**Does Order Word Matter?**

A Literature Review on the Importance of Word Order in Referring Expression Generation Algorithms.

**Floris Geerts**

Supervised by Rick Nouwen

A 7,5 ECTS thesis presented for the Bachelor degree in Artificial Intelligence

Utrecht University

Faculty of Humanities
Utrecht University
Netherlands
July 2021

**Abstract**

A big part within natural language generation is referring expression generation. It is concerned with modelling human behavior when it comes to referring expressions. Several algorithms have been built on such models. Recent findings have shown that there are cross-linguistic differences and that word order is important when it comes to referring expressions. I have found that although existing algorithms handle word order in their own way, they are insufficient to completely mimic human behavior and an additional algorithm might be required.

# Contents

# 1 Introduction

Natural Language Generation (NLG in short) has always been an important part of artificial intelligence. NLG is concerned with constructing computer systems that use a non-linguistic representation to produce text. Simplified: creating humanlike text using models and algorithms. Referring Expression Generation (REG) is a subtask of NLG. As the name suggests, REG is concerned with generating referring expressions, used to direct attention to a target (oftentimes an object). An example of a referring expression task can be seen in figure 1. The objective is to describe the target object, in this case the big circle. For REG, this means that it needs to create an expression that distinguishes the target object from the distractors (other objects in the set). Several algorithms have been created to generate referring such expressions, each with their own strengths and weaknesses.

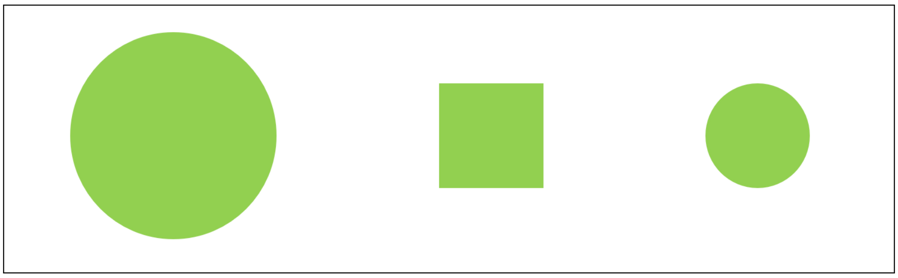

Figure 1: Example set 1, target object is the big circle

A big topic within REG is overspecification. When someone describes an object, he or she might use more words than necessary to identify the object. For our example (figure 1), one could utter "the big green circle"; mentioning the color of the object redundantly. Humans tend to overspecify often and existing algorithms deal with this phenomenon in different ways. Some ignore overspecification entirely, while others try to mimic human behavior on this regard.

Recent research (Rubio-Fernandez, 2019; Rubio-Fernandez et al., 2020; Rubio-Fernandez & Jara-Ettinger, 2020) expanded the REG task. They found that there is a difference between languages when it comes to humans using referring expressions. This has to do with the fact that some languages place their nouns after their adjectives, while in other languages the noun is followed by the adjectives. They found that this difference in word order is responsible for differences in overspecification. Speakers of languages where the noun is followed by adjectives tend to overspecify less often and in different ways. Following these findings, my research question is: to which extent do existing REG algorithms consider word order and cross-linguistic differences? This question can be broken into several parts, as I will do in this study. Firstly it is of importance to know how existing REG systems and algorithms words, which we will do in chapter 2. Secondly, in chapter 3, we will delve deeper into the way humans use referring expressions and the differences between languages. Chapter 4 will discuss how well human behavior is implemented in existing REG algorithms, with a focus on the recently found importance of word order and cross-linguistic differences. This chapter concludes with a possible implementation of these findings.

## 2 Referring Expression Generation

A lot of different algorithms have been invented to tackle Referring Expression Generation (REG). Before we look at a few specific algorithms, it will be useful to look at underlying concepts for most of these algorithms. Grice's philosophy of cooperative interaction is fundamental to much of REG. His so-called maxims were originally designed as a broad model for human interaction, but for REG it is interesting to which extent they should be implemented in algorithms.

### 2.1 Gricean Maxims

An early base for REG models are the so-called Gricean Maxims, invented by Grice (1975). These four maxims were originally designed (together with the cooperative principle) to function as a model for meaningful human interaction. Later, since REG is about just that, these maxims were used as inspiration for REG algorithms. They are rational principles that can enable people to achieve effective communication. Both listeners and speakers can reason to which extent these maxims are and should be followed.

