The AAAI 2017 Spring Symposium on
Computational Construction Grammar and Natural Language Understanding
Technical Report SS-17-02

Semantically-Driven Coreference Resolution with Embodied Construction Grammar

Vivek Raghuram, Kelly Shen, Ethan Goldberg, Sidney Oderberg
International Computer Science Institute
Berkeley, CA 94704

Sean Trott
University of California, San Diego

Abstract

Reference resolution is a challenging problem in Natural Language Understanding and is the focus of Winograd problems, which pose particularly difficult instances of this task (Levesque 2011). We demonstrate solutions to several difficult reference resolution tasks posed by Winograd problems using Embodied Construction Grammar and the ECG 2.0 system (Raghuram et al. 2016). We are able to take advantage of ontological constraints, grammatically marked properties, and relationships between frames in order to accurately perform reference resolution. By addressing these problems we gained valuable insights to guide future development on the ECG 2.0 system.

Introduction

Many efforts in Natural Language Processing focus on statistical learning and do not integrate insights from research on how humans process and understand language. Fortunately, however, Natural Language Understanding (NLU) is moving towards models that are rooted in studies of language and cognition, as well as metrics that aim to objectively measure human-like language understanding in machines.

One example of such a metric is the Winograd Challenge (Levesque 2011). The challenge arose in response to growing criticism of the Turing Test as a measurement of machine intelligence. Rather than assessing intelligence and language understanding by observing interactions with a human interlocutor, the Winograd Challenge requires a computer to resolve an ambiguous pronoun with its antecedent. Consider the sentence:

The city council refused to grant the protesters a permit because they (feared/advocated) violence.

Here, "they" can be alternately co-indexed with "the city council" or "the protesters", depending on which of the words in parentheses is chosen.

Crucially, these cases cannot be resolved with purely syntactic or statistical models; reference resolution requires "common sense" or "world knowledge". The primary advantages of this challenge over the Turing Test are:

1. It allows an *objective* measurement of the machine's competence, as opposed to a subjective measurement by a human judge.
2. The nature of the task requires a way to represent knowledge and perform reasoning, whereas contestants in the Turing Task can recycle canned phrases without "understanding."

However, there are disadvantages as well. Beyond simply the difficulty of crafting the sentences, these types of challenges often encourage models geared toward solving a narrow task (e.g. reference resolution), rather than generalizable models of language use and understanding. The end result, then, is a program that performs well on the Winograd Challenge but is not necessarily useful for other NLU domains, and does not necessarily inform our knowledge of language understanding in general.

We take a contrasting approach. We have developed a generalizable system of language understanding that can be extended to various domains, such as robotics (Trott, Eppe, and Feldman 2016) and computer gaming (Eppe et al. 2016). This system uses Embodied Construction Grammar, a neurologically plausible model of language use (Bergen and Chang 2005). In extending our system to address the challenge, we arrived at several valuable, more general insights, which inform both current and future approaches to construction grammar engineering and reference resolution.

Background: ECG 2.0

The ECG 2.0 system includes a generalizable framework for natural language understanding, as well as software tools to help extend this framework to various domains and applications. Previous papers have described this system in considerable detail (Eppe et al, 2016); this paper will focus on the two components of the framework that are most relevant to the Winograd Challenge: 1) Embodied Construction Grammar (ECG); and 2) the Specializer. Our current release of the ECG 2.0 system can be found on Github[1] (Raghuram and Trott 2016).

[1] https://github.com/icsi-berkeley/ecg_homepage/wiki

Embodied Construction Grammar

Embodied Construction Grammar (ECG) is a formalism for describing how language is used, and how that usage creates meaning (Bergen and Chang 2005) (Feldman 2010). Like other construction grammars, it rests on an argument that language consists of form-meaning pairs. In ECG, meaning is represented by embodied schemas; the collected schemas form a lattice of embodied world knowledge. These schemas are activated in constructions, which describe how the constituents of grammatical patterns map to elements from the schemas. A set of constructions and schemas constitute a grammar. We developed a "core" grammar, which contains constructions and schemas that are domain-independent; this grammar can be used as a scaffolding for building out grammars for new domains and contexts.

