UNIVERSITY OF BAMBERG

Applied Computer Science Degree Program in the
Faculty of Information Systems and Applied Computer Sciences

Bachelor's Thesis

Incremental Concept Learning in Grounding Dialog – Investigating Colour Learning with PRAGR

by

Matthias Müller

Supervisor:
Prof. Dr. Diedrich Wolter

August 2022

Abstract

In this paper, I present three incremental learning approaches using the *Probabilistic Reference And Grounding* mechanism PRAGR. I investigate how prototypes for colours can be adjusted for better understanding and generation of referring expressions in agents and if considering the context, i.e. the other objects in the scene, improves the learning performance. It is discussed how training examples can be dynamically generated while participants interact with the system in an abstract dialog, thereby, labeling the instances. The learning approaches are compared based on the amount of change to the model they need and the prediction power of the models trained with the corresponding algorithm. The hypothesis that a context-sensitive approach exceeds context-insensitive approaches by these criteria could neither be proven nor disproven due to challenges arising with concept learning in PRAGR, especially hierarchies in concepts and robustness.

Contents

1 | Introduction

Mary is resting on the sofa, relaxing after a long day at work. She decides to read the new book she got for her birthday last week, and asks her personal assistant robot Amanda: Could you pass me that red book on my desk? Amanda scans the desk and identifies that two books seem to match that description. She asks: Do you mean the one in front of the coffee cup? Slightly annoyed, Mary replies: No, not the orange one, the red one. Amanda understands and confirms: Oh, okay. I'll get it., moves to the desk, grabs the book, and brings it to Mary. This example is adapted from Mast et al. [MFW16].

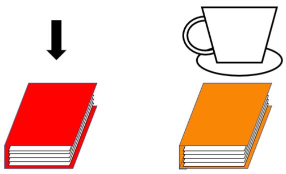

Figure 1.1: Possible abstract representation of the two books in the initial example.

For the successful interaction between Mary and Amanda, they had to refer to objects, in this case, books, by choosing appropriate attributes of these objects. Mary decided to pick the attribute colour, in her case red. Alternatively, she might have used expressions like ”the thick book” or spacial properties like ”the one in front of the coffee cup”, as Amanda did. Such descriptions are called *Referring Expressions* (RE). For Amanda to understand REs and to generate REs herself that Mary can understand, she needs an internal model of how humans, or in this case Mary in particular, use and generate REs. In the example conversation at the beginning, there was a misunderstanding. Amanda initially chose the wrong book. However, she was not confident enough that her choice was the correct one, so she asked Mary for confirmation. What can Amanda learn from this conversation, i.e. how can she adapt her internal model. Intuitively, we could picture a situation as in Figure 1.1 and argue that Amanda has now an example of what red is and an example what orange is.

To assess this, it must be considered how referring expressions are being generated. This will be discussed further in Chapter 2. For the moment it is sufficient to recognize that the chosen attributes depend on the context. The following example shall illustrate this phenomenon. However, keep in mind that the colours are chosen very specifically and their representation may differ on your display or in a printed version.

In the situation depicted in Figure 1.2, Mary may describe the indicated book also as red. In the situation depicted in Figure 1.3, Mary may pick the colour orange to describe her desired book, even though the colour of the indicated book has not changed. The HSL values are marked below. In the second example, red is not as distinctive as in the first example, as the distractor, the book in front of the coffee cup, may also be considered reddish. This has implications for the learning process. Should the indicated book still be an example for a colour category and if yes, for which, red or orange or both?

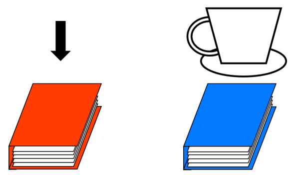

Figure 1.2: Context is relevant for REs. The indicated book could be described as the red book. Chosen HSL values: (10, 1, 0.5) and (150, 1, 0.5)

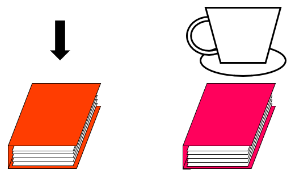

Figure 1.3: The indicated book could be described as the orange book. Chosen HSL values: (10, 1, 0.5) and (240, 1, 0.5)

This is, of course, a constructed example, but it demonstrates that the selection of attributes for a referring expression does depend on the context. In this paper, I want to explore what an agent can infer from the concepts of its internal model from a dialog, especially from misunderstandings. I will use PRAGR [MFW16] as the internal model and limit the domain to colours exclusively. In particular, I propose a context-sensitive incremental learning approach and compare it to other more naive i.e. context-insensitive approaches, if and how the internal model is adapted to an individual. The hypothesis is that the context-sensitive approach converges quicker and in a more stable way than the context-insensitive approaches. In the process, it will also be analyzed how relevant test examples can be generated.

Any implementations can be reviewed at: https://github.com/MuellerMatthiasGER/BachelorsThesis

2 Referring Expression Generation

First, I want to discuss referring expression generation and its properties which are related to the learning process or building its foundation.

2.1 Criteria for a Referring Expression Generating Algorithm

The Gricean Maxims [Gri75] indicate requirements for REs in general. Relevant to this work are:

Quality A referring expression must be an accurate description of the intended referent [Van16].

Quantity A referring expression should contain enough information to enable the hearer to identify the object referred to, but not more information [Van16].

Relevance A referring expression should not mention attributes that have no discriminatory power and, hence, do not help distinguish the intended referent from the members of the contrast set [Van16].

Dale and Reiter further [DR95] proposed a set of criteria that a referring expression generating algorithm should satisfy: First, the communicative goal should be satisfied, i.e. the hearer, or more precisely the addressee, should be able to identify the targeted object. Second, the algorithm should generate referring expressions that do not lead the human hearer or reader to make false conversational implicatures in the sense of Grice [Gri75]. Consider the following statements:

(a) Sit by the table.

(b) Sit by the brown wooden table.

If there is only one table visible, both statements are clear about where to sit. However, one hearing statement b might wonder, why the table was so specifically described and if he or she missed something [DR95]. Note that, if confronted with the statement a, it would be an incorrect conclusion for a learning algorithm to make implications for the concepts brown or wooden, i.e. to treat the observed table as a negative example for these concepts. Just because something was not mentioned is not a reason for it not to be true. Third, the algorithm, if it is to be of practical use, should be computationally efficient. For pure REG this is a clear requirement. Consider the introduction example again. If Amanda takes quite some time to process Mary's request, Mary would not use her personal robot assistant in the first place, but get the book herself.

One could ask if the last performance requirement has to be met by the learning process as well. A short-term adjustment is necessary for good usability. For instance, Amanda should be able to deal with the concepts, Mary just used from that point forward in the current conversation. Mary may add: Amanda, I changed my mind. Could you bring me the orange book instead? Mary would be annoyed if Amanda would not understand this request and had to ask which book she means, as she has just called the other book orange. The short-term adjustment to the current situation might differ from the ideal adjustment to the internal model. The example in Figure 1.3 illustrate this. If this was the setting in the initial example and Mary had called the books orange and pink, Amanda should be able to refer to the indicated book as orange. However, this does not mean that Amanda should always call this colour orange in the future, see Figure 1.2. The calculation of how the internal model should be changed ideally is allowed to take longer than the current conversation, as further conversations using at least part of the involved concepts might be necessary to determine the ideal parameters. The learning approach in this paper is a proposition for a long-term ideal adjustment of the internal model. The mentioned short-term adjustments could be achieved by keeping track of the current conversation in memory.

The learning process has to meet an additional criterion. It should be robust against single outliers. For instance, if one calls a red object green, the agent should not change its concepts of colours drastically, i.e. by introducing a second region for green inside the region for red or, even worse, expanding the green region across yellow and orange to red. Note, that such misunderstandings do not need to happen with malicious intent. Maybe the interacting human did not realize that there was a misunderstanding, as his request was met either way e.g. because the object could be identified by other characteristics or maybe the human does not care about his request anymore. Also, the human might have red-green colour blindness the agent does not know about. Robustness is mentioned and discussed throughout this paper but is not the main focus. Keep this in mind when using the suggested approaches.

2.2 Deviating Concepts in Individuals

When Mary wants to refer to the intended book, she has an internal concept of the colour. When she learned the concept of the colour in this case red, it is probably similar to the people in her surroundings but not identical, as we can hardly convey concepts but communicate with symbols, here with the symbol *red*. For successful communication Mary has not only to consider her own assignment of the colours, so red_{mary} and $orange_{mary}$ but also of Amanda's assignments red_{amanda} and $orange_{amanda}$. If Mary knew that Amanda had problems with identifying colours, she might use different attributes in the first place. Moreover, the individual perception of colour depends on the distribution of the colour cones in the eye, which differs even between humans who are not colour blind [RW99]. The perception of people may also change in a certain domain if they are more exposed to that domain [GS12].

The fact that the assignment of attributes and referring is not universally identical, shows the importance of a learning process. In the end, it is irrelevant what Mary thinks is the 'right' referring expression. She even might call the book a completely different colour. There is no such thing as a correct or incorrect RE, only successful and unsuccessful communication [Mas17].

2.3 The Classic Paradigm of Referring Expression Generation

A *Referring Expression* (RE) is a natural language utterance indicating a particular object within a scene, e.g. "the person on the right" [Yu+18]. If there is only one instance of an object type in the scene, REs can provide additional information. When there are multiple instances of an object type, a RE distinguishes the targeted object from other instances, thereby helping to localize the correct instance [NMD16]. This proposes three challenges for a computer system, comprehending REs, generating REs, and infering other informations from REs, i.e. learning. This paper focuses on the latter one.

Referring Expression Generation (REG) addresses the second challenge, i.e. that computer systems can create REs of their own which is necessary for communication with humans. In the classic view of REG the challenge is to find a subset of the set of valid properties for the targeted object such that the subset is sufficient to uniquely identify the object and distinguish it from the other objects in the set of observed or considered objects. Those other objects are called *distractors* [MFW16]. Van Deemter [Van16] defined REG as follows: Given a finite domain D with objects $d_1, d_2, \ldots, d_n$ with attributes $a_1, a_2, \ldots, a_n$ where each object is defined by a number of attribute-value pairs which are true of this object, find a set of attribute-value pairs whose conjunction is true for the targeted object but not for any distractor [Mas17]. The context is defined as the set of objects currently in the attention of the hearer [DR95].

In this approach the properties are crisp, meaning that a book is either pink, red, or orange. The properties of an object are determined independently from the context, i.e. the other objects in the mental scene. Note, that this contradicts with the setting in the example before in Figure 1.2 and 1.3 in which the same book got assigned different attributes depending on the context, one time red and one time orange.

2.4 The Probabilistic Reference and Grounding Mechanism

To solve the problem with crisp properties Mast et al. [Mas+14][Mas17] propose the **P**robabilistic **R**eference **A**nd **GR**ounding (PRAGR) mechanism. They distinguish between *acceptability*, *discriminatory power* and *appropriateness*. At first, these values are determined for each attribute. Therefore, in the following the term description can be understood as an extension of an attribute, e.g. the colour red.

