Towards Addressing the Winograd Schema Challenge — Building and Using a Semantic Parser and a Knowledge Hunting Module

Arpit Sharma, Nguyen H. Vo, Somak Aditya & Chitta Baral
School of Computing, Informatics & Decision Systems Engineering
Arizona State University
Tempe, AZ 85281, USA
{asharm73, nhvo1, saditya1, chitta}@asu.edu

Abstract

Concerned about the Turing test's ability to correctly evaluate if a system exhibits human-like intelligence, the Winograd Schema Challenge (WSC) has been proposed as an alternative. A Winograd Schema consists of a sentence and a question. The answers to the questions are intuitive for humans but are designed to be difficult for machines, as they require various forms of commonsense knowledge about the sentence. In this paper we demonstrate our progress towards addressing the WSC. We present an approach that identifies the knowledge needed to answer a challenge question, hunts down that knowledge from text repositories, and then reasons with them to come up with the answer. In the process we develop a semantic parser (www.kparser.org). We show that our approach works well with respect to a subset of Winograd schemas.

1 Introduction

With significant advances and success in many Artificial Intelligence subfields, and instances of gaming the Turing test, there is now a need to more clearly define how to evaluate when a system is replicating humanoid intelligence. The Winograd Schema Challenge (WSC) [Levesque *et al.*, 2011] is one such attempt. The WSC corpus consists of a set of pairs of a sentence and a question such that the answer to the question is about resolving a definite pronoun or possessive adjective to one of its two probable co-referents in the sentence. The co-referents belong to same gender and they have a number agreement between them (both are either singular or plural). These make it harder to resolve the pronoun to its antecedent. The sentence also contains a *"special word"* which when replaced by another word (*alternate word*), the answer to the question also changes. An example of such a pair of Winograd Schema sentences is as follows.
Sentence: *The man couldn't lift his son because he was so* ***heavy***. **Question:** *Who was heavy ?* **Answer:** *son*
Sentence: *The man couldn't lift his son because he was so* ***weak***. **Question:** *Who was weak ?* **Answer:** *man*
The motivation behind this challenge is to evaluate human-like reasoning in machines. For example, one of the many ways in which humans answer the first sentence above is by using the commonsense that "someone who could not be lifted may be heavy". Thus a human-like intelligent system that can answer such questions correctly needs to have this knowledge and the ability to reason with that knowledge. Very recently, this aspect of human-like intelligence has also been emphasized in several other (besides the Winograd challenge) research initiatives such as the call by Paul G. Allen Family Foundation, and the Big Mechanism call of DARPA. Since the existing knowledge repositories (such as CYC, ConceptNet, and AURA) are not comprehensive enough, and a comprehensive knowledge base may be too unwieldy, we propose the following approach to address this aspect of human-like intelligence: 1. determining what kind of (commonsense) knowledge is needed; 2. finding a way to acquire that knowledge [Knowledge Hunting]; and 3. reasoning with that knowledge [Reasoning].

In this paper we make a start with respect to WSC and illustrate our approach with respect to a subset of it. In particular, we identify two specific categories of WSC schemas and focus on those. In the process we develop several needed tools, such as a non domain-specific semantic parser (K-Parser, demo available at *www.kparser.org*), a Knowledge Hunting module that is able to automatically extract knowledge needed for the two categories of WSC schemas that we focus on, and a reasoning module that reasons with the K-Parser output. The semantic parser integrates various natural language processing aspects, and incorporate ontologies and knowledge. It is of independent interest, but here we only describe it to the extent used in this paper. We illustrate a simple graph based reasoning engine to answer the WSC schema questions by using the semantic representation of both the input text and the automatically extracted knowledge.

2 The Corpus

The WSC corpus consists of 282 sentence and question pairs. We evaluated our technique on a subset of the WSC corpus. The subset consists of a significant number of Winograd schemas that require two specific kind of commonsense knowledge as mentioned below:
• **Direct Causal Events - Event-event causality**: In this category, the commonsense knowledge required has two mutually causal events (*explained* and *convince* in the example below) such that a pronoun participates in one of the event and

its candidate co-referent participates in another. For example, for the text "*Sid explained his theory to Mark but he could not convince him .*" and the question "*Who could not convince ?*", the expected answer is "*Sid*". One way in which humans resolve *he* to *Sid* is by using the commonsense knowledge: **IF** (*X explained S to Y but Z could not convince*) **THEN** (*Z=X* **i.e.** agent of *explained*=agent of *could not convince*).

