NOTICE

This is the author's of a work accepted for publication by Springer. The final publication is available at www.springerlink.com:

https://doi.org/10.1007/978-3-030-04284-4_12

A Marker Passing Approach to Winograd Schemas

Johannes Fähndrich, Sabine Weber, Hannes Kanthak, and Sahin Albayrak

Technische Universität Berlin
Johannes.Faehndirch@dai-labor.de
www.fähndrich.de

Abstract. This paper approaches a solution of Winograd Schemas with a marker passing algorithm which operates on an automatically generated semantic graph. The semantic graph contains common sense facts from data sources form the semantic web like domain ontologies e.g. from Linked Open Data (LOD), WordNet, Wikidata, and ConceptNet. Out of those facts, a semantic decomposition algorithm selects relevant facts for the concepts used in the Winograd Schema and adds them to the semantic graph. Markers are propagated through the graph and used to identify an answer to the Winograd Schema. Depending on the encoded knowledge in the graph (connectionist view of world knowledge) and the information encoded on the marker (for symbolic reasoning) our approach selects the answers. With this selection, the marker passing approach is able to beat the state-of-the-art approach by about 12%.

Keywords: Semantic Web · LOD · Winograd Schema · Common Sense Reasoning · Symbolic Connectionist AI

1 Introduction

Artificial Intelligence (AI) helps to solve ever more complex problems. The use of increasingly sophisticated software enables us to automate many tedious tasks, perform better research and grasp a better understanding of the world, e.g., playing Go [30], or fighting cancer [9]. But *"Despite all these developments, the promises of strong artificial intelligence set forth in the 1960s have not been fulfilled."* [31, p. 7], meaning that AI is not able to understand natural language [33], construct plans on dynamic domains [12], or do common sense reasoning like humans [27]. These second kinds of problems are solved by a so-called "strong AI" [25][1]. A strong AI is able to learn new problem-solving skills in new domains. The adaption to new domains is one of the differences to special purpose AI where a chess AI is, e.g., unable to drive a car.
One of the reasons for human intelligence might be the ability to think. Having a language to formulate thoughts, meaning and ideas, helps us to handle unknown situations with adaptiveness and dynamic behavior. Part of the capacity to think is reasoning, which does not always "obey the rules of classical logic" but gives us our common

[1] *Strong AI* (sometimes called *full AI* or *hard AI*) [14, p. 260] refers to a human level intelligence.

sense [11]. The foundation for a language to think is a representation of meaning. Consequently, research in AI analyzes how methods from mathematics, linguistics, psychology, philosophy and computer science can be used to create machines with the ability to represent meaning.
A main source of world knowledge is the semantic web consisting of multiple ontologies which are created and maintained by multiple organizations [32]. Collecting knowledge from the semantic web and merging it into one representation can be seen as a first step to reasoning on with the LOD which includes Information sources like DBPedia [2] or Freebase or YAGO [1].
The use of world knowledge in common sense reasoning has multiple applications [15]. One application is to answer commonsense reasoning questions like the test questions called Winograd Schemas. The *Commonsense Reasoning - Winograd Schema Challenge (WSC)*[2] tests the best approaches solving Winograd Schemas.
This paper describes our approach to solving Winograd Schemas: Using knowledge extracted form the Semantic Web to create an connections representation of facts. This connectionist representation is then used for pragmatic inference. For the evaluation, we will use the Common-sense Reasoning Winograd Challenge dataset.
In the following we will use the example Winograd Schema, as an running example:
"**The trophy would not fit in the brown suitcase because it was too big (small). What was too big (small)?**" Answer 0: **the trophy** Answer 1: **the suitcase**
In this example the answer changes depending of the used adjective: big (the trophy) or small (the suitcase).

2 State-of-the-art

The Winograd Schema Challenge was first proposed in the year 2011 by Hector Levesque [16] as a test for machine intelligence as an alternative to the Turing Test. On the first glance, the Winograd Schema Challenge seems like a task in anaphora resolution. However, rather than to be solvable with only grammatical and semantic relations, it requires world knowledge and common sense reasoning. This section will look at the newest AI approaches to solve Winograd Schemas and describe the available data sets.
Related Work
After the initial Winograd Schema Challenge described by Levesque [16], multiple approaches to solving Winograd Schemas where published. The result of our analysis of related work is shown in Fig. 1. One of the first approaches by Rahman and Ng [23] combines eight different methods, e.g., using the Google search engine and comparing the number of results.
Starting in 2015 a new era of approaches were introduced by Sharma et al. [26]. They developed an approach which combines existing methods for knowledge collection and extends them with a semantic parser. Also in 2015 Sharma et al. [22] used statistical methods with the focus on sentence predicates. Since the statistical analysis of Winograd Schema evades the challenge of common sense reasoning a change in the competition dataset was needed.

