Winventor: A Machine-driven Approach for the Development of Winograd Schemas

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

Keywords: Winograd Schema Challenge, Pronoun Resolution, Natural Language Processing.

Abstract: The Winograd Schema Challenge — the task of resolving pronouns in certain carefully-constructed sentences — has recently been proposed as a basis for a novel form of CAPTCHAs. Such uses of the task necessitate the availability of a large, and presumably continuously-replenished, collection of available Winograd Schemas, which goes beyond what human experts can reasonably develop by themselves. Towards tackling this issue, we introduce *Winventor*, the first, to our knowledge, system that attempts to fully automate the development of Winograd Schemas, or at least to considerably help humans in this development task. Beyond describing the system, the paper presents a series of three studies that demonstrate, respectively, *Winventor*'s ability to replicate existing Winograd Schemas from the literature, automatically develop reasonable Winograd Schemas from scratch, and aid humans in developing Winograd Schemas by post-processing the system's suggestions.

1 INTRODUCTION

A number of challenges have been proposed in the literature towards encouraging the development of systems that will automate or enhance basic human abilities and increase the extent to which humans can relate and interact with them. Among those, the Winograd Schema Challenge (WSC) (Levesque et al., 2012) is concerned with the ability to resolve a definite pronoun to one of its possible co-referents in certain carefully-constructed sentences, which in certain cases could be argued to require the use of commonsense knowledge. Although humans can easily solve this type of task, the performance of automated approaches is still significantly lacking (Sharma et al., 2015).

This disparity in the performance of humans and machines on the WSC suggests, on the one hand, the need for more research in identifying ways to endow machines with the necessary knowledge (commonsense or otherwise (Michael, 2013)) that would allow machines to perform better. The importance of this suggestion becomes even more clear when one considers that the general problem of anaphora resolution remains a significant task for many natural language understanding applications (Deepa and Deisy, 2017). On the other hand, this disparity presents an opportunity for the use of WSC as the basis for developing CAPTCHAs (automated tests that aim to distinguish humans from computers), as proposed in recent work (Isaak and Michael, 2018).

In either case above, a large collection of available Winograd Schemas would seem to be a prerequisite, or at least a facilitator, for further work and progress. This collection could be used in the former case, for example, to develop data-driven approaches to the knowledge acquisition task. In the latter case of developing CAPTCHAs, one could argue that even the availability a large collection is insufficient, and one could further expect that this collection would be replenished in an ongoing manner, to prevent machines that simply memorize the answers to previously seen Winograd Schemas to pass the test.

Motivated by the difficulty of having human experts develop Winograd Schemas from scratch, this work introduces *Winventor* (see Figure 1), a system that can facilitate the continuous development of Winograd Schemas, either as a fully-automated process, or as a tool to be used by human experts. In the sections that follow, we first present some aspects of the architecture of *Winventor*, and we then proceed to present the results from three studies that we undertook to evaluate the system's performance on replicating existing Winograd Schemas from a well-known

[a] https://orcid.org/0000-0003-2353-2192

WSC dataset (Rahman and Ng, 2012), on developing new Winograd Schemas, and on helping humans develop new Winograd Schemas. To the best of our knowledge, this is the first published work to report results on the feasibility of this approach.

2 BACKGROUND ON THE WSC

The Winograd Schema Challenge (WSC) was proposed as an alternative to the Turing Test. Rather than basing the test on the sort of short free-form conversation suggested by the Turing Test, a machine claiming to pass the WSC should be able to demonstrate that it is thinking without having to pretend to be somebody (Levesque et al., 2012; Levesque, 2014). During the first, and only one so far, competition that was held for the WSC (Morgenstern et al., 2016), the challenge was found to be extremely difficult for machines (Isaak and Michael, 2016), since machines that perform well would presumably need to be capable of supporting a wide range of commonsense reasoning that would span many AI application domains. According to Levesque (2014), the AI community need to return to their roots in Knowledge Representation and Reasoning for language and from language because when humans are at their *best behaviour* they undertake knowledge-intensive activities such as responding to WSC questions.

Winograd Schemas comprise two Winograd Halves, with each Winograd Half consisting of a sentence, a definite pronoun (or a question), two possible pronoun targets, and the correct pronoun target. The task is to identify the correct pronoun target given the rest of the information in a Winograd Half. The challenging nature of the task derives from several constraints that each schema obeys: the two pronoun targets belong to the same gender, and both are either singular or plural; additionally, the difference between a pair of halves is a special word or a small phrase, which when replaced by another word causes the correct answer to flip from one pronoun target to the other. Pairs of halves that do not obey all the constraints are known as "Winograd Schemas in the broad sense" (Levesque et al., 2012).

The following WSC schema illustrates the nature of the challenge: *i) sentence: "The cat caught the mouse because it was clever." / Pronoun: "it" / Pronoun-Targets: "cat, mouse" / Special-Word: "clever" ii) sentence: "The cat caught the mouse because it was careless." / Pronoun: "it" / Pronoun-Targets: "cat, mouse" / Special-Word: "careless"*. In some cases, one could consider a Winograd Half without identifying a special word, and without making provisions for the existence of the other half, and this task has been referred to as a PDP, or a pronoun disambiguation problem (Morgenstern et al., 2016).

