Experience and Prediction: A Metric of Hardness for a Novel Litmus Test

Article *in* Journal of Logic and Computation · February 2021

DOI: 10.1093/logcom/exab005

2 authors:

Nicos Isaak

18 PUBLICATIONS **32** CITATIONS

SEE PROFILE

Loizos Michael

126 PUBLICATIONS **903** CITATIONS

SEE PROFILE

Experience and Prediction: A Metric of Hardness for a Novel Litmus Test

Nicos Isaak[1 [0000−0003−2353−2192]] and Loizos Michael[1,2]

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

*This paper has been accepted and published by the Journal of Logic and Computation. The authenticated and peer-reviewed version of this paper is available online at
https://doi.org/10.1093/logcom/exab005

Abstract. In the last decade, the Winograd Schema Challenge (WSC) has become a central aspect of the research community as a novel litmus test. Consequently, the WSC has spurred research interest because it can be seen as the means to understand human behavior. In this regard, the development of new techniques has made possible the usage of Winograd schemas in various fields, such as the design of novel forms of CAPTCHAs.

Work from the literature that established a baseline for human adult performance on the WSC has shown that not all schemas are the same, meaning that they could potentially be categorized according to their perceived hardness for humans. In this regard, this *hardness-metric* could be used in future challenges or in the WSC CAPTCHA service to differentiate between Winograd schemas.

Recent work of ours has shown that this could be achieved via the design of an automated system that is able to output the hardness-indexes of Winograd schemas, albeit with limitations regarding the number of schemas it could be applied on. This paper adds to previous research by presenting a new system that is based on Machine Learning (ML), able to output the hardness of any Winograd schema faster and more accurately than any other previously used method. Our developed system, which works within two different approaches, namely the random forest and deep learning, is ready to be used as an extension of any other system that aims to differentiate between Winograd schemas, according to their perceived hardness for humans. At the same time, along with our developed system we extend previous work by presenting the results of a large-scale experiment that shows how human performance varies across Winograd schemas.

Keywords: Winograd Schema Challenge · Schema Hardness · Machine Learning · Random Forest · Deep Learning.

1 Introduction

Since the late fifties, the AI community is concerned with endowing machines with commonsense and reasoning [25, 43]. To that end, a number of challenges have been proposed to advance the field of AI, aiming to behoove AI researches build systems able to help or replace humans in day-to-day life. One of these challenges is the WSC, a carefully-crafted pronoun resolution task and a variant of the well known recognizing-textual-entailment challenge (RTE) [10] that is able to capture basic human abilities.

Given a Winograd schema, people can anticipate and reason about causes and effects [33], and tell you who did what to whom, when, where, and why [13]. For instance, if someone tells us that,"The city councilmen refused the demonstrators a permit because they feared violence." and asks as "who feared violence?", we can easily infer that the correct answer is the "city councilmen". This example shows how humans, through commonsense and reasoning, can answer such questions. On the other hand, we know that current AI do not have that day to day commonsense and reasoning that humans do [13, 25].

In a recent work, Bender [2] established a human baseline for the WSC, where it was shown that adults can tackle the challenge with a mean of 92%. Along with the results, the authors have shown the importance of having humans to evaluate the schemas upon designing, as not all schemas have the same perceived hardness for humans [2, 28]. Given that the WSC was developed to help humans design systems that mimic human behavior, it seems that shedding light on the perceived human hardness for schemas would be useful for the challenge itself. In this regard, this metric of hardness could be used, i) to categorize schemas according to the strengths and weaknesses of a particular group of participants, and, ii) in the WSC CAPTCHA service that uses Winograd schemas to identify humans from bots [19].

In a past work of ours, we approached this problem by reusing the *Wikisense* system [17] for resolving Winograd Schemas, and, roughly, by using the amount of training data it requires to correctly answer a given schema as an indicator of its hardness. This resulted in a system [18] that correlates well with the performance of humans, 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 very time consuming. To do that, we compared the Wikisense-approach results to humans' performance on a dataset of 144 schemas [2].

In this work, we consider a new novel approach called WinoReg (from Winograd-Regression) which, through machine-learning, can deliver faster and more accurate results than our previous work. To that end, we build a new system that works within two different approaches, i) the Random-Forest approach, which directly relates with feature engineering, and, ii) the Deep-Learning approach, which requires access to the hardness indexes of more Winograd schemas. In this regard, we extended Bender's work with a study that we designed and undertook, which involved 306 crowdsourced workers and 943 schemas.

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 a

number of works in the literature that have developed WSC-related systems, which we have re-implemented as needed [7, 17, 30, 32, 35].

In the next sections, we start by presenting the challenge itself. We continue with our motivation section followed by the human-adult performance section. A high-level analysis of WinoReg's architecture is outlined in the fifth section, whereas a more detailed analysis of the Random Forest and the Deep-Learning approach is given in the next two sections. We present the experiments along with our results in section eight. Finally, in the next sections, we present some highlights of previous work along with potential implications and recommendations for future research.

2 Challenge Basics

Broadly speaking, the WSC is about resolving ambiguities because the information needed is not grammatically present in the examined schemas. Consequently, the Winograd schemas comprise of two Winograd halves, with each half consisting of a sentence, a definite pronoun or a question, two possible pronoun targets (answers), and the correct pronoun target [24]. The following schema (a pair of halves) illustrates the key characteristics of the challenge:

– – First-half: *Sentence: The city councilmen refused the demonstrators a permit because they feared violence. Question: Who feared violence? Answers: The city councilmen, The demonstrators. Correct Answer: The city councilmen.*
– – Second-half: *Sentence: The city councilmen refused the demonstrators a permit because they advocated violence. Question: Who advocated violence? Answers: The city councilmen, The demonstrators. Correct Answer: The demonstrators*.

Given just one of the halves, the aim is to resolve the definite pronoun through the question to one of its two co-referents. To avoid trivializing the task, the co-referents are of the same gender, and both are either singular or plural. The two halves differ in a special word or phrase that critically determines the correct answer. Schemas that do not *strictly* follow these rules are called "schemas in the broad sense".

It is believed that the WSC can provide a more meaningful measure of machine intelligence when compared to the Turing Test [25]. This happens because of the presumed necessity of reasoning with commonsense knowledge to identify how the special word or phrase affects the resolution of the definite pronoun. The challenge is already in full swing with other AI challenges that aim to tackle the goal of endowing machines with human commonsense and reasoning. By extension, it is believed that a system that contains the commonsense knowledge to correctly resolve Winograd schemas should be capable of supporting a wide range of AI applications [24].

3 Motivation

It is widely believed that Winograd Schemas are easy for humans and hard for machines because they require the use of commonsense knowledge to correctly resolve the definite pronoun [25]. According to Levesque, in every schema, you need to have background knowledge that is not revealed in the words of the sentence to be able to clarify what is going on [24].

Broadly speaking, due to schema discrepancies, 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. Additionally, with every single schema any potentially developed system should presumably be able to demonstrate how humans tackle it, meaning, that, there are different kinds of schemas.

What we know about the perceived human hardness index on the WSC is largely based on Bender's work [2], who, through an experiment he undertook, identified that human adults tackle the WSC with a mean of 92%. In a past work of ours [18], towards answering the previous question, we started by considering the *Wikisense* system [17], which is a commonsense & reasoning system able to resolve a number of Winograd schemas. Basically, Wikisense parses each examined schema to identify the necessary keywords to search for relevant Wikipedia sentences. Next, for every Wikipedia sentence, it returns semantic scenes, which are triples based on nominal-subjects and direct-objects returned by a dependency parser. These semantic scenes are fed to a Learner that constructs the necessary knowledge, which can be searched through a Reasoner for the tackle of the challenge.

Given that Wikisense gets its training data in real time from the English Wikipedia, we developed a new system —Wikisense-based approach— whose performance improves as it gets more training data while its trying to resolve a given schema. Specifically, we have found that the amount of training data needed for the resolving of a given schema correlates positively with the perceived human hardness index of that specific schema. However, the resulting model was able to offer the hardness index on only 57% of our tested schema halves, which is in direct relation with the keyword implementation of Wikisense that is based on the semantic analysis of the given schemas; If Wikisense cannot extract a keyword then the Wikisense-based approach cannot return the hardness index of the examined schema. Additionally, because of its dependency on training during query-answering, it was found that the *Wikisense*-based system needs, on average, eight hours to output the hardness index of given schema.

