

Generating Relevant Referring Expressions with GAIA: A Givenness Advised Incremental Algorithm

Mark Higger
mhigger@mines.edu
Colorado School of Mines
Golden, Colorado, USA

Tom Williams
twilliams@mines.edu
Colorado School of Mines
Golden, Colorado, USA

ABSTRACT

Referring Expression Generation (REG) is the language generation task of selecting attributes to refer to a target entity. While REG is well-studied in linguistics, its introduction into robotics brings new challenges. For real-world robotic environments, robots may have access to a multitude of irrelevant objects that exist outside the scope of conversation, and traditional REG disambiguates the target referent from all other entities, regardless of relevance. While some newer REG methods take relevance into consideration, they are largely limited to potential referents that are part of the same conversation. In this work, we propose using cognitive statuses to inform the relevance of each entity for REG, narrowing down possible distractors based on cognitive relevance introducing our Givenness Advised Incremental Algorithm (GAIA) which leverages cognitive status for REG. This allows a flexible and enhanced REG, accounting for the context of entities both inside a conversation and within the larger scale environment.

CCS CONCEPTS

• **Computer systems organization** → **Robotics**; • **Computing methodologies** → **Natural language generation**; **Natural language processing**.

KEYWORDS

natural language generation, referring expression generation

ACM Reference Format:

Mark Higger and Tom Williams. 2024. Generating Relevant Referring Expressions with GAIA: A Givenness Advised Incremental Algorithm. In *Companion of the 2024 ACM/IEEE International Conference on Human-Robot Interaction (HRI '24 Companion)*, March 11–14, 2024, Boulder, CO, USA. ACM, New York, NY, USA, 4 pages. <https://doi.org/10.1145/3610978.3640631>

1 INTRODUCTION

Referring expression generation (REG) is “the task of selecting words or phrases to identify domain entities” [15]. REG is a very distinct and one of the most well-studied tasks in natural language generation (NLG)[19]. While there are many approaches to REG, the motivation behind how most of these approaches work is rooted in the Gricean Maxims, which describe how and why we choose

the language we use to communicate. These maxims, while notoriously not formalized [19], consist of quantity, how much is said, quality, or how truthful the words are, relation, or how relevant the words are to the conversation, and manner, or how clearly the information is conveyed [8]. Early REG algorithms, such as Full Brevity [3] focused on optimizing for Gricean quantity while holding the other maxims true by choosing the minimum number of descriptors needed to uniquely identify the target referent. It does this by finding which attributes narrow down the most number of other distractors and choosing these attributes iteratively. However, there are fundamental issues with this approach [19]. The first is that choosing the minimum number of attributes is an NP-hard problem and becomes very complex computationally as the number of potential distractors increases. The second, and perhaps more important factor is that there is substantial evidence that humans fundamentally do not generate the minimum number of descriptors, but rather generate descriptors based on latent preference for attributes [13].

This idea of preference has led to the gold standard for REG, the Incremental Algorithm (IA) [15]. IA mechanically works very similar to Full Brevity, opting to minimize the number of words said, satisfying the Gricean maxim of quantity, however, it does so with the additional constraint of preference order. That is, it goes through each attribute in preference order and finds which attributes narrow down potential distractors and only adds those attributes to the expression if they narrow down potential distractors. In this way, it satisfies the constraint of the Gricean maxim of quantity, while taking into account latent preference. While this algorithm is the standard for REG, it has many major flaws and limitations. One of these limitations of IA is that it always assumes a one-shot referring expression that is generated without context and consequently generates a unique description of a target entity in comparison to *all* other entities. This is problematic in two different, but also similar ways. The first is that for very large sets of entities, many entities may not be relevant at all to the current conversation, but using IA would still compare a target entity to *all* other entities, including irrelevant ones. Take the example of two conversants in a classroom talking and one asks another for a whiteboard marker. In this scenario, since there are whiteboard markers in many rooms, IA would generate an expression like “The {whiteboard} {marker} {in classroom 208}”. In this case, it would be unnecessary to specify that the marker lies in classroom 208, because it is implied by the fact that both conversants are already in classroom 208, and know that a black marker exists there. While this type of context is obvious to humans it becomes a significant challenge in larger-scale robotics where a robot may be aware of a considerable number of similar, but irrelevant entities.