Gricean Maxims (Grice, 1975) in no particular order, but numbered for claritiy's sake:

1. Maxim of Quality
   Do not say what you believe to be false
   Do not say that for which you lack adequate evidence

2. Maxim of Quantity
   Make your contribution as informative as is required
   Do not make you contribution more informative than is required

3. Maxim of Relevance
   Be relevant

4. Maxim of Manner
   Avoid obscurity of expression
   Avoid ambiguity
   Be brief
   Be orderly

These maxims feel quite intuitive and they can be used as a base to build on for referring expressions. As for the maxim of quality, it is relevant to describe an object the way you perceive it. Uttering “square” when referring to a circle is not helpful to the listener. The maxim of quantity states that all information given is useful and information that is not should be excluded. For an example, look back at figure 1. If the target object is the big circle, someone might utter “big green circle”. The word “big” is informative because it distinguishes the big circle from the small circle. However, the word “green” does not provide any information since all objects in the set are green. Following the maxim of quantity, ”green” should not be included in the expression. We can see that these maxims conflict with the way humans use referring expressions. The maxim of quantity states that no more information than required should be given and the maxim of manner advises to be brief. In other words, no overspecification is allowed.

### 2.2 Algorithms

Existing algorithms handle the Gricean Maxims in different ways. Before delving into the algorithms themselves, we must first formalize information in such a way that a computer or algorithm can use it. It will give a better understanding of the way the information is fed to the algorithm.

We represent attributes of objects as an attribute-value pair. For example, for each of the objects in Figure 2 (which is a copy of figure 1, duplicated for readability), we could create a set of pairs. In this case, I use three attributes per object but this can extend to any number of attributes per object.

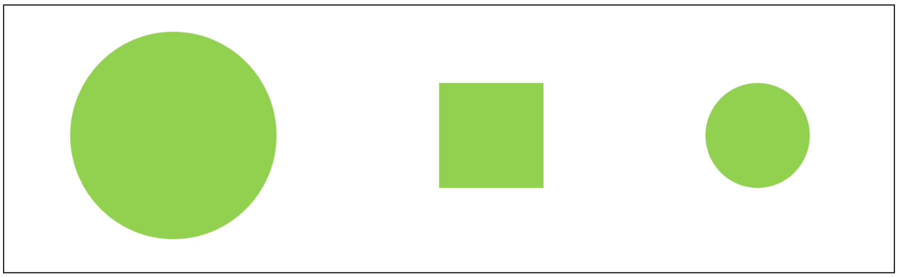

Figure 2: Example set 1, target object is the big circle

{⟨color, green⟩, ⟨shape, circle⟩, ⟨size, big⟩}
{⟨color, green⟩, ⟨shape, square⟩, ⟨size, small⟩}
{⟨color, green⟩, ⟨shape, triangle⟩, ⟨size, small⟩}

The computer must now figure out a way to discriminate the distractors from the target. In other words, creating a distinguishing description of the object being referred to.

It is important to note that a head noun is always included in a referring expression. One would not just say the word "green" when referring to an object, a head noun is needed for grammatical reasons. You could argue that someone might say "green object", which works grammatically. However, this clashes with the maxim of quantity and the maxim of manner. It is more efficient to say "green circle" than "green object".

### 2.2.1 Early Algorithms

Some traditional algorithms simply strive to create the shortest possible referring expression that identifies the target object. The **Full Brevity algorithm** (Dale, 1992) is one of those algorithms. Following both the Gricean maxim of quantity and maxim of manner closely, it tries to generate the shortest distinguishing expression. It first tries to find a distinguishing description using only one property. When that proves impossible, the algorithm tries the same thing, but now using two properties. When this fails, it tries three properties, etcetera. Unfortunately, this way of searching is proven to be a NP-hard task (Dale & Reiter, 1995). This task is computationally very expensive and cannot be completed in polynomial time and thus, when the number of properties of the target object and number of attributes in the shortest possible referring expression become very large, computers will have a hard time completing the search for the shortest referring expression. An algorithm that does not have this NP-hard problem is the **Greedy Heuristic algorithm** (Dale, 1992). This algorithm runs in polynomial time, but does not always produce the referring expression with the fewest attributes. In short, this algorithm tries different attributes (attribute-value pairs) and, if they apply to the target object, adds them to a list. It goes down the list of attributes from the attributes that rule out the most number of the distractors to ones that rule out the least number of distractors. The algorithm works incrementally; words are added to the expression one by one. Every time an attribute is added (and some of the distractors are ruled out), the list updates according to the new discriminatory power of the attributes that are left. The discriminatory power of attributes changes because the set of distractors changes after each attribute that is added to the description. This always creates a correct, distinguishing referring expression, just not always the shortest possible one (which might actually be an upside, as we will discuss later). If we look at example set 1, the Greedy Heuristic algorithms will order the attributes in order of discriminatory power. In this case, "big" rules out two distractors (only one if we take the head noun "circle" into account), while "green" rules out none. The algorithms thus adds "big" to the description. As explained before, a head noun is always included. The expression becomes "big circle". The algorithms checks whether this is a distinguishing description (which it is) and stops, giving "big circle" as output. The **Local Brevity algorithm** (Reiter, 1990a) is a way to check if the generated expression can be shortened. It is often used in combination with another algorithm like the Greedy Heuristic algorithm. After this first algorithm has generated a referring expression, which might well not be the shortest possible expression, local brevity will be able to shorten this expression to an expression that is the shortest expression possible. To obtain such an expression, an algorithm checks whether it is possible to create a shorter correct expression by replacing a set of attributes for a single new component. It does this iteratively, meaning that once it has found an improvement, it runs again on this improvement. This way, the algorithms keeps checking for new components to replace old sets until a shortest referring expression is found. This can be done in polynomial time, but as stated before, local brevity can only be applied to an already generated referring expressions. It functions to check and improve such an expression, not create one of its own.