Broadly speaking, the Wikisense-based approach results are disproportional to the demand of new developed schemas, in the literature. For instance, in a recent work of ours we have demonstrated how the WSC can form a novel form of CAPTCHAs [19], with the ultimate goal of bringing more AI researchers to work on the challenge. Like in every other CAPTCHA service, there is a high demand of new Winograd schemas which could serve as the means to identify fraudulent actions. In this regard, systems like the Wikisense-based approach could be used to make sure that the CAPTCHA service would display harder schemas to solve in the case of possible fraudulent actions. Furthermore, in the case of humans it could be used to ensure that the generated instances are not overly demanding. Additionally, there are systems that are already in full-swing with the Wikisense-based approach, meaning that they already use its mechanisms to differentiate schemas according to their perceived hardness for humans. In this regard, WinoFlexi [20], which is a crowdsourced collaboration platform for the development of Winograd schemas from scratch, leverage Wikisense-based approach to provide feedback to workers regarding the *quality* of their developed schemas.

This raises many questions regarding whether we should look for alternative solutions which are based on different techniques. In this regard, to find a faster and

more accurate way to output the hardness index of Winograd Schemas, we consider WinoReg, which is a system based on machine a learning approach. Through experience and prediction WinoReg learns how to predict the hardness of a given schema based on two different approaches, i) the Random-Forest, and ii) the Deep-Learning approach. Before proceeding with WinoReg's architecture, it is interesting to briefly review Bender's work regarding human perceived hardness on the Winograd schemas.

4 Human-adult Performance on the WSC

Bender [2], through an experiment he undertook, which involved the participation of adult English speakers, identified that human adults tackle the WSC with a mean of 92% . Furthermore, it was found that adults need, on average, 15 seconds to answer a given schema.

To the best of our knowledge, this is the only set available to provide us with the necessary training and testing data [18]. In his work, he used schemas developed by experts —Levesque et al.'s dataset [24, 28]– which, at the time of writing, consisted of 144 schemas (288 Schema Halves). The experiment ran on Amazon's Mechanical Turk where 407 adult speakers, who speak English fluently, participated. Results showed that adult speakers are, on average, able to correctly resolve 92% of the Winograd schemas, which sets the bar very high, compared to what systems can achieve [30, 32, 27, 22]. On the other hand, in the experiments it was shown that there are schemas that are harder to resolve than others; for instance there are schemas that humans scored a mean of 45%. A detailed analysis of human performance on each individual WSC instance (accuracy) is available from: `https://github.com/benderdave/wsc-exp.git`.

5 High-level Analysis of WinoReg's Architecture

Here, we perform a high level analysis of WinoReg's Architecture (see Figure 1). WinoReg works in two operational modes, namely, the random-forest, and the deep learning mode. In both modes, it outputs the hardness of any schema through regression analysis, where it examines the relationship between the schema halves and the perceived human hardness indexes[2].

After the training it uses the learned model for faster computation during its deployment. Regarding the feature engineering of the Random-Forest approach, these features come from a number of works in the literature that have developed WSC-related systems, which we have re-implemented as needed [7, 17, 30, 32, 35]. Specifically,within the Random Forest mode, WinoReg analyzes each schema to output a required number of features. Next, all of the features are given as an input to the learned model to output the hardness of a schema half. On the other hand, within the Deep-Learning approach, WinoReg does not require to estimate the values of features, meaning that any given schema can be given directly to the model to acquire its hardness index. In both cases, WinoReg can load a schema from a schema-database to output its hardness index, which is a value in the range of 0-1. Compared to Wikisense-based approach, none schema is discarded.

In the next sections, we will show how Winventor 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.

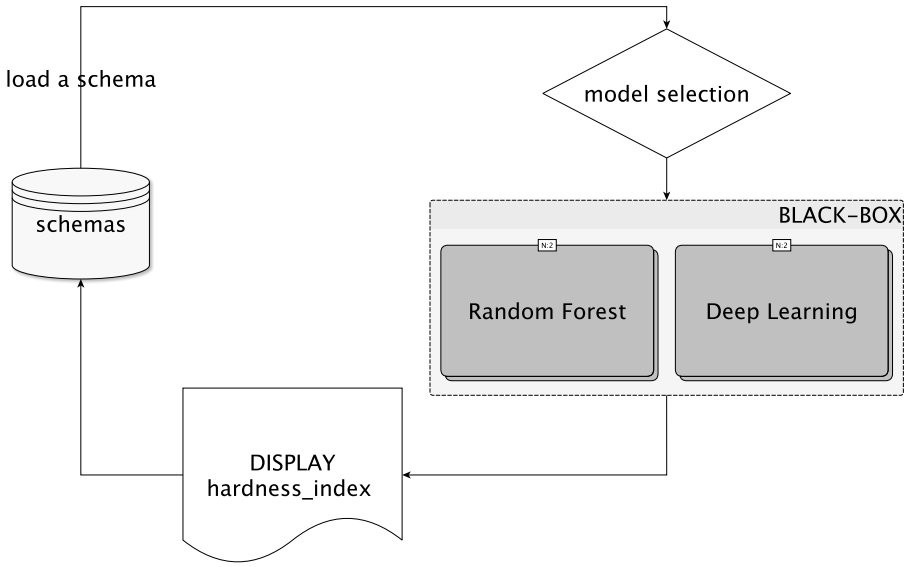

Fig. 1: WinoReg's Architecture to compute the hardness indexes of Winograd schemas. The black-box shows that the system can work in two distinct modes.

6 WinoReg: A Random Forest Approach

Within this approach *WinoReg*, is based on training a regression model with the use of Decision Trees. We use, in particular, the Random Forest algorithm [12], which was introduced in 2001 [6]. The Random Forest algorithm, which involves the construction of an ensemble of Decision Trees, each trained on random subsets of the data, showed significant improvements in accuracy of different kinds of problems[6]. A recent line of research showed that it is one of the best algorithms that maintain high imputation performance on linear regression across a range of performance metrics [42]. Like any other Machine Learning algorithm, the focus of random forest algorithm is to form a rule with reasonable accuracy, which could be used as a prediction tool on future data [31]. In this regard, we aim to train a Random Forest algorithm able to estimate the perceived human hardness index of Winograd schemas (see Figure 2).

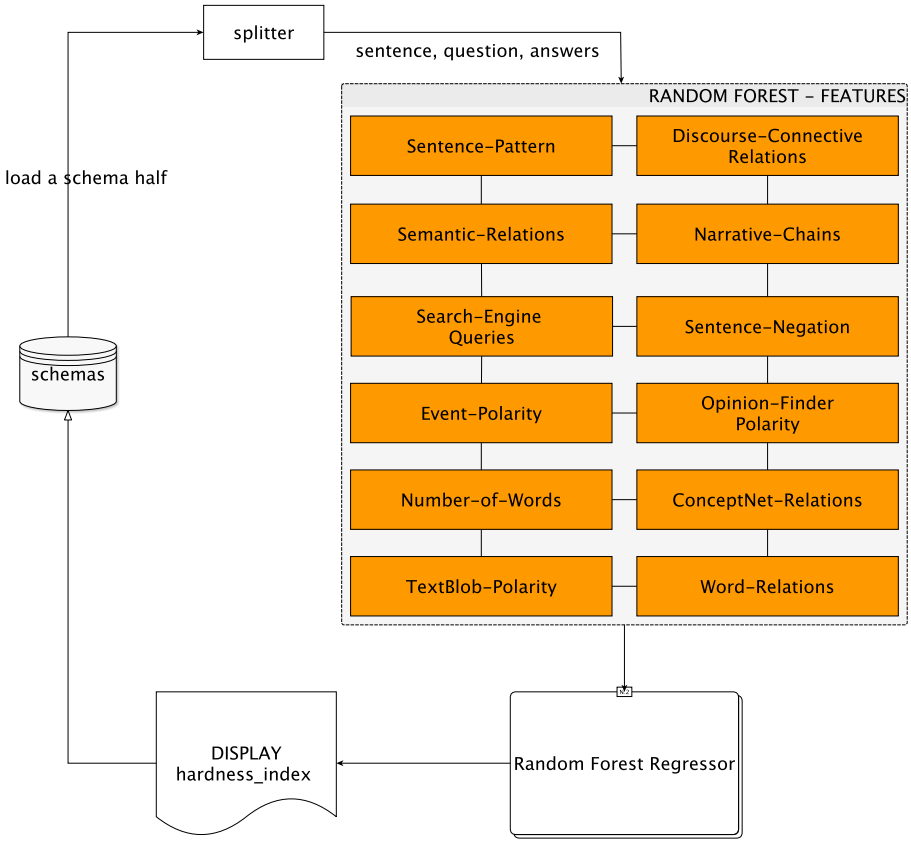

Fig. 2: WinoReg's Architecture based on Random-Forest: Given a Winograd schema WinoReg outputs the perceived human hardness index.

