EPiC Series in Computing

Volume 72, 2020, Pages 46–58

GCAI 2020. 6th Global Conference on Artificial Intelligence (GCAI 2020)

WinoReg: A New Faster and More Accurate Metric of Hardness for Winograd Schemas

Nicos Isaak[1] and Loizos Michael[2]

[1] Open University of Cyprus
nicos.isaak@st.ouc.ac.cy
[2] Open University of Cyprus &
Research Center on Interactive Media,
Smart Systems, and Emerging Technologies
loizos@ouc.ac.cy

Abstract

The Winograd Schema Challenge (WSC), the task of resolving pronouns in certain carefully-structured sentences, has received considerable interest in the past few years as an alternative to the Turing Test. In our recent work we demonstrated the plausibility of using commonsense knowledge, automatically acquired from *raw text* in English Wikipedia, towards computing a metric of hardness for a limited number of Winograd Schemas.

In this work we present *WinoReg*, a new system to compute hardness of Winograd Schemas, by training a Random Forest classifier over a rich set of features identified in relevant WSC works in the literature. Our empirical study shows that this new system is considerably faster and more accurate compared to the system proposed in our earlier work, making its use as part of other WSC-based systems feasible.

1 Introduction

Artificial Intelligence is concerned with the study of intelligent forms of behavior and with how systems that acquire and manipulate commonsense knowledge can be created [14, 21]. Towards that end, a number of challenges have been proposed that aim for the development of systems that will replace or substitute basic human abilities. One of these challenges is the Winograd Schema Challenge (WSC) [13], an alternative to the well-known Turing test [1, 14]. Rather than having to show intelligence by emulating a human during a free-form conversation, as suggested by the Turing Test, a machine should be able to show that it is thinking in a more constructive direction, away from the possibility of deception and trickery [13, 14].

One dimension in the work of tackling the WSC is the evaluation of the hardness of a particular Winograd Schema. In our previous work, we approached this question by reusing the *Wikisense* system [7] for resolving Winograd Schemas, and, roughly, using the amount of training data it requires to correctly answer a given Winograd Schema as an indicator of its hardness. This *Wikisense*-based hardness [8] was shown to correlate well with the performance of humans in solving the WSC across different Winograd Schemas. Notably, the system operates

without any pre-training, and thus, in particular, it does not require access to the performance of humans as an input. On the other hand, it does require knowledge of the correct answer to the Winograd Schema whose hardness it is evaluating.

We have used the *Wikisense*-based system as part of the feedback mechanism of a crowdsourcing platform that we have created for the development of Winograd Schemas of high quality [10]. Our experience with its use has revealed two key drawbacks: i) it requires large amounts of training data from the English Wikipedia, which leads to long delays, and ii) it is able to output the hardness only for a subset of Winograd Schemas. This work aims to develop an alternative way of computing the hardness metric that addresses these shortcomings.

Naturally, we have chosen to consider a complementary approach to the *Wikisense*-based system. *WinoReg*, our new proposed system, proceeds by first training a regression model using Random Forests, and then using the learned model for faster computation during its deployment. During the training phase, *WinoReg* requires access to the performance of humans on each of the Winograd Schemas in the training set. Unlike the *Wikisense*-based system, however, it does not require access to the correct answer to each of these Winograd Schemas. Features provided as input to the system come from a number of works in the literature that have developed WSC-related systems [4, 7, 16, 17, 18], which we have re-implemented as needed.

2 Motivation and Methodology

Winograd Schemas are pairs of Winograd Halves, each consisting of a sentence, a definite pronoun or a question, two possible pronoun targets, and the correct answer [13, 14], as illustrated next: *1.) Sentence: Erica called Jennifer on the phone because she was not responding to email. Question: Who was not responding to email? Answers: Jennifer, Erica. Correct Answer: Jennifer. 2.) Sentence: Erica called Jennifer on the phone because she was not able to email. Question: Who was not able to email? Answers: Jennifer, Erica. Correct Answer: Erica.*

Having access to only one Schema Half, the objective is to resolve the definite pronoun to one of its two co-referents. The avoid trivializing the task, the co-referents belong to the same gender, and both are either singular or plural. What makes the task challenging is that the two Schema Halves differ only with respect to a special word or phrase, whose change between the two Schema Halves also changes the correct answer. Winograd Schemas are presumed to be easy for humans and hard for machines, precisely because of the supposition that they require the use of commonsense knowledge to understand the dependence of the correct answer on the special word or phrase. Even so, not all Winograd Schemas are equally easy or hard for humans, and the task of being able to predict their hardness index is an interesting question.

