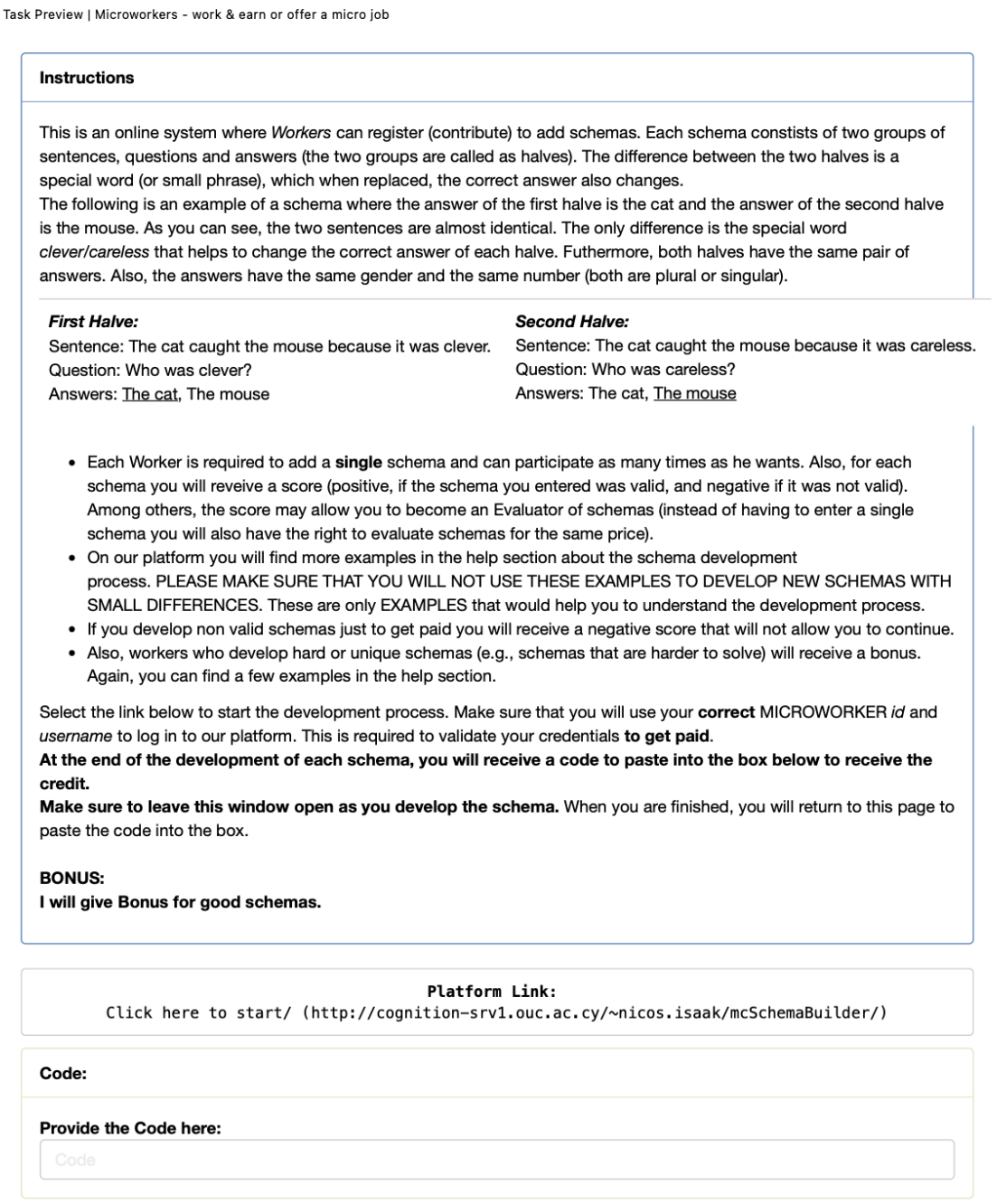
Our experimental evaluation shows that WinoFlexi supports the development of *valid* schemas, costing approximately $1.91 per schema. Considering the challenge difficulties, we believe that this is a fair cost. The mean response time across all workers was 1.48 minutes (after training), and the average time for the best worker was 1.66 minutes. Given that, with every login, each worker was allowed to develop one schema, many workers seem to have pre-developed their schemas so that it would not have taken them enough time on WinoFlexi. That said, the workers reported timings should be taken with a grain of salt. Sixty percent of the bonuses were offered to the top five workers. We believe that our adopted approach leads to more bonus opportunities for workers who submit good quality schemas.

Evaluators were not observed to show a preference for the evaluation process over the schema design process. Although the evaluation process seems more straightforward, workers might have preferred the schema design process for the following reasons: i) they were more familiar with the schema design process than the evaluation process; ii) through the schema design process, they were eligible for rewards, such as cash bonuses; iii) they did not want to leave other Contributors unpaid or lower their score.

The general picture emerging from the analysis above is that WinoFlexi is a platform where workers can collaborate for the schema development process. However, considering this approach, there is a key question that we have not addressed yet: How does the quality of the developed schemas compare to that of schemas developed by experts? To answer this question, we undertook a quantitative and qualitative analysis.

Quantitative Analysis

Here, we test the quality of WinoFlexi's schemas and compare them with the quality of those developed by experts. Initially, we test if existing coreference resolution systems can tackle


Task Preview | Microworkers - work & earn or offer a micro job

Instructions

This is an online system where *Workers* can register (contribute) to add schemas. Each schema constists of two groups of sentences, questions and answers (the two groups are called as halves). The difference between the two halves is a special word (or small phrase), which when replaced, the correct answer also changes.
The following is an example of a schema where the answer of the first halve is the cat and the answer of the second halve is the mouse. As you can see, the two sentences are almost identical. The only difference is the special word *clever/careless* that helps to change the correct answer of each halve. Futhermore, both halves have the same pair of answers. Also, the answers have the same gender and the same number (both are plural or singular).

First Halve:
Sentence: The cat caught the mouse because it was clever.
Question: Who was clever?
Answers: The cat, The mouse

Second Halve:
Sentence: The cat caught the mouse because it was careless.
Question: Who was careless?
Answers: The cat, The mouse

- Each Worker is required to add a **single** schema and can participate as many times as he wants. Also, for each schema you will reveive a score (positive, if the schema you entered was valid, and negative if it was not valid). Among others, the score may allow you to become an Evaluator of schemas (instead of having to enter a single schema you will also have the right to evaluate schemas for the same price).
- On our platform you will find more examples in the help section about the schema development process. PLEASE MAKE SURE THAT YOU WILL NOT USE THESE EXAMPLES TO DEVELOP NEW SCHEMAS WITH SMALL DIFFERENCES. These are only EXAMPLES that would help you to understand the development process.
- If you develop non valid schemas just to get paid you will receive a negative score that will not allow you to continue.
- Also, workers who develop hard or unique schemas (e.g., schemas that are harder to solve) will receive a bonus. Again, you can find a few examples in the help section.

Select the link below to start the development process. Make sure that you will use your **correct** MICROWORKER *id* and *username* to log in to our platform. This is required to validate your credentials **to get paid**.
At the end of the development of each schema, you will receive a code to paste into the box below to receive the credit.
Make sure to leave this window open as you develop the schema. When you are finished, you will return to this page to paste the code into the box.

BONUS:
I will give Bonus for good schemas.

Platform Link:
Click here to start/ (http://cognition-srv1.ouc.ac.cy/~nicos.isaak/mcSchemaBuilder/)

Code:

Provide the Code here:

Code

https://taskv2.microworkers.com/preview?type=task&campaignId...d34008c55c_HG&taskId=91d34008c55c_HG_5cea972404c396_27372140 Page 1 of 1

Figure 6.6 The ad we placed on the MicroWorkers platform to attract workers.

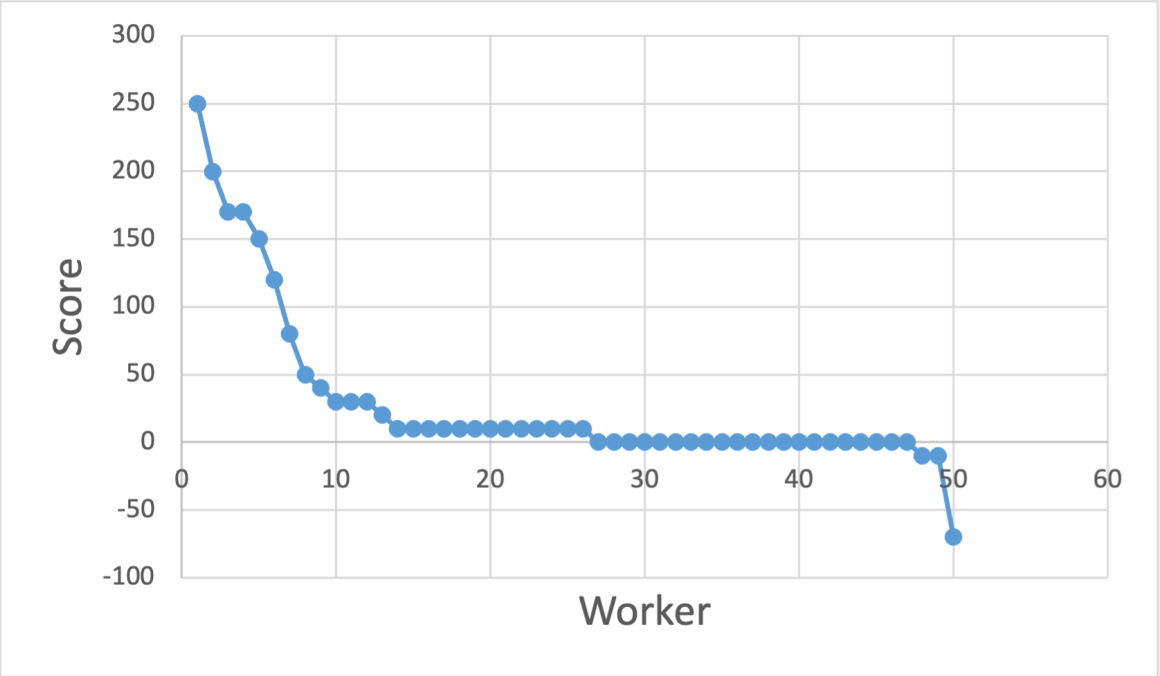


Figure 6.7 Workers score for the schema development process.

Schema	Sentences	Questions	Pronoun Targets
1	Erica called Jennifer on the phone because she was not responding to email.	Who was not responding to email?	Jennifer
	Erica called Jennifer on the phone because she was not able to email.	Who was not able to email?	Erica
2	If Rachel listened to Mrs. Sheila, she would have given her full marks.	Who would give full marks?	Mrs. Sheila
	Had not Rachel ignored Mrs. Sheila, she would have got full marks.	Who would have got full marks?	Rachel
3	The martial artist defended himself from the drug dealer because he was violent.	Who was violent?	The drug dealer
	The martial artist defended himself from the drug dealer because he was under attack.	Who was under attack?	The martial artist
4	George will congratulate Steve if he deserves it.	If who deserves it?	Steve
	George will congratulate Steve if he sees him.	If who sees him?	George
5	Chris helped Joe lift the bed because needed a favour.	Who needed a favour?	Chris
	Chris helped Joe lift the bed because he owed him a favour.	Who was owed the favour?	Joe

Table 6.1 Snapshot of the Contributors' developed schemas on WinoFlexi.

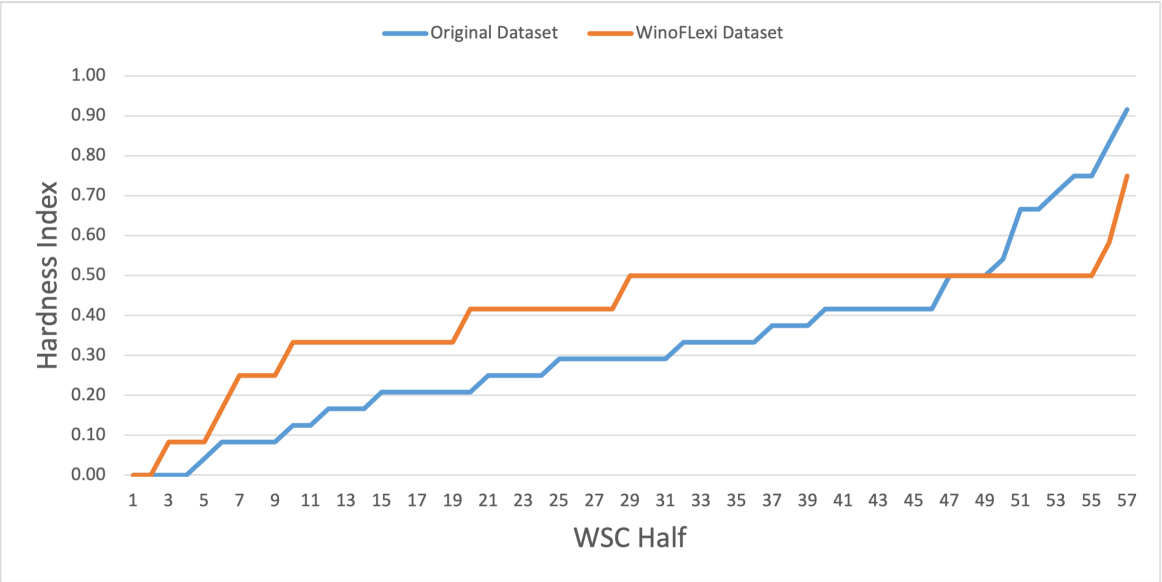


Figure 6.8 Hardness index variability across 57 halves of the original-dataset and 57 halves of the WinoFlexi-dataset. Each group is sorted based on the hardness index.

WinoFlexi's schemas in the same regard as the original WSC dataset's schemas. Then, we compare the hardness and the structure of WinoFLexi's schemas to that of the original WSC dataset.

Our baselines are three **coreference resolution** systems that were used in previous chapters, namely the Stanford CoreNLP system, Wikisense, and Knowledge Parser (K-Parser) (Sharma et al., 2015). Showing a positive correlation between the three systems' performance on the original and the WinoFlexi dataset would offer evidence that WinoFlexi can be used to develop schemas of good quality. We randomly selected 50 schemas (100 halves) from each dataset (see Table 6.2). On the original-dataset, Stanford CoreNLP correctly resolves 37% halves, incorrectly resolves 39% of them, and does not make any decision on the remaining 23%. On the WinoFlexi-dataset, it correctly resolves 44% halves, incorrectly resolves 44% of them, and does not make any decision on the remaining 12%. Wikisense correctly resolves 59% halves of the original-dataset, incorrectly resolves 31% of them, and does not make any decision on the remaining 9%. On the WinoFlexi-dataset, it correctly resolves 56% halves and incorrectly resolves 44%. K-Parser correctly resolves 38% halves of the original-dataset, incorrectly resolves 36%, and does not make any decision on the remaining 26%. On the other hand, on the WinoFlexi-library, it correctly resolves 37% halves, incorrectly resolves 37% of them, and does not make any decision on the remaining 26%. Comparing the results shows that the three systems' performance on the WinoFlexi-dataset is analogous to their performance on the original-dataset. According

	Original-dataset			WinoFlexi-dataset		
	Correct	Wrong	Unresolved	Correct	Wrong	Unresolved
Stanford CoreNLP	0.37	0.39	0.23	0.44	0.44	0.12
Wikisense	0.59	0.31	0.09	0.56	0.44	0
K-Parser	0.38	0.36	0.26	0.37	0.37	0.26

Table 6.2 Results of Stanford CoreNLP, K-Parser, and Wikisense on the original dataset and the WinoFlexi dataset.

to our results, the two datasets have correlation coefficients of 0.925 (Stanford CoreNLP), 0.987 (Wikisense), and 0.995 (K-Parser), respectively. The results provide evidence that our developed schemas are of the same or similar quality as the original dataset schemas.

Regarding the **Hardness-Metric** experiment, we randomly selected 57 halves of the WinoFlexi-dataset and compared their hardness index to that of 57 halves of the original-dataset taken from our previous work in Chapter 5. Figure 6.8 shows in more detail how the computed hardness index varies across halves, suggesting that, indeed, the two sets have comparable average hardness indices and analogous variability in their hardness indices. The general picture emerging from the analysis shows that although our workers were not initially familiar with the schema development process, through WinoFlexi's mechanisms, they were trained to design schemas of good quality. Furthermore, the data presented here provide evidence that the WinoFlexi schemas avoid Levesque et al. (2012) pitfalls, meaning that the schemas' questions are neither too obvious, nor are their answers not obvious enough.