• **Causal Attributive**: In this category, the commonsense knowledge required has an event and a participant entity in that event has an associated attribute that is causally related to the event. For example, for the text "*Pete envies Martin because he is very successful.*" and the question "*Who is successful ?*", the expected answer is "*Martin*". A kind of commonsense knowledge required to get this answer is of the form: *X envies Y* **possibly because** *Y* **has trait** *successful*. Here *Y* is the participant entity and *envies* is the event in the sentence.

A total of 71[1] WSC corpus sentences are categorized in the above mentioned categories. The remaining 211 sentences[2] do not fall into these categories because the kind of commonsense knowledge mentioned above is not enough to solve them. For example, for the sentence "*There is a pillar between me and the stage, and I can not see it*" and the question "*What can not I see ?*" the required commonsense knowledge includes: "*If something big comes in between me and the stage then my sight is blocked; pillar represents something big; and if my sight is blocked then I can not see.*" As we can see that this commonsense knowledge does not fit into the above two categories.

3 Our Approach

Our approach is based on three main tasks, namely, semantic parsing of text, automatic extraction of commonsense knowledge about the input text and using a graph based reasoning on the semantic representations of both the input and the commonsense knowledge to get the answer to the WSC schema questions. The sections below explain each of these tasks along with the tools and techniques that we developed.

3.1 Semantic Parsing: K-Parser (kparser.org)

A semantic representation of text is considered good if it can express the structure of the text, can distinguish between the events and their environment in the text, uses a general set of relations between the events and their participants, and is able to represent the same events or entities in different perspectives. Keeping these features in mind, we have developed a graph based semantic parser (Knowledge Parser or K-Parser) that produces a semantic representation of the input text with the following properties:

• Has an acyclic graphical representation for English text. The representation is easy to read.

• Has a rich ontology (KM [2004]) to represent semantic relations (Event-Event relations such as *causes*, *caused_by*, Event-Entity relations such as *agent*, and Entity-Entity relations such as *related_to*).

[1]including #4, #6, #72 and #23 from *http://www.cs.nyu.edu/davise/papers/WS.html*

[2]including #34, #35, #41, #48, #50 and #43

• Has special relations (*instance_of* and *prototype_of*) to represent the existential and universal quantification of entities.

• Has two levels of conceptual class information for words.

• Accumulates semantic roles of entities based on PropBank framesets [Palmer *et al.*, 2005].

• Has tenses of the verbs in the input text.

• Has other features such as an optional Co-reference resolution, Word Sense Disambiguation and Named Entity Tagging.

Algorithm 1 demonstrates the steps used to create the semantic graph in K-Parser. The algorithm consists of five pri-

Algorithm 1 K-Parser Algorithm

```
1: procedure CreateGraph(text)
2:     S ← ExtractSynDeps(text)
3:     G ← SemanticMapping(S, L_KM)
4:     for all node v ϵ G do
5:         G ← G + GetClass(v, WS_v)      ▷ WS_v=word sense of v
6:     for all edge e ϵ G do
7:         EdgeLabelCorrection(G)
8:     G ← AddFeature(G, SRL_ent)         ▷ Semantic Roles of Entities
9:     G ← AddFeature(G, CR_ent)          ▷ Co-reference Resolution
10:    return G                           ▷ The Semantic Graph
```

mary modules. The first module $ExtractSynDeps$ is used to extract the syntactic dependency graph from the input text. We used Stanford Dependency Parser [De Marneffe *et al.*, 2006] for this purpose.

The second module, $SemanticMapping$ is used to map the syntactic dependency relations to KM relations [2004] and a few newly created relations. There are three methods used for semantic mapping. First, we used mapping rules to map syntactic dependencies into semantic relations. For example the *nominal subject* dependency is mapped to *agent* relation. Second, we developed a multi-class multilayer perceptron classifier for disambiguating different senses of prepositions and assign the semantic relations appropriately. The training data for classification is taken from The Preposition Project [Litkowski, 2013] and the sense ids for prepositions are manually mapped to KM relations. Third, we used the discourse connectives in the text to label the event-event relations. We labeled different connectives with different labels. For example, the coordinate connectives such as *but*, *and*, *comma (,)* and *stop(.)* are labeled as *next_event*. Other connectives are also labeled based on their effect, such as *because* and *so* are labeled *caused_by* and *causes* respectively.