[2] `http://commonsensereasoning.org/winograd.html` last visited on 30.07.2018

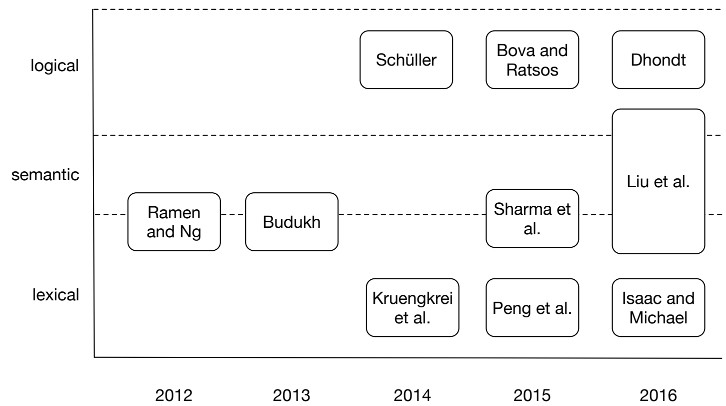

Fig. 1. Classification of state-of-the-art in time.

After the rules for the construction of Winograd Schemas have been changed with the suggestions of Levesque et al. [16], approaches building solely on statistical analysis are no longer capable of mastering the challenge. After this change, the first official Common Sense Reasoning Winograd Schema Challenge organized by the New York University was conducted. With the improved data set, the results of the known approaches dropped to the extent that the best result in 2016 was as low as 58%, with a random score of 48% [6]. One of the most promising approaches in 2016 was Liu et al. which reached in 2017 Liu et al. [17] a result of 61.7%. No other results are reported by [8] or [24].

Data Set

In this section, we will describe the data set on which we evaluate our approach. There has been a multitude of Winograd data sets over the last few years. Rahman and Ng [23] present a manually created data set of 941 sentences. This data set had the caveat of being solvable without common sense reasoning.

The most recent Winograd Schema dataset is based on the work of Morgenstern et al. [19]. It contains 60 Winograd Schemas and is more difficult than the others because the example schemas have been selected with the criteria to not be solvable by statistics. This focus on difficult schemas has been done to ensure that the approaches explicitly do not use a black box or statistical models to approximate answers. This dataset has been used at the IJCAI 2016 in the last official Winograd Schema Challenge and therefore will be used in this work.

3 Approaching Winograd Schemas with Decomposition and marker passing

The best approach on Winograd Schemas so far is to train an Artificial Neural Network (ANN) as shown by Liu et al. [17]. We approach Winograd Schemas with a similar approach in two parts: First, automatically creating a semantic graph for each schema, and second, using marker passing to select the right pronoun resolution. This approach is similar to ANNs because a network is used to encode semantic features and the markers describe activation. Each node is activated like in an ANN and passes markers to the concepts it is in relation with. The markers encode symbolic information like the activation of a node in the graph, and therefore simulate neural behavior but with more detail then in an ANN like in [17]. This section describes how the semantic graph is built and how the marker passing is configured in our approach.

3.1 Decomposing

The first phase collects all information available and creates a semantic graph, which forms the knowledge base for this approach. This collection of connectionist information is called Decomposition. A Decomposition is a process of looking up a concept in the given information sources. As shown in Figure 2 the input of the decomposition is a set of information sources like WordNet, Wikipedia or domain ontologies, which then are used to build semantic graphs[3] .
The lookup of a concept is done by collecting all semantic relations known to the concepts. These relations could be, e.g., synonyms, hypernyms or meronyms. The semantic relations are completed with the concepts making up the definition of a concept, e.g., as described in WordNet. If an added concept was not present in the graph before the concepts will be decomposed iteratively until a termination criterion is met. The termination criteria were selected to be the iteration depth of two since the graph becomes intractable afterward.
This resulting semantic graph is domain specific and depends on the concepts which are decomposed. The decomposition is the process of looking up a given word in the given resources and adding all found concepts and relations to the resulting graph.

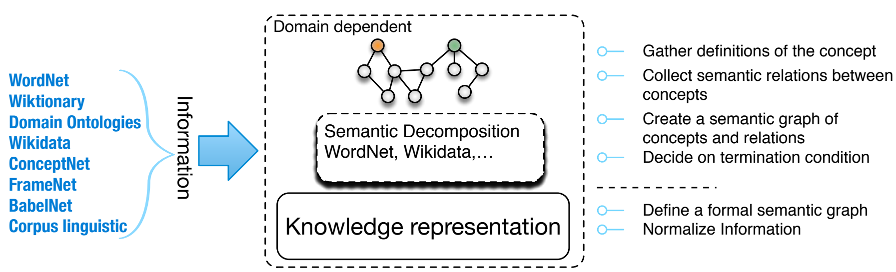

Fig. 2. Abstract description of the semantic decomposition to automatically create a semantic graph.

Depending on the decomposed concepts, the resulting graph consists of different concepts. This has been proven to be useful for different problems, e.g., semantic distance

[3] `git@gitlab.tubit.tu-berlin.de:johannes_faehndrich/semantic-decomposition.git` for access please contact the author.

measures [10]. The questions for the decomposition is now on which concepts to decompose regarding Winograd Schemas. The next subsection will explain how we created the knowledge graph which is used by the marker passing.