Regarding the **Schema-Structure**, we compared the structure of all the crowd-generated schemas (WinoFlexi dataset) to that of all the expert-generated schemas (original dataset) as a way to determine if using crowdworkers sacrifices quality in exchange for scalability. For this experiment, we used the Sentence-Structure Identifier (see Chapter 5) to identify each developed half's sentence type and pattern. The results showed that 9% of the crowd-schemas are based on simple sentences, 8% on compound sentences, and 83% on complex sentences (see Figure 6.9). On the other hand, 41% of the expert-schemas are based on simple sentences, 14% on compound sentences, and 45% on complex sentences. Most of the developed schemas (both expert and crowd) are based on complex sentences. The expert-schemas that were designed with complex sentences had 30% "Cause/Effect", 8% "Comparison/Contrast", 1% "Place/Manner", 4% "Possibility/Condition", 18% "Relation", and 39% "Time" relationships. On the other hand, the crowd-schemas had 52% "Cause/Effect", 1% "Comparison/Contrast", 2% "Possibility/Condition", 1% "Relation", and 44% "Time" relationships. The results provide evidence that with *WinoFlexi*'s help, the crowd was able to develop quality schemas based on a variety of sentence patterns, similar to the expert-developed schemas. The results

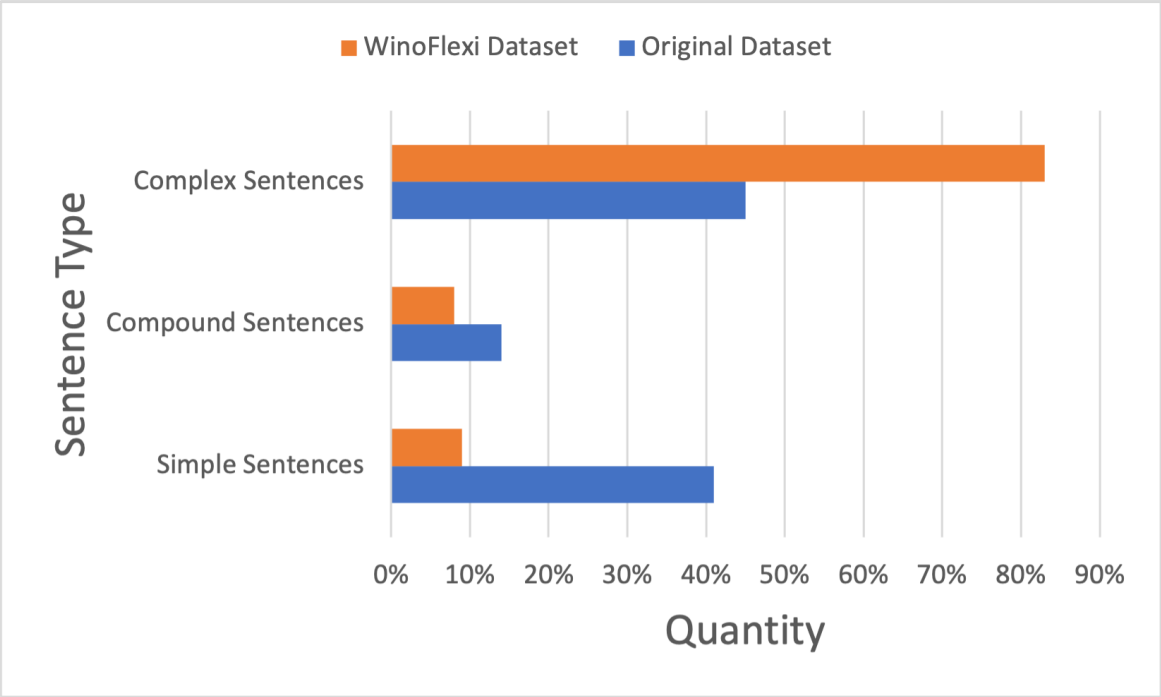


Figure 6.9 WinoFlexi's Dataset Sentences-Types.

point to a positive income of having an extended typology of this kind to instruct WinoFlexi's contributors to develop schemas of diverse types, which would also enable the qualitative analysis of the generated schemas. Additionally, the fact that crowd-schemas are not based on simple sentences, like the expert-schemas are (41%), might show that the crowd did not sacrifice quality in exchange for scalability. Another reason might be that complex sentences have a syntactic structure that is easier to indicate and use as a "skeleton" for WSC schema creation than simple sentences where a skeleton is not easily detectable. Still, considering the challenge difficulties, it seems that *WinoFlexi* can motivate and inspire researchers for the faster development of new schemas.

Qualitative Analysis

Based on the valid developed schemas and taking into account comments received from Contributors, we present below a qualitative analysis of *WinoFlexi*'s outputs. We begin with observations we obtained regarding the evaluation procedure and continue with comments we received and relate to factors like inspiration, creativity, enjoyment, and curiosity of WinoFlexi's workers.

Certain WinoFLexi's outputs suggest that the schema **evaluation process** might need to be optimized, and more than one Evaluator might need to evaluate schemas. For instance,

the following was mistakenly considered as a valid schema: *1.) Sentence: Karen loved going to salons to get her nails done. They always looked so nicely decorated. Question: What looked nicely decorated? Answers: The Salons, The Nails. 2.) Sentence: Karen loved going to salons to get her nails done. They always looked so nicely manicured. Question: What looked nicely manicured? Answers: The Salons, The Nails*. This schema cannot be considered valid because the second half is resolvable with selectional restrictions; salons cannot be manicured.

One of the problems in the schema development process is the lack of **inspiration and creativity**. It seems that the collective intelligence of the crowd can eliminate those factors. For instance, the workers developed schemas which are based on a variety of subjects, like cartoon heroes (e.g., spiderman, hulk), animals (hyenas, rabbits, dogs, zebras), hospitals (e.g., psychiatrists, medications), people in general (boolies, fights, burglars, homework, schools), things (cards, drains, golf). The following schema is a great example of a *scene* between Spiderman and Hulk: *1.) Sentence: Spiderman spun his web around the Hulk because he was falling. Question: Who was falling? Answers: Hulk, Spiderman. 2.) Sentence: Spiderman spun his web around the Hulk because he was annoyed. Question: Who was annoyed? Answers: Hulk, Spiderman*.

Finally, it seems that certain workers were motivated by an intrinsic incentive such as **enjoyment and curiosity** for new knowledge and not only from potential rewards (Elmalech et al., 2016). More broadly, the following comments motivated us to continue the building of WinoFlexi:

- Contributor Comments (username: Member0xx): ...*"I am terribly sorry, on my most recent schema I accidentally selected the wrong option. The schema is about putting a shirt in the dryer. I hope it is something you can fix. Thank you for your time and allowing a platform to develop these schemas, I very much enjoy trying to figure out new ways to create a valid schema..."*

- Contributor Comments (username: Member1xxxxx): ...*"Hey I noticed your company paid me for my clickworker submissions but I did them incorrectly accidentally and have endeavoured to submit correct Schemas since that time - Just a heads up!..."*

- Evaluator Comments (username: Member_7xxxxxxxx): ...*"Your schema answers are too obvious, the second part of the sentence adds nothing to the schema..."*

6.2.5 Expert Analysis

Acknowledging that the crowd evaluation process is primarily subjective based on the information provided regarding WinoFlexi's evaluation factors, we present a more detailed analysis of the collected schemas after sending and receiving feedback from an expert in the field (Davis, 2021).

The results showed that the number of valid schemas was significantly smaller than our analysis. Specifically, although 83 schemas were found to be valid, overall, a handful of them compares favorably in terms of quality to that of WinoGrande (Sakaguchi et al., 2020) —for instance, "Sentence: Kevin paid/invoiced George for the job he did. Question: Who did the job? Answers: George, Kevin" and "Sentence: Ronnie pushed past Bart because he was in the way/a hurry. Question: Who was in the way/hurry? Answers: Bart, Ronnie".

Regarding the structure of the valid schemas, 60 of them seem to follow a pretty fixed form "something about X and Y because he/she/it is Q" where Q is generally an adjective, occasionally a participle or prepositional phrase, though there is nothing wrong with that as WSC273 contains versions of this.

According to the feedback received, in the following paragraphs, we provide a thorough analysis of why a number of valid schemas should have been considered invalid in the first place.

Ambiguous Schemas

Seventeen schemas consist of at least one ambiguous half, where the two possible pronoun targets are more or less equally likely to be the referent of the definite pronoun. For example, in the schema "Sentence: Callum tripped Joe because he was angry/clumsy. Question: Who was angry? Answers: Callum, Joe" the half with the special word "clumsy" is ambiguous, meaning that the definite pronoun can refer to both pronoun targets. For another, in the schema "Sentence: Charlie called Ed because he was smart/less intellectually inclined. Question: Who was smart/less intellectually inclined? Answers: Ed, Charlie", in each half, the definite pronoun can refer to both pronoun targets.

Schemas without Pronouns

Ten schemas consist of at least one half that has no pronoun. For instance, we can see the lack of pronouns in the half "Sentence: Emily needed Peggy to help with the assignment so everyone contributed. Question: Who needed to contribute? Answers: Peggy, Emily".

Schemas Resolved by Selectional Restrictions

Four additional schemas were identified to have at least one half that can be resolved by selectional restrictions or considerations of frequency. For example, in the following schema, "Sentence: I hung the painting in the living room in order to display/decorate it. Question: To display what? Answers: painting, the living room" the pronoun target "paintings" cannot be decorated. Another example is the schema "Sentence: The bull charged the dog because it was angry/barking. Question: Who was angry? Answers: Bull, Dog", because the "bulls" cannot bark.

Difficult to Understand Schemas

Five schemas were found to have at least one half that is hard to understand with either resolution of the pronoun. For instance, in the schema "Sentence: Carol chose to play against Laura because she was slow/fast. Question: Who was slow/fast? Answers: Laura, Carol" the second half (with the special word "fast") is hard to distinguish whether it refers to Laura or Carol. Another example is the schema "Sentence: Erica explained to Sue that she did not have any fun at the party/that she was one of her closest friends. Question: Who did not have any fun/was one of her closest friends? Answers: Erica, Sue", where the second half was found difficult to understand.

Similarly, four additional schemas were classified as too confusing to determine the best pronoun target of each half, meaning that the same actual answer could be given to both halves. For instance, in the schema "Sentence: Erica assisted Jennifer with her homework because she wanted/needed help. Question: Who wanted/needed help? Answers: Jennifer, Erica" the two pronoun targets are equally expected to be the referent of the definite pronoun.

Repetitive Schemas

Three schemas seemed to be repetitions with trivial changes though this does not disqualify their validness:

- "Sentence: The cat tried to catch the mouse, but it was too slow/fast. Question: Who was slow/fast? Answers: The cat, The mouse".
- "Sentence: The spider caught the fly because it was determined/lazy. Question: Who was determined/lazy Answers: The spider, The fly".
- "Sentence: The Spider caught the Fly because it was sneaky/distracted. Question: Who was sneaky/distracted Answers: The spider, The fly".

Schemas written with Incorrect English

Two schemas were found to be written in incorrect English. For instance, the schema "Sentence: Jeff borrowed his instruction booklet to John because he was helpful/confused. Question: Who was helpful/confused? Answers: Jeff, John", and the schema "Sentence: The car crashed into the tree because it was in the way/reckless. Question: Who was in the way/reckless? Answers: The tree, The car".

Syntactically Resolvable Schemas

Two schemas were characterized as easy to disambiguate by using syntactic considerations. For instance, in the schema "Sentence: Whilst running past a trash can John kicked it over, the next day Andrew cleaned it up/ignored the mess. Question: Who kicked over/ignored the trash can? Answers: John, Andrew" both halves can be syntactically resolved as the answers are clearly stated in each sentence.

Gender Resolvable Schemas

Finally, a schema was found easy to disambiguate just by checking the gender agreement of each pronoun target with the definite pronoun. For instance, the schema "The jockey liked the horse because it ran fast/he made it run fast. Question: Who was fast? Answers: horse, jockey" can be easily resolved because in the first half, the definite pronoun "it" can only refer to the "horse" whereas in the second half, "he" can only refer to "jockey".

Final Thoughts

Concerning the feedback received, most of the remarks can be addressed, as WinoFlexi's mechanisms can easily be modified to enhance the schema development process.

In our defense, ambiguous and difficult-to-understand schemas relate to the ambiguity of language, which is not easy to address. People think in mental models using visual or auditory images and use language to communicate ideas, meaning that words and thoughts cannot be the same (Pinker, 2005). In principle, even sentences with slight differences have different meanings that each worker might interpret differently, directly relating to the difficulty of the challenge itself.

Regarding the problem of gender and syntactically resolvable schemas, this might be addressed by increasing our workers' training and testing period. Concerning the development of schemas without definite pronouns, we can update WinoFlexi's mechanism to include additional tests to identify and notify workers if a definite pronoun was not included in the designed schemas.

Consequently, to address the problem with the development of schemas with incorrect language, we can request workers who have better knowledge of the English language. In this regard, a better qualification task from the Microworkers platform or by WinoFlexi's mechanisms might reduce the problem.

6.2.6 Discussion

After presenting our work and given that, we began our discussion by introducing the WinoGrande dataset (Sakaguchi et al., 2020), below, we present the key differences between the two datasets and methodologies used. Firstly, given that WinoFlexi is a complete system that can be used independently or connected to crowdsourcing platforms, via WinoFlexi's mechanisms, the MW workers not only completed a training task that familiarized them with the process but were continuously evaluated based on test questions displayed during the schema development process. Compared to the WinoGrande dataset, where workers wrote pairs of sentences using anchor words, the WinoFlexi schemas were created by workers based on controlled mechanisms that led them to design quality schemas consisting of pairs of sentences and questions. For instance, compare the following WinoGrande schema *1.) Sentence: Ian volunteered to eat Dennis's menudo after already having a bowl because _ despised eating intestine., option1: Ian, option2: Dennis 2.) Sentence: Ian volunteered to eat Dennis's menudo after already having a bowl because _ enjoyed eating intestine., option1: Ian, option2: Dennis* to WinoFlexi's results *Sentence: Erica called Jennifer on the phone because she was not responding to email/not able to email., Question: Who was not responding to email/not able to email? Answers: Jennifer, Erica.* Secondly, in WinoGrande, the authors enhanced the creativity of their workers by priming them by a randomly chosen topic as a suggestive context. However, in WinoFlexi, the whole procedure was controlled by various mechanisms, like the ban-score, rewards, and the schema hardness, which enhanced workers' inspiration, creativity, enjoyment, and curiosity. Thirdly, compared to WinoGrande, where workers were advised to keep twin sentence length between 15 and 30 words while maintaining at least 70% word overlap between a pair of twins, in WinoFlexi, this was automatically controlled by the system similarity and lemmatization mechanisms. Fourthly, in comparison with WinoFlexi, where workers were allowed to develop schemas of various domains, in WinoGrande, workers were advised to design schemas of two domains: (i) social commonsense, which involves two same-gender people with contrasting attributes, and (ii) physical commonsense, which involves physical objects with contrasting properties. Finally, compared to WinoFlexi, which characterizes workers with two distinct roles to ensure that the developed schemas follow the WSC rules, with WinoGrande, the authors validated each

collected schema through a specific set of three crowd workers —to ensure that humans could easily infer pronoun referents in these sentences.

6.3 The Winventor Approach

6.3.1 Introduction

In this section, we present Winventor, a system able to promote the original goals of the WSC through the development of Winograd instances (schemas/halves). Winventor is a machine-driven approach that tries to automate the schema development process to considerably help humans in the development task. It combines NLP and deep learning into a flexible system able to produce new Winograd instances, which could be used to enhance the creativity and motivation of human experts for the development of schemas. In this regard, Winventor can facilitate the continuous development of Winograd schemas. To the best of our knowledge, this is the first published work to report results on the feasibility of this approach.

Winventor's architecture is based on three major approaches: NLP, deep learning, and a blended approach. In each case, we undertake several experiments regarding the a priori appropriateness of our system as a schema development mechanism. Our empirical evaluation suggests that the blended approach, which combines deep learning and NLP, can provide us with more schemas than the other two approaches. The design of appropriate systems for our particular task and the evaluation of the developed Winograd instances is the focus of the rest of this section.

6.3.2 Winventor's High-level Architecture

We begin with a high-level overview of Winventor by presenting how the engine works (see Figure 6.10). If Winventor cannot develop a Winograd schema, it only generates a Winograd half that consists of a sentence, a definite pronoun, a question that indirectly points to the definite pronoun, and the two pronoun targets. Schemas that do not obey all constraints are known as "Winograd Schemas in the broad sense" (Levesque et al., 2012). In this regard, we developed Winventor to work in two different modes: strict or relaxed. With the strict mode enabled, Winventor develops schemas that strictly follow the WSC rules, whereas, with the relaxed mode, it may also develop schemas where the pronoun targets do not have to share the same gender.

At first, Winventor loads an English sentence to evaluate if it can develop a schema. Winventor utilizes the sentence to output the definite pronoun and the two pronoun targets with one of the three specified approaches: NLP, deep learning, and the blended approach (see

BlackBox in Figure 6.10). If this is not possible, the current sentence is rejected. Otherwise: i) it proceeds with the question development, using a tool from the literature; ii) it constructs the first half by placing together the sentence, the question, and the two pronoun targets; iii) it finds the special word in the first sentence, generates the question, and develops the second half. More details on this procedure are given next.

Source of Sentences

To develop schemas automatically, it is necessary to have access to a source of sentences. Winventor can use any source, local or online, which can provide a bulk amount of English sentences. In its current version, Winventor is built on an extensible framework that allows access to a broad collection of nearly 88 million sentences from the English Wikipedia (see Wikipedia corpus in Chapter 3).

Question Generator

One of the most challenging parts of the WSC is to come up with appropriate questions. The question generation task is a very challenging process that dates back to 1976 (Wolfe, 1976). According to Levesque et al. (2012), while doing so, we must avoid two major pitfalls. The first pitfall concerns questions whose answers are, in a certain sense, too obvious. The second and more troubling pitfall concerns questions whose answers are not obvious enough.

To tackle this, Winventor uses the Heilman and Smith (2009) question generator[2], a system able to generate questions based on a given piece of text. This question generator is freely available, easily customizable, and, at the same time, able to generate questions with a ranking strategy. Specifically, *Winventor* uses the question generator with the *"–keep-pro and –just-wh"* flags enabled. *Keep-pro* keeps questions with unresolved pronouns, and *Just-wh* excludes boolean questions from the output. Next, it selects the pronoun targets related to the pronoun given as the answer to the best question. By way of illustration, in the following example, *"The cat caught the mouse because it was clever"*, Winventor, via Heilman and Smith's question generator returns the following questions: i) "What caught the mouse because it was clever, the cat, 2.32"; ii) "What did the cat catch because it was clever?, null,2.23"; iii) "What was clever?, it, 0.97". In the end, Winventor selects the third question, as it is the only one that has as an answer the definite pronoun *it*.