### 2.2.2 Incremental Algorithm

Humans often provide more information than needed to describe an object. If our goal for REG is to replicate human behavior, we should include overspecification and incremental processing in our models. For this reason, work by Rosh (1978) advocates an extension to the four existing maxims, which is useful for REG. He argues that there are three levels in which a word or object can be classified. A superordinate-, basic- and subordinate class. An example would be:

Superordinate: animal

Basic Level: dog

Subordinate: Golden Retriever

Rosh found that basic level objects are the most inclusive level of classification and should thus be preferred over the other two levels. Even if naming just "animal" would suffice (meaning none of the distractors are animals), people still prefer to use the word "dog" instead. Dale & Reiter (1995a) argue that as a fifth maxim, we could then add:

5. Lexical preference: Use basic-level and other lexically preferred classes whenever possible

Using this fifth maxim, Dale & Reiter (1995a) came up with an algorithm which aimed to mimic human behavior more accurately: the **Incremental algorithm**. This algorithm functions like the Greedy Heuristic algorithm in the way that it also works incrementally. However, unlike the greedy heuristic algorithm, Dale & Reiter (1995) aim to prioritize words people prefer when referring to an object. So instead of looking for the most discriminating properties (like the Greedy Heuristics algorithm), it prioritizes lexically preferred properties. It uses a knowledge base with interface functions. The function `MoreSpecificValue` finds a value that is more specific than the value that it has as input. When you input "dog" into `MoreSpecificValue`, it will (in our example) return "Golden Retriever". `BasicLevelValue` returns the basic level value of an object. When inputting "Golden Retriever", `BasicLevelValue` returns "dog". `Userknows` returns "true" if the listener knows or can easily determine that a certain attribute applies to the object. For example, the listener knows that a golden retriever is a dog and that the target object is a golden retriever. When you input "dog" into `Userknows` (asking whether the target object is a dog), it will return true. The algorithms uses these functions in the knowledge base to generate a referring expression.

The most important function that the incremental algorithm uses is `PreferredAttributes`. This is the function that determines the order of lexical preference of attributes in the domain. The order of the elements in this list are domain dependent and are determined by empirical investigation. Lexical preference can vary from domain to domain. For example, in some domains color will be higher on the lexical preference scale than in other domains. The algorithm iterates through the list of attributes, from most lexically preferred to least. For each attribute, it checks whether the attribute rules out any of the distractors that were not yet ruled out. If this is the case, the attribute-value pair is added to the description. Unlike the Greedy Heuristic algorithm, the list of lexically preferred objects will not be updated in between every iteration since lexical preference does not change when attributes are added to the expression. The algorithm continues iterating through the list of attributes until all distractors are ruled out. Like the Greedy Heuristic algorithm, it will never backtrack; once an attribute is added to the description, it will not be removed later. Even if it becomes redundant because of later added attributes. As an example, we can once again look at example set 1. The way the Incremental algorithm creates a referring expression is dependent on how `PreferredAttributes` is defined. Let us assume (for the sake of this example) that color is preferred over size. The algorithm would first add the color of the target object to the description, in this case "green". It checks whether "green circle" is distinguishing. It is not, so the algorithm continues and adds the word "big" to the description. "Green big circle" is distinguishing for the target, so the algorithm stops. Note that this is not the shortest possible description and that it differs from the description the Greedy Heuristic algorithm came up with.

However, even though this algorithm possibly mimics human behavior better than the previously mentioned algorithms, Viethen & Dale (2006) show that it still differs from human speech significantly. In their study, they compared algorithms to humans in a description task (using 16 colored drawers in a 4x4 grid). They found that the incremental algorithm fails to completely mirror human behavior. The algorithm stops when a distinguishing description has been found, but humans often go beyond that, adding redundant words to their description. An example of this phenomenon would be the human expression that was uttered frequently: "the yellow drawer in the top right corner". The incremental algorithm would generate one of the following: "the yellow drawer in the corner", "the top right yellow drawer" or "the drawer in the top right corner". Even though all of those are correct referring expressions, they do not replicate human behavior precisely. Viethen & Dale (2006) admit that this could also be explained by their modelling of the knowledge base. For example, cornerhood was modelled as a distinct property with discriminatory power. However, humans might have used the word corner as a nominal head to a phrase. It is intuitive to, when having said "the yellow drawer in the top right", add the word "corner" to the end of the phrase. The knowledge base might not have given the algorithm a completely fair chance to mimic human behavior and Viethen & Dale conclude that not only the algorithm is of importance, also how the knowledge base underneath the algorithm is constructed is of significance.