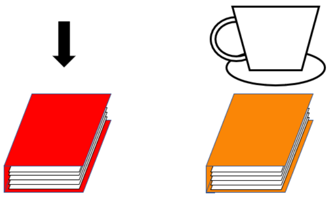

Figure 1.1: Possible abstract representation of the two books in the initial example.

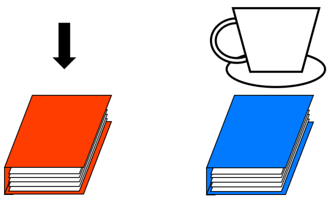

Figure 1.2: The indicated book could be described as the red book.

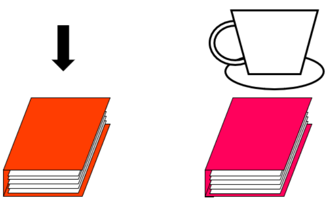

Figure 1.3: The indicated book could be described as the orange book.

The different metrics can be conceived as conditional probabilities. *Acceptability* indicates how good a description D fits a given object x. In terms of conditional probabilities this can be expressed as $P(D|x)$ [MFW16]. For instance, the indicated book in Figure 1.1 would be assigned an acceptability value close to 1, as it is a perfect example of red. It might be a potential prototype for red. In comparison, the indicated book in Figure 1.2 and 1.3 can be considered as red and orange but for neither it is a perfect example. Therefore, the acceptability value for this book would be a medium to high value with respect to the range between 0 and 1.

Discriminatory power expresses how well the description discriminates the intended object from the distractors. In terms of conditional probabilities, given a description of how likely is a particular object the intended one $P(x|D)$ [MFW16]. For instance, in Figure 1.3 the description *red* would not receive a high discriminatory power value because the left book is not a perfect match for red and the colour of the right book is close to being red.

Those two metrics are interrelated by the Theorem of Bayes:

$$P(x|D) = \frac{P(D|x)P(x)}{P(D)} \tag{2.1}$$

$P(x)$ is the probability that the object x was selected. We can integrate further context information like salience or previously referred objects in the current or past dialogs. For simplification, we can assume that each object is equally likely to be referred to. Thus, $P(x) = \frac{1}{N}$ where N is the total number of objects in the scene. $P(D)$ expresses the probability that the description D is used as the referring expression. We can determine it by calculating the acceptability of D for all objects in the scene, i.e. by the law of total probability:

$$P(D) = \frac{\sum_{i=1}^{N} P(D|x_i)}{N} \tag{2.2}$$

Note that in the usual law of total probability we would include $P(x_i)$. Therefore, this formula holds only with the previous assumption that we do not integrate further context information and all objects are equally likely selected. The value of $P(D)$ is high if the description D matches many objects well. An example would be the description *red* in Figure 1.3. This reduces the value of the discriminatory power formula shown in Equation 2.1 as expected.

When the listener gets a description D, the object which fits the description best is a good candidate for the referred object [MFW16]. If we want to integrate context knowledge as before, we would need to append the factor $P(x)$ as in the previous examples.

$$x^* := \arg\max_{\mathrm{x}} P(D|x) \tag{2.3}$$

The speaker aims for effectiveness in communication by choosing a referring expression that does not just distinguish the targeted object from the distractors, i.e. maximizing discriminatory power $P(x|D)$, but is also a good description for the targeted object if it is standalone, i.e. maximizing acceptability $P(D|x)$. Since these two terms do not necessarily optimize for the same outcome, Mast et al. [MFW16] introduce a balancing parameter $\alpha \in [0, 1]$ to obtain a single optimization formula for the best description:

$$D_x^* := \arg\max_{\mathrm{D}} (1-\alpha)P(x|D) + \alpha P(D|x) \tag{2.4}$$

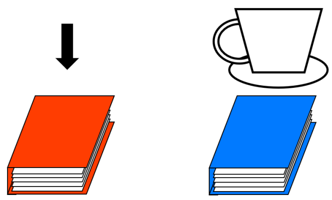

Figure 1.2: The indicated book could be described as the red book.

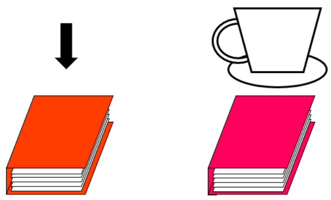

Figure 1.3: The indicated book could be described as the orange book.

If the description is only optimized for discriminatory power (DP), it may use attributes that are only marginally acceptable [MFW16], for instance, calling the indicated book in Figure 1.2 and 1.3 the brown book. Even though brown is not a good description for the book without any context, it is discriminatory because the distractor book in front of the coffee cup is definitely not brown. Nevertheless, a listener will either refute the RE due to its bad match or wonder why it was unnecessarily specified as brown, thus, creating false conversational implicatures in the sense of Grice [Gri75], see Section 2.1. Maximizing purely for DP may also cause overspecification and thus violate the Gricean Maxim of Quantity [Gri75], see Section 2.1. For instance, in the introduction example, Mary could have described the book as the red, thick book on the left with the title …written by the author …which is necessarily specific. Mast and Wolter [MW13] identified that a choice of $\alpha \in [0.1, 0.4]$ leads to intuitive descriptions.

This example description has an additional characteristic. It addresses multiple properties of the object from different domains, so-called complex descriptions [MFW16]. Formally speaking, a complex description is a set of tuples containing a feature and the extension of this attribute, e.g. $D_{book} = \{(colour, red), (thickness, thick), \dots\}$. The tuples are considered conjunctions. In this example, the book is both red *and* thick. If D only contains a single entry, it is called a basic description [MFW16]. The acceptability of a complex description D, $P(D|x)$, can be determined if each basic description d_i in D can be assigned an acceptability value $P(d_i|x)$ and the features of d_i are independent from each other. According to Lawry and Tang [LT09], feature dimensions belonging to different domains can be assumed to be independent. For instance, the thickness of a book can be assumed to be independent of its colour. By the laws of probability, $P(D|x)$ can then be calculated by multiplication of the individual acceptability values for $P(d_i|x)$.

2.5 Scope of this Paper

The PRAGR model provides possibilities for RE comprehension and generation. As described in Section 2.2 concepts, and therefore REs, can slightly differ between individuals. However, PRAGR uses a static model in the background to determine the acceptability value $P(D|x)$ for its calculations. Humans can adjust their language and expressions to the communicating person. This paper proposes solutions how PRAGR can be extended such that the underlying model can be adjusted to individuals. PRAGR could store these models for the next conversation with the corresponding person.

One of the most influencial algorithms for REG is the *Incremental Algorithm* by Dale and Reiter [DR95]. The Algorithm iterates through all potential attributes in a predetermined order is based on assumptions about human preferences [Mas17]. This order might also be adjusted to an individual in analogy to this paper. The *Incremental Algorithm* tries to eliminate distractors by choosing a value for the potential attributes, thereby, having the problems of using crisp properties as discussed in Section 2.3. However, one aspect might be useful to the approach proposed in this paper in the future. If an object has several possibilities for a given attribute, the value is chosen based on high discriminatory power and basic level categories. This might be a useful extension to the proposed approaches in this paper, as hierarchies between categories are not considered for sake of simplicity, see Section 3.3.

The proposed approaches in this paper are presented in three parts. First, Chapter 3 explains how scenes can be created that are suitable for the learning process. In particular, I will discuss how the domain colour can be modelled, how to limit the colour selection to keep the project feasible, and how colours can be chosen such that deviations in the categories are revealed. Chapter 4 introduces three different incremental learning algorithms for an underlying model used by PRAGR and discusses problems associated with them. Two of the algorithms are context-insensitive, the last one is context-sensitive. Chapter 5 investigates the hypothesis of this paper, i.e. that since REs depend on the context, the context-sensitive learning approach is superior to the other learning approaches. They are compared by two metrics, amount of necessary change to the model and correct prediction power. The first metric is motivated by two assumptions. The initial model is already good. Therefore, no great changes are needed. The other assumption is that a good learning algorithm quickly adapts to the user such that after a few inputs less changes are necessary the model. The motivation for the correct prediction power metric is that a good learning algorithm should increase the correct predictions made by the model over time.

3 | Colour as a Feature

In the example in Figure 1.3 we might call the indicated book *red-orange* or *reddish-orangish* to resolve possible misunderstandings in communication. However, it is not trivial to assign an acceptability value to this description, as red and orange are of the same domain and thus not statistically independent. Since this paper is mainly interested in investigating possible learning approaches rather than producing REs of actual use, the possibility by PRAGR of selecting multiple features is not considered here. Only the domain of colour is considered and only single colours are assigned. Nevertheless, colour is a good choice for exploring learning because there are well comprehensible conceptual models of how colour can be represented for instance within the HSL colour space.

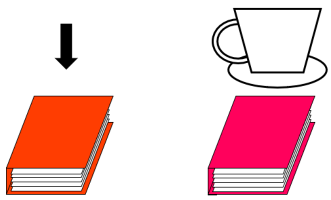

Figure 1.3: The indicated book could be described as the orange book.

This section describes how features can be mathematically modeled, how the colour space from which colours are selected is limited for this paper, and what to consider when generating colour learning samples.

3.1 General Modelling of Features

According to Gärdenfors [Gar04] concepts can be seen as convex regions in an n-dimensional conceptual space composed of one or more dimensions [Mas17]. Some cannot exist independently. For instance, the separation of hue, saturation, and lightness does not make sense when using the HSL colour space. A hue value cannot be assigned a colour not even generic terms like reddish because this only holds when saturation and lightness are in a certain range. Otherwise, the resulting colour will be white, gray, or black. A set of dimensions that can be separated from all other dimensions form a domain [Gär11]. While not being separable from each other, the dimensions hue, saturation, and lightness can be considered separately as a set thereby forming the domain colour.

Gärdenfors further suggest to model categories as prototypes and determining the membership of an instance by Voronoi tesselation [Gar04]. However, human categorization appears to be more complex. For instance, there are expressions like a face turning red, blue, or white. Even though the actual colours of the faces are still quite similar we tend to use a significant amount of the whole colour spectrum. Gärdenfors suggests that the colour selection depends on the relevant contrast set, in this case, the colours of human faces [Gar04]. Thereby, the colour red might differ whether the context is a red face or a red mug. Note that this is a different context sensitivity than in the examples in the beginning. This paper investigates the context-sensitive implications for learning colours but for the whole colour spectrum. The here addressed problem is not further considered in this paper.

Further, Voronoi tesselation divides the conceptual space into equal sections, not in terms of the size of the sections but that all concepts are mutually exclusive on the same hierarchical level. However, concepts often have more complex structures. For instance, can specific colours like crimson be considered as a type of the more general colour red? However, even these hierarchies are not distinct, for example, maroon might be considered red, brown, or both [Mas17].