[2]http://www.cs.cmu.edu/~ark/mheilman/questions/

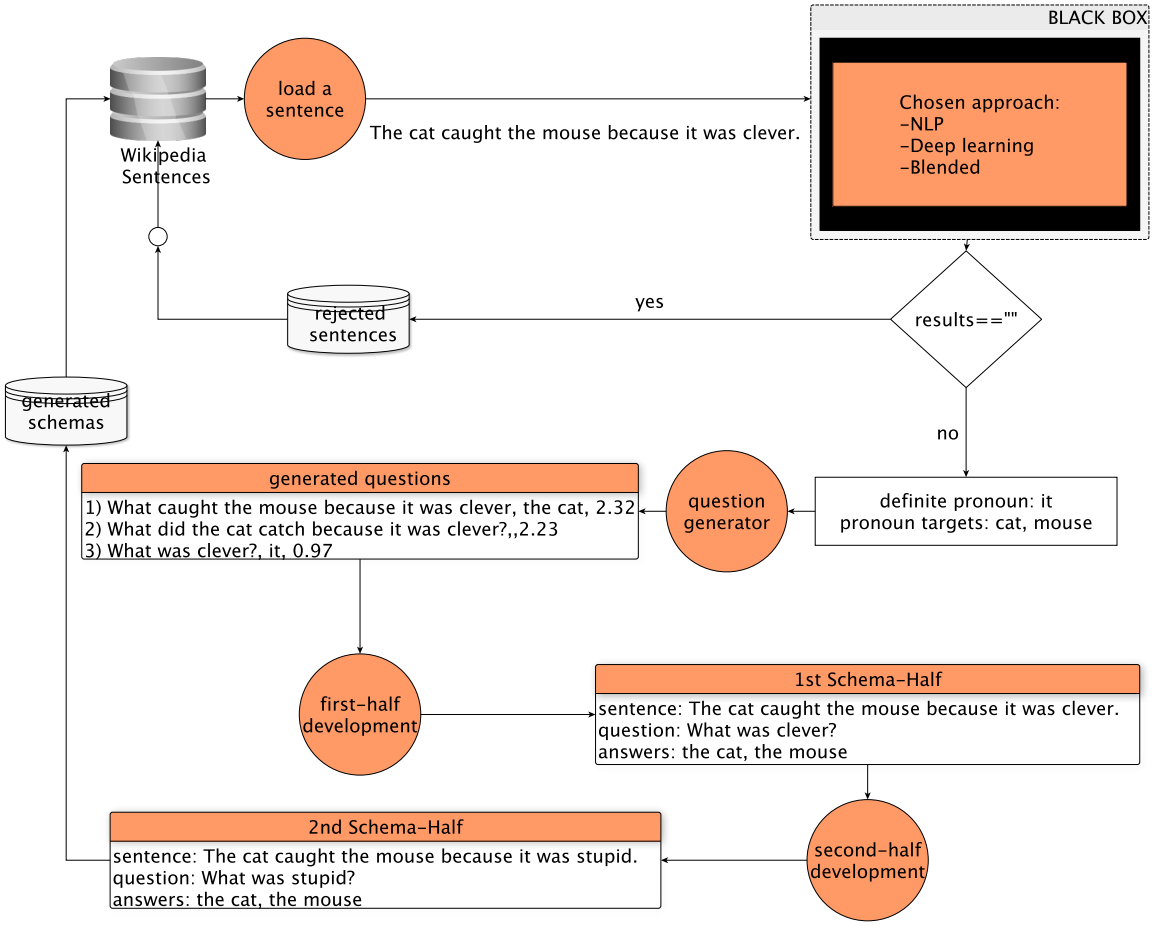


Figure 6.10 Winventor's high-level architecture: A system that automates the schema development process.

Developing the Winograd Half

The next step for Winventor is the development of Winograd halves, meaning, pairs of sentences, questions, and pronoun targets. For each sentence and depending on the approach used, Winventor might construct several halves, the number of which relates to the question generator results and the possible pronoun-target pairs. In this regard, for each valid pronoun-target pair, Winventor develops several halves, reordered by their significance (see *first-half development* in Figure 6.10).

Developing the Winograd Schema

Winventor develops schemas by considering that they are constructed so that there is a special word in each sentence, which, when replaced by another word, the answer also changes (Morgenstern et al., 2016). Hence, for every Winograd half, it considers the following: i) it parses the question to identify the special word, which is a verb/adjective that participates in the questions' triple relation (e.g., the word clever from the question "Who was clever"); ii) it returns the antonym of the special word, found in the previous step (e.g., from "clever" to "stupid"), and iii) it modifies the returned word, in the question and the sentence, to match the tense of the second half (see *second-half development* in Figure 6.10). Regarding the triples, these are semantic scenes of the type *subject, verb, object* that are created through the sentence/question's subjects and objects (see Chapter 3). For instance, the triples [cat, caught, mouse] and [who, was, clever], which were used for the development of the schema in Figure 6.10, were created from the parser's *nsubj and dobj* relations *(abbreviations of "nominal-subject" and "direct-object"*). Obviously, one could make the sentences sound more natural-sounding by taking into account work in progress in the Natural Language Generation (NLG) field (Gatt and Krahmer, 2018).

Below, we will show how Winventor analyzes Wikipedia sentences to select the definite pronoun and the pronoun targets, based on three approaches. In the first part, we will discuss how the engine handles its semantics to develop schemas with various NLP tools, and in the second part, we will show how deep learning comes into play. In the third part, we will show how the blending of the two approaches can be used to enhance the schema development process.

6.3.3 The NLP Approach

Winventor uses various NLP tools to determine, in a way, the meaning of each sentence (Chowdhury, 2003). This approach helps select the definite pronoun and the pronoun targets based on the semantic analysis of a given piece of text. For instance, via various NLP tools,

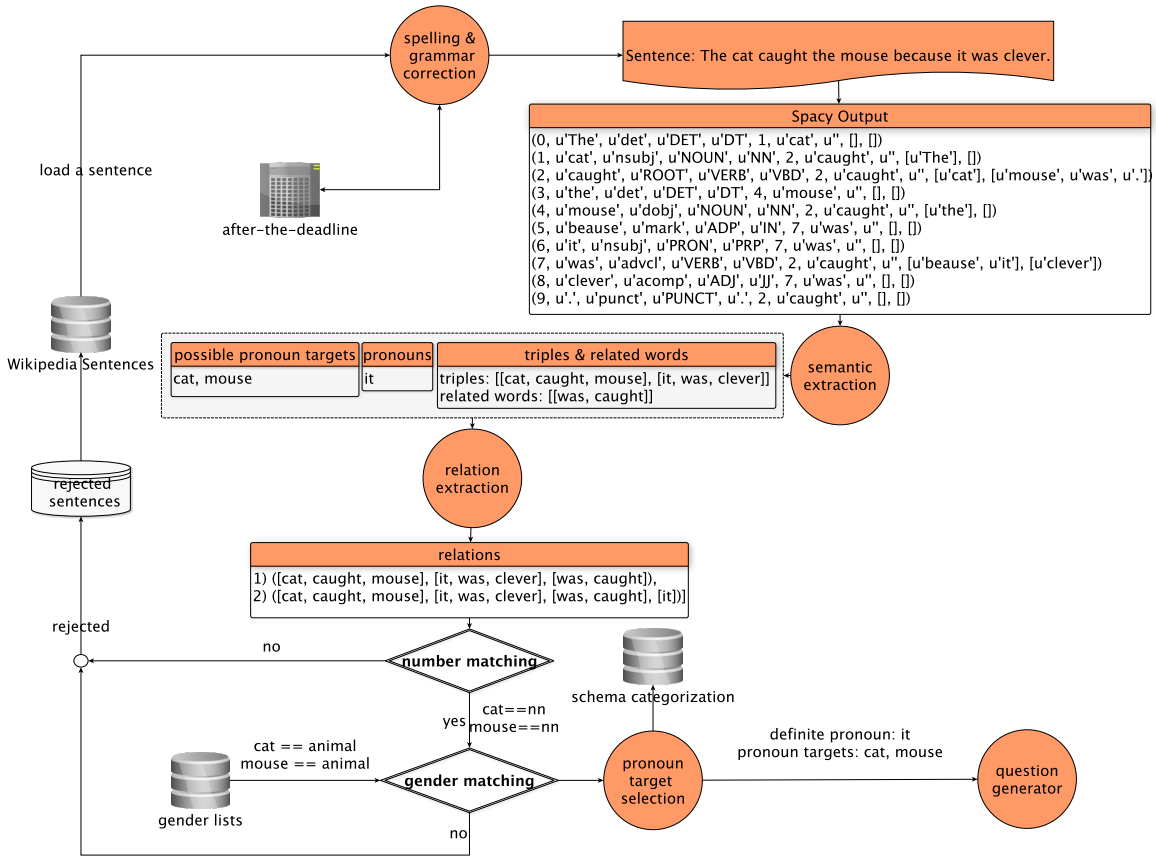


Figure 6.11 A schema development process by Winventor using various NLP tools. The NLP section ends just before the question generator comes into play (see question-generator in Figure 6.10).

Winventor will acquire sentences with good structure to select pronoun targets that agree in gender, number and participate in relations with other words. In the sequel, we will introduce the NLP components of Winventor by presenting how it generates Winograd instances from scratch (see Figure 6.11).

Spelling Correction

It is well-known that sentences found from online sources, like Wikipedia, might suffer from abbreviations, spelling errors, and misspellings of words. For instance, it was found that the percentage of misspellings of words on Wikipedia, relative to content, consistently increases year after year (Stacey, 2011). To avoid these kinds of problems, Winventor makes use of two tools from the literature. The first is the Google language-detection[3] library, which helps Winventor acquire only English sentences. The second is a download version of

[3]https://pypi.org/project/langdetect/

After-the-Deadline[4] language-checker, which automatically corrects spelling and grammar errors. Tools like After-the-Deadline offer efficient and effective ways of enhancing grammar accuracy and learning (Mudge, 2010).

Word Relations

With the term word-relations, we refer to semantic relations that can be concluded from a given text. While this task, which is necessary for developing schemas, is trivial for humans, it is quite challenging and difficult for machines. As we have seen in previous chapters, the semantic relations of any given piece of text are considered good if they can output essential relationships between the events and their participants (e.g., relations between subjects and objects in a sentence), albeit there is still no clear path to this goal (Schubert, 2015). To build good relations, we must consider various facts, like grammatical role, number, gender, and syntactic structure, that dependency parsers can give (Budukh, 2013). As in our other developed tools, Winventor utilizes the spaCy dependency parser to develop semantic relations from the Wikipedia sentences.

Through spaCy, Winventor parses each sentence to develop triples, related-words, and pronoun relations. *Related-words* are based on verbs that have a direct relation between them. For instance, the *caught-was* relation shows an indirect connection of the *nsubj* cat and the *dobj* mouse to the *adjective* clever (see *2nd and 7th line of the spaCy output* in Figure 6.11). *Pronoun-relations* are relations where the pronoun targets (nouns or proper-nouns) are related to other words, via pronouns (see *relations* in Figure 6.11). If at least one pronoun and two nouns or two proper nouns exist (possible pronoun targets), we proceed to the next step. Otherwise, we proceed to the next sentence.

Pronoun-Target Selection

A challenging task for Winventor is to obtain the possible pronoun targets from each examined sentence. According to what the challenge dictates (Levesque et al., 2012), the possible pronoun targets should be either a pair of nouns or proper-nouns that agree in gender and number. Winventor's approach to discerning a list of possible pronoun targets includes the following: i) it utilizes spaCy's entity recognition system to search for proper nouns, ii) it searches some pre-downloaded gender-lists to find nouns that have the same gender, and, iii) via spaCy dependency parser, it selects only nouns and/or proper-nouns that agree in number (compound nouns are selected accordingly). The final result is to develop as many schemas

[4]http://www.afterthedeadline.com.

as it can from each examined sentence. For each developed schema, Winventor keeps track of three variables/flags, showing the relations that govern the pronoun targets:

- numberAgreement: This variable equals 1 if the two nouns/proper-nouns agree in number, otherwise 0.
- genderAgreement: Likewise, this equals 1 if the two pronoun targets have the same gender.
- pronounGenderAgreement: This variable equals 1 if the two pronoun targets' gender agrees with the target pronoun, otherwise 0. To complete this task, we consider the following: The third-person singular personal pronouns, *he/him/his*, refer to the masculine gender, whereas *she/her(s)* refer to the feminine gender. On the other hand, pronouns like *they/them/their(s)* refer to the neutral gender, and the pronouns *it/its* refer to the neuter gender (in the case of companion animals, the pronouns *he/she* may also be used).

Pronoun-Target Appropriateness

In order to identify the appropriateness of each pronoun target pair, Winventor does the following: i) as previously mentioned, it keeps track of the number, gender, and the pronoun-gender agreement, ii) it stores the number of the triple relations that the pronoun targets participate in, and iii) it utilizes the Mitkov (1998) aggregation score, which can create a ranking list of nouns, according to some preferences. Mitkov's work showed that when we have limited background knowledge, like in our case, we can consider five salience indicators, identified empirically, to select the best pronoun targets:

- Definiteness: *Definiteness* refers to definite nouns (specific and non-general nouns), meaning that these kinds of nouns should get a higher preference compared to other nouns. Definite noun phrases' score equals 0, whereas indefinite ones are penalized by -1. For instance, in the sentence "The man couldn't lift his son because he was sick", the definite pronoun [he] has more chances to refer to the "man" than the "son" since "the man" is a definite noun [man: 0, son: -1].
- Indicating-verbs: *Indicating verbs* relate to nouns that are followed by verbs that are members of a specific Verb set (e.g., discuss, consider, investigate). These nouns' score equals 1, otherwise 0. According to Mitkov (1998), the verbs listed above are good indicators because of the importance of the noun phrases that follow them. For instance, in the sentence "A good map will show the building because it is famous."

the “building” gets a higher preference than the “map” because the building follows an indicative-verb (show) [map: 0, building: 1].

- Lexical-Reiteration: *Lexical Reiteration* refers to repeated synonymous noun phrases where they get a higher preference. A noun’s score equals 2 if it is repeated twice or more, 1 if it is repeated once, and 0 if not. For instance, in the sentence “Although there are various kinds of maps, a good map will show the building because it is famous.”, the “map” gets a higher preference than the “building” because the word “map” is repeated once [map: 1, building: 0].

- Non-prepositional: *Non-prepositional* nouns are given a higher preference than prepositional nouns. A non-prepositional noun’s score equals 0, whereas a prepositional noun’s score equals -1. In this regard, noun phrases, which are parts of prepositional phrases, are penalized because they are usually indirect than direct objects (Mitkov, 1998). Take, for example, the following sentence: “You must insert the sd-card into a camera making sure at the same time it is compatible”. Here, as the “camera” is a prepositional noun, it is penalized with -1 [sd-card: 0, camera: -1], meaning that the definite pronoun [it] more likely refers to the sd-card than the camera.

- Collocation: *Collocation* refers to nouns that have identical collocation patterns with the definite pronoun. Specifically, these nouns get a higher preference than other nouns—collocation nouns’ score equals 2, otherwise 0. For instance, in the sentence “To start the engine, you have to press the button. In case you want to switch off the engine, you have to press it again”, the “button” gets a higher preference than the “engine” because the “button” has a similar collocation to the definite pronoun [it] [engine: 0, button: 2].

- Immediate-Reference: *Immediate-Reference* refers to sentences that have a specific structure, where noun phrases immediately placed before specific conjunctions (and, or, before, after) followed by pronouns are more likely to be the referents of those pronouns —an immediate-noun phrase has a score of 2, otherwise 0. For instance, in the sentence “The cat caught the mouse and held it firmly till the end.” the “mouse” is more likely to be the pronoun’s [it] referent [cat: 0, mouse: 2]. For another, this indicator can be seen as a modification of the collocation indicator and can also be found in imperative constructions (Mitkov, 1998).

Completing the Schema

As shown in section 6.3.2, after the selection of the best pronoun target, Winventor parses the sentence through the Heilman and Smith (2009) question generator and selects the one that has as answer the definite pronoun. Finally, it develops the two halves, constructs the schema, and adds it to the schema database. Based on this approach, each developed schema is automatically classified into predefined categories and added to a schema-categorization DB (see Figure 6.11). The categorization is done according to each sentence subject (e.g., Schwarzenegger - terminator - protection, birds - food) and the pronoun target pairs' types (e.g., gpe, gerund, loc, country, facility, norp, org, etc.). Additionally, *Winventor* keeps track of the rejected sentences with the following flags (see rejected-sentences in Figure 6.11):

- *Nouns and proper-nouns have not been found.*
- *Target Pronoun relations have not been found.*
- *Questions have not been formed.*
- *Not an English sentence.*
- *This was artificially created for previous WSC.*

6.3.4 The Deep-Learning Approach

As we have seen in Chapter 5, deep learning refers to a class of different techniques that allow computational models to learn representations of data (LeCun et al., 2015). In this regard, we aim to train three deep learning models to help Winventor in the schema development process. Specifically, we train: 1.) the sentence model for selecting sentences, 2.) the pronoun model for selecting the definite pronoun in each examined sentence, and 3.) the pronoun-targets model for selecting the best pronoun target pair in each examined sentence. For developing a schema/half, our algorithm starts with the sentence model to select an appropriate sentence. It then utilizes the pronoun model to select the best definite pronoun from the previously selected sentence and continues with the pronoun-targets model to select the best possible pair of answers. In the sequel, we will introduce the deep learning models with the datasets used for training and testing purposes (see Figure 6.12).

Dataset Preparation

Deep learning algorithms cannot understand the meaning of the text but only map the statistical structure of written language, which is supposedly sufficient to solve simple textual

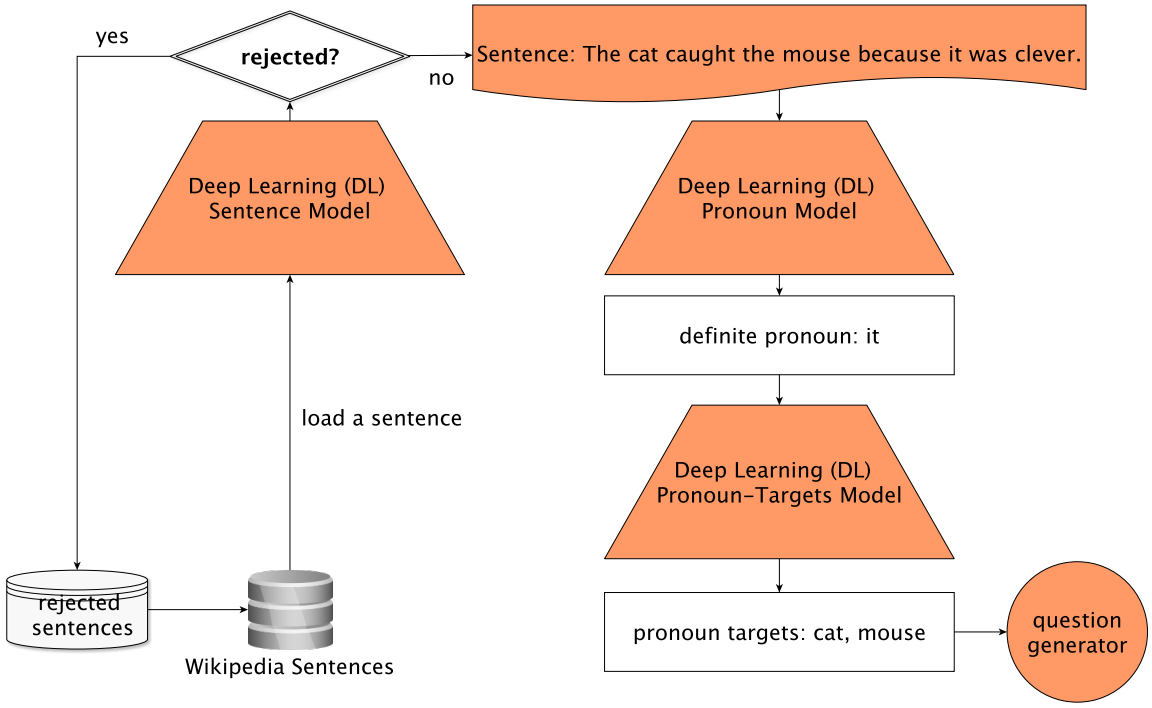


Figure 6.12 A schema development process by Winventor using deep learning. The deep learning section ends just before the question generator comes into play (see question-generator in Figure 6.10).