In our first attempt towards answering this question [8] we started by considering the *Wikisense* system [7] for the WSC, and whose performance improves as its gets more training data that are chosen based on the particular Winograd Schema it is trying to answer. We have shown that the amount of training data needed to correctly answer a Winograd Schema correlates positively with the performance of humans in correctly answering that Winograd Schema (even though the system never gets access to the performance of humans), suggesting that the performance of the particular automated approach could be used as a metric of hardness for WSC instances. On the other hand, since *Wikisense* is not always able to correctly answer a Winograd Schema (which is to be expected by design of the WSC), our *Wikisense*-based system was able to offer a hardness index on only 57% of our tested Schema Halves. Furthermore, because of its model-free approach and its reliance on training during query-answering, the *Wikisense*-based system needs, on average, 8 hours to output the hardness index of each given Winograd Schema.

To the best of our knowledge, no other system exists that outputs the hardness index of Winograd Schemas. This seems disproportional to the demand of new developed Winograd Schemas in the literature. For instance, a recent study [9] showed that the WSC can form the basis of a new type of CAPTCHA, which might encourage more AI researchers to work on the problem of actually trying to tackle the WSC. WSC CAPTCHAs could use the hardness index of Winograd Schemas to ensure that the generated instances are not overly demanding for human users. Furthermore, WinoFlexi, a new collaborative system for the development of new Winograd Schemas [10], already leverages the *Wikisense*-based hardness tool to generate *seeds* that are used as a feedback to human contributors, for labeling schemas with a hardness score which indirectly shows if a Winograd Schema is considered easy to answer or not.

Towards further addressing the need for the evaluation of the hardness of Winograd Schemas, we introduce *WinoReg*, a system based on training a regression model with the use of Decision Trees. We use, in particular, the Random Forest algorithm [6], which involves constructing an ensemble of Decision Trees, each trained on random subsets of the data. Recent research [20] showed that the Random Forest algorithm consistently maintains high imputation performance over the benchmark linear regression across a range of performance metrics. In our case, when presented with a new Winograd Schema, every Decision Tree votes on the hardness of the input, and the average of their votes is the predicted hardness of the ensemble.

Since the aim is to estimate the hardness index of a Schema Half, which indirectly refers to the ability to determine the correct answer, *WinoReg* expects features related to the sentence, the question, and the two candidate antecedents. In contrast to the *Wikisense*-based system, *WinoReg* does not require access to the correct answer. Given a question that indirectly shows the target pronoun and two candidate antecedents, we aim to train a regressor that uses features derived from the two candidates, such that the correct antecedent is assigned a higher rank.

3 Feature Engineering

To train *WinoReg* we engineer 50 features from 12 components (see Fig. 1). Most of the features are based on systems previously built in an attempt to tackle the WSC [4, 8, 16, 17]. Given that the majority of the systems are not open-source [4, 16, 17], we had to develop these components from scratch. *WinoReg* utilizes the *spaCy* (`https://spacy.io`) dependency parser to turn raw text into semantic relations. These relations act, in turn, as the basic feed for the feature development. We use *spaCy*, like in previous works [7], to output various relations between the sentence, the question and the two pronoun targets and use them in our feature engineering (see *Semantic Extraction* in Fig. 1). For instance, consider the following Schema Half (referred to later as *catch* example): *Sentence: The cat caught the mouse because it was clever. Question: Who is clever? Answers: The cat, The mouse.* Via *spaCy* we built scenes like *catch(cat,mouse), was(it,clever)* which means that *a cat caught a mouse*, and *something/someone is clever.*

3.1 Negation

Previous work has shown that *negation* plays an important role in the WSC [7]. Following that line of research, through *spaCy* we analyze each Schema Half to estimate if the two candidates (pronoun targets) and the pronoun are governed by negation. This is done through the sentence and question triples of the Schema Half (like in the *catch* example). Here, we create two binary features (*STN* for the two candidates, *QTN* for the pronoun) that contain the value of 1 if negation exists, and the value of 0 if it does not.

