




Blending NLP and Machine Learning for the Development of Winograd Schemas

Nicos Isaak¹(✉)  and Loizos Michael^{1,2}

¹ Open University of Cyprus, Nicosia, Cyprus
nicos.isaak@st.ouc.ac.cy, loizos@ouc.ac.cy

² CYENS Center of Excellence, Nicosia, Cyprus

Abstract. The Winograd Schema Challenge (WSC), a novel litmus test for machine intelligence, has been proposed to advance the field of AI. Over the last decade, AI researchers have become increasingly interested in this challenge. While a common and trivial task for humans, studies have shown that the WSC is still difficult for current AI systems. Tackling the challenge would likely require access to a sufficiently rich set of Winograd schema examples, which are currently limited in their number and too cumbersome to create completely manually. Towards addressing these limitations, we propose a machine-driven approach for the development of large numbers of schemas. Our empirical evaluation suggests that our developed system, which blends the advantages of Machine Learning and Natural Language Processing, is able to automatically develop Winograd schemas autonomously, or considerably help humans in the development task.

Keywords: Winograd Schema Challenge · Schema development · machine learning · Deep learning

1 Introduction

The Winograd Schema Challenge (WSC) [15], the task of resolving definite pronouns in carefully-constructed sentences, has been proposed to advance the field of AI [16]. It is believed that systems able to tackle the WSC will be able to support a wide range of commonsense and reasoning tasks that will help us understand human behaviour itself [16]. It seems that tackling the WSC will play a significant role in a wide range of current AI applications, as a step towards the development of machines that will automate or enhance basic human abilities—a traditional goal of AI that was laid back in the late 1950s [19].

Scholars seem to agree that the WSC is quite trivial for humans, but at the same time it is quite difficult for machines [21,28], due to the acknowledged lack of their commonsense reasoning abilities. In this line of research, in a recent work we have demonstrated the possibility of using the WSC as a novel form of CAPTCHAs [10]. This kind of challenge might spur research interest in

anaphora resolution which remains an essential task for the Natural Language Understanding (NLU) community [6]. Although the use of the WSC as a means to bring more researchers in the AI field is very important [10, 22], it would seem necessary that for this to happen one would require access to a good source of newly developed Winograd schemas, which itself has its challenges [21].

Aiming to develop a new system that is able promote the original goals of the WSC through the development of high quality schemas, this work presents Winventor (see Fig. 1). Winventor is a machine-driven approach that automates the schema development process and considerably helps humans in the development task. It combines NLP tools and deep learning into a flexible system able to produce efficiently a number of new Winograd schemas, which could be used to enhance the creativity and motivation of human experts for the development of schemas that were formerly designed by Winventor.

To lay a foundation for a machine-aided schema development process, we start by explaining the key challenge of the task, and continue by describing our system. Winventor’s architecture is based on three major approaches: based on NLP, based on deep learning, and a blended approach. In each case, we undertake several experiments regarding the a priori appropriateness of our system as a schema development mechanism. Our empirical evaluation suggests that the blended approach, which combines deep learning and NLP, can provide us with more schemas than the other two approaches. Finally, we review the implications of our results along with potential directions for future research.

The current paper extends an earlier version [12] presented at the 12th International Conference on Agents and Artificial Intelligence (ICAART). Compared to the conference paper, which was based on NLP-only techniques, we enhanced the schema development process through Deep Learning. In this regard, we developed some original ideas to blend Deep Learning with NLP, and the resulting system was able to provide larger numbers of schemas, while being 92% faster than our initial approach.

2 Problem Definition

The WSC consists of pairs of halves and the objective is to resolve a definite pronoun in each half. Each half comprises a sentence, a question and two possible pronoun targets or answers. The pronoun targets belong to the same gender and both are either plural or singular. Ostensibly, in each half there is a special word that when replaced by another word the answer also changes.

The WSC was named after Terry Winograd because of a well known example that was taken from his doctoral thesis [4], justified in terms of machine translation (“The city councilmen refused to give the women a permit for a demonstration because they [feared/advocated] violence”). The following schema (a pair of halves) illustrates the modified example, which meets the challenge rules:

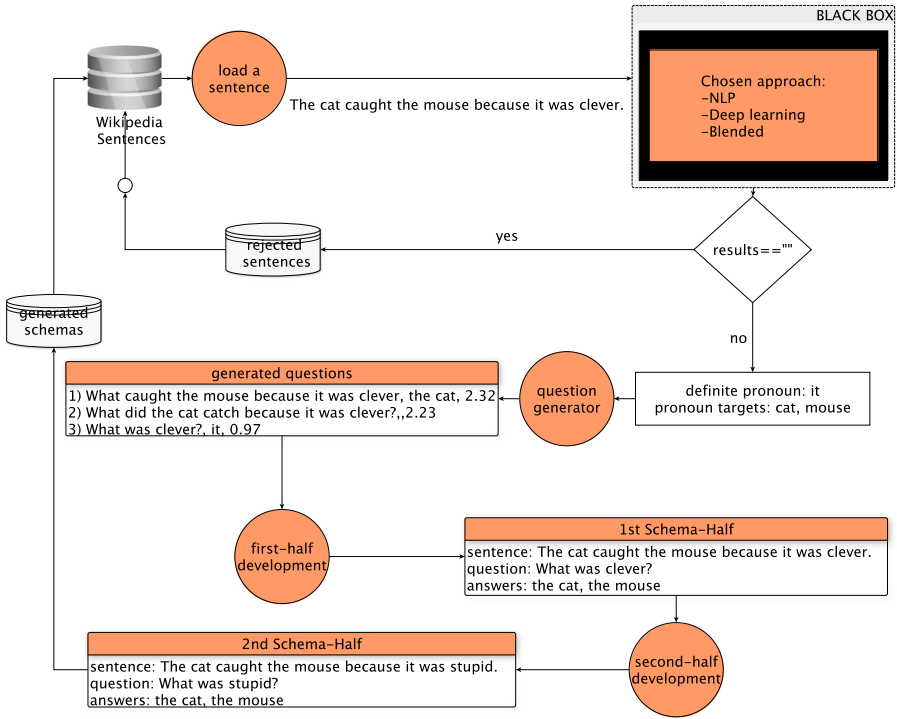


Fig. 1. Winventor’s high-level architecture: a system that automates the schema development process (adapted from [12]).

- 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.

It is believed that the WSC can provide a meaningful measure of machine intelligence, exactly 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 [15,16]. According to Levesque [15], 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. 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 (e.g., machine translation).

As stated in the literature, constructing a WSC corpus is a laborious job, requiring creativity, motivation, and inspiration [21]. In addition to this, as far as we know, only two WSC datasets are widely available: the Rahman and Ng’s dataset [25], consisting of 943 schemas (1886 halves), and the Levesque et al.’s dataset [15], consisting of 150 schemas (300 halves). It seems that a machine-driven approach for the development of schemas, which is a fertile area of research, would presumably help the community work on those WSC problems that require schemas, supporting and promoting at the same time further research on the WSC.

3 High-Level Architecture

In this section, we start with a high-level overview of Winventor by presenting how the engine works (see Fig. 1). If Winventor cannot develop a schema, it only develops a schema half that consists of a sentence, a definite pronoun, a question that indirectly points to the definite pronoun, and the two pronoun targets. Schemas that do not obey all constraints are known as “Winograd Schemas in the broad sense” [15]. In this regard, we developed Winventor to work in two different modes: strict or relaxed. With the strict mode enabled, Winventor develops schemas that strictly follow the WSC rules, whereas with the relaxed mode it may also develop schemas where the pronoun targets do not have to share the same gender.