Although the AI community has sought to promote the WSC through specialized competitions, constructing a WSC corpus is a laborious job, requiring creativity, motivation, and inspiration (Morgenstern et al., 2016). Perhaps unsurprisingly, then, there seem to exist only two WSC datasets that have been widely used: Rahman & Ng's dataset (Rahman and Ng, 2012) consisting of 942 schemas, and Levesque et al.'s dataset (Levesque et al., 2012) consisting of 150 schemas. It would seem that the *automated development* of schemas would help overcome the limitations of the manual construction of schemas, supporting regularly-held competitions and promoting more research on the WSC.

3 SYSTEM ARCHITECTURE

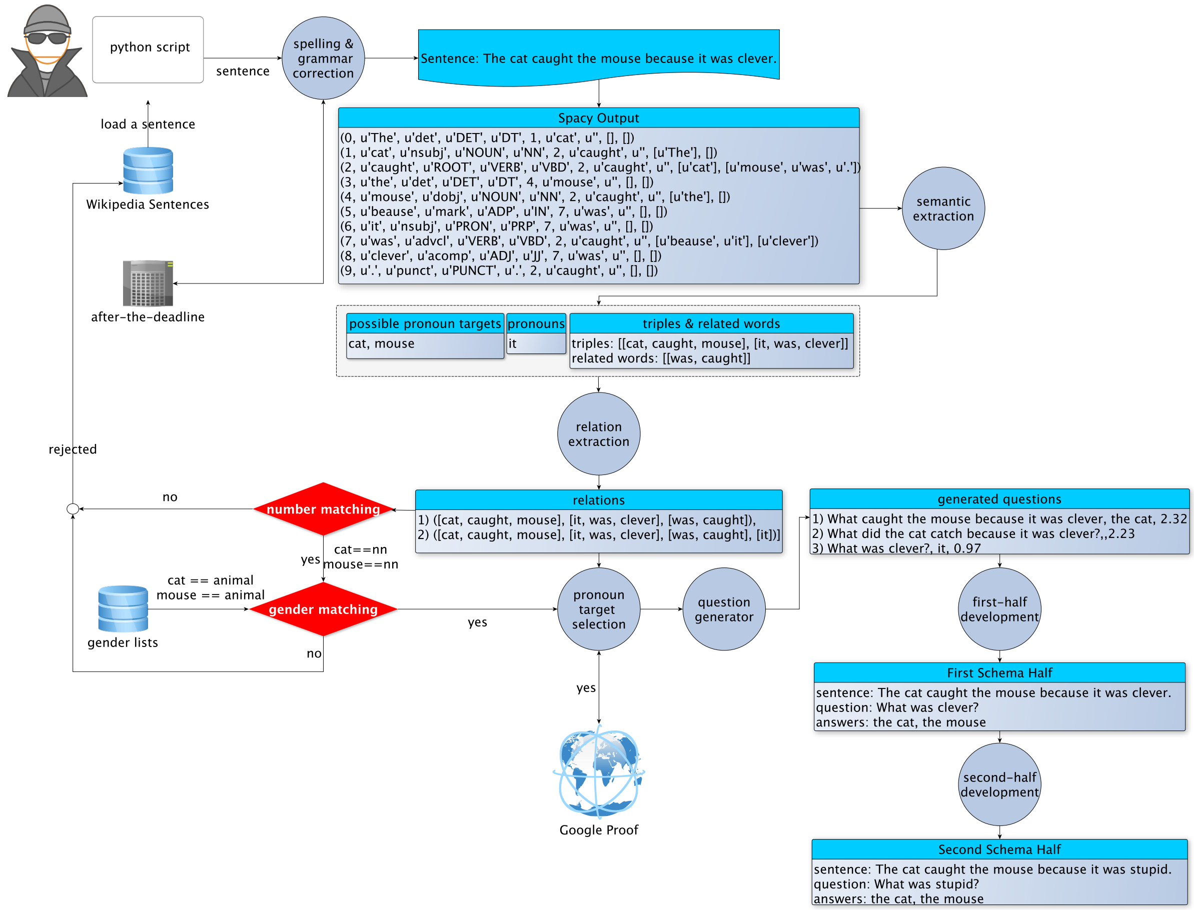
In this section, we briefly discuss the main elements of our approach by presenting how the engine works, and how it handles its semantics to develop schemas (see Figure 1).

3.1 Definitions and Conventions

We follow the constrained definition of the WSC which requires that the ambiguous word is a pronoun and that the two referents are noun phrases that occur in the sentence. If *Winventor* cannot develop a schema, it only develops a schema half (PDP). For each half *Winventor* has to develop the sentence, the definite pronoun, the question that indirectly refers to the definite pronoun, and the two pronoun targets (the correct pronoun target has to be selected by humans). To build *Winventor*, we created a framework that allows access to a broad collection of nearly 88 Millions of sentences from the English Wikipedia (Isaak and Michael, 2016). The system runs on Wikipedia sentences non-stop, and outputs its schemas in an online database.

3.2 Spelling & Grammar Correction

Since our sentences come from the English Wikipedia, it is possible that we might find sentences with spelling errors; abbreviations and misspellings of words are common examples of the informal nature of some Wikipedia texts. For this reason, *Winventor* parses each examined sentence through a double check. At first, it uses the Google language detection library to check the language of



Sentence: The cat caught the mouse because it was clever.