Mast and Wolter [MW13] propose not to determine the category of an instance by choosing the class of the closest prototype, i.e. by Voronoi tesselation but to use cognitively motivated similarity functions instead. The similarity $s_{i,j}$ is expressed as an exponentially decaying function depending on the distance of an instance to a prototype.

$$s(i,j) := e^{-c \cdot d(i,j)^2} \tag{3.1}$$

This approach has two main advantages. First, categorization is not a binary task anymore, e.g. an instance either is red or it is not, but it can be determined how reddish this instance is. Since the similarity value is always a number between zero and one, it can be used as the acceptability value of an instance for this category. Note that even though the function in 3.1 resembles the standard deviation, it is not a proper probability distribution as it does not integrate into one. However, since we are not interested in true probabilities but rather use the Theorem of Bayes as an approximation for the model, this is not a problem for our purpose. This will become relevant again in Section 4.1 when discussing the *Statistical Learning Approach*.

3.2 Modelling Colour

There are different ways to describe colour models, for instance by their physical properties. The RGB colour model is grounded in the additive mixture of red, green, and blue light in various intensities, e.g. used for displays. The CMYK colour model describes the subtractive properties of colours, e.g. used in printers. Those systems are very useful in their respective domain. However, they do not serve as a good system for the internal model of our REG agent, as they represent different properties than humans intuitively use to categorize colours. The HSL colour space representation aims to solve this problem. It describes colours using hue, which corresponds to basic colour terms like *red, yellow, green, blue, ...*, as well as saturation and lightness, which corresponds to the perceived brightness of the colours to express for instance *dark-red, pale-green, light-blue*. HSL has the further advantage of being directly mappable to RGB which might be helpful for the evaluation as the experiments in this paper are conducted with artificially generated colours viewed on a display. The HSL system is not perceptually uniform, i.e. changes in hue of a fixed amount might be perceived as steps of different size [Mas17]. This will be relevant for the generation of test examples, see Section 3.4.

To categorize colours in the HSL space, PRAGR needs a function for the acceptability of an input colour depending on its hue, saturation, and lightness values. The proposed formula by Mast et al. [MFW16] expresses defined colours as open balls with $C = (h_c, h_r, s_c, s_r, l_c, l_r)$ where the index c represents the center of the ball, i.e. of the exponentially decaying function described in 3.1 of the corresponding dimension and the index r represents the radius of the ball, i.e. the inflection point of the exponentially decaying function. This can be interpreted such that (h_c, s_c, l_c) determine the position of the prototype and (h_r, s_r, l_r) determines how deviations of the prototype are weighted. Thereby, the following function can be used as a value for acceptability:

$$g_C(h, s, l) = e^{-\frac{1}{2}((\frac{h-h_c}{h_r})^2+(\frac{s-s_c}{s_r})^2+(\frac{l-l_c}{l_r})^2)} \tag{3.2}$$

For achromatic prototypes like *white, gray, black* is the hue value irrelevant. Therefore, it can be omitted when calculating the acceptability of such a prototype for a colour.

Using an exponentially decaying function creates an additional problem when using PRAGR. One property of this function type is that they converge to zero but never reach it. This can be problematic for the expressiveness of the discriminatory power. If a category is merely acceptable for the target object, e.g. acceptability value of 0.1, but the category is not acceptable for the distractor object, e.g. acceptability value of 0.01, the category would achieve a discriminatory power of $\frac{0.1}{0.1+0.01} \approx 0.91$. Keep in mind that the discriminatory power is weighted by $1 - \alpha$, thereby accounting for sixty to ninety percent of the overall score for the merely acceptable category, see Section 2.4. This paper is mainly concerned with the situation in which the target and distractor objects are similar, see Section 3.4. Such situations are especially affected by this problem as the category of the target object when viewed isolated without the distractor objects, has a low discriminatory power on purpose. To mitigate this problem, I introduced a lower boundary, called `lower_bound` in the code, of 0.02. An acceptability value cannot undercut this boundary. Thereby, I introduce a baseline for the discriminatory power. If there are two objects, one target and one distractor object, a category that would undercut the lower boundary for both objects achieves a discriminatory power of $\frac{0.2}{0.2+0.2} = 0.50$. Note that it is irrelevant that fifty percent seems to be a high value. Since no category can achieve a lower value for the discriminatory power and the scores are only compared with each other, it only transforms the range of possible values. However, the fixed lower bound induces problems in the learning process. This is further explained in Section 4.4.2.

3.3 Limiting the Colour Selection

Mast et al. [MFW16] parameterize their prototypes by transforming colour intervals from previous experiments ([FMG15], [Fal+13]) to the used open ball notation, see Section 3.2. To account for hierarchical relationships between colours, e.g. accepting *light-red* as a specific form of *red*, the basic colours *red, orange, yellow, green, ...*, they call them *rainbow colours*, span a wide range of lightness and saturation, thereby overlapping the corresponding *light, dark, pale* colours of the same base colour. For fixed values, this approach is a good solution to call most red objects *red*, but if there are two red objects, *red* becomes less distinctive. Thereby, the model can use the terms *light-red* and *dark-red*. Despite that, the solution has some drawbacks.

First, *light-red* and *red* still compete. I suggest that *red* is not a good description if a distractor would receive a description of a subconcept of red like *light-red*. We would probably call the intended object for example *dark-red* for clear communication. The values of their model may be chosen in a way that this case hardly happens, as *red* would receive a low discriminatory value. However, during the learning process, the values of the properties will change and, thereby, resolve this balance. Second, if all objects in a scene have the same transformation with respect to their base colour, for instance, all objects have light colours, the property light can be neglected conforming to the Gricean Maxim of Quality, see Section 2.1. This is not a severe problem for static values as used by Mast et al. [MFW16] because PRAGR could refer to the correct object even if *light* is neglected and because it would be fine to call the object *light* risking a small overspecification. This becomes a problem when we want to learn to adapt the prototype values based on the input, i.e. the dialog. Without further changes, a base colour would compete with its hierarchical subcolours, e.g. *red* might get enlarged, and *light-red* might get downsized as the consequence of such a situation. Thereby, the aforementioned balance between those two prototypes might get damaged. Third, if the agent adjusts the range of the spectrum that a particular individual considers as *red*, the agent should not have to learn this for *light-*, *pale-* and *dark-red* again but infer this information. How these problems can be resolved is not covered in this paper. One possible approach could be to model *light, dark, pale* as transformations on base colours in the saturation and lightness dimension.

In this paper, these problems are mitigated by using a reduced set of colours. The colour set contains the base colours *red, orange, yellow, green, turquoise, blue, purple, pink*, i.e.the rainbow colours $vQCD_2$ used by Mast et al. [MFW16]. For simplification, saturation is set to 100% in all prototypes and all generated colours. This has the further advantage that the adaption of the prototypes during the learning process can be visualized in a comprehensible two-dimensional graph. The colour set is extended with colours that have trivial names and which only differ on the lightness dimension. Mast et al. [MFW16] defined equivalent colours, for instance *dark-orange* ≡ *brown* and *dark-yellow* ≡ *olive*. Since there already exist two good examples for colours with trivial names with lightness values below 50, I chose additional dark colours with trivial names for other hue values, in particular *leave-green (blattgrün), navy-blue (marinblau), indigo, bordeaux*. The corresponding lightness parameters l_c and l_r were adopted from the dark colours $vQCD_4$ used by Mast et al. [MFW16]. Since these are only the initial values for the learning process, precise values for the hue values were not necessary and were chosen to roughly match the corresponding hue range. The colour set could have been extended also with light colour prototypes. However, to be consistent it would be meaningful to find colours with trivial names for the whole hue spectrum. Finding good examples that most people have a good notion of turned out to be difficult. Achromatic colours were also not considered. *Gray* cannot be created when saturation is set to 100%. The initial values of the prototypes can be found in Appendix A.1.

Figure 3.1 visualizes the initial model. Since the acceptability functions are exponentially decaying, the ellipses do not indicate fixed borders. Rather, they mark the inflection point of the acceptability function which is around $g_C(h_c \pm h_r, s_c, l_c) = e^{-\frac{1}{2}} \approx 0.607$.

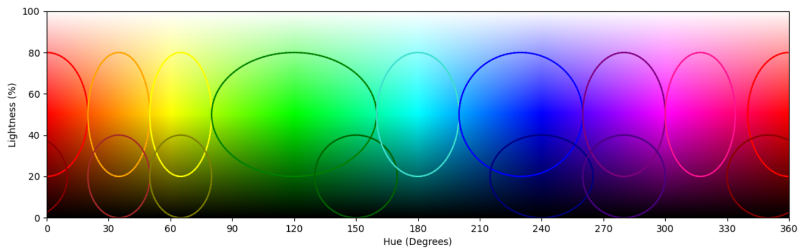

Figure 3.1: Visualization of the initial model. The ellipses mark the inflection point of the acceptability function which is around 0.607.

3.4 Colour Generation

The core of this paper is to investigate how the context influences referring expression generation and, as a consequence, how we can adapt the internal model considering the context. Since for simplicity, the domain is limited to colours in this paper, the interesting situations are when the distractor objects are similar to the referred object. For instance, whereas in Figure 1.3 the similar coloured distractor book influenced the referring description tending towards *orange*, the description for the indicated book in Figure 1.2 probably only depends on the acceptability of the description for the referred object. Therefore, we need a system that generates colours with this property. For simplicity, the system shall only generate two similar colours, in the code called *adjacent colours*. One will later be used as the target, the other one as the distractor object.

The straightforward approach would be to choose a random hue value for the first colour and experimentally determine an adequate step size that is added to the hue value to obtain the second colour. This approach has a major drawback. Keep in mind that the HSL colour system is not perceptually uniform, see Section 3.2, meaning that the perceived difference differs for steps of equal size in the hue dimension depending on the hue region. For example, the initial values for the prototypes of *red* and *orange* differ by 35 units while the *green* is the predominant prototype in a span of 80 units, see Figure 3.1. A second consequence is that if the initial hue value is chosen by a uniform distribution over the hue dimension, this value would be classified as *green* in about 22% of all instances ($\frac{80}{360} \approx 0.222$) by the initial model. This approximation only considers the eight base colours, see Section 3.3.

To mitigate these problems, the used colour generation system does not choose the initial hue value by a uniform distribution of the whole hue spectrum and the step sizes are not universal for the whole hue spectrum. To address the first issue, the system chooses a random prototype at first and then picks a hue value by a uniform distribution over its region for the first colour. The region is bounded by the deviation of the radius h_r around the center h_c of the chosen prototype. Keep in mind that these are not fixed borders but rather describe the inflection point to the exponentially decaying function, see Section 3.2. Consulting Equation 3.2, colours at this border are assigned an acceptability value of $g_C(h_c \pm h_r, s_c, l_c) = e^{-\frac{1}{2}} \approx 0.607$. When only considering the initial values for the eight base colours, the probability of a hue value being chosen for the first colour is fairly distributed over the hue spectrum, as the initial prototypes line up nicely back to back in the beginning, see Figure 3.1. Note that this might change due to the learning process. Furthermore, the system picks any of the prototypes not only the base colours to be consistent with treating every prototype equally and not forming hierarchies, see Section 3.2. Since the prototypes of the dark colours do not line up nicely back to back as the base colours do, this might induce a selection bias. However, this is considered to be less severe as the problems with the naive approach described above.