3.1 A Simplified Example

At first Winventor loads an English sentence to evaluate if it can develop a schema. Winventor utilizes the sentence to output the definite pronoun and the two pronoun targets with one of the three specified approaches: using only NLP, using only deep learning, and a blended approach (see BlackBox in Fig. 1). If this is not possible, the current sentence is rejected. Otherwise: i) it proceeds with the question development, using a tool from the literature; ii) it constructs the first schema half by placing together the sentence, the question, and the two pronoun targets; iii) it finds the special word in the first sentence, generates the question, and develops the second schema half. More details on this procedure are given next.

Wikipedia Sentences: To be able to automatically develop schemas it is important to have access to a source of sentences. The Winventor framework can use any source, local or online, which can provide a bulk amount of English sentences. In its current version, Winventor is built on an extensible framework that allows access to a broad collection of nearly 88 million sentences from the English Wikipedia [9].

Developing the Schema-Half Questions: One of the most difficult parts of the challenge is to come up with appropriate questions [15]. According to Levesque, while doing so we must avoid two major pitfalls: 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 might be a stretch to do that since the question generation task is a very challenging and tedious process that dates back to 1976 [30].

To tackle this, Winventor uses the Heilman and Smith question generator¹ [7], a system able to generate questions based on a given piece of text. This question generator is freely available, easily customizable, and, at the same time, able to generate questions with a ranking strategy. Specifically, *Winventor* uses the question generator with the “*-keep-pro* and *-just-wh*” flags enabled. *Keep-pro* keeps questions with unresolved pronouns and *Just-wh* excludes boolean questions from the output. At the end, it selects the pronoun targets that relate to the pronoun that is given as the answer of the best question. For instance, in the next example “*The cat caught the mouse because it was clever*”, Winventor, via Heilman and Smith’s question generator, returns the following questions: i) “What caught the mouse because it was clever, the cat, 2.32”; ii) “What did the cat catch because it was clever?, ,2.23”; iii) “What was clever?, it, 0.97”. In the end, it selects the third question, as it is the only one that has as answer a definite pronoun: it.

Completing the Schema-Half: The next step for Winventor is the development of schema halves, meaning, pairs of sentences, questions, and pronoun targets. For each sentence and depending on the returned results (based on the approach used), Winventor might construct several schema halves. The number of the schema halves relates to the question generator results and the possible pronoun-target pairs. Specifically, for each valid pronoun-target pair, Winventor develops a number of schema halves, reordered by their significance (see *first schema-half* in Fig. 1).

Completing the Schema: Winventor develops schemas by keeping in mind that they are constructed so that there is a special word, in each sentence, which when replaced by another word, the answer also changes [21]. Hence, for every schema half it considers the following: i) it parses the question to identify the special word, which is a verb/adjective that participates in the questions’ triple relation (e.g., the word *clever* from the question “Who was clever”); ii) it returns the antonym of the special word, found in the previous step (e.g., from “clever” to “careless”), and iii) it modifies the returned word, in the question and the sentence, to match the tense of the second schema half (see *second-half* in Fig. 1). Regarding the triples, these are semantic scenes of the type *subject, verb, object* that are created through the sentence/question’s subjects and objects [9]. For instance, the triples [cat, caught, mouse] and [who, was, clever], which were used for the development of the schema in Fig. 1, were created from the parser’s *nsubj* and *dobj* relations (abbreviations of “nominal-subject” and “direct-object”).

¹ <http://www.cs.cmu.edu/~ark/mheilman/questions/>.

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

3.2 Developing Schemas Through NLP

Winventor makes use of various NLP tools to determine the meaning of each sentence [3]. This approach helps select the definite pronoun along with the pronoun targets based on the semantic analysis of a given piece of text. For instance, via various NLP tools, Winventor will be able to acquire sentences with good structure to select pronoun targets that agree in gender, number, and participate in relations with other words. In the sequel, we will introduce the major NLP components of Winventor by presenting how it generates schemas from scratch (see Fig. 2).

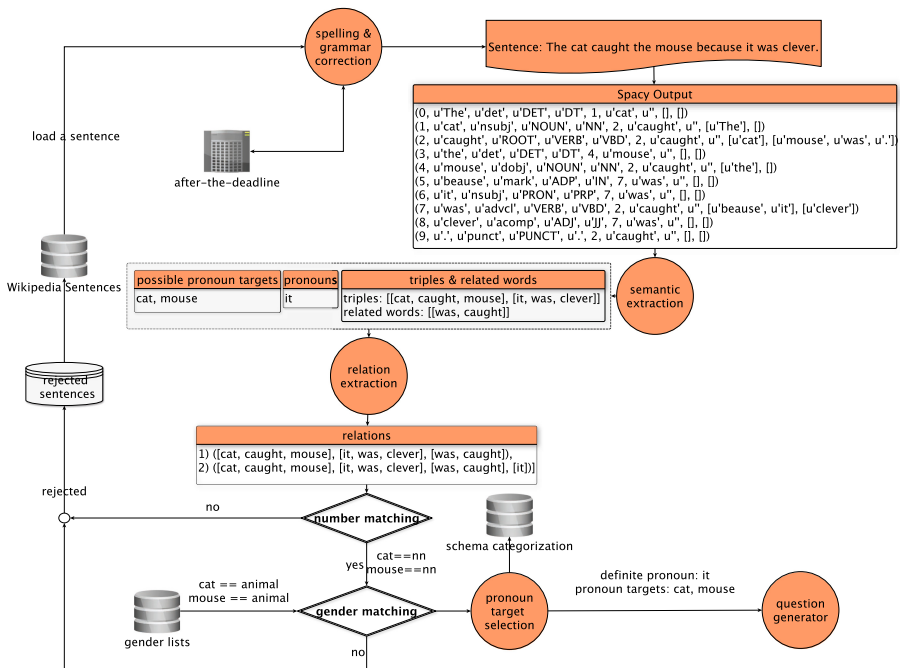


Fig. 2. A schema development process by Winventor using various NLP tools (adapted from [12]). The NLP section ends just before the question generator comes into play (see question-generator in Fig. 1).

Orthography and Spelling Correction: It is well-known that sentences found from online sources, like Wikipedia, might suffer from abbreviations, spelling errors, and misspellings of words. For instance, it was found that the percentage of misspellings of words on Wikipedia, relative to content, consistently increases year after year [29]. To avoid these kinds of problems, Winventor makes use of two tools from the literature. The first is the Google language-detection² library, which helps Winventor acquire only English sentences. The second is the After-the-Deadline³ language-checker, which automatically corrects spelling and grammar errors. Tools like After-the-Deadline offer efficient and effective ways of enhancing grammar accuracy and learning [23].

Sentence Word Relations: With the term word-relations we refer to semantic relations that can be concluded from a given text. While this task, which is necessary for the development of schemas, is very common and trivial for humans, it is quite challenging and difficult for machines. According to the literature, semantic relations of any given piece of text, are considered good if they can output essential relationships between the events and their participants [28], albeit, there is still no clear path to this goal [27]. To build good relations we have to consider various facts, like grammatical role, number, gender, and syntactic structure that can be given by dependency parsers [1, 9]. In this regard, Winventor utilizes the spaCy⁴ dependency parser to develop semantic relations from the Wikipedia sentences.

Through spaCy, Winventor parses each sentence to develop triples, related-words, and pronoun relations. *Related Words* are based on verbs that have a direct relation between them. For instance, the *caught-was* relation shows an indirect connection of the *nsubj* cat and the *dobj* mouse to the *adjective* clever (see *2nd and 7th line of the spaCy output* in Fig. 2). *Pronoun Relations* are relations where the pronoun targets (nouns or proper-nouns) are related to other words, via pronouns (see *relations* in Fig. 2). 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.