The third module, $GetClass$ accumulates two level of classes for each node in the output of Semantic Mapping function. Word Sense Disambiguation [Basile *et al.*, 2007] along with the lexical senses from WordNet [1995] are used for this task. The fourth module, $EdgeLabelCorrection$ corrects the mappings done by the mapping function by using class information extracted by the third function. For example, if there is a relation *is_possessed_by* between two nodes with their superclass as *person*, then the relation is corrected to *related_to* (because a person can not possess another per-

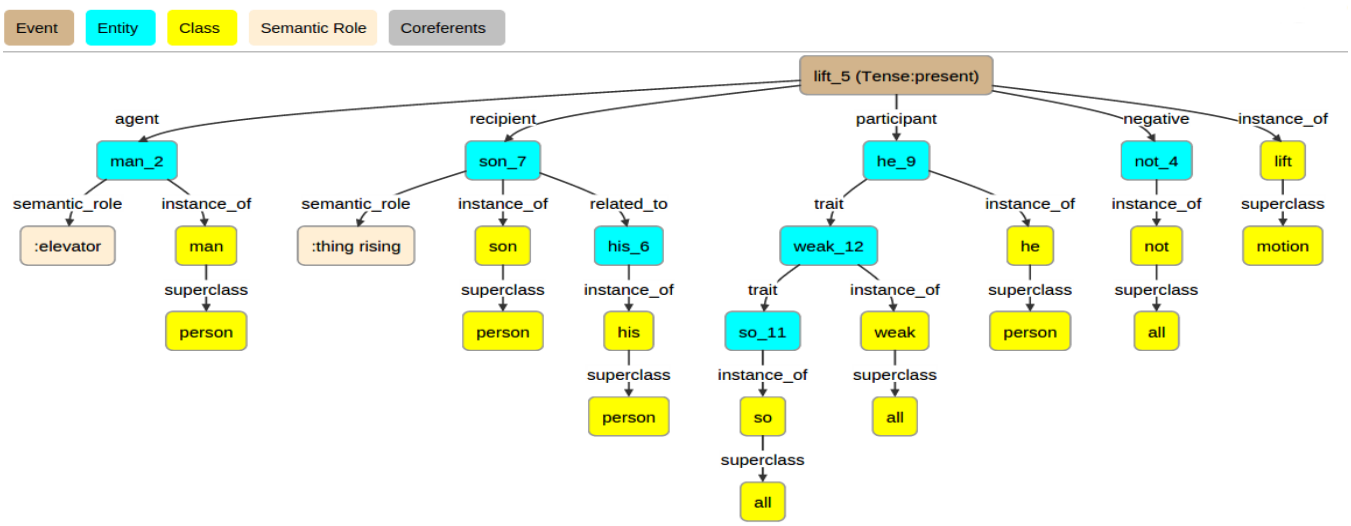

Figure 1: K-Parser output for *"The man could not lift his son because he was so weak.*

son). Lastly, the $AddFeature$ module is used to implement other features such as semantic roles of the entities by using Propbank Framesets [Palmer *et al.*, 2005]. An option for co-reference resolution is also provided in the system which uses the Stanford Co-reference resolver [2010]. Furthermore, many other tools are also used at various steps in the above mentioned algorithm, such as Named Entity Tagging, WordNet [1995] database and Weka statistical classifier library [Dimov *et al.*, 2007]. Figure 1 demonstrates the output of K-Parser for the sentence *The man could not lift his son because he was so weak.* We also defined an algorithm consisting of a set of rules to match a question's semantic representation with that of the respective sentence's and extract the *pronoun to be resolved* from the sentence.

Kparser is a general semantic parser which has been used beyond addressing WSC. In particular, it has been used in parsing text in image interpretation and engineering design specification, and is being used in interactive planning.

3.2 Knowledge Hunting

The questions in Winograd Schema Challenge can be easily answered by human beings using simple knowledge that they have learned over the years. One of the ways they learn is by reading. For example, a Guardian article[3] states that *"There is evidence that reading can increase levels of all three major categories of intelligence."* For example in the Winograd Schema, **Sentence:** *The man couldn't lift his son because he was so weak.* **Question:** *Who was so weak?*; the answer to the question can be achieved by using the commonsense knowledge that *if X could not lift Y* then *X may be weak.*

In our system we try to imitate this. The only difference is that we retrieve the commonsense knowledge in an on-demand manner, and only relevant to the given sentence and the question. We do that by creating string queries from the concepts in the sentence and the question and use the queries to retrieve sentences from a text repository.