These algorithms all try to mimic human behavior in their own way, but only the Greedy Heuristic algorithm and the Incremental algorithm include overspecification in their model. To extend on this attempt, let us look at what current literature states on human behavior when it comes to referring expressions.

# 3 Human Behaviour

Let us look at an example in which someone is tasked to describe the circle from figure 3. He or she could respond with describing the circle as just “the circle”. There will be no doubt which object the person is referring to, since there are no other circles. However, someone might instead utter: “the green circle”. In this case, with only one circle present, mentioning the fact that the circle is green is redundant for finding the correct object. The person is **overspecifying**; giving more details about the object than required. Although the name might suggest that the extra information is unnecessary and should not be given, research has shown that overspecification does have a purpose. When trying to describe something to someone, it is advantageous if the hearer can identify the object quickly.

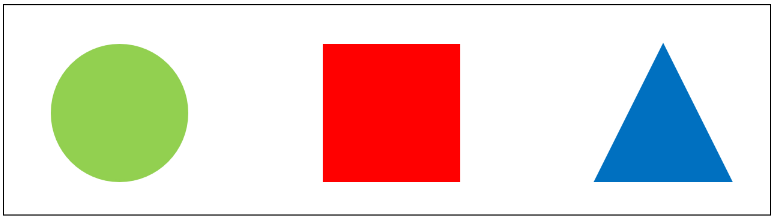

Figure 3: Example set 3, target object is the circle

## 3.1 Overspecification

Overspecification can help with achieving a quick find, especially when the mentioned attribute is fully discriminating (Fukumura, 2018), which is the case in figure 3. ”Green” is rules out all distractors. Even though it is mentioned redundantly, it provides a quicker find of the target. While hearing a referring expression, someone obtains information in a linear way; not all information is obtained at once, but rather it is processed word for word.

### 3.1.1 Incremental processing

Using an eye-tracking experiment, Eberhard et al. (1995) found that eye movements are closely time locked to the words. In our example, when hearing the phrase “green circle”, one would, upon hearing the word green, first look for a green object. People do not process “green circle” as a whole, but rather in order. In other words, information is processed **incrementally**. For incremental processing, word order is important. When, after hearing the word “green”, ”circle” is added to the description, the person starts looking for a circle within the set of green objects. The attention is first directed to the color, and then to the shape of an object. Looking for the color green is easier than looking for the shape of a circle (Pechmann, 1989). Not only color helps with finding an object quicker. There is a plethora of ways to overspecify an object.

### 3.1.2 Different kinds of overspecification

There are several factors in play which influence the way people describe a target. Firstly, people tend to mention the color of an object more often than any other trait (Pechmann, 1989). Degen et al. (2019) give an example with three pins. They present two sets of three pins. In one of them, the distinguishing feature of the target is its size. In the other, the color is different from the color of the two distractor pins. When naming the first target, speakers often include a redundant color adjective. However, in the second example, size will barely by mentioned. This idea is supported by several studies, such as a study by Tarenskeen et al, 2015. This study not only included size and color, but also pattern. Their finding was (among others) that participants overspecified color and pattern substantially more often than size. Furthermore, research by Koolen et al. (2013) found that participants overspecify color more often when the scene variation is high, compared to when the scene variation (variation between distractors) is low. As an example, someone is less likely to mention the color of an object if there are not many different colors present in the set. When a lot of different colors are present, the color of the target object is more likely to be mentioned redundantly.

Another influence on overspecification is the typicality of color (or any other attribute). In other words, how regular is the color of the object for that given object. Examples of color typicality would be a yellow banana, green grass or white snow. Westerbeek et al. (2015) found that the less typical the color is for the target object, the likelier a person is to mention its color. They use a banana as an example. En absence of other bananas, a yellow banana is not likely to be named by color. In contrast, color is very likely to be mentioned when that same banana is blue. A brown banana is an in-between case, since it is a more natural color for a banana to have, but still not as frequent of an occurrence as yellow bananas. This feature of typicality also applies to features other than color. Mitchell & Reiter (2013) extended the typicality feature to also include shape and size. In scenarios where participants could use either shape or material to describe an object, they were significantly more likely to mention the atypical attribute over the typical one.