	Part of Speech Tagging
sentence	The city councilmen refused the demonstrators a permit because they feared violence.
part of speech	DET NOUN NOUN VERB DET NOUN DET NOUN ADP PRON VERB NOUN
synonym-positive example	DET ADJ NOUN NOUN VERB DET NOUN DET PROPN NOUN ADP PRON VERB NOUN
synonym-negative example	DET NOUN VERB PART DET DET ADP PRON VERB

Table 6.3 A sentence transformation example for developing the training and testing dataset of our sentence model.

tasks (Chollet, 2017). On the other hand, we know that in problems where data is limited, deep learning often is not an ideal solution (Marcus, 2018). In this regard, we employed a data synthesis/augmentation procedure to increase our training data size. To that end, we used the original WSC dataset (Levesque et al., 2012), which at the time of writing consisted of 150 schemas and ended up with 30,000 schemas.

The Sentence Model

This model utilizes a classifier that is responsible for selecting appropriate sentences for the development of schemas (see *DL Sentence Model* in Figure 6.12). Given any English sentence, our sentence model returns a value in the range of 0-1, where values > 0.5 indicate high-grade (suitable) sentences for developing schemas. Valid or high-grade sentences are eligible for further processing to develop schemas, whereas non-valid are not.

To train our classifier, we used training data with positive and negative examples. Positive examples refer to sentences used in developing the WSC original dataset (Levesque et al., 2012). In contrast, negative examples refer to sentences that cannot be used in the development of schemas. To increase the number of positive-examples we proceeded as follows: 1.) we parsed each sentence and removed the punctuation characters, 2.) for every noun, adjective, verb, and adverb, we developed a list with their synonyms, 3.) based on a random combination of their synonyms, we developed a list of new sentences, and, 4.) via spaCy, we replaced the words of each examined sentence with their part of speech (part-of-speech tagging). Through the part-of-speech tagging, our model does not need to use *knowledge transfer* between various domains, which is a characteristic feature for several deep learning approaches (Liu et al., 2016). Regarding the negative examples, for every positive sentence, we developed a negative one: i) by randomly removing some words, and ii) by randomly reordering its tagging (see Table 6.3).

The Pronoun Model

A critical problem within the schema development process is selecting the definite pronoun, which directly relates to selecting the pronoun targets. To that end, we developed the pronoun

model, which is responsible for selecting the definite pronoun in sentences returned by the sentence model (see *DL Pronoun Model* in Figure 6.12). Given any tagged-English sentence with multiple pronouns, this model returns the best possible pronoun, which could be used as our definite pronoun. Specifically, for each sentence with a (marked) pronoun, this model returns a confidence score in the range of 0-1; the higher the score, the higher the confidence for the specific pronoun.

To increase our training set, we have followed a similar procedure to the previous model. Regarding the construction of the positive examples, we have used the valid sentences from our sentence model but with the position of the definite pronoun marked. For instance, for the half's sentence, *The city councilmen refused the demonstrators a permit because they feared violence* our algorithm would return "DET NOUN NOUN VERB DET NOUN DET NOUN ADP <PRON> VERB NOUN". We have followed a similar procedure for the construction of the negative examples, where, for each positive sentence, we build a new negative one with its tagging shuffled. For instance, in our previous example, this would result in "DET NOUN DET NOUN <PRON> NOUN VERB ADP NOUN VERB DET NOUN".

The Pronoun-Targets Model

This model is responsible for selecting the best pronoun target pair (answers) in sentences, previously selected by the pronoun model (see *DL Pronoun-Targets Model* in Figure 6.12). Recall that the WSC is about resolving the definite pronoun to one of *two* possible pronoun targets in each schema. Hence, in each examined sentence, this model aims to output the best answer pair to construct the schema. Given any tagged English sentence with two parts marked, this model returns a confidence score in the range of 0-1 that indirectly shows the best pair for developing the schema.

For training purposes and specifically for building our positive examples, in all of the synonym sentences, a pronoun target pair was marked. For instance, in the example used in our previous models, our algorithm would return "DET <NOUN NOUN> VERB DET <NOUN> DET NOUN ADP PRON VERB NOUN", with the position of the two pronoun targets marked. We have followed a similar procedure for the construction of the negative examples, where, for each positive sentence, we build a new negative one with its tagging shuffled.

Completing the Schema

We continue to discuss how Winventor develops schemas via the deep learning approach. At the start, each Wikipedia sentence is validated by the sentence model, where for every

valid sentence (> 0.5), it proceeds to the next step to search for the definite pronoun (see Algorithm 2). Winventor replaces every sentence word by its part-of-speech, marks the pronoun (<PRON>), and parses it through the pronoun model to retrieve its score; this process is repeated for every pronoun in the sentence, and at the end, it selects the pronoun with the biggest score. The next step is to find the best pronoun-target pair of the sentence that indirectly relates to the definite pronoun. To that end, Winventor randomly creates all the combinations of two, three, and four words. Then, for every combination, it marks the combination's words in the sentence (part-of-speech) and parses it through the pronoun-targets model to retrieve its score. In the end, it selects the best pair, which is the pair with the highest score. After selecting the sentence, the definite pronoun, and the pronoun target pair, Winventor develops the two halves, following the same procedure stated in the previous sections (see Section 6.3.3). The only difference within this approach is that each developed schema cannot be automatically classified into predefined categories in order to be added to the schema-categorization DB.

Algorithm 2 Schema development via deep learning

```
1: sentences = loadDatasetHalf1Sentences (RahmanNg)
2: for sentence in sentences do
3:     validSentence = checkSentMODEL (sentence)
4:     if validSentence <= 0.5 then continue
5:     bestPronoun = findTheBestPronoun (sentence, pronounMODEL)
6:     bestAnswerPair = findBestAnswerPair (sentence, answerMODEL)
7:     question = buildQuestion (sent)
8:     half1 = finalizeSchema (sent, bestPronoun, bestAnswerPair, question)
9:     half2 = buildHalf2 (sent, bestPronoun, bestAnswerPair, question)
10: end for
```

6.3.5 The Blended Approach

This section describes how we blend the NLP and the deep learning approach with the ultimate goal of developing a more efficient and effective solution. In particular, we modified the pronoun-target selection process based on factors described in the previous sections (see Algorithm 3). Specifically, we replaced the pronoun-targets model with the gender, number, pronoun-gender, and triple factors to select the best answer pair (see Figure 6.13).

Thus, the blended approach proceeds as follows: 1.) via the sentence model, it parses Wikipedia sentences to select an appropriate sentence for the development of a schema; 2.) through the pronoun model, it returns the definite pronoun of the examined sentence; 3.) from the sentence it selects only nouns or proper-nouns and builds all the possible combinations

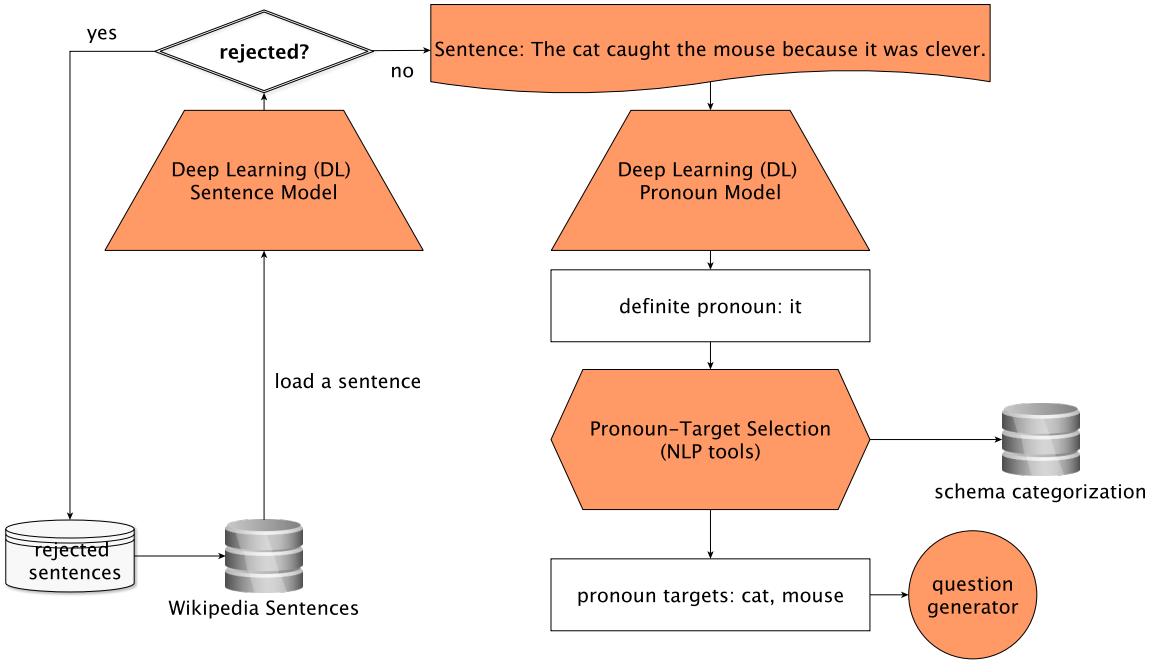


Figure 6.13 A schema development process by Winventor using deep learning and various NLP tools (for a further explanation on the NLP tools, see Algorithm 3). The process ends just before the question generator comes into play (see question-generator in Figure 6.10).

(see *relations* in Algorithm 3); 4.) at the same time, it searches for possible compound-nouns and replaces each noun accordingly; 5.) next, for every pair of answers, it estimates a score value where it adds 1 if they are both members of the same number-class. It does the same, in case the two candidates share the same gender, participate in triples (*as subj and dobj*), and have a pronoun-gender agreement with the definite pronoun; 6.) it adds the score to a list of scores (see *answersScore* in Algorithm 3); 7.) The last step returns the best answer pair, which is the pair with the highest score.

Completing the Schema

The blended approach generates the questions and develops the two halves following the same procedure stated in the previous sections (see Section 6.3.3). Furthermore, similarly to the NLP approach and contrary to the deep learning approach, each developed schema is automatically classified into predefined categories and added to the schema-categorization DB (see Figure 6.13).

Algorithm 3 Blended pronoun-target pair selection

```
1: function FINDBESTANSWERPAIR(sentence)
2:     relations = returnPairs (["NOUN", "PROPN"], doubleRelations)
3:     compounds = findCompoundNouns ()
4:     pairs=match (compounds, relations )
5:     for pair in pairs do
6:         num = checkNumberAgreement (pair)
7:         gnd = checkGenderAgreement (pair)
8:         pga = checkPronounGenderAgreement (pair)
9:         trp = checkTriples(pair)
10:        score = m1+m2+num+gnd+pga+trp
11:        answersScore.append(score)
12:    end for
13:    return bestAnswerPair = pairs[answersScore.index(max(answersScore))]
14: end function
```

6.4 Experimental Evaluation

This section describes the results from several studies that we undertook to evaluate Winventor's performance in developing schemas based on the aforementioned approaches. Each of the following subsections reports on one of the approaches.

6.4.1 The NLP Approach

Here, we present the results obtained by applying the NLP approach. We describe the results from three studies that we undertook to evaluate the system's performance on replicating existing Winograd schemas from the DPR dataset (Rahman and Ng, 2012), on developing new Winograd schemas from scratch, and on helping humans develop new Winograd schemas.

Schema Replication

In this experiment, we have tested Winventor on replicating schemas from the DPR dataset (Rahman and Ng, 2012). Recall that this is a dataset of 943 schemas, where each half consists of a sentence, a definite pronoun (instead of the question), and two possible pronoun targets. The average sentence length of the database was 14 words. For this experiment, the *strict* mode was disabled, as this is a dataset developed under the "broad" flag. By giving Winventor the sentence of the first half of each schema, we wanted to evaluate if it could produce similar results as in the dataset. For each sentence, Winventor was requested to develop all possible schemas, storing at the same time all of the developed relations and factors (e.g., Mitkov-score, gender, number, and pronoun-gender-agreement variables).

Schemas: The results revealed that 416 sentences resulted in 990 halves, where 848 were schemas. More than two hundred schemas (254 halves of which 214 are schemas) were found to match the DPR dataset, meaning that they have the same definite pronoun and the same pronoun targets. At the same time, our system rejected 527 sentences for the following reasons: 1.) Nouns and proper-nouns have not been found (10 sentences); 2.) Target Pronoun relations have not been found (502 sentences); 3.) Questions have not been formed (13 sentences); 4.) Not an English sentence (2 sentences were wrongly identified). The large number of rejected sentences shows that further gains could be achieved via a more accurate semantic analysis of each sentence. For instance, over fifty percent of the sentences were rejected because of pure parsing—*Target Pronoun relations have not been found*.

Pronoun Targets: Regarding the pronoun targets, 122 halves were identified as proper-noun problems and 132 as noun problems. Among the proper-noun halves, it was found that 33% had more than two proper-nouns in each sentence. Similarly, 70% of the noun problems were found to have more than two nouns in each sentence. The positive difference in favor of the noun problems might suggest that resolving proper-nouns is more challenging than resolving nouns (Budukh, 2013).

Definite Pronoun: We further analyzed our results regarding the cases where Winventor correctly resolved the definite pronoun but not the correct pronoun targets. On average, we have found that each sentence identified as a proper-noun problem contains four proper-nouns, and each sentence identified as a noun problem contains five nouns. It seems that the increased number of possible pronoun targets might have led Winventor to the wrong conclusions. Further analysis has shown that the average sentence length for the examined sentences was increased. Specifically, schemas characterized as proper-noun problems contain, on average, thirteen words, and schemas characterized as noun problems contain nineteen words. At the same time, in the halves where Winventor correctly identified both the definite pronoun and the pronoun targets, the average length is twelve words for the proper-noun problems and fourteen words for the noun problems.

Question Development: Although the DPR dataset did not include questions, *Winventor* was able to produce schemas with valid questions (see Table 6.4). This result shows that the parsing of sentences through the question generator and, at the same time, the selection of the best appropriate question returned useful results.

	Sentence	Pronoun	Question
1	Tony helped Jeff because he wanted to help.	he	Who wanted to help?
2	The security team locked the scientists inside the building because they had to keep confidential information inside.	they	Who had to keep confidential information inside?
3	Sam helped Davey fortify their bunker because he thought the Mexicans were invading?	he	Who thought the Mexicans were invading?
4	Tiger Woods dropped Randy as his caddy because he was not satisfied with his work?	he	Who was not satisfied with his work?

Table 6.4 A snapshot of *Winventor's* developed questions on the DPR dataset.

Non-matching Schemas: Our results showed that Winventor developed 990 halves from 416 sentences, meaning that multiple schemas/halves were developed for each sentence. On the other hand, our analysis showed that only 254 halves (214 schemas) matched the DPR dataset, meaning that 74% of the halves were among those rejected as non-matching halves. Recall that there are sentences containing multiple nouns, proper nouns, and pronouns, which means that there is a big chance to lead to the development of more than one schema/half. For instance, in the DPR dataset, we have the following halves: i) *Sentence: Arnold Schwarzenegger cannot terminate John Conner, because he is protecting him. Definite-Pronoun: he, Answers: Arnold Schwarzenegger, John Conner*, and, ii) *Sentence: Arnold Schwarzenegger cannot terminate John Conner, because he is the leader of the resistance. Definite-Pronoun: he, Answers: Arnold Schwarzenegger, John Conner*. Although Winventor did not manage to build the requested schema, it returned the following results: i) *Sentence: Arnold Schwarzenegger cannot terminate John Conner, because he is protecting him. Definite-Pronoun: he, Question: Who is protecting him? Answers: Arnold Schwarzenegger, John Conner*, and ii) *Sentence: Arnold Schwarzenegger cannot terminate John Conner, because he is protecting him. Definite-Pronoun: him, Question: Who is he protecting? Answers: Arnold Schwarzenegger, John Conner*. As we can see, the question of the second half refers to a different pronoun than the DPR's half. Given that the DPR dataset was developed under the "broad" flag, Winventor's halves can be taken together to form a new *schema-like* example, albeit different from the original schema —it is called a *schema-like* example because having two pronoun resolution problems with the same sentence and different pronouns does not form a Winograd schema.

Selecting the best Halves: Given that for any sentence, multiple schemas might be created, many open questions remain regarding the fastest automated way to select the best ones (e.g., selecting the 254 halves from our database of 990 halves). To that end, we further analyzed the relation between the developed halves and different *factors* (e.g., Mitkov-score, triple, gender, and pronoun-gender agreement). The results showed a direct relation between our

factors and the selection of the best half. For instance, if we select all the halves that agree on gender, number, participate in triples, and have a pronoun-gender agreement, we have an 89% success rate. Furthermore, our results showed the importance of the triple factor (nsubj-dobj); it was shown that if we remove the triple factor, the success rate drops to 85%. Additionally, our analysis showed that if we select the halves according to their Mitkov-score, we have an 82% success rate, meaning that Mitkov's theory seems to work well when we have limited background knowledge.

Schema Development

Within this experiment, we investigated Winventor's appropriateness in developing new Winograd schemas from scratch. To that end, we analyzed schemas developed from Wikipedia sentences, with a survey that we designed and undertook. The schemas were developed with the *strict* flag enabled, meaning that they had to consist of a sentence, a question, and two possible pronoun-targets that agreed in gender, number, and had a pronoun-gender agreement. At the time of the experiment, *Winventor* had already searched 20000 sentences from the Wikipedia dataset to develop 500 schemas.

Experiment Design: For our experiment, we selected the Microworkers (MW) platform[5], which, as seen in the previous chapters, can be considered as one of the best available crowdsourced platforms. In this regard, we designed a questionnaire using LimeSurvey[6] and posted the link on the MW platform (see Figure 6.14). We divided our questionnaire into two sections. The first section consisted of twenty randomly selected Winograd halves, whereas the second consisted of ten Winograd *schemas*. Examples that were included in the first section were excluded from the second one. The questionnaire started with the first section and continued with the second one, where each half/schema was displayed on a single screen, followed by the question; in each example, three choices were displayed side-by-side: i) Valid Schema - Easy to Solve, ii) Valid Schema - Hard to Solve, iii) Non-Valid Schema. Furthermore, all participants were informed that they could not change a submitted answer once the survey started. Additionally, before taking the survey, each participant had to do the following: 1.) Read a consent form and agree to participate; 2.) Read an introduction guide about the WSC and pass a one-question training phase to get familiarized with the task; 3.) Select their age and their English language literacy level.