Importantly, ECG is computationally implemented. The ECG Analyzer (Bryant 2008) is a best-fit constructional parser that uses an ECG grammar to produce a semantic analysis of an input sentence. This semantic analysis is called a Semantic Specification (SemSpec), and contains information about how words in the sentence activate meaningful bindings in the embodied schemas. Roles bound to the same value are co-indexed to reveal the relationships within the SemSpec. An example SemSpec for the sentence "Jane thanked Mary" can be seen in Figure 1.

Core Specializer

The SemSpec produced by the ECG Analyzer is a rich data structure containing both constructional and semantic information about a sentence, but it does not contain any information about the wider discourse context (e.g. previous sentences that have been uttered), nor does it attempt to resolve pronouns with their antecedents. Instead, pronouns are marked as referring to some unknown antecedent; the resolution process is done by the Core Specializer.

The Core Specializer receives a SemSpec from the ECG Analyzer and is responsible for performing several key functions:

1. Extracting task-relevant information: the SemSpec often contains information that is unnecessary to solve a task.
2. Keeping track of the wider discourse context and entities that have been referenced in previous sentences.
3. Performing anaphora resolution.

The Specializer produces a data structure called an Action Specification (ActSpec) - this is a dictionary-like structure that can be used for various tasks, such as controlling a robot, answering queries, performing inference, and more. Like the ECG Analyzer and core grammar, the Core Specializer is domain-independent; if needed, domain-dependent changes can be accomplished through the use of declarative templates, which point out which information is necessary to extract for the ActSpecs (Raghuram and Trott 2016).

The anaphora resolution system is able to resolve pronouns both within and across sentences. As discussed in previous work (Trott, Eppe, and Feldman 2016), (Trott et al. 2015), (Oliva et al. 2013), the system exploits semantic constraints supplied by the ECG grammar, as well as heuristics about syntactic patterns, to often correctly identify the antecedent of a pronoun.

Figure 1: SemSpec for the sentence, "Jane thanked Mary"

Until now, the reference resolution system has been primarily used in the context of human-robot dialogue; below, we discuss how the Winograd Challenge helped us identify its limitations, which in turn has led to a more robust anaphora resolution process.

Taxonomy of Winograd Problems

Winograd problems can be classified by the fundamental, often implicit constraints that make them intuitively solvable by humans. Most Winograd problems involve some combination of these core issues. We identified some of the constraints that appear most frequently across previously published Winograd problems (Davis 2011). Using these foundational issues, the problems were categorized so as to determine the components necessary to achieving a valid reference resolution. Special attention was paid to the problems which are resolvable using minimal world knowledge. The core issues in our taxonomy are as follows:

Agent/Patient Distinction A single object cannot serve as both the agent and patient of a non-reflexive action. Enforcing this assertion frequently condenses the set of available assignments of pronouns to antecedents.

Example: *"Sid explained his theory to Mark but he couldn't understand him."*

"He" and "Him" must refer to different antecedents. As a result, assigning one forces the assignment of the other.

Ontological Congruence The agent/patient must be ontologically compatible with its corresponding verb. In other words, a noun must have a meaning and linguistic category that makes sense in its role as an agent or patient for its verb.

Example: *"Anne gave birth to a daughter last month. She is a very charming baby."*

The patient of giving birth must be a baby. The second sentence tells us that "She" is a baby so "She" must reference the daughter.

Agent and Patient Properties The agent and/or patient may have properties that make it specifically consistent with a verb. Such properties may not be defined within the word's ontology, but are instead denoted by information contained within the evoked schema.

Example: *"The older students were bullying the younger ones, so we rescued them."*

The first clause designates the younger students as being in danger. This classification disambiguates the pronoun as the more valid object of the verb "rescue."

Relationship Frames Some Winograd problems evoke common frames specifying the relationships between the agent and patient. Some frequently appearing frames include spatial relationships, commercial transactions, food consumption, and ownership of objects.

Example: *"Joan made sure to thank Susan for all the help she had received."*

Resolving the pronoun "she" requires an understanding of the relationship between a giver and receiver of a favor, which can be described within a frame akin to the commercial transaction frame.

Scale Comparison The correct assignment of a pronoun may necessitate the comparison of a particular scalar value held by multiple antecedents. Common simple scalars include height, speed, and weight.