Pronoun-Target Selection: A challenging task for Winventor is to obtain the possible pronoun targets from each examined sentence. According to what the challenge dictates [15], the possible pronoun targets should be either a pair of nouns or proper-nouns that agree in gender and number. Winventor’s approach to discerning a list of possible pronoun targets includes the following: i) it utilizes spaCy’s entity recognition system to search for proper nouns, ii) it searches some pre-downloaded gender-lists to find nouns that have the same gender, and, iii) via spaCy’s dependency parser, it selects only nouns and/or proper-nouns that agree in number. The final result is to develop as many schemas as it can from

² <https://pypi.org/project/langdetect/>.

³ <http://www.afterthedeathline.com>.

⁴ <https://spacy.io>.

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

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

Pronoun-Target Appropriateness: In order to identify the appropriateness of each pronoun target pair, Winventor does the following: i) as previously mentioned, it keeps a track of number, gender, and the pronoun-gender agreement, ii) it stores the number of the triple relations that the pronoun targets participate in, and iii) it utilizes the Mitkov aggregation score [20], which is able to create a ranking list of nouns, according to some preferences. Mitkov’s work showed that when we have limited background knowledge, like in our case, we can consider five salience indicators to select the best pronoun targets: 1.) *Definiteness* refers to definite nouns, meaning that this kind of nouns should get a higher preference, in comparison to other nouns. Definite noun phrases’ score equals 0, whereas indefinite ones are penalized by -1 . 2.) *Indicating verbs* relate with nouns that are followed by verbs that are members of a specific Verb set (e.g., discuss, consider, investigate). These nouns’ score equals 1, otherwise 0. 3.) *Lexical Reiteration* refers to repeated synonymous noun phrases where they get a higher preference. A noun’s score equals 2 if it is repeated twice or more, 1 if it is repeated once, and 0 if not. 4.) *Non-prepositional* nouns are given a higher preference than prepositional nouns. A non-prepositional noun’s score equals 0, whereas a prepositional noun’s score equals -1 . 5.) *Collocation and Immediate-Reference* refers to nouns with identical collocation patterns, where they get a higher preference (Collocation nouns’ score equals 2, otherwise 0).

Completing the Schema: As shown in Sect. 3.1, after the selection of the best pronoun target, Winventor parses the sentence through the Heilman and Smith question generator and selects the one that has as answer the definite pronoun (see Sect. 3.1). Finally, it develops the two schema halves, constructs the schema and adds it to the Schema Database. Based on this approach, each developed schema is automatically classified into predefined categories and added to a schema-categorization DB (see Fig. 2). The categorization is done according to each sentence subject (e.g., Schwarzenegger - terminator - protection, birds - food) and the types of the pronoun target pairs (e.g., gpe, gerund, loc, country, facility, norp, org, etc.). Additionally, *Winventor* keeps track of the rejected

sentences with the following flags: 1.) Nouns and proper-nouns have not been found; 2.) Target Pronoun relations have not been found; 3) Questions have not been formed; 4.) not an English sentence; 5.) This was artificially created for previous WSC (see rejected-sentences DB in Fig. 2).

3.3 Developing Schemas via Deep Learning

Deep learning refers to a class of different techniques that allow computational models to learn representations of data through multiple levels of abstraction [14]. As stated in the literature, deep learning is extremely good at finding complex data structures and is, therefore, suitable for different fields [14]. In this regard, we aim to train three deep learning models to help Winventor in the schema development process. Specifically, we train: 1.) the sentence model for the selection of sentences, 2.) the pronoun model for the selection of the definite pronoun, and 3.) the pronoun-targets model for the selection of the best pronoun-target pair, from each examined sentence.

For the development of a schema-half/schema, our algorithm starts with the sentence model to select an appropriate sentence, continues with the pronoun model to select the best definite pronoun from the previously selected sentence, and, finally, ends with the pronoun-targets model to select the best possible pair of answers. In the sequel, we will introduce the deep learning models with the datasets used for their training and testing (see Fig. 3).

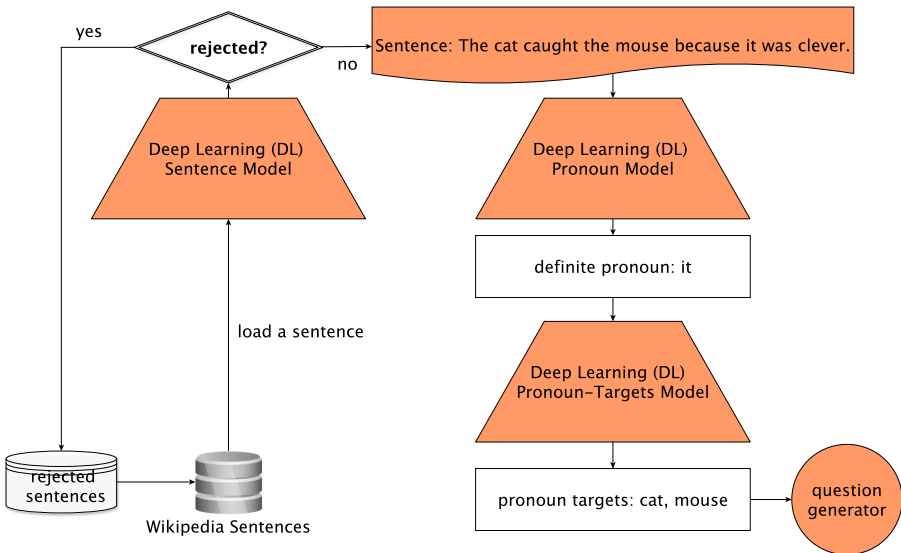


Fig. 3. A schema development process by Winventor using deep learning. The deep learning section ends just before the question generator comes into play (see question-generator in Fig. 1).

Dataset Preparation: The central aspect of deep learning is that the layers of a deep neural network are automatically learned from data using a general-purpose learning process [14]. In this sense, deep learning algorithms are not capable of understanding the text models but only map the statistical structure of written language, which is supposedly sufficient to solve simple textual tasks [2]. On the other hand, we know that in problems where data are limited, deep learning often is not an ideal solution [18]. In this regard, we employed a data synthesis/augmentation procedure to increase the size of our training data. In an attempt to use a different training set than that of Rahman and Ng’s, we begun with Levesque et al.’s dataset [15], which consists of 150 schemas, and ended up with 30,000 schemas.

Sentence Model: This model utilizes a classifier that is responsible for selecting appropriate sentences for the development of schemas (see *DL-Sentence-Model* in Fig. 3). Given any English sentence, our sentence model returns a value in the range of 0–1, where values >0.5 indicate high-grade (suitable) sentences for the development of schemas. Valid or high-grade sentences are eligible for further processing for the development of schemas, whereas non-valid are not.

To train our classifier, we used training data with positive and negative examples. Positive examples refer to sentences that were used in the development of the Levesque et al.’s dataset [15], whereas negative examples refer to sentences that cannot be used in the development of schemas. To increase the number of positive-examples we proceeded as follows: 1.) we parsed each sentence and removed the punctuation characters, 2.) for every noun, adjective, verb, and adverb, we developed a list with their synonyms, and, 3.) based on a random combination of their synonyms, we developed a list of new sentences, 4.) via spaCy, we replaced the words of each examined sentence with their part of speech (part-of-speech tagging). Through the part-of-speech tagging, our model does not need to use knowledge transfer between various domains which is a characteristic feature for many deep learning approaches [17]. Regarding the negative examples, for every positive sentence, we developed a negative one: i) by randomly removing some words, and ii) by randomly reordering its tagging (see Table 1).