Next, the problem of dynamic step sizes is addressed. Under the assumption that all prototypes cover their corresponding region and are equal, i.e. there are no hierarchical structures, the hue radius h_r of the previously chosen prototype can be assumed to be an adequate reference point. In this paper, the step size, executed in a random direction, is equivalent to the hue radius h_r multiplied with a factor, here 1.2. The step size is randomly added or subtracted from the hue value of the first colour resulting in the hue value for the second colour. During the dialog, the user can express that he or she cannot distinguish the two colours. The program adapts to this information by increasing the step size of 5 units per indistinguishable colour pair when the same prototype is chosen again, i.e. the prototype from which the first hue value was picked.

This procedure for the step size introduces a new problem that can be explained by considering the adjacent prototypes for *yellow* and *green*. Let the initial hue value be chosen to be inside *green* and close to its radius to the left. Without this countermeasure the hue value for the second colour could be the radius of *green* times the factor 1.2 to the left, i.e. $h_2 = 80 - 40 \cdot 1.2 = 32$. See Appendix A.1 for initial values. The second colour would be considered *orange*, thereby skipping the whole *yellow* region. This can be mitigated by keeping track of when the step size leaves the region of the current prototype and adjusting it to the next prototype from this point on forward. This might be possible if only the base colours are considered, which line up nicely back to back. If the colour selection is extended, this turned out to be too complicated for the colour generation. Therefore, the drawback was accepted.

The saturation value is always set to 100%, see Section 3.3. The lightness values are chosen randomly in the range between 30% and 60% respectively. Lightness is limited to this range, as this is the lightness range the chosen prototypes compete with the most.

4 Incremental Concept Learning

This section introduces the proposed learning approaches. The first two, the *Statistical Learning Approach* and the *Naive Learning Approach* are context-insenstive in contrast to the third one, the *Context-Sensitive Learning Approach*. Afterwards, problems of these algorithms are discussed.

4.1 Statistical Learning Approach

The first incremental learning approach is motivated by the structure of the function used for calculating the acceptability of a prototype for an instance, presented in Equation 3.2. This formula is similar to a multivariate normal distribution of independent random variables, i.e. with covariances being 0. This can be justified in the HSL colour space as hue, saturation and lightness can be viewed as indepenent. The correction term in front is missing to normalize the integral to 1 such that it can be a probability distribution. This can be expressed by

$$X \sim N\left(\begin{pmatrix} h_c \\ s_c \\ l_c \end{pmatrix}, \begin{pmatrix} h_r^2 & 0 & 0 \\ 0 & s_r^2 & 0 \\ 0 & 0 & l_r^2 \end{pmatrix}\right) \tag{4.1}$$

where the random variable X represents the distribution of instances of a specific category by a normal distribution around the corresponding prototype. Therefore, the missing correction term for a dimension only depends on the corresponding radius value h_r, s_r or l_r. Under the assumption that this approximation resembles the real distribution of observed examples, we can learn the values of a category by a maximum likelihood estimation overall observed examples labeled in a dialog as this category. Consulting a standard Statistics textbook, the maximum likelihood estimators for μ and σ^2 in a normal distribution are:

$$\widehat{\mu}_n = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{4.2}$$

$$\widehat{\sigma}_n^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \widehat{\mu}_n)^2 \tag{4.3}$$

These formulas have the problem that they contain x_i, i.e. all observed instances up to this point in time. We could limit the number of stored examples. A randomly chosen subset of the instances should still resemble the same normal distribution. However, storing examples is not the desired goal of learning, but rather finding abstractions and rules, in this case handling cumulated data like the mean value or the variance. For this paper in particular, there is an additional problem. Since this paper only covers adjusting existing categories and not learning new ones, these x_i values do not exist. Therefore, we need a way to calculate the new mean, denoted as $\widehat{\mu}_{n+1}$, and the new variance, denoted as $\widehat{\sigma}_{n+1}^2$, when a new instance, denoted as x_{n+1}, is observed, which does not depend on knowing previous instances $x_i, i \in \{1, \ldots, n\}$ but only their cumulated information, i.e. μ_n and σ_n^2. This can be calculated by[1]

1: The proofs for these interrelations can be found in Appendix B.

$$\begin{aligned}\widehat{\mu}_{n+1} &= \frac{1}{n+1}\sum_{i=1}^{n+1} x_i \\ &= \frac{1}{n+1}(n \cdot \mu_n + x_{n+1}) \end{aligned} \tag{4.4}$$

$$\begin{aligned}\widehat{\sigma}_{n+1}^2 &= \frac{1}{n+1}\sum_{i=1}^{n+1}(x_i - \widehat{\mu}_{n+1})^2 \\ &= \frac{1}{n+1}(n \cdot (\sigma_n^2 + \Delta_\mu^2) + (x_{n+1} - \widehat{\mu}_{n+1})^2) \end{aligned} \tag{4.5}$$

where Δ_μ is the amount how $\widehat{\mu}_{n+1}$ changed with respect to μ_n i.e. $\Delta_\mu = \widehat{\mu}_{n+1} - \mu_n$. Note that the larger n becomes, the less influence a new observation can exert. This can be justified by the fact that the agent can have more and more confidence in its prototype. To allow flexibility of the model we can set an upper limit. For this paper, this has the further advantage that we do not have to keep track of n, as we can assume that the initial model has the highest permitted confidence. The upper limit, called `accumulated_number` in the code, is set to 15 for this paper. Keep in mind that when using the Equations 4.4 and 4.5 this does not mean that only 15 instances are considered. Since we always calculate with the cumulated values μ_n and σ_n^2, all observed instances are considered, however, not with equal weight. Instead, the more recent an observation, the higher its weight. This is achieved by setting the fraction in the beginning to $\frac{1}{\texttt{accumulated\_number}}$ and respectively $n = \texttt{accumulated\_number} - 1$. Thereby, the current observation accounts for $\frac{1}{15}$ of the resulting estimated values for $\widehat{\mu}_{n+1}$ and $\widehat{\sigma}_{n+1}^2$.

Since hue, saturation, and lightess are independent, the formulas can be applied to each dimension individually. Keep in mind that the ranges of the three dimensions differ. While saturation and lightness take values between 0% and 100% and do not need additional care, hue is modelled as an angle, which starts over after 360° with 0°.

The *Statistical Learning Approach* uses every observation as a training example regardless of if the communicating parties had a misunderstanding and regardless of the context, i.e. the distractor objects are ignored. The pseudocode of the statistical algorithm is shown in 1.

Algorithm 1 Statistical Learning Approach

function StatisticalLearning(M: model, T: target, C_{user}: category)
    $n \leftarrow$ accumulated_number $- 1$
    $h_c \leftarrow h_c(C_{user})$
    $h_r \leftarrow h_r(C_{user})$
    $\widehat{h_c} \leftarrow \frac{1}{n+1}(n \cdot h_c + \text{hue}(T))$ ▷ Note that h_c and $\widehat{h_c}$ are angles
    $\Delta_h \leftarrow \widehat{h_c} - h_c$
    $\widehat{h_r} \leftarrow \sqrt{\frac{1}{n+1}(n \cdot (h_r^2 + \Delta_h^2) + (\text{hue}(T) - \widehat{h_c})^2)}$
    $s_c \leftarrow s_c(C_{user})$
    $s_r \leftarrow s_r(C_{user})$
    $\widehat{s_c} \leftarrow \frac{1}{n+1}(n \cdot s_c + \text{saturation}(T))$
    $\Delta_s \leftarrow \widehat{s_c} - s_c$
    $\widehat{s_r} \leftarrow \sqrt{\frac{1}{n+1}(n \cdot (s_r^2 + \Delta_s^2) + (\text{saturation}(T) - \widehat{s_c})^2)}$
    $l_c \leftarrow l_c(C_{user})$
    $l_r \leftarrow l_r(C_{user})$
    $\widehat{l_c} \leftarrow \frac{1}{n+1}(n \cdot l_c + \text{lightness}(T))$
    $\Delta_l \leftarrow \widehat{l_c} - l_c$
    $\widehat{l_r} \leftarrow \sqrt{\frac{1}{n+1}(n \cdot (l_r^2 + \Delta_l^2) + (\text{lightness}(T) - \widehat{l_c})^2)}$
    if BorderChange($M, \widehat{h_c}, \widehat{h_r}, \widehat{s_c}, \widehat{s_r}, \widehat{l_c}, \widehat{l_r}$)< changeLimit **then**
        AdaptModel($M, \widehat{h_c}, \widehat{h_r}, \widehat{s_c}, \widehat{s_r}, \widehat{l_c}, \widehat{l_r}$)
    else
        Log Failure
    end if
end function

4.2 Naive Learning Appraoch

In contrast, the remaining two approaches, namely the *Naive Learning Approach* and the *Context-Sensitive Learning Approach*, only act in case of misunderstandings. The *Naive Learning Approach* also ignores the distractor objects, thus the name *Naive*. Here, the acceptability values for each category are determined for the target object T. The category with the highest value is picked as the referring expression. If this category C_{agent} differs from the description C_{user} the user picked, the values $(h_c, h_r, s_c, s_r, l_c, l_r)$ of the two involved categories C_{agent} and C_{user} are adjusted such that the C_{user} achieves a higher acceptability value than C_{agent}.