## 3.2 Cross-linguistic differences

Recent studies (Rubio-Fernandez, 2019; Wu et al., 2020) have found differences between languages when it comes to referring expressions. They found that English speakers use color adjectives more redundantly than Spanish speakers. This can be explained by a difference in grammatical sequencing in these languages. We can split languages in two groups. One where the noun is placed after the adjectives (prenominal, noun-adjective languages), like English, and one where the order is reversed; the noun is followed by the adjectives (postnominal, adjective-noun languages), like Spanish. In Spanish, most adjectives are placed after the noun, but not always. For example, possessive adjectives are placed before nouns (”my bag”). However, since the default position for most attributive adjectives is postnominal, I will consider Spanish as a strictly noun-adjective language.

Incremental processing of information can be used to explain the differences between languages. The word that is first mentioned, will be processed first. The order in which a combination of noun and adjectives is presented is relevant for the way humans process information. As a note for this chapter, difference between these languages stem from a difference in grammar of these languages, not from a native difference between the people use these languages. When Spanish speakers were tested in English, their linguistic behavior was equal to that of native speakers of English (Rubio-Fernandez et al., 2020).

### 3.2.1 Incremental processing in different languages

Efficiency is key for REG and this incremental processing of language often (not always, as will be discussed later) allows a quicker find of a target (Spivey et al., 2001; Fukumura & Carminati, 2021). This is confirmed by a finding by Rubio-Fernandez & Jara-Ettinger (2020), using eye-tracking with participants from different languages. Participants were given a sheet with four pictures and then fed a referring expression. Participants would start looking at objects with a certain attribute as soon as that attribute was mentioned. In line with this finding, a difference was found between languages for a one competitor case: a case were one of the four items has something in common with the target object, albeit kind or property. When the property (adjective) was the distinguishing factor for the target object (in other words, there was a property competitor present), speakers of languages that place adjectives before nouns were quicker to look at it. When the kind (noun) was distinguishing, prenominal (noun-adjective) language speakers were quicker with looking at the target object.

A second finding was that participants would anticipate the adjective to be distinguishing. To explain this, figure 4 resembles an example of two competitors that was used in the experiment (although the actual objects used by Rubio-Fernandez & Jara-Ettinger (2020) were different). The target object is once again the green circle. The red circle is a kind competitor and the green square is a property (in this case color) competitor. For the different languages, the referring expression given was “green circle” or “circle green” (in the appropriate languages). Once again, incremental processing takes place, but interestingly it differs between the two types of languages. Upon hearing the word green, adjective-noun languages would already be looking primarily at the green circle. This has to do with predicting the intention of the speaker. The fact that green is distinguishing two circles from one another, tends humans to anticipate that the speaker is going to mention the circle. In other words, the listener expects that the speaker does not use the color redundantly, but to distinguish the two circles. For noun-adjective languages, where the referring expression is “circle green” this phenomenon does not occur. Upon hearing circle, the listener can not predict which circle will be mentioned. This seems intuitive at first glance, but it is a significant difference between the two types of languages. The logic that is applied for adjective-noun languages can just as well be (but is not) applied to noun-adjective languages. When hearing the word “circle”, one can (as with adjective-noun languages) expect the speaker to not use words redundantly and intend the green circle. If it was the red circle the speaker was referring to, the word circle would be redundant and just the color red would suffice. However, when referring to an object, it is highly unusual to leave out the actual object (head noun) you are referring to. This does mean that only speakers of adjective-noun languages can use this predicting skill to infer the object, by anticipating the intended function of the adjective. It shines a light on complex differences between languages; word order is linked with predicting the speaker’s intention. It also

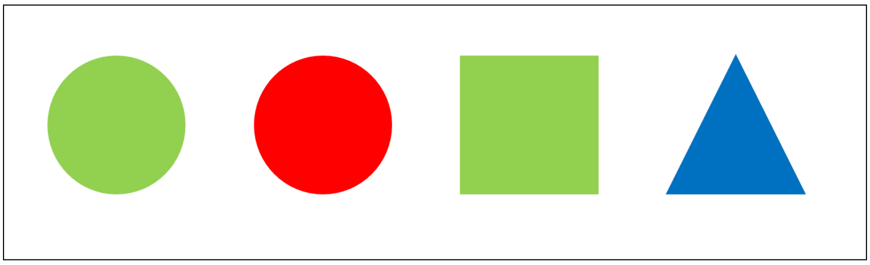

Figure 4: Example set 2, target object is the green circle

reveals when the incremental processing of language can hinder processing time. If (instead of the green circle) the target is the square in figure 4, uttering “green square” (in an adjective-noun language) will lengthen the time it takes to find the target, since the word green will direct the listener to the green circle. The redundantly named attribute creates temporary ambiguity. When the attribute is not distinctive of the target object, a shorter description (just “square”) can lead to shorter processing time (Rubio-Fernandez et al., 2020).

