DRESDEN UNIVERSITY OF TECHNOLOGY

Research project

On Commonsense Domains within the Winograd Schema Challenge

Aneta Koleva
(Mat.-No.: 4734043)

Supervisors: Prof. Dr. Sebastian Rudolph   Dr. Emmanuelle Dietz

Dresden, June 10, 2019

Abstract

We investigate how to formalize commonsense reasoning in machines by analyzing The Winograd Schema Challenge (WSC). WSC is a complex coreference resolution task and requires applying knowledge on commonsense reasoning. In addition, to identify the main challenges of the WSC, we provide a survey of the different state-of-the-art approaches and their methods for addressing the WSC. After that, we perform an elaborate analysis of the WSC problems and identify six categories of commonsense knowledge required for their resolving. These categories might be helpful for further developments in the formal specification of characterizing meta-knowledge, that is required to solve the WSC.

Contents

1 Introduction

The Winograd Schema Challenge (WSC) was proposed by Levesque et al. [LDM12] as a new test in Artificial Intelligence (AI) and possibly as an alternative to the Turing test. The test captures a difficult pronoun disambiguation problem which is an easy task for humans, but remains a still unsolved challenge for computers. The WSC is formulated in such manner that it requires genuine understanding of real world situations and intelligence for solving it. For this reason it is argued that a computer that is able to solve the WSC with human-like accuracy must be able to perform human-like thinking [LDM12]. Different types of commonsense knowledge and reasoning are required to solve the Winograd Schema problems. Thus, the WSC has been proposed as a method for testing automated commonsense reasoning. This idea of designing a machine which could apply commonsense reasoning was first proposed by McCarthy [McC60]. He recognized that commonsense reasoning is a trait of intelligence and therefore tried to express it in formal logic so that it can be used for building a truly intelligent machine. Since then, formalizing commonsense reasoning has been an open challenge in the AI field.

Inspired by an example from Terry Winograd [AW72], the WSC corpus consists of sentences like the following:

S1: The city councilmen refused the demonstrators a permit because they feared violence.

S2: The city councilmen refused the demonstrators a permit because they advocated violence.

The task in the WSC is to identify the correct referent of the pronoun *they*. The difference between the sentences is the special word, in this case *feared/advocated*. Depending on this word, the referent of the pronoun *they* changes. In the first sentence it referes to *the city councilmen* and in the second to *the demonstrators*. This characteristic is what makes the WSC task a restricted form of the coreference resolution problem. The goal in the coreference resolution problem is to identify all the correct antecedents for a pronoun by relying on information about the gender and the number of the pronoun [Cry11]. In the WSC problems the candidates antecedents are always the same gender and number as the ambiguous pronoun, so this information is not sufficient for resolving the correct referent. In order to identify the referent of the ambiguous pronoun, one needs to have knowledge about the relations between the nouns, the verb phrase and the special word from the Winograd sentence.

Various approaches have been suggested for solving the WSC. More generally, the studies of tackling the WSC can be divided into two categories: Machine learning and Knowledge-based. The division is based on the main techniques applied for obtaining the correct answer. In the first category are the approaches which rely on machine learning and deep learning techniques ([RN12, LJL$^+$16b, TL18]). Indeed, approaches from this category ([TL18, RWC$^+$19] are the most recent ones to perform the best, with reported accuracy of over 70%. In the second category are the approaches which rely on knowledge-based systems ([SB16, ECT$^+$18, Sch14]). These require to have formally represented knowledge and procedures that should be able to reason with that knowledge. Many of the previously proposed approaches [SVAB15, Sch14, LJL$^+$16b, ECT$^+$18] recognized that answering the WSC correctly requires additional knowledge to what is in the given sentence. However, to the best of our knowledge few approaches have analyzed the knowledge in the available WSC sentences so far.

For humans, commonsense reasoning comes naturally because of the available background knowledge and because of the understanding about the surrounding world that humans have. In order for machines to be able to appear as if they would do commonsense reasoning, a huge amount of non-domain specific knowledge is needed [MMS$^+$02]. To address this issue, there have been attempts to develop repositories of common knowledge such as Cyc [Len95] and ConceptNet [LS04]. However, it is unclear whether these knowledge bases can ever be completed or if they contain all the necessary information for commonsense reasoning.

Motivated by the need for background knowledge we analyzed the sentences from the WSC corpus and identified six different categories of commonsense reasoning. We describe the process of annotating the WSC problems with these categories and we describe their characteristics.

The rest of this report is structured as follows: In the next chapter we introduce the WSC and explore approaches from both categories which have made significant contributions towards solving the WSC. In chapter 3 we discuss in more details the approach Sharma and Baral [SB18] and their proposed Reasoning Algorithm (RA). We then present the result of our analysis along with examples and description for the identified categories. In this section we also analyze the semantic graphs of the WSs which we analyzed thoroughly and formalized. After that we discuss all the challenges and findings that we came across during this work. Finally, we give concluding remarks and ideas for potential future work.

2 Background

In this chapter we introduce the structure of the WSC and the In section 2.1 we introduce the features of the WSC as proposed by Levesque et al. [LDM12]. Additionally we explain the dataset of Pronoun Disambiguation Problems (PDPS) which was proposed by [MJ15]. Finally, we discuss what makes the WSC so hard for solving and what are the main limitations of this challenge. In section 2.2 we analyze the different state-of-the-art approaches and give a summary of the results from their evaluation.

2.1 The Winograd Schema Challenge

The WSC was originally conceived in 2012 [LDM12] and since then it has caught the attention of many researchers from different areas. However, the first and so far only WSC competition was held in 2016 as part of IJCAI[1]. Before the competition, the rules for executing and evaluating the participants were presented by Morgenstern et al. [MDJ16]. The competition consisted of two rounds, one qualifying and one final, each with 60 questions. For the qualifying round, the questions were randomly chosen from the PDPs dataset and for the final round the questions were from the WSC dataset. In the qualifying round none of the participants achieved accuracy of 90%, which was the requirement for advancing to the final round.

2.1.1 Description

A Winograd Schema (WS) consists of three main parts:

1. A sentence that consists of:
    - Two noun phrases of the same semantic class and of the same gender
    - One ambiguous pronoun that could refer to either of the antecedent noun phrases
    - A special word such that when changed, the resolution of the pronoun is changed
2. A question, possibly containing the special word, asking about the referent

[1] http://ijcai-16.org/

of the ambiguous pronoun

3. Two possible answers corresponding to the noun phrases in the sentence

The WSC dataset currently consists of 285[2] such WSs. A typical example of a WS is the following one:
Example 2.1.1:

S: The trophy does not fit into the brown suitcase because it is too small.

Q: What is too small?

A: The suitcase/the trophy

Here, the special word is the adjective at the end of the sentence which can be *small* or *big*. The ambiguous pronoun is *it* and the two antecedents are *trophy* and *suitcase*. The adjective in the question depends on the chosen special word in the sentence. While it is obvious that answering the question in this example is easy for an adult English speaker, it still requires thinking to derive the correct answer. In order to identify the correct referent of *it*, one needs to use world knowledge about the size of objects, more specifically that a small object can fit into a large one, but not the other way around. Hence, if a machine is able to resolve the pronoun correctly, we can conclude that background knowledge was involved. Answering the question in this example is known as a problem of pronoun disambiguation. However, the sentences in the WSs are distinct in that one special word, which when changed causes the other noun phrase to be the correct referent. For example, consider the twin sentence for the WS given in *Example 2.1.1*:

S: The trophy does not fit into the brown suitcase because it is too large.

Q: What is too large?

The possible answers are the same as given in *Example 2.1.1*, but here the correct answer is *the trophy*. Having the special word in the sentences prevents the solver to rely on sentence structure and word order for finding the answer to the question. Moreover, Levesque et al. [LDM12] explain another feature of the WS sentences, which they call *Google proof*: The idea is to prevent finding the answer by using statistical learning on large corpora of text or by just typing the question and the answers into a search engine, such as Google. Here is an example of a sentence that is not Google proof:
Example 2.1.2:

S: Tom's books are full of mistakes. Some of them are quite [foolish/worthless].

Q: What are [foolish/worthless]?

A: The mistakes/the books.

[2] https://cs.nyu.edu/faculty/davise/papers/WinogradSchemas/WSCollection.xml

Example 2.1.2 has been provided as a failed example[3] because the term *foolish mistake* is commonly used and can be returned as an answer when querying Google. Additional feature is that resolving a WS should be easy for a human reader. For this reason, human performance is expected to be near 100%. Indeed, Bender [Ben15] conducted a large online experiment and reported 92% success rate for humans. For the execution and evaluation part of the WSC, two distinct datasets have been formulated and published by Morgenstern et al. [MDJ16]. The first dataset consists of pronoun disambiguation problems (PDPs)[4] that are taken from examples found in literature, biographies, essays and news or have been constructed by the organizers of the competition. The second dataset consists of WSs[5] which have been constructed by the challenge organizers. The following is an example of the PDPs:
Example 2.1.3:

Text: When they had eventually calmed down a bit, and had gotten home, Mr. Farley put the magic pebble in an iron safe. Some day they might want to use it, but really for now, what more could they wish for?