[5]www.microworkers.com
[6]https://cognition.ouc.ac.cy/surveys/

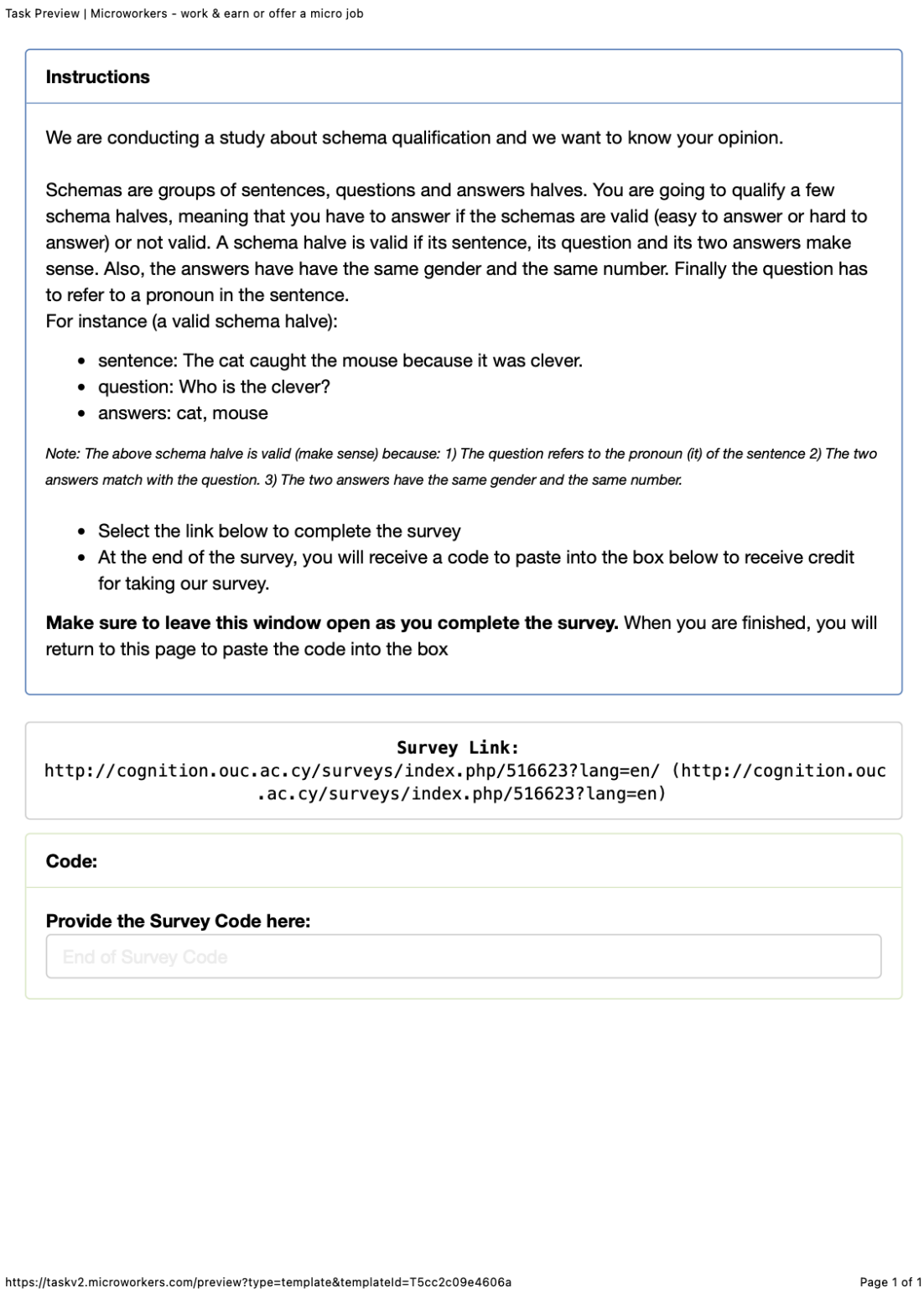


Figure 6.14 The ad we placed on the MicroWorkers platform to attract workers to validate Winventor's schemas/halves.

Participants: Our experiment was performed during May 2019, where a total of one hundred MW workers were recruited, aged between 18 and 65+ (none was discarded). Our participants were residents of English-speaking countries (United States, United Kingdom, Canada, New Zealand) and were screened through a qualification task from the Microworkers platform. In terms of their knowledge of the English language, 81% reported that it was "very good", 18% that it was "good". The total cost of our campaign was $250.

Results: In the first section, the participants characterized the halves as *valid* with a mean of 69% (σ = 0.15). In the second section, they characterized the schemas as *valid* with a mean of 73% (σ = 0.17). It seems that the positive difference in favor of the schemas might have happened not because of the quality of the schemas, which are harder to develop, but because of the following reasons: i) the participants were able to see the two halves at the same time, which seems to help them understand the meaning of the schema, and ii) sentences that were found appropriate for the development of schemas might have a simpler structure. Generally speaking, our results must be taken with a grain of salt as our study participants were not experts in the field. Specifically, we are not claiming that this system can develop schema/halves without the need for reviewing. For instance, in order to validate the next half, we need to change a word in the question and enhance the first answer (question: *is* to *causes*, answers: *stick* to *the use of the back of the stick*): *Sentence: If the back side of the stick is used, it is a penalty, and the other team will get the ball back. Question: What is a penalty? Answers: the stick, the ball*. Another example is the following schema which was considered valid by 81% of the participants: *1.) Sentence: Federer consistently beat Nadal since he was the better tennis player. Question: Who was the better tennis player? Answers: Federer, Nadal. 2.) Sentence: Federer consistently beat Nadal since he was the better tennis spectator. Question: Who was the better tennis spectator? Answers: Federer, Nadal*. Although it makes sense, after human reviewing, we could easily modify the second half as follows: *Sentence: Federer consistently beat Nadal since he was the worst tennis player. Question: Who was the worst tennis player? Answers: Federer, Nadal*.

Winventor as a Co-Worker

Within this experiment, we evaluated if Winventor can assist humans in the schema development process. To delineate it from the previous experiment, we asked ten colleagues who have prior experience developing schemas to design new halves from scratch in a specified period of time. For the sake of simplicity, participants were asked to develop only Winograd halves. To investigate whether this a priori appropriateness of Winventor as a co-worker can be justified, we divided the experiment into two sections. The experiment started with the

first section, where participants were asked to develop as many halves as possible without Winventor's help, in ten minutes; these were called non-guided halves. The participants continued with the second section, where the experiment was then replicated under conditions, in which we gave them access to fifteen randomly selected halves that Winventor developed —the results were called guided halves.

On average, we found that Winventor helped participants develop twenty halves, whereas, without Winventor's help, they only developed seven halves. Ostensibly, a sentence analysis that we undertook, showed that Winventor helped them develop halves based on different sentence patterns/types (see Table 6.5).

Using the Sentence-Structure Identifier we saw in previous chapters, we analyzed our results based on each half's sentence structure. The results yielded some interesting findings. Twenty-nine percent of the guided halves are based on compound sentences, 44% on complex sentences, 26% on compound-complex sentences, and 1% on simple sentences (see (a) of guided halves in Table 6.6). On the other hand, 33% of the non-guided halves are based on compound sentences, 63% on complex sentences, and 4% on compound-complex sentences (see (a) of non-guided halves in Table 6.6).

Without any help from Winventor, the participants mostly developed halves that follow the pattern "A DID X TO Y BECAUSE HE/SHE WAS Q", which is extremely overused (91% of the complex and 50% of the compound-complex sentences); these are the halves that follow the "Cause/Effect" relationship. Specifically, the halves that were designed with complex sentences had 91% "Cause/Effect", and 9% "Time" relationship (see (b) of non-guided halves in Table 6.6). The non-guided halves that were designed with compound-complex sentences had 50% "Cause/Effect" relationship and 50% "Time" relationship (see (c) of non-guided halves in Table 6.6). On the contrary, the guided halves that were designed with complex sentences had 9% "Cause/Effect", 11% "Comparison/Contrast", 2% "Place/Manner", 2% "Possibility/Condition", 36% "Relation", and 40% "Time" relationship (see (b) of guided halves in Table 6.6). The guided halves that were developed based on the compound-complex pattern showed the following results: 3% "Cause/Effect", 12% "Comparison/Contrast", 10% "Place/Manner", 13% "Possibility/Condition", 42% "Relation", and 20% "Time" relationship (see (c) of guided halves in Table 6.6). Regarding the compound sentences, 19% of the guided halves are arranged as "SV, and SV", 37% as "SV, but SV", 14% as "SV, or SV", 12% as "SV, so SV" and 18% as "SV; but, SV". At the same time, 5% of the non-guided halves are arranged as "SV, for SV", 37% as "SV, and SV" and 58% as "SV, but SV". The results provide convincing evidence that with Winventor's help, the participants were able to develop halves based on various sentence patterns/types; the complete opposite happened without Winventor's help (non-guided halves).

Our observations show that Winventor motivates and inspires participants to develop richer and more diverse schema/halves in the shortest time possible. The results are in line with WinoFlexi's results, where it was shown that schemas developed by crowdworkers have a similar hardness to those developed by experts.

6.4.2 The Deep-Learning Approach

In this section, we present the results of the deep learning approach. We begin by presenting the results regarding our models' training and then continuing by applying the methodology to develop schemas. For these experiments, we trained and evaluated our system on the original WSC dataset (Levesque et al., 2012). We divided our samples into a training and a testing set following the ratio of 70%-30% and evaluated our three models. Initials results showed an accuracy of 89% on the sentence selection process, 94% on the pronoun selection process, and 91% on the pronoun-target selection process.

Schema Replication

In this experiment, we have tested Winventor on replicating schemas from the DPR dataset (Rahman and Ng, 2012). Winventor loads all sentences from the first half of each schema and tests if it can produce the same or similar results as the second half of each schema. Here, in contrast to the NLP approach, Winventor develops one schema/half for each examined sentence (see Algorithm 2).

Sentence Model: The results revealed that the sentence model rejected only 170 sentences, achieving 82% accuracy, which is very near our initial training and testing results. Compared to our previous results (527 rejected sentences), it seems that the deep learning approach works better, meaning that it can correctly validate which sentences are appropriate for the development of schemas.

Definite Pronoun Model: In 96% of the cases (745 sentences), Winventor returned the correct pronoun. The results are in line with our training and testing results, meaning that our model can correctly identify the definite pronoun in sentences with multiple pronouns.

Pronoun Targets Model: Contrary to our expectations, Winventor returned the correct answers in only 9% of the cases (74 sentences). On the other hand, this is in line with the challenge difficulties and design purposes. Recall that the whole idea behind the WSC is to develop systems that can resolve the definite pronoun to one of its two coreferences, in each

	Automatically developed halves
1	Your governors are unjustifiably killing people and they only write the crime of the killed person to inform you.
	Who only write the crime of the killed person to inform you?
	The governors, The people
2	This river may have been shaped by God, or glaciers, or the remnants of the inland sea, or gravity or a combination of all, but the Army Corps of Engineers controls it now.
	What does the Army Corps of Engineers control now?
	The river, The island sea
3	Some do not eat grains, believing it is unnatural to do so, and some fruitarians feel that it is improper for humans to eat seeds as they contain future plants, or nuts and seeds, or any foods besides juicy fruits.
	What contain future plants?
	The grains, The nuts
4	The Greeks hiding inside the Trojan Horse were relieved that the Trojans had stopped Cassandra from destroying it, but they were surprised by how well she had known of their plan to defeat Troy.
	Who were surprised by how well she had known of their plan to defeat Troy?
	The Greeks, The Trojans
5	The reintroduction of a permanent diaconate has permitted the Church to allow married men to become deacons but they may not go on to become priests.
	Who may not go on to become priests?
	The men, The deacons

	Halves developed by humans with Winventor's help
1	Because of a misunderstanding Hitler had with Stalin, he attacked his country, misjudging the level of preparation needed to withstand harsh weather conditions, and subsequently that misunderstanding had cost him the war.
	Who the misunderstanding cost the war?
	Hitler, Stalin
2	Even though Meredith was the one who had committed the fraud, Andrea wanted to fix everything, so she confessed and went to jail.
	Who went to jail?
	Meredith, Andrea
3	Some fruitarians feel that it is improper for humans to eat seeds as they contain future plants.
	What contains future plants?
	Grains, Humans
4	This river may have been shaped by God, or glaciers, or the remnants of the inland sea, but the Army Corps of Engineers controls it now.
	What does the Army Corps of Engineers control now?
	The river, The island sea
5	While Oliver was having a party, Doug was in the city, saving some people, and trying to prove he was in fact the vigilante the police was looking for.
	Who was the vigilante?
	Oliver, Doug
6	Since everybody could always rely on Tommy, they expected him to have a plan, and so did John, but unfortunately he got shot during this specific operation by their worst enemy.
	Who got shot?
	John, Tommy

	Halves developed by humans without Winventor's help
1	Jack gave John the book, although he didn't need it.
	Who didn't need the book?
	Jack, John
2	My cat hates my dog because it is jealous.
	Who is jealous?
	My cat, My dog
3	Alice tried to reach her mother's head but she was too short.
	Who was too short?
	Alice, Her mother
4	Mary tried to calm her mother, but she was really stressed.
	Who was stressed?
	Mary, Her mother
5	Kids talk to their parents but sometimes they are too busy to listen.
	Who are busy?
	The kids, The parents
6	Jane gave Christina the necklace before she died.
	Who died?
	Jane, Christina

Table 6.5 A subset of the Winograd halves developed by humans with and without Winventor's help. The first five are a subset of the examples given to inspire humans in the development of quality Winograd halves.

	Guided halves	Non-Guided halves
a) Sentence Pattern		
simple sentences	1%	-
compound sentences	29%	33%
complex sentences	44%	63%
compound-complex	26%	4%
b) Complex Sentence Type		
cause/effect	9%	91%
comparison/contrast	11%	-
place/manner	2%	-
possibility/condition	2%	-
relation	36%	-
time	40%	9%
c) Compound-Complex Sentence Type		
cause/effect	3%	50%
comparison/contrast	12%	-
place/manner	10%	-
possibility/condition	13%	-
relation	42%	-
time	20%	50%

Table 6.6 Sentence patterns of halves that were developed based on guided-halves —designed with Winventor's help— and non-guided halves. In the first example (a) we see the developed number of simple, compound, complex, and compound-complex sentences, of the guided and non-guided halves. In the second (b) and third (c) example we see the number of complex and compound-complex sentences, regarding their sentence type.

half. In this regard, it seems that it might be a stretch to find the correct pronoun target in sentences with multiple candidates.

Schemas: Results showed that 745 sentences resulted in 162 schemas. This is in line with our Pronoun-Targets model results because the question generator automatically rejects questions that have as answers possible pronoun targets (e.g., the notification "A noun is in the question" was returned in 1698 cases). Given that our previous results showed that 416 sentences resulted in 254 schemas, it seems that the NLP approach can provide us with more schemas than the deep learning approach.

6.4.3 The Blended Approach

Below, we present the results by applying the methodology described in the blended approach section (see Section 6.3.5). Specifically, we performed an analysis regarding Winventor's ability to replicate and develop schemas from scratch. Additionally, we performed a speed analysis comparison between the two approaches.

Schema Replication

Within this experiment, we report results based on Winventor's ability to replicate schemas from the DPR dataset (Rahman and Ng, 2012). Like before, the results are expressed in terms of accuracy.

Results showed that in 50% of the cases (389), Winventor selected the correct answer pair, which is 40% more than the deep learning approach (see Table 6.7). Our analysis showed that Winventor was able to develop 332 halves that match the DPR dataset; 70% of them (234) were found to be schemas. In the case of halves, this means 27% more than the NLP and 158% more than the deep learning approach. Furthermore, in the case of schemas, this means 10% more than the NLP and 159% more than the deep learning approach.

We observed that if we remove any of the NLP factors, the performance is further reduced, showing the importance of every single factor in the schema development process. The results ultimately show that our blended approach replicates more schemas than both the other methods. On the other hand, our findings show that the development per sentence ratio of the NLP approach is better than the blended approach. According to our findings, 61% of the NLP approach's sentences were successfully used in the development of halves. In contrast, in the blended approach, only 43% of the sentences resulted in halves. This suggests that the NLP approach works better with the question generator mechanism. This may have occurred because the question generator needs to successfully output the semantic relations

of a given piece of text to develop the questions; It seems that sentences rejected by the NLP approach have a too complex structure to be used with the question generator (Heilman and Smith, 2009). The results might suggest that a better question generator could lead to the development of more schemas/halves.

We also performed a speed analysis experiment. Since the availability of more schemas directly relates to the ability to run a WSC-based CAPTCHA service (see Chapter 4), Winventor needs to develop schemas at a sufficiently fast pace. Our results showed that the blended approach could return results in 1.5 hours instead of 5 hours for the NLP approach, meaning that Winventor can develop, on average, three schemas per minute.

Schema Development

Within this experiment, we report the results of Winventor's blended approach to developing schemas from scratch. In this regard, we fed Winventor with the same Wikipedia dataset, like in Section 6.4.1, and compared the two approaches. Specifically, we randomly selected 2000 Wikipedia sentences that were previously used in the NLP approach.

In contrast to previous findings, within this subset of Wikipedia sentences, the NLP approach returned 23 halves, of which 16 were schemas. On the other hand, the blended approach returned 39 halves, of which 25 were schemas. At the same time, 1587 sentences were rejected by our sentence model (79%), whereas 1978 sentences were rejected by the NLP approach (99%). On average, the blended approach provided 52% more halves and 44% more schemas than the NLP approach. In general, regarding the number of the developed schemas, the performance was a little disappointing. The prime cause of this discrepancy seems to be the structure of the sentences found on the Web. This realization is in line with the previous section, where Winventor could replicate more schemas as humans designed the sentences used. Furthermore, not surprisingly, there were some discrepancies due to our sentence model limitations. Recall that in previous examples, all of the sentences were validated as humans manually designed them. On the other hand, as some Wikipedia sentences did not include pronouns, our deep learning sentence-model mistakenly identified them as valid sentences. This might lead to the conclusion that our data augmentation process was not sufficient, meaning that more valid sentences are required in order to do better training.

One of the most surprising results from our analysis is the number of the developed schemas compared to the time needed. According to our results, the blended approach parsed 2000 sentences in 1 hour, whereas the NLP approach required 12 hours; the results show that the blended approach is 91.67% faster than the NLP approach. In general, although performance was not perfect, we still believe that results highlighted the importance of

	rejected sentences	used sentences	matching answers	matching schemas	matching halves
NLP	527	416	254	212	254
DL	170	745	75	27	38
BL	170	745	389	234	332

Table 6.7 Results of the developed schemas/halves based on various approaches (NLP, deep learning, and blended approach) that match the DPR dataset (943 schemas). Regarding the initially-rejected sentences of the deep learning and blended approaches, there is an additional number of 28 sentences where our pronoun-model did not manage to correctly identify the definite pronoun.

blending machine learning and semantic analysis to achieve better results. Overall, the results ultimately show that we could enhance the schema development process via the interaction between the two approaches. This also shows the possibilities of combining the two approaches in future challenges, which is already in full swing with recent research in the AI field (Marcus, 2018).