# 4 Word order in REG

The findings of cross-linguistic differences reveal the importance of word order for referring expressions. Before looking at how well the discussed algorithms handle word order, it is important for clarity's sake to make a distinction between two different types of incrementality.

## 4.1 Incremental generation versus incremental processing

Incremental generation is the process of generating (in this case) an expression incrementally. It starts with only a head noun and adds adjectives one by one. This way, a string is generated word for word. When talking about processing information incrementally, once again, it means that the information is being processed word for word, with previous words impacting how following words are interpreted (looking for a circle within the set of green objects when hearing "green circle"). Even though the definition of both types of incrementality is very similar, it is important to realize that the two types do not go hand in hand necessarily. Assume that an algorithm has generated an expression incrementally. This does not mean that this string can also easily be processed incrementally. In other words, the word order generated incrementally is not necessarily the word order that is easiest to process incrementally.

## 4.2 Word order in algorithms

For each of the algorithms discussed earlier, we will look at how well they cope with the finding of the importance of word order. As such, for completeness sake, I will include the Full Brevity algorithm in this analysis.

Overspecification is in no way implemented in the Full Brevity algorithm, let alone word order. This algorithm sees all referring expressions with the same number of attributes as equal and does not differentiate between different word orders.

The Greedy Heuristic algorithm is more interesting in this regard because it works incrementally. Although not directly, the greedy heuristic algorithm does consider word order. As explained earlier, it first looks for the most discriminatory attribute. After that attribute is added to the description, it once again looks for the attribute with the highest discriminatory power, but now only looking at the distractors that are not ruled out by the first attribute. The word order of the generated referring expression will be from most discriminatory attribute to the least (discriminatory attribute that still rules out distractors). For an example, let us look at figure 5. The target object is the big green circle. "Green" rules out two distractors, while "big" only rules out one. The first attribute that is added to the description is "green". Because this is not yet a distinguishing description (we have ruled out the two red objects, but not the small circle), the algorithm adds the now most dicriminatory attribute left; "big" to the description. The referring expression that the Greedy Heuristic algorithm will come up with would thus be: "green big circle". That is, if the algorithm is working with an adjective-noun language. For a noun-adjective language, we could just move the head noun to the front of the expression and leave the rest as it is; "the circle green big" (would be the English translation). However, as shown in the chapter before, the differences between languages are more complex than this and just moving the head noun might well not suffice in creating a humanlike expression for noun-adjective languages.

The Local Brevity algorithm is a step back from the Greedy Heuristic algorithm since it removes all overspecification. On top of that, it does not change the order of the words in the referring expression. When it comes to word order, it does not improve on the Greedy Heuristic algorithm. It only shortens the referring expression to a minimum (distinguishing description).

The Incremental algorithm has similar ideas to the Greedy Heuristic algorithm. Even though the algorithm creates a referring expression incrementally, this will not necessarily be the most

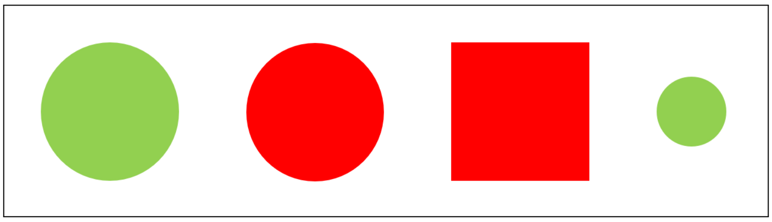

Figure 5: Example set 3, target object is the big green circle

humanlike word order. The quality of the word order depends on the knowledge base on which this algorithm is build. Look at our previous example (Figure 5). The way the Incremental algorithm describes the target depends on the underlying knowledge base. If “big” is coded as being more lexically preferred than “green” (in other words, “big” is higher on the `PreferredAttributes` list than “green), the algorithm will create the expression “the big green circle”. However, if the two are reversed in lexical preference, the expression will be: “the green big circle”. It boils down to how well the `PreferredAttributes` list works for the given domain. In some domains, color might be lexically preferred over size, while in others size is higher on the list of lexical preference. Dale & Reiter (1995a) do not give any details about how this ordered list of elements based on preference is defined, but rather suggest that it will typically be determined by empirical investigation. So, given a domain of objects, one would observe the way humans refer to the target object and process the results to create the `PreferredAttributes` list. The list is not made with word order specifically in mind. It will not consider which words are likely to be mentioned before other words order wise.

Differences between languages are not represented directly in the Incremental algorithm. However, because `PreferredAttributes` is determined by empirical investigation, cross-linguistic differences can be considered. The results of such empirical investigation can differ from language to language. If it is conducted with Spanish speakers, the list `PreferredAttributes` can differ from the list created with an investigation with English speakers. Although Dale & Reiter (1995a) do not mention this difference in languages specifically, they do mention the domain dependency of `PreferredAttributes`. This domain can include the language in which the algorithm performs.