Algorithm 2 Naive Learning Approach

function NaiveLearning(M: model, T: target, C_{user}: category)
  $C_{agent} \leftarrow \arg\max_{C} g_C(T)$ ▷ Category with highest acceptability
  if $C_{agent} \neq C_{user}$ **then**
    $\vec{x} \leftarrow \vec{0}$ ▷ Change of C_{agent}
    $\vec{y} \leftarrow \vec{0}$ ▷ Change of C_{user}
    for $i \leftarrow 1$ to max-iterations **do**

      Choose $\vec{x}$ and $\vec{y}$
      $\widetilde{C}_{agent} \leftarrow C_{agent} + \vec{x}$
      $\widetilde{C}_{user} \leftarrow C_{user} + \vec{y}$
      Minimize BorderChange($M, \vec{x}, \vec{y}$) with
      Constraint $g_{C_{user}}(T) > g_{C_{agent}}(T)$ + aimedDifference

      if minimum reached **then**
        if BorderChange($M, \vec{x}, \vec{y}$) < changeLimit **then**
          AdaptModel($M, \vec{x}, \vec{y}$)
        else
          Log Failure
        end if
        return
      end if
    end for
    Log Failure
  end if
end function

For the difference to be significant, such that the agent would pick hopefully the same category as the user when confronted with the same inputs after the learning, the optimizer shall aim for a difference for at least 0.01, called `aimed_difference` in the code:

$$g_{C_{user}}(T) > g_{C_{agent}}(T) + \texttt{aimed\_difference} \quad (4.6)$$

There are infinite ways of adjusting those twelve parameters such that the inequality holds. Under the assumption that the current model is close to the ideal one, I suggest that the adjustment with the least changes to the current model is the best choice. I do not consider the absolute changes of the parameters $(h_c, h_r, s_c, s_r, l_c, l_r)$ of both categories to be a good metric. For instance, if you want to adjust the right border[2] of the hue dimension of a category by 10 units to the right, h_c and h_r must be increased by 10 units in total. How the 10 units are distributed to h_c and h_r is irrelevant to the outcome of the right border. However, if only one of the parameters is increased by 10 units, the left border is also moved by 10 units, even though this change is irrelevant to the current desired adjustment. If both h_c and h_r are increased by 5 units each, the left border does not move. Therefore, I suggest that the absolute change of the borders of the involved categories is a better metric for the minimization problem. I

2: Note that border is in the sense of Section 3.4, i.e. it is rather the inflection point of the acceptability function.

call the left hue border h_0 and the right hue border h_1 as done by Mast et al. [MFW16], respectively for saturation with s_0, s_1 and lightness with l_0, l_1. To further aim for minimal change, I use the sum of the squared changes in each border instead of the absolute change.

An overview for the *Naive Learning Approach* is shown in Algorithm 2. For the minimization task, i.e. line 8 - 12 in 2, the `minimize` function of the `scipy.optimize` python package is used. This approach has flaws that also apply to the *Context-Sensitive Learning Approach*. Therefore, the problems are discussed in Section 4.4.

4.3 Context-Sensitive Learning Approach

Algorithm 3 Context-Sensitive Learning Approach

function ContextSensitiveLearning(M: model, T: target, C_{user}: category)
    $C_{agent} \leftarrow \arg\max_{C} score_C(T)$ ▷ Category selected by PRAGR
    if $C_{agent} \neq C_{user}$ **then**
        $\vec{x} \leftarrow \vec{0}$ ▷ Change of C_{agent}
        $\vec{y} \leftarrow \vec{0}$ ▷ Change of C_{user}
        for $i \leftarrow 1$ to max-iterations **do**

            Choose $\vec{x}$ and $\vec{y}$
            $\widetilde{C}_{agent} \leftarrow C_{agent} + \vec{x}$
            $\widetilde{C}_{user} \leftarrow C_{user} + \vec{y}$
            Minimize BorderChange($M, \vec{x}, \vec{y}$) with
            Constraint $score_{C_{user}}(T) > score_{C_{agent}}(T) + \text{aimedDiff}$

            if minimum reached **then**
                if BorderChange($M, \vec{x}, \vec{y}$) < changeLimit **then**
                    AdaptModel($M, \vec{x}, \vec{y}$)
                else
                    Log Failure
                end if
                return
            end if
        end for
        Log Failure
    end if
end function

The *Context-Sensitive Learning Approach* is similar to the *Naive Learning Approach*. Its main difference is that it uses the score that PRAGR assigns each category instead of the acceptability values. Since the score considers the distractor objects by using the discriminatory power value, this approach is called *context-sensitive*. Whereas the *naive* approach is activated when the category with the highest acceptability value assigned by the agent differs from the chosen category by the user, the *context-sensitive* approach uses the final score of each category for the trigger instead of the acceptability value. The constraining inequality that should be met by the optimizer is:

$$score_{C_{user}}(T) > score_{C_{agent}}(T) + \texttt{aimed\_difference} \tag{4.7}$$

The pseudocode shown in 3 is similar to the *Naive Learning Approach*. Changes are highlighted.

4.4 Problems of the Learning Approaches

In this section, I will discuss some unaddressed problems of the proposed learning approaches.

4.4.1 Selection of Adapted Categories

This problem is relevant for the *Naive* and the *Context-Sensitive Learning Approach*. I will explain the problem by using the final PRAGR score, i.e. the metric of the context-sensitive approach. The problem can be transferred to the naive approach by replacing "score" with "acceptability value" in the following.

When a user enters a category for the target object and the agent assigned a higher score to another category, the learning process is initiated. The optimizer tries to adapt the parameters $(h_c, h_r, s_c, s_r, l_c, l_r)$ of those two parameters, see Section 4.3. Even if the adaption is successful, it does not mean that the agent would choose the same category as the user just did, when confronted with the same situation again after the learning process. The reason for this is that the learning process only considers the category that got assigned the highest score. However, there might be other categories that got assigned a higher score than the category the user picked. For instance, if the agent assigns *red* to the target object but the user chooses *yellow*, chances are that *orange* got assigned a higher score than *yellow* as well. For the learning process only *red* and *yellow* are considered. There are two main reasons to exclude *orange* in the learning process. First, the changes to the model might be much bigger. Besides the fact that *orange* is an additional category that must be adapted, the necessary change for *yellow* might be bigger because now it also has to exceed the score of *orange*. In this example, one might argue that the difference between *red* and *yellow* is so big that this example should be treated as an outlier and the model should not be changed much. Robustness is further discussed in Section 4.4.3. Nevertheless, the problem is more relevant when the model shall also consider more specific colours for instance *salmon* because here the categories are located more closely together.

The implementation can deal with multiple constraints at once. Therefore, one would need to create multiple goals as they are called in the code.

This might also be useful for another situation. In the initial example when Amanda chose the wrong book at first, Mary did not just say that she wants the other book but she also identified the other book as *orange*. This additional information could also be used by the learning algorithm. For sake of simplicity, this is not further considered in this paper.

4.4.2 Consequences of the Lower Bound in PRAGR

Introducing a lower bound for the acceptability values in PRAGR improves the quality of the generated referring expressions, see Section 3.2. However, it also induces a quite severe problem for the naive and the context-sensitive learning approach. They use an optimizer to determine the necessary adaption. Since the lower bound sets the acceptability function to a fixed value in regions further away from the prototype, the deviation of the acceptability function in these regions is constant at zero. Thereby, the optimizer cannot determine in which direction to search for the solution. The optimizer often reaches its iteration limit. In those cases no learning takes place. This can be interpreted that the user input would require too much change in the current model and is therefore neglected. However, this is not a good solution for achieving robustness. The problem may be mitigated by choosing a dynamic lower bound depending on the distance to the prototype rather than choosing a fixed value. Thereby, a similar effect might be achieved when using a leaky ReLU activation function compared to a regular ReLU function. Introducing an acceptability function with a leaky lower bound is not covered in this paper.

4.4.3 Robustness

One problem with robustness concerning the statistical approach is the assumption the distribution of the observed examples can be approximated with the underlying normal distribution. This assumption can be questioned. For instance, you might be in an environment in which a certain shade of a colour occurs disproportionally often. The agent should not infer that this is the only shade of this colour. Figure 4.1 shows the learned model after it has observed twenty times the colour value of the right border of the initial green category. The dotted lines indicate the model before the learning to visualize the adjustment. Thereby, the agent unlearned shades of green that tend towards yellow, even though the inputs were correctly labeled as green.

The *Naive Learning Approach* also suffers from robustness. Figure 4.2 shows the learned model when the naive learning approach observed the colour of the initial prototype of orange which was labeled yellow. The yellow category now spans the entire hue spectrum. The acceptability values of the target object changed from 1 to 0.94 for orange and from 0.14 to 0.96 for yellow, thus, meeting the goal. There might also be a solution that takes less change in the model, e.g. by moving orange and yellow to the left. Maybe the optimizer got stuck in a local minimum. Since the goal could be met, the found solution was applied. However, a robust implementation should not accept such a dramatic change.

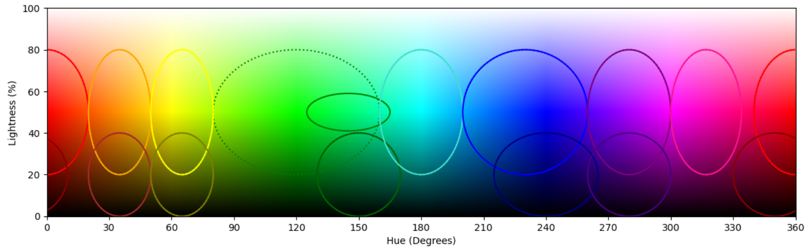

Figure 4.1: Internal model with statistical learning after observing twenty times the colour (h: 160, s: 100, l: 50), i.e. the right border of the initial green category

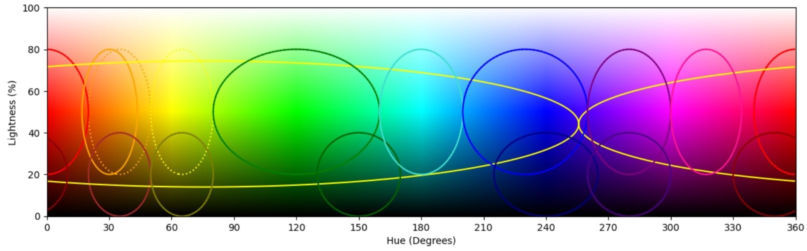

Figure 4.2: Internal model with naive learning after the colour (h: 35, s: 100, l: 50), i.e. the initial orange prototype, was labeled yellow

The *Context-Sensitive Learning Approach* can successfully adapt to the situation before. The learned model is shown in Figure **??**. Even if the target object is called green, the context-sensitive approach can adapt in a meaningful way. When turquoise is assigned, the learning process fails due to the problem with the fixed lower bound mentioned in Section 4.4.2. Even though this prevents the model from adapting to an unsuitable input, this is not considered a proper implementation of robustness.

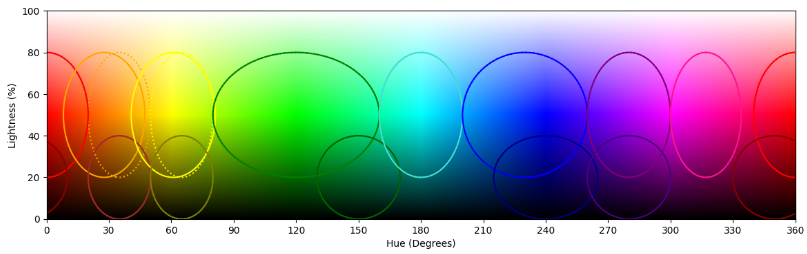

Figure 4.3: Internal model with naive learning after the colour (h: 35, s: 100, l: 50), i.e. the initial orange prototype, was labeled yellow

5 | Evaluation

To determine further problems when using the algorithms and to compare the different approaches, an experiment was conducted.