Table 1. A sentence transformation example for the development of the training and testing dataset of our sentence model.

	Part of speech tagging
Sentence	The city councilmen refused the demonstrators a permit because they feared violence
Part of speech	DET NOUN NOUN VERB DET NOUN DET NOUN ADP PRON VERB NOUN
Synonym-positive example	DET ADJ NOUN NOUN VERB DET NOUN DET PROPON NOUN ADP PRON VERB NOUN
Synonym-negative example	DET NOUN VERB PART DET DET ADP PRON VERB

Pronoun Model: A key problem within the schema development process is the selection of the definite pronoun, as this directly relates with the selection of the pronoun targets. To that end, we developed the pronoun-model, which is responsible for selecting the definite pronoun in sentences that were returned by the sentence model. Given any tagged-English sentence with multiple pronouns, this model returns the best possible pronoun, which could be used as our definite pronoun. Specifically, for each sentence with a (marked) pronoun, this model returns a confidence score in the range of 0–1; the higher the score, the higher the confidence for the specific pronoun.

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

Pronoun-Targets Model: This model is responsible for the selection of the best pronoun target pair (answers), in sentences that were selected by the pronoun model. Recall that the WSC is about resolving the definite pronoun to one of *two* possible pronoun targets, in each schema. Hence, in each examined sentence, this model aims to output the best answer pair to be used in the construction of the schema. Given any tagged English sentence, with two words marked, this model returns a confidence score in the range of 0–1 that indirectly shows the best pair for the development of the schema.

For training purposes and specifically for the building of our positive examples, in all of the synonym sentences the correct pronoun target pair was marked. This resulted in pairs of multiple words, as in some schemas the correct answers consisted of compound nouns. For instance, in the example used in our previous models our algorithm would return “DET <NOUN NOUN> VERB DET <NOUN> DET NOUN ADP PRON VERB NOUN”, with the position of the two pronoun targets marked. For the construction of the negative examples, we have followed a similar procedure, where, for each positive sentence, we build a new negative one with its tagging shuffled.

Schema Development: We continue to discuss how Winventor develops schemas via the deep learning approach. At the start, each Wikipedia sentence is validated by the sentence-model, where for every *valid* sentence (>0.5) it proceeds to the next step to search for the definite pronoun (see Algorithm 1). Winventor replaces every sentence word by its part-of-speech, marks the pronoun (<PRON>) and parses it though the pronoun-model to retrieve its score; this

process is repeated for every pronoun in the sentence and at the end it selects the pronoun with the biggest score. The next step is to find the best pronoun-target pair of the sentence that indirectly relates to the definite pronoun. To that end, Winventor randomly creates all the combinations of two, three, and four words. Then, for every combination, it marks the combination's words in the sentence (part-of-speech) and parses it through the pronoun-targets model to retrieve its score. At the end, it selects as the best pair the pair with the highest score. After the selection of the sentence, the definite pronoun, and the pronoun target pair, Winventor develops the two schemas halves, following the same procedure as stated in the previous sections (see Sect. 3.2). The only difference within this approach, is that each developed schema cannot be automatically classified into predefined categories to be added to the schema-categorization DB.

Algorithm 1. Schema development via deep learning.

```

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

```

3.4 The Blended Approach

In this section, we describe how we blended the NLP and deep learning approaches with the ultimate goal of developing a more efficient and more effective solution. In particular, we modified the pronoun-target selection process based on factors described in the previous sections (see Algorithm 2), by replacing the deep learning solution for that task with the gender, number, pronoun-gender, and triple factors, in order to select the best answer pair (see Fig. 4).

Thus, the blended approach proceeds as follows: 1.) via the sentence model it parses Wikipedia sentences to select an appropriate sentence for the development of a schema; 2.) through the pronoun model it returns the definite pronoun of the examined sentence; 3.) from the sentence it selects only nouns or proper-nouns and builds all the possible combinations (see *relations* in Algorithm 2); 4.) at the same time, it searches for possible compound-nouns and replaces each noun accordingly; 5.) next, for every pair of answers, it estimates a score value where it adds 1 if they are both members of the same number-class. It does the same, in case the two candidates share the same gender, participate in triples (*as subj and dobj*), and have a pronoun-gender agreement with the definite pronoun; 6.) it adds the score to a list of scores (see *answersScore* in Algorithm 2); 7.) In the last step, it returns the best answer pair, which is the pair with the highest score.

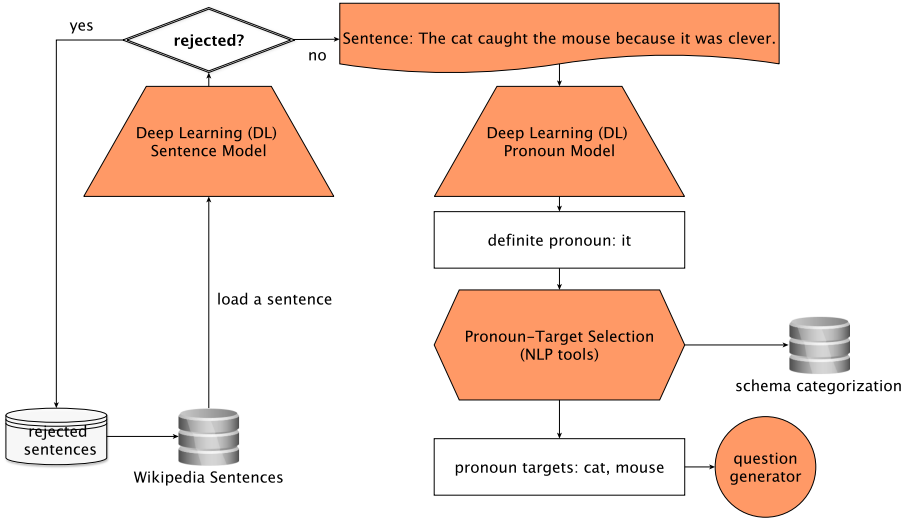


Fig. 4. A schema development process by Winventor using deep learning and NLP tools (for a further explanation on the NLP tools, see Algorithm 2). The process ends just before the question generator comes into play (see question-generator in Fig. 1).

Completing the Schema: The blended approach generates the questions and develops the two schema halves following the same procedure as stated in the previous sections (see Sect. 3.2). Furthermore, similarly to the NLP approach and contrary to the deep learning approach, each developed schema is automatically classified into predefined categories and added to the schema-categorization DB (see Fig. 4).

4 Experiments and Results

In this section, we describe the results from several studies that we undertook to evaluate Winventor’s performance on the development of schemas, based on the aforementioned approaches. Each of the following subsections reports on one of the approaches.

4.1 NLP Approach

Here we describe the results from three studies that we undertook to evaluate Winventor’s performance, on replicating existing Winograd Schemas from a well-known WSC dataset, on developing new Winograd Schemas from scratch, and on helping humans develop new Winograd Schemas.

Algorithm 2. Blended pronoun-target pair selection.

```

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

```

Schema Replication: In this experiment, we have tested Winventor on replicating schemas from Rahman and Ng’s dataset [25], which is a challenging dataset of 943 schemas, where each schema half consists of a sentence, a definite pronoun (instead of question), and two possible pronoun targets. The average sentence length of the database was 14 words. For the purpose of this experiment the *strict* mode was disabled, as this is a dataset that was developed under the “broad” flag. By giving Winventor the sentence of the first half of each schema, we wanted to evaluate if it can produce similar results as in the dataset. For each sentence, Winventor was requested to develop all the possible schemas, storing at the same time all of the developed relations and factors (e.g., Mitkov-score, gender, number, and pronoun-gender-agreement variables).