Snippet: to use it

Answers: magic pebble/safe

Along with the text, a snippet with the ambiguous pronoun that needs to be resolved and the possible answers are provided. As in *Example 2.1.3*, a PDP may consist of more than one sentence and it can also have more than two possible answers. Similar to resolving WSs, considerable use of commonsense reasoning is required when answering the PDPs. Since the structure of the PDPs is not always consistent, resolving these problems is more difficult than resolving WSs.

Having the two different sets allows the evaluation of the participating systems in the challenge to be done gradually. If a system successfully pases the questions from the first dataset, then it will be challenged with the WSs set. Morgensten et al. [MDJ16] argue that if a system can answer the problems from the PDP set correctly, then it will surely have success when answering the WSs set. Additionally, the PDPs are taken from various literature sources and may include different aspects of commonsense knowledge which would otherwise not be included in the created WSs.

2.1.2 Main challenges and limitations

We now analyze and discuss the main challenges that an approach for addressing the WSC must overcome in order to achieve good results. Furthermore, we will identify and discuss the main limitations of

[3] https://cs.nyu.edu/faculty/davise/papers/WinogradSchemas/WSfailed.html

[4] https://cs.nyu.edu/faculty/davise/papers/WinogradSchemas/PDPChallenge2016.xml

[5] https://cs.nyu.edu/faculty/davise/papers/WinogradSchemas/WSCollection.html

the WSC.

Finding the correct noun phrase for an ambiguous pronoun is a known problem in the Natural Language Processing (NLP) research field. Coreference resolution problem depends on different features, such as grammatical role, number, gender, syntactic structure and the distance between the referent and the pronoun. Although existing automatic systems rely on such grammatical features, they do not use world knowledge in the resolution process. Consequently, the Stanford resolver CoreNLP [LPC$^+$11] achieved accuracy of around 55% on the dataset provided by Rahman and Ng [RN12], which is just above the random baseline (50%).

From the examples given earlier, it is clear that background knowledge is required in order for a human reader to derive the correct answer. Since the sentences from both datasets, PDPs and WSs, are non-domain specific it is difficult to determine what kind of background knowledge is needed. Therefore, the first challenge imposed by the WSC is how to obtain knowledge for commonsense reasoning. The second is how to formalize this knowledge such that all important information are preserved. Lastly, how to reason on top of formalized knowledge is the third main challenge when trying to solve the WSC.

Regarding the limitations of the WSC, as a first and probably most important limitation is the number of available sentences. Because the process of creating new WS is difficult, the number of available sentences is very small. Moreover no training data is provided for any of the datasets which makes it hard to approach the task as a machine learning problem. Another limitation is the constraint posed by the language of the sentences. For some other languages, as explained by Davis [Dav16], the disambiguation of a pronoun for different genders may not be as clear as in English. Nevertheless, there are translation of 144 WSs in Japanese[6] and 107 WSs in French [7].

A third limitation is the lack of a more detailed evaluation protocol. In the proposal for the WSC competition the evaluation is measuring the accuracy of predicting the correct answers from both the PDPs and WSs datasets. However, Trichelair et al. [TEC$^+$18] argue that this evaluation does not measure whether commonsense reasoning was involved in the prediction of the correct answer. For this reason, they propose a different evaluation protocol which additionally includes measures of the accuracy on a changed subset and a consistency score. The changed subset contains WSs with switched antecedents such as in the following example.

Example 2.1.4:

- **S: Emma did not pass the ball to Janie although she saw that she was open.**
- **S': Janie did not pass the ball to Emma although she saw that she was open.**

[6] https://cs.nyu.edu/faculty/davise/papers/WinogradSchemas/collection_ja.pdf

[7] http://www.llf.cnrs.fr/winograd-fr

The consistency score is the percentage of predicted correct answer after the antecedents have changed. This new evaluation protocol gives a better understanding on how the models perform and opportunity to analyze in more details the achieved results.

2.2 Related Work

Since the publication of the WSC in 2012 many approaches for solving it have been proposed. For a structured overview, we categorized the existing approaches as follows: The first category are the approaches which employ machine learning and in particular deep learning techniques. The second category covers the approaches which use formalized background knowledge and rules. In this section we will discuss some of the proposed solutions from both categories which had notable contributions as well as approaches which achieved state-of-the-art results.

2.2.1 Machine learning approaches

The authors of the WSC were skeptical about the usage of learning methods being sufficient to resolve the WSC. Their concern was that these methods do not employ any reasoning which they believe is essential for solving the WSC. Nevertheless, there have been some implementations which rely on these methods and still achieve relatively good results.

One early contribution towards these approaches is the work by Rahman and Ng [RN12]. Along with their machine learning framework, an additional dataset was provided. This dataset was constructed by undergraduate students and consists of 943 WSs twin sentences divided into training (70%) and test (30%) sets. In the framework, a ranking-based approached was used. In other words, a Ranking Support Vector Machine (SVM) model [Joa02] is trained given pronoun and the two possible antecedents. The model ranks the antecedents and should assign a higher rank to the correct one. After the training phase, the same model is applied to the instances from the test set. Although the evaluation of the framework on the additional dataset showed that 73% of the test set was answered correctly, this approach was not tested on the original Winograd dataset. Nevertheless, the provided additional dataset will later serve for the training and testing of learning models from other proposals.

More recent proposals employ deep learning techniques. Such as the sequence ranking task suggested by Optiz and Frank [JA18]. The problem of choosing one of the possible antecedents for the missing pronoun is translated to a classification task which distinguishes a more preferred solution from a less preferred solution. As a first step, the ambiguous pronoun in the given sentence is substituted with the two possible solutions. For *Example 2.1.1* this would result in the following two sentences:

S1: The trophy does not fit into the brown suitcase because the trophy is too small.

S2: The trophy does not fit into the brown suitcase because the suitcase is too small.

After that, a neural model designed by Optiz and Frank [JA18], called a Siamese neural network, is specified. The model compares the two -representations of the input sentences and a ranking function that ranks them. For the training of the network, the additional dataset provided in [RN12] is used. Since this set is small and in order to prevent the model to memorize the noun phrase candidates, an anonymization technique is used. With this technique, during training the model omits the noun phrase candidates. This forces the model to focus to the rest of the sentence in order to identify a more general meaning. During the evaluation, the test on data which was anonymized had significantly better results than when anonymization was not applied. In contrast to [RN12], the model is tested on both the original Winograd dataset and the test set from the additional dataset. The model achieved 56% accuracy on the Winograd dataset and 63% accuracy on the additional dataset.

The implementation by Liu et al. [LJL$^+$16b] which competed and had the best results at the WSC held in 2016 also used deep neural network classifier to predict the answers. In the proposed framework, named Knowledge Enhanced Embeddings (KEE), three commonsense knowledge bases (ConceptNet [LS04], WordNet [Mil95] and CauseCom [LJL$^+$16a]) were used to extract commonsense knowledge. The extracted knowledge was then incorporated as knowledge constraint during the word embedding training process. An example of such knowledge constraint is the following semantic similarity inequality: *similarity(happy,glad)* $>$ *similarity(happy,sad)*. For the training process the skip-gram model [MCCD13] was used, which learns to predict the context given a target word. The KEE model was used for feature extraction from the PDP test problems. After feature extraction, two different solvers were employed for extracting the correct answer. The first solver relied on an unsupervised method for calculating the semantic similarity between the extracted embeddings and the antecedent candidates. The second solver used the extracted embeddings for supervised training of a neural network classifier. The experiments were initially conducted only on the PDP set and showed that the two solvers combined achieve 66.7%. During the WSC, the KEE model was trained on a small Wikipedia text corpus and the second solver was used, achieving 58.3% accuracy which was the best result at that time.

The method described by Trinh and Le [TL18], uses language models based on neural networks to capture commonsense knowledge. They use unsupervised learning to train neural networks on different large datasets. Based on what was learned during the training phase, the language models are able to assign probabilities to given text. The idea here is to reduce the coreference resolution problem to a decision which relies on probabilities. As a first step, a substitution as in [JA18], explained above, is done on the input sentence with the two possible antecedents. Next, the pre-trained language models assign one of the two different scores, full or partial, to these sentences. The full score is obtained by computing the probability of the substitution on the full sentence while the partial is the probability of the part of the sentence with the special word, conditioned on the substitution in the antecedent. The sentence with the higher probability is assumed to be the one with the correct answer. The conducted experiments showed that the models which assigned partial scoring were the most successful ones. According to the results of the experiments, the language models correctly resolved 70% of the PDP dataset and 63.7% of the WSC dataset. While these are very good results for the WSC, compared to the previous state-of-the-art [LJL$^+$16b], this method relies on learning with no inferential reasoning involved. Thanks to available computational power and the access to large datasets this method achieves very good results. At the same time, the training process of the networks is what makes this method very expensive. Moreover, it has been suggested as one possible extension of the WSC to add a requirement for the system to provide explanations of how the answers have been chosen [MJ15]. This would be a challenge for this particular approach and the other machine learning approaches since no explanation for a reasoning process is provided.