#### 4.2.1 Greedy Heuristic- versus Incremental algorithm

It leaves the question which algorithm handles word order better; the Greedy Heuristic algorithm or the Incremental algorithm? When it comes to cross-linguistic differences, the Incremental algorithms is a clear winner. Having `PreferredAttributes` be dependent on the domain (and thereby also language), avoids having to deal with these differences. It does beg the question how realistic or practical this empirical investigation is if it also needs to be conducted over several languages.

For word order more generally, the central question for the comparison between these algorithms is: which words do people use early in a referring expression; ones that have a high discriminatory power or ones that are lexically preferred? A study by Fukumura (2018) has an argument for both algorithms. He started with two hypotheses, one of discriminatory efficiency and one of availability-based ordering. The first states that attributes with a high discriminatory power will be used early in a referring expression (by humans). This would be in favor of the Greedy Heuristic algorithm. The second hypothesis states that the most available adjectives are produced early in an expression. This (although a bit more loosely) corresponds to the Incremental algorithms. Three experiments found that there is some truth in both hypotheses. For now, both algorithms have their own strengths, but a combination of both concepts might be needed to fully mimic human word order.

### 4.3 Implications for REG

Even though both the Greedy Heuristic algorithm and the Incremental algorithm indirectly consider word order, they leave some things to be desired. These algorithms were not designed with word order specifically in mind, leaving room for improvements. There are many factors at play when it comes to the way humans use word order, such as: word combinations that are common, context (the sentence in which the referring expression is embedded) and grammar. On top of that, cross-linguistic differences have been proven to be quite complex. The Incremental algorithm can handle them if language is included in the domain, but the actual practicality seems to lack. In the sense that the empirical research needed for each domain might prove to be too time-consuming or impractical. It might be too difficult to just have word order embedded in an algorithm, without addressing it specifically.

A possible solution could be an algorithm that places a generated referring expression in the most humanlike order. Such an algorithm would work like the Local Brevity algorithm, upgrading an already existing referring expression. For the sake of readability, let us call this algorithm Word Order algorithm. After another algorithm (for example the Greedy Heuristic algorithm) has generated a referring expression, the Word Order algorithm would have the generated expression as its input. It is tasked with finding the most accurate word order for the given expression and giving that word order as an output. To achieve this, it would not only need to find the best word order for the given domain, but also take into consideration which language the expression is in. An example can be seen in figure 6. The target in the first set is once again the circle that is big and green. The Greedy Heuristic algorithm generates “green big circle”. The Word Order algorithm takes this as input and improves on the word order. For this example, we assume that “big green circle” is preferred over “green big circle”. The Word Order algorithm would thus produce “big green circle”. Creating such an algorithm will be a hard task, given the complexity of word order as well as considerations for different languages; both discussed earlier in this study.

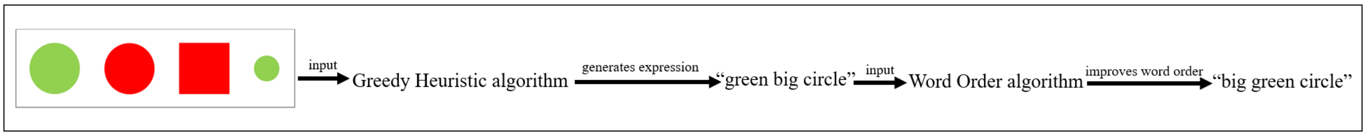


Figure 6: High level concept Word Order algorithm

Even if we succeed in creating the Word Order algorithm, it is still dependent on the first algorithm and might thus not be able to fully reflect different languages. The attributes that will be included in the final referring expression are chosen by the first algorithm. If this algorithm does not consider the difference between verb-adjective or adjective-verb languages, it might not include certain words that would be included in the other type of language. For example, in Spanish, “green circle” (in reversed order and in Spanish) might be a preferred expression. In English, in the same domain, maybe “big green circle” is preferred. If the first algorithm creates either one of these expressions, the proposed Word Order algorithm will have a hard time if it also needs to adjust the expression to the other type of language. It might have to leave out or add attributes to the expression, which, on top of being a complex task, somewhat nullifies the work the previous algorithm has done. To avoid this, it is important that the original algorithm already considers which language the final expression will be in. The Incremental algorithm seems best suited. A combination of the Incremental algorithm and Word Order algorithm might be able to fully mimic human word order and reflect cross-linguistic differences.

# 5 Discussion & Conclusion

In light of recent research, this study has discussed how well existing algorithms consider word order in their referring expression generation. I found that these algorithms do not completely suffice on this subject. Although word order is rooted in the incremental way these algorithms function, they do not take word order into consideration specifically. I found that, since word order is such a complex subject, it might be needed to be addressed separately. Following this, I proposed that a seperate algorithm for humanlike word order might be necessary.