This work is licensed under a Creative Commons Attribution International 4.0 License.

A similar contextual issue is that within a single conversation entities may require fewer descriptors if they have already been mentioned. For example, a block-movement task, where a robot is asking a human to move block [b] to another location. In this case, the robot might want to find the best way to generate a description of the block for the following utterances:

U1: “I want you to grab [b]”

U2: “and move [b] over there”

If using IA to generate the description for [b] the robot might find that the best description is “green round block”, creating the utterances “I want you to grab the *green round block*” and “and move the *green round block* over there”. Because [b] is directly repeated and at the center of attention, a more natural referring expression for U2 might be “and move *that block* over there”. Notably in this example, the attributes used for the description of [b] were reduced from green, round, block, to just block. This means that by directly using IA we are adding unneeded and irrelevant information about [b] in U2, violating the Maxim of quantity and relation. In this paper, we argue that both the contextual issue of narrowing down descriptors for repeated entities as well as the larger scale issue of comparing to irrelevant entities are directly tied together. Furthermore, both of these issues can be directly addressed by utilizing cognitive models of the entities to narrow down the relevant entities used under consideration in IA.

2 RELATED WORKS

2.1 Referring Expression Generation

Referring expression generation is an extremely well-studied field in linguistics. In particular, the Incremental Algorithm (IA) is a standard REG algorithm that iteratively goes through properties of known entities to create a description of a target referent using the unique set properties that best differentiate the referent from all other entities [3, 14]. One of the major drawbacks of IA is that it is only designed for a one-shot referring model. That is, it does not take into account the context of the conversation for repeated entity references, potentially creating overly specified and unnatural references. Another major drawback of IA is that it must compare the target referent to all other known entities, often limiting its use to smaller closed-world environments [11]. This may be acceptable in limited linguistic tasks where known entities are confined to a specific conversation or text, but is not acceptable in larger-scale human-robot interactions, where a robot may know about a wide variety of objects, many of which may not be relevant to the current conversation.

This issue has been widely noted in the computational linguistics community and is addressed by moving away from IA and instead using methods specifically designed for contextualized REG. Historically one of the ways this has been achieved is using accessibility theory [1, 2, 7], which uses discreet levels of accessibility for particular linguistic markers to determine how accessible a particular entity is. For example, a pronoun or full name might relate to a ‘high’ level of accessibility [12]. By modeling accessibility in this way, references can be generated that utilize the context of the conversation rather than just attributes to generate more reasonable referring expressions. More recently, the computational linguistics community has moved to more machine learning and deep learning-based

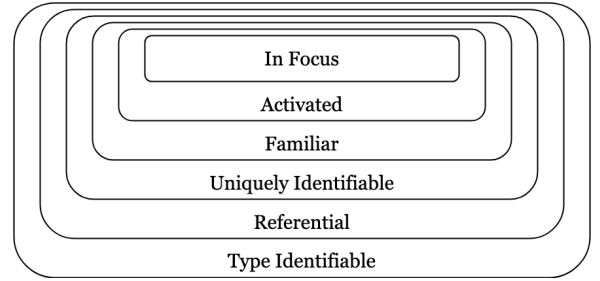


Figure 1: Graphical representation of the Givenness Hierarchy higher levels of cognitive status are a subset of all lower statuses

methods. Specifically, many modern REG algorithms utilize linguistic features, such as word choice and recency for learning-based methods [6, 16, 17]. While both accessibility and learning-based methods can address the problem of short-term repetition of reference, they are not scalable to larger contexts. For example, a robot in a long-term care facility might want to refer to a particular coffee mug. In this case, these current REG algorithms might compare to every mug in the robot’s knowledge base, which includes all the mugs that might exist in a large facility, most of which will be completely irrelevant to the robot’s current situation. Some of the learning methods might take into account how recently the mug was last referred to and be ‘smart’ enough to differentiate it from only recently mentioned mugs. However, in the case where the mug hasn’t even been mentioned yet, the robot may need to differentiate the target mug from every other mug it is aware of, even if the target referent is the only mug in the room. While it may be possible to expand the feature space of the REG methods to include physical features such as distance and retrain them to incorporate these new features, there may be a more fundamental, simple, and tractable way to encode this type of relevance.