We now use the above mentioned example to explain the two step process. The first step is to create a query set by using the representation of the given sentence and question. Following are the sets of queries that are created:

• The first set of queries is created by using formal representations of both the Winograd sentence and the question. All the nodes from the question's formal representation (except the ones which represent "Wh" words) are mapped into the formal representation of the given sentence. From the mapped output, all the words/nodes which do not specify a nominal entity are extracted and their different combinations are joined together using a wild card (.*) and quotes (""). An example query for the *lift* example mentioned above is, *".*could not lift.*because.*weak.*"*.

• The second set of queries is created by replacing the verbs in the previously created set of queries by their synonyms. For example, a new query for the *lift* example that is generated is: *".*could not raise.*because.*weak.*"*, where *raise* is a synonym of *lift*.

Finally, a combined set of queries is formed by merging the above two sets. The second sub-step in the commonsense knowledge extraction process is to automatically search a large corpus of English text using the queries and extract the sentences which contain all the words (in any form) in the respective query. We used the Google search engine API along with sentence splitting to get such sentences from the textual part of WWW but the searching can be performed on other text repositories as well. The idea here is to extract the sentences which contain the commonsense knowledge that is required to answer the question about the given Winograd sentence. One of the sentences extracted from Google by using the above mentioned queries for the *lift* example is "*She could not lift it because she is a weak girl.*" In general, in English language, when a sentence has two mentions of the same pronoun (e.g. *she*) then, they both represent the same

[3]http://www.theguardian.com/books/2014/jan/23/can-reading-make-you-smarter

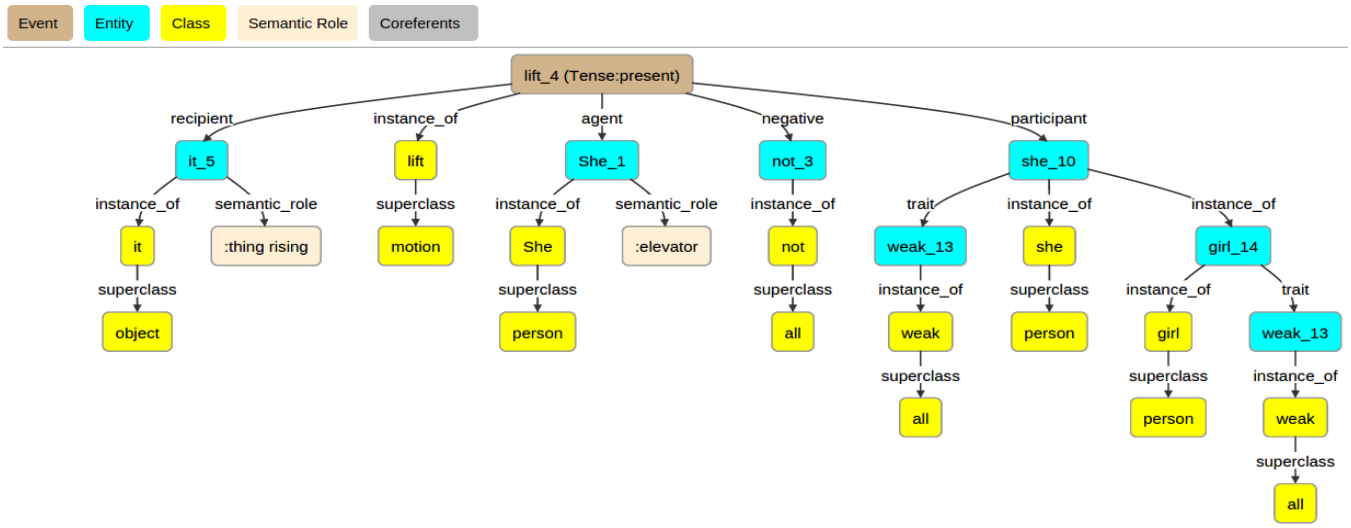

Figure 2: K-Parser output for *"She could not lift it because she is a weak girl.*

entity. We use that in post-processing. Such post processing is performed on the semantic representation of the sentence. Figure 2 demonstrates the output of K-Parser for the sentence "*She could not lift it because she is a weak girl.*"

3.3 Reasoning on Semantic Graphs

This section explains the module that answers the input questions using logical reasoning on the K-Parser output. The module uses the semantic representation graphs of the given sentence, and the sentences containing commonsense knowledge obtained by the knowledge hunting step. As mentioned in Section 2, we focus on the two categories of sentences: *Direct Causal Events* and *Causal Attributive*. We use similar reasoning (i.e., matching) techniques for both of them. Each given sentence and corresponding commonsense knowledge sentences are translated into the semantic representation graphs, by using the K-Parser system. They are then matched using a set of rules and logical constructs.