6.1 Feature Preparation

Within ML, we need to transform our data to find the appropriate representations to make it more manageable to the task at hand [11]. As we want to estimate the hardness index of a Schema Half, which indirectly relates to the selection of the correct answer, our Random-Forest approach expects features related to the schema half parts — sentence, question, and the two pronoun targets (candidates). Compared to *Wikisense*-based system *WinoReg* does not make use of the correct answer of each schema.

To train our system we use 50 features from 12 components that we built from scratch. The majority of these features, are based on non open-source systems, from the literature, that were previously used for the tackle of the challenge [7, 18, 30, 32]. Most of these features relate to semantic relations that are taken from each examined schema. In this regard, our system uses the *spaCy* [3] dependency parser to turn raw-text into semantic relations; these are relations that show how the sentence words are related to each other. According to the literature, the semantic relations are considered good if they can express the structure of the text and can differentiate, at the same time, between the events and their participants [35]. 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 [17, 21].

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 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 detailed analysis of the feature development process is given in the next paragraphs.

6.2 Sentence-Pattern

In a recent work, we have shown shown that the structure of each schema-half's sentence plays an important role in its quality [20]. It seems that schemas that are developed using a variety of sentence patterns/types —complex, compound-complex–, are preferable than schemas that are based on simple types. In this regard, to design our first feature we make use of a tool that is able to output the sentence-type of each examined schema [20] (stored as ST). Given any English sentence, the tool is able to output its type which can be either a simple, a compound, a complex, or a compound-complex sentence. At the same time, it outputs it's pattern/clause (e.g., "SV because SV", "SV and SV because SV.", "Cause/Effect"), which directly relates to the connectors that each sentence uses between its clauses (stored as SP). Hence, within this component we are able to engineer two features, namely ST & SP.

[3] `https://spacy.io`

6.3 Sentence-Negation

It is widely accepted that *negation* plays an important role into capturing the semantics of text, as it is used to reverse the polarity of parts of a statement [5, 17]. To encompass these kind of rules, we analyze each Schema Half to estimate if the two candidates and the definite pronoun are governed by negation; this is done via the sentence & question triples of the Schema Half (see *catch* example). In this regard, from the *negation*-component we create two binary features, (*STN* for the two candidates, and *QTN* for the definite pronoun) that contain the value of 1, if negation exists, and, otherwise, the value of 0.

6.4 Schema's Semantic-Relations

This component directly relates to the semantic relations of a given text. As stated in the literature [32], via web queries we might phase precision and recall problems. Specifically, when a pronoun target and a verb appears next to each other, it does not mean that a subject-verb relation exists between them —precision problem. On the other hand, these queries fail to obtain subject-verb relations where a pronoun target and verb are not close to each other—recall problem. To eliminate these kind of problems, we search Wikisense's Wikipedia-corpus to see how many times each pronoun target appears as subject or as an 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 times; Otherwise, if the pronoun appears as an object, we search to find which pronoun target appears most of the times 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, otherwise 2 if it has the same role with the second pronoun target. If we cannot determine their roles then SEM equals -1.

6.5 Number-of-Words

It seems that the sentence length of each schema directly relates to the resolution of the definite pronoun [21], where schemas with longer sentences tend to be harder to answer. In accordance with our findings, we engineered a feature that directly relates, in terms of words, to the length of each sentence (SL).

6.6 Word-Relations

Word-relations features relate to candidate-independent, and candidate-dependent relations, where, according to previous works [32], they seem to play an important 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; these 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 the candidate-dependent features we engineer three features, namely HN, VF, and AF. Specifically, HN contains the number of the head words of the two candidates that were returned by the dependency parser; if we cannot determine the two candidates in the 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.

6.7 Search-Engine Queries

Recent work has shown that search-engine queries are able to provide us with world knowledge, which is useful for the tackle of the challenge [30, 32, 35]. For instance, in the *catch* example, we can acquire world knowledge to learn that someone who is 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 schema we build six queries, namely QR1: A1VQ, QR2: A2VQ, QR3: A1VQW, QR4: A2VQW, QR5: JA1, QR6: JA2; A1 and A2 are the two candidates, 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, using the number of hits that were returned by the search engine, we built eight binary features —GL1i1, GL1i2, GL2i1, GL2i2, GL3i1, GL3i2, GL4i1, GL4i2—, as in Rahman & Ng [32]. The first two features (GL1i1, GL1i2) are computed from QR1 & QR2, the next two (GL2i1, GL2i2) from QR3 & QR4, and the third (GL3i1, GL3i2) from QR5 & QR6 (The last two features are computed based on the results returned from all of the queries). For instance, if the absolute value of —QR1, QR2—, is bigger than the threshold of 20% in favor of the first candidate, then, GL1i1 equals 1 and GL1i2 equals 0; 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 originally introduced [32].

To avoid problems with proper-names (persons) where we cannot retrieve search query hints we make use of Framenet [1]. As stated in other works [7, 32], it is unlikely that search engines will return meaningful counts for persons. In this regard, in a schema where the candidates 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 successful search from Framenet we replace the persons with their Framenet roles. Consequently, we form six queries and search the Google engine to generate eight features: GLF1i1, GLF1i2, GLF2i1, GLF2i2, GLF3i1, GLF3i2, GLF4i1, GLF4i2.

6.8 ConceptNet-Relations

ConceptNet is a freely available semantic commonsense toolkit [26]. 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 [26, 36, 39]. In contains concepts about common basic knowledge about various facts, connected with other facts, using different kind of relations (e.g., *relatedTo, AtLocation, IsA, PartOf*) [7]. Our system, makes use of ConceptNet to find possible relations between the two candidates 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 relatedness value of the first candidate is greater then the value of the second candidate; if the opposite exists then then value of CN equals 2, and, if we cannot find any difference, it equals -1. Additionally, like before, we consider Framenet [1] for issues with proper names, and create the CNF feature, where its values are being computed in the same way as the CN values.

6.9 Discourse-Connective Relations

As reported by Rahman & Ng [32], causal relations, which are signaled by discourse connectives, show the world knowledge between events. For instance, in the 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 help us resolve the definite pronoun ”it” to the lion.

For each schema half, we search the *Wikisense* 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 candidates, 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 the number of occurrences is at least 100 then we proceed to the next step [32]; iii) if X is a verb, then it resolves the pronoun to the candidate that has the same role as the definite 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. To encode this heuristic decision we create a binary feature (CNT); CNT equals 1 if the definite pronoun is resolved to the first candidate, and 2 if it is resolved to the second candidate. Otherwise, in case we cannot resolve the definite pronoun, CNT equals -1.

6.10 Event-Chaining via Narrative Chains

Narrative-chains are sequences of events, in a story that shows the role of the protagonist/actor, which is denoted as *-s*: subject or *-o*: object [7, 32]. To the best of our knowledge, one of the best available narrative-chain datasets is the Chambers and Jurafsky's narrative chains [8]; these are ordered sets of 12 events (verbs) centered around a common protagonist that show its role in the chain (subject or object).

For every schema half, we determine the events the two candidates and the definite pronoun participate in along with their protagonist role (subject or object). For instance, in the next 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.*, via Wikisense mechanisms we output two triples: i) refused (x-subject, y-object), ii) advocate (they-subject, violence-object). Hence, in this example, 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, and for each such pair, we extract all the chains that contain both elements (refuse and advocate). For instance, in our example, Chambers & Jurafsky 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* cannot find narrative chains containing both elements, it runs again the same procedure but with a similarity mechanism enabled. In the end, we create a feature (NCH) that equals 1 if the answer is the first candidate, and 2 if it is the second candidate. Otherwise, if we cannot output triples or find any narrative chains, it equals -1.