5.1 Experiment Setup

In the first part of the experiment, the participants were confronted with a description of the following task. They will see scenes with two discs each, one of which is indicated with an arrow. Their task is to give a description of this disc by choosing one colour of the colour set, see Section 3.3, such that another person could identify the target disc without seeing the arrow. The participants were informed that the goal is not to give a description as precisely as possible, but a successful communication. Since the colour set contains specific colour names, the participants could inform themselves about unknown colour terms. Since the colours are generated such that they are similar to the distractor, see Section 3.4, the participants can enter a question mark instead of a colour term to indicate that the colours are too close and that they were not able to find a suitable description. Internally, the category with the highest acceptability value of the target colour got an `add_diversity` property which is increased by five units in such a case. When choosing another colour based on this prototype, the `add_diversity` value is added to the hue component of the second colour. Therefore, the colours are more diverse only in the concerned region. The used instructions[1] in the experiment can be reviewed in Appendix A.2.

1: The experiment was conducted in German. Therefore, the instructions are also written in German.

After entering at least 60 valid descriptions (entering a question mark is not considered a valid description), the participants were introduced to the second task. They are confronted with the same setting as before, but this time none of the two discs were marked by an arrow. Instead, a description, i.e. a colour term, was given. The participants were asked to enter whether they think the left disc or the right disc is the referred object by entering `1` for the left disc and `2` for the right disc. Alternatively, they could enter "left" or "right" (in German). If they think that the description matches none of the discs or that referred object can not be uniquely identified, they should enter `0` or "none" (in German). The participants were not informed that the selection is a subset of 25 scenes with their own descriptions in random order. Therefore, the third option "none" was never the right answer. The second part is not used for evaluating the learning approaches but to validate the inputs. The motivation is that at least one self should be able to identify an object by his or her own description. Finally, the colour perception of each participant was tested by using the Ishihara test, i.e. the plates number 1, 13, 23, and 29 [Ish87]. The result of a participant with deviating colour perception would have been interesting but was not observed.

The experiment included twelve participants. The participants were separated into three groups, each got assigned a different learning algorithm. The setting of the participants did not differ depending on their assigned learning algorithm. Therefore, the inputs of the participants should not be significantly influenced by the learning algorithm. The only difference is that different learning algorithms adapt the prototypes in different ways. This should not affect the input of the participants but only on the colour generation as the colours are generated based on the prototypes. Therefore, the inputs of a participant can be simulated for the other learning algorithms as well, as every learning algorithm should be able to handle any pair of colours that a participant can separate. The effective group size is therefore twelve rather than four. Ten participants were male, two were female. Unfortunately, the genders were not equally represented. However, for this evaluation, this should not be a significant problem. It is investigated how the algorithms adapt to human input, not the other way around. A good learning algorithm should be able to adapt to any person regardless of gender. This paper is interested if the proposed solutions work and if context-sensitivity makes a difference in general. For an application of actual use, this factor should be considered.

To mitigate the problem of robustness, an upper limit for allowed border change is set to 750 units. If an adaption would cause a border change of more than 750 units, it is considered to be an unwanted outlier. Therefore, the model is not changed and reports a failure like if the optimizer reaches its iteration limit. The limit value was experimentally determined. The `simulator.py` script allows simulating other learning algorithms to the given inputs for different upper limits. Note that the unit considers the sum of the squared border changes.

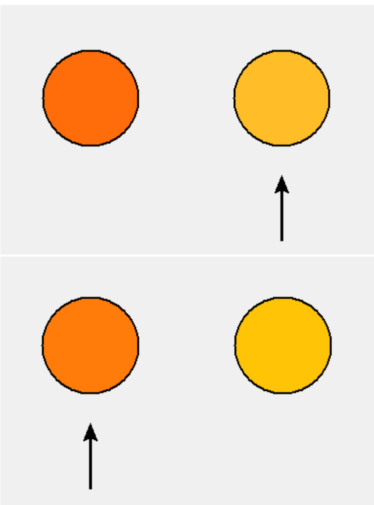

Figure 5.1: One person called both objects in very similar settings orange.

5.2 Discussion

The validation rate in the second part of the experiment, i.e. how many descriptions were correctly associated with the referred object, is between 0.84 and 1.0, meaning that participants made between zero and four mistakes out of 25 samples. Any worse result would have been rejected. In retrospect, the required validation rate maybe should have been higher, as robustness was not a core aspect of this paper. Some inconsistencies are probably by accident. For instance, one person called both objects *orange* in two very similar scenes in the course of their test run, see Figure 5.1. Keep in mind, that the system does not learn across participants, so everyone starts with the initial model. If these two inputs came from different persons, this would not be as problematic, besides effects as discussed in Section 4.4.3. But these two inputs came from the same person, so the same model has to deal with them.

The task description is very specific about what the participants are supposed to do, for instance, the goal is not to give an accurate description, but that another person could identify the object without the indication arrow. This note was introduced on purpose such that participants do not neglect the distractor object and comply with the Gricean Maxim relevance as mentioned in Section 2.1. However, there might be a bias in the formulation of the task. Even though the task description was accurate, explicitly mentioning that the goal is not giving a description as precise as possible but a successful communication, might be a bias against the acceptability value, used by the *Naive Learning Approach* and preferring the score value, used by the *Context-Sensitive Approach*. An indicator for this proposition is shown in Figure 5.2. The indicated object was described as *turquoise* even though this is a merely acceptable description. The naive learning algorithm made large changes to the model to adapt it only to this input, as shown in Figure 5.3. The context-sensitive algorithm could handle the input well, see Figure 5.4, which might be an indicator for the bias.

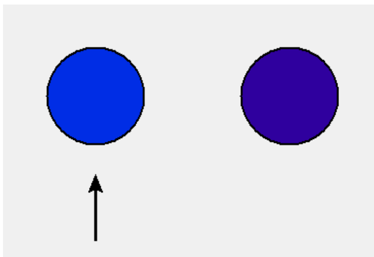

Figure 5.2: The indicated object was described as *turquoise* even though this is a merely acceptable description.

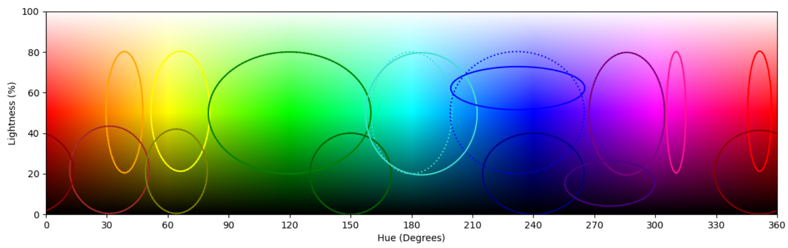

Figure 5.3: The naive algorithm made large changes to adapt to the input from Figure 5.2

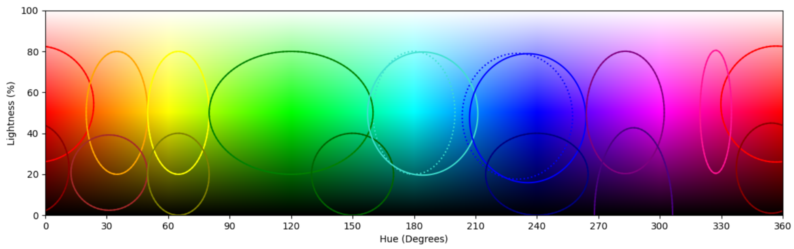

Figure 5.4: The context-sensitive algorithm could handle the input from Figure 5.2 well.

The experiments also revealed another problematic situation. Take for instance the scene depicted in Figure 5.5. The indicated disc was called *purple*. However, the hue value of the disc was even beyond the pink prototype from the perspective of the *purple* prototype. The optimizer solves this problem by compressing the *pink* prototype, as shown in Figure 5.6. Since the degree of how quick the exponential acceptance function decreases depends on the radius of the prototype, the optimizer finds a solution where the acceptability value of *purple* is higher than the one of *pink* because pink decreases very quickly through this adaption. In this example, this is not a good adaption because purple and pink are both basic colour terms. Note that there might be examples where one prototype is completely inside another. This might happen for very specific shades of a colour. Mast et al. [MFW16] use the example that *crimson* is a subtype of *red*. This might be modeled such that *crimson* is surrounded by red. This phenomenon concerns especially the *Naive Learning Approach* but also the *Context-Sensitive Learning Approach* might compress prototypes. *Pink* was the most frequent compressed prototype.

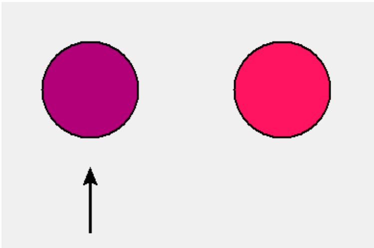

Figure 5.5: The indicated disc with a hue value of 319 was called *purple* but it is even beyond the *pink* prototype.

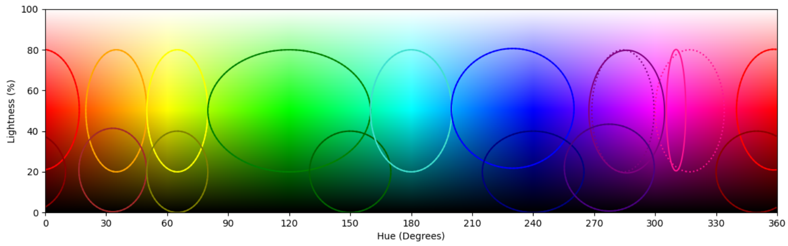

Figure 5.6: The input of Figure 5.5 caused the model to compress the *pink* prototype.

Since the sample size per learning algorithm is only $n = 4$ participants, an evaluation and comparison of the learning algorithms which separate the participants based on the used learning algorithm are not very meaningful. Therefore, the inputs of each participant are also applied to the other learning algorithms. The justification is explained in Section 5.1. Moreover, since the order of the inputs should not make a difference, the inputs of each participant are applied in a simulation to all three learning approaches in three different random orders for each learning approach. Thereby, from the input set of one participant, there are nine different learning traces simulated. The main research question of this paper is if a context-sensitive approach converges faster than context-insensitive approaches, i.e. the internal model approaches the user's model more quickly. Since the model is not changed if the description of the user and the description of the agent based on the corresponding metric are equivalent, the average border change should decrease over time. Figure 5.7 visualizes the outcome of the simulations. The first row shows the average border change for the different learning approaches. The second line shows the percentage of how often the adaption failed for the different learning approaches. Since a failed adaption does not change the model, i.e. the border change is zero, an approach that often fails would be preferred, as fails are treated as correctly predicted. To account for this problem, an error is treated as a border change of the upper border change limit, i.e. 750 units.