2.2 Givenness Hierarchy

The Givenness Hierarchy is a hierarchical mapping of cognitive statuses about the relevance of entities such as topics, concepts, or objects [9]. Fundamentally this allows us to give discreet categories to entities based on their cognitive relevance to the conversation in a way that does not rely exclusively on verbal language used. Specifically, an entity can have the cognitive statuses consisting of:

- (1) In Focus: Entity is at the center of attention
- (2) Activated: Entity is represented in working memory, but is not necessarily the center of attention.
- (3) Familiar: Entity is represented in memory, while not necessarily being represented in working memory.
- (4) Uniquely Identifiable: Entity can be accessed uniquely, without necessarily being represented in memory
- (5) Referential: Entity can be accessed, but not necessarily accessed uniquely
- (6) Type Identifiable: The type of entity can be accessed, but not necessarily an instance of the entity

Importantly, these cognitive statuses are hierarchical in that all entities that are of a particular cognitive status also encapsulate all

of the lower cognitive statuses. For example, an ‘Activated’ object is also ‘Familiar’, ‘Uniquely Identifiable’, ‘Referential’, and ‘Type Identifiable’, as shown in figure 1. This hierarchy has previously been directly leveraged to play a pivotal role in reference *understanding* to allow for narrowing down the possible target entities in very large, open-world environments based on cognitive status [20]. We believe that similarly the Givenness Hierarchy can be applied to reference *generation* by delimiting distractors based directly on cognitive status. In the previous example, a mug that is in the room would have the cognitive status of ‘Familiar’ while most other mugs the robot is aware of would likely fall under either ‘Uniquely Identifiable’, ‘Referential’, or ‘Type Identifiable’, and therefore the robot only needs to differentiate the target mug from only other Familiar mugs, and not *all* other mugs. While there is some recent research that utilizes this cognitive status model for generating either referring form [5, 10], or for planning referential sentences [18], currently there is no research that utilizes the Givenness Hierarchy for REG. We believe that utilizing cognitive statuses provides a simple and effective way to reduce distractors to a level where using IA at a large scale is not only feasible but desirable.

3 ALGORITHM AND WALKTHROUGH

In this section, we present GAIA, a Givenness-Advised Incremental Algorithm, which is a modified version of the Incremental Algorithm (IA) [4] which leverages the Givenness Hierarchy [9] to reduce the number of distractors assessed by IA. This implementation is done by simply ruling out distractors that have a cognitive status lower than the cognitive status of the target referent and then feeding into the standard IA. In this way, we develop a simple and tractable way to significantly reduce the number of distractors needed to be assessed and reduce the amount of properties needed to formulate a unique description of the target referent.

Notation

D	Incrementally built up list of descriptors
P	Queue of all properties in preference order consisting of $\{p_0, \dots, p_i\}$
M	Robot model of all entities in the environment consisting of $\{m_0, \dots, m_n\}$, where each entity contains values for each property (v_{pi})
m_t	Target entity for referring expression
c_m	Cognitive status for entity m
$v_{m,p}$	property p value for entity m
X	Incrementally pruned set of distractors