Recently, a similar method which relies on deep learning has been published by Radford et al. [RWC$^+$19]. This method uses an unsupervised learning on a language model which is trained using 1.5 billion parameters and a dataset of 8 million web pages. In order to ensure the quality in the training dataset, a new web scrape was created which scrapes data only from pages with content created and modified by humans, for example blogs. The model, named GPT-2, uses Transformer based architecture [VSP$^+$17] which is an encoder-decoder structure with self-attention layers. Although this method is controversial because the implementation is not publicly available, the authors claim that their language model outperforms the current state-of-the-art in different NLP tasks (reading comprehension, summarization of text, translation). For solving the WSC, they used the GPT-2 model in the same setting as Trinh and Le [TL18] and also achieved better results when applying the partial scoring. The reported accuracy of 70.70% on the WSC corpus is the current state-of-art result.

2.2.2 Knowledge-based approaches

The authors of the WSC are from the Knowledge Representation and Reasoning area, so it is not surprising that they suggest the use of knowledge-based systems as a possible way of addressing the WSC. At the core of the knowledge-based approaches lies the idea of formulating rules which can be used for resolving the missing pronoun. To be able to apply the rules during a reasoning process, structured background knowledge is required.

One of the earliest proposals that rely on this methodology was by Schüller [Sch14] in which Relevance

Theory (RT) [WS02] is combined with Knowledge Graphs (KG). Inspired by Schank's model for Knowledge Representation [Sch72], Schüller proposed a framework for combining nodes of KG and reasoning on the resulting graph. In this framework a KG data structure based on the domains for labeling the nodes and labeling the dependencies in the graph is defined. Additionally two sets of constraints and conditions for the KG structure are defined in order to enforce certain linguistic and structural properties. An example of such condition is that all nodes in the graph are of the same type. This data structure is then used for representing the input sentence and the background knowledge. The input sentence represented as a KG and a background KG related to it is activated, after which the two are joined in a third KG. Finally, applying concepts from RT during the reasoning process over the third graph, a resulting graph is extracted which represents the correct solution for the input sentence. Schüller [Sch14] presents the evaluation on 4 WSs. The downside of this approach is that the graphs for the input sentence and for the knowledge have been manually constructed which limits the possibility for evaluation and it misses the core point of the challenge. However, the idea to represent the given Winograd sentence as a dependency graph served later as a starting point to other approaches such as [SVAB15].

In Sharma et al. [SVAB15] the authors recognized two different types of commonsense knowledge which are needed in order to resolve the pronoun in the input sentence. 71 WSs have been divided into two categories: the first category is called the Direct Causal Events and the second one is called Causal Attribute. Furthermore, a non domain-specific semantic parser is developed, called KParser[8]. This parser uses different NLP techniques, such as syntactic dependency parsing, sense disambiguation and discourse parsing for preprocessing the input sentence and produces a semantic graph representation of the sentence. In the next step, a set of queries based on the concepts in the sentences and in the question is created, which is later used to search a large corpus of text. The goal of the search is to automatically extract sentences with relevant commonsense knowledge for the input sentence and represent them as a semantic graph. Finally, different reasoning processes[9] are applied to the sentences from the two categories and by comparing the graphs of the input sentence and the extracted sentences, the predicted answer is found. During the evaluation of this approach, 53 sentences were answered with 4 of them answered incorrectly. Although in [SVAB15], the focus is on a subset of sentences, the results from the evaluation are promising. This leaves open the possibility to identify more types of commonsense reasoning which would help in the extraction of the background knowledge and the reasoning process.

The work by Sharma and Baral [SB18] follows this direction. Although this approach is currently under review and still not published, we decided to consider it in more detail because it seems to be a promising knowledge-based approach. In contrast to the two types of reasoning identified by Sharma

[8] www.kparser.org

[9] The difference is in the predicates that extract nodes from the semantic graphs.

et. al [SVAB15], Sharma and Baral [SB18] present twelve different knowledge types which capture the entire corpus of the WSC. How these twelve knowledge types were identified and which are they will be explained in the next chapter. Another contribution of the work is the provided reasoning algorithm which will also be explained in the next chapter.
Following are the steps of this approach:

1. The given Winograd sentence and question are represented as semantic graphs and encoded as ASP rules.

2. The extraction of the background knowledge is the same as in [SVAB15]. Sentences similar to the input WS are retrieved and the knowledge is extracted in the same graph format as one of the identified knowledge types and encoded in ASP.

3. The reasoning algorithm matches the graph from the input sentence with the graph from the background knowledge, which again results in a graph.

4. This resulting graph is merged with the graph representation of the question.

5. An answer is retrieved by matching the node which represents the ambiguous pronoun from the question graph with the node which represents an entity from the resulting graph.

The evaluation is done separately for the WS and for the additional dataset from Rahman and Ng [RN12]. The knowledge extraction algorithm retrieved relevant knowledge for only 100 WSs, so the reasoning algorithm was evaluated only over those WSs and all of them were answered correctly. For the additional dataset, 138 sentences which are covered by the identified knowledge types were evaluated. The needed background knowledge was encoded manually. In this case, the reasoning algorithm answered correctly 80.45%, whereas the remaining 19% were answered wrong because of a parsing error and not a reasoning one. The main drawback of this approach is the missing background knowledge which limits the evaluation part.

A very recent approach, which achieves state-of-the-art results, is a knowledge hunting framework proposed by Emami et al. [ECT$^+$18]. The focus here is on solving all the instances without relying on any existing knowledge bases or statistical methods for capturing the commonsense knowledge. To this end they have developed a framework which is divided into four stages.

1. Using syntactic parsing of the given sentence the antecedent candidates, the verb which connects them as well as the verb phrase which contains the missing pronoun are identified.

2. Based on the identified elements from the first step, queries are generated. In addition, a semantic similarity algorithm is developed which filters out elements from the generated query sets based on their relevance to other words. An example of a query set for *Example 2.1.1* is the following:

{"doesn't fit into", "brown", "fit"}

3. The results from the search engines are processed and only sentences with a structure similar to the input sentence are kept.

4. Finally, the retrieved sentences are scored according to a scoring system designed by the authors. The scores are assigned to both possible antecedents and the one with the higher score is assumed to be the correct one.

The experiments done during the evaluation of this framework show the a correct answer is given for 119 WSs. Furthermore, the precision, recall and F1 score is calculated and compared with results from previous approaches ([SB16, LJL^{+}17]) and the knowledge-hunting framework outperformed them, achieve a higher F1 score and higher recall.

Table 2.1 presents all the previously analyzed approaches. For each approach the size of the evaluation set and the size of correctly answered problems is represented with the number of sentences and the percentage this number represents for the used evaluation dataset. If the evaluation was not provided for any of the datasets, then there is NA for Not Available. In the last column are some remarks for each of the approaches. It can be observed that almost all of the approaches are evaluated on different or on smaller datasets. For the machine learning approaches the main disadvantage is the small number of available problems. In the case of the knowledge based approaches, the general drawback is that the directly the test set is analyzed and based on these observations the models are build. Moreover, none of these approaches achieves an accuracy close to 90%, which means there is room for improvement for both machine learning and knowledge-based approaches.

Technique	**PDPs** Size | Correct	**WSC** *additional dataset Size | Correct	**Remarks**
Supervised ranking SVM model [RN12]	NA	NA 282* - 30% | 205* - 73%	-provided additional dataset set -no evaluation on WSC dataset
Classification task with NN [JA18]	NA	282 - 100% | 157 - 56% 282* - 30% | 177* - 63%	-first to use substitution of the pronoun with the antecedents
Knowledge Enhanced Embeddings (KEE) [LJL$^+$16b]	60-100% | 40 - 66.7%	NA NA	-best results in the 2016 WSC competition
Google's language model [TL18]	60-100% | 42 - 70%	273 - 100% | 173 - 63.7% NA	-no reasoning involved in the discovery of the correct answer -state-of-the-art for PDPs
OpenAI language model [RWC$^+$19]	NA	273 - 100% | 193 - 70.70% NA	-current state-of-the-art for WSC -requires a lot of data for training -results are not reproducible
Graphs with Relevance theory [Sch14]	NA	4 - 2.6% | 4 - 100% NA	-manual construction of graphs -first representation of WS as dependency graph
2 identified categories [SVAB15]	NA	71 -25% | 49 - 69% NA	-first attempt of identifying commonsense knowledge types -developed the KParser
Semantic relations categories [SB18]	NA	100 - 34%| 100 - 100% 138* - 14%| 111 - 80%	-provided Reasoning Algorithm -identified 12 commonsense types which capture the entire WSC
Knowledge hunting framework [ECT$^+$18]	NA	273 - 100%| 119 - 43.5% NA	-refined query generation -developed an algorithm for scoring the retrieved sentences

Table 2.1: Summary of the analyzed approaches

3 An approach to Commonsense Knowledge Types Identification and Reasoning for the WSC

Intrigued by the remarkably achieved accuracy of the Sharma and Baral [SB18] approach, we now describe this work in more details. In the previous chapter, in section 2.2 we outlined the steps of the approach. Here, we first focus on the process of identification of the presented commonsense knowledge types. Then, we discuss the proposed reasoning algorithm together with the worked out example. Finally, we give our observations and comments about this approach.