Spacy Output

```
(0, u'The', u'det', u'DET', u'DT', 1, u'cat', u'', [], [])
(1, u'cat', u'nsubj', u'NOUN', u'NN', 2, u'caught', u'', [u'The'], [])
(2, u'caught', u'ROOT', u'VERB', u'VBD', 2, u'caught', u'', [u'cat'], [u'mouse', u'was', u'.'])
(3, u'the', u'det', u'DET', u'DT', 4, u'mouse', u'', [], [])
(4, u'mouse', u'dobj', u'NOUN', u'NN', 2, u'caught', u'', [u'the'], [])
(5, u'beause', u'mark', u'ADP', u'IN', 7, u'was', u'', [], [])
(6, u'it', u'nsubj', u'PRON', u'PRP', 7, u'was', u'', [], [])
(7, u'was', u'advcl', u'VERB', u'VBD', 2, u'caught', u'', [u'beause', u'it'], [u'clever'])
(8, u'clever', u'acomp', u'ADJ', u'JJ', 7, u'was', u'', [], [])
(9, u'.', u'punct', u'PUNCT', u'.', 2, u'caught', u'', [], [])
```

possible pronoun targets	pronouns	triples & related words
cat, mouse	it	triples: [[cat, caught, mouse], [it, was, clever]] related words: [[was, caught]]

relations

1) ([cat, caught, mouse], [it, was, clever], [was, caught]),
2) ([cat, caught, mouse], [it, was, clever], [was, caught], [it])]

generated questions

1) What caught the mouse because it was clever, the cat, 2.32
2) What did the cat catch because it was clever?,,2.23
3) What was clever?, it, 0.97

First Schema Half

sentence: The cat caught the mouse because it was clever.
question: What was clever?
answers: the cat, the mouse

Second Schema Half

sentence: The cat caught the mouse because it was stupid.
question: What was stupid?
answers: the cat, the mouse

Figure 1: A Schema Development from *Winventor* for the Sentence *The cat caught the mouse because it was clever*.

the text. Next, it uses language-checker tools like *after-the-deadline*[1] to correct grammar and spelling errors.

3.3 Semantic Relations

According to Sharma et al. (2015), a semantic representation of text is considered good if it can express the structure of the text, can differentiate between the events and their context in the text, and uses a set of relations between the events and their participants. In anaphora resolution, among others, grammatical role, number, gender and syntactic structure play a role (Amsili and Seminck, 2017). *Winventor* utilizes a dependency parser (spaCy) to turn raw text into semantic relations. We use spaCy[2] to parse the sentence, to locate the candidate pronoun targets, and to develop triples and related words (see *spaCy-Output* in Figure 1). The triples are semantic scenes that are created through the sentence's subjects and objects (Isaak and Michael, 2016). For instance, in our example sentence (see *triples* in Figure 1) the triple [cat, catch, mouse] is created by the *nsubj & dobj* relations *(abbreviations of "nominal subject" and "direct object")*. Also, prior the schema building, it is important to search for *pronoun-noun* relations & *pronoun-proper-noun* relations; these are the relations where nouns are related to other nouns via pronouns and/or other words (see *relations* in Figure 1). If at least one pronoun exists, and two nouns or two proper nouns exist (possible pronoun targets), we proceed to the next step, otherwise we proceed to the next sentence. To illustrate, for the sentence *"The cat caught the mouse because it was clever"*, *Winventor* will return *two relations*; these relations show that the nouns *cat, mouse* are related to the pronoun *it*, the verb *caught*, and the adjective *clever* (see *relations* in Figure 1).

[1] https://afterthedeadline.com
[2] https://spacy.io

3.4 Selecting the Pronoun Targets

Each sentence's pronoun targets will be either two proper-nouns or two nouns. Also, the two pronoun targets have to belong to the same gender, and both have to be either singular or plural.

3.4.1 Gender & Number Factor

For the *proper-nouns* we are using the spaCy's entity recognition system. SpaCy's default model identifies a variety of named and numeric entities, including locations, organizations, products and persons. Via spaCy, *Winventor* locates and classifies named entities into predefined categories; in case of persons, each pair has to consist of two males or two females. On the other hand, we know that gender represents an axis along which a word class that of nouns can be classified into different types. While grammatical gender was an essential category in Old English, Modern English is mostly based on natural gender that is reflected mainly in pronouns (Huddleston and Pullum, 2005). It appears that it lacks a system of gender consensus within the noun phrase where the choice of pronoun is determined based on semantics rather than on any formal assignments.

Recognized that there is not always a gender agreement in English between nouns and their modifiers, we developed *Winventor* to work in two different modes (austere/not). If the *austere mode* is disabled, *Winventor* can develop schemas that its pronoun targets may not agree e.g., in gender, although it keeps a track of the *genderAgreement, pronounGenderAgreement and numberAgreement* flags. Alternatively, the pronoun targets have to strictly follow the WSC rules. For the *numberAgreement* we check if the two nouns *agree* in number through the dependency parser. Next, for the *genderAgreement* we search some pre-downloaded gender lists to see if the two nouns have the same gender; these are lists that were downloaded from several online-sources where nouns are classified according to their gender (e.g., masculine, feminine, neutral). If we are unable to locate the two nouns in our lists, we can either search our lists with the synonym or the similarity factor. For the *pronounGenderAgreement* we consider the following: The third-person singular personal pronouns are gender specific where *he/him/his* refer to the masculine gender, and *she/her(s)* to the feminine gender. Furthermore, the singular pronouns *they/them/their(s)* refer to the neutral gender (e.g., people, animals). Additionally, the pronouns *it/its* refer to the neuter gender (e.g., objects, animals); in the case of companion animals, the gendered pronouns *he/she* may be used, as they would be for a human.