GAIA starts equivalently to IA, where in line 1 the list of distractors (X) is initialized to all entities (M) except the target entity (m_t). However, lines 2– 9 deviate from IA, in that they use the Givenness Hierarchy cognitive status of the target entity (c_t) and the cognitive status of each distractor (c) to reduce the number of distractors. This is achieved in line 6 where if the cognitive status of the distractor is lower than the cognitive status of the target referent it is removed from the list of distractors. That is, if the target referent has a cognitive status of ‘Activated’, then an entity with a cognitive status of ‘Familiar’ would be ruled out from the distractor list, but an entity with a cognitive status of ‘In Focus’ would not. While this step is simple and straightforward it directly allows applicability of the IA to interactive robotic applications. For example, in a hospital setting a robot may have knowledge representation of every object in the hospital giving all known entities

Algorithm 1 GAIA: Givenness-Advised Incremental Algorithm

```

1:  $X = M \setminus m_t$  // Set distractors equal to all entities except target referent
2:  $c_t = m_t[c]$  // Get cognitive status of target referent
3: // Remove all distractors who's cognitive status is lower than the target referent
4: for  $x$  in  $X$  do
5:    $c_x = x[c]$ 
6:   if  $c_x < c_t$  then
7:      $X = pop(x)$ 
8:   end if
9: end for
10: // Use the incremental algorithm to find the description using the remaining distractors
11:  $D = \text{new Queue}()$  // Initialize the Description
12: while  $X \neq \emptyset$  and  $P \neq \emptyset$  do
13:   // For each property in preference order, find the new set of potential distractors
14:    $p = pop(P)$ 
15:    $v_{m_t,p} = m_t[p]$ 
16:    $X' = \emptyset$ 
17:   for  $x = pop(X)$  do
18:     // Add to the new distractor list any entity who has the same property value as the target referent
19:      $v_{x,p} = x[p]$ 
20:     if  $v_{m_t,p} == v_{x,p}$  then
21:        $X' = push(x)$ 
22:     end if
23:     // Only add the property value to the description if the new distractor list is smaller than the old one
24:     if  $X' \neq X$  then
25:        $D = push(v_{m_t,p})$ 
26:        $X = X'$ 
27:     end if
28:   end for
29: end while
30: return  $D$ 

```

a cognitive status of least ‘Referential’ or ‘Uniquely Identifiable’. However, the robot may have more direct knowledge of all entities in the same room as the robot making them ‘Familiar’. If the robot then needs to refer to a particular cup within the current room (giving it a ‘Familiar’ cognitive status), it can eliminate the ‘Referential’ or ‘Uniquely Identifiable’ entities the robot is aware of such that it only needs to differentiate its target cup from other ‘Familiar’ cups. This stands in stark contrast with other REG methods which would try to differentiate the target cup from every other cup in the hospital and allows for the viability of using the IA as-is, which has been proven extremely simple and effective [19].

The rest of GAIA directly follows the procedure of IA. Once the initial distractors (X) have been set, an empty description (D) of the target referent is created in line 11. Then all properties (P) are iterated through to eliminate distractors in preference order until either there are either no remaining distractors or all properties have been iterated. Then for each property, the corresponding attribute value for the target referent ($v_{m_t,p}$) is found in line 15.

For example the target entity might have the attribute *color* with the value *green*. Then a list of new distractors (X') is initialized as empty. All entities in the previous list of distractors are iterated through, and the new list of distractors is populated with all of the previous distractors whose property values are the same as the target referent. From the previous example, any other current distractor with the *color* of *green*, becomes a new distractor. Finally, in lines 24–27 the property value is only added to the description of the target referent if it actually ruled out any distractors, or in other words if the new distractor list is different than the previous distractor list. Following the example, the *color* of *green* is only added to the referring expression if and only if it reduces the number of distractors. While other REG algorithms may not be implementable at the scale needed for future robotic environments, by using cognitive status to eliminate distractors, GAIA can be leveraged in large open-world environments to provide concise and accurate referring expressions.