6.11 Event-Polarity with Heuristic Rules

Word polarity, which has been widely studied in the NLP field [14], can help us to resolve the definite pronoun in specific schema halves [7, 30, 32]. This is a straightforward procedure that can be summarized in three steps: i) find the polarity of the definite pronoun; ii) determine the polarity of the two candidates; iii) select the candidate that has the same polarity as the definite pronoun. To find the polarity values we use the Wilson et al. subjectivity lexicon [45], a lexicon that assigns to various events their polarity, such as negative, positive, or neutral.

Let us use the following example to explain the procedure we follow to assign the polarity values: *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 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. subjectivity lexicon [45], 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 *demonstrators* becomes positive. Additionally, we know that the polarity of the event *advocate* in the subjectivity lexicon is positive, hence 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 lead us to resolve the definite pronoun —they— to *demonstrators*.

The event-polarity procedure lead us to the engineering of six binary features, namely, RP1i1, RP1i2, RP2i1, RP2i2, RP3i1, RP3i2. Initially, all of these features are set to zero. The first two features, RP1i1, 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 & RP2i2) are the concatenation of the polarity values, determined for both the definite pronoun and the two candidates; 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 [17, 32]. 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 enhance the polarity features by creating an additional feature (RPTL) that shows the best pronoun target.To that end, we simply 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.

6.12 Event-Polarity with OpinionFinder

This is a machine-based polarity that uses a sentiment-analyzer to resolve the definite pronoun of a schema-half. As stated in other works [30, 32], instead of using a heuristic approach to estimate the polarity values, we use OpinionFinder [44], which is a machine driven approach able to perform subjectivity analysis. With tools like OpinionFinder we can easily annotate phrases with their contextual polarity values. To that end, we compute the OpinionFinder polarity features in the same way we did with the rule-based polarity features, and create seven features (OP1i1, OP1i2, OP2i1, OP2i2, OP3i1, OP3i2, OPTL).

6.13 Event-Polarity with TextBlob

Given that our previous polarity features are based on similar approaches, namely, Wilson et al. subjectivity lexicon [45] and Wilson et al. OpinionFinder [44], here, we use another, simpler polarity mechanism —called TextBlob-Polarity [4]. This is an NLP library that can process textual data and output, among others, the events' polarity values. Specifically, with the TextBlob's sentiment analysis we return the polarity of the verb that governs the two candidates, and the polarity of the verb that governs the definite pronoun. Finally, we create two features (TBSPOL, TBQPOL) that can be either neutral, positive, or negative.

7 WinoReg: A Deep-Learning Approach

Within this approach we train WinoReg using deep learning (see Figure 3), which is another increasingly popular method inspired by the biological brain [4, 11, 23]. 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 [34], albeit the term deep learning with the current resurgence started in 2006 [4, 37].

[4] `https://textblob.readthedocs.io/en/dev/`

Techniques that incorporate deep learning have been steadily gaining in popularity [4]. In this line of research, deep learning have won numerous contest, in pattern and image recognition, and achieved promising results on different NLP tasks [11, 34]. With deep learning algorithms, machines could learn good representations of data to help NLP tasks enormously. Specifically, deep learning seems to help in building constitutionality into Machine Learning models, just like human languages do to give meaning to complex ideas [37]. 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 with the schema hardness. In this regard, here, we train WinoReg within a Deep-Learning approach that is able to estimate the perceived human hardness indexes of Winograd schemas (see Figure 3).

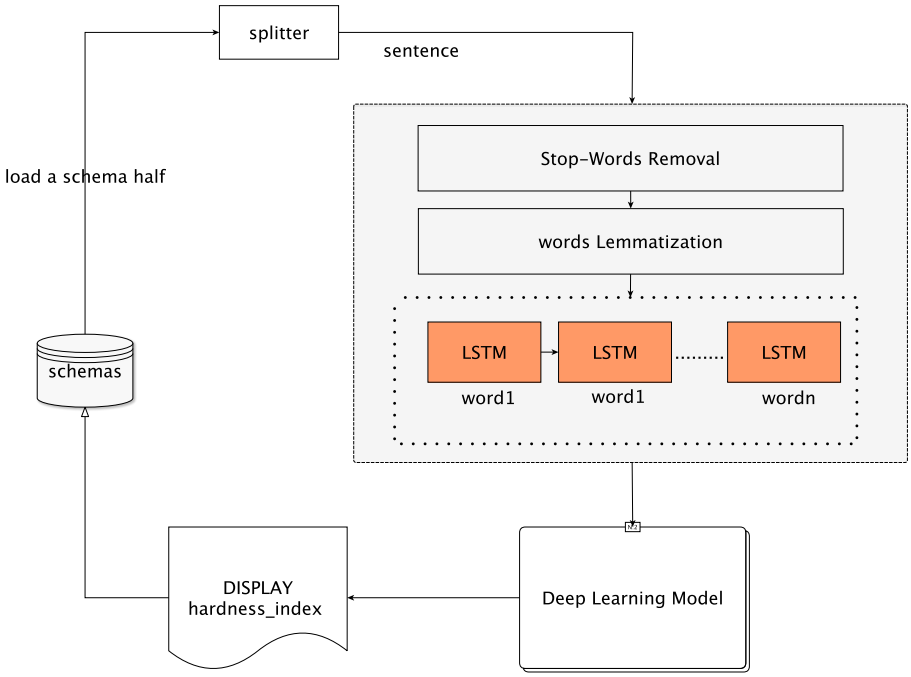

Fig. 3: WinoReg's Architecture based on Deep-Learning: Given a Winograd schema WinoReg outputs the perceived human hardness index.

7.1 Data Enhancement via Crowdsourcing

Although it is debatable [13], it is widely accepted that deep learning killed feature engineering, which is time-consuming and brittle [37]. As stated in the literature, most of the times conventional ML techniques require considerable domain expertise for

feature engineering [23]. On the other hand with deep learning, the amount of skill required for feature engineering reduces as the amount of training data increases [4].

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 [4]. Additionally, if we can provide deep learning with sufficient amount of data [23], it will also reduce the generalization error, aka over-fitting [4].

In our case, the availability of training data is limited since we only have access on 143 schemas. In this regard, to increase Bender's training data we run an experiment on the **MicroWorkers (MW)** platform [5], which offers a reliable solution for various fields and research purposes [29]; Amazon Turk, which was used in Bender's experiments, was not available in our region. In the next lines, we will explain how we designed and ran our experiment along with our results.

7.1.1 Dataset: According to the literature, only two well-known datasets exist, i) the Davis et al dataset, which was used in Bender's experiment, and, ii) the Rahman & Ng's dataset, which consists of 943 schemas. Given that all of the Davis at al. dataset was used in Benders experiment, we designed our experiment based on Rahman & Ng's dataset. The main difference between the two datasets is the missing of questions in Rahman & Ng's schemas —it only contains the definite pronoun. To match Bender's experiment we manually developed and added the necessary questions in all of the schemas. For the sake of simplicity, in our questionnaire, we use only the first half of each schema.

7.1.2 Materials: For the design of the questionnaire we used LimeSurvey software from our lab server [6]. All materials used in the experiment, including the schema halves used, are available online [7].

7.1.3 Participants: The questionnaire started in April 2020 and ran for two months. According to our results, a total of 306 participants from English speaking countries attempted and finished the task. 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 (see 7.1.4). The total cost of our campaign was $322. In the end, every schema half was answered by at least 30 participants.

7.1.4 Design: We built the questionnaire and posted the link on the Microworkers platform. A total of 943 schema halves were included, where, each half was displayed on a single screen. Each schema-half's sentence was displayed at the top, followed by the question, and the two possible answers that were displayed alongside (see Figure

[5] https://www.microworkers.com
[6] http://limesurvey.org
[7] https://github.com/NicosCg/wsc-experiment

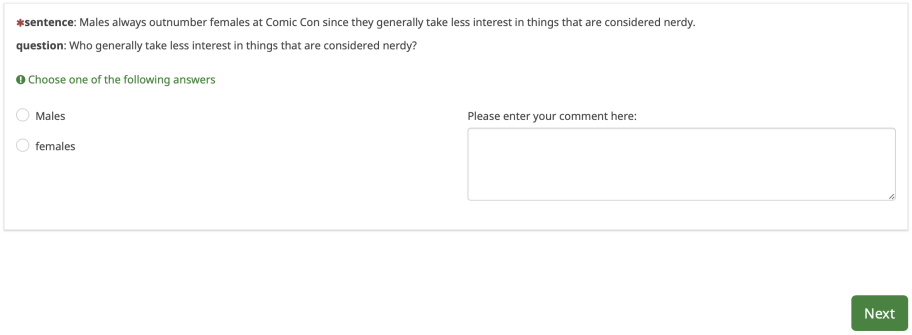

Fig. 4: Screenshot of experiment window.