Schemas: The results revealed that 416 sentences resulted in 990 halves where 848 were schemas. More than two hundred schemas (254 schema halves of which 214 are schemas) were found to match with the Rahman and Ng’s dataset, meaning that they have the same definite pronoun and the same pronoun targets. At the same time our system rejected 527 sentences, for the following reasons: 1.) Nouns and proper-nouns have not been found (10 sentences) 2.) Target Pronoun relations have not been found (502 sentences) 3.) Questions have not been formed (13 sentences) 4.) Not an English sentence (2 sentences were wrongly identified). 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, over fifty percent of the sentences were rejected because of pure parsing: *Target Pronoun relations have not been found.*

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

Table 2. A snapshot of *Winventor*'s developed questions on Rahman and Ng dataset.

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

noun problems might suggest that resolving proper-nouns is more challenging than resolving nouns [1].

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

Question Development: Although the original dataset did not include questions, *Winventor* was able to produce schemas with valid questions (see Table 2). This shows that the parsing of sentences through the question generator, and, at the same time, the selection of the best appropriate question, returned useful results.

Non-matching Schemas: Our results showed that *Winventor* was able to develop 990 halves from 416 sentences, meaning that for each sentence multiple schemas were developed. On the other hand, our analysis showed that only 254 halves (214 schemas) were found to match the original dataset, meaning

that 74% of the schema halves were among those that were rejected as non-matching schema halves. Recall that there are sentences that contain multiple number of nouns, proper-nouns and pronouns, which means that there is a big chance to have sentences that could lead to more than one schema. For instance, in the original schema dataset we have the following halves: i) *Sentence: Arnold Schwarzenegger cannot terminate John Conner, because he is protecting him. Definite-Pronoun: he, Answers: Arnold Schwarzenegger, John Conner*, and, ii) *Sentence: Arnold Schwarzenegger cannot terminate John Conner, because he is the leader of the resistance. Definite-Pronoun: he, Answers: Arnold Schwarzenegger, John Conner*. Although Winventor did not manage to build the requested schema, it returned the following results: i) *Sentence: Arnold Schwarzenegger cannot terminate John Conner, because he is protecting him. Definite-Pronoun: he, Question: Who is protecting him? Answers: Arnold Schwarzenegger, John Conner*, and ii) *Sentence: Arnold Schwarzenegger cannot terminate John Conner, because he is protecting him. Definite-Pronoun: him, Question: Who is he protecting? Answers: Arnold Schwarzenegger, John Conner*. As we can see, the question of the second schema half, which was returned by Winventor, refers to a different pronoun than the original schema half. Given that the original dataset was developed under the “broad” flag, these two halves can be taken together to consist a new valid schema, albeit different from the original one.

Selecting the Best Schemas: Given that for any sentence multiple schemas might be created, many open questions remain regarding the fastest way to select the best ones (for instance, to select the 254 halves from our database of 990 halves). To that end, we further analyzed the relation between the developed halves and different *factors* (e.g., Mitkov-score, triple, gender and pronoun-gender agreement). The results showed a direct relation between our factors and the selection of the best schema halves. For instance, if we select all the schema halves that agree on gender, number, participate in triples, and have a pronoun-gender agreement, we have an 89% success rate. Furthermore, our results showed the importance of the triple factor (nsubj-dobj); it was shown that if we remove the triple factor the success rate drops to 85%. Additionally, our analysis showed that if we select the schemas according to their Mitkov-score, we have an 82% success rate, meaning that Mitkov’s theory works well when we have limited background knowledge.

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

Design: For our experiments we selected the Microworkers (MW) platform⁵, which can be considered as one of the best available crowdsourced platforms [8, 24]. Specifically, we designed a questionnaire using LimeSurvey⁶ and posted the link on the MW platform. We divided our questionnaire into two sections, where the first section consisted of twenty randomly selected Winograd halves, whereas the second consisted of ten Winograd *schemas*; every single example was automatically developed by Winventor. Examples that were included in the first section were excluded from the second one. The questionnaire started with the first section and continued with the second one, where each half/schema was displayed on a single screen, followed by the question; in each example three choices were displayed side-by-side: i) Valid Schema - Easy to Solve, ii) Valid Schema - Hard to Solve, iii) Non-Valid Schema. Furthermore, all participants were informed that once the survey started, they could not change a submitted answer. Additionally, before taking the survey, each participant had to do the following: i) read a consent form and agree to participate, ii) select their age and their English language literacy level, and iii) pass a training phase to get familiarized with the task. In the training task, which consisted of few examples similar to that of our questionnaire, immediate feedback (correct or incorrect) was given after each trial.

Participants: Our experiment was performed during May 2019, where a total of one hundred MW workers were recruited, aged between 18 and 65. Our participants were residents of English speaking countries, and were screened by means of a qualification task from the Microworkers platform. The total cost of our campaign was \$250.

Results: In the first section the participants characterized the schema halves as *valid* with a mean of 69% ($\sigma = 0.15$). In the second section they characterized the schemas as *valid* with a mean of 73% ($\sigma = 0.17$). It seems that the positive difference in favor of the schemas might have happened not because of the quality of the schemas, which are harder to develop, but because of the following reasons: i) the participants were able to see the two halves at the same time, which seems to help them understand the meaning of the schema, and ii) sentences that were found appropriate for the development of schemas might have simpler structure. Generally speaking, we believe that our results must be taken with a grain of salt. Specifically, we are not claiming that this system can be used to develop schema/halves without the need of reviewing. For instance, in order to validate the next schema-half we need to change a word in the question (*is to causes*): *sentence: If the back side of the stick is used, it is a penalty and the other team will get the ball back. question: What is a penalty? answers: the stick, the ball.*

Winventor as an Assistant: Within this experiment we evaluated if Winventor can assist humans in the schema development process. To delineate it from

⁵ www.microworkers.com.

⁶ <http://limesurvey.org>.

the previous experiment, we asked ten colleagues who have prior experience in developing schema halves to design new schemas from scratch, in a specified period of time. For the sake of simplicity, participants were asked to develop only schema halves. In order to identify Winventor’s a priori appropriateness as a *teammate*, we divided the experiment in two sections. The experiment started with the first section, where participants were asked to develop as many schema-halves as they can without Winventor’s help, in ten minutes; these were called non-guided schema-halves. They continued with the second section where the experiment was then replicated under conditions in which we gave them access to fifteen randomly selected schema halves, developed by Winventor; the results were called guided schema halves.

Results: On average, we found that Winventor helped participants develop twenty schema halves, whereas without Winventor’s help, they only developed seven schema-halves. Ostensibly, a schema sentence analysis that we undertook, showed that Winventor helped them develop schema halves that are based on different sentence patterns/types (see Table 3). These tests revealed that the guided developed schemas have a variety of sentence types (29% based compound sentences, 44% on complex sentences, 26% on compound-complex sentences, 1% on simple sentences). On the other hand, regarding the non-guided schema-halves, results showed that 33% of them are based on compound sentences, 63% on complex sentences and 4% on compound-complex sentences.