3.4.2 Selecting Pronoun Targets

For each schema, *Winventor* has to select the best pair of the pronoun targets. Therefore, for every pair of nouns or proper-nouns, it stores the boolean values of *genderAgreement, pronounGenderAgreement and numberAgreement*. Additionally, it stores if the pronoun targets participate in triple relations. Finally, we use an aggregation mechanism to give to each target a score (Mitkov, 1998). According to Mitkov (1998), noun preferences play a definitive role in hunting down the pronoun targets from a set of possible candidates when we have limited background knowledge. From his work, we used five indicators that are related to salience: 1) *Definiteness*: where noun phrases are regarded as definite if the head noun is modified by a definite article. 2) *Indicating Verbs*: if a verb is a member of a specific *Verb_set (e.g., discuss)*, we consider the first NP following it as the preferred antecedent. 3) *Lexical Reiteration*: repeated synonymous noun phrases are likely candidates for the antecedent. 4) *nonPrepositional*: a pure, non-prepositional noun phrase is given a higher preference. 5) *Collocation & ImmediateReference*: This preference is given to candidates which have an identical collocation pattern with a pronoun. According to Mitkov, in a sentence with multiple pronoun targets (candidates) the pair with the highest aggregate score can be rated as the best pair.

3.5 Question Generator

According to Levesque et al. (2012), the most problematic aspect of the WSC is to come up with a list of appropriate questions. In the field of Automatic Question Generation, most systems focus on the text-to-question task, where a set of content-related questions are generated based on a given text. For the question development, we use the Heilman and Smith (2009) system, which generates factual questions with an overgenerating and ranking strategy, such as Name Entity Recognizer and Wh-movement Rules. It is a tool that is freely available, and it generates *weighted* questions based on a given text. *Winventor* uses the question generator with the *"–keep-pro & –just-wh"* flags enabled. *Keep-pro* keeps questions with unresolved pronouns and *Just-wh* excludes boolean questions from the output (questions that can be answered with a simple yes/no). In case of generated questions, *Winventor* leaves only the questions that have as answers unresolved pronouns. In the end, it selects the best pronoun target pair that relate to the pronoun which is the answer of the most significant question (the question with the *bigger-weight*). If more pairs can be selected, then it develops more schemas/halves.

3.6 Schema Development

Winograd schemas are constructed so that there is a special word that can be substituted, causing the other candidate pronoun referent to be correct (Morgenstern et al., 2016). The existence of the special word is one way to ensure that test creators do not unwittingly construct a set of problems in which ordering of words can be used to help the disambiguation process. According to Levesque with a bit of care on the selection of the special word it is possible to come up with Winograd schema questions that exercise different kinds of expertise (Levesque, 2014).

To find the special word we use a simple heuristic approach which is similar to the work of Budukh (2013). Specifically, we parse the question of the first *half* to find the verb that participates in the triple relation (e.g., the word *fear* in the triple *feared (who, violence)*, of the question "Who feared violence?"). If we have an auxiliary verb, we take the rightmost word of the triple, which is the dobj (e.g., the word *clever* in the triple *was (it, clever)*, of the question "Who is the clever?"). Next, we find the best antonym of the returned word via the *similarity* factor, and modify it to match the tense of the first half's question. To end up with the schema, we modify the first half's sentence and question to develop the second half.

3.7 Schema Categorization

Via the schema categorization, *Winventor* classifies the developed schemas into predefined categories, regarding the pronoun targets (e.g., names of organizations, locations). It also gathers information about the *schema subject*, the *gender*, the *number* of the pronoun targets, and the *Mitkov* scores. With this categorization challenge organizers can get new ideas, for the development of new schemas.

4 EVALUATION

In this section, we present the results that were obtained by applying the methodology described in this paper. We describe the results from three studies that we undertook to evaluate the system's performance on replicating existing Winograd Schemas from a well-known WSC dataset (Rahman and Ng, 2012), on developing new Winograd Schemas, and on helping humans develop new Winograd Schemas.

4.1 Automated Evaluation

In the first experiment, we tested Winventor on a well-known WSC dataset (Rahman and Ng, 2012); this is a challenging dataset of 943 schemas where each half consists of a sentence, a definite pronoun, and two pronoun targets. Our goal is to feed Winventor with the sentences of the first half of each schema, to evaluate if it can produce the same, or similar results. Also, for the purposes of this experiment the *austere* mode is disabled.

4.1.1 Results and Discussion

According to our results *Winventor* was able to develop 990 halves, where 416 were based on unique sentences; 848 of the 990 were schemas. At the same time, it rejected 527 sentences, More than two hundred halves (254/416 halves –61%), of which 214 are schemas, match with schemas/halves of the WSC dataset meaning that they have the same definite pronoun and the same pronoun targets. Thirty-three percent of the returned halves that were identified as proper-noun problems (122 halves), have more than two proper nouns in each sentence (we can say that, on average, each sentence contains three proper-nouns). Also, 70% of the halves that were identified as noun problems (132 halves), have more than two nouns in each sentence (on average, each sentence contains four nouns). The results show that it is more complicated to find the correct target pronouns in the case of proper-nouns than in the case of nouns. This is in line with other works which consider the resolving of proper nouns more difficult and more challenging (Isaak and Michael, 2016; Budukh, 2013). On the other hand, in the halves where Winventor was able to identify the correct pronoun but not the correct pronoun targets, we found the following: 95% of the noun problems have more than two nouns in each sentence (665 sentences), and all of the proper-noun problems (33 sentences) have more than two proper-nouns. On average, each sentence that was identified as a proper-noun problem contains four proper-nouns, and each sentence that was identified as a noun problem contains five nouns; the increased number of nouns and proper-nouns might be one of the reasons why Winventor failed to identify the correct pronoun targets. Another reason is the average sentence length in words; here, the average sentence length is thirteen words for the proper-noun problems and nineteen words for the noun problems. In the correctly identified halves, the average length is twelve words for the proper-noun problems, and fourteen words for the noun problems.