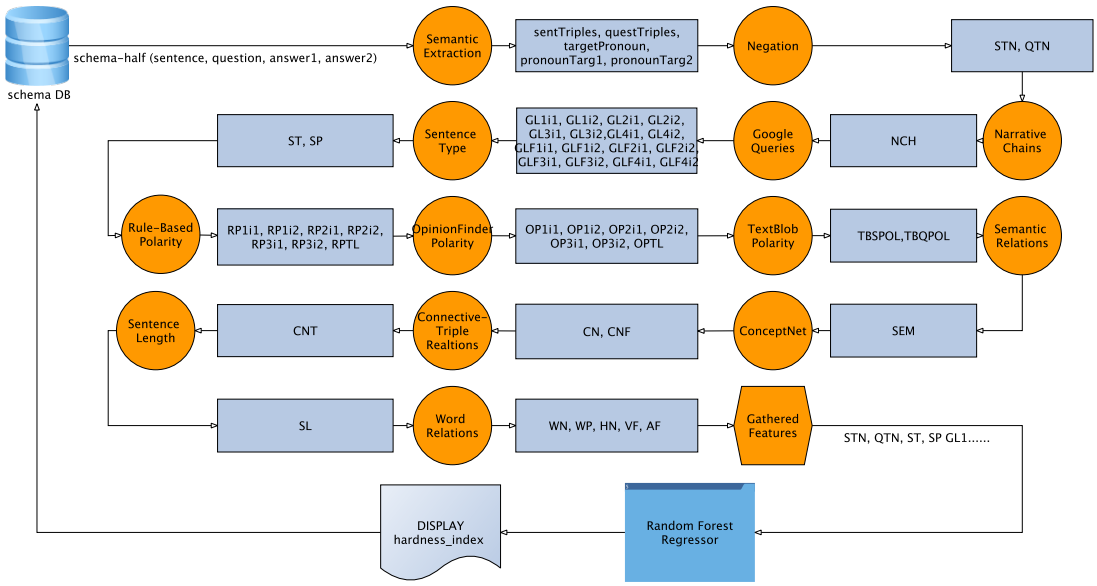

Figure 1: *WinoReg*'s architecture for a faster metric of hardness for Winograd Schemas. The various parts of the architecture are marked in orange eclipses, and are discussed in Section 3.

3.2 Narrative Chains

According to Budukh and Rahman et al. [4, 17], *Narrative chains* provide us with the story containing an event-based description of the participation of a common central actor called the protagonist. *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. We use spaCy, to output the subject and the object and after that we use Chambers and Jurafsky's (size 12) narrative chains [5]; these are ordered sets of 12 events (verbs) centered around a common protagonist.

Consider the following Schema Half: *Sentence: The city councilmen refused the demonstrators a permit because they advocated violence. Question: Who advocated violence? Answers: The city councilmen, The demonstrators.* Aiming to encode the kind of knowledge provided by Chambers and Jurafsky scripts we do the following: 1) via *Wikisense* mechanisms [7] we determine the event(s) that the two candidates participate in (e.g., refuse-?), and the event the target pronoun participates in with its role (advocate-s) 2) we pair each event participated by each candidate with each event participated by the pronoun [(refuse, advocate-s)] 3) from Chambers and Jurafsky's, for each such pair, we extract all the chains that contain both elements. For instance, in our example, Chambers & Jurafsky narrative chain contains *refuse-o and advocate-s*. In other words, the protagonist in this chain is the object of a refuse event and the subject of an advocate event (the demonstrators). In previous works the algorithm basically gives out no decision if there is no narrative chain matching between each event participated

by each candidate with each event participated by the pronoun [(refuse, advocate-s)]. In this work if *WinoReg* cannot find narrative chains containing both elements, it runs again the same procedure but with a similarity mechanism enabled ($sim >= -0.8$). Here we create a feature, NCH, and compute its value as follows. For our example, since *the demonstrators* are predicted to be the pronoun target, the value of NCH is set to 2 (second antecedent). Otherwise, it would be set to 1 (first antecedent). Finally, we note that NCH will be set to -1 if the pronoun and the two candidates do not participate in events, or no narrative chains can be extracted.

3.3 Google Queries

Google queries have been used to acquire a world knowledge, to learn patterns of word usage to tackle to WSC [16, 17, 18]. Consider the *catch* example: Humans resolve *it* to cat by exploiting the world knowledge that *clever catch/grab* easily other things. To get this kind of knowledge, we acquire patterns of word usage from the WWW through queries [17, 18]. Peng et al. [16], for each variation of nouns (plural and singular) and verbs (different tenses), create different queries and average the counts over all queries to create feature vectors. On the other hand, Rahman et al. [17] create triples that are based on the clauses preceding and following a discourse connective Conn, respectively in each sentence.

In our case, we follow a similar approach but without the need of a discourse connective in each sentence. Let a Schema Half be denoted by the following: A1 and A2 are the two pronoun targets (candidates), VQ be the verb that governs the pronoun (in the question), W be the sequence of words following VQ in the question, J is the adjective following a verb-to-be in the question. From these values, we generate six queries: QR1: A1VQ, QR2: A2VQ, QR3: A1VQW, QR4: A2VQW, QR5: JA1, QR6: JA2. For instance, in our *catch* example six queries are generated: (QR1) "cat was"; (QR2) "mouse was"; (QR3) "cat was clever"; (QR4) "mouse was clever"; (QR5) "clever cat"; and (QR6) "clever mouse". More specifically, using the counts returned by Google, we create eight binary features (GL1i1, GL1i2, GL2i1, GL2i2, GL3i1, GL3i2, GL4i1, GL4i2) whose values are determined by heuristic rules, respectively [17]. For the first two binary features (GL1i1, GL1i2), if the difference between the two queries is bigger than the threshold of 20% in favor of the first candidate (QR1 > QR2) then GL1i1 is set to 1 and GL1i2 to 0; otherwise, if the opposite exists then GL1i1 is set to 0 and GL1i2 to 1. According to Rahman et al. [17] the threshold ensures that a heuristic decision is made only if the difference between the two queries is significant. The other features can be estimated in the same way. More details about the process can be found in the paper that introduced the method [17].