4). Additionally, their was a comment section for participants to offer any comments they might have. All of the participants were informed that once the survey started, they could not change a submitted answer. Compared to Bender, our workers were not given an immediate feedback (correct or incorrect) after each trial, nor, by extension access on their updated score.

Our questionnaire consisted of 10 sections that ran independently. Each section included 100 unique schema-halves except for the tenth, which included the last 43 schemas of the dataset. Each participant was allocated only one position, meaning that they were allowed to 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, their English language literacy level, and pass a training phase to get familiarized with the task; in the training phase immediate feedback (correct/incorrect) was given to the participants. Ostensibly, instructions were given as a warning not to sacrifice accuracy for speed.

To avoid problems related to cheating we also included a number of test questions that were randomly displayed among the other schemas. As dealing with cheating in crowdsourcing platforms is a major challenge [20], test questions were used to verify if a given worker indeed holds a particular skill [9, 15]. Via an adaptive interjection of test questions at any time in any given place we aimed on the selection of the answers of really motivated participants. In this regard, in the end we selected only the answers of participants who at least scored 70% on the test questions. Note that all participants were a priori informed about the test question mechanism.

The testing phase consisted of 10 schema halves that were designed specifically to select, in the end, the answers of the best participants. According to Bender, a lot of schemas suffer from ambiguity, meaning that it is difficult even from humans to answer them [2]; this is related to the fact that the design of schemas is too difficult and troublesome [28]. In this regard, the testing questions were designed in a way to directly show the correct pronoun antecedent (correct answer), 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.*

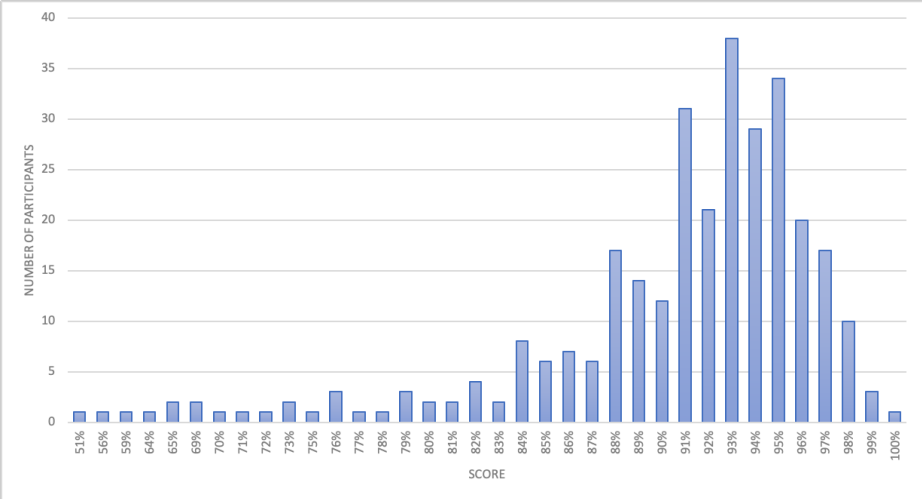

Fig. 5: Questionnaire results: Distribution of scores.

7.1.5 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 schema-half (see Figure 5). Our experiments are in line with Bender's results [2], 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 differentiate a lot. Specifically, in Bender's work it was noted that the majority of Rahman and Ng's seems to be easy Winograd schemas. On the other hand, our results do not seem to confirm their observation. In fact, it seems that the hardness indexes of the two datasets are similar, meaning that human adults make the same effort to solve them.

7.2 Fine-tuned dataset

To get more data for our Deep-Learning approach we took the 943 schema halves of the Rahman & Ng's dataset 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 Rahman & Ng's dataset over 286 schema halves of the Davis et al. dataset—, through oversampling, we increased the number of observations of the Davis et al. dataset; these were copies of the existing schema halves, excluding the 100 schemas used for testing purposes. The whole process resulted in 1872 schema halves, which were used for training purposes.

7.3 WinoReg's Deep-Learning Architecture

WinoReg's deep-learning architecture is based on LSTM networks (see Figure 3), an updated version of RNNs that are capable of learning long-term dependencies [16];

Specifically, LSTM networks may be also interpreted as something similar to a computer memory [41]. As stated in the literature, LSTM neural networks perform really well in the field of language modeling (LM) [41] which can be used to solve various NLP tasks [22]. 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 absent of features, with LSTM networks WinoReg can learn the joint probability function of sequences of words in a given sentence [3], and at the same time, take into account all of the predecessor words [40, 41] to output the perceived human hardness index of any given schema.

Within this approach, WinoReg split each examined schema-half to select the sentence, as this is the only input-value that is needed for our Deep-Learning approach (see Figure 3). 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.

8 Experimental Evaluation

In this section, we present our results by applying the methodology described in this paper. In this regard, we undertook several experiments to investigate if WinoReg can be used to automatically differentiate between Winograd schemas based on their perceived hardness for humans. We start by presenting WinoReg results based on the Random-Forest approach and continue with the Deep-Learning approach.

8.1 Random-Forest

Here, by using the data from Bender's study [2], we examine whether the performance of the Random-Forest approach can be predictive of the hardness of the WSC instances for humans. The results are reported on the testing set, which comprises 30% of the Davis et al. dataset (288 schema halves), expressed in terms of accuracy and correlation coefficient. For comparison purposes, the testing set is identical to the one that was used in our first work [18]. According to Bender's results, the human adult bar on the testing set (100 schemas halves) is 91%.

8.1.1 Results & Discussion

The fixed baseline: For comparison purposes, we trained our Random Forest algorithm with only one feature, which is the human adult bar of 91% and, like in our first work we tested it on the first 100 Schema Halves. Not surprisingly, our results show an achievement of 90.87%, but with a Correlation Coefficient with the adults results, of -1 (see the Fixed-Baseline in Fig. 6).

Wikisense-based Hardness: Recall that the Wikisense-based hardness is able to return results only for 57% of the examined schemas with a correlation coefficient of 38%

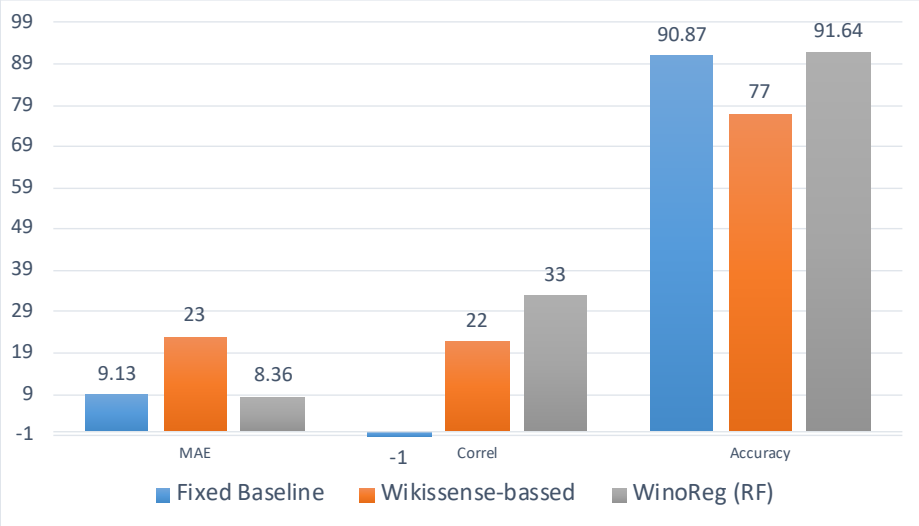

Fig. 6: Results of the Fixed Baseline, the Wikisense-based hardness, and WinoReg, which was trained based on the Random-Forest approach.