Example: *"The delivery truck zoomed by the school bus because it was going so fast."*

The pronoun refers to the antecedent that is traveling at a higher speed. The phrase "zoomed past" implies that the truck is moving faster than the bus, allowing the schema to be resolved using a comparison of speed.

Our Approach

After analyzing the Winograd problems, we determined three main ways to address a subset of the problems they present using the ECG 2.0 system: enforcing ontological constraints, using grammatically marked object property constraints and using the novel idea of bridging schemas.

```
schema CreateObstacleSchema
  subcase of CausalTransitiveAction
  roles
    portal: RD
  constraints
    portal <--> patient
    causalAgent.ontological-category <-- @removableEntity
    portal.ontological-category <-- @portal


schema RemovingSchema
  subcase of TransitiveAction
  roles
    remover
    obstacle: RD
  constraints
    protagonist <--> remover
    patient <--> obstacle
    obstacle.ontological-category <-- @removableEntity
```

Figure 2: Schemas used to encode obstacles and the removal of an obstruction

Additionally, we implemented a simple rule to keep distinct the agents and patients of non-reflexive actions. We applied these techniques to simplified versions of the Winograd problems. Sentences were simplified to remove extraneous language but retain the same meaning and difficulty in the reference resolution task. All example sentences presented in the subsequent sections run in the existing system.

Ontological Constraints

Problems posed by ontological congruence and agent and patient properties can be addressed by taking into account the ontological categories of the referents and constraints imposed by the evoked ECG schema. The ECG analyzer can recognize categories that are stored in an ontological lattice. By adding the right categories to the lattice and constraining the roles of an ECG schema to those categories, we can provide the system with enough information to resolve the references. We can then use the requirement for ontological congruence to guide reference resolution in the specializer. Similar kinds of constructional constraints have been used in understanding and learning languages like Mandarin where omitted material is common (Mok and Bryant 2006).

Consider the sentence, *"The hair clogged the drain; John [cleaned/removed] it."* Here the choice of the action "cleaned" or "removed" will cause "it" to refer to either the drain or the hair respectively. Were a human to read this sentence, they would easily know how to resolve the reference because of their understanding that a clogged portal can be cleaned, whereas a clogging entity can be removed from its corresponding portal. This understanding can be encoded in schemas for an obstacle (figure 2).

If the sentence discusses removing, it invokes the RemovingSchema (figure 2) for removal. "It" is the patient of the RemovingSchema and so must be a removableEntity. Therefore "it" can only refer to the causalAgent of the CreateObstacleSchema. Using this constraint, we utilize the specializer to correctly resolve the pronoun against the appropriate referent in the specializer. The process similarly works if the

```
schema RelativeScale
  subcase of Relation
  roles
    larger
    smaller
    property
  constraints
    larger <--> entity1
    smaller <--> entity2
    relation <--> property

schema ContainSchema
  subcase of Motion
    evokes SPG as spg
    evokes RelativeScale as rsc
  roles
    container: RD
    contents: RD
  constraints
    container <--> rsc.larger
    contents <--> rsc.smaller
    rsc.property <-- @size
    contents <--> mover
    container <--> spg.landmark
    container.ontological-category <-- @container
    contents.ontological-category <-- @physicalEntity
```

Figure 3: Schemas used to encode containment and relative size

sentence discussed cleaning.

We also applied this process to the sentence *"Jane played the song with the flute ; she [loved/had] it."* Here we made songs both valuable things and also abstract things. Therefore, one can love the song but not possess it.

Grammatically Marked Properties

Many Winograd problems, including those involving a scale comparison, provide the necessary properties for reference resolution through the language in the sentence. These grammatically marked properties enable some basic inference to accurately perform the reference resolution process.

We focused on problems that involved a relative scale. Here two objects have a property measured on a scale (e.g. speed, size) where one is higher on the scale than the other. Using this concept we produced the RelativeScale schema which relates the properties of two objects.

Consider the sentence, *"Jane passed Mary ; she was too [fast/slow]."* "Passing" Mary invokes a RelativeMotion schema which in turn evokes the RelativeScale schema (figure 3) using the speed property. We say Jane has a larger speed and Mary has a smaller speed. Therefore, the one who was fast or slow must be the one who was relatively fast or relatively slow.