Furthermore, we analyzed our results based on the sentence structure of each schema-half. Regarding the complex and compound-complex sentences, this is a list of six different types of relationships along with the connectors they use: 1.) Cause/Effect 2.) Comparison/Contrast 3.) Place/Manner 4.) Possibility/Condition 5.) Relation 6.) Time. Results highlighted that the guided schema-halves are based on a variety of relationships, which is much richer than the non-guided schema-halves (see Fig. 5). All in all, non-guided schemas were mostly designed using the cause/effect relationship (70.5%) with the connectors “because, since, so that”. The rest of them were designed by using connectors of “Time” relationship (e.g., after, as, before, since, when, whenever, while, until). Regarding the guided schemas, our results showed that they were designed based on a much richer set of connectors: 6% “Cause/Effect”, 11.5% “Comparison/Contrast”, 6% “Place/Manner”, 7.5% “Possibility/Condition”, 39% “Relation”, and 30% “Time” relationship (see Fig. 5). We also incorporated a similar analysis for the schema-halves that are based on compound sentences. As anticipated, our analysis showed that the guided schema-halves, compared to non-guided schema-halves, were developed based on a variety of relationships. Specifically, 19% of them are arranged as “SV, and SV” (S: subject, V:verb), 37% as “SV, but SV”, 18% as “SV; but, SV”, 14% as “SV, or SV” and 12% as “SV, so SV”. On the other hand, 58% of non-guided schema-halves are arranged as “SV, but SV”, 37% as “SV, and SV” and 5% as “SV, for SV” (see Fig. 6).

Our observations show that Winventor seems to motivate and inspire participants develop richer and more diverse schema halves, in the shortest time

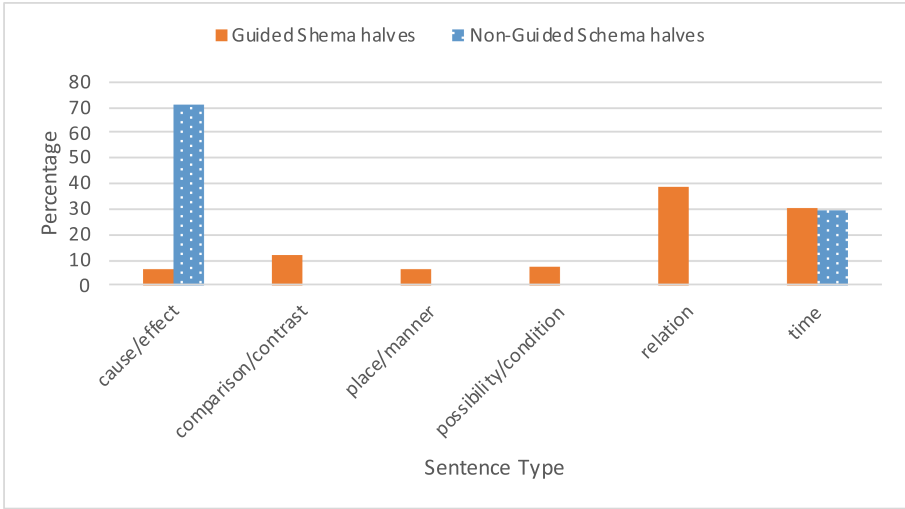


Fig. 5. Complex and compound-complex sentence types that were developed based on guided-schema halves (designed with Winventor’s help) and non-guided schema halves.

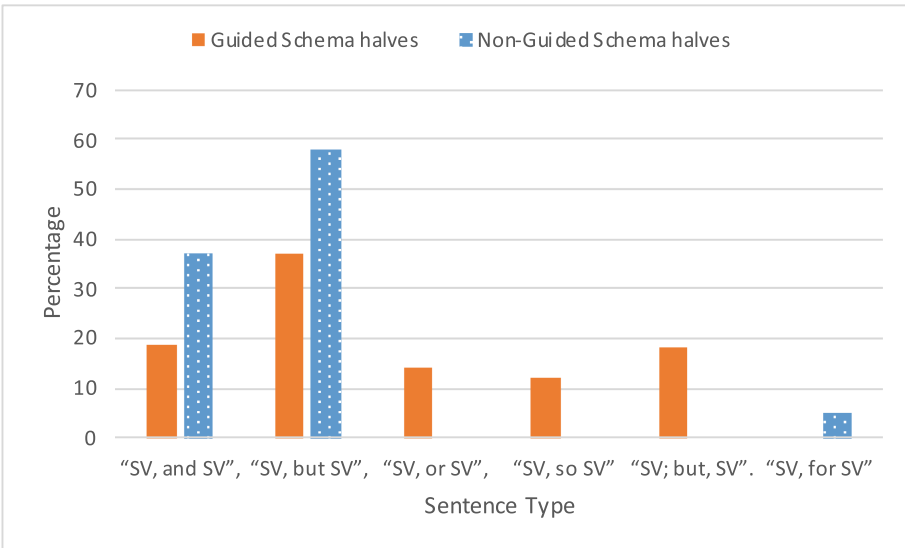


Fig. 6. Compound sentence types that were developed based on guided-schema halves (designed with Winventor’s help) and non-guided schema halves.

possible. The results are inline with a recent work that we undertook [11], where it was shown that schemas developed by crowdworkers have a similar hardness to those developed by experts.

4.2 Deep Learning Approach

In this section, we present the results of the deep learning approach. We begin by presenting the results regarding our models' training and then continue by applying the methodology on the development of schemas. For the purpose of these experiments, we trained and evaluated our system on Levesque et al.'s dataset [15]. We divided our samples into a training and a testing set following the ratio of 70%–30% and evaluated our three models. Initial results showed an accuracy of 89% on the sentence selection process, 94% on the pronoun selection process, and 91% on the pronoun-target selection process.

Schema Replication: Within this experiment, we have tested if our proposed approach is able to replicate schemas from Rahman and Ng's dataset. That is, Winventor loads all sentences from the first half of each schema, and tests if it can produce the same or similar results as the second half of each schema. Here, in contrast to the NLP approach, Winventor develops one schema/schema-half for each examined sentence (see Algorithm 1).

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

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

Pronoun Targets: Contrary to our expectations, Winventor returned the correct answers in only 9% of the cases (74 sentences). On the other hand, this is in line with the challenge difficulties and design purposes. Recall that the whole idea behind the WSC is to develop systems that can resolve the definite pronoun to one of its two conferences, in each schema-half. In this regard, it seems that it might be a stretch to find the correct pronoun targets in sentences with multiple candidates.

Schemas: Results showed that 745 sentences resulted in 162 schemas. This is in line with our Pronoun-Target results as the question generator automatically rejects questions that have as answers possible pronoun targets (e.g., the notification "A noun is in the question" was returned in 1698 cases). Regarding our previous results, which showed that 416 sentences resulted in 254 valid schemas, it seems that our NLP-approach can provide us with more schemas than our deep learning models. On the other hand, considering the fact that: i) the deep learning method achieved better results on the selection of both sentences and

Table 3. A subset of the schemas that were developed by humans with and without Winventor's help. The first five examples are a subset of the schemas that were given in order to inspire humans in the development of new Winograd schemas.