Consider our running example, where the semantic representation of the input sentence is shown in Figure 1, of the common sense sentence is shown in Figure 2, and the question is "Who was so weak?". From the question and the graph of input sentence (Figure 1), we first find out that the *pronoun to be resolved* is *he_9*. We also find the event directly connect to it ($lift\_5$), and the trait/property associated with it ($weak\_12$).

Next we extract similar features from the commonsense knowledge sentence (Figure 2). We then match event $lift\_4$ and trait $weak\_13$ to event $lift\_5$ and trait $weak\_12$ in the other graph. Next, we reason that the entity she_10 (Figure 2) must match to he_9 (Figure 1). Since the commonsense knowledge sentence (in Figure 2) is simpler and there is no ambiguity, we determine that she_10 and she_1 refer to the same entity. From this, we deduce that the *agent* of event *lift* is also its *participant*. Applying this knowledge on the input sentence (Figure 1), we have that the *agent* man_2 is the co-referent of he_9, which is what we were looking for.

We have used Answer Set Programming (ASP) [Gelfond and Lifschitz, 1988; Baral, 2003] to define the rules and constructs for reasoning. The semantic graphs are translated into the ASP constructs by using quaternary predicate *has*. For example, *has(winograd,X,R,Y)* means that (1) there exists an edge in the semantic graph of the WSC sentence has end nodes X and Y; and (2) the semantic relation between them is R. Alternatively, we use *has(commonsense,X,R,Y)* if the edge is in the commonsense knowledge graph. We also use *toBeResolved(Pronoun)* to define the *pronoun to be resolved*.

The following section gives more details about the reasoning rules. (A version of the ASP rules are available at [Sharma, 2014].)

General Properties:

Three types of nodes: $Events$, $Entities$, and $Classes$ (in five possible types - see the legend in Figure 1) are used to define the set of properties mentioned below.

- The basic transitivity relationship between two event nodes is defined if an event node is reachable from another event node, traversing along any directed edge in the semantic graph.
- If two different nodes in different sentences (Winograd or Commonsense) are instances of the same class then they are defined as cross context siblings. For example $crossContextSiblings(lift\_5, lift\_4)$ follows from Figure 1 & 2.
- A node in any semantic graph is defined to have negative polarity if it has an outgoing edge labeled as *negative*. For example $negativePolarity(lift\_5)$ follows from Figure 1.

Type Specific Properties:

Type1: Causal Attributive:

- In the semantic graph of the Winograd sentence, if there is an event node connected to the *pronoun to be resolved* with an edge labeled *participant*, and there exists an edge $has(winograd, X, trait, B)$; then we define a predicate $attSubgraph(winograd, A, X, B)$. For example,

$attSubgraph(winograd, lift\_5, he, weak)$ follows from figure 1.

• A predicate that extracts nodes from the semantic representation of the commonsense knowledge graph is defined. I.e., $attSubgraph(commonsense, A, X, B)$, if there exists $attSubgraph(winograd, A1, X1, B1)$ where $crossContextSiblings(A, A1)$ and $crossContextSiblings(B, B1)$. For example $attSubgraph(commonsense, lift\_4, she\_10, weak\_13)$ follows from figure 1 and 2.

• Finally the co-referent of the pronoun to be resolved in the predicate $hasCoreferent(X, C)$ if $attSubgraph(commonsense, A1, X1, B1)$, $has(commonsense, A1, R, X1)$ and there exists an edge in the winograd sentence's semantic graph $has(winograd, A, R, C)$, $(C \neq X)$, where $crossContextSiblings(A, A1)$. For example, $hasCoreferent(he\_9, man\_2)$ follows from figure 1.

Type2: Direct Causal Events The reasoning for this type is similar to the other type with a different set of predicates and their definitions.

• There are two event nodes connected transitively in the semantic graph of the Winograd sentence. First step is to identify the similar chain of two transitive event nodes from commonsense sentence's semantic graph by using the general properties defined in the previous section.

• A subgraph from the semantic representation of the given Winograd sentence is extracted which consists of the *pronoun to be resolved* and the events and entities required to resolve it to its antecedent. A similar subgraph, based on the general properties and matching event nodes extracted in the previous step is extracted from the semantic representation of the commonsense sentence.

• Finally, both the subgraphs extracted in the previous steps are compared (similar to Type1 reasoning mentioned above) and the resolution of pronoun is done.

4 Evaluation and Error Analysis

Although, the purpose of this paper is to present a novel technique that hunts for commonsense knowledge and uses it to answer difficult questions, we have evaluated our approach in parts and as a whole. We first evaluated the K-parser component by itself and then evaluated the whole system with respect to WSC.