6.5 Chapter Summary

In this chapter, we investigated the possibility of building systems that considerably help humans in the schema development task.

We started with WinoFlexi, an online system built explicitly for the development of Winograd schemas. Our approach is based on the crowd's cooperation to develop Winograd schemas from scratch, rewarding the best workers who successfully build them. Despite the acknowledged difficulty of the task when assigned to individuals, our empirical evaluation offers evidence that online crowdsourced platforms and systems like *WinoFlexi* might offer a viable alternative in developing schemas of high quality. As a hard problem the WSC is, we believe that WinoFlexi will help the research community develop schemas from scratch, which would be a further step toward addressing the challenge.

We continued with Winventor, a machine-driven approach for developing Winograd instances (schemas/halves). Given that schemas' development is troublesome even for humans, Winventor mainly comes into play as an assistant for the schema development process. Our experiments offer evidence that Winventor can develop schemas within two approaches: i) the pure NLP approach, which provides a limited number of schemas, albeit with multiple variations, and ii) the blended approach, which provides a bigger number of schemas, albeit one for every single sentence. In either case, the variability generally stems from which method is used. The evidence from this study suggests that humans could utilize systems like Winventor in the schema development process. In this regard, we want to point

out that Winventor's results must be taken with a grain of salt as it cannot develop quality Winograd instances without human reviewing.

Related-Publications

1. Isaak, N. and Michael, L. (2019). WinoFlexi: A Crowdsourcing Platform for the Development of Winograd Schemas. In Liu, J. and Bailey, J., editors, AI 2019: Advances in Artificial Intelligence, pages 289–302, Cham. Springer International Publishing.

2. Isaak., N. and Michael., L. (2020). Winventor: A Machine-driven Approach for the Development of Winograd Schemas. In Proceedings of the 12th International Conference on Agents and Artificial Intelligence - Volume 2: ICAART,, pages 26–35. INSTICC, SciTePress.

3. Isaak, N. and Michael, L. (2021a). Blending NLP and Machine Learning for the Development of Winograd Schemas. In Rocha, A. P., Steels, L., and van den Herik, J., editors, Agents and Artificial Intelligence, pages 188–214, Cham. Springer International Publishing.

7

Conclusions and Future Work

Here, we review the implications of our results along with potential directions for future research.

7.1 Thesis Summary

The work presented here is involved with the Winograd Schema Challenge (WSC), a novel litmus test for machine intelligence that focuses on human behavior. As a new challenge in the field, the WSC has been proposed as a conceptually and practically appealing alternative to other AI challenges. Passing the challenge requires resolving pronouns in certain sentences where the use of world knowledge and the ability to reason seem necessary. To that end, on each half, each developed system should demonstrate how humans can tackle it in order to get closer to the AI goal of endowing machines with commonsense reasoning abilities.

The contribution of this thesis is fourfold:

- First, concerning the WSC, and as Levesque (2014) suggested, we developed a system that tackles the WSC by emphasizing knowledge representation and reasoning without treating English text as a monolithic source of information. For this purpose, our developed system uses logical inferences to answer Winograd schemas.

- Second, given that there is still a lot of room for improvement and that there are no silver bullets regarding the endowment of machines with commonsense reasoning abilities, we utilized the WSC to build a novel form of CAPTCHA. We expect that WSC-based CAPTCHAs will promote the challenge to people of various academic disciplines (Marcus and Davis, 2019), so that they could work on the problem of actually solving it, and perhaps, in the process, help towards the building of machines able to reason with commonsense knowledge.

- Third, given that not all schemas can be tackled with the same ease and that future challenges should be organized according to how humans answer them (Bender, 2015), we have designed two systems that automatically differentiate between Winograd instances according to their perceived human hardness.

- Fourth, given that the development of schemas is a challenging and complicated process even for humans (Morgenstern et al., 2016), we have designed two systems that mainly come into play as human assistants for the schema development process.

On a second front, our developed methods and tools, which refer to different periods of AI history, might help bring together a new generation of AI researchers who appreciate both classical AI and machine learning (Marcus and Davis, 2019), which might contribute to a faster solution of the problem.

7.1.1 Tackling the WSC with Logical Inferences

In this work, we have argued that most scholars try to solve the WSC through pure statistical approaches without invoking commonsense knowledge and reasoning like humans do (see Chapter 2). Given that human language is more than statistics or word patterns (Adger, 2019), and that the WSC was proposed as the means to understand human behavior, we have developed a system that focuses on classical AI to tackle the challenge with promising results. Put simply, we have shown that our developed system, Wikisense, focuses on commonsense knowledge, which can be retrieved and learned via a supervised learning approach, called auto-didactic (Michael, 2010). In this regard, Wikisense shows how learning and reasoning through knowledge acquisition can fruitfully interact for pronoun resolution.

Wikisense utilizes the Websense engine, which can output logical inferences so that humans can relate and interact with it (Isaak, 2011; Michael, 2013). The engine can respond to user queries provided in natural language text, with inferences implied by the given queries according to the collective human knowledge. According to Levesque (2014), for every schema, we need to think of people's behavior as a natural phenomenon to be explained. In this regard, we have shown that Wikisense utilizes the Websense engine's *logical inference* mechanism in order to tackle Winograd schemas. At the same time, the generated knowledge file based on which the logical inferences are produced (see Table 3.1 in Chapter 3) can explain Wikisense's decisions, which is crucially important for developing AI systems (Wooldridge, 2020).

For the Wikisense implementation and to strengthen its knowledge and reasoning abilities, we focused on human knowledge and reasoning skills. We argued that when someone has to figure out the meaning of a sentence to resolve pronouns or answer questions, they mainly

focus on the sentence structure (Adger, 2019). In this regard, to train Wikisense to utilize each sentence's structure, we used two parsers in the field, spaCy, and the Stanford parser. After Wikisense gains access to the English Wikipedia, via the two parsers and the *Scene Constructor* component, it develops first-order semantic scenes —logical *formulae*, similar to Prolog rules, which are like McCarthy's vision for logic-based AI (Wooldridge, 2020).

It was shown that for any given Winograd half, a specified number of Wikipedia sentences is acquired and transformed into semantic scenes. The sentence acquisition relates to specific keywords produced through several methods, such as synonyms and antonyms that enhance the whole process. Next, the semantic scenes are given as input to the Learner component to building its knowledge. Afterward, Wikisense uses its inference mechanism to tackle the examined halves.

All in all, our study provided an insight into how learning and reasoning through knowledge acquisition can fruitfully interact towards pronoun resolution. Results showed that, although there is still room for improvement, the Wikisense approach works well with respect to the WSC.

7.1.2 Using the Winograd Schema Challenge as a CAPTCHA

Given the difficulties of the challenge and aiming to bring more AI research in the field, in Chapter 4 we examined the task of utilizing the WSC as a novel form of CAPTCHA. We have argued that state-of-the-art systems based on machine learning solve narrow WSC subsets, side-stepping, from the main objective that every single half should tell us about how humans behave. Given that we can always build more challenging Winograd schemas (Cozman and Munhoz, 2020) with an infinite number of English sentences (Adger, 2019), the WSC is an ongoing challenge that statistical solutions will not take up for years to come.

To lay a foundation for a WSC-based CAPTCHA, through a study we designed and undertook, we compared how human performance, usability, and time needed for solving a WSC-based CAPTCHA relates to how humans perform on other types of CAPTCHAs. To design the study, we used a representative sample of various types of CAPTCHAs, including ones based on text, images, and math, along with a WSC-based CAPTCHA service developed from scratch.

Our findings indicate that WSC-based CAPTCHAs are not only justified in terms of their acceptability by human users, but they are generally faster, easier to solve, and equally entertaining as the most typical existing CAPTCHA tasks. In this regard, WSC-based CAPTCHAs might encourage researchers of various disciplines to work on actually trying to solve the WSC and perhaps, in the process, help build machines able to reason with commonsense knowledge.

7.1.3 Metrics of Hardness to Differentiate Between Winograd Instances

Given that not all Winograd schemas are equally easy or hard for humans, in Chapter 5, we argued that predicting their hardness index is an interesting task. First, Bender (2015), who established a human baseline for the WSC, argued that future Winograd challenges should have humans evaluate the schemas upon designing, as not all schemas have the same perceived hardness for humans. On a second front, having access to a tool that could potentially parse schemas to output their perceived human hardness index could be used to make sure that a WSC-based CAPTCHA service displays harder schemas to solve in the case of possible fraudulent actions.

In this regard, two different approaches have been proposed to output the perceived human hardness indexes of Winograd instances.

The Wikisense-based Approach

In the first approach, we have shown how the performance of an automated approach correlates positively with human performance, suggesting that the performance of that particular approach could be used as a metric of hardness for WSC instances.

In support of the claim that the automated approach *varies* across WSC instances in a manner analogous to human performance, we presented evidence from two studies. The first refers to Bender's study, which involves adult native speakers, while the second study was designed as part of this work and involves non-native teenage speakers. Regarding the human performance data, we determined the human hardness index of a WSC instance as the percentage of people from a certain group that resolved it incorrectly.

For the automated approach, we utilized Wikisense to design the Wikisense-based approach. Using the data from the two aforementioned studies, we examined whether the performance of the *Wikisense* system could be predictive of the hardness of WSC instances for humans. As a baseline, we compared the system's predictive ability against other coreference resolution systems, the Stanford CoreNLP (Manning et al., 2014) and the Illinois-Coreference-Resolver (Bengtson and Roth, 2008; Peng et al., 2015). Results showed that the performance of the *Wikisense*-based approach *varies* across WSC instances in a manner that resembles the variability of the human performance more closely than what other systems achieve.

The WinoReg Approach

Given that machine learning approaches are excellent at correlation tasks (Marcus and Davis, 2019; Mitchell, 2019; Wooldridge, 2020), in our second approach, we designed a system

(WinoReg) that outputs the perceived hardness index of any Winograd half. This system was designed to address the limitations of the Wikisense-based approach in order to deliver faster, more accurate results. For this purpose, we developed WinoReg to work within two different modes, namely the random forest (WinoReg_RF) and the deep learning (WinoReg_DL) mode.

WinoReg_RF proceeds by first training a regression model based on the random forest algorithm and then using the learned model for faster computation during its deployment. Features provided as input to the system came from several works in the literature that we re-implemented as needed. To train WinoReg_RF, we used features from various components that relate to each examined WSC instance. In this regard, we used features that directly relate to semantic relations or the sentence pattern of each examined half. Next, using the data from Bender's work, we examined whether the performance of the WinoReg_RF system could be predictive of the WSC instance hardness for humans. Results showed that WinoReg_RF is more useful for achieving faster and better accuracy on the hardness of Winograd instances.

Regarding the WinoReg_DL approach, we showed how we could train our system using deep learning, an extremely valuable tool for correlation tasks. Given that deep learning killed feature engineering, which is time-consuming (Socher et al., 2012), we have shown that via deep learning we could achieve better and faster results than our previously used methods.

7.1.4 Designing new Winograd Instances from Scratch

The development of schemas is a challenging task, requiring inspiration, creativity, and motivation (Morgenstern et al., 2016). Moreover, tackling the challenge would likely require access to a sufficiently rich set of Winograd schemas, which are currently limited in their number and too cumbersome to create manually. In order to address these limitations, in Chapter 6, we proposed two different approaches, a crowdsourced-based and a machine-driven approach.

The WinoFlexi Approach

In the first approach, we presented WinoFlexi, a flexible online collaboration system that allows members of crowdsourcing platforms to collaborate on the development of Winograd schemas from scratch. WinoFlexi uses a combination of tools to enhance the schema-development process. We showed that crowdsourced workers could contribute to the development of schemas or evaluate their quality. In this regard, we showed that WinoFlexi utilizes various quality mechanisms to ensure the quality of the developed schemas.

To test the quality of the developed schemas, we used three well-known coreference resolution systems. The comparison of the results showed that the three systems' performance in the WinoFlexi-library is analogous to their performance in the original WSC_ library. Additionally, we compared the hardness indexes of the WinoFlexi dataset to that of the original WSC_ dataset, where our results showed the two sets to have comparable average hardness indexes. Taken altogether, we have shown that even though our workers were not initially familiar with the schema development process, through WinoFlexi's mechanisms, they were trained to design schemas similar to that of experts. Furthermore, according to comments received, certain workers found the development of schemas enjoyable, and were motivated at the same time by intrinsic incentives like amusement and curiosity for new knowledge.

The Winventor Approach

In the second approach, we developed Winventor, a machine-driven system that blends the advantages of deep learning and NLP. We have shown that Winventor can develop Winograd instances automatically (schemas or halves) and considerably help humans in the development task.

We started with the NLP approach, which uses various NLP tools to build Winograd instances from scratch. Given an English sentence, it searches for the definite pronoun and the pronoun targets based on the semantic analysis of its text. Given that, for any sentence, we can have various pairs of pronoun targets that meet certain criteria, it uses some predefined filters in order to rank each pronoun-target appropriateness. Then, through the Heilman and Smith (2009) question generator, it generates questions to build schemas/halves.

We continued with the deep learning approach, based on which three models were built. To that end, we trained a sentence, a definite pronoun, and a pronoun-target model, all of which are applied to each examined sentence to build Winograd schemas/halves. To increase the training dataset, we employed a data synthesis/augmentation procedure. As before, we used the Heilman and Smith (2009) question generator to generate questions to build schemas/halves.

Finally, we introduced the Blended approach, which combines NLP and deep learning more efficiently and effectively to develop schemas/halves. In other words, we replaced the pronoun-target model of the deep learning approach with the pronoun target selection process of the NLP approach.

To evaluate Winventor, we undertook experiments related to replicating and developing new Winograd schemas/halves from scratch and helping humans in the schema development process. The results showed that Winventor seems to motivate and inspire participants

to develop richer and more diverse halves in the shortest time possible. Comparing the three approaches results showed that the pronoun-targets model thwarts the deep learning approach's full potential, directly related to the challenge difficulties. Overall, the Blended approach provided better results than the NLP approach.

All in all, results showed that Winventor could be used within two different approaches: 1) the pure NLP approach, which provides a limited number of schemas/halves, albeit with multiple variations, 2) the blended approach, which provides a larger number of schemas/halves, albeit one for every single sentence. Of course, given that the schema development process is a troublesome and tedious task, we argued that Winventor does not purport to replicate the thought process of humans in the development of schemas but to help them in this challenging task considerably.

7.2 Future Work

The fact that many researchers from different parts of the world are interested in the WSC shows how significant the challenge is. However, the means behind the WSC make it challenging and troublesome, meaning that we are nowhere close to understanding and implementing the unfolding human mechanisms when tackling Winograd instances. We believe that work on different aspects of the WSC, like this thesis, will bring us closer towards the building of machines able to reason with commonsense knowledge. Below, we present future research directions regarding tackling, utilizing, and developing Winograd Instances from scratch.

7.2.1 Tackling the WSC with Logical Inferences

Although with Wikisense, we provided an insight into how learning and reasoning through knowledge acquisition could fruitfully interact for pronoun resolution, concerning the WSC, there is still room for improvement. Researchers could build on this work to enhance its capabilities. Future possible tasks for performance improvement may include implementing a better keyword-generator, which might lead to the acquisition of richer knowledge to benefit the system's commonsense reasoning ability. Furthermore, maybe Wikipedia's substitution with another knowledge resource might help the acquisition of richer knowledge. In this line of research, one could also combine multiple knowledge resources to build a knowledge aggregation mechanism that could offer better results.

Moreover, future research recommendations involve studying possible enhancements of Wikisense's commonsense reasoning abilities. Recall that for any given Winograd half,

Learner is rebuilding the acquired knowledge from the ground up. In this regard, learning through chaining might accelerate the Reasoners' prediction performance because the Learner's knowledge will be more comprehensive than without chaining. Furthermore, for any given half, Wikisense will use the previously acquired accumulated knowledge learned until that specific moment, which might help break down a more complex task into a sequence of simpler interconnected steps. For instance, the necessary knowledge for tackling a Winograd half might be found in the accumulated knowledge used to tackle previously seen halves.

Furthermore, given that humans do not reason with first-order logic but other frameworks, such as mental models or argumentation, it might be useful to apply those frameworks to bring Wikisense closer to Explainable AI (XAI), and make it more transparent. XAI refers to making something clear or justifying an action or a belief as a new way to transparency, hopefully as a way to heighten accountability (Edwards and Veale, 2017). In regards to XAI, it is believed there are four kinds of systems: opaque, interpretable, comprehensible, and truly-explainable systems (Doran et al., 2017). Opaque systems are the well-known black-boxes that do not explain their decisions. Interpretable systems are those where users can mathematically analyze the mechanism behind their decisions. Comprehensible systems through symbols enable evidence of how a decision is made or a conclusion reached. Finally, truly-explainable systems refer to the optimal goal, meaning systems that can explain their decision-making process using a human-like form of reasoning (Doran et al., 2017).

Based on the categories above, Wikisense is somewhere between an interpretable and a comprehensible system that needs to be moved nearer to the "right side of the spectrum". To do that, we might need to focus on argumentation, which is how humans justify or cancel a decision they previously made. Argumentation aims to give reasons to conclusions, as it can be seen as the process of providing possible explanations to given observations (Kakas and Michael, 2020). Therefore, before designing new systems, it is better to make sure that we know what human needs are. Of course, being able to explain is not a simple task, as humans think "out of the box", meaning that machines handle explicit knowledge better than implicit knowledge (Rutjes et al., 2019).

According to Kakas et al. (2016), "argumentation logic", a proposed framework that is closer to natural human reasoning than classical logic, can be used to combine human and automated reasoning of machines. Similarly, work from the literature has demonstrated that it might be possible to address this need through a first-order argumentation framework, which could be implemented as a decision support system to help humans analyze conflicting first-order information (Besnard and Hunter, 2005). In this regard, Wikisense's learner and reasoner could be easily modified with an open-ended dialectic process of argumentation,

attuned to human reasoning (Kakas et al., 2016). This framework could be used to give a different meaning to symbolic knowledge having at the same time the flexibility of human reasoning. Moreover, in the sense of XAI, this new framework could be used for the construction of acceptable arguments in an attempt to support Wikisense's conclusions on given WSC instances.