Sentence	Question	Answers
Automatically developed schemas		
1 Your governors are unjustifiably killing people and they only write the crime of the killed person to inform you	Who only write the crime of the killed person to inform you?	The governors, The people
2 This river may have been shaped by God, or glaciers, or the remnants of the inland sea, or gravity or a combination of all, but the Army Corps of Engineers controls it now	What does the Army Corps of Engineers control now?	The river, the island sea
3 Some do not eat grains, believing it is unnatural to do so, and some fruitarians feel that it is improper for humans to eat seeds as they contain future plants, or nuts and seeds, or any foods besides juicy fruits	What contain future plants?	The grains, The nuts
4 The Greeks hiding inside the Trojan Horse were relieved that the Trojans had stopped Cassandra from destroying it, but they were surprised by how well she had known of their plan to defeat Troy	Who were surprised by how well she had known of their plan to defeat Troy?	Greeks, Trojans
5 The reintroduction of a permanent diaconate has permitted the Church to allow married men to become deacons but they may not go on to become priests	Who may not go on to become priests?	The men, The deacons
Schemas that were developed from humans with Winventor's help		
1 Because of a misunderstanding Hitler had with Stalin, he attacked his country, misjudging the level of preparation needed to withstand harsh weather conditions, and subsequently that misunderstanding had cost him the war	Who the misunderstanding cost the war?	Hitler; Stalin
2 Even though Meredith was the one who had committed the fraud, Andrea Wanted to fix everything, so she confessed and went to jail	Who went to jail?	Meredith; Andrea
3 Some fruitarians feel that it is improper for humans to eat seeds as they contain future plants	What contain future plants?	Grains, humans
4 This river may have been shaped by God, or glaciers, or the remnants of the inland sea, but the Army Corps of Engineers controls it now	What does the Army Corps of Engineers control now?	the river; the island sea
5 It is allowed by the Church married men to become deacons but they may not go on to become priests	Who may not go on to become priests?	Men; deacons
6 Since everybody could always rely on Tommy, they expected him to have a plan, and so did John, but unfortunately he got shot during this specific operation by their worst enemy	Who got shot?	John; Tommy
Schemas that were developed from humans without Winventor's help		
1 Jack gave John the book, although he didn't need it	Who didn't need the book?	Jack; John
2 My cat hates my dog because it is jealous	Who is jealous?	My cat; My dog
3 Alice tried to reach her mother's head but she was too short	Who was too short?	Alice; her mother
4 Mary tried to calm her mother, but she was really stressed	Who was stressed?	Mary; her mother
5 Kids talk to their parents but sometimes they are too busy to listen	Who are busy?	The kids; the parents
6 Ice cream is really nice with sirup, especially when it's caramel flavoured	What is caramel flavoured?	The ice cream; the sirup

Table 4. Number of developed schemas/schema-halves based on various approaches (NLP, deep learning, and blended approach) that match Rahman and Ng’s dataset (943 schemas). Regarding the initially-rejected sentences of the deep learning and blended approaches, there is an additional number of 28 sentences where our pronoun-model did not manage to correctly identify the definite pronoun.

	Rejected sentences	Used sentences	Matching answers	Matching schemas	Matching halves
NLP	527	416	254	212	254
DL	170	745	75	27	38
BL	170	745	389	234	332

definite pronouns, and ii) the question generator directly relates with the selection of the best pronoun targets, it seems that better pronoun targets could lead to the development of more schemas. That is, it appears that the pronoun-targets model is the one that thwarts the full potential of our deep learning approach.

4.3 Blended Approach

Below, we present the results by applying the methodology described in the blended approach section (see Sect. 3.4). Specifically, we performed an analysis regarding Winventor’s ability in replicating and developing schemas from scratch. Additionally, we performed a speed analysis comparison between the blended and the NLP approach, which indirectly relates to the availability of schemas.

Schema Replication: Within this experiment, we report results based on Winventor’s blended-mechanism on replicating schemas from Rahman and Ng’s dataset. Like before, the results are expressed in terms of accuracy.

Results showed that in 50% of the cases (389) Winventor selected the correct answer pair, which is 40% more than the deep learning approach (see Table 4). Regarding the schema development process, our analysis showed that Winventor was able to develop 332 schema halves that match the Rahman and Ng’s dataset; 70% of them (234) were found to be schemas. In the case of schema halves, this means 27% more than the NLP, and 158% more than the deep learning approach. Furthermore, in the case of schemas, this means 10% more than the NLP, and 159% more than the deep learning approach.

We observed that if we remove any of the NLP factors the performance is further reduced, showing the importance of every single factor in the schema development process. The results ultimately show that our blended approach replicates more schemas than both the other methods, which is very important considering the challenge difficulties. On the other hand, our findings would seem to show that the development/sentence ratio of the NLP approach is better than

the blended approach. According to our findings, 61% of the sentences of the NLP approach were successfully used in the development of schema halves, whereas in the blended approach only 43% of the sentences resulted in schema halves. This suggests that the NLP approach works better with the question generator mechanism. This may have occurred because the question generator needs to successfully output the semantic relations of a given piece of text in order to develop the questions; It seems that sentences that were rejected by the NLP approach are very difficult to be used with the question generator [7]. The results might suggest that a better question generator could lead to the development of more schemas.

We also performed a speed analysis. Since the availability of more schemas directly relates to the ability to run a WSC-based CAPTCHA service [10], it is important for Winventor to be able to develop schemas at a sufficiently fast pace. Our results showed that the blended approach is able to return results in 1.5 h instead of 5 h for the NLP approach, meaning that Winventor can develop, on average, 3 schemas per minute.

Schema Development: Within this experiment, we report results of Winventor’s blended-approach, on developing schemas from scratch. In this regard, we fed Winventor with the same Wikipedia dataset, like in Sect. 4.1, and compared the two approaches. Specifically, we randomly selected 2000 Wikipedia sentences that were previously used for the NLP approach.

In contrast to previous findings—recall that the NLP approach returned 23 schema halves of which 16 were schemas—the blended approach returned 39 schema halves of which 25 were schemas. At the same time, 1587 sentences were rejected from our sentence model (79%), whereas 1978 sentences were rejected by the NLP approach (99%). On average, the blended approach provided 52% more schema-halves and 44% more schemas than the NLP approach. In general, regarding the number of the developed schemas, the performance was a little disappointing. The prime cause of this discrepancy seems to be due to the structure of the sentences found on the Web. This realization is in line with the previous section, where Winventor was able to replicate more schemas, as the sentences that were used were designed by humans. Furthermore, not surprisingly though, there were some discrepancies due to our sentence model limitations. Recall that in previous examples all of the sentences were validated as they were manually designed by humans. On the other hand, as some Wikipedia sentences did not include pronouns, our deep learning sentence-model mistakenly identified them as valid sentences. This might lead to the conclusion that our data augmentation process was not sufficient, meaning that more valid sentences are required in order to do better training.

One of the most surprising results to emerge from our analysis is the number of the developed schemas compared to the time needed. According to our results, the blended approach parsed 20000 sentences in 1 h, whereas the NLP approach required 12 h; the results show that the blended approach is 91.67% faster than the NLP approach. In general, although performance was not perfect, we still

believe that results highlighted the importance of mixing machine learning and semantic analysis to achieve better results. In this regard, and based on both the Wikipedia dataset at hand (88 million sentences) and our current results (39 schema halves from 20000 sentences), it seems that Winventor could provide us, approximately, with 1.7 million schema halves or 1 million schemas when applied on the entire Wikipedia dataset. However, we are aware that these numbers are not guaranteed, as this depends on the structure of the sentences found on the Web. Overall, the results ultimately show that via the interaction between the two approaches we were able to enhance the schema development process. This also shows the possibilities of combining the two approaches in future challenges, which is already in full swing with recent research in the field of AI [18].

5 Related Work

The first and only Winograd Schema Challenge that took place in 2016 required the organizers to manually develop a collection of 89 Winograd schemas. For evaluation purposes they designed a questionnaire where participants were requested to resolve the schemas [5]. As stated in the literature, the development of Winograd schemas was found to be troublesome, difficult and too burdensome to do on a yearly or biennial basis [21]. The challenge, which was designed based on the questionnaire results, consisted of two rounds, where, the first one included 60 Winograd halves (as pronoun disambiguation problems, or PDPs) and the second 60 Winograd schemas.