This type of inference can be made robust to negation as well. Consider the sentence, *"The trophy does not fit into the suitcase ; it is too [small/large]."* This sentence evokes a schema for Containment (figure 3) where the container must be larger than the contained object. We know the pronoun "it" must be relatively small (or large).

The sentence works similarly to the previous example, however here the event is negated. The trophy does *not* fit. We infer that this is caused by the relative size constraints being violated. Therefore, during reference resolution, we resolve values in order to violate the constraints (rather than to meet them). With this implementation in place, the specializer accurately resolves "it" to the suitcase if "it" was too small or to the trophy if "it" was too big.

Bridging Schemas

```
schema BridgeSchema
  roles
    bridgeKind: @bridgeKind
    bridgePatient
    bridgeAgent
    bridgeTheme

schema GratitudeSchema
  subcase of TransitiveAction, BridgeSchema
  roles
    actionary: @thank
    theme: RD
  constraints
     bridgePatient <--> agent
     bridgeAgent <--> patient
     bridgeKind <-- @thanks

schema ObjectTransfer
  subcase of ComplexProcess, A123, TransitiveAction
  roles
   process1: ReleaseHold
   process2: EstablishHold
   giver
   getter
   theme: RD
   actionary: @objecttransfer
   constraints
     agent <--> giver
     patient <--> getter
     process1.actedUpon <--> theme
     process2.actedUpon <--> theme
```

Figure 4: The Gratitude schema inherits the BridgeSchema, allowing it to bind multiple related schemas

A large number of Winograd problems involve relationship frames that are related to other frames. The relationships between the frames are what allow the bindings to be resolved. For example, consider the situation where Jane thanks Mary for a gift. The act of Jane thanking Mary is related to an earlier object transfer with Mary giving a gift to Jane. Therefore the receiver of the object transfer is the thanker and the giver is the one being thanked. To address this situation, we added a new kind of schema to the grammar and specializer called the BridgeSchema (figure 4). This general use schema is intended to bridge between two related ECG schemas.

A "bridging schema" is one that inherits from the special schema called the BridgeSchema. Inheriting from BridgeSchema acts as a hint to the specializer to try to match the bridging schema with another compatible schema if available. The schema with which a bridging schema is successfully matched is called the "matched schema". Additionally inheriting from BridgeSchema provides a number

of roles which relate the roles of the bridging schema and the matched schema.

Consider the sentence "Jane thanked Mary for flute that had been given [to/by] her." As explained earlier, we know from our understanding of gratitude that Jane is thanking Mary for an action performed by Mary which had Jane as the patient of the action. This knowledge can be formalized in a Gratitude schema that captures both the meaning of gratitude as well as how the members of a gratitude act would be related in a prior motivating act. The GratitudeSchema is a bridging schema and so we bind participants to the roles provided by the BridgeSchema. Notice the agent is bound to the bridgePatient and patient is bound to the bridgeAgent. The reason for this switch will become apparent in the specializer. For now, the analyzer is able to parse this sentence using the GratitudeSchema and the ObjectTransfer schema and produce semspecs without any of the references resolved.

The specializer performs the reference resolution by matching the GratitudeSchema with the ObjectTransfer schema. This is done by crawling the semspec to find all bridging schemas and then greedily matching them to the first compatible schema in the semspec. Matching is conducted by first looking at the bridgeKind field of the bridging schema. This field indicates, via a mapping, what general class of schemas to attempt matching with. For example, the bridgeKind, "thanks", maps to TransitiveAction so only schemas that are subcases of transitive action will be considered. Next, the roles inherited from BridgeSchema are paired with the corresponding roles in the candidate schema. For example, the bridgeAgent role is paired with the agent role in ObjectTransfer. From there the roles are checked to ensure they do not have conflicting values and are ontologically compatible. If all checks are passed, the candidate schema is the matched schema and the references are resolved by unifying the paired roles. For example, the value of agent is set to that of bridgeAgent. In the case of the sentence above, the specializer matches GratitudeSchema with the ObjectTransfer schema, allowing us to determine that the giver must have been Susan and the receiver was Joan.