We consider Framenet [2] for Google issues with proper names. It is unlikely that Google will return meaningful counts for persons [4, 17]. Consider the following Schema Half: *Sentence: Paul called George, but he wasn't successful. Question: Who wasn't successful?, Answers: Paul, George.* To generate meaningful queries, we can replace these proper names with their roles. Via *Wikisense* we generate the sentence triple the two candidates participate in (e.g., call(Paul, George)). Next we search Framenet for NP.EXT - NP.OBJ relations, where NP.EXT shows the subjects and NP.OBJ the objects for the FrameNet frame corresponding to the event (call). Finally we replace the name with its FrameNet role (NP.EXT=Speaker, NP.OBJ=Addressee). Consequently, we replace the two names with their extracted semantic roles, and generate the search queries from the resulting sentence in the same way as before; the features that are generated are GLF1i1, GLF1i2, GLF2i1, GLF2i2, GLF3i1, GLF3i2, GLF4i1, GLF4i2.

3.4 Sentence Type

Recent work has shown that the structure of sentences plays an important role in the quality of each Schema Half [10]. For this experiment, we use a tool that identifies the sentence type of each designed schema [10]. Given as input an English sentence, it outputs its type which can be either a simple, a compound, a complex, or a compound-complex sentence (stored in feature ST). Additionally it outputs it's pattern/clause (e.g., "SV because SV", "SV and SV because SV.", "Cause/Effect"), which is stored in feature SP.

3.5 Rule-Based Polarity

There are Schema Halves where we can resolve the target pronoun by comparing the two pronoun targets according to their polarity [4, 16, 17]. The basic idea is the following: 1) Take the question and the verb that governs the two pronoun targets 2) Find the polarity of the pronoun through the question, and the polarity of each pronoun target, and 3) assign the answer to the pronoun target that has the same polarity as the pronoun.

Consider the following Schema Half: *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 Schema Half: councilmen = subject of the event refuse, demonstrators = object of the event refuse, they = subject of the event advocate. We find the polarity of the *refuse* event from the Wilson et al. subjectivity lexicon [23]. The polarity given in subjectivity lexicon is negative. Hence the polarity of the object *demonstrators* becomes positive and the polarity of the deep subject *councilmen* becomes negative. The target pronoun *they* participates in the event *advocate*. The polarity of the event *advocate* in the subjectivity lexicon is positive. Since target pronoun is the subject of the event *advocate*, its polarity becomes positive. As we see from the above that the polarity of both the pronoun and the *demonstrators* is the same. Hence we resolve target pronoun *they* to *demonstrators*.

We create six features, RP1i1, RP1i2, RP2i1, RP2i2, RP3i1 and RP3i2. If the rank value of the pronoun or the rank value of one or both of the candidate antecedents cannot be determined, the values of all six binary features will be set to zero. For instance, RP1i1 and RP1i2 have to do with the two pronoun targets, respectively. To compute RP1i1 and RP1i1, which are binary features, we follow a rule-based procedure that has to do with the pronoun target (candidate) that has the same polarity with the pronoun. For example, since the second candidate (demonstrators) is the correct answer and not the first, RP1i1=0 and RP1i2=1. The value of RP2i1 and RP2i2 are the concatenation of the polarity values determined for the pronoun and the two candidates. For instance, RP2i1=negative-positive, and RP2i2=positive-positive. To compute RP3i1 and RP3i2, we simply take the previous features of RP2i1 and RP1i2 and append, if exists, the polarity reversing connective such as although; according to the literature [7, 17] these are connectives that can flip the polarity. To determine if a Schema Half has polarity reversing connectives we use a component from *Wikisense* [7]. If a polarity reversing connective exists we simply take RP2i1 and RP2I2 and append the connective to it (RP3i1 = RP2i1 + connective, RP3i2 = RP2i2 + connective). Additionally, we create another feature RPTL that shows the best pronoun candidate, as follows: First, we estimate the score of the first candidate by adding its binary features (Target1score = RP1i1). Next we do the same for the other candidate (Target2score = RP1i2). If Target1score > Target2score then the value of RPTL is 1. Otherwise, if Target2score > Target1score, the value of RPTL is set to 2. If Target1score==Target2score then the value of RPTL is set to -1.

3.6 OpinionFinder Polarity

It seems that the polarity values that were computed with the rule-based polarity could also be calculated with a sentiment analyzer [16, 17]. For this reason we use OpinionFinder [22], which is a system able to perform subjectivity analysis, like annotating phrases with their contextual polarity values. We compute the OpinionFinder polarity features in the same way we did with the rule-based polarity features, without the need to heuristically search for the polarity values, and create seven features (OP1i1, OP1i2, OP2i1, OP2i2, OP3i1, OP3i2, OPTL).