Decomposing Winograd Schemas At first, each word of the Winograd Schema (WGS) is decomposed, and the resulting graphs are merged. This graph forms the basis of semantic information, the facts we know about the words used in the WGS. This first decomposition thus contains all semantic information available to our approach, including synonyms, antonyms, meronyms, hypo- and hypernyms of each word. Depending on the information sources connected to the decomposition, additional concepts and relations can be part of this graph.
The result of decomposing of the word "suitcase" in our example is shown in Figure 3:

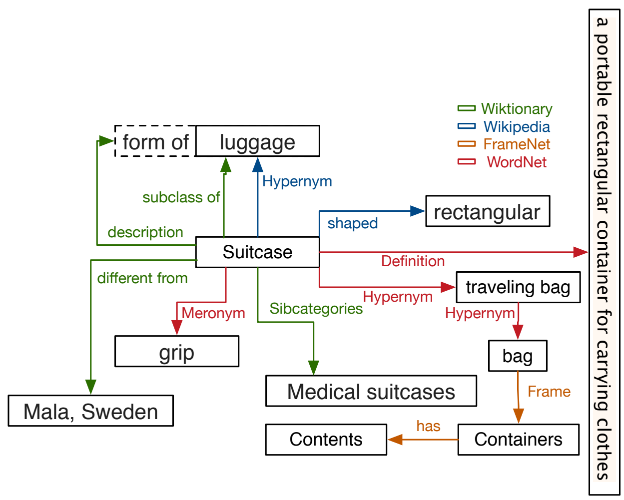

Fig. 3. Example decomposition of the word suitcase.

In Figure 3 the colors denote the source of information. The node of the graph represent concepts, and the edges the relations between them. The decomposition contains not only semantic but also syntactic information.
Now it is time to add other information which is already contained in the WGS. The next sections will describe how syntactic, semantic roles and Named Entity Recognition (NER) information is added. We start out with syntactic information.

Syntax The syntactic information contains the information which constructs sentences out of words. Without syntax the word order would not matter, inflection is ignored, and references cannot be followed. Adding syntactic information is done by using a state of the art NLP library called CoreNLP[4]. CoreNLP [18] has been chosen because of its performance on English texts regarding grammatical analysis like Part-of-Speech (POS) and Named Entity Recognition (NER) tagging as well as basic dependency identification. Figure 4 shows one example output of the CoreNLP framework for a dependency analysis of our example sentence. Here the colors describe the different POS, and the arrows describe the syntactic relations between the words. These nodes and edges are added to our decomposition graph, which includes syntactic information into the purely semantic result of the decomposition.

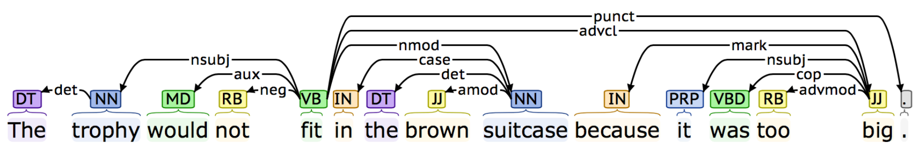

Fig. 4. CoreNLP output of a dependency tree for our example phrase.

The syntactic analysis creates an abstract syntax tree with additional nodes and relations to all words of the WGS. Those nodes and relations are added to the decomposition graph and with that allow the marker to pass over them. Enabling our approach to take syntactic relations into the count.

Semantic Roles Semantic role labeling analyses verbs and annotates roles involved. In our example the verb "to fit" has two roles: The object which is fitting and the thing it is fits into. We use PropBank[5] to determine semantic roles. We integrate those as edges into the decomposition graph. Especially for the resolution of Winograd Schemas, those roles are essential because Winograd Schemas are those border cases where the assignment of roles is ambiguous. If the program assigning semantic roles can guess right here, the battle is mostly won: In many cases, this would create a direct connection between the ambiguous pronoun and one of the answer candidates.

NER If semantic decomposition is applied to names, information enters the semantic graph that is not helpful for the resolution of the Winograd Schema (e.g., etymological information about the names of people mentioned in the sentence). This makes it necessary to recognize names entities and exclude them from the decomposition. Rather than the named entity itself the assigned named entity tag is decomposed, e.g., "Person". This leads to more semantically useful links in the graph, adding qualities of people or organizations linked to the names, which can, in turn, lead to meaningful connections to other parts of the graph.
The resulting graph decomposes all words which are not stop words and connects them to the given answers. A simplification of this graph is shown in Figure 5.

[4] `https://stanfordnlp.github.io/CoreNLP/` last visited 12.08.2018
[5] `https://propbank.github.io/` last visited 12.08.2018

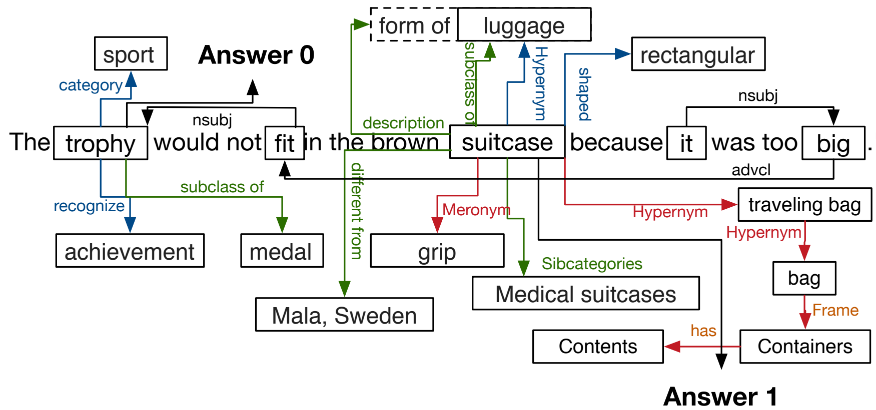

Fig. 5. Example decomposition graph of a Winograd Schema.