4.1 K-Parser Evaluation

We developed K-Parser by using the training sentences collected from many sources such as the example sentences from Stanford dependency manual [2008] and dictionary examples for sentences with conjunctions. Though our initial intention behind developing K-Parser was to solve WSC, we realized that a non domain-specific semantic parser could help us achieve our goal and would be a general contribution to the NLP community. We evaluated the K-Parser's output for the test sentences from WSC against the manually created gold standard representations. We identified five important categories to assess the accuracy of K-Parser. The categories are Number of Events, Number of Entities, Number of Classes, Number of Event-Event Relations and Number of Event-Entity Relations. We defined Precision and Recall of our system based on the above categories, Precision = $t_1/(t_1 + t_2 + t_3)$, Recall = $t_1/(t_1 + t_2 + t_4)$, where, t_1 = identified and relevant and the label is correct; t_2 = identified and relevant and the label is wrong; t_3 = identified, but not relevant; t_4 = not identified, but relevant.

Table 1: Evaluation Results table

	Precision	Recall
Events	0.94	0.92
Entities	0.97	0.96
Classes	0.86	0.79
Event-Event Relations	0.91	0.79
Event-Entity Relations	0.94	0.89

Table 1 shows the evaluation results for K-Parser.

4.2 Overall System Evaluation and Error Analysis

There are 282 total sentence and question pairs in the Winograd Schema Challenge corpus. Out of those, we identified a total of 71 sentences from both the categories *Causal Attributive* and *Direct Causal Events*. Among the 71 pairs, our system is able to answer 53 and the remaining 18 pairs are left unanswered. Out of the 53 answered, 49 are correctly answered. Four of them are incorrectly answered because the commonsense knowledge found was inappropriate. For example the commonsense knowledge "*I paid the price for my stupidity. How grateful I am*" was found for the Winograd Schema sentence, *Bob paid for Charlie's college education, he is very grateful*. In this sentence, there is only one entity in the commonsense sentence (presented by the words, *I* and *my*. As mentioned earlier, these words are post processed as one in our semantic representation). Hence, this particular extracted knowledge is not appropriate for the given sentence.

It must also be noted that our system does not return any answer if no commonsense knowledge is found or it is not sufficient to answer the question. This property is advantageous because it provides the ability to use another commonsense knowledge source and this process is repeatable. Furthermore, if the system finds multiple answers for the same question, then the support for each answer is calculated in terms of count and the answer with maximum support is the final answer.

5 Related Works

The above sections explain two main components of our approach towards solving the WSC. In this section we compare the tools we developed along with our approach in whole with the currently existing tools and techniques that attempt to solves these problems. Hence, we divide the comparison section in two categories mentioned below.

5.1 Semantic Parsers

The semantic parsing systems available today belong to many categories. For example, there are systems [Berant and Liang, 2014; Fader *et al.*, 2014] that aim at translating factual questions into query representations. These queries are useful

to extract the answer to the given questions from a factual knowledge base such as Freebase. Another category consists of Semantic Role Labeling systems such as [Punyakanok *et al.*, 2004] and SEMAFOR parser [Das *et al.*, 2010]. While they assign semantic roles to entities and events in the text, they lack the causal or non-causal relations between events or actions. Furthermore, these systems do not correctly process the implications, quantifications and conceptual class information about the text (eg. "John" is an instance of "person" class). The category of semantic parsers that are non-domain specific can be further classified to two classes. The first class consists of systems which translate a given text into a logical language. For example, Boxer [Bos, 2008] translates English sentences into first order logic. Despite its many advantages, it does not capture the event-event and event-entity relations in the text. The inclusion of the homonym-hypernym information and resolution of identical meaning words are important for downstream reasoning. Such ontological information about entities or similarities between connectives are also not captured in the Boxer system. The second class consists of parsers which translate the given text into a graph based representation. Each node in the graph represents a word or concept and the edges (obtained from a predefined or newly defined ontology) represent the relation between those concepts. For example Flanigan et al., [2014] illustrate a semantic parser that translates natural language strings into Abstract Meaning Representation (AMR) [Banarescu *et al.*, 2013]. AMR has limitations such as it does not have meaningful relation labels (e.g. *ARG1, ARG2,...* instead of *agent, recipient...*) and event-event relations. TRIPS parser [Allen *et al.*, 2007; Dzikovska *et al.*, 2003] also falls in the class of graph based semantic parsers. It encodes features such as the conceptual classes of the words, quantification of entities, and representation of the participants of an event. However, event-event relations are not captured by this system.