4 CONCLUSION

While REG is a well-studied topic in linguistics, the incorporation of REG into robotics brings along new challenges that need to be addressed. In this work, we explore how in robotic settings, the large databases of knowledge that a robot can access render traditional REG algorithms such as IA ineffective. This is because REG algorithms need to compare a target referent to all other possible entities, even if many of those entities are not relevant in the current context. To address this, we propose the use of the Givenness Hierarchy to inform a cognitive status for each entity. With this cognitive status, we can infer how relevant each entity is, not only to a particular conversation but even to a particular reference within a conversation. In this way, we can use cognitive status to rule out any distractor entities that have a lower cognitive status than the target entity. In this paper, we specifically leverage this principle for GAIA, which rules out distractors based on cognitive status to enhance the effectiveness IA. In this way, we can greatly enhance the performance of referring expression generation for more efficient and natural language for robots.

ACKNOWLEDGMENTS

This work has been supported in part by the Office of Naval Research grant N00014-21-1-2418.

REFERENCES

- [1] Mira Ariel. 1988. Referring and accessibility. *Journal of linguistics* 24, 1 (1988), 65–87.
- [2] Mira Ariel. 2001. Accessibility theory: An overview. *Text representation: Linguistic and psycholinguistic aspects* 8 (2001), 29–87.
- [3] Robert Dale. 1989. Cooking up referring expressions. In *27th Annual Meeting of the association for Computational Linguistics*. 68–75.
- [4] Robert Dale and Ehud Reiter. 1995. Computational interpretations of the Gricean maxims in the generation of referring expressions. *Cognitive science* 19, 2 (1995), 233–263.
- [5] Gabriel Del Castillo, Grace Clark, Zhao Han, and Tom Williams. 2023. Exploring the Naturalness of Cognitive Status-Informed Referring Form Selection Models. In *Proceedings of the 16th International Natural Language Generation Conference*. 269–278.
- [6] Andrew Gargett, Konstantina Garoufi, Alexander Koller, and Kristina Striegnitz. 2010. The GIVE-2 Corpus of Giving Instructions in Virtual Environments.. In *LREC*.
- [7] Talmy Givón. 1983. Topic continuity in discourse. *Topic continuity in discourse* (1983), 1–498.
- [8] Herbert P Grice. 1975. Logic and conversation. In *Speech acts*. Brill, 41–58.
- [9] Jeanette K Gundel, Nancy Hedberg, and Ron Zacharski. 1993. Cognitive status and the form of referring expressions in discourse. *Language* (1993), 274–307.
- [10] Zhao Han and Tom Williams. 2023. Evaluating Cognitive Status-Informed Referring Form Selection for Human-Robot Interactions. In *2023 Annual Meeting of the Cognitive Science Society (CogSci)*.
- [11] Emiel Krahmer and Kees Van Deemter. 2012. Computational generation of referring expressions: A survey. *Computational Linguistics* 38, 1 (2012), 173–218.
- [12] Ann Kronrod and Orit Engel. 2001. Accessibility theory and referring expressions in newspaper headlines. *Journal of Pragmatics* 33, 5 (2001), 683–699.
- [13] Thomas Pechmann. 1989. Incremental speech production and referential over-specification. (1989).
- [14] Ehud Reiter. 1990. The computational complexity of avoiding conversational implicatures. In *28th annual meeting of the association for computational linguistics*. 97–104.
- [15] Ehud Reiter and Robert Dale. 1997. Building applied natural language generation systems. *Natural Language Engineering* 3, 1 (1997), 57–87.
- [16] Fahime Same, Guanyi Chen, and Kees van Deemter. 2023. Models of reference production: How do they withstand the test of time? *arXiv preprint arXiv:2307.14817* (2023).
- [17] Fahime Same and Kees van Deemter. 2020. A linguistic perspective on reference: Choosing a feature set for generating referring expressions in context. In *Proceedings of the 28th international conference on computational linguistics*. 4575–4586.
- [18] Kevin Spevak, Zhao Han, Tom Williams, and Neil T Dantam. 2022. Givenness hierarchy informed optimal document planning for situated human-robot interaction. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 6109–6115.
- [19] Kees Van Deemter. 2016. *Computational models of referring: a study in cognitive science*. MIT Press.
- [20] T Williams and M Scheutz. 2019. A givenness hierarchy theoretic approach. *The Oxford handbook of reference* (2019), 457.