3.7 TextBlob Polarity

To test the polarity results of Wilson et al. subjectivity lexicon [23] and OpinionFinder [22] we use another, simpler, polarity mechanism. Though sentiment analysis we return the polarity of the verb that governs the two candidates, and the polarity of the verb that governs the pronoun. To do that we use *TextBlob* (`https://textblob.readthedocs.io/en/dev/`), which is a python library for processing textual data. We create two features (TBSPOL, TBQPOL) that each can contain one of the following values: neutral, positive, negative.

3.8 Semantic Relations

To fix possible query based problems of Google we compute another feature that is based on semantic relations. It seems that using web search queries has potential precision and recall problems. According to Rahman et al. [17]: i) the fact that a pronoun target and a verb appear close to each other in a search query does not mean that a subject-verb relation exists between them (precision problem) ii) these search queries fail to obtain subject-verb relations where a pronoun target and verb are not close to each other (recall problem). To address this, we use the Wikipedia corpus [7] to see how many times each pronoun target appears as subject or as an object. If the pronoun is governed by a "to be" verb then we cannot determine these values [17]. If the pronoun appears as a subject we search to find which candidate appears as a subject most of the times; for instance, if the first candidate appears more times as a subject then the feature SEM=1. Otherwise, if the pronoun appears as an object we search to find which antecedent appears most of the times as an object. If we cannot determine the values we set SEM to -1.

3.9 ConceptNet

ConceptNet is a semantic network of concepts, where nodes are concepts and the edges are the relations between them. It is a freely available semantic network that describes general human knowledge and how it is expressed in natural language [19]. ConceptNet provides a large set of background knowledge about different facts connected with other facts using relations such as *relatedTo*, *AtLocation*, *PartOf*, *IsA*, etc. According to Budukh [4] the strength of ConceptNet is that it contains concepts about everyday basic knowledge, cultural knowledge, scientific knowledge etc.

WinoReg uses conceptNet API to find the relatedness between the two pronoun targets and the verb/adjective that governs the pronoun. To resolve the problem of proper nouns we are using frameNet in the same way we used it with Google queries. For every antecedent we search for its *relatedness* with the verb/adjective that governs the pronoun. This is a ConceptNet function that returns a boolean value (value ≤ 1). At the end we create a feature (CN) that equals 1 if the value of the first candidate is greater than the value of the second candidate. If

the value of the second candidate is greater than the value of the first candidate then the value of CN is set to 2, and otherwise it set to -1. Additionally, we consider Framenet [2] for issues with proper names like we did with Google Queries, and create the CNF feature. Its values are being computed in the same way as the CN values.

3.10 Connective-Triple Relations

Humans can exploit world knowledge via causal relations, signaled by discourse connectives, between events [17]. Consider the following Schema Half: *Sentence: Bob paid for Charlie's college education because he is very generous, Question: Who is generous?, Answers: Bob, Charlie.* For instance, we might easily resolve the pronoun *he* by exploiting the world knowledge that there is a causal relation (by the discourse connective because) between the pay event and the generous event.

From every Schema Half, we collect the best triple and search for their frequencies of occurrences in the *Wikisense* corpus [7]. Each triple is of the form (V, Cn, X), where Cn is a discourse connective, V is a verb in the clause that governs the two candidate antecedents, and X is a stemmed verb or an adjective that governs the pronoun. Basically, each triple indicates a relation between V and X through the discourse connective. In the same way we find triples from every sentence in the *Wikisense* corpus and search for the frequencies of the best triple (also via similarity). According to Rahman et al. [17]: i) if the the number of occurrences is at least 100 then we proceed to the next step ii) if X is a verb, then it resolves the pronoun to the candidate that has the same role as the pronoun, otherwise, if the sentence does not involve comparison and X is an adjective, it resolves the pronoun to the candidate that serves as the subject of V. We create a binary feature, CNT, that encodes this heuristic decision (CNT=1 if the pronoun is resolved to the first antecedent, otherwise CNT=2 if the pronoun is resolved to the second antecedent). If we cannot resolve the pronoun then CNT=-1.

3.11 Sentence Length

Recent work has shown that the sentence length of each schema plays an important role in the the resolution of the target pronoun [11]. Given that the schemas that are built on sentences that have a big number of words are harder to resolve, we create a feature that holds the number of words of each Schema Half sentence (SL).

3.12 Word Features

Word features are features that can be divided into two categories: i) the antecedent-independent features, and ii) the antecedent-dependent features. According to previous works [17], these are very important features for the tackle of the challenge. At first, we must assume that each Schema Half sentence contains a connective (Cn). For the antecedent-independent features we create two features (WN, WP). The WN feature, contains the number of words in the sentence (except the candidate antecedents and the Cn). For the WP feature, we pair each word appearing before Cn with each word appearing after Cn, excluding adjective-noun pairs, noun-adjective pairs, and the two candidate antecedents. In the end, the WP feature contains the number of the pairs.