To highlight the progress of the learning approaches, a linear regression is used. Note that the behaviour of the border change is not linear. The regression is only used as an indicator for the trend of the border change. All approaches achieve a negative trend. However, a slope of -0.04 of the *Naive Learning Approach* is hardly progress. The *Context-Sensitive Learning Approach* achieved a slope of -0.29. This can be interpreted that with each observed input, the adaption of the model decreased by 0.29 units on average. This is a small approximation to the model of the user considering that context-sensitive approach adjusted the model

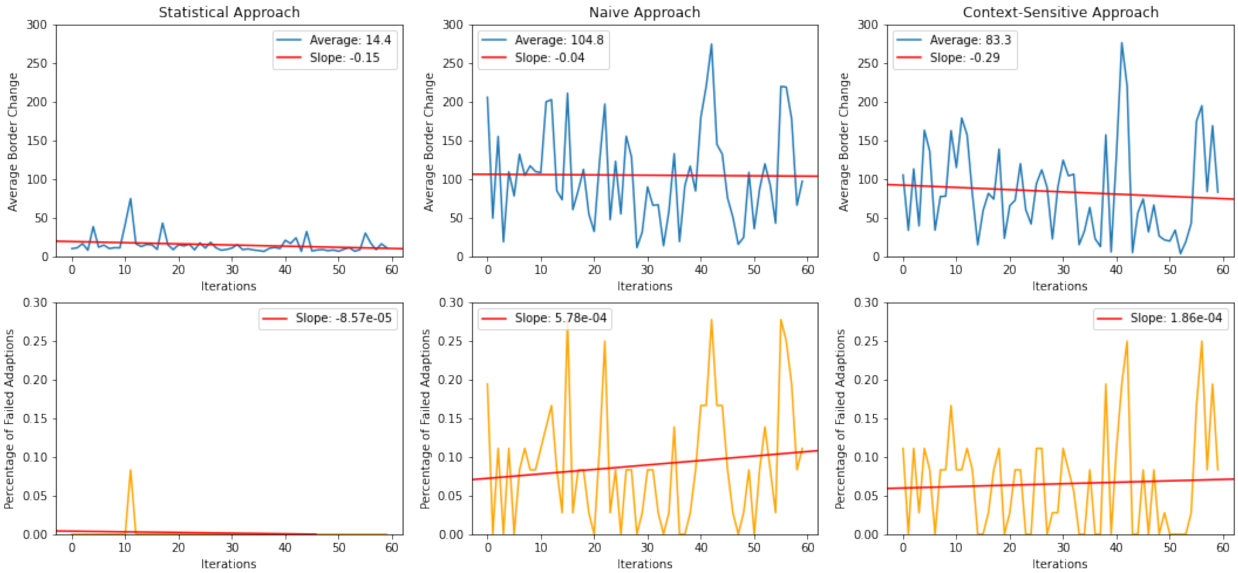

Figure 5.7: The first row shows the average border change for the different learning approaches. The second line shows the percentage of how often the adaption failed for the different learning approaches. The upper border change limit is 750. The lowest bound for the acceptability value is 0.02.

by 83.3 units on average per observed input. Even though the *Context-Sensitive Learning Approach* could not achieve a desirable adaption, it still exceeded the *Naive Learning Approach* not only by slope but also in terms of average adjustment. The naive algorithm adjusted the model by 104.8 units on average per observed input. One reason for this might be that the naive approach has a worse intrinsic robustness than the context-sensitive approach as described by the example in Figures 5.2, 5.3 and 5.4. The worse intrinsic robustness might also be an explanation for why the average errors of the naive approach increased over time, as the model got modified to an extent that the algorithm could not resolve again.

In the current analysis, the statistical approach seems to exceed the other approaches as the model is less adapted during the learning process. However, whereas the naive and the context-sensitive approach adjust the model when the prediction with regards to the used metric, i.e. acceptability or PRAGR score, differs from the description of the user, the statistical approach adapts the model after each observed input regardless of the prediction. To compare the statistical and the context-sensitive approach, the correct prediction rates are considered, as shown in Figure 5.8. An input is counted as correctly predicted if PRAGR would assign the same description as the user before the adaption of the model to the input. Keep in mind that the behaviour is not linear but a linear regression is used to visualize a trend. Since both approaches start at around 0.60, this can be considered to be the baseline of the initial model. The statistical approach hardly improves over time. To conclude that the statistical approach does not adapt properly, different values for the number of considered instances for the maximum likelihood estimation must be tested. An accurate parameter fitting is not in the scope of this paper.

The context-sensitive approach not only achieved a higher average correct prediction rate of 0.662 but also a slope three times as high as the statistical approach. However, it could not achieve a desirable adaption rate.

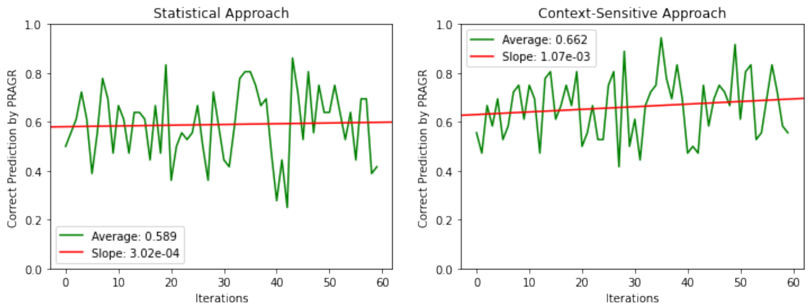

Figure 5.8: Comparison of the statistical and the context-sensitive approach in terms of correct predictions by PRAGR

Finally, the effects of the lower bound for the acceptability values are investigated by simulating the inputs again for each learning algorithm three times each in different random orders, this time with the lower bound disabled. The results are shown in Figures 5.9 and 5.10. Without the lower bound the context-sensitive approach hardly improved over time both in terms of average border change and correct prediction rate. The lower bound does not significantly improve or hinder the naive approach. While the average border change decreased by 14.5 units in comparison to the simulation using the lower bound, the slope got worse by 0.08 units. The correct prediction rate of the statistical approach changed. The lower bound does not influence the statistical approach itself, but it has an influence on the prediction of PRAGR. However, despite there being a visible improvement, keep in mind that the correct prediction rates are below the one the statistical approach scored using the lower bound. Therefore, the lower bound has a positive influence on the learning process.

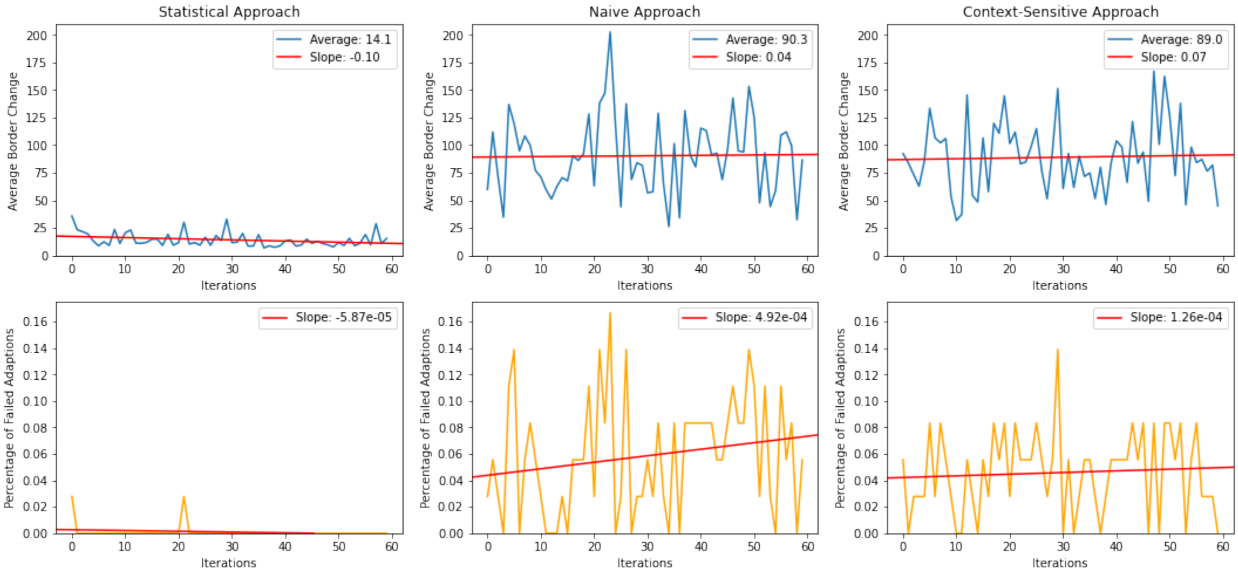

Figure 5.9: The first row shows the average border change for the different learning approaches. The second line shows the percentage of how often the adaption failed for the different learning approaches. The upper border change limit is 750. The lowest bound for the acceptability value is disabled.

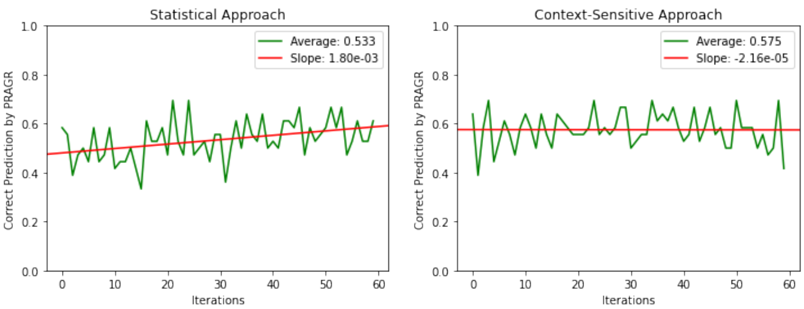

Figure 5.10: Correct predictions by PRAGR using the context-sensitive approach with disabled lower bound for the acceptability value

6 | Conclusion

None of the proposed learning approaches could provide an improvement to the internal model as good as desired. The statistical approach has a useful intrinsic robustness, since every input is weighted to the current model and has, therefore, a limited influence. However, the premise that the distribution of the observed examples can be approximated with the underlying normal distribution, cannot be met in a real life scenario as explained in Section 4.4.3.

Neglecting hierarchies of concepts creates problems especially with learning algorithm. As described in Section 3.3, there might be possibilities to avoid these problems in static models as used by Mast et al. [MFW16]. However, a learning algorithm has to know or learn the hierarchies as well such that the algorithm knows two things. First, two basic colour classes are not allowed to be adapted as described in Section 5.2 and Figure 5.6, where *pink* gets compressed such that *purple* also exceeds *pink* on both ends of the hue range of *pink*. Second, the algorithm has to know how to adapt the model if a colour competes with one of its subforms, e.g. *crimson* and *red*. Even though the colour selection tried to mitigate this problem for this paper, choosing which prototypes to adjust is still very challenging.

The naive and the context-sensitive approach turned out not to be sufficiently robust. Especially the naive approach adjusted prototypes to unrealistic extends in some cases. The context-sensitive approach performed the best as stated in the hypothesis. However, the experiments could not prove that the reason for this is that using a context-sensitive approach is superior in concept-learning with PRAGR. Other reasons might be a biased task description or the better intrinsic robustness in the context-senstive approach. After all, the context-sensitive approach could not achieve a good improvement for the internal models as well. Future work could investigate how robustness can be modelled in the learning approaches and how hierarchies in concepts can be modelled in PRAGR such that not all category are treated equally.