3.1 Identification of Commonsense Knowledge Types

The first step of the Sharma and Baral [SB18] approach is to represent the input Winograd sentence and question as semantic graphs. To achieve this, the KParser from [SVAB15] is used. To illustrate the results of the semantic parsing, we now analyze the graphs for the sentence and question from *Example 3.1.1* which are shown in Figure 3.1 and Figure 3.2.

Example 3.1.1

S: The man could not lift his son because he was weak.

Q: Who was weak?

In these graphs, the different colors of the nodes indicate the predefined class of nodes to which they belong. Nodes with red color represent events which correspond to the verbs in the sentence. Nodes with blue color represent entities and qualities of entities, and the nodes with gray color represent conceptual classes. The labels on the directed edges are the semantic relations between the different nodes in the graph. The number next to a word refers to the position of the word in the sentence.

For the representation of the question, one additional conceptual class labeled as "q" is introduced. In this conceptual class are all nodes which represent the question words from the WSC problems, such as: *Who, Which and What*.

From the observation of these two graphs, Sharma and Baral [SB18] come to conclusion that the missing

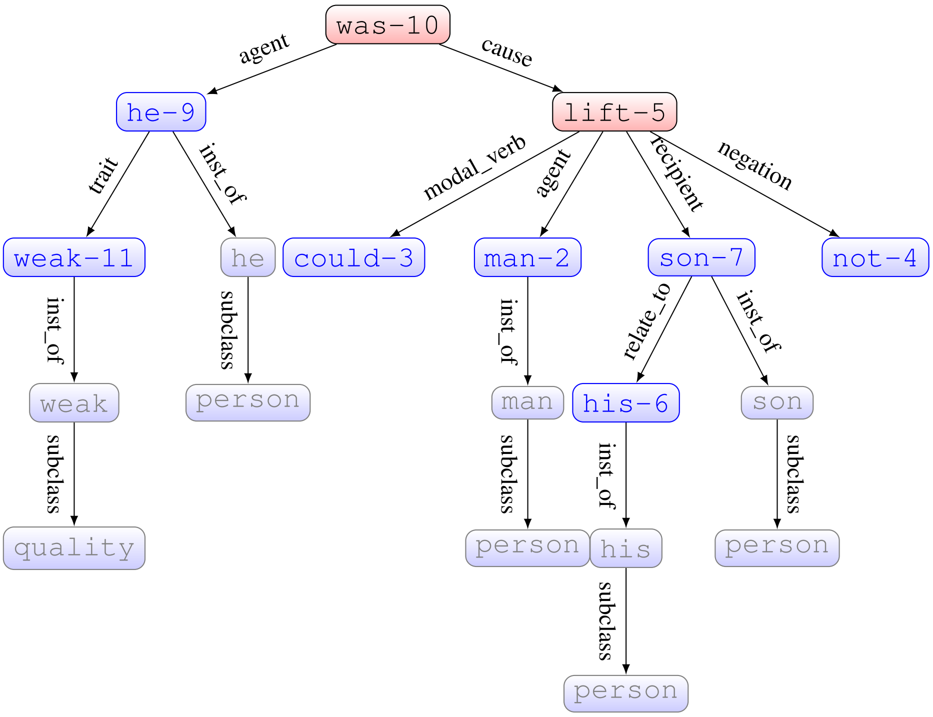

Figure 3.1: “The man couldn’t lift his son because he was so weak."

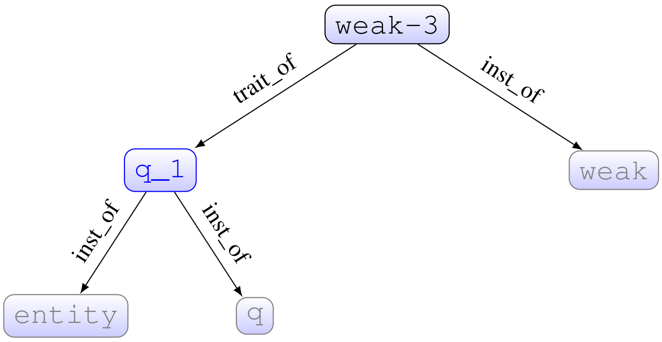

Figure 3.2: “Who was weak?"

knowledge required in order to answer the question, must connect the trait of an entity being weak with its inability to lift. In Figure 3 is the representation of this knowledge as shown in [SB18].

By applying this observation to all problems from the WSC corpus, Sharma and Baral [SB18] identified 12 different knowledge types. From these, the first 10 knowledge types share the same structure as they are all based on different interactions between actions and properties. Each of the 10 knowledge types consists of three parts: the first and third part are sentences containing entities, properties or actions, whereas the second part is a semantic relation (*causes, prevents* or *followed by*) which connects them. The last two knowledge types have different structure because one of them requires multiple knowledge and the other one is based on the conditional likelihood of a previous event. For the representation of the knowledge type shown in Figure 1.3, the first part is *weak y*, the third part is *y lifts* and the semantic relation is *prevents*. The presented identified knowledge types are the following:

> Property *prevents* Action, Action1 *prevents* Action2, Action1 *causes* Action2, Property *causes* Action, Action *causes* Property, Property1 *causes* Property2, Action1 *followed by* Action2, Action *followed by* Property, Property *followed by* Action, Co-existing Action(s) and Property(s), Statement1 *is more likely than* Statement2, Multiple Knowledge.

3.2 Reasoning Algorithm

We will now explain the proposed Reasoning Algorithm (RA) which was used for answering the WSC problems by using the previously explained knowledge types. The RA together with a running example is available online[1]. The code for the RA algorithm can be found in AppendixA.

The RA is based on Answer Set Programming (ASP) [Lif08]. The RA is split into two phases, each including several steps. Each step, results in creating new ASP rules. In the second phase, the graph representation of the WSs question is matched with the merged representation from the first phase. With this matching, the possible answer should be retrieved.

- **Phase 1: Generating merged representation**
  In this phase, the graph representation of the Winograd sentence and the background knowledge are analyzed and the following information from both is extracted and merged.

  1. All the nodes which are an instance of a defined class of nodes (for example: person, motion, description) in the KParser are identified as constant nodes.
  2. Nodes with parents as constant nodes are identified.

[1] https://drive.google.com/file/d/1WN0T98HaMFhWEEIH-3AlWoIPxAdFYlT_/view

3. Nodes with children as constant nodes are identified.
4. The cross domain siblings in both representations are identified. These are constant nodes which appear in both graphs and they both are instances of nodes of the same type. Furthermore the cross domain siblings ave the same number of child/parent nodes which are connected through the same semantic relations.
5. A merged representation is generated

- **Phase 2: Extracting possible answers**

  In this phase, the graph representation of the WSs question is projected on the merged representation from the previous step. The goal is to retrieve the nodes which are cross domain clones of the unknown nodes (q) in the graph representation of the question.

  1. Constant nodes are identified from both graphs.
  2. Nodes with children/parent as constant node are identified.
  3. Cross domain siblings of the nodes from the question graph are identified.
  4. Using the previous nodes, the cross domain clones of the nodes from the question graph are identified.

The reasoning algorithm can reason over 10 of the 12 identified categories.

3.3 Observations

The encoding the example available in the supplementary document resulted in correct answer. After that we tried a few different sentences and the reasoning algorithm worked correctly. When the rule that defines the knowledge type is commented out, again a correct answer is retrieved. Our conclusion is that the RA does not even use this rule in the reasoning process. Rather it relies on hard coded information about the agent and the recipient of the action.

4 Categorization of Commonsense Knowledge

Motivated by the identified flaw in the Reasoning Algorithm explained in Section 3?, we present a different conceptual analysis which focuses on the categorization of the problems from the WSC corpus. Similarly as in Sharma et al. [SVAB15] and in Sharma and Baral [SB18], we identify different categories of knowledge which are needed in order to answer the questions from the WSs. A detailed comparison between the categories from the previous approaches and our categories will follow in the discussion section. We identified six categories with which all problems from the WSC corpus were categorized.