Figure 5 simplifies the decomposition for the ease of understanding. This graph is extended with syntactic edges discussed in the next section.
If no information source of the decomposition contains information about a given concept, the graph only contains the concept it self. If a concept has additionally no syntactic relations to the rest of the graph, the concept is not connected to the graph.

3.2 Marker Passing

Marker passing is the generalization of spreading activation [5] which models how semantic memory [4, 28] is used for reasoning in humans. One theory about reasoning in humans claims that humans think in concepts in a connectionist way [13]. Concepts are abstract representations of things, meaning less detailed, a model of which properties we remember if we think of something. This something could be e.g. a *dog*, or *love*, but in all cases it contains connections to other concepts, like *legs* or *feelings*. But a concept is abstract because it is a model of the real dog. It contains relevant information, not every hair of the dog, not every moment of feeling we had, but only those relevant. The connectionist part, is that the concept is connected to other concepts, with relations. A relation could be e.g. has as in a *dog* has four *legs*, or e.g. *is-a* as *love* is a *choice* or *feeling*. Those connections are part of the meaning of the concepts. Thus if we connect a dog with *hurt* or *fear* the meaning of *dog* is different then if we connect it with *protection* or *puppies*. Therefore, the meaning of concepts is subjective. But meaning is not only subjective but also context dependent [20, 21]. Context dependent means that regarding the current situation, the meaning of concepts changes. Good examples are ambiguous words like *bank*, where we get money or where we sit on our surfboard. The concepts and how they are connected, is given by the decomposition. Thus the semantic graph shows what was learned. The questions answered by the marker passing is: how can this graph be used to implement symbolic reasoning [3, 29]?

This principle is transferred to machine learning algorithms which then can be used on artificial semantic graphs for reasoning. The basic idea is that markers are placed on

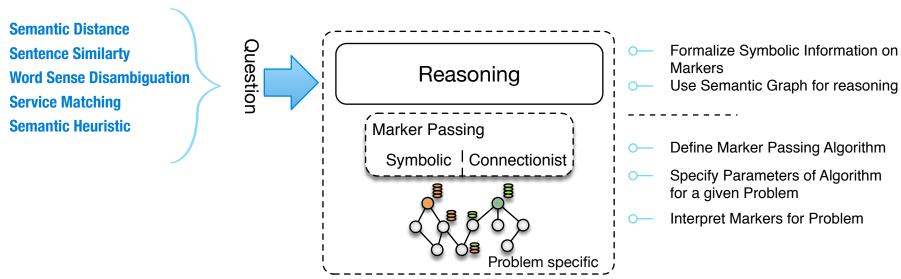

Fig. 6. Reasoning on semantic graphs.

concepts of interest in a semantic graph, and are passed over the edges connecting the concepts to other concepts. This creates a marker distribution over the graph. Such an algorithm is quite general and can be adapted to many applications. In this work we adapt it to the solution of WGSs. The markers include application-specific information like an activation level or an edge history over which the marker traveled so far. Additional application-specific properties of the algorithm are encoded in the configuration of the marker passing. Those properties could be the placement of the initial markers, or when an concept is being regarded as active. This application specificness is shown in Fig. 6: the marker passing is specialized by a questions we want to answer. For the Winograd Schemas, this question is: What is the right resolution of the ambiguous pronoun? After configuring the marker passing, the marker passing algorithm then passes markers from their start location to connected nodes, regarding those application specific rules.
Figure 7 shows the abstract marker passing algorithm. The main loop (called a Pulse) is executed until a termination condition is met. Each Pulse consists of selecting the active nodes, which pass their markers, the passing of the markers to neighboring nodes and the integration of the nodes into those neighbors.
The generic marker passing algorithm has variation points which allow specialization for different areas of application. These parameters are dependent on each other. Simple examples of such variation points are the $selectActiveConcepts$ function, the $terminationCondition$ and the $markers$: Here the $selectActiveConcepts$ function needs to interpret the $markers$ to decide if a concept is active or not. These examples show that the variation points can inter-depend. The needed variation points of the marker passing are the following:

Active Concept: is a concept which has markers on it.
Passing Concept: is a concept which has been selected by the activation function to pass markers in the next pulse.
Data: describes the marker and with that the information available to the marked node.
Pulse size: selecting which nodes pass markers.

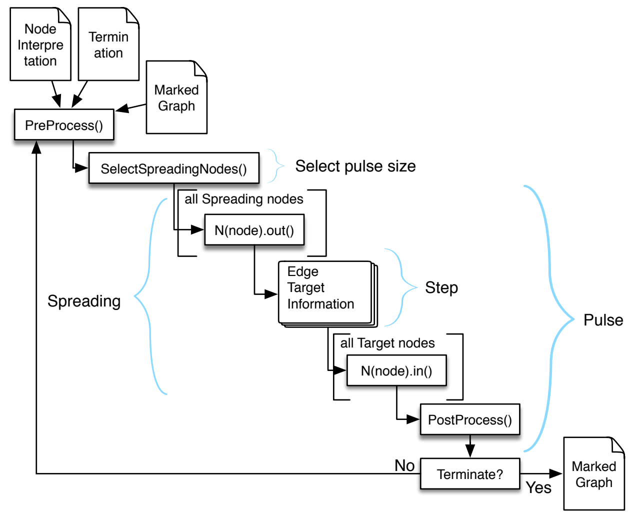

Fig. 7. Marker Passing Algorithm

In-Function: describes how the node handles incoming markers.
Out-Function and After-Send: describe how the nodes pass markers and what happens on the node after passing them.
Edge-Function: describes how the edges handle markers passed over them.
Termination Condition: describes when to stop the passing of markers.
Pre- and Post-Adjustment: describes what happens before and after a pulse.