Next, we compute the antecedent-dependent features (HN, VF, AF). HN contains the number of the head words of the candidate antecedents that were returned by the dependency parser (spaCy); if we cannot determine the candidate antecedents in a sentence then the HN feature is set to 0 (the two candidates might be written/mentioned differently). Subsequently, the VF

feature contains the number of the verbs and JF the number of the adjectives modifying the two candidate antecedents.

4 Empirical Evaluation

Dataset: We report results on the test set, which comprises around 30% of the Winograd dataset which was developed by experts in the field [15]. This is the same set (100 Schema Halves) on which the *Wikisense*-based hardness experiment has been tested on [8]. At the time of writing, the Winograd dataset comprises 144 schemas (288 Schema Halves).

Evaluation Metrics: Results are expressed in terms of accuracy, and correlation coefficient.

Human Performance on the WSC: Like we did in our first work [8], here we present evidence from one study in support of the claim that the performance of the *WinoReg* system *varies* across WSC instances in a manner analogous to the performance of adult native speakers. In terms of the performance of humans on the WSC, the literature [3] establishes a baseline with adult speakers — residents of the United States — who speak English fluently. Bender shows that native English speakers are, on average, able to correctly resolve 92.1% of the WSC instances; 91%, if we consider only the first 100 WSC instances, which is the subset on which our previous system was tested on. A detailed analysis of human performance on each individual WSC instance (accuracy) is available from: https://github.com/benderdave/wsc-exp.git.

Using the data from the aforementioned study, we examine in this section whether the performance of the *WinoReg* system can be predictive of the hardness of the WSC instances for humans.

4.1 Results and Discussion

Coreference System	MAE	Correl	Accuracy	Schema Halves
Fixed Baseline	9.13	-1	90.87	100
Wikisense-based	23	0.22	77	100
WinoReg	8.36	0.33	91.64	100

Table 1: Results of the Fixed Baseline, the *Wikisense*-based hardness, and *WinoReg*

The fixed baseline: Our first baseline is a resolver that chooses the human adult bar as the schema hardness index for each Schema Half. Given that the human adult bar is set to 91%, if we train our Random Forest Regressor with only one feature and test it on the first 100 Schema Halves then our results would be as follows: Mean Absolute Error: 9.13 degrees and Correlation Coefficient (with the adults results): -1 (see Table 1).

***Wikisense*-based Hardness:** A summarizing of the results of the *Wikisense*-based Hardness system [8], which is trained using different sets of training data, is shown in Table 1. The *Wikisense*-based system was able to return results for only 57% of the tested Winograd Schemas with a correlation coefficient of 38%. If we take as granted the human adult bar on the WSC, which is 91% for the unresolved schemas, then the system achieves an accuracy of 77%, with a correlation coefficient of 22%.

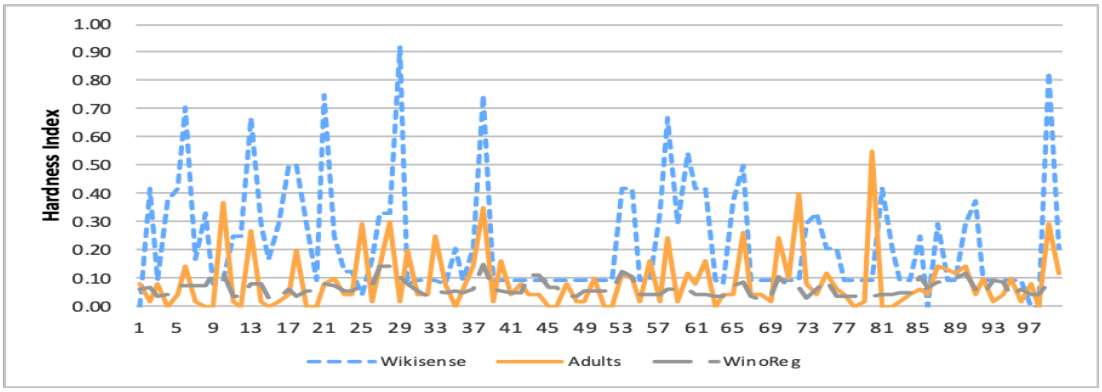

Figure 2: Variability of our developed *WinoReg* hardness index and *Wikisense*-based hardness index across the 100 WSC instances on which it was computed, in relation to the variability of the human hardness index for adults.

***WinoReg*:** Results of *WinoReg*, which is trained using the features described in Section 3, are shown in Table 1. *WinoReg* achieves an accuracy of 91.64% which is the same with the human adult accuracy, significantly outperforming the *Wikisense*-based system by 14.64% in accuracy and by 11% in correlation coefficient. Furthermore, if we use only the 57 Schema Halves the *Wikisense*-based system was able to resolve, the correlation coefficient of *WinoReg* and adults rises to 47% (which is by 9 percentage points bigger than what the *Wikisense*-based system was able to achieve (38%)). Our analysis ultimately shows that the performance of the *WinoReg* approach *varies* across WSC instances in a manner that resembles the variability of the human performance more closely than what other systems can achieve. This can be seen in Fig. 2 which depicts in more detail how the computed hardness index and the human hardness index vary across WSC instances, suggesting that indeed, certain WSC instances that are easier or harder for humans are accordingly labeled as such by the computed hardness index of *WinoReg*.