In this case the bridging schema contained the referents. However, this method also works to resolve references that appear in the bridging schemas themselves. Consider the sentence, *"Jane asked Mary a question; She [answered/repeated] it."* In this case, we know the act of answering necessitates an earlier Communication where the listener of the answer is the speaker. Therefore we create a DiscourseAnswer schema that is a subcase of BridgeSchema with the agent and patient reversed as before. The specializer then matches the DiscourseAnswer against the initial Communication schema. From there the pronouns in DiscourseAnswer are resolved by binding them to paired roles in the Communication schema. The process works similarly in the case of repeating the question.

The BridgeSchema represents a powerful new tool introduced to the system and is an area of active exploration.

Related Work

Solving General Winograd Problems

A key feature of Winograd problems is that they must be resolved through semantics, not syntax. Any machine that solves these kinds of reference resolution problems must be able to determine the semantic content of the sentence as well as have a sophisticated world knowledge in order to correctly determine the bindings. Developing these abilities is a very difficult problem in itself and is the focus of most attempts to address Winograd problems. Several approaches involve querying the world-wide web for relevant information to help in understanding the sentence. This knowledge is often taken from other sources and combined with other knowledge acquired through syntactic and semantic analysis as well as machine learning models (Rahman and Ng 2012; Sharma 2014; Sharma et al. 2015; Peng, Khashabi, and Roth 2015). While these attempts have made progress to more robust reference resolution systems, they are still far from perfect.

Winograd as a Motivating Problem

Winograd problems have also been used to inform a discussion about novel reference resolution techniques. Schüller 2014 showed how relevance theory can be used to address a number of Winograd problems. Bailey et. al 2015 demonstrated how to use correlation between sentences to justify solutions to Winograd problems. These discussions are mainly theoretical and are not attempts to solve the general challenge. This is similar to our approach to Winograd problems as a motivation to improving reference resolution in the ECG 2.0 system.

Conclusion

The design of the ECG 2.0 system enables solving difficult reference resolution problems. We demonstrate this ability by using Winograd problems as a target domain to be solved by the system. We classified the challenges posed by the Winograd problems into different categories and then developed solutions to address several of them. With no changes to the core system we are able to take advantage of ontological constraints. Simple extensions of the core specializer enabled taking advantage of grammatically marked properties for scale inference and using Bridging Schemas to connect the roles of related frames. Preliminary implementations of Bridging Schemas demonstrate how powerful they can be in the reference resolution process.

Limitations

ECG 2.0 is a framework for inherently domain specific systems and so will never be a universal solution to the Winograd Challenge. However, by capturing all the language of a specific domain, our system can accurately solve difficult reference resolution tasks. More work is needed to determine the generality of the solutions described in this paper.

As we have discussed, some of the Winograd problems are solvable only by having a base of world or cultural knowledge. Consider the following example: *"In the middle*

of the outdoor concert, the rain started falling, [and/but] it continued until 10. What continued until 10? In order for a language understanding system to recognize the subtle difference in using "and" or "but" and then pick the correct noun, it would need to know about rain and what it means to be outdoors. The requisite world knowledge is that people attending outdoor concerts are exposed to falling water and that this is an undesired condition. While some of the world knowledge problems may be solvable by just adding to the vocabulary, ones like these require understanding people's preferences.

Future Work

Further Development on Bridging Schemas In their current state Bridging Schemas have proven to be a powerful tool in reference resolution. As it stands we only have one kind of Bridging Schema that satisfies the needs of the examples shown above. However, we believe that ultimately we will need a few general types of Bridging Schemas.

Additionally, Bridging Schemas introduce their own problem akin to reference resolution. Matching a Bridging Schema to another schema is currently done by checking the compatibility of their respective roles. However, there may be more sophisticated ways to find the appropriate match.

We believe that further development on Bridging Schemas will yield better solutions to these problems.

Understanding Actions Schemas by Using X-nets Not all Winograd problems can be resolved simply through reference resolution. Some involve being able to understand the procedures and outcomes of a particular action. Take, for example, the following sentence: *"The dog chased the cat, which ran up a tree. It waited at the [top/bottom]. Which waited at the [top/bottom]?"*. Without understanding what it means to "run up" a tree, our existing system will have no notion of where the agent and patient of the sentence should be located at the end of the action.