A | Experiment Conditions

A.1 Initial Prototype Configuration

Colourname	Hue (h_c, h_r)	Saturation (s_c, s_r)	Lightness (l_c, l_r)
red (rot)	(0,20)	(100,50)	(50,30)
orange (orange)	(35,15)	(100,50)	(50,30)
brown (braun)	(35,15)	(100,50)	(20,20)
yellow (gelb)	(65,15)	(100,50)	(50,30)
olive (olive)	(65,15)	(100,50)	(20,20)
green (grün)	(120,40)	(100,50)	(50,30)
leave-green (blattgrün)	(150,20)	(100,50)	(20,20)
turquoise (türkis)	(180,20)	(100,50)	(50,30)
blue (blau)	(230,30)	(100,50)	(50,30)
navy-blue (marinblau)	(240,25)	(100,50)	(20,20)
purple (lila)	(280,20)	(100,50)	(50,30)
indigo (indigo)	(280,20)	(100,50)	(20,20)
pink (pink)	(317,17)	(100,50)	(50,30)
bordeaux (bordeaux)	(350,20)	(100,50)	(20,20)

A.2 Task Description

The following texts show the tasks as they were described in German when collecting the input data.

A.2.1 Task 1: Entering Colournames

Im Folgenden wirst du Szenen mit jeweils zwei Scheiben sehen. Eine Scheibe ist mit einem schwarzen Pfeil gekennzeichnet. Deine Aufgabe ist es, die Scheibe mit nur einer Farbe zu beschreiben, sodass eine andere Person, die dieselbe Szene ohne Markierungspfeil sieht, das markierte Objekt eindeutig nur anhand deiner Beschreibung identifizieren kann. Es geht nicht darum, die markierte Scheibe möglichst präzise zu beschreiben. Es gibt kein richtig oder falsch! Entscheidend ist nur deine persönliche Auffassung, mit welcher Beschreibung eine andere Person das markierte Objekt möglichst sicher identifizieren kann.

Für die Beschreibung kannst du jeweils eine der folgenden Farben nutzen:

- Rot

- Orange
- Braun
- Gelb
- Olive
- Grün
- Blattgrün
- Türkis
- Blau
- Marinblau
- Lila
- Indigo
- Pink
- Bordeaux

Falls dir einige dieser Farbbezeichnungen unbekannt sind oder du dir nicht sicher bist, kannst du dich jetzt vor Beginn selbstständig mit diesen vertraut machen.

Solltest du der Meinung sein, in einer Szene keine passende Beschreibung zu finden, bspw. weil die beiden Objekte deiner Meinung nach zu ähnlich sind, dann schreibe in das Beschreibungsfeld ein **Fragezeichen (?)**.

Diese Aufgabe wird etwa 10 Minuten dauern.

A.2.2 Task 2: Validating the Inputs

Nun übernimmst du die andere Rolle. Du wirst wieder Szenen mit jeweils zwei Scheiben sehen. Dieses Mal ist keine Scheibe markiert. Stattdessen ist eine Beschreibung gegeben. Deine Aufgabe ist zu entscheiden, welche Scheibe durch die Beschreibung referenziert werden soll. Gib dazu in das Eingabefeld entweder **links** bzw. **1** oder **rechts** bzw. **2** ein.Wenn du der Meinung bist, dass die Beschreibung keine der beiden Scheiben beschreibt oder uneindeutig ist, gib in das Eingabefeld **keine** bzw. **0** ein.

B | Proofs for Statistical Learning Approach

$$\widehat{\mu}_{n+1} = \frac{1}{n+1} \sum_{i=1}^{n+1} x_i \tag{B.1}$$

$$= \frac{1}{n+1}((\sum_{i=1}^{n} x_i) + x_{n+1}) \tag{B.2}$$

$$= \frac{n}{n+1} \cdot \frac{1}{n} \sum_{i=1}^{n} x_i + \frac{1}{n+1} \cdot x_{n+1} \tag{B.3}$$

$$= \frac{n}{n+1} \cdot \mu_n + \frac{1}{n+1} \cdot x_{n+1} \tag{B.4}$$

$$= \frac{1}{n+1}(n \cdot \mu_n + x_{n+1}) \tag{B.5}$$

$$\widehat{\sigma}^2_{n+1} = \frac{1}{n+1} \sum_{i=1}^{n+1} (x_i - \widehat{\mu}_{n+1})^2 \tag{B.6}$$

$$= \frac{1}{n+1}(\sum_{i=1}^{n} (x_i - \widehat{\mu}_{n+1})^2 + (x_{n+1} - \widehat{\mu}_{n+1})^2) \tag{B.7}$$

$$= \frac{n}{n+1} \cdot \frac{1}{n} \sum_{i=1}^{n} (x_i - \widehat{\mu}_{n+1})^2 + \frac{1}{n+1}(x_{n+1} - \widehat{\mu}_{n+1})^2 \tag{B.8}$$

$$= \frac{n}{n+1} \cdot \frac{1}{n} \sum_{i=1}^{n} (x_i - (\mu_n + \Delta))^2 + \frac{1}{n+1}(x_{n+1} - \widehat{\mu}_{n+1})^2 \tag{B.9}$$

$$= \frac{n}{n+1} \cdot \frac{1}{n} \sum_{i=1}^{n} ((x_i - \mu_n) - \Delta)^2 + \frac{1}{n+1}(x_{n+1} - \widehat{\mu}_{n+1})^2 \tag{B.10}$$

$$= \frac{n}{n+1} \cdot (\sigma_n^2 + \Delta_\mu^2) + \frac{1}{n+1}(x_{n+1} - \widehat{\mu}_{n+1})^2 \tag{B.11}$$

$$= \frac{1}{n+1}(n \cdot (\sigma_n^2 + \Delta_\mu^2) + (x_{n+1} - \widehat{\mu}_{n+1})^2) \tag{B.12}$$

$$\frac{1}{n}\sum_{i=1}^{n}((x_i-\mu_n)-\Delta)^2 = \frac{1}{n}\sum_{i=1}^{n}((x_i-\mu_n)^2 - 2\cdot(x_i-\mu_n)\Delta+\Delta^2) \tag{B.13}$$

$$= \frac{1}{n}\sum_{i=1}^{n}(x_i-\mu_n)^2 - \frac{2}{n}\sum_{i=1}^{n}(x_i-\mu_n)\Delta + \frac{1}{n}\sum_{i=1}^{n}\Delta^2 \tag{B.14}$$

$$= \sigma_n^2 - \frac{2\Delta}{n}\sum_{i=1}^{n}(x_i-\mu_n) + \Delta^2 \tag{B.15}$$

$$= \sigma_n^2 - \frac{2\Delta}{n}(\sum_{i=1}^{n}x_i - n\cdot\mu_n) + \Delta^2 \tag{B.16}$$

$$= \sigma_n^2 - 2\Delta(\frac{1}{n}\sum_{i=1}^{n}x_i - \mu_n) + \Delta^2 \tag{B.17}$$

$$= \sigma_n^2 - 2\Delta(\mu_n-\mu_n) + \Delta^2 \tag{B.18}$$

$$= \sigma_n^2 + \Delta^2 \tag{B.19}$$

References

[DR95] R Dale and E Reiter. Computational interpretations of the Gricean maxims in the generation of referring expressions. In: *Cognitive science* 19(2):(1995), 233–263 (pages 3, 5, 8).

[FMG15] Z Falomir, L Museros, and L Gonzalez-Abril. A model for colour naming and comparing based on conceptual neighbourhood. An application for comparing art compositions. In: *Knowledge-Based Systems* 81:(2015), 1–21 (page 12).

[Fal+13] Z Falomir et al. Measures of similarity between objects based on qualitative shape descriptions. In: *Spatial Cognition & Computation* 13(3):(2013), 181–218 (page 12).

[Gar04] P Gardenfors. Conceptual spaces as a framework for knowledge representation. In: *Mind and matter* 2(2):(2004), 9–27 (page 9).

[Gär11] P Gärdenfors. Semantics based on conceptual spaces. In: *Indian Conference on Logic and Its Applications*. Springer. 2011, pp. 1–11 (page 9).

[GS12] RL Goldstone and JY Son. *Similarity.* Oxford University Press, 2012 (page 5).

[Gri75] HP Grice. Logic and conversation. In: *Speech acts*. Brill, 1975, pp. 41–58 (pages 3, 7).

[Ish87] S Ishihara. *Test for colour-blindness*. Kanehara Tokyo, Japan, 1987 (page 26).

[LT09] J Lawry and Y Tang. Uncertainty modelling for vague concepts: A prototype theory approach. In: *Artificial Intelligence* 173(18):(2009), 1539–1558 (page 7).

[Mas17] V Mast. Referring expression generation in situated interaction. PhD thesis. Universität Bremen, 2017 (pages 5, 6, 8–11).

[MFW16] V Mast, Z Falomir, and D Wolter. Probabilistic Reference and Grounding with PRAGR for Dialogues with Robots. In: *Journal of Experimental & Theoretical Artificial Intelligence* 28(5):(2016), 889–911 (pages 1, 2, 5–7, 11–13, 21, 28, 33).

[MW13] V Mast and D Wolter. A probabilistic framework for object descriptions in indoor route instructions. In: *International conference on spatial information theory*. Springer. 2013, pp. 185–204 (pages 7, 10).

[Mas+14] V Mast et al. Referential grounding for situated human-robot communication. In: *Proceedings of DialWatt/SemDial*:(2014) (page 6).

[NMD16] VK Nagaraja, VI Morariu, and LS Davis. Modeling context between objects for referring expression understanding. In: *European Conference on Computer Vision*. Springer. 2016, pp. 792–807 (page 5).

[RW99] A Roorda and DR Williams. The arrangement of the three cone classes in the living human eye. In: *Nature* 397(6719):(1999), 520–522 (page 5).

[Van16] K Van Deemter. *Computational models of referring: a study in cognitive science*. MIT Press, 2016 (pages 3, 5).

[Yu+18] L Yu et al. Mattnet: Modular attention network for referring expression comprehension. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 1307–1315 (page 5).

Declaration of Authorship

Ich erkläre hiermit gemäß § 9 Abs. 12 APO, dass ich die vorstehende Bachelorarbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Des Weiteren erkläre ich, dass die digitale Fassung der gedruckten Ausfertigung der Bachelorarbeit ausnahmslos in Inhalt und Wortlaut entspricht und zur Kenntnis genommen wurde, dass diese digitale Fassung einer durch Software unterstützten, anonymisierten Prüfung auf Plagiate unterzogen werden kann.

Bamberg, den ______________________ ____

Matthias Müller