The number of the produced schemas does not mean that the developed halves that did not end up being schemas are not useful. We have to keep in mind that generating of schemas is difficult and troublesome even for humans. Furthermore, although the original dataset did not include questions, *Winventor* produced schemas with valid questions (e.g., see the first two schemas in Table 1). Additionally, this library was developed under the broad sense which means that schemas do not differ only in one word, hence it is difficult to match them (e.g., see the last two schemas in Table 1 that differ only in one word, but have the same meaning). Regarding the big number of rejected sentences, it shows that further gains could be achieved via more accurate semantic analysis of each sentence. For instance, 53% of 943 sentences were rejected because *Winventor* was not able to develop relations between pronouns and pronoun-targets.

We further analyzed the relationship between developed halves and different factors (e.g., gender, triple, Mitkov-score). The main purpose was to find the impact of these factors on the selection process of the halves that were developed with the correct pair of pronoun targets (e.g., 254 halves from a database of 990 halves). The results showed interesting findings: If for each half we select the pronoun targets that agree in gender, number, have a pronoun-gender agreement, and they participate in triples, we have an 89% success rate. On the other hand, if we remove the triple factor the success rate drops to 85%; this might show the important role of the semantic scenes (triple factor) for the tackle of the challenge. Regarding the Mitkov-score, if we select the pronoun targets that have bigger Mitkov-score we have an 82% success rate; it seems that the results are in line with Mitkov's work, meaning that its theory works when we have limited background knowledge.

4.2 Evaluation by Humans

In this section we investigate whether this a priori appropriateness of this approach as a tool to develop new Winograd schemas can be justified in terms of its qualification by humans. For these experiments, we used *freshly* developed schemas from *Winventor*. The schemas were developed with the *austere* flag enabled, meaning that answers that did not match in gender, number, and pronoun-gender agreement were rejected. At the time of the experiment *Winventor* had already searched 20000 sentences and developed 500 schemas.

4.2.1 Platform & Participants

The Microworkers (MW) platform[3] was launched in May, 2009 and can be considered as a viable platform for across a range of experimental tasks. From an employers perspective the MW platform offers features which can influence the completion time of a campaign. MW uses gold question automatically to track the performance of workers and rate their answers with a trust score.

A single experiment was performed online –in May 2019– using MW, where one hundred people participated. Participants were adult residents of the United States, United Kingdom, Canada, New Zealand who speak English fluently (aged between 18 and 65). Participants were also screened by means of a qualification task from the Microworkers platform, for their fluency. Out of 100 participants everyone managed to attempt and finish the task. The participants were paid $2.50 each.

4.2.2 Training Task

To maximize the quality of our results, we gave all participants a training task. This preliminary task familiarized them with the experiment by asking them to answer a few Winograd schemas/halves. Participants were required to correctly answer all the questions before proceeding to the questionnaire. Immediate feedback (correct or incorrect) was given after each trial. All participants had to pass the same number of training examples, to continue on to the testing task; none was discarded.

4.2.3 Design & Procedure

The participation was anonymous, and it lasted about 16 minutes. Although we could not monitor the process we asked them to take their time and answer honestly. Participants were given instructions explaining the task and directing them to answer each question without sacrificing accuracy. They also read an informed consent form and agreed to participate in the study. Also, they were told that once they answered a schema they could not go back to change their answer. We divided our questionnaire into two sections: For the first section, we randomly selected twenty Winograd *halves* from our database, and for the second ten Winograd *schemas* (schema *halves* that were selected in the first section were not included in the second one). The questionnaire started with the *half* section and continued with the *schema* section. Each half/schema was displayed on a single

[3] www.microworkers.com

Table 1: A snapshot of *Winventor's* Developed Schemas on the Rahman & Ng Dataset. The first two match 100% with the Dataset while the last two have slight differences.

System		Sentence	Pronoun	Question	Answers
1	Rahman & Ng	Caroline quickly defeated Sue because she regained her confidence. Caroline quickly defeated Sue because she lost her confidence.	she	-	Caroline, Sue
1	Winventor	Caroline quickly defeated Sue because she regained her confidence. Caroline quickly defeated Sue because she lost her confidence	she	Who regained her confidence? Who lost her confidence?	Caroline, Sue
2	Rahman & Ng	Saudi Arabians do not respect the Bangladeshis because they are poor. Saudi Arabians do not respect the Bangladeshis because they are rich.	they	-	Saudi Arabians, Bangladeshis
2	Winventor	Saudi Arabians do not respect the Bangladeshis because they are poor. Saudi Arabians do not respect the Bangladeshis because they are rich	they	Who are rich? Who are poor?	Saudi Arabians, Bangladeshis
1	Rahman & Ng	The cat caught the mouse because it was clever. The cat caught the mouse because it was careless.	it	-	cat, mouse
1	Winventor	The cat caught the mouse because it was clever. The cat caught the mouse because it was stupid.	it	Who is the clever? Who is the stupid?	cat, mouse
2	Rahman & Ng	Apple defeated Microsoft in the war because they were creative. Apple defeated Microsoft in the war because they lack creativity.	they	-	Apple, Microsoft
2	Winventor	Apple defeated Microsoft in the war because they were creative. Apple defeated Microsoft in the war because they were untalent	they	Who were creative? Who were untalented?	Apple, Microsoft

screen, followed by the question where three choices were displayed side-by-side i) Valid Schema - Easy to Solve ii) Valid Schema - Hard to Solve iii) Non-Valid Schema.

