SAARLAND UNIVERSITY

Faculty of Mathematics and Computer Science
Department of Computer Science
BACHELOR THESIS
Degree Program: Computer Science (English)

Understanding Pragmatic Reasoning Through Eye Movement Patterns in Reference Games

submitted by
Tymur Mykhalievskyi
Saarbrücken
April 2025

Advisors:
Prof. Dr. Vera Demberg
Computer Science and Computational Linguistics
Saarland University
Saarbrücken, Germany

Dr. John Duff
Computer Science and Computational Linguistics
Saarland University
Saarbrücken, Germany

Reviewer 1: Prof. Dr. Vera Demberg
Reviewer 2: Prof. Dr. Sven Apel

Saarland University
Faculty MI – Mathematics and Computer Science
Department of Computer Science
Campus - Building E1.1
66123 Saarbrücken
Germany

Declarations

Acknowledgements

Abstract

Contents

Introduction

If one says “I am going to Munich this week. My mother lives there.”, you will interpret this as meaning they are visiting their mother, even though it is not explicitly stated. This is called an implicature, without explicitly stating something one can still deliver the information. Human communication is full of such implicit constructions. One reason for this may be to save cognitive effort. What rules do people unconsciously follow during communication to make it more efficient?

In 1975 a British philosopher Paul Grice finalized four types of maxims (**?**). Maxim of Quantity: Provide as much information as required, do not provide more information than required. Maxim of Quality: Be truthful, only say that for which you have adequate evidence. Maxim of Relation: be relevant. Maxim of Manner: avoid ambiguity. Going back to the example with traveling, we can assume a speaker is obeying the maxims. Therefore, the information is relevant and the right amount is provided, so the second sentence about where the mother lives is not just a disconnected fact. Hence, we build an implicature that one is visiting their mother.

One way to study this is through reference games. In these games, participants engage in a collaborative task, often involving the identification or description of objects, where effective communication and reasoning play key roles. Over the years, these reference games have become a popular experimental paradigm to explore how individuals reason about others’ intentions and strategies in communication (**??**). A simple example is presented in **??**. Imagine someone is talking to you and uses the word “blue” to refer to one of these objects. Which object are they talking about? If you answered blue square, congrats, it is considered the correct solution. Do not worry, if you are confused. If we consider the possibilities of the speaker, the two completely unambiguous messages available to them are “green” and “circle”. Hence, if they would have referred to one of the other objects, there are clear messages to do that. Thus, we are left with messages “blue” and “square”. Although the message “blue” corresponds to two objects, the blue circle can be referred to unambiguously by using message “circle”. Similar logic can be applied to the message “square”. So both of them can be inferred to point to the blue square. All this reasoning is built upon the Gricean maxims, as we expect from the speaker to be as concise, unambiguous, relevant and truthful as they can be.

On the other hand, one could notice that the reference games are not as intuitive as the traveling example. It is still a limitation that we will have to keep for now. And this study could also shed light on this problem by understanding what exactly people are doing to solve this kind of problems.

In order to deepen the understanding, a formal model was developed, it is called Rational Speech Act model. It tries to mimic a recursive sequence of reasoning between speaker and listener (**?**). Despite significant progress in understanding the cognitive processes underlying these tasks, much remains unknown about the specific strategies individuals employ when solving particular problems within these games.

This study seeks to expand on prior research by incorporating a novel dimension: tracking participants’ eye gaze during reference games. Eye gaze offers valuable insight into how people process information, make decisions, and employ strategies. By capturing where and when participants direct their attention, we can gain a deeper understanding of the cognitive mechanisms at play, including how individuals prioritize certain visual cues and how these cues influence their reasoning strategies. This approach has shown to be a very insightful tool (**?**).

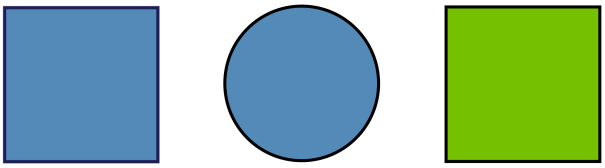


Figure 1: An example of reference game. Same example is shown in **?**. A speaker utters "blue", which object are they referring to?