In a recent work, which in part served as a motivation for this one, we have demonstrated the possibility of using the WSC as a novel form of CAPTCHA [10]. While designing good CAPTCHAs is a tedious task, through an experiment that we designed and undertook we showed that a Winograd CAPTCHA is generally faster to solve than, and equally entertaining with, the most typical existing CAPTCHA tasks. The ultimate goal of that work was to attract security researchers to participate in future challenges for tackling the WSC. As this CAPTCHA service requires multiple Winograd schemas to be displayed on a daily basis, it is in direct relation with what Winventor seeks to do: to offer a continuously-replenished pool of Winograd schemas.

Davis [4] demonstrated the possibility of using the Winograd schemas as a machine translation challenge. According to the author, Winograd schemas with special gender characteristics of their answers could be used to advance the machine translation field. Consider, for instance, the following schema-half, a slightly modified example from Davis et al.’s dataset, to see how it can be used in a translation from English to French: “The city councilmen refused to give the women a permit for a demonstration because they [feared/advocated] violence”. In the first sentence (with “feared” as the special word), the definite pronoun “they” would refer to councilmen and would be translated to “ils” in French, whereas in the second sentence (with “advocated” as the special word), “they” would refer to women and would be translated to “elles” in French. In this regard, Winventor could be used to provide us with schemas that could enhance a translation-schema database.

Our experimental set up bears a resemblance to the one proposed in another work [11], where it was shown that workers who collaborate on crowdsourcing platforms could develop Winograd schemas of high quality, similar to that of experts. Compared to this work, where we are able to construct high numbers of draft machine-generated schemas, workers are able to produce a limited number of schemas but of higher quality. It seems that the collaboration of the crowd with systems like Winventor could potentially help overcome the limitations of the automated development of Winograd schemas.

Recent work has shown a significant improvement on the WSC by fine-tuning large pre-trained language models, such as BERT (Bidirectional Encoder Representations from Transformers) [13]. That work introduces a method for generating large-scale WSC-like examples—although not exactly WSC schemas, like in our work—by masking repeated occurrences of nouns (130 million examples, downscaled to 2.4 million). Their large developed dataset (MASKEDWIKI) indirectly shows that an automated way for the development of schemas will be helpful to the research community.

The importance of an automated way to develop Winograd schemas is not unrelated to WINOGRANDE, a large-scale dataset of 44 thousand examples collected via crowdsourcing [26]. To prevent the development of the same schemas, workers are primed by a randomly chosen topic from a WikiHow article. The idea of a randomly chosen topic shows the importance of Winventor’s categorization dataset.

6 Conclusion and Future Work

We have presented Winventor, a machine-driven approach for the development of Winograd schemas. Given that the development of schemas is hard and troublesome even for humans, Winventor comes into play as a schema replenishment mechanism, and as an assistant for the schema design process. Our experiments offer evidence that this can be achieved with two different approaches, the pure NLP approach, which provides a limited number of schemas, albeit with multiple variations, and a blended approach, which provides a bigger number of schemas, albeit one for every single sentence. In either case, the variability generally stems from which method is used. The evidence from this study suggests that systems like Winventor could act as teammates to further enhance the schema development process by humans. Winventor does not purport to replicate the thought process of humans in the development of schemas, as there is still no clear path yet on how this could be achieved. Future studies will have to identify other mechanisms to help humans and machines produce efficiently more schemas. Perhaps a better question generator, able to develop questions for more schema halves, would further help the schema development process. Furthermore, schemas could be offered to the crowd for further validation, leading to an interaction that would amplify human and machine intelligence by combining their complementary strengths.

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

1. Budukh, T.U.: An Intelligent Co-reference Resolver for Winograd Schema Sentences Containing Resolved Semantic Entities. Master’s thesis, Arizona State University (2013)
2. Chollet, F.: The future of deep learning. *Future* **8**, 2 (2017)
3. Chowdhury, G.G.: Natural language processing. *Ann. Rev. Inf. Sci. Technol.* **37**(1), 51–89 (2003)
4. Davis, E.: Winograd schemas and machine translation (2016)
5. Davis, E., Morgenstern, L., Ortiz, C.: Human tests of materials for the winograd schema challenge 2016 (2016)
6. Deepa, K.A., Deisy, C.: Statistical pair pruning towards target class in learning-based anaphora resolution for tamil. *Int. J. Adv. Intell. Paradigms* **9**(5–6), 437–463 (2017)
7. Heilman, M., Smith, N.A.: Question Generation via Overgenerating Transformations and Ranking. Carnegie-Mellon Univ Pittsburgh Pa Language Technologies Inst, Technical report (2009)
8. 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)
9. 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](http://dblp.uni-trier.de/db/conf/stairs/stairs2016.html?$$IsaakM16)
10. 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>
11. Isaak, N., Michael, L.: WinoFlexi: a crowdsourcing platform for the development of winograd schemas. In: Liu, J., Bailey, J. (eds.) AI 2019. LNCS (LNAI), vol. 11919, pp. 289–302. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-35288-2_24
12. 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>
13. 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](https://arxiv.org/abs/1905.06290) (2019)
14. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. *Nature* **521**(7553), 436–444 (2015)

15. Levesque, H., Davis, E., Morgenstern, L.: The winograd schema challenge. In: Thirteenth International Conference on the Principles of Knowledge Representation and Reasoning (2012)
16. Levesque, H.J.: On our best behaviour. *Artif. Intell.* **212**, 27–35 (2014)
17. Liu, Q., et al.: Probabilistic reasoning via deep learning: neural association models. arXiv preprint [arXiv:1603.07704](https://arxiv.org/abs/1603.07704) (2016)
18. Marcus, G.: Deep learning: a critical appraisal (2018)
19. Michael, L.: Machines with websense. In: Proceeding of 11th International Symposium on Logical Formalizations of Commonsense Reasoning (Commonsense 13) (2013)
20. Mitkov, R.: Robust pronoun resolution with limited knowledge. In: Proceedings of the 17th International conf. on Computational linguistics-Volume 2, pp. 869–875. Association for Computational Linguistics (1998)
21. Morgenstern, L., Davis, E., Ortiz, C.L.: Planning, executing, and evaluating the winograd schema challenge. *AI Mag.* **37**(1), 50–54 (2016)
22. Morgenstern, L., Ortiz, C.: The winograd schema challenge: evaluating progress in commonsense reasoning. In: Twenty-Seventh IAAI Conference (2015)
23. Mudge, R.: The design of a proofreading software service. In: Proceedings of the NAACL HLT 2010 Workshop on Computational Linguistics and Writing: Writing Processes and Authoring Aids, pp. 24–32. Association for Computational Linguistics (2010)
24. Peer, E., Samat, S., Brandimarte, L., Acquisti, A.: Beyond the Turk: An Empirical Comparison of Alternative Platforms for Crowdsourcing Online Research. In: Diehl, K., Carolyn Yoon, D. (eds.) *NA - Advances in Consumer Research*, vol. 43, pp. 18–22. MN, Association for Consumer Research (2015)
25. 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 2012, Association for Computational Linguistics, Stroudsburg, PA, USA (2012). <http://dl.acm.org/citation.cfm?id=2390948.2391032>
26. Sakaguchi, K., Bras, R.L., Bhagavatula, C., Choi, Y.: WINOGRANDE: an adversarial winograd schema challenge at scale. arXiv preprint [arXiv:1907.10641](https://arxiv.org/abs/1907.10641) (2019)
27. Schubert, L.K.: Semantic representation. In: Twenty-Ninth AAAI Conference on Artificial Intelligence (2015)
28. 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)
29. Stacey, J.: Text mining wikipedia for misspelled words (2011)
30. Wolfe, J.H.: Automatic question generation from text-an aid to independent study. In: Proceedings of the ACM SIGCSE-SIGCUE Technical Symposium on Computer Science and Education, pp. 104–112 (1976)