5.2 Winograd Schema Challenge

There are few published techniques that aim at solving the Winograd Schema Challenge or a similar corpus. One of them is demonstrated by Rahman et al., [2012] which uses a number of techniques to resolve pronouns in a Winograd Schema like corpus. Their system uses various techniques and combine their results on a corpus of 941 such schema. There are a few issues with the techniques used in their system. For example a technique used by them creates string queries from given sentences and finds the support for the queries from Google search results. The issue with this technique is that sometimes the queries do not justify the outcome of the technique. For example, a query for the sentence *Lions eat zebras because they are predators* is *"Lions are predators"*. It makes sense to find the support for lions being predators based on this query. But if the sentence is changed to *Lions eat zebras because they are hungry* then the support for the query *"Lions are hungry"* is not capable of justifying the fact that *they* in the sentence refers to *Lions*. This is because *zebras* are equally likely to be hungry if no context is provided. A different work that attempts to solve the WSC by using the modifications of above mentioned techniques along with new ones is demonstrated in [Budukh, 2013]. However the techniques performs decently on the WSC, the deeper analysis of results leads to a conclusion that most of the correct answers do not follow humanoid reasoning, in fact they are correct by chance.

Another work by Schuller [2014], demonstrates a graph based technique and performs experiments on 4 out of 141 Winograd Schema pairs. It converts the given Winograd sentence to a dependency graph using Stanford dependency parser and then manually creates a background knowledge dependency graph which is required to answer the question. The main contribution of this work is to formalize a way to combine both a given sentence dependency graph and the *manually created background knowledge* dependency graph in using relevance theory and then use Answer Set Programming (ASP) to extract the answer. But, unlike our approach, commonsense knowledge is not automatically extracted by this system.

6 Conclusion and Future Work

In this paper we report on our development of a semantic parser (K-Parser) and a knowledge hunting mechanism and their use in addressing the WSC. Given a WSC schema our knowledge hunting mechanism searches a text repository to find needed knowledge for that schema. For example, for the sentence in Figure 1 and the question "Who was so weak", our system finds the sentence in Figure 2. The knowledge in that sentence about X cannot lift Y when X is weak is needed to answer the question. Indeed, our system uses that knowledge in answering the question. Whether this knowledge is a deep background knowledge or is an evidence from the text is a matter of where the line is drawn regarding when an evidence becomes knowledge. Regardless, our usage of knowledge of the form "X cannot lift Y when X is weak" is an example of commonsense reasoning and is more meaningful than bag-of-words or related statistical approaches commonly used in NLP and NLU.

Although we achieved a notable accuracy on the two identified WSC categories (that covers 71 of 282 schemas), a lot more remains to be done. As future work, we are working on identifying other categories of Winograd Schema sentences. We are also trying to create a knowledge repository of commonsense knowledge by extracting the knowledge from text repositories. But, additional success on these may imply that there is a need to develop a harder challenge schema that would require deeper knowledge and more involved commonsense reasoning. We believe that to obtain more involved knowledge from text, there is a need to go beyond standard fact extraction approaches to semantic translation of text. The research call by the Paul G. Allen foundation, and other research initiatives (such as the "Machine Reading" initiative) have similar motivations and implications.

Acknowledgements

We thank NSF for the DataNet Federation Consortium grant OCI-0940841 and ONR for their grant N00014-13-1-0334 for partially supporting this research.

References

[Allen *et al.*, 2007] James Allen, Mehdi Manshadi, Myroslava Dzikovska, and Mary Swift. Deep linguistic processing for spoken dialogue systems. In *Proceedings of the Workshop on Deep Linguistic Processing*, pages 49–56. ACL, 2007.

[Banarescu *et al.*, 2013] Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. Abstract meaning representation for sembanking. In *Proc. of the Linguistic Annotation Workshop and Interoperability with Discourse*. Citeseer, 2013.

[Baral, 2003] Chitta Baral. *Knowledge representation, reasoning and declarative problem solving*. Cambridge university press, 2003.

[Basile *et al.*, 2007] Pierpaolo Basile, Marco Degemmis, Anna Lisa Gentile, Pasquale Lops, and Giovanni Semeraro. The jigsaw algorithm for word sense disambiguation and semantic indexing of documents. pages 314–325, 2007.

[Berant and Liang, 2014] Jonathan Berant and Percy Liang. Semantic parsing via paraphrasing. In *Proceedings of ACL*, 2014.