4.2 Speed Analysis

Recall that there are crowdsourcing systems that leverage the *Wikisense*-based hardness tool to generate *seeds* that are used as a feedback to help contributors develop harder/easier schemas (depending on the hardness index) [10]. Given that the hardness index of each developed schema seems to play an important role on the quality of the developed schemas, and directly relates to the amounts that is paid to crowd workers, it is crucial to have access to the hardness index without delays.

***Wikisense*-based Hardness:** Summarizing the results of the *Wikisense*-based system [8], which is trained using different sets of training data after each Winograd Half is presented for evaluation, it requires on average 8 hours for every Schema Half.

***WinoReg*-based Hardness:** Summarizing the results of *WinoReg*, which is based on a machine learning approach that is trained on a variety of features prior to its use on a given Winograd Half, it requires on average 1.6 minutes for every Schema Half, without a significant training time for the construction of the model. The results ultimately show that with *WinoReg* we can deliver the hardness index of a Winograd Schema 300 times faster than the *Wikisense*-based approach.

Feature	Correlation Coefficient	Accuracy
All	33.1%	91.64%
-Narrative Chains	28.4%	91.47%
-Google Queries	29.8%	91.56%
-Rule-Based Polarity	28%	91.47%
-OpinionFinder Polarity	31%	91.61%
-Connective-Triple Relations	22%	91.3%
-Semantic Relations	30.5%	91.53%
-Word Relations	27.57%	91.53%
-ConceptNet	32%	91.62%
-Negation	31%	91.57%
-Sentence Type	23.7%	91.4%
-TextBlob Polarity	24.2%	91.45%
-Sentence Length	12.7%	91.17%

Table 2: Results of feature decrement experiments

4.3 Feature Importance

Based on the contribution of each feature used in *WinoReg* we performed a feature analysis. As shown in Table 2, where each row presents the performance of the model trained on all types of features except for the one shown in that row, the correlation coefficient drops significantly whichever feature type is removed. This suggests that all feature types are contributing positively.

It seems that the Connective-Triple, the Sentence-Type, the Sentence-Length, the TextBlob-Polarity, and the Word-Relations are the most useful features. This is in line with previous works that showed that the sentence length and the sentence type of each schema play an important role on the quality of the schema [10, 11]. Additionally, according to Rahman et al., the Word-Relations feature is one of the most important ones in their work [17].

As to the TextBlob-Polarity, it seems that it is better in capturing the polarity context of the candidate antecedents and the pronoun than the rule-based polarity and the OpinionFinder polarity features. Regarding the OpinionFinder-Polarity feature our analysis agrees with other studies [17]; the reason might be attributed to the fact that the corpus on which OpinionFinder was trained was quite different from ours.

It is somewhat surprising that, unlike in other studies [17], Google-based features are not among the most useful features. The reason might be attributed to the fact that the corpus on which the Google feature was used in previous studies was easier than our training set [17]. Furthermore, ConceptNet and Negation features seem to offer the least. Regarding the Negation feature, this might have happened because we were only able to determine if negation exists in only 41% of the Schema Halves. Although ConceptNet seems to have played an important role in other WSC studies [4], our results indicate that it is not among the most useful features. This might have happened because its similarity factor is based on different relations that cannot easily capture the semantics of each sentence.

5 Conclusion and Future Work

We have shown how a feature-based system that can be built and trained on a set of Winograd Schemas can form the basis for deriving a data-driven metric of hardness for WSC instances. Evidence that the system's computed hardness index is correlated with the perceived human hardness was offered through one study from the literature. Our experimental results suggest that *WinoReg* is more useful for achieving faster and better accuracy on the hardness of the Winograd Schemas, compared to systems previously used. On the other hand, our system still has a lot of room for improvement. In particular, our feature analysis indicates that further gains could be achieved via more accurate semantic analysis.

WinoReg can be used by WSC competition organizers, who wish to group sentences in terms of their human hardness. As an example application, the designers of CAPTCHAs could utilize our system to display schemas organized by their hardness index. Furthermore, experts who want to design new Winograd Schemas, as pursued, for example, in [10], could use our system to ensure that the generated schema hardness indexes are computed and displayed faster to their workers to ensure quality and to possibly reduce their schema design costs. Future studies will have to identify mechanisms through which we can develop systems like *WinoReg* that can achieve better accuracy with higher correlation coefficient. The use of other techniques like deep learning (e.g., [12]) might offer a viable alternative. Maybe, by calibrating the confidence scores predicted by neural network models would yield a faster and more flexible method for predicting the WSC hardness.

Acknowledgments

This work 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.