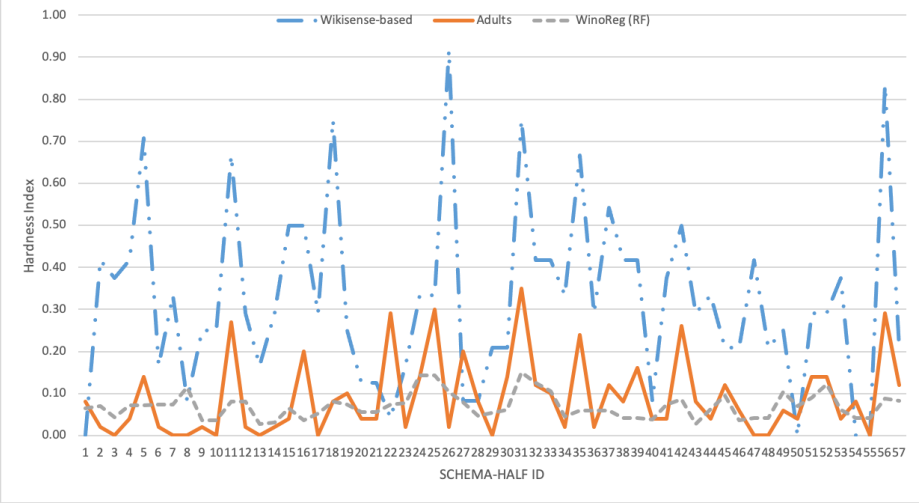

Fig. 7: Variability of WinoReg and Wikisense-based hardness-index across the 57 WSC instances on which the Wikisense-based approach originally was computed (in relation to the variability of the human hardness-index).

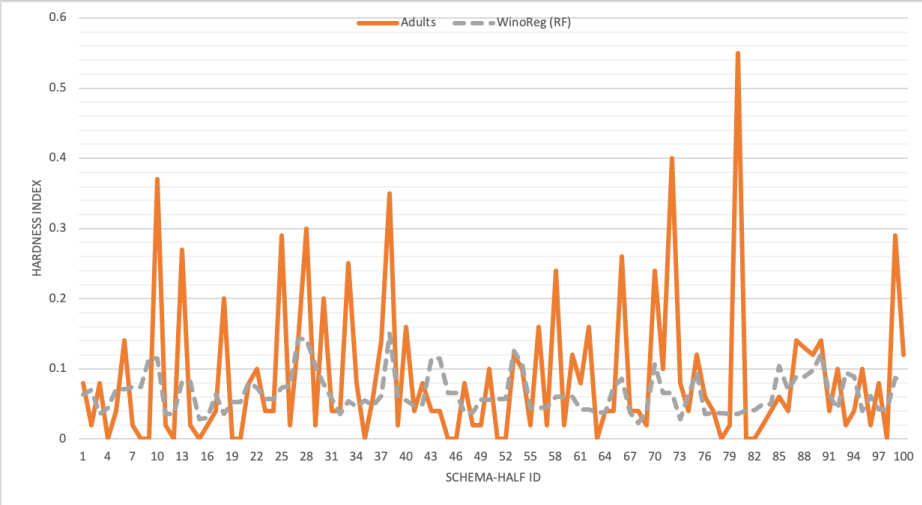

Fig. 8: Variability of the WinoReg hardness-index and the perceived human hardness-index across our testing set (100 schema halves). WinoReg is trained based on the Random-Forest approach.

[18]. It seems that Wikisense, the engine behind the system, was unable to output the necessary keywords to search the Wikipedia corpus. Given that the human adult bar on our testing set is 91% (for the unresolved schemas), we can assume that Wikisense-based achieves an accuracy of 77% on all of the remain schemas, with a correlation coefficient of 22% (see Fig. 7).

WinoReg: The general picture emerging from the analysis is that WinoReg can achieve an accuracy of 91.64%, significantly outperforming the *Wikisense*-based approach by 14.64% in accuracy and by 11% in correlation coefficient (see Fig. 8). To make a better comparison between WinoReg and Wikisense-based approach, we compared the two methods only on the 57 Schema Halves the *Wikisense*-based system was able to resolve. In this regard, the correlation coefficient of *WinoReg* and humans rises to 47%, which is 9 percentage points bigger than what the *Wikisense*-based system 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 instances 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, Fig. 8 and Fig. 7 that depict 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 *WinoReg*.

8.1.2 Speed Analysis

Given that the hardness index plays an important role in the quality of the developed schemas and that there are crowdsourcing and machine driven approaches that already

leverage the Wikisense-based hardness mechanism [20], it is crucial to have access to the hardness index without delays. In this regard, we performed a speed analysis to show how fast WinoReg can provide us with the hardness index of Winograd schema-halves. Compared to the Wikisense-based approach, which requires on average 8 hours for every Schema Half, it was found that WinoReg can return the hardness index of a schema half, on average, in 1.6 minutes; this is the time needed for the estimation of the required features that are fed to the Random-Forest model. The results ultimately show that *WinoReg* can deliver the hardness index of schemas 300 times faster than the Wikisense-based approach.

8.1.3 Feature Analysis

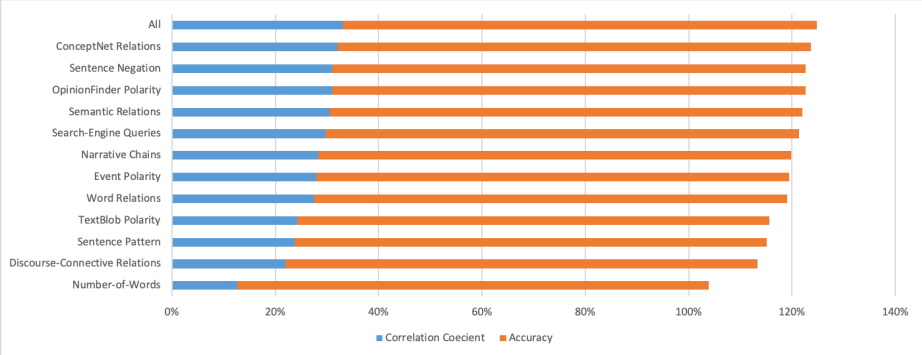

Fig. 9: Results of feature decrement experiments. We can see the performance of the model trained on all types of features except for the one shown in that row.

Here, we present the results that were obtained from an analysis of the features used to train our Random Forest model. Consequently, as shown in Fig. 9, 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 significantly 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-Pattern, the TextBlob-Polarity, and the Word-Relations are the most important features. This is in line with previous studies, where it was shown that features like the sentence length, sentence pattern and word relations play an important role on both the quality of the schemas and the tackle of the challenge [20, 21, 32].

Regarding the TextBlob-Polarity, our results show that it is better in capturing the polarity context than the other polarity features, which is strange as it is not commonly used in the literature. Regarding the OpinionFinder, previous works have stated, that this might happens because it was trained on a completely different training set [32].

Contrary to our expectations, and unlike what other studies have mentioned [32], Search-Engine-based features are not among the most useful 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 [7], 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 was able to determine if negation exists in only 41% of the Schema Halves.

8.2 Deep learning

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

8.2.1 Results & Discussion

System	MAE	Correl	Accuracy	Schema Halves
Wikisense-based	23	0.22	77	100
WinoReg-RF	8.36	0.33	91.64	100
WinoReg-DL	0.673	0.39	93.27	100

Table 1: Results of the Fixed Baseline, the *Wikisense*-based hardness, and *WinoReg* based on both the Random-Forest and the Deep-Learning Approach.

Our tests show that there is a positive correlation between WinoReg results and the perceived human hardness-indexes, across the Winograd schemas (see Table 1). Specifically, within the Deep-Learning approach, WinoReg can achieve an accuracy of 93.27% with a correlation coefficient of 39%.

Compared to Wikisense-based approach, WinoReg, within the Deep-Learning approach can achieve a higher correlation coefficient of 17%. Additionally, if we compare the two systems on the 57% of the schemas the Wikisense-based approach was able to solve, the correlation-coefficient difference rises to 10%, in favor of WinoReg (38% vs 48%).