4.1 Commonsense domains

For solving the problems from the WSC corpus, different types of commonsense knowledge need to be applied. As discussed before, for a system to be able to do commonsense reasoning, background knowledge from different domains is required. In order to identify these domains of knowledge we approached the WSC corpus inductively and we identified six categories.Two annotators thoroughly analyzed the entire WSC corpus[1] and agreed on six different categories of knowledge. Five of these categories are associated with knowledge from a specific domain, whereas the sixth category requires knowledge from multiple domains. In a second process, all WSs were annotated with these previously identified categories. The annotation was based on the additional knowledge that is applied in order to answer the question. To decide which sentence belongs to which category, we rely on the context of the sentence and on the adjective or the special word. The identified categories are specific to the WSC corpus and they are related to what was discovered during the analysis done by the annotators. We will now explain the six categories in more detail.

Physical To resolve the WSs from this category additional knowledge about some physical feature which is a trait of the ambiguous pronoun is needed.

[1] https://cs.nyu.edu/faculty/davise/papers/WinogradSchemas/WSCollection.html

Category	**Example**
1. Physical	**S:** John couldn't see the stage with Billy in front of him because he is so **[short/tall]**. **Q:** Who is so [short/tall]?
2. Emotional	**S:** Frank felt **[vindicated/crushed]** when his longtime rival Bill revealed that he was the winner of the competition. **Q:** Who was the winner of the competition?
3. Interactions	**S:** Joan made sure to thank Susan for all the help she had **[given/received]**. **Q:** Who had [given/received] help?
4. Comparison	**S:** Joe's uncle can still beat him at tennis, even though he is 30 years **[older/younger]**. **Q:** Who is [older/younger]?
5. Causal	**S:** Pete envies Martin **[because/although]** he is very successful. **Q:** Who is very successful?
6. Multiple knowledge	**S:** Sam and Amy are passionately in love, but Amy's parents are unhappy about it, because they are **[snobs/fifteen]**. **Q:** Who are [snobs/fifteen]?

Table 4.1: Types of knowledge

Emotional This category captures all WSs that include some emotion. In other words, resolving these WSs requires some knowledge about emotions and their characteristics.

Interactions These WSs are related to abstract forms of interaction between the subject and the object in the sentence. In contrast to the WSs from the *Physical* and *Emotional*categories, resolving these WSs does not include knowledge about emotions or characteristics of physical objects.

Comparison In this category are all the WSs where the special words are usually antonyms. The comparison is mostly between the subject and the object in the sentence.

Causal Although many of the WSs express causality, we categorize here only those for which the causality is expressed more explicitly. These sentences contain linking words such as *so* and *because*.

Multiple knowledge Lastly, in this category are all the WSs which do not belong to any of the other categories. For resolving these WSs knowledge from different domains is required.

The annotation of the WSs in some cases was very straightforward and easy to decide. For instance, if we analyze the example for the *Emotions* category in Table 4.1 the special word can be *vindicated* or *crushed*, which we know both are emotions. In addition, the verb *felt*, which is a past form of the verb *to feel*, is an emotion indicator. To answer the question “Who was the winner of the competition?", a deeper understanding of both emotions is required. Furthermore, knowledge about which emotions are related to winning is needed. Overall, the reasoning process for answering the questions in this category involves knowledge about emotions.

In comparison, the annotation of the WSs from the *Interactions* category was challenging. The WSs in this category are about inter-human relationships, often between the subject and the object in the sentence. To solve a WS from this category, applying knowledge about more abstract activities is necessary. Some examples for such activities are: to cooperate, to promise, to convince, to refuse, etc. That is to say, when resolving these WSs there is no reasoning about emotions, comparison or exchange of any physical objects involved. The sentence given as an example from this category in Table 4.1 is annotated with category *Interactions* because of the special word. *Giving/Receiving* help is an abstract form of interaction between two different entities. This activity cannot be categorized as physical, emotional, causal or as comparison.

Additionally, these categories can be divided further. Since in the *Physical* category we have the highest number of WSs, we will analyze this category in more details. The WSs from this category, contain different physical features which characterize the ambiguous pronoun. After carefully analyzing these features we identified five subcategories: *size, height/width, weight, space* and *time*.

In the *size* subcategory is the WS from Example *Example 2.1.1*, which was discussed in the Background section. The activity in the sentence from this WS depends on the physical trait of the pronoun it.

S: The trophy does not fit into the brown suitcase because it is too small.

This physical trait is also the special word which can be *small* or *large*. For resolving these WSs, knowledge about the characteristics of an object of a certain size is required. For Example *Example 2.1.1*, this knowledge should capture that a small object cannot fit into a large one, while a large one can fit small object.

Similar to this subcategory are the *height/width* and the *weight* subcategories. The WSs in these subcategories have a special word which characterize the ambiguous pronoun with certain height, width or weight such as *short/tall*, *wide/narrow*. Since these are all similar traits, we assume that their characterizations can be formalized in a similar way.

For solving the WSs from the *space* subcategory, spatial reasoning and in some cases spatial-temporal reasoning is needed. Since the knowledge required for resolving the WSs from this subcategory has to include rules for navigating and understanding time and space, we consider the knowledge needed for this category to be the hardest one for formalization among the subcategories of the *Physical* category. For example, consider the following WS:

Example 3.1

S: Tom threw his schoolbag down to Ray after he reached the [top/bottom] of the stairs.

Q: Who reached the [top/bottom] of the stairs?

A: Tom/Ray.

The sentence in this example contains information about a specific location in space which can be *top of the stairs* or *bottom of the stairs*. In addition, it states something about an event which happened *after* a certain event. Therefore, answering the question from this example requires applying reasoning about both space and time. Other WSs are simpler, where only knowledge about space is needed. These WSs have a special words such as: *above/bellow, in/out*.

The WSs from the *time* subcategory require temporal reasoning. Formalization of knowledge about change over time and chronology of events is needed for solving these WSs. For annotating these WSs we relied on the context and we identified events which happened at different points in time.

Let us consider the following WS which is example from the weight subcategory :

Example 3.2

S: The man could not lift his son because he was so [weak/heavy].

Q: Who was [weak/heavy]?

A: The man/the son.

While heavy is a description of the weight of an object/a person, weak on the other hand is a description of the strength. Both special words represent a physical trait of the ambiguous pronoun, but they are not from the same subcategory.

Evaluation To evaluate the agreement of the annotation we used Cohen's kappa statistic [Coh60], which is a measure for an inter-rater agreement. Namely, Cohen's kappa measures the agreement between two annotators regarding the annotation of N items with C mutually exclusive categories. It is defined as follows:

$$\kappa = (p_o - p_e)/(1 - p_e) \tag{4.1}$$

Category	**Annotator 1**	**Annotator 2**
Physical	36	39
Emotions	7	9
Interactions	44	24
Comparison	19	26
Causal	16	18
Multiple knowledge	28	34

Table 4.2: Annotation results

where p_o is the observed accuracy among the annotators and p_e is the expected accuracy when both annotators assign categories randomly. Each of the two annotators analyzed all the problems from the WSC corpus and annotated them with one of the identified six categories. For the calculation of the Cohen's kappa score, we used the sklearn metrics package[2] that includes an implementation of Cohen's kappa.

Results The calculated Cohen's kappa score for the annotation of the WSs is 0.66. According to the description for relative strength of agreement from Landis and Koch [LGK77], this score is interpreted as substantial agreement. In our case this means, that both annotators largely agree on the annotation of WSs with the provided six categories.

The numbers of WSs in each category, as annotated by the two annotators, is shown in Table 4.2. From the table we can be observed that both annotators agree on high number of WSs being in the *Physical* category. There is also strong agreement for both the *Emotional* and the *Causal* category. However, there is a significant difference of 20 WSs in the annotation for the *Interactions* category. Lastly, from both annotators, around 30 WSs were annotated with the category *Multiple knowledge*.

4.2 Semantic representation and ASP encoding

As the *Physical* category was the one on which both annotators agreed on, we will further investigate WSs that belong to this category. For extracting the semantic graphs we used the KParser[3]. Two examples of such semantic graphs[4], are shown in Figure 4.1 and 4.2.

[2] https://scikit-learn.org/stable/modules/generated/sklearn.metrics.cohen_kappa_score

[3] kparser.org

[4] Slightly simplified such that only relevant information is represented.

We can observe that the semantic representations of the sentences are indeed very similar. Figure 4.1 represents the semantic graph for the sentence:

S: The trophy does not fit into the brown suitcase because it is too small.