References

[1] Dan Bailey, Amelia Harrison, Yuliya Lierler, Vladimir Lifschitz, and Julian Michael. The Winograd Schema Challenge and Reasoning about Correlation. In *Working Notes of the Symposium on Logical Formalizations of Commonsense Reasoning*, 2015.

[2] Collin F Baker, Charles J Fillmore, and John B Lowe. The Berkeley Framenet Project. In *Proceedings of the 17th international conference on Computational linguistics-Volume 1*, pages 86–90. Association for Computational Linguistics, 1998.

[3] David Bender. Establishing a Human Baseline for the Winograd Schema Challenge. In *MAICS*, pages 39–45, 2015.

[4] Tejas Ulhas Budukh. An Intelligent Co-Reference Resolver for WInograd Schema Sentences Containing Resolved Semantic Entities, 2013.

[5] Nathanael Chambers and Dan Jurafsky. Unsupervised Learning of Narrative Event Chains. In *Proceedings of ACL-08: HLT*, pages 789–797, 2008.

[6] Hannah Fry. *Hello World: How to be Human in the Age of the Machine*. Random House, 2018.

[7] Nicos Isaak and Loizos Michael. Tackling the Winograd Schema Challenge Through Machine Logical Inferences. In David Pearce and Helena Sofia Pinto, editors, *STAIRS*, volume 284 of *Frontiers in Artificial Intelligence and Applications*, pages 75–86. IOS Press, 2016.

[8] Nicos Isaak and Loizos Michael. A Data-Driven Metric of Hardness for WSC Sentences. In Daniel Lee, Alexander Steen, and Toby Walsh, editors, *GCAI-2018. 4th Global Conference on Artificial Intelligence*, volume 55 of *EPiC Series in Computing*, pages 107–120. EasyChair, 2018.

[9] Nicos Isaak and Loizos Michael. Using the Winograd Schema Challenge as a CAPTCHA. In Daniel Lee, Alexander Steen, and Toby Walsh, editors, *GCAI-2018. 4th Global Conference on Artificial Intelligence*, volume 55 of *EPiC Series in Computing*, pages 93–106. EasyChair, 2018.

[10] Nicos Isaak and Loizos Michael. WinoFlexi: A Crowdsourcing Platform for the Development of Winograd Schemas. In Jixue Liu and James Bailey, editors, *AI 2019: Advances in Artificial Intelligence*, pages 289–302, Cham, 2019. Springer International Publishing.

[11] Nicos Isaak. and Loizos Michael. Winventor: A Machine-Driven Approach for the Development of Winograd Schemas. In *Proceedings of the 12th International Conference on Agents and Artificial Intelligence: ICAART,*. INSTICC, SciTePress, 2020.

[12] Vid Kocijan, Ana-Maria Cretu, Oana-Maria Camburu, Yordan Yordanov, and Thomas Lukasiewicz. A Surprisingly Robust Trick for Winograd Schema Challenge, 2019.

[13] Hector Levesque, Ernest Davis, and Leora Morgenstern. The Winograd Schema Challenge. In *Proceedings of the 13th International Conference on the Principles of Knowledge Representation and Reasoning*, 2012.

[14] Hector J Levesque. On Our Best Behaviour. *Artificial Intelligence*, 212:27–35, 2014.

[15] Leora Morgenstern, Ernest Davis, and Charles L. Ortiz. Planning, Executing, and Evaluating the Winograd Schema Challenge. *AI Magazine*, 37(1):50–54, 2016.

[16] Haoruo Peng, Daniel Khashabi, and Dan Roth. 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, 2015.

[17] Altaf Rahman and Vincent Ng. 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, 2012. Association for Computational Linguistics.

[18] Arpit Sharma, Nguyen H Vo, Somak Aditya, and Chitta Baral. 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, 2015.

[19] Robyn Speer, Joshua Chin, and Catherine Havasi. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[20] Marcus Suresh, Ronnie Taib, Yanchang Zhao, and Warren Jin. Sharpening the BLADE: Missing Data Imputation Using Supervised Machine Learning. In Jixue Liu and James Bailey, editors, *AI 2019: Advances in Artificial Intelligence*, pages 215–227, Cham, 2019. Springer International Publishing.

[21] Leslie G. Valiant. Knowledge Infusion. In *Proceedings of the 21st National Conference on Artificial Intelligence - Volume 2*, AAAI'06, pages 1546–1551. AAAI Press, 2006.

[22] Theresa Wilson, Paul Hoffmann, Swapna Somasundaran, Jason Kessler, Janyce Wiebe, Yejin Choi, Claire Cardie, Ellen Riloff, and Siddharth Patwardhan. Opinionfinder: A System for Subjectivity Analysis. In *Proceedings of HLT/EMNLP 2005 Interactive Demonstrations*, pages 34–35, 2005.

[23] Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. Recognizing Contextual Polarity in Phrase-Level Sentiment Analysis. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, 2005.