4.2.4 Results and Discussion

Based on the results, the participants characterized the Winograd halves as *valid* with a mean of 69% (σ = 0.15) . Moreover they marked the Winograd schemas as *valid* with a mean of 73% (σ = 0.17) . In the first section, the Winograd halves were characterized as *valid - easy to solve* by 46% of the participants and as *valid - hard to solve* by 23% of the participants. In the second section, the Winograd schemas were characterized as *valid - easy to solve* by 55% of the participants and as *valid - hard to solve* by 18% of the participants. The positive difference in favor of the schemas might have happened because the participants were able to see the two halves at the same time and not because of the quality of the schemas, which are harder to develop (it seems that it helped them to understand the meaning of the schema halves). According to Levesque et al. (2012), it is clear enough that to build quality Winograd schemas we need to have the following: i) The first pitfall concerns questions whose answers are in a certain sense too obvious. ii) The second and more troubling pitfall concerns questions whose answers are not obvious enough. It seems that our developed schemas are in line with the WSC rules, meaning that they cannot be considered as easy or hard schemas (see *a) Examples of Valid Schema-Halves* in Table 2). On the other hand, we are not claiming that this system can be used to develop schema/halves without the need of reviewing (see *b) Examples of Valid Schema-Halves (after some minor modifications)* in Table 2); the fact that sentences are taken from Wikipedia pages for the schema development process might yield pronominal coreferences that appear unbound (Antunes et al., 2018).

4.3 Winventor as a Co-worker

To get feedback regarding Winventor's usefulness as a co-worker, we asked ten colleagues that have participated in earlier WSC-related experiments of ours, and have prior experience in developing Winograd Schemas. The participants were asked to develop halves/PDPs with and without Winventor's help (see Table 3). For this experiment we gave them an assignment which consists of two sections. In the first section we asked them to develop as many PDPs as they can in 10 minutes without any help from Winventor (called non-guided PDPs). In the second section we gave them access to 15 randomly selected Winventor PDPs and asked them to develop as many PDPs as they can in the same time (called guided PDPs).

4.3.1 Results and Discussion

Based on the results, Winventor helped the participants develop 20 PDPs in 10 minutes on average. Without Winventor they had developed an average of 7 PDPs. Based on the remarks submitted by the participants in our study, Winventor helped them develop PDPs that are based on different sentence patterns/types. To analyze the guided PDPs and compare them to non-guided PDPs we used a tool that we designed in another work (Isaak and Michael, 2019) which is based on the English Grammar (Seely, 2013). This is a tool that refers to the sentence-pattern of each designed PDP. It can take as input any English sentence and output its pattern/type

Table 2: A snapshot of *Winventor's* Developed Halves (PDPs) on the Wikipedia Sentences. The first six examples show valid halves that were developed automatically by *Winventor*, without human reviewing. The last two are examples that required some reviewing.

	Sentence	Pronoun	Question	Answers
	a) Examples of Valid Schema-Halves			
1	Your governors are unjustifiably killing people and they only write the crime of the killed person to inform you.	they	Who only write the crime of the killed person to inform you?	The governors, The people
2	The Greeks hiding inside the Trojan Horse were relieved that the Trojans had stopped Cassandra from destroying it, but they were surprised by how well she had known of their plan to defeat Troy.	they	Who were surprised by how well she had known of their plan to defeat Troy?	Greeks, Trojans
3	Captain Cardenas subsequently told reporters that the cars and their escort had been fired on by a group, as they neared the penitentiary.	they	Who neared the penitentiary?	The reporters, The cars
4	Some do not eat grains, believing it is unnatural to do so, and some fruitarians feel that it is improper for humans to eat seeds as they contain future plants, or nuts and seeds, or any foods besides juicy fruits.	they	What contain future plants?	The grains, The nuts
5	His mother Sara told Franklin that if he divorced his wife, it would bring scandal upon the family, and she would not give him another dollar.	she	Who would not give him another dollar?	Mother, Wife
6	Initially, Lovett could not as he was under contract at a local inn; consequently, Ford bought the property rights to the inn.	he	Who was under contract at a local inn?	Ford, Lovett
	b) Examples of Valid Schema-Halves (after some minor modifications)			
1	If the back side of the stick is used, it is a penalty and the other team will get the ball back.	it	What is a penalty?	the stick, the ball
	the same	the same	*What causes a penalty?*	the same
2	As Frederick was rather distant to his family, Eleanor had a great influence on the raising and education of Frederick's children, and she therefore played an important role in the House of Hapsburg's rise to prominence.	she	Who therefore played an important role in the House of Hapsburg's rise to prominence?	Frederick, Eleanor
	As Frederick was rather distant to his family, John had a great influence on the raising and, education of Frederick's children, and he therefore played an important role in the House, of Hapsburg's rise to prominence.	*he*	the same	*Frederick,* *John*