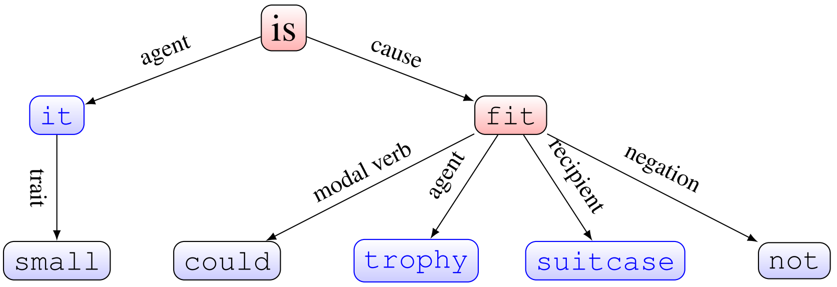

Figure 4.1: “The trophy doesn’t fit into the brown suitcase because it’s too small."

Figure 4.2 shows the semantic graph for the sentence:

S: The man couldn’t lift his son because he was so weak.

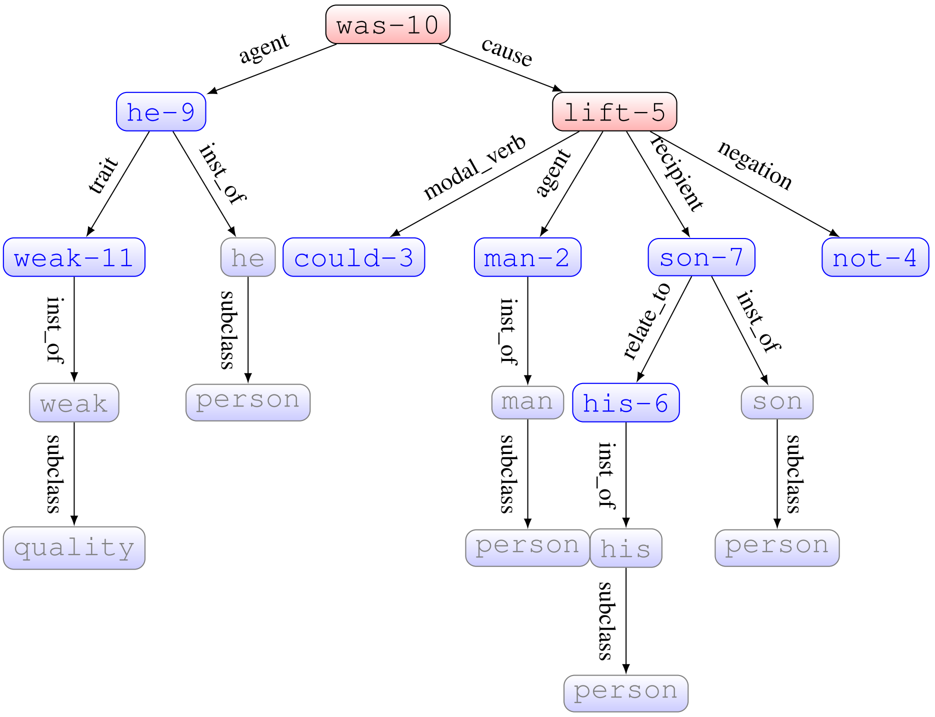

Figure 4.2: “The man couldn’t lift his son because he was so weak."

In both representations, the main action (*fit, lift*) is connected with the agent and the recipient of the action as well as with the modifiers of the action. Additionally, the ambiguous pronoun (*it, he*), associated with a physical trait (*small, weak*), is represented as the cause of the main action. The KParser has an option to show *Coreference*, that is to find the coreferents for the pronouns in the given sentence. However, almost for all of the WS sentences that we analyzed, as coreferent to the ambiguous pronoun was selected the *agent* of the main action, which was not correct.

We used these semantic graph representations as guidance for encoding the sentences, the questions and the background knowledge with ASP rules.

The following rules represent the ASP rules for the knowledge obtained from the sentence represented with the semantic graph in Figure 4.2.

sent(he, instance_of, entity)
sent(man, instance_of, entity)
sent(son, instance_of, entity)
sent(lift, instance_of, action_lift)
sent(lift, agent, man)
sent(lift, recipient, son)
sent(weak, trait_of, he)
not sent(lift, modifier, could)

The predicate **sent** is short for **sentence** which indicates that this is the origin of these facts. Similarly, we use **know** for predicates from the background knowledge and **quest** for predicates from the question. In the same way the rules for the other semantic graph are defined. We also formalized the background knowledge. For this purpose we specified two ASP rules which characterized the physical traits from the examples.

- **know(weak, trait_of, X) :- know(lift, agent, X), not know(lift, modifier, could)**
- **know(small, trait_of, X) :- know(fit, recipient, X), not know(fit, modifier, could)**

Lastly, the ASP encoding of the question for the sentence in Figure 4.2 is the following:
quest(weak, trait_of, Q)
quest(Q, instance_of, entity)

The predicates in the ASP rules for the question include a variable **Q** which is a substitution for the question word in the sentence, in this case for **Who**.

Since developing an reasoning algorithm is not a central part of our work, we used the Reasoning Algorithm provided from Sharma and Baral [SB18] to test the two example sentences from the *Physical* category. The algorithm is accessible online as a supplementary document[5] for the paper.

The returned answer for both sentences was the correct answer. When considered the twin sentence with the switched special word, the Reasoning Algorithm again retrieved the correct answer. Furthermore,

[5] https://drive.google.com/file/d/1WN0T98HaMFhWEEIH-3AlWoIPxAdFYlT_/view

when the rule describing the background knowledge is removed, there is no answer was retrieved.

5 Discussion

An initial objective of this project was to identify the main challenge that needs to be resolved in order to tackle the WSC. From studying the existing approaches, we recognized the need for having a formal representation of a specific background knowledge during the reasoning process in the knowledge-based approaches. To this end, we proposed a categorization of the sentences for which similar knowledge is required for sentences belonging in the same category. We assume that for the different categories there can be different formalized rules which can be used during the reasoning process.

The result of the evaluation of the WSs annotation indicate that all of the identified categories were recognized by the two annotators. Although not a large number of WSs are in the *Emotions* and *Causal* categories, there is a strong agreement between the annotators about the WSs belonging in these two categories. For the *Causal* category, this indicates that even though many of the WSs seem causal, it is not so difficult to discriminate the sentences in which the cause and effect are explicit. The *Physical* category contains largest number of WSs for which both annotators agree. Indeed, the observation of the semantic graphs for WSs from this category showed that there is a similarity in the knowledge that is represented with these WSs. As discussed earlier, the identification of the WSs which belong in the *Interactions* category was quite difficult. This is clearly shown in the large difference between the annotators in the annotation results. While one annotator placed 44 WSs in this category, the other annotator 20 of those 44 WSs distributed in all other five categories. Therefore, we can conclude that the identified categories are not exclusive, i.e. some of the WSs can belong to more than one category.

The identified categories differ from the ones presented by Sharma and Baral [SB18] in the way they are defined. We rely on the context of the WS sentences, whereas in [SB18] the categorization is based on the structure of the sentence. Namely, we analyzed the entire WSC corpus and identified what is the commonsense knowledge that is needed for answering the question and from which category it would be. In Sharma and Baral [SB18] the categories are identified using a more general approach that is, they all have the same structure *X prevents/follows/causes Y*.

One unanticipated finding was that the rule-based Reasoning Algorithm given in Sharma and Baral [SB18] does not work as expected. In the supplementary document[1] for the paper the ASP code for

[1] https://drive.google.com/file/d/1WN0T98HaMFhWEEIH-3AlWoIPxAdFYlT_/view

the Reasoning Algorithm together with an example is given. The example is the WS from Example 3.2. After encoding this example and assuring the correct answer, we tried to test it with several different WSs. For this, changes in the sentence and background knowledge representation were made and the correct answer was retrieved in all tried examples. What came as a surprise was that the correct answer was retrieved even when *the key rule* was removed. The key rule is the one which formalizes the knowledge type of the WSs. As explained in Sharma and Baral [SB18], for the given example this rule is what characterizes the knowledge type *Property prevents Action*. In the example given in the supplementary document this rule is encoded as: **has_k(weak_1,prevents,lifts_5)**. Intrigued by this result, we analyzed more closely the Reasoning Algorithm. What we discovered is that the answer returned depends on whether in the background knowledge there is information about the *agent* or the *recipient* of the action.

Since the Reasoning Algorithm is well encoded, we decided to change only the encoding of the background knowledge so it would capture the characteristics to one of our identified categories. Having these rule in the background knowledge and then running the Reasoning Algorithm returns the correct answer. In contrast to the rule from the example in Sharma and Baral [SB18], when the rule that formalizes the physical trait is removed from the background knowledge, no answer is retrieved.

A good semantic graph representation of the WS is of high importance for correctly formalizing the available knowledge. To understand better the choice of *agent* and the *recipient* for the WS sentences, we analyzed more closely the work of the KParser. Although in most of the sentences that we parsed a correct graph representation was returned, in some of them there were many inconsistency. For example consider the following sentence:

S: Joan made sure to thank Susan for all the help she had given.

When parsed with the KParser, the semantic graph representation of this sentence consists of three disconnected graphs. Moreover, in the case when there are two consequent sentences in one WS the result from the KParser is too complicated for analyzing, let alone for formalizing in ASP.