There is plenty of evidence that word order is of importance for REG, strengthened by cross-linguistic studies. However, the specifics seem to lack. 'In what order do humans place their words when referring to an object?' is a central question I could not find an extensive answer to in this study. Although research by Fukumura (2018) certainly is a first step, future research could try to find a deeper understanding of the word order we use in referring expressions. This is no easy task. As I explained before, there might be a lot to the way humans phrase referring expressions. On top of that, it might differ from language to language, even apart from the difference in verb-adjective and adjective-verb languages. Certain common word combinations might exist in one language, but not in the other. A word order that sounds intuitive in one language might make way less sense in another.

Additional research can be done on how to construct an algorithm that places a referring expression in humanlike order (previously called Word Order algorithm). The algorithm needs a certain knowledge base to build on. For this, the previous question (in what order do humans place their words when referring to an object?) needs more research. When research on this topic is extended, a knowledge base can be built. A next step would be to include cross-linguistic differences, possibly making different algorithms for different languages. A Word Order algorithm for Spanish could be created separately from an algorithm made for English. The knowledge base might need to differ from language to language and possibly be completely domain dependent. The Word Order algorithm could be combined with the Incremental algorithm to mimic human referring expressions accurately.

I will conclude with a broader question on REG. Is it even necessary that algorithms mimic human behavior precisely? Maybe an algorithm like the Local Brevity algorithm is already perfect because it creates the shortest possible distinguishing expression. Do we need to consider over-specification, word order and cross-linguistic differences in REG? For me, the answer is yes. REG tries to reflect human behavior. It is not just a tool for creating algorithms, but also a model for the way humans refer to objects. For that reason, it is important to research all parts of human speech on this regard, including word order. It gives us a deeper understanding of natural language.

For algorithms specifically, it is important to understand that the current algorithms already create correct distinguishing expressions. In that way, they are completely sufficient. It boils down to efficiency; how quick can someone find the target object, given a generated description. More humanlike generated referring expressions leads to shorter search times. Updating or creating an algorithm to include the findings of cross-linguistic differences and the importance of word order will be beneficial for visual search, but for now, current algorithms do their job just fine.

## References


Dale, R. (1992). *Generating referring expressions: Building descriptions in a domain of objects and processes.*

Dale, R., & Reiter, E. (1995a). *Computational Interpretations of the Gricean Maxims in the Generation of Referring Expressions.*

Degen, J., Hawkins, R. D., Graf, C., Kreiss, E. & Goodman, N. D. (2020). *When redundancy is useful: A Bayesian approach to "overinformative" referring expressions.*

Eberhard, K. M., Spivey-knowlton, M. J, Sedivy, J. C. & Tanenhaus, M. K. (1995). *Eye movements as a window into real-time spoken language comprehension in natural contexts.*

Fukumura, K. (2018). *Ordering adjectives in referential communication.*

Fukumura, K. & Carminati, M. N. (2021). *Overspecification and incremental referential processing: An eye-tracking study.*

Grice, H.P. (1975). *Logic and conversation.*

Koolen, R., Goudbeek, M., & Krahmer, E. (2013). *The Effect of Scene Variation on the Redundant Use of Color in Definite Reference.*

Mitchell, M. & Reiter, E. (2013). *Typicality and Object Reference.*

Pechmann, T. (1989). *Incremental speech production and referential overspecification.*

Reiter, E. (1990a). *The computational complexity of avoiding conversational implicatures.*

Reiter, E., & Dale, R. (1997). *Building applied natural language generation systems.*

Rosch, E. (1978). *Principles of Categorization.*

Rubio-Fernandez, P. (2019). *Overinformative Speakers Are Cooperative: Revisiting the Gricean Maxim of Quantity.*

Rubio-Fernandez, P., & Jara-Ettinger, J. (2020). *Incrementality and efficiency shape pragmatics across languages.*

Rubio-Fernandez, P., Mollica, F. & Jara-Ettinger, J. (2020). *Speakers and Listeners Exploit Word Order for Communicative Efficiency: A Cross-Linguistic Investigation.*

Spivey, M. J., Tyler, M. J., Eberhard, K. M. & Tanenhaus, M. K. (2001). *Linguistically mediated visual search.*

Tarenskeen, S., Broersma, M., & Geurts, B. (2015). *Overspecification of color, pattern, and size: salience, absoluteness, and consistency.*

Viethen, J. & Dale, R. (2019). *Algorithms for Generating Referring Expressions: Do They Do What People Do?*

Westerbeek, H., Koolen, R. & Maes, A (2015). *Stored object knowledge and the production of referring expressions: the case of color typicality.*

Wu, S. A., & Gibson, E. (2020). *Word Order Predicts Cross-Linguistic Differences in the Production of Redundant Color and Number Modifiers.*