As shown in Table 1, the Deep-Learning approach has an advantage over the Random-Forest approach. Our results highlighted that WinoReg results correlate better to human adult results in the case of Deep-Learning than the Random-Forest approach. Specifically, the Deep-Learning approach outperforms the Random-Forest approach by 2% in accuracy and 6% in correlation coefficient. Additionally, the fact that the deep learning

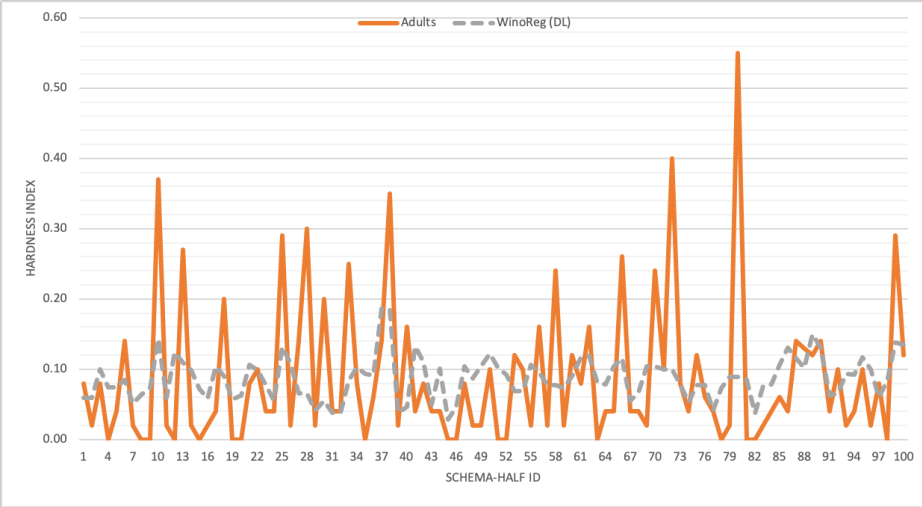

Fig. 10: Variability of the WinoReg hardness-index and the perceived human hardness-index across our testing set (100 schema halves). WinoReg is trained based on the Deep-Learning approach.

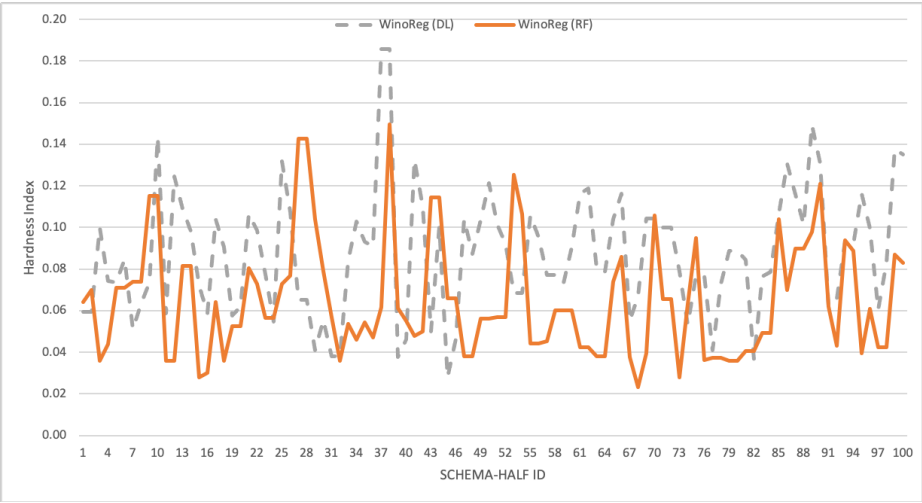

Fig. 11: Variability of WinoReg approaches, in relation to the perceived human hardness-index across our testing set (100 schema halves).

does not require feature engineering, offers it a compelling advantage over the Random-Forest approach.

The general picture emerging from the analysis is that the performance of WinoReg, when trained based on the Deep-Learning approach, *varies* across Winograd schemas in a way that resembles the variability of the human performance more closely than what other approaches could achieve (see Fig. 10). In this regard, certain WSC instances that are easier or harder for humans are respectively identified as such by *WinoReg*. Broadly speaking, both approaches have a compelling advantage over the Wikisense-based system. Specifically, further analysis we undertook showed that both WinoReg approaches correlate positively with a correlation coefficient of 22%, suggesting that schemas that are easier or harder for the Random-Forest approach are accordingly labeled as such by the Deep-Learning approach (see Figure 11).

Speed Analysis: Recall that having access to schema hardness indexes is crucial for both the quality of the schemas and the CAPTCHA service. In this regard, we performed a speed analysis to identify how fast the Deep-Learning approach can provide us with results. Our analysis revealed that within the Deep-Learning approach WinoReg can return results, on average, in 1.6 msec for any given schema, which means 60000 times faster the Random Forest and 18m times faster the Wikisense-based approach. This means that WinoReg, within the Deep-Learning approach, is able to return results in real-time with no further delays.

9 Related Work

The work presented here is not irrelevant with a recent work, where we have demonstrated the possibility of using crowdsourced workers for the development of Winograd schemas [20]. WinoFlexi is an online collaboration platform where workers collaborate with the help of various tools that enhance the schema development process. One of this tools is the Wikisense-based approach, which indirectly helps workers develop schemas of various hardness indexes, albeit with big delays. For instance, if the majority of the developed schemas of a crowdworker are considered easy, WinoFlexi prompts them to develop schemas that are harder to solve. Unfortunately, this might lead to high development costs because the WinoFlexi's notification mechanism depends on the Wikisense-based approach which is time consuming. In this regard, if we replace Wikisense-based approach with WinoReg, it will further reduce the schema development costs. Particularly, with WinoReg (DL), WinoFlexi will be able to access the hardness-indexes of schemas in real-time.

In another work, which is relevant to the previous one [21], we have designed a system that offers full pipeline for automated or semi-automated design of schemas. At the same time, it considerably help humans in the schema development task. Evidence from that study has shown that the developed system is able to automatically design large amounts of schemas, albeit of lower quality to that developed by humans [20]. On the other hand, it was shown that the system is able to considerably motivate and inspire humans for the development of high-quality schemas. In this regard, WinoReg could be used to help humans develop schemas of various hardness-indexes.

The first and only Winograd schema challenge was organized back in 2016, as a side event of IJCAI. According to the organizers, the design of schemas was found to be too troublesome and difficult to be handled at regular intervals for short periods of time [28]. Through an experiment the organizers evaluated the hardness of each examined schema (consisted of 89 problems with 9 subjects) which helped in the organization of the challenge. Their results have shown a 91% of achievement which is in line with ours and Bender's results. They categorized their schemas according to the number of the correct answers given by participants, which resembles the way WinoReg works. In this regard, instead of using human participants, they could use WinoReg mechanisms which could save them time and money —Although participants were paid for their participation, the authors did not mention the amount payed.

In a recent work we have demonstrated how Winograd schemas can form a novel form of CAPTCHAs [19]. Specifically, by providing motivation for a detailed form of WSC-based CAPTCHAs we have shown that this kind of CAPTCHAs are equally entertaining and useful like the other form of CAPTCHAs. The WSC-based CAPTCHAs are generated as the means to identify humans from bots, and at the same time to prevent automated processes from performing illicit actions. WinoReg can contribute by organizing schemas according to their perceived hardness for humans to be displayed accordingly by the CAPTCHA service. Specifically, a variety of various mechanisms can straighten the production of harder schemas to solve in the case of performed fraudulent actions.

10 Conclusion and Future Work

This paper has investigated the possibility of building a system that can output the perceived human hardness index of any Winograd schema in the shortest time possible. Our results have shown that this is possible via the training of a system that is based on two different approaches, namely, the Random-Forest and the Deep-Learning approach. We have provided evidence of that by comparing WinoReg results with two studies, one from the literature [2] and one that we designed and undertook. Results have shown that WinoReg 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 33% correlation coefficient, whereas with the Deep-Learning approach 93.27% of accuracy with 39% correlation coefficient. Even though the results of two approaches seems close, the strong benefit of the Deep-Learning approach lies in the response time of the model, which is 60000 times faster than the Random-Forest model.

WinoReg can be used by researchers or challenge organizers to group schemas in terms of their perceived human hardness indexes. Specifically, WinoReg can be used by CAPTCHA organizers to ensure that the generated schemas are not overly demanding for human users. Additionally, WinoReg can be used in systems that pursue the development of Winograd schemas from scratch, like in [20, 21], to ensure that a variety of schemas would be developed. We suggest that future studies should examine the impact of systems like WinoReg in other AI fields. For instance, in the field of machine translation, systems like WinoReg could be used to identify sentences that are harder to translate, in order to acquire better feedback from people. In this regard, WinoReg can

help with the problem many translation services face, of where to focus their attention to make end-users aware of the quality [38].