[Bos, 2008] Johan Bos. Wide-coverage semantic analysis with boxer. In *Proceedings of the 2008 Conference on Semantics in Text Processing*, pages 277–286. ACL, 2008.

[Budukh, 2013] Tejas Ulhas Budukh. An intelligent coreference resolver for winograd schema sentences containing resolved semantic entities. Master's thesis, Arizona State University, 2013.

[Clark *et al.*, 2004] Peter Clark, Bruce Porter, and Boeing Phantom Works. Km - the knowledge machine 2.0: Users manual. *Department of Computer Science, University of Texas at Austin*, 2004.

[Das *et al.*, 2010] Dipanjan Das, Nathan Schneider, Desai Chen, and Noah A Smith. Semafor 1.0: A probabilistic frame-semantic parser. *Language Technologies Institute, School of Computer Science, Carnegie Mellon University*, 2010.

[De Marneffe and Manning, 2008] Marie-Catherine De Marneffe and Christopher D Manning. Stanford typed dependencies manual. *URL http://nlp. stanford. edu/software/dependencies manual. pdf*, 2008.

[De Marneffe *et al.*, 2006] Marie-Catherine De Marneffe, Bill MacCartney, Christopher D Manning, et al. Generating typed dependency parses from phrase structure parses. In *Proceedings of LREC*, volume 6, pages 449–454, 2006.

[Dimov *et al.*, 2007] Rossen Dimov, Michael Feld, Dr Michael Kipp, Dr Alassane Ndiaye, and Dr Dominik Heckmann. Weka: Practical machine learning tools and techniques with java implementations. *AI Tools SeminarUniversity of Saarland, WS*, 6(07), 2007.

[Dzikovska *et al.*, 2003] Myroslava O Dzikovska, James F Allen, and Mary D Swift. Integrating linguistic and domain knowledge for spoken dialogue systems in multiple domains. In *Proc. of IJCAI-03 Workshop on Knowledge and Reasoning in Practical Dialogue Systems*, 2003.

[Fader *et al.*, 2014] Anthony Fader, Luke Zettlemoyer, and Oren Etzioni. Open question answering over curated and extracted knowledge bases. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1156–1165. ACM, 2014.

[Flanigan *et al.*, 2014] Jeffrey Flanigan, Sam Thomson, Jaime Carbonell, Chris Dyer, and Noah A Smith. A discriminative graph-based parser for the abstract meaning representation. 2014.

[Gelfond and Lifschitz, 1988] Michael Gelfond and Vladimir Lifschitz. The stable model semantics for logic programming. In *ICLP/SLP*, volume 88, pages 1070–1080, 1988.

[Levesque *et al.*, 2011] Hector J Levesque, Ernest Davis, and Leora Morgenstern. The winograd schema challenge. In *AAAI Spring Symposium: Logical Formalizations of Commonsense Reasoning*, 2011.

[Litkowski, 2013] Ken Litkowski. The preposition project corpora. Technical report, Technical Report 13-01. Damascus, MD: CL Research, 2013.

[Miller, 1995] George A Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

[Palmer *et al.*, 2005] Martha Palmer, Daniel Gildea, and Paul Kingsbury. The proposition bank: An annotated corpus of semantic roles. *Computational linguistics*, 31(1):71–106, 2005.

[Punyakanok *et al.*, 2004] Vasin Punyakanok, Dan Roth, Wen-tau Yih, and Dav Zimak. Semantic role labeling via integer linear programming inference. In *Proceedings of the 20th international conference on Computational Linguistics*, page 1346. ACL, 2004.

[Raghunathan *et al.*, 2010] Karthik Raghunathan, Heeyoung Lee, Sudarshan Rangarajan, Nathanael Chambers, Mihai Surdeanu, Dan Jurafsky, and Christopher Manning. A multi-pass sieve for coreference resolution. In *Proceedings of the 2010 Conference on EMNLP*, pages 492–501. ACL, 2010.

[Rahman and Ng, 2012] Altaf Rahman and Vincent Ng. Resolving complex cases of definite pronouns: the winograd schema challenge. In *Proceedings of the 2012 Joint Conference on EMNLP and CoNLL*, pages 777–789. ACL, 2012.

[Schuller, 2014] Peter Schuller. Tackling winograd schemas by formalizing relevance theory in knowledge graphs. International Conference on Principles of Knowledge Representation and Reasoning, 2014.

[Sharma, 2014] Arpit Sharma. Solving winograd schema challenge: Using semantic parsing, automatic knowledge acquisition and logical reasoning. Master's thesis, Arizona State University, 2014.