In this way, the relevant concepts can receive markers, and after the algorithm has finished passing markers, the result can be interpreted. Algorithm 1 shows a more detailed view of the main Pulse loop, and describes our extension of the spreading activation algorithm of Crestani [5]. Active concepts are defined by getActiveConcepts().
In the abstract representation of the marker passing algorithm in Figure 7 the inputs are parts of the specification of the marker passing algorithm. In the more concrete pseudo code, in Algorithm 1 the input is a data type called "NodeData" which describes a graph created by the semantic decomposition with markers on the nodes. As input this markers are the start markers, and as output this is the resulting graph with its marker distribution. Now lets look at what the algorithm does step by step:

Line 2 - 4: All passing concepts activate their out-function and the result to the current pulse stored in the variable $pulse_{out}$. This is the input for the edge functions of the appropriate relations of the next step.
Line 5 - 9: Each marker passed by the current pulse is given to the appropriate relation it is passed to, and this relation activates its edge-function. The result of the edge-function is added to the pulse which is used as input for the in-functions of the targets of this relations.

Algorithm 1 Marker Passing Algorithm

Name: MarkerPassing **Input:** NodeData M **Output**:NodeData

1: $pulse_{out}$ = new Map<Concept,$(Edge, Markers)^* >$();
2: **for all** sourceConcept $\in$ getPassingConcepts(M) **do**
3: $pulse_{out}$.addAll(outFuntion(M, sourceConcept));
4: **end for**
5: $pulse_{in}$ = new Map<Concept,$(Edge, Markers)^* >$();
6: **for all** $e \in pulse_{out}.keyset()$ **do**
7: $pulse_{in}$.addAll(edgeFunction(M,e,$pulse_{out}.get(e)$));
8: **end for**
9: **for all** targetConcept $\in pulse_{in}.keyset()$ **do**
10: M = inFunction(M, targetConcept, $pulse_{in}$.get(targetConcept));
11: **end for**
12: **for all** sourceConcept $\in$ getPassingConcepts(M) **do**
13: M = afterSend(M, sourceConcept);
14: **end for**
15: **return** M

Line 10 - 12: The concepts which are targets of the relations passing markers are given the markers passed to them and activate their in-function.

Line 13 - 15: The after-send-function is activated to fix the markers on the source concepts if needed.

In our running example we can see in Figure 8 Step 1 that the initial markers have been placed at the pronoun. With each pulse the markers pass over the edges to other concepts, until they reach e.g. the position of step n in Figure 8.

The Step n could be one outcome of the marker passing which then needs interpretation. How the markers are interpreted is subject of the next section.

This algorithm is used to perform inference on the graphs created by the Winograd Schema. The specialization of the algorithm passes markers over all edges, weighted by edge specific weights. The parameters are chosen like in the experiment regarding a semantic similarity measure [10]. These parameters let the markers pass through the graph and propagate like measuring a semantic similarity. This lets the markers pass through, e.g., big, fit and trophy, to select an answer. The parameters specialized for this experiment with Winograd Schemas are shown in Table 1.

The parameters in Table 1 have been established in experiments where each parameter has been analyzed, and an selection has been made depending on the expressibility of the edges. The other link weights propagate markers as a factor on the activation described on the marker. With the additional links (Semantic roles, NER, Syntax,...) new weights had to be introduced. The selection of those new link weights is not an optimization on the test set, but an general setup of the approach. The same has been done for semantic distance in [10], where the weights of a general semantic distance measure are reused in our approach here. Those weights for the newly involved edges are not specific to the WSC but, e.g., can also be used for sentence similarity or Word Sense Disambiguation or the relaxation of search queries on knowledge bases [7]. The

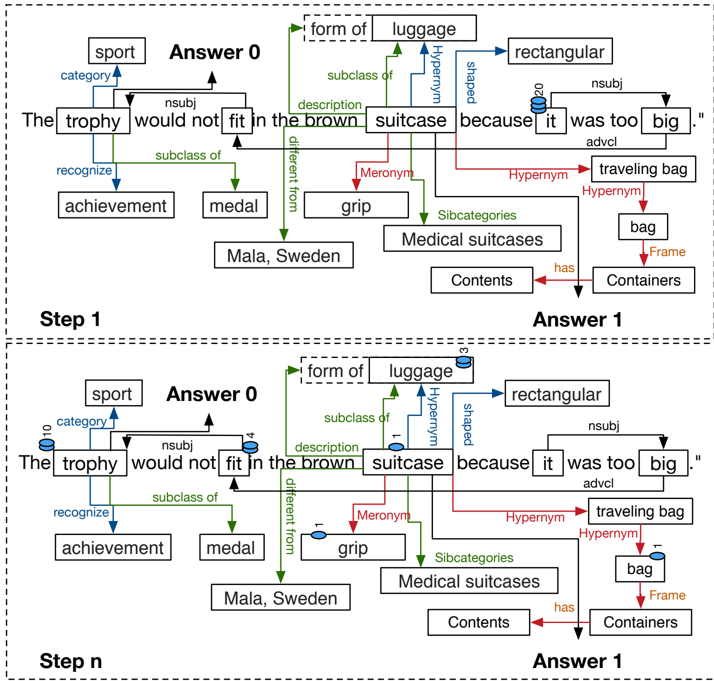

Fig. 8. Marker passing Example.

degradation factor of negative activation reduces the effect of negative activation in general.