6 Conclusion and Future Work

Conclusion In this report we presented a detail analysis of the WSC. We introduced the main components of a WS and identified the need for additional, background knowledge as a main challenge for correctly resolving one. We presented an extensive literature review on the existing different proposals and their contributions towards solving the WSC. Looking into the different approaches, we highlighted that neither the Machine learning based nor the Knowledge-based approaches alone are sufficient for achieving high accuracy on the WSC. Therefore, with the intention to improve the process of extracting and formalizing relevant knowledge, we analyzed in more details the WSC problems and identified different categories of WSs. The categories that we presented can enhance the work on both the Machine learning and the Knowledge-based approaches or can be used as a bridge between them. We found the process of identification of the categories and deciding the right category for the WSs to be quite challenging. It is not surprising that so far too little attention has been paid to the content of the WSs. In addition, we used an existing reasoning algorithm, which when adapted as for one of the identified categories retrieves correct answers for WSs. It is important to note that our work presented here is an analysis of the WSC and not an attempt towards solving it. Moreover, due to the limited number of WSC problems, the presented categories were identified in a “backward-engineering" manner, that is by considering directly the test set.

Future Work To support the assumption that identifying the category of a WS can improve the extraction of relevant additional knowledge, the characteristics of the meta-concepts from each category ought to be formalized. Analyzing the semantic graphs for the WS sentences could be a good starting point for recognizing these characteristics. Afterwards, these formalized concepts need to be implemented and tested in an appropriate reasoning algorithm. Another interesting extension would be to explore an implementation of knowledge-enhanced neural networks [MPC18] with commonsense knowledge from the different categories and test their performance. Finally, since the existing semantic parsers often provide incomplete and disconnected graphs, it would be worth considering different representations of the input Winograd sentence and question.

Part of the aim of this project is to develop a hypothesis which can improve the process of extracting relevant knowledge for the reasoning process. Recently, as proposed by Ma et al. [MPC18], applying knowledge injection during the training of deep neural networks can lead to improvement of the result of the neural network. Using commonsense knowledge databases, different neural networks can be pre-trained with knowledge from a specific category. After analyzing the Winograd input sentence, a network trained with the knowledge from the identified category can be used in the process of extracting relevant background knowledge. Additionally, formalized rules describing the characteristics of a category can be fed to neural networks before training [RDG18]. In this way, the rules would be a guidance for the network during the training phase. Not only could this speed up the training of the network by reducing the amount of required data, but might also support explanations for the predicted answer. An approach that exploits the advantage of formalized rules, capturing characteristics of different areas of background knowledge, can possibly merge together the strengths of both categories of approaches, Machine Learning and Knowledge-Based Systems.

References

[AW72] Terry Allen Winograd. Understanding natural language. *Cognitive Psychology*, 3:1–191, 01 1972.

[Ben15] David Bender. Establishing a human baseline for the winograd schema challenge. In *Proceedings of the 26th Modern AI and Cognitive Science Conference 2015, Greensboro, NC, USA, April 25-26, 2015.*, pages 39–45, 2015.

[Coh60] Jacob Cohen. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1):37–46, 1960.

[Cry11] David Crystal. A dictionary of linguistics and phonetics, sixth edition. *The Modern Language Journal*, 76:25–26, 01 2011.

[Dav90] Ernest Davis. *Representations of Commonsense Knowledge*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1990.

[Dav16] Ernest Davis. Winograd schemas and machine translation. *CoRR*, abs/1608.01884, 2016.

[DM15] Ernest Davis and Gary Marcus. Commonsense reasoning and commonsense knowledge in artificial intelligence. *Commun. ACM*, 58(9):92–103, 2015.

[ECT$^+$18] Ali Emami, Noelia De La Cruz, Adam Trischler, Kaheer Suleman, and Jackie Chi Kit Cheung. A knowledge hunting framework for common sense reasoning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 1949–1958, 2018.

[JA18] Opitz Juri and Frank Anette. Addressing the winograd schema challenge as a sequence ranking task. In *Proceedings of the First International Workshop on Language Cognition and Computational Models*, pages 41–52. Association for Computational Linguistics, 2018.

[Joa02] Thorsten Joachims. Optimizing search engines using clickthrough data. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, July 23-26, 2002, Edmonton, Alberta, Canada*, pages 133–142, 2002.

[LDM12] Hector J. Levesque, Ernest Davis, and Leora Morgenstern. The winograd schema challenge.

In *Principles of Knowledge Representation and Reasoning: Proceedings of the Thirteenth International Conference, KR 2012, Rome, Italy, June 10-14, 2012*, 2012.

[Len95] Douglas B. Lenat. CYC: A large-scale investment in knowledge infrastructure. *Commun. ACM*, 38(11):32–38, 1995.

[LGK77] J Landis and Gary G. Koch. The measurement of observer agreement for categorical data. *Biometrics*, 33:159–74, 04 1977.

[Lif08] Vladimir Lifschitz. What is answer set programming? In *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence, AAAI 2008, Chicago, Illinois, USA, July 13-17, 2008*, pages 1594–1597, 2008.

[LJL+16a] Quan Liu, Hui Jiang, Zhen-Hua Ling, Si Wei, and Yu Hu. Probabilistic reasoning via deep learning: Neural association models. *CoRR*, abs/1603.07704, 2016.

[LJL+16b] Quan Liu, Hui Jiang, Zhen-Hua Ling, Xiaodan Zhu, Si Wei, and Yu Hu. Combing context and commonsense knowledge through neural networks for solving winograd schema problems. *CoRR*, abs/1611.04146, 2016.

[LJL+17] Quan Liu, Hui Jiang, Zhen-Hua Ling, Xiaodan Zhu, Si Wei, and Yu Hu. Combing context and commonsense knowledge through neural networks for solving winograd schema problems. In *2017 AAAI Spring Symposia, Stanford University, Palo Alto, California, USA, March 27-29, 2017*, 2017.

[LPC+11] Heeyoung Lee, Yves Peirsman, Angel Chang, Nathanael Chambers, Mihai Surdeanu, and Dan Jurafsky. Stanford's multi-pass sieve coreference resolution system at the conll-2011 shared task. In *Conference on Natural Language Learning (CoNLL) Shared Task*, 2011.

[LS04] Hugo Liu and Push Singh. Conceptnet—a practical commonsense reasoning tool-kit. *BT technology journal*, 22, 06 2004.

[McC60] John McCarthy. Programs with common sense. Technical report, Cambridge, MA, USA, 1960.

[MCCD13] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781, 2013.

[MDJ16] Leora Morgenstern, Ernest Davis, and Charles L. Ortiz Jr. Planning, executing, and evaluating the winograd schema challenge. *AI Magazine*, 37(1):50–54, 2016.

[Mil95] George A. Miller. Wordnet: A lexical database for english. *Commun. ACM*, 38(11):39–41, 1995.

[MJ15] Leora Morgenstern and Charles L. Ortiz Jr. The winograd schema challenge: Evaluating progress in commonsense reasoning. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA.*, pages 4024–4026, 2015.

[MMS$^+$02] John McCarthy, Marvin Minsky, Aaron Sloman, Leiguang Gong, Tessa A. Lau, Leora Morgenstern, Erik T. Mueller, Doug Riecken, Moninder Singh, and Push Singh. An architecture of diversity for commonsense reasoning. *IBM Systems Journal*, 41(3), 2002.

[MPC18] Yukun Ma, Haiyun Peng, and Erik Cambria. Targeted aspect-based sentiment analysis via embedding commonsense knowledge into an attentive LSTM. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 5876–5883, 2018.

[RDG18] Soumali Roychowdhury, Michelangelo Diligenti, and Marco Gori. Image classification using deep learning and prior knowledge. In *The Workshops of the The Thirty-Second AAAI Conference on Artificial Intelligence, New Orleans, Louisiana, USA, February 2-7, 2018.*, pages 336–343, 2018.

[RN12] Altaf Rahman and Vincent Ng. Resolving complex cases of definite pronouns: The winograd schema challenge. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, EMNLP-CoNLL 2012, July 12-14, 2012, Jeju Island, Korea*, pages 777–789, 2012.

[RWC$^+$19] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. `https://github.com/openai/gpt-2`, 2019. [Online; accessed 04-03-2019].

[SB16] Arpit Sharma and Chitta Baral. Automatic extraction of events-based conditional commonsense knowledge. In *Knowledge Extraction from Text, Papers from the 2016 AAAI Workshop, Phoenix, Arizona, USA, February 12, 2016.*, 2016.

[SB18] Arpit Sharma and Chitta Baral. Commonsense knowledge types identification and reasoning for the winograd schema challenge. `https://www.semanticscholar.org/paper/Commonsense-Knowledge-Types-Identification-and-for/1151e8ceafdd292fbf70db5cbca20a805a3ecacb`, 2018. [Online; accessed 25-02-2019].