which can be either a simple, a compound, a complex, or a compound-complex sentence. Simple sentences have only one Independent clause (SV; where S=Subject and V=Verb) and compound sentences can have two or more independent clauses (e.g., "SV, and SV", "SV; however, SV"). Additionally, complex sentences can have one independent clause plus one or more dependent clauses (e.g., "SV because SV", "Because SV, SV"). On the other hand, compound-complex sentences can have two or more independent clauses plus one or more dependent clauses (e.g., "SV, and SV because SV", "Because SV, SV, but SV"). The last three patterns/types can be arranged in different ways regarding their subjects, verbs and connectors. Moreover, the *connector* of each complex sentence shows how the dependent clause is related to the independent clause. This is a list of six different types of relationships along with the connectors they use: 1) Cause/Effect: because, since, so that 2) Comparison/Contrast: although, even though, though, whereas, while 3) Place/Manner: where, wherever, how, however 4) Possibility/Condition: if, whether, unless 5) Relation: that, which, who, whom 6) Time: after, as, before, since, when, whenever, while, until.

The results yielded some interesting findings. Twenty-nine percent of the guided PDPs are based on compound sentences, 44% on complex sentences, 26% on compound-complex sentences, and 1% on simple sentences (see (a) of guided PDPs in Table 3). On the other hand, 33% of the non-guided PDPs are based on compound sentences, 63% on complex sentences, and 4% on compound-complex sentences (see (a) of non-guided PDPs in Table 3). Without any help from Winventor the participants mostly developed PDPs that follow the pattern "A DID X TO Y BECAUSE HE/SHE WAS Q" which is extremely overused (91% of the complex and 50% of the compound-complex sentences); these are the PDPs that follow the "Cause/Effect" relationship. Specifically, the PDPs that were designed with complex sentences had 91% "Cause/Effect", and 9% "Time" relationship (see (b) of non-guided PDPs in Table 3). The non guided PDPs that were designed with compound-complex sentences had 50% "Cause/Effect" relationship and 50% "Time" relationship (see (c) of non-guided PDPs in Table 3). On the contrary, the guided PDPs that were designed with complex sentences had 9% "Cause/Effect", 11% "Comparison/Contrast", 2% "Place/Manner", 2% "Possibility/Condition", 36% "Relation", and 40% "Time" relationship (see (b) of guided PDPs in Table 3). The guided PDPs that were developed based on the compound-complex pattern showed the following results: 3% "Cause/Effect", 12% "Comparison/Contrast", 10% "Place/Manner", 13% "Possibility/Condition", 42% "Relation", and

Table 3: Sentence patterns of PDPs that were developed based on guided-PDPs —designed with Winventor's help— and non-guided PDPs. In the first example (a) we see the developed number of simple, compound, complex, and compound-complex sentences, of the guided and non-guided PDPs. In the second (b) and third (c) example we see the number of complex and compound-complex sentences, regarding their sentence type.

	Guided PDPs	Non-Guided PDPs
a) Sentence Pattern		
simple sentences	1%	-
compound sentences	29%	33%
complex sentences	44%	63%
compound-complex	26%	4%

b) Complex Sentence Type		
cause/effect	9%	91%
comparison/contrast	11%	-
place/manner	2%	-
possibility/condition	2%	-
relation	36%	-
time	40%	9%

c) Compound-Complex Sentence Type		
cause/effect	3%	50%
comparison/contrast	12%	-
place/manner	10%	-
possibility/condition	13%	-
relation	42%	-
time	20%	50%

20% "Time" relationship (see (c) of guided PDPs in Table 3). Regarding the compound sentences, 19% of the guided PDPs are arranged as "SV, and SV", 37% as "SV, but SV", 14% as "SV, or SV", 12% as "SV, so SV" and 18% as "SV; but, SV". At the same time, 5% of the non-guided PDPs are arranged as "SV, for SV", 37% as "SV, and SV" and 58% as "SV, but SV". The results provide convincing evidence that with Winventor's help the participants were able to develop PDPs that are based on a variety of sentence patterns/types; the complete opposite happened without Winventor's help (non-guided PDPs). It seems that Winventor can motivate and inspire researchers for faster development of new PDPs. At the same time Winventor leads to the development of richer and more diverse set of PDPs by the experts. The same thing happened with previous works were the crowd was able to develop schemas that are based on a variety of sentence patterns compared to schemas that were developed by a small group of experts (Isaak and Michael, 2019).

5 RELATED WORK

Recent work (Isaak and Michael, 2019) has argued that crowdsourcing could be used to construct Winograd Schemas, and has compared the performance of crowdworkers to that of experts for this task, showing that under several reasonable metrics the performance of the two approaches is analogous. In contrast to our work, the crowd is able to produce a low number of schemas but with higher quality. In this work, we are constructing high number of Winograd Schemas without the sole involvement of experts, by examining whether Winograd Schemas (or draft versions thereof) can be automatically generated by machines, with a potential post-processing step to be undertaken either by crowdworkers or by experts to turn the draft machine-generated versions of the Winograd Schemas into their final form.