References

1. Baker, C.F., Fillmore, C.J., Lowe, J.B.: The Berkeley Framenet Project. In: Proceedings of the 17th international conference on Computational linguistics-Volume 1. pp. 86–90. Association for Computational Linguistics (1998)
2. Bender, D.: Establishing a Human Baseline for the Winograd Schema Challenge. In: MAICS. pp. 39–45 (2015)
3. Bengio, Y., Ducharme, R., Vincent, P., Jauvin, C.: A Neural Probabilistic Language Model. Journal of machine learning research **3**(Feb), 1137–1155 (2003)
4. Bengio, Y., Goodfellow, I., Courville, A.: Deep learning, vol. 1. MIT press (2017)
5. Blanco, E., Moldovan, D.: Some Issues on Detecting Negation From Text. In: Twenty-Fourth International FLAIRS Conference (2011)
6. Breiman, L.: Random forests. Machine learning **45**(1), 5–32 (2001)
7. Budukh, T.U.: An Intelligent Co-reference Resolver for Winograd Schema Sentences Containing Resolved Semantic Entities. Master's thesis, Arizona State University (2013)
8. Chambers, N., Jurafsky, D.: Unsupervised Learning of Narrative Event Chains. In: ACL. vol. 94305, pp. 789–797. Citeseer (2008)
9. Christoforaki, M., Ipeirotis, P.: Step: A Scalable Testing and Evaluation Platform. In: Proceedings of the 2nd AAAI Conference on Human Computation and Crowdsourcing (2014)
10. Dagan, I., Glickman, O., Magnini, B.: The Pascal Recognising Textual Entailment Challenge. In: Machine Learning Challenges Workshop. pp. 177–190. Springer (2005)
11. François, C.: Deep Learning with Python (2017)
12. Fry, H.: Hello World: How to be Human in the Age of the Machine. Random House (2018)
13. Gary Marcus: Beyond Deep Learning with Gary Marcus. [online] (2019), https://hbr.org/podcast/2019/10/beyond-deep-learning-with-gary-marcus
14. Hassan, A., Radev, D.: Identifying Text Polarity Using Random Walks. In: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics. pp. 395–403. Association for Computational Linguistics (2010)
15. Hirth, M., Hoßfeld, T., Tran-Gia, P.: 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. pp. 322–329. IEEE (2011)
16. Hochreiter, S., Schmidhuber, J.: Long Short-Term Memory. Neural computation **9**(8), 1735–1780 (1997)
17. 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
18. 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
19. 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

20. Isaak, N., Michael, L.: WinoFlexi: A Crowdsourcing Platform for the Development of Winograd Schemas. In: Liu, J., Bailey, J. (eds.) AI 2019: Advances in Artificial Intelligence. pp. 289–302. Springer International Publishing, Cham (2019)
21. Isaak., N., Michael., L.: 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,. pp. 26–35. INSTICC, SciTePress (2020). https://doi.org/10.5220/0008902600260035
22. Kocijan, V., Cretu, A.M., Camburu, O.M., Yordanov, Y., Lukasiewicz, T.: A Surprisingly Robust Trick for Winograd Schema Challenge. arXiv preprint arXiv:1905.06290 (2019)
23. LeCun, Y., Bengio, Y., Hinton, G.: Deep Learning. Nature **521**(7553), 436–444 (2015)
24. Levesque, H., Davis, E., Morgenstern, L.: The Winograd Schema Challenge. In: Thirteenth International Conference on the Principles of Knowledge Representation and Reasoning (2012)
25. Levesque, H.J.: On Our Best Behaviour. Artificial Intelligence **212**, 27–35 (2014)
26. Liu, H., Singh, P.: ConceptNet — A Practical Commonsense Reasoning Tool-Kit. BT technology journal **22**(4), 211–226 (2004)
27. Liu, Q., Jiang, H., Evdokimov, A., Ling, Z.H., Zhu, X., Wei, S., Hu, Y.: Probabilistic Reasoning via Deep Learning: Neural Association Models. arXiv preprint arXiv:1603.07704 (2016)
28. Morgenstern, L., Davis, E., Ortiz, C.L.: Planning, Executing, and Evaluating the Winograd Schema Challenge. AI Magazine **37**(1), 50–54 (2016)
29. Peer, E., Samat, S., Brandimarte, L., Acquisti, A.: In: Diehl, K., Carolyn Yoon, D. (eds.) Beyond the Turk: An Empirical Comparison of Alternative Platforms for Crowdsourcing Online Research. NA - Advances in Consumer Research, vol. 43, pp. 18–22. MN : Association for Consumer Research (2015)
30. Peng, H., Khashabi, D., Roth, D.: Solving Hard Coreference Problems. In: Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. pp. 809–819 (2015)
31. Probst, P., Wright, M.N., Boulesteix, A.L.: Hyperparameters and Tuning Strategies for Random Forest. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery **9**(3), e1301 (2019)
32. Rahman, A., Ng, V.: 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. pp. 777–789. EMNLP-CoNLL '12, Association for Computational Linguistics, Stroudsburg, PA, USA (2012), http://dl.acm.org/citation.cfm?id=2390948.2391032
33. Sap, M., Le Bras, R., Allaway, E., Bhagavatula, C., Lourie, N., Rashkin, H., Roof, B., Smith, N.A., Choi, Y.: ATOMIC: An Atlas of Machine Commonsense for If-Then Reasoning. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 33, pp. 3027–3035 (2019)
34. Schmidhuber, J.: Deep Learning in Neural Networks: An Overview. Neural networks **61**, 85–117 (2015)
35. Sharma, A., Vo, N.H., Aditya, S., Baral, C.: 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. pp. 25–31 (2015)
36. Singh, P., Lin, T., Mueller, E.T., Lim, G., Perkins, T., Zhu, W.L.: Open Mind Common Sense: Knowledge Acquisition From the General Public. In: OTM Confederated International Conferences" On the Move to Meaningful Internet Systems". pp. 1223–1237. Springer (2002)
37. Socher, R., Bengio, Y., Manning, C.D.: Deep learning for NLP (without magic). In: Tutorial Abstracts of ACL 2012. pp. 5–5. Association for Computational Linguistics (2012)

38. Specia, L., Turchi, M., Cancedda, N., Dymetman, M., Cristianini, N.: Estimating the Sentence-Level Quality of Machine Translation Systems. In: 13th Conference of the European Association for Machine Translation. pp. 28–37 (2009)
39. Speer, R., Chin, J., Havasi, C.: Conceptnet 5.5: An open multilingual graph of general knowledge. In: Thirty-First AAAI Conference on Artificial Intelligence (2017)
40. Sundermeyer, M., Ney, H., Schlüter, R.: From Feedforward to Recurrent Lstm Neural Networks for Language Modeling. IEEE/ACM Transactions on Audio, Speech, and Language Processing **23**(3), 517–529 (2015)
41. Sundermeyer, M., Schlüter, R., Ney, H.: Lstm Neural Networks for Language Modeling. In: Thirteenth annual conference of the international speech communication association (2012)
42. Suresh, M., Taib, R., Zhao, Y., Jin, W.: Sharpening the BLADE: Missing Data Imputation Using Supervised Machine Learning. In: Liu, J., Bailey, J. (eds.) AI 2019: Advances in Artificial Intelligence. pp. 215–227. Springer International Publishing, Cham (2019)
43. Valiant, L.G.: Knowledge Infusion. In: Proceedings of the 21st National Conference on Artificial Intelligence - Volume 2. pp. 1546–1551. AAAI'06, AAAI Press (2006), http://dl.acm.org/citation.cfm?id=1597348.1597438
44. Wilson, T., Hoffmann, P., Somasundaran, S., Kessler, J., Wiebe, J., Choi, Y., Cardie, C., Riloff, E., Patwardhan, S.: Opinionfinder: A System for Subjectivity Analysis. In: Proceedings of HLT/EMNLP 2005 Interactive Demonstrations. pp. 34–35 (2005)
45. Wilson, T., Wiebe, J., Hoffmann, P.: 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)