[Sch72] Roger C. Schank. Conceptual dependency: A theory of natural language understanding. *Cognitive Psychology*, 3(4):552 – 631, 1972.

[Sch14] Peter Schüller. Tackling winograd schemas by formalizing relevance theory in knowledge graphs. In *Principles of Knowledge Representation and Reasoning: Proceedings of the Fourteenth International Conference, KR 2014, Vienna, Austria, July 20-24, 2014*, 2014.

[SVAB15] Arpit Sharma, Nguyen Ha Vo, Somak Aditya, and Chitta Baral. Towards addressing the winograd schema challenge - building and using a semantic parser and a knowledge hunting module. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015*, pages 1319–1325, 2015.

[TEC$^+$18] Paul Trichelair, Ali Emami, Jackie Chi Kit Cheung, Adam Trischler, Kaheer Suleman, and Fernando Diaz. On the evaluation of common-sense reasoning in natural language understanding. *CoRR*, abs/1811.01778, 2018.

[TL18] Trieu H. Trinh and Quoc V. Le. A simple method for commonsense reasoning. *CoRR*, abs/1806.02847, 2018.

[VSP$^+$17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 6000–6010, 2017.

[WS02] Deirdre Wilson and Dan Sperber. Relevance theory. In L. Horn and G. Ward, editors, *The Handbook of Pragmatics*, pages 607–632. Blackwell, 2002.

A Reasoning Algorithm

This is a supplementary document which contains the Answer Set Programming ASP Code for the Reasoning Algorithm A Worked Out Example ASP Code for the Reasoning algorithm: Defining domain of variables

```
k_val(X) :- has_k(X,R,Y).
k_val(Y) :- has_k(X,R,Y).
```

Extracting constant nodes from graphical representations

```
s_const(X) :- has_s(X,instance_of,I).
q_const(X) :- has_q(X,instance_of,I).
k_class(X) :- has_k(A,instance_of,X).
k_const(X) :- not k_class(X), k_val(X).
```

Extracting constant nodes which has constant parent nodes
from graphical representations of sentence, question and
knowledge

```
s_has_par(X) :- has_s(P,R,X), s_const(X), s_const(P).
q_has_par(X) :- has_q(P,R,X), q_const(X), q_const(P).
k_has_par(X) :- has_k(P,R,X), k_const(X), k_const(P).
```

Extracting constant nodes which has constant children nodes
from graphical representations of sentence, question and
knowledge

```
s_has_child(X) :- has_s(X,R,C), s_const(X), s_const(C).
q_has_child(X) :- has_q(X,R,C), q_const(X), q_const(C).
k_has_child(X) :- has_k(X,R,C), k_const(X), k_const(C).
```

Extracting cross-domain siblings from a knowledge's
representation
to a sentence's representation

```
not_k_s_crossdom_sib(X,Y) :- has_k(X,instance_of,I1),
has_k(X,instance_of,I2),
has_s(Y,instance_of,I1),
not has_s(Y,instance_of,I2),
I1!=I2.
not_k_s_crossdom_sib(X,Y) :- has_k(X,instance_of,I1),
has_k(X,instance_of,I2),
I1!=I2,
not has_s(Y,instance_of,I1),
has_s(Y,instance_of,I2).
not_k_s_crossdom_sib(X,Y) :- has_k(X,instance_of,I1),
has_k(X,instance_of,I2),
I1!=I2,
not has_s(Y,instance_of,I1),
not has_s(Y,instance_of,I2),
s_const(Y).
k_s_crossdom_sib(X,Y) :- has_s(Y,instance_of,I),
has_k(X,instance_of,I),
s_const(Y),
k_const(X),
not not_k_s_crossdom_sib(X,Y).
```

Extracting cross-domain clones from a knowledge's representation

to a sentence's representation

```
k_s_crossdom_clone(X,Y) :- k_s_crossdom_sib(X,Y),
has_k(Pj,Rj,X),
has_s(Pj_prime,Rj,Y),
k_s_crossdom_clone(Pj,Pj_prime),
k_const(Pj),
has_k(X,Rk,Cj),
has_s(Y,Rk,Cj_prime),
k_s_crossdom_sib(Cj,Cj_prime),
k_const(Cj).
k_s_crossdom_clone(X,Y) :- k_s_crossdom_sib(X,Y),
not k_has_par(X),
has_k(X,Rk,Cj),has_s(Y,Rk,Cj_prime),
k_s_crossdom_sib(Cj,Cj_prime),
k_const(Cj).
```

```
k_s_crossdom_clone(X,Y) :- k_s_crossdom_sib(X,Y),
has_k(Pj,Rj,X),
has_s(Pj_prime,Rj,Y),
k_s_crossdom_clone(Pj,Pj_prime),
k_const(Pj),
not k_has_child(X).
k_s_crossdom_clone(X,Y) :- k_s_crossdom_sib(X,Y),
not k_has_par(X),
not k_has_child(X).
```

Generating a merged representation of a sentence and a knowledge

```
has_m(X,R,Y) :- has_s(X,R,Y).
k_covered(X) :- k_const(X),
s_const(Y),
k_s_crossdom_clone(X,Y).
k_not_all_covered :- k_const(X),
not k_covered(X).
k_all_covered :- not k_not_all_covered.
has_m(X,R,Y) :- has_s(X,R,Y1),
has_s(X2,R2,Y),
Y1!=Y,
k_s_crossdom_clone(Y_k,Y1),
k_s_crossdom_clone(Y_k,Y),
k_all_covered.
has_m(X,R,Y) :- has_s(X1,R,Y),
has_s(X,R2,Y2),
X1!=X,
k_s_crossdom_clone(X_k,X1),
k_s_crossdom_clone(X_k,X),
k_all_covered.
```

Extracting constant nodes from graphical representation

```
has_const(X) :- has_m(X,instance_of,I).
```

Extracting constant nodes which has constant parent nodes

from a merged representation

```
m_has_par(X) :- has_m(P,R,X), m_const(X), m_const(P).
```

Extracting constant nodes which has constant children nodes

from a merged representation

```
m_has_child(X) :- has_m(X,R,C), m_const(X), m_const(C).
```

Extracting cross-domain siblings from a question's representation

to a merged representation

```
not_q_m_crossdom_sib(X,Y) :-
has_q(X,instance_of,I1),
has_q(X,instance_of,I2),
has_m(Y,instance_of,I1),
not has_m(Y,instance_of,I2),
I1!=I2, I1!=q, I2!=q.
not_q_m_crossdom_sib(X,Y) :-
has_q(X,instance_of,I1),
has_q(X,instance_of,I2),
not has_m(Y,instance_of,I1),
has_m(Y,instance_of,I2),
I1!=I2, I1!=q, I2!=q.
not_q_m_crossdom_sib(X,Y) :-
has_q(X,instance_of,I1),
has_q(X,instance_of,I2),
not has_m(Y,instance_of,I1),
not has_m(Y,instance_of,I2),
m_const(Y),
I1!=I2, I1!=q, I2!=q.
q_m_crossdom_sib(X,Y) :-
has_m(Y,instance_of,I),
has_q(X,instance_of,I),
not not_q_m_crossdom_sib(X,Y),
m_const(Y),
q_const(X).
```

Extracting cross-domain clones from a question's representation

to a merged representation

```
q_m_crossdom_clone(X,Y) :- q_m_crossdom_sib(X,Y),
has_q(Pj,Rj,X),
has_m(Pj_prime,Rj,Y),
q_m_crossdom_clone(Pj,Pj_prime),
q_const(Pj),has_q(X,Rk,Cj),
has_m(Y,Rk,Cj_prime),
q_m_crossdom_sib(Cj,Cj_prime),
q_const(Cj).
q_m_crossdom_clone(X,Y) :- q_m_crossdom_sib(X,Y),
not q_has_par(X),
has_q(X,Rk,Cj),
has_m(Y,Rk,Cj_prime),
q_m_crossdom_sib(Cj,Cj_prime),
q_const(Cj).
q_m_crossdom_clone(X,Y) :- q_m_crossdom_sib(X,Y),
has_q(Pj,Rj,X),
has_m(Pj_prime,Rj,Y),
q_m_crossdom_clone(Pj,Pj_prime),
q_const(Pj),
not q_has_child(X).
q_m_crossdom_clone(X,Y) :- q_m_crossdom_sib(X,Y),
not q_has_par(X),
not q_has_child(X).
```

Extracting the answers to the input question

```
q_covered(X) :- q_const(X),
m_const(Y),
q_m_crossdom_clone(X,Y).
q_not_all_covered :-
not q_covered(X),
q_const(X).
q_all_covered :- not q_not_all_covered.
ans(Q,X) :- q_m_crossdom_clone(Q,X),
has_q(Q,instance_of,q),
q_all_covered.
```

Making answers visible in the terminal

```
#show ans/2.
```

B Annotation

appendix with our annotations