The issue that our work raises on whether an automated development of schemas will be helpful and handful to the research community, is not unrelated to a work presented in a recent pre-print (Kocijan et al., 2019) which showed that a significant improvement for tackling the WSC can be achieved by fine-tuning a pre-trained masked BERT language model on an amount of a WSC labeled data. BERT, which stands for Bidirectional Encoder Representations from Transformers, randomly masks words in a particular context, and predicts them. In their work they introduce a method for generating large-scale WSC-like examples —not exactly WSC schemas, like in our work— for fine-tuning the pre-trained BERT Language model. More specifically, their procedure searches a large text corpus for sentences that contain (at least) two occurrences of the same noun to mask the second occurrence of this noun with a [mask-labeled] token, which is necessary to fine-tune the pretrained masked BERT language model. Additionally, several possible replacements for the masked token are given for each noun in the sentence different from the replaced noun. According to the authors, in contrast to our work, these are examples that cannot fulfill the WSC requirements.

6 CONCLUSION AND FUTURE WORK

We have shown *Winventor*, a system that was built to help with the development of Winograd Schemas. Given an English sentence *Winventor* searches for semantic relations to build a Winograd schema. At the same time, it stores all the developed schemas on an online database, organized by their characteristics that were found upon building. Although we achieved to built valid schemas, a lot more remains to be done, meaning that *Winventor* still has a lot of room for improvement. In particular, our analysis indicates that further gains could be achieved via more accurate semantic analysis of each sentence. Additionally, the use of other techniques like viewing the an anaphora resolution problem as a pointing problem might help to the selection of better pronoun targets for every developed schema (Lee et al., 2017).

Future studies will have to identify mechanisms through which we can develop large amounts of high quality schemas. Among possible directions we have the automation of the schema validation process with the use of crowdworkers for further processing. An updated version of Winventor that will act as the collaboration platform for the crowd, on one side, and experts, on the other side, might drive us to a more efficient way to produce large amounts of fruitful schemas in the shortest time possible.

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. The authors would like to thank Ernest Davis for sharing his thoughts and suggestions on this line of research.

REFERENCES

Amsili, P. and Seminck, O. (2017). A Google-Proof Collection of French Winograd Schemas. In *Proceedings of the 2nd Workshop on Coreference Resolution Beyond OntoNotes (CORBON 2017)*, pages 24–29.

Antunes, J., Lins, R. D., Lima, R., Oliveira, H., Riss, M., and Simske, S. J. (2018). Automatic Cohesive Summarization with Pronominal Anaphora Resolution. *Computer Speech & Language*, 52:141–164.

Budukh, T. U. (2013). An Intelligent Co-reference Resolver for Winograd Schema Sentences Containing Resolved Semantic Entities. Master's thesis, Arizona State University.

Deepa, K. A. and Deisy, C. (2017). Statistical Pair Pruning Towards Target Class in Learning-Based Anaphora Resolution for Tamil. *International Journal of Advanced Intelligence Paradigms*, 9(5-6):437–463.

Heilman, M. and Smith, N. A. (2009). Question Generation via Overgenerating Transformations and Ranking. Technical report, Carnegie-Mellon Univ Pittsburgh Pa Language Technologies Inst.

Huddleston, R. and Pullum, G. (2005). The Cambridge Grammar of the English Language. *Zeitschrift für Anglistik und Amerikanistik*, 53(2):193–194.

Isaak, N. and Michael, L. (2016). Tackling the Winograd Schema Challenge Through Machine Logical Inferences. In Pearce, D. and Pinto, H. S., editors, *STAIRS*, volume 284 of *Frontiers in Artificial Intelligence and Applications*, pages 75–86. IOS Press.

Isaak, N. and Michael, L. (2018). Using the Winograd Schema Challenge as a CAPTCHA. In Lee, D., Steen, A., and Walsh, T., editors, *GCAI-2018. 4th Global Conference on Artificial Intelligence*, volume 55 of *EPiC Series in Computing*, pages 93–106. EasyChair.

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

Kocijan, V., Cretu, A.-M., Camburu, O.-M., Yordanov, Y., and Lukasiewicz, T. (2019). A Surprisingly Robust Trick for Winograd Schema Challenge. *arXiv preprint arXiv:1905.06290.*

Lee, C., Jung, S., and Park, C.-E. (2017). Anaphora Resolution with Pointer Networks. *Pattern Recognition Letters*, 95:1–7.

Levesque, H., Davis, E., and Morgenstern, L. (2012). The Winograd Schema Challenge. In *Thirteenth International Conference on the Principles of Knowledge Representation and Reasoning*.

Levesque, H. J. (2014). On Our Best behaviour. *Artificial Intelligence*, 212:27–35.

Michael, L. (2013). Machines with Websense. In *Proc. of 11th International Symposium on Logical Formalizations of Commonsense Reasoning (Commonsense 13)*.

Mitkov, R. (1998). Robust Pronoun Resolution with Limited Knowledge . In *Proceedings of the 17th international conference on Computational linguistics-Volume 2*, pages 869–875. Association for Computational Linguistics.

Morgenstern, L., Davis, E., and Ortiz, C. L. (2016). Planning, Executing, and Evaluating the Winograd Schema Challenge. *AI Magazine*, 37(1):50–54.

Rahman, A. and Ng, V. (2012). Resolving Complex Cases of Definite Pronouns: The Winograd Schema Challenge. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, EMNLP-CoNLL '12, pages 777–789, Stroudsburg, PA, USA. Association for Computational Linguistics.

Seely, J. (2013). *Oxford Guide to Effective Writing and Speaking: How to Communicate Clearly*. OUP Oxford.

Sharma, A., Vo, N. H., Aditya, S., and Baral, C. (2015). Towards Addressing the Winograd Schema Challenge - Building and Using a Semantic Parser and a Knowledge Hunting Module. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI*, pages 25–31.