Marker Passing Result Interpretation After the marker passing has terminated moving markers on the semantic graph, it is time to interpret the result. The interpretation of the marker distribution on the semantic graph depends on the problem solved. For our example application of the Winograd Schemas, we count the amount of markers set on each answer. We then select the answer with the most total amount of activation on all markers. This maximal activation represents a mix of semantic and syntactic distances. The distance is measured, because each time a concept is activated, its markers are split up between multiple edges. Passing the markers to multiple edges, means that ever less activation is carried on markers with each puls. This reduction of activation results in an abstract distance measure.

Here multiple interpretations have been tested: maximal average activation, maximal peak activation, the maximal sum of activation over the history of a concepts activation. The best performing interpretation was to use the marker activation divided through the total activation on all answers.

Table 1. Parameters of the marker passing algorithms which differ from [10].

Parameter	Value
degradation factor (negative)	0.05
syntaxLinkWeight	0.5
contrastLinkWeight	-0.5
NERLinkWeight	0.3
roleLinkWeight	-0.94
vnRoleLinkWeight	0.3
vnRoleLinkWeight	0.3

4 Evaluation

We evaluated our approach on the Winograd Schema Challenge [6]. For this evaluation, we have used parameters of the marker passing which have been learned for semantic similarity [10]. Those parameters are not specialized to the data set. Other parameters have been specialized, e.g., the decomposition depth, or the start marker allocation. Smaller experiments have shown that the placement of markers on the pronoun instead on the answer possibilities performs better in solving the WSC. The results of our experiments compared to the state-of-the-art are shown in Table 2 [17].

Table 2. Approaches and their performance in 2016 Winograd Schema Challenge [17, Table 4], and the result of Liu et al. [17].

Approach	Result
Our Approach	74%
Liu et al. [17]	62%
Quan Liu	58%
Nikos Isaak	48%
Patrick Dhondt	45%
Denis Robert	32%

These results show that the inference on the semantic and syntactic graph can distinguish the ambiguous pronouns. Furthermore, the interpretation of the markers lets us evaluate the ratio of the amount of activation placed on each answer possibility, which gives us insight into the confidence of our approach.

The evaluation has been done in multiple smaller experiments. The different components have been evaluated to their impact to the result. As an example the placement of the start markers. The start marker placement could be on the pronoun or an the answers. Putting the start markers on the answers, which have multiple edges in the graph, spreads the markers across the graph. This spread is caused by multiple use of the answers e.g. names and the on average high edge count of the answer nodes. This

[6] `https://cs.nyu.edu/faculty/davise/papers/WinogradSchemas/WSCollection.xml` last visited 12.08.2018

spread of markers can be reduced by placing the markers on the ambiguous pronoun, which in average has less edges then an answer node. Because the markers did not have to find the e.g. one edge connecting the pronoun to the sentence, the graph was not flooded with markers, resulting an better performance then placing start markers on the answers.
Another example of an additional experiment is the investigation of the influence of the different information sources. Regarding the Winograd Schema challenge using only WordNet, Wiktionary and Wikidata worked best.
The use of syntactic relations, in addition to the semantic relations of the decomposition connects many concepts (especially the pronoun) to the rest of the sentence. These additional relations increase the performance of the approach significantly.

5 Conclusion

The paper presents our unsupervised approach to solving Winograd Schemas. Our approach includes two parts: Part one is connectionist, where resources like Wikipedia or WordNet are used to collect semantic information about the words used and build up a semantic graph. The second part is a symbolic one, where marker passing is used to traverse the created semantic and syntactic graph. The combination of symbolic and connectionist approaches allows the approach to be adapted to multiple problems. The experiments in this paper show that it is possible to use this approach to beat the state-of-the-art in the Winograd Schema Challenge. Some of the wrongly answered schemas have properties in common. One example is the following schemas:

– Mark heard **Steve's** feet going down the ladder. The door of the shop closed after-him. He ran to look out the window.
– So Mark slept. It was daylight when he woke with **Warren's** hand uponhisshoulder.
– Papa looked down at the **children's faces**, so puzzled and sad now. It was bad enough that they had to be denied so many things because he couldn't affordthem.

All of those schemas have a "'s" in one of the answers. The apostrophe seems to confuse the syntactic analysis and the decomposition independent of the being singular or plural. On a positive note, our approach is able to handle multiple sentences, handle multi word answers like "life and soul" like in:

– Lionel is holding captive a scientist, **Dr. Vardi**, who has invented a device that turns animals invisible; Lionel plans to use it on Geoffrey and send him to steal nuclear material from an army vault.
– I sat there feeling rather like a chappie I'd once read about in a book, who murdered another cove and hid the body under the dining-room table, and then had to be the **life and soul** of a **dinner party**, withitthere all the time.