7.2.2 Using the Winograd Schema Challenge as a CAPTCHA

Regarding the WSC-based CAPTCHA, although designing good CAPTCHAs is a tedious task, we expect this work to be a good starting point for the future design of these kinds of CAPTCHAs. Therefore, we would encourage researchers to register and utilize our WSC-based CAPTCHA[1] service in their labs or personal web pages in order to prevent bots from performing fraudulent actions. We believe that this will encourage researchers of various disciplines to work on the WSC and perhaps, in the process, help build machines able to reason with commonsense knowledge.

One could consider specific extensions to strengthen the security level of WSC-based CAPTCHAs without sacrificing their ease of use. For example, to prevent automated bots from performing illicit and fraudulent actions, one could require the correct resolving of a specified number of halves in a row, with an additional banning and blocking of IP addresses that might repeatedly try random answers to pass these kinds of CAPTCHAs. Furthermore, like within the Google reCAPTCHA, one could use various techniques (e.g., mouse movement) to display the WSC-based CAPTCHA only when they suspect possible fraudulent actions. This will also increase the human acceptability rate, encouraging at the same time more researchers to work on the problem.

7.2.3 Metrics of Hardness to Differentiate Between Winograd Instances

Systems like WinoReg and the Wikisense-based approach can be used by researchers or challenge organizers to group schemas regarding their perceived human hardness indexes. Moreover, they could be integrated into CAPTCHA services to ensure that the generated schemas are not overly demanding for human users and in systems that pursue the development of Winograd schemas from scratch to ensure that various schemas would be developed. Finally, we suggest that future studies examine the impact of systems like WinoReg and the Wikisense-based approach in other AI fields. For example, in machine translation, they could be used to identify sentences that are harder to translate to acquire better feedback from people. In this regard, our developed systems can help with the problem many translation

[1] http://cognition.ouc.ac.cy/ws_builder

services face, regarding the focus of their attention to making end-users aware of the quality (Specia et al., 2009).

7.2.4 Designing new Winograd Instances from Scratch

Regarding the development of new Winograd instances from scratch, among possible directions for future research of interest would be the automation of parts of the schema development and validation process without taking humans out of the loop. According to the literature, human-machine cooperation (HCM) (Hoc, 2000) is necessary to introduce new stakes, which was done a long time ago with human-computer interaction. On the one hand, machines could provide autonomous agents for different tasks, and, on the other hand, humans control the activity with the optimum goal to produce better outcomes (Gil et al., 2019; Hoc, 2000). Furthermore, by amplifying human and machine intelligence, we can address some of their weaknesses (Vandenhof, 2019). In this regard, Winventor's schemas could be offered to the WinoFlexi for further processing and validation, leading to an interaction that would amplify human and machine intelligence by combining their complementary strengths. This human-machine teaming up might yield a more diverse set of schemas, expanding at the same time the creativity of crowdworkers. Moreover, inspired by having humans and machines act as associates, rather than supervisors and tools, we could bring the schema development process into a new era, where both humans and machines are evolving from supervisors and assistants to associates (de Visser et al., 2018).

7.3 Discussion

Given that, in this thesis, we have implemented a system that is based on commonsense reasoning, and at the same time, we have used machine learning to both develop and differentiate between Winograd instances, here, let us do our best to explain how we could potentially combine them into a new powerful system to take on the challenge.

On the one hand, we have commonsense reasoning that covers the traditional classical AI, and, on the other hand, we have machine learning (e.g., deep learning), the backbone of modern AI (Wooldridge, 2020). As we have previously seen, classical AI and machine learning have both positives and negatives. Each of these approaches has had success in specific areas. However, both have limitations in achieving AI's long-term goal of endowing machines with commonsense reasoning abilities, like those found in humans (Mitchell, 2019).

Take, for example, classical AI. The 1970s MYCIN expert system (Shortliffe and Buchanan, 1975), one of the most celebrated expert systems of the period, intended to be a doctor's assistant in providing advice about blood diseases (Mitchell, 2019; Wooldridge, 2020). MYCIN was able to come up with a diagnosis by combining logic and probabilistic reasoning, explaining at the same time its reasoning process, which is essential for applications of AI. To handle uncertainty, MYCIN used certainty factors (Wooldridge, 2020), which is something similar to what Wikisense's Learner uses. In the same regard, systems like DENDRAL (Feigenbaum et al., 1970), a chemical-analysis expert system, or XCON/RI (McDermott, 1980), a production-rule-based system for ordering computer components, attracted at the same time much interest in the AI community. A problem those systems faced relates to the fact that we cannot always use simple written rules to capture the knowledge of complex environments or handle environments that change over time (Wooldridge, 2020).

Furthermore, rule-based systems do not seem to handle incomplete knowledge very well. Rules written by humans heavily rely on subconscious knowledge, which programmed rules cannot easily capture. This is something that other classical AI systems are also faced with. For instance, systems like ConceptNet, where humans build knowledge bases consisting of written rules about the world, do not seem to work in various cases. For example, rules that are made up based on intuition, when applied to complex scenarios, can lead to absurd outcomes. According to Mitchell (2019), if we are not consciously aware of knowing something, how can we form a rule to *teach* it to a computer. In short, it is not an easy task to get experts to formulate what they know in a hyper-precise way that computers require, meaning that it is a mystery of how we could do this.

On the other hand, let us take deep learning. If we give it large amounts of training data and substantial computational resources, the results are great. There are numerous challenges that, although classical AI systems cannot handle (e.g., we do not know how to specify certain rules), can be easily tackled by deep learning techniques —e.g., the ImageNet competition (Deng et al., 2009). Of course, as we saw in previous Chapters, although these kinds of systems can produce great results, they do not comprehend the meaning of what we ask them (Mitchell, 2019). Their computational innards are so complex that no one completely understands how they work (Adger, 2019). At the same time, they also miss compositionality (Adger, 2019; Marcus and Davis, 2019), meaning they have no obvious ways of performing logical inferences.

According to Marcus and Davis (2019), we cannot get to the moon by climbing taller trees successively, as there are no silver bullets. In this regard, we must combine the two methods into building hybrid systems that will use the best of the two worlds in ways we have yet to discover —like the Blended approach in the previous Chapter, which provided the

best results. For example, parts of the brain might seem to do something like deep learning, but other areas seem to operate at a much higher level of abstraction (Marcus, 2018). For another, we know that humans blend multiple information sources when reasoning about future outcomes (Téglás et al., 2011). In this regard, building hybrid systems might get us closer to the long-term goal of AGI, though this is not going to be an easy task.

To that end, we might need to find ways to interpret machine learning decisions in order to supply them as rules into logic-based systems. As we have previously seen, tackling a Winograd half, without explaining why, is not the solution to the problem. On the other hand, drawing inferences in the form of written rules from any machine learning technique would arguably help us trust the model's prediction. The upshot of a complete commonsense reasoner would be remarkable for the AI community, albeit this gain may only be recognized once a significant part of the outcome has been developed (Davis and Marcus, 2015).

Finally, let us close this thesis by saying that the complete solution of the WSC will take time. As one of the AI pioneers, Marvin Minsky, said, "easy things for humans are hard for machines". The fact that we as humans unconsciously sense the abstract structure of sentences when we hear them gives us no easy solution to transfer that ability to machines. Years ago, previous works demonstrated, that in order to achieve pronoun resolution one had to be able to do everything else, and that once everything else is done, pronoun resolution will come *freely* and *automatically* (Hobbs, 1978). We hope that our work will positively contribute to this task. Through other future extensions and the cooperation with other researchers, maybe one day the solution of the WSC will come *freely* and *automatically*.

Bibliography

Ackerman, E. (2016). Winograd Schema Challenge Results: AI Common Sense Still a Problem, for Now. *Spectrum*.

Adger, D. (2019). *Language Unlimited: The Science Behind Our Most Creative Power*, volume 1. Oxford University Press.

Amsili, P. and Seminck, O. (2017). A Google-proof Collection of French Winograd Schemas. In *Proceedings of the 2nd Workshop on Coreference Resolution Beyond OntoNotes (CORBON 2017)*, pages 24–29, Valencia, Spain. Association for Computational Linguistics.

Anvesh Sinha, S. T. (2016). Review Paper on Different CAPTCHA Techniques. *IJCST Vol. 7*.

Bailey, D., Harrison, A., Lierler, Y., Lifschitz, V., and Michael, J. (2015). The Winograd Schema Challenge and Reasoning about Correlation. In *AAAI Spring Symposia*.

Baker, C. F., Fillmore, C. J., and Lowe, J. B. (1998). The Berkeley Framenet Project. In *Proceedings of the 17th international conference on Computational linguistics-Volume 1*, pages 86–90. Association for Computational Linguistics.

Baral, C. (2003). *Knowledge Representation, Reasoning and Declarative Problem Solving*. Cambridge university press.

Belk, M., Germanakos, P., Fidas, C., Holzinger, A., and Samaras, G. (2013). Towards the Personalization of CAPTCHA Mechanisms Based on Individual Differences in Cognitive Processing. In *Human Factors in Computing and Informatics*, pages 409–426. Springer.

Bender, D. (2015). Establishing a Human Baseline for the Winograd Schema Challenge. In *MAICS*, pages 39–45.

Bengio, Y., Ducharme, R., Vincent, P., and Jauvin, C. (2003). A Neural Probabilistic Language Model. *Journal of machine learning research*, 3(Feb):1137–1155.

Bengio, Y., Goodfellow, I., and Courville, A. (2017). *Deep Learning*, volume 1. MIT press.

Bengtson, E. and Roth, D. (2008). Understanding the Value of Features for Coreference Resolution. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 294–303. Association for Computational Linguistics.

Bernard, T. and Han, T. (2020). Mandarinograd: A Chinese Collection of Winograd Schemas. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 21–26, Marseille, France. European Language Resources Association.

Besnard, P. and Hunter, A. (2005). Practical First-order Argumentation. In *Proceedings of the National Conference on Artificial Intelligence*, volume 20, page 590. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999.

Bhagavatula, C., Bras, R. L., Malaviya, C., Sakaguchi, K., Holtzman, A., Rashkin, H., Downey, D., Yih, S. W.-t., and Choi, Y. (2019). Abductive Commonsense Reasoning. *arXiv preprint arXiv:1908.05739*.

Blanco, E. and Moldovan, D. (2011). Some Issues on Detecting Negation From Text. In *Twenty-Fourth International FLAIRS Conference*.

Bock, K., Patel, D., Hughey, G., and Levin, D. (2017). unCaptcha: A Low-Resource Defeat of reCaptcha's Audio Challenge. In *Proceedings of the 11th USENIX Conference on Offensive Technologies*, pages 7–7. USENIX Association.

Bos, J. and Markert, K. (2005). Recognising Textual Entailment with Logical Inference. In *HLT/EMNLP 2005, Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference, 6-8 October 2005, Vancouver, British Columbia, Canada*, pages 628–635. Association for Computational Linguistics.

Breiman, L. (2001). Random Forests. *Machine learning*, 45(1):5–32.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. (2020). Language Models are Few-Shot Learners.

Budukh, T. U. (2013). *An Intelligent Co-Reference Resolver for Winograd Schema Sentences Containing Resolved Semantic Entities*. Arizona State University.

Chambers, N. and Jurafsky, D. (2008). Unsupervised Learning of Narrative Event Chains. In *ACL*, volume 94305, pages 789–797.

Chollet, F. (2017). The Future of Deep Learning. *future*, 8:2.

Chomsky, N. (1959). A Review of BF Skinner's Verbal behavior. Reprinted in JA Fodor & JJ Katz. 1964. *The Structure of Language: Readings in the Philosophy of Language*, pages 547—578.

Chowdhury, G. G. (2003). Natural Language Processing. *Annual review of information science and technology*, 37(1):51–89.

Christoforaki, M. and Ipeirotis, P. (2014). Step: A Scalable Testing and Evaluation Platform. In *Proceedings of the 2nd AAAI Conference on Human Computation and Crowdsourcing*.

Cozman, F. and Munhoz, H. (2020). The Winograd Schemas from Hell. In *Anais do XVII Encontro Nacional de Inteligência Artificial e Computacional*, pages 531–542, Porto Alegre, RS, Brasil. SBC.

Dagan, I., Glickman, O., and Magnini, B. (2005). The Pascal Recognising Textual Entailment Challenge. In *Machine Learning Challenges Workshop*, pages 177–190. Springer.

Davis, E. (2016). Winograd Schemas and Machine Translation.

Davis, E. (2021). Personal Communication.

Davis, E. and Marcus, G. (2015). Commonsense Reasoning and Commonsense Knowledge in Artificial Intelligence. *Commun. ACM*, 58(9):92–103.

Davis, E., Morgenstern, L., and Ortiz, C. (2016). Human Tests of Materials for the Winograd Schema Challenge 2016.

Davis, E., Morgenstern, L., and Ortiz, C. L. (2017). The First Winograd Schema Challenge at Ijcai-16. *AI Magazine*, 38(3):97–98.

De Marneffe, M.-C., MacCartney, B., Manning, C. D., et al. (2006). Generating Typed Dependency Parses from Phrase Structure Parses. In *Proceedings of LREC*, volume 6, pages 449–454.

De Marneffe, M.-C. and Manning, C. D. (2008). The Stanford Typed Dependencies Representation. In *Proceedings of COLING 08 Workshop on Cross- Framework and Cross-Domain Parser Evaluation*, pages 1–8. Association for Computational Linguistics.

de Visser, E. J., Pak, R., and Shaw, T. H. (2018). From Automation to Autonomy: The Importance of Trust Repair in Human–Machine Interaction. *Ergonomics*, 61(10):1409–1427.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). ImageNet: A Large-Scale Hierarchical Image Database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL-HLT*.

Doran, D., Schulz, S., and Besold, T. R. (2017). What Does Explainable AI Really Mean? A New Conceptualization of Perspectives.

Edwards, L. and Veale, M. (2017). Slave to the Algorithm: Why a Right to an Explanation Is Probably Not the Remedy You Are Looking For. *Duke L. & Tech. Rev.*, 16:18.

Elmalech, A., Sarne, D., David, E., and Hajaj, C. (2016). Extending Workers' Attention Span Through Dummy Events. In *Fourth AAAI Conference on Human Computation and Crowdsourcing*.

Elson, J., Douceur, J. R., Howell, J., and Saul, J. (2007). Asirra: A CAPTCHA that Exploits Interest-Aligned Manual Image Categorization. In *ACM Conference on Computer and Communications Security*, volume 7, pages 366–374.

Emami, A., De La Cruz, N., Trischler, A., Suleman, K., and Cheung, J. C. K. (2018). A Knowledge Hunting Framework for Common Sense Reasoning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1949–1958, Brussels, Belgium. Association for Computational Linguistics.

Emami, A., Trichelair, P., Trischler, A., Suleman, K., Schulz, H., and Cheung, J. C. K. (2019). The KnowRef Coreference Corpus: Removing Gender and Number Cues for Difficult Pronominal Anaphora Resolution. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3952–3961, Florence, Italy. Association for Computational Linguistics.

Etzioni, O., Banko, M., and Cafarella, M. J. (2006). Machine Reading. In *AAAI*, volume 6, pages 1517–1519.

Fähndrich, J., Weber, S., and Kanthak, H. (2018). A Marker Passing Approach to Winograd Schemas. In *Joint International Semantic Technology Conference*, pages 165–181. Springer.

Feigenbaum, E. A., Buchanan, B. G., and Lederberg, J. (1970). On Generality and Problem Solving: A Case Study Using the DENDRAL Program.

Fidas, C. A., Voyiatzis, A. G., and Avouris, N. M. (2011). On the Necessity of User-Friendly CAPTCHA. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2623–2626. ACM.

François, C. (2017). *Deep Learning with Python*. Manning Publications Company.

Gadiraju, U., Möller, S., Nöllenburg, M., Saupe, D., Egger-Lampl, S., Archambault, D., and Fisher, B. (2017). Crowdsourcing Versus the Laboratory: Towards Human-Centered Experiments Using the Crowd. In *Evaluation in the Crowd. Crowdsourcing and human-centered experiments*, pages 6–26. Springer.

Gary Marcus (2019). Beyond Deep Learning with Gary Marcus. [online]. https://hbr.org/podcast/2019/10/beyond-deep-learning-with-gary-marcus.

Gatt, A. and Krahmer, E. (2018). Survey of the State of the Art in Natural Language Generation: Core tasks, applications and evaluation. *Journal of Artificial Intelligence Research*, 61:65–170.

Gelfond, M. and Lifschitz, V. (1988). The Stable Model Semantics for Logic Programming. In Kowalski, R., Bowen, and Kenneth, editors, *Proceedings of International Logic Programming Conference and Symposium*, pages 1070–1080. MIT Press.

Gil, Y., Honaker, J., Gupta, S., Ma, Y., D'Orazio, V., Garijo, D., Gadewar, S., Yang, Q., and Jahanshad, N. (2019). Towards Human-Guided Machine Learning. In *Proceedings of the 24th International Conference on Intelligent User Interfaces*, IUI '19, page 614–624, New York, NY, USA. Association for Computing Machinery.

Harwell, D. (2018). AI Models Beat Humans at Reading Comprehension, but They've Still Got a Ways to Go. *The Washington Post*.

Hasan, W. K. A. (2016). A Survey of Current Research on Captcha. *International Journal of Computer Science and Engineering Survey (IJCSES)*, 7(3):141–157.

Hassan, A. and Radev, D. (2010). Identifying Text Polarity Using Random Walks. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 395–403. Association for Computational Linguistics.

He, P., Liu, X., Chen, W., and Gao, J. (2019). A Hybrid Neural Network Model for Commonsense Reasoning. In *Proceedings of the First Workshop on Commonsense Inference in Natural Language Processing*, pages 13–21, Hong Kong, China. Association for Computational Linguistics.

Heilman, M. and Smith, N. A. (2009). Question Generation via Overgenerating Transformations and Ranking. Technical report, Carnegie-Mellon Univ Pittsburgh Pa Language Technologies Inst.

Hernandez-Castro, C. J. and Ribagorda, A. (2010). Pitfalls in CAPTCHA Design and Implementation: The Math CAPTCHA, a case study. *computers & security*, 29(1):141–157.

Hirth, M., Hoßfeld, T., and Tran-Gia, P. (2011). Anatomy of a Crowdsourcing Platform — Using the Example of microworkers.com. In *Proceedings of the 5th International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing*, pages 322–329. IEEE.

Hobbs, J. R. (1978). Resolving Pronoun References. *Lingua*, 44(4):311–338.