Our approach to adding this ability to understand action will be the integration of X-nets with the ECG system, where X-nets are an action-focused extension of the general Petri net state model (Doubleday, Trott, and Feldman 2017). With the addition of X-nets, the ECG system will be able to run a simulation of the actions found in the Winograd problems. For the given example, a simulation of the cat running up the tree would result in a model of the outcome: the cat has gone from the ground to the top of the tree and the dog has remained on the ground. Then, whether the question asks which animal is at the top or which is at the bottom, our system will be able to use the resulting model to provide the correct answer.

This approach, in terms of how it fits in with Natural Language Understanding, is justified by findings on human action simulation (Bergen and Wheeler 2010). It was found that mental simulations are run when processing sentences involving actions. These simulations are produced by some of the same parts of the brain that are active when humans would actually perform the action. As such, we believe that running action simulations is key for our system to truly understand action-based language.

Acknowledgements

We thank Professor Jerome Feldman for all his guidance and support, and the reviewers for their thoughtful suggestions.

References

Bailey, D.; Harrison, A.; Lierler, Y.; Lifschitz, V.; and Michael, J. 2015. The Winograd Schema Challenge and Reasoning about Correlation. (Association for the Advancement of Artificial Intelligence).

Bergen, B. K., and Chang, N. 2005. Embodied Construction Grammar in Simulation-Based Language Understanding. *Construction grammars: Cognitive grounding and theoretical extensions* 147–190.

Bergen, B., and Wheeler, K. 2010. Grammatical aspect and mental simulation. *Brain and Language* 112(3):150–158.

Bryant, J. E. 2008. Best-Fit Construction Analysis. *Analysis*.

Davis, E. 2011. Collection of winograd schemas.

Doubleday, S.; Trott, S.; and Feldman, J. 2017. Processing Natural Language About Ongoing Actions in ECG. *AAAI Spring Symposium*.

Eppe, M.; Trott, S.; Raghuram, V.; Feldman, J.; and Janin, A. 2016. Application-Independent and Integration-Friendly Natural Language Understanding. *EPiC Series in Computing* 41(GCAI 2016. 2nd Global Conference on Artificial Intelligence):340–352.

Feldman, J. 2010. Embodied language, best-fit analysis, and formal compositionality. *Physics of Life Reviews* 7.

Levesque, H. J. 2011. The Winograd Schema Challenge. *Artificial Intelligence* (1989):63–68.

Mok, E. H., and Bryant, J. 2006. A Best-Fit Approach to Productive Omission of Arguments. *In proceedings of the Berkeley Linguistics Society*.

Oliva, J.; Feldman, J.; Gilardi, L.; and Dodge, E. 2013. Ontology driven contextual best fit in embodied construction grammar. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 8114 LNCS:133–151.

Peng, H.; Khashabi, D.; and Roth, D. 2015. Solving Hard Coreference Problems. *Proc. of NAA* 61801:809–819.

Raghuram, V., and Trott, S. 2016. Ecg homepage. https://github.com/icsi-berkeley/ecg_homepage/wiki, accessed 09/27/16.

Raghuram, V.; Shen, K.; Goldberg, E.; Oderberg, S.; and Trott, S. 2016. Ecg winograd repo. https://github.com/icsi-berkeley/ecg_winograd, accessed 09/27/16.

Rahman, A., and Ng, V. 2012. Resolving complex cases of definite pronouns: the winograd schema challenge.

Schüller, P. 2014. Tackling Winograd Schemas by Formalizing Relevance Theory in Knowledge Graphs. *International Conference on Principles of Knowledge Representation and Reasoning (KR)* (1):358–367.

Sharma, A.; Vo, N. H.; Aditya, S.; and Baral, C. 2015. Towards addressing the winograd schema challenge - Building and using a semantic parser and a knowledge hunting

module. *IJCAI International Joint Conference on Artificial Intelligence* 2015-January(July):1319–1325.

Sharma, A. 2014. Solving Winograd Schema Challenge: Using Semantic Parsing, Automatic Knowledge Acquisition and Logical Reasoning. (October).

Trott, S.; Appriou, A.; Feldman, J.; and Janin, A. 2015. Natural Language Understanding and Communication for Multi-Agent Systems. *AAAI Fall Symposium* 137–141.

Trott, S.; Eppe, M.; and Feldman, J. 2016. Recognizing Intention from Natural Language : Clarification Dialog and Construction Grammar. *Workshop on Communicating Intentions in Human-Robot Interaction*.