Additionally our approach still performs well if more than two answer possibilities are part of the schema. Here 67% of the schemas with three or more answers have been solved successfully.

5.1 Future Work

The here presented approach combines prior knowledge in the form of a knowledge source like the LOD, Wikipedia or WordNet with a reasoning algorithm extending modern ANN. Since the marker passing can be configured to use weights in relations, which changes the marker distribution on the graph, the weights can be learned. This learning of weights would specialize the resulting algorithm to the given problem, and most likely reduce its generality. With this loss in generality, the results would be as specific as the results produced by black box ANN approaches.
Another drawback of our approach is the dependence of the marker passing on the decomposition. Thus if the performance of the approach is not as expected, it might at first be unclear if a faulty decomposition or a misconfigured marker massing is the cause. Solving such development problems needs experience on which data sources contain which kind of information so that the decomposition can be changed to fit the needs of the problem. Additionally, the developer needs a sufficient understanding of the marker massing, and the effect of a change in parameters have on the result. During the design of the algorithm, the needed information in the decomposition can be estimated, and with that information, the needed data sources can be specified. Based on the available information in the semantic graph the marker passing can be specified.
It can be argued that the use of knowledge sources which are open to public debate include unverified information. This unverified information can lead the algorithm to be biased towards beliefs of the authors of such knowledge sources. Furthermore is the amount of knowledge sources in which a piece of information has been stated of interest to the decomposition since we do not yet identify and remove duplicate information.
Additionally, we do not use the full extent of the knowledge available, since the multi-lingual information, presented, e.g., in Wikidata is neglected. The extension to use multi-lingual information of all Wikipedia, BabelNet, and Wikidata is part of future work.
Acknowledgement: "This work is a part of the NeMo project. NeMo has received funding from the European Union's Horizon 2020 research & innovation programme under grant agreement no 713794. Content reflects only the authors' view and European Commission is not responsible for any use that may be made of the information it contains"

References

1. Arenas, M., Grau, B.C., Kharlamov, E., arnas Marciuka, Zheleznyakov, D.: Faceted search over rdf-based knowledge graphs. Web Semantics: Science, Services and Agents on the World Wide Web **37-38**, 55 – 74 (2016). https://doi.org/https://doi.org/10.1016/j.websem.2015.12.002, `http://www.sciencedirect.com/science/article/pii/S1570826815001432`
2. Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., Ives, Z.: DBpedia: A Nucleus for a Web of Open Data. The Semantic Web **4825**(Chapter 52), 722– (2007). https://doi.org/10.1007/978-3-540-76298-0-52
3. Austin, J.: Distributed associative memories for high-speed symbolic reasoning. Fuzzy Sets and Systems **82**(2), 223–233 (1996). https://doi.org/10.1016/0165-0114(95)00258-8, `http://eprints.whiterose.ac.uk/1871/1/austinj18.pdf`

4. Collins, A., Quillian, R.: Retrieval time from semantic memory. Journal of Verbal Learning and Verbal Behavior **8**(2), 240–247 (1968). https://doi.org/10.1016/S0022-5371(69)80069-1, http://linkinghub.elsevier.com/retrieve/pii/S0022537169800691
5. Crestani, F.: Application of Spreading Activation Techniques in Information Retrieval. Artificial Intelligence Review **11**(6), 453–482 (1997). https://doi.org/10.1023/A:1006569829653
6. Davis, Ernest, Morgenstern, Leora, Ortiz, Charles: The First Winograd Schema Challenge at IJCAI-16. AI Magazine **38**(3), 97–98 (Oct 2017). https://doi.org/10.1609/aimag.v38i4.2734, https://dblp.org/rec/journals/aim/DavisMO17
7. Ecke, A., Pealoza, R., Turhan, A.Y.: Similarity-based relaxed instance queries. Journal of Applied Logic **13**(4, Part 1), 480 – 508 (2015). https://doi.org/https://doi.org/10.1016/j.jal.2015.01.002, http://www.sciencedirect.com/science/article/pii/S1570868315000038, workshop on Weighted Logics for AI - 2013
8. Emami, A., Trischler, A., Suleman, K., Cheung, J.C.K.: A Generalized Knowledge Hunting Framework for the Winograd Schema Challenge. NAACL-HLT (2018), https://dblp.org/rec/conf/naacl/EmamiTSC18
9. Esteva, A., Kuprel, B., Novoa, R., Ko, J., Swetter, S., Blau, H., Thrun, S.: Dermatologist-level classification of skin cancer with deep neural networks. Nature **542**(7639), 115–118 (2017). https://doi.org/10.1038/nature21056, http://www.nature.com/doifinder/10.1038/nature21056
10. Fähndrich, J., Weber, S., Ahrndt, S.: Design and Use of a Semantic Similarity Measure for Interoperability Among Agents. In: Multiagent System Technologies. pp. 41–57. Springer International Publishing (2016)
11. Furbach, U., Schon, C.: Commonsense Reasoning Meets Theorem Proving. In: Agents and Computational Autonomy, pp. 3–17. Springer International Publishing, Cham (2016)
12. Ghallab, M., Nau, D., Traverso, P.: The actor's view of automated planning and acting: A position paper. Artificial Intelligence **208**, 1–17 (2014). https://doi.org/10.1016/j.artint.2013.11.002, http://linkinghub.elsevier.com/retrieve/pii/S0004370213001173
13. Jones, M.N., Willits, J., Dennis, S.: Models of Semantic Memory, Models of Semantic Memory, vol. 1. Oxford University Press (2015). https://doi.org/10.1093/oxfordhb/9780199957996.013.11
14. Kurzweil, R.: The singularity is near. Gerald Duckworth & Co (2005)
15. Lecue, F.: Applying Machine Reasoning and Learning in Real World Applications. Reasoning Web **9885**(5), 241–257 (2016). https://doi.org/10.1007/978-3-319-49493-7-7, http://link.springer.com/10.1007/978-3-319-49493-7-7
16. Levesque, H., Davis, E., Morgenstern, L.: The winograd schema challenge. In: Proceedings of the Thirteenth International Conference on Principles of Knowledge Representation and Reasoning. vol. 46, pp. 552–561 (2011)
17. Liu, Q., Jiang, H., Ling, Z.H., Zhu, X., Wei, S., Hu, Y.: Combing Context and Commonsense Knowledge Through Neural Networks for Solving Winograd Schema Problems. Association for the Advancement of Artificial Intelligence (2017), http://dblp.org/rec/journals/corr/LiuJLZWH16
18. Manning, C., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S., McClosky, D.: The Stanford CoreNLP Natural Language Processing Toolkit. ACL (2014), http://dblp.org/rec/conf/acl/ManningSBFBM14
19. Morgenstern, L., Davis, E., Ortiz Jr, C.: Planning, Executing, and Evaluating the Winograd Schema Challenge. AI Magazine (2016), https://dblp.org/rec/journals/aim/MorgensternDO16