Hoc, J.-M. (2000). From Human – Machine Interaction to Human – Machine Cooperation. *Ergonomics*, 43(7):833–843. PMID: 10929820.

Hochreiter, S. and Schmidhuber, J. (1997). Long Short-Term Memory. *Neural computation*, 9(8):1735–1780.

Hong, S. J. and Bennett, B. (2020). Tackling Domain-Specific Winograd Schemas with Knowledge-Based Reasoning and Machine Learning.

Isaak, N. (2011). A First Attempt of the Creation Of a Commonsense Conclusion Web Engine (In Greek). Master's thesis, Open University of Cyprus.

Joachims, T. (2002). Optimizing Search Engines Using Clickthrough Data. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 133–142.

Joulin, A., Grave, E., Bojanowski, P., Douze, M., Jégou, H., and Mikolov, T. (2016). FastText.zip: Compressing Text Classification Models. *arXiv preprint arXiv:1612.03651*.

Kakas, A. and Michael, L. (2020). Abduction and Argumentation for Explainable Machine Learning: A Position Survey. *arXiv preprint arXiv:2010.12896*.

Kakas, A. C., Michael, L., and Toni, F. (2016). Argumentation: Reconciling Human and Automated Reasoning. In *Bridging@ IJCAI*.

Kimmig, A., Bach, S., Broecheler, M., Huang, B., and Getoor, L. (2012). A Short Introduction to Probabilistic Soft Logic. In *Proceedings of the NIPS Workshop on Probabilistic Programming: Foundations and Applications*, pages 1–4.

Klein, T. and Nabi, M. (2019). Attention Is (not) All You Need for Commonsense Reasoning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4831–4836.

Kocijan, V., Camburu, O.-M., Cretu, A.-M., Yordanov, Y., Blunsom, P., and Lukasiewicz, T. (2019a). WikiCREM: A Large Unsupervised Corpus for Coreference Resolution. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4294–4303.

Kocijan, V., Cretu, A.-M., Camburu, O.-M., Yordanov, Y., and Lukasiewicz, T. (2019b). A Surprisingly Robust Trick for the Winograd Schema Challenge. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4837–4842, Florence, Italy. Association for Computational Linguistics.

Kocijan, V., Lukasiewicz, T., Davis, E., Marcus, G., and Morgenstern, L. (2020). A Review of Winograd Schema Challenge Datasets and Approaches.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet Classification With Deep Convolutional Neural Networks. *Advances in neural information processing systems*, 25:1097–1105.

Le, T., Kim, J., and Kim, H. (2017). An Effective Intrusion Detection Classifier Using Long Short-Term Memory with Gradient Descent Optimization. In *2017 International Conference on Platform Technology and Service (PlatCon)*, pages 1–6.

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep Learning. *Nature*, 521(7553):436–444.

Lenat, D. (2008). The Voice of the Turtle: Whatever Happened to AI? *AI Mag.*, 29:11–19.

Levesque, H., Davis, E., and Morgenstern, L. (2012). The Winograd Schema Challenge. In *Thirteenth International Conference on the Principles of Knowledge Representation and Reasoning*.

Levesque, H. J. (2014). On Our Best Behaviour. *Artificial Intelligence*, 212:27–35.

Lin, S.-C., Yang, J.-H., Nogueira, R., Tsai, M.-F., Wang, C.-J., and Lin, J. (2020). Tttttackling Winogrande Schemas. *arXiv preprint arXiv:2003.08380*.

Liu, H. and Singh, P. (2004). ConceptNet — A Practical Commonsense Reasoning Tool-Kit. *BT technology journal*, 22(4):211–226.

Liu, Q., Jiang, H., Evdokimov, A., Ling, Z.-H., Zhu, X., Wei, S., and Hu, Y. (2016). Probabilistic Reasoning via Deep Learning: Neural Association Models. *arXiv preprint arXiv:1603.07704*.

Liu, Q., Jiang, H., Evdokimov, A., Ling, Z.-H., Zhu, X., Wei, S., and Hu, Y. (2017). Cause-Effect Knowledge Acquisition and Neural Association Model for Solving A Set of Winograd Schema Problems. In *IJCAI*, pages 2344–2350.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. (2019). RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv preprint arXiv:1907.11692*.

Loria, S. (2018). TextBlob Documentation. *Release 0.15*, 2.

Manning, C. D., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S. J., and McClosky, D. (2014). The Stanford CoreNLP Natural Language Processing Toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60.

Marcus, G. (2018). Deep Learning: A Critical Appraisal.

Marcus, G. and Davis, E. (2019). *Rebooting AI: Building Artificial Intelligence We Can Trust*. Vintage.

McCarthy, J., Minsky, M. L., Rochester, N., and Shannon, C. E. (2006). A Proposal for the Dartmouth Summer Research Project on Artificial Intelligence. *AI magazine*, 27(4):12–12.

McDermott, J. P. (1980). RI: an Expert in the Computer Systems Domain. In *AAAI*, volume 1, pages 269–271.

Melo, G., Imaizumi, V., and Cozman, F. (2019). Winograd Schemas in Portuguese. In *Anais do XVI Encontro Nacional de Inteligência Artificial e Computacional*, pages 787–798, Porto Alegre, RS, Brasil. SBC.

Michael, L. (2009). Reading Between the Lines. In *IJCAI*, pages 1525–1530.

Michael, L. (2010). Partial Observability and Learnability. *Artif. Intell.*, 174(11):639–669.

Michael, L. (2013). Machines with Websense. In *Proc. of 11th International Symposium on Logical Formalizations of Commonsense Reasoning (Commonsense 13)*.

Michael, L. and Valiant, L. G. (2008). A First Experimental Demonstration of Massive Knowledge Infusion. In *Principles of Knowledge Representation and Reasoning: Proceedings of the Eleventh International Conference, KR 2008, Sydney, Australia, September 16-19, 2008*, pages 378–389. AAAI Press.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G., and Dean, J. (2013). Distributed Representations of Words and Phrases and Their Compositionality. *arXiv preprint arXiv:1310.4546*.

Minsky, M. (2007). *The Emotion Machine: Commonsense Thinking, Artificial Intelligence, and the Future of the Human Mind*. Simon and Schuster.

Mitchell, M. (2019). *Artificial Intelligence: A Guide for Thinking Humans*. Penguin UK.

Mitchell, T. (2005). Reading the Web: A Breakthrough Goal for AI. *AI Magazine*, 26(3):12–16.

Mitkov, R. (1998). Robust Pronoun Resolution with Limited Knowledge . In *Proceedings of the 17th international conference on Computational linguistics-Volume 2*, pages 869–875. Association for Computational Linguistics.

Moreno, L., González, M., and Martínez, P. (2014). CAPTCHA and Accessibility-Is This the Best We Can Do?. In *WEBIST (2)*, pages 115–122.

Morgenstern, L. (2021). The Importance of WINOGRANDE: Technical Perspective. *Commun. ACM*, 64(9):98.

Morgenstern, L., Davis, E., and Ortiz, C. L. (2016). Planning, Executing, and Evaluating the Winograd Schema Challenge. *AI Magazine*, 37(1):50–54.

Morgenstern, L. and Ortiz, C. (2015). The Winograd Schema Challenge: Evaluating Progress in Commonsense Reasoning. In *Twenty-Seventh IAAI Conference*.

Mudge, R. (2010). The Design of a Proofreading Software Service. In *Proceedings of the NAACL HLT 2010 Workshop on Computational Linguistics and Writing: Writing Processes and Authoring Aids*, pages 24–32. Association for Computational Linguistics.

Nguyen, V. D., Chow, Y.-W., and Susilo, W. (2014). On the Security of Text-Based 3d Captchas. *Computers & Security*, 45:84–99.

Opitz, J. and Frank, A. (2018). Addressing the Winograd Schema Challenge as a Sequence Ranking Task. In *Proceedings of the First International Workshop on Language Cognition and Computational Models*, pages 41–52.

Parikh, A. P., Täckström, O., Das, D., and Uszkoreit, J. (2016). A Decomposable Attention Model for Natural Language Inference. *arXiv preprint arXiv:1606.01933*.

Peer, E., Samat, S., Brandimarte, L., and Acquisti, A. (2015). Beyond the Turk: An Empirical Comparison of Alternative Platforms for Crowdsourcing Online Research. *ACR North American Advances*.

Peng, H., Khashabi, D., and Roth, D. (2015). Solving Hard Coreference Problems. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 809–819.

Pennington, J., Socher, R., and Manning, C. D. (2014). Glove: Global Vectors for Word Representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Pinker, S. (2005). So How Does the Mind Work? *Mind & Language*, 20(1):1–24.

Prakash, A., Sharma, A., Mitra, A., and Baral, C. (2019). Combining Knowledge Hunting and Neural Language Models to Solve the Winograd Schema Challenge. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6110–6119.

Probst, P., Wright, M. N., and Boulesteix, A.-L. (2019). Hyperparameters and Tuning Strategies for Random Forest. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 9(3):e1301.

Radford, A., Narasimhan, K., Salimans, T., and Sutskever, I. (2018). Improving Language Understanding by Generative Pre-Training.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. (2019). Language Models Are Unsupervised Multitask Learners. *OpenAI blog*, 1(8):9.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. (2019). Exploring the Limits of Transfer Learning With a Unified Text-to-Text Transformer. *arXiv preprint arXiv:1910.10683*.

Rahman, A. and Ng, V. (2012). Resolving Complex Cases of Definite Pronouns: The Winograd Schema Challenge. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, EMNLP-CoNLL '12, pages 777–789, Stroudsburg, PA, USA. Association for Computational Linguistics.

Rovatsos, M., Gromann, D., and Bella, G. (2018). The Taboo Challenge Competition. *AI Magazine*, 39(1):84–87.

Ruan, Y.-P., Zhu, X., Ling, Z.-H., Shi, Z., Liu, Q., and Wei, S. (2019). Exploring Unsupervised Pretraining and Sentence Structure Modelling for Winograd Schema Challenge. *arXiv preprint arXiv:1904.09705*.

Ruder, S., Peters, M. E., Swayamdipta, S., and Wolf, T. (2019). Transfer Learning in Natural Language Processing. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Tutorials*, pages 15–18.

Rudinger, R., Naradowsky, J., Leonard, B., and Van Durme, B. (2018). Gender Bias in Coreference Resolution. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 8–14, New Orleans, Louisiana. Association for Computational Linguistics.

Rutjes, H., Willemsen, M., and IJsselsteijn, W. (2019). Considerations on Explainable AI and Users' Mental Models. In *CHI 2019 Workshop: Where is the Human? Bridging the Gap Between AI and HCI*. Association for Computing Machinery, Inc.

Sakaguchi, K., Bras, R. L., Bhagavatula, C., and Choi, Y. (2020). WinoGrande: An Adversarial Winograd Schema Challenge at Scale. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 8732–8740. AAAI Press.

Sap, M., Le Bras, R., Allaway, E., Bhagavatula, C., Lourie, N., Rashkin, H., Roof, B., Smith, N. A., and Choi, Y. (2019). ATOMIC: An Atlas of Machine Commonsense for If-Then Reasoning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3027–3035.

Schmidhuber, J. (2015). Deep Learning in Neural Networks: An Overview. *Neural networks*, 61:85–117.

Schubert, L. K. (2015). Semantic Representation. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*.

Schüller, P. (2014). Tackling Winograd Schemas by Formalizing Relevance Theory in Knowledge Graphs. In *Proceedings of the Fourteenth International Conference on Principles of Knowledge Representation and Reasoning*, pages 358–367.

Seals, B. T. (2017). unCAPTCHA Defeats Google CAPTCHA. [Online; accessed August-2018].

Sharma, A. (2019). Using Answer Set Programming for Commonsense Reasoning in the Winograd Schema Challenge.

Sharma, A., Vo, N. H., Aditya, S., and Baral, C. (2015). Towards Addressing the Winograd Schema Challenge - Building and Using a Semantic Parser and a Knowledge Hunting Module. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI*, pages 25–31.

Shortliffe, E. H. and Buchanan, B. G. (1975). A Model of Inexact Reasoning in Medicine. *Mathematical biosciences*, 23(3-4):351–379.

Singh, P., Lin, T., Mueller, E. T., Lim, G., Perkins, T., and Zhu, W. L. (2002). Open Mind Common Sense: Knowledge Acquisition From the General Public. In *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*, pages 1223–1237. Springer.

Sivakorn, S., Polakis, J., and Keromytis, A. D. (2016). I'm not a human: Breaking the Google reCAPTCHA. *Black Hat,(i)*, pages 1–12.

Socher, R., Bengio, Y., and Manning, C. D. (2012). Deep learning for NLP (without magic). In *Tutorial Abstracts of ACL 2012*, pages 5–5. Association for Computational Linguistics.

Specia, L., Turchi, M., Cancedda, N., Dymetman, M., and Cristianini, N. (2009). Estimating the Sentence-Level Quality of Machine Translation Systems. In *13th Conference of the European Association for Machine Translation*, pages 28–37.

Speer, R., Chin, J., and Havasi, C. (2017). CConceptNet 5.5: An Open Multilingual Graph of General Knowledge. In *Thirty-First AAAI Conference on Artificial Intelligence*.

Speer, R. and Havasi, C. (2012). Representing General Relational Knowledge in Conceptnet 5. In *LREC*, pages 3679–3686.

Stacey, J. (2011). Text Mining Wikipedia for Misspelled Words.

Sundermeyer, M., Ney, H., and Schlüter, R. (2015). From Feedforward to Recurrent Lstm Neural Networks for Language Modeling. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23(3):517–529.

Sundermeyer, M., Schlüter, R., and Ney, H. (2012). Lstm Neural Networks for Language Modeling. In *Thirteenth annual conference of the international speech communication association*.

Suresh, M., Taib, R., Zhao, Y., and Jin, W. (2019). Sharpening the BLADE: Missing Data Imputation Using Supervised Machine Learning. In Liu, J. and Bailey, J., editors, *AI 2019: Advances in Artificial Intelligence*, pages 215–227, Cham. Springer International Publishing.

Technoblog.org (2017). Google no Captcha + INVISIBLE reCaptcha – First Experience Results Review. [Online; accessed August-2018].

Themistocleous, C. (2009). Written Cypriot Greek in Online Chat: Usage and Attitudes. In *Proceedings of the 8th International Conference on Greek Linguistics*, volume 30, pages 473–488. University of Ioannina.

Trichelair, P., Emami, A., Cheung, J. C. K., Trischler, A., Suleman, K., and Diaz, F. (2018). On the Evaluation of Common-Sense Reasoning in Natural Language Understanding. *arXiv preprint arXiv:1811.01778*.

Trinh, T. H. and Le, Q. V. (2018). A Simple Method for Commonsense Reasoning. *arXiv preprint arXiv:1806.02847*.

Tung, B. L. (2017). Google Algorithm Busts CAPTCHA With 99.8 Percent Accuracy. [Online; accessed August-2018].

Téglás, E., Vul, E., Girotto, V., Gonzalez, M., Tenenbaum, J., and Bonatti, L. (2011). Pure Reasoning in 12-Month-Old Infants as Probabilistic Inference. *Science (New York, N.Y.)*, 332:1054–9.

Valiant, L. G. (2006). Knowledge Infusion. In *Proceedings of the 21st National Conference on Artificial Intelligence - Volume 2*, AAAI'06, pages 1546–1551. AAAI Press.

Vandenhof, C. (2019). A Hybrid Approach to Identifying Unknown Unknowns of Predictive Models. In *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*, volume 7, pages 180–187.

Verger, B. R. (2017). Google Just Made the Internet a Tiny Bit Less Annoying. [Online; accessed August-2018].

Von Ahn, L., Blum, M., Hopper, N. J., and Langford, J. (2003). CAPTCHA: Using Hard AI Problems For Security. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 294–311. Springer.

Wang, A., Pruksachatkun, Y., Nangia, N., Singh, A., Michael, J., Hill, F., Levy, O., and Bowman, S. R. (2019a). Superglue: A Stickier Benchmark for General-Purpose Language Understanding Systems. *arXiv preprint arXiv:1905.00537*.

Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., and Bowman, S. (2018). GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Wang, S., Zhang, S., Shen, Y., Liu, X., Liu, J., Gao, J., and Jiang, J. (2019b). Unsupervised Deep Structured Semantic Models for Commonsense Reasoning. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 882–891.

Webster, K., Recasens, M., Axelrod, V., and Baldridge, J. (2018). Mind the Gap: A Balanced Corpus of Gendered Ambiguous Pronouns. *Transactions of the Association for Computational Linguistics*, 6:605–617.

Wilson, T., Hoffmann, P., Somasundaran, S., Kessler, J., Wiebe, J., Choi, Y., Cardie, C., Riloff, E., and Patwardhan, S. (2005a). Opinionfinder: A System for Subjectivity Analysis. In *Proceedings of HLT/EMNLP 2005 Interactive Demonstrations*, pages 34–35.

Wilson, T., Wiebe, J., and Hoffmann, P. (2005b). Recognizing Contextual Polarity in Phrase-Level Sentiment Analysis. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*.

Winograd, T. (1972). Understanding Natural Language. *Cognitive Psychology*, 3(1):1–191.

Wolfe, J. H. (1976). Automatic Question Generation From Text-An Aid to Independent Study. In *Proceedings of the ACM SIGCSE-SIGCUE technical symposium on Computer science and education*, pages 104–112.

Wooldridge, M. (2020). *The Road to Conscious Machines: The Story of AI*. Penguin UK.

Yan, J. and El Ahmad, A. S. (2007). Breaking Visual CAPTCHAs with Naive Pattern Recognition Algorithms. In *Computer Security Applications Conference, 2007. ACSAC 2007. Twenty-Third Annual*, pages 279–291. IEEE.

Yan, J. and El Ahmad, A. S. (2008). Usability of CAPTCHAs or Usability Issues in CAPTCHA Design. In *Proceedings of the 4th symposium on Usable privacy and security*, pages 44–52. ACM.

Ye, Z.-X., Chen, Q., Wang, W., and Ling, Z.-H. (2019). Align, Mask and Select: A Simple Method for Incorporating Commonsense Knowledge into Language Representation Models. *arXiv preprint arXiv:1908.06725*.

Zhang, H. and Song, Y. (2018). A Distributed Solution for Winograd Schema Challenge. In *Proceedings of the 2018 10th International Conference on Machine Learning and Computing*, ICMLC 2018, page 322–326, New York, NY, USA. Association for Computing Machinery.

Zhao, J., Wang, T., Yatskar, M., Ordonez, V., and Chang, K.-W. (2018). Gender Bias in Coreference Resolution: Evaluation and Debiasing Methods. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 15–20, New Orleans, Louisiana. Association for Computational Linguistics.