20. Neely, J.H.: Semantic priming and retrieval from lexical memory: Roles of inhibitionless spreading activation and limited-capacity attention. Journal of Experimental Psychology: General **106**(3), 226–254 (1977). https://doi.org/10.1037//0096-3445.106.3.226, http://content.apa.org/journals/xge/106/3/226
21. Pace-Sigge, M.: Spreading Activation, Lexical Priming and the Semantic Web. Early Psycholinguistic Theories, Corpus Linguistics and AI Applications, Springer (Jun 2018)
22. Peng, H., Khashabi, D., Roth, D.: Solving Hard Coreference Problems. Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (2015), http://dblp.org/rec/conf/naacl/PengKR15
23. Rahman, A., Ng, V.: Resolving Complex Cases of Definite Pronouns: The Winograd Schema Challenge. In: Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning. pp. 777–789 (2012)
24. Richard-Bollans, A., Álvarez, L.G., Cohn, A.G.: The Role of Pragmatics in Solving the Winograd Schema Challenge. COMMONSENSE (2017), https://dblp.org/rec/conf/commonsense/Richard-Bollans17
25. Searle, J.: Minds, brains, and programs. Behavioral and brain sciences **3**(3), 417–424 (1980). https://doi.org/10.1017/S0140525X00005756, http://www.journals.cambridge.org/abstract_S0140525X00005756
26. Sharma, A., Vo, N.H., Aditya, S., Baral, C.: Towards Addressing the Winograd Schema Challenge-Building and Using a Semantic Parser and a Knowledge Hunting Module. In: International Joint Conference on Artificial Intelligence. pp. 1319–1325 (2015)
27. Shastri, L., Ajjanagadde, V.: From simple associations to systematic reasoning: A connectionist representation of rules, variables and dynamic bindings using temporal synchrony. Behavioral and brain sciences **16**(03), 417–451 (2010). https://doi.org/10.1017/S0140525X00030910, http://www.journals.cambridge.org/abstract_S0140525X00030910
28. Smith, E., Shoben, E., Rips, L.: Structure and process in semantic memory: A featural model for semantic decisions. Psychological Review **81**(3), 214–241 (1974). https://doi.org/10.1037/h0036351
29. Sun, R.: A connectionist model for commonsense reasoning incorporating rules and similarities. Knowledge Acquisition **4**(3), 293–321 (Sep 1992). https://doi.org/10.1016/1042-8143(92)90020-2
30. Wang, F.Y., Zhang, J., Zheng, X., Wang, X., Yuan, Y., Dai, Y., Zhang, J., Yang, L.: Where does AlphaGo go: from church-turing thesis to AlphaGo thesis and beyond. IEEE/CAA Journal of Automatica Sinica **3**(2), 113–120 (2016). https://doi.org/10.1109/JAS.2016.7471613
31. de Winter, J., Dodou, D.: Why the Fitts list has persisted throughout the history of function allocation. Cognition, Technology & Work **16**, 1–11 (2014). https://doi.org/10.1007/s10111-011-0188-1, http://dx.doi.org/10.1007/s10111-011-0188-1
32. Yamaguchi, A., Kozaki, K., Yamamoto, Y., Masuya, H., Kobayashi, N.: Semantic Graph Analysis for Federated LOD Surfing in Life Sciences. JIST **10675**(5), 268–276 (2017). https://doi.org/10.1007/978-3-319-70682-5-18, http://link.springer.com/10.1007/978-3-319-70682-5-18
33. Yampolskiy, R.: AI-Complete, AI-Hard, or AI-Easy - Classification of Problems in AI. In: Twenty-third Midwest Artificial Intelligence and Cognitive Science Conference. pp. 94–101 (2012), http://ceur-ws.org/Vol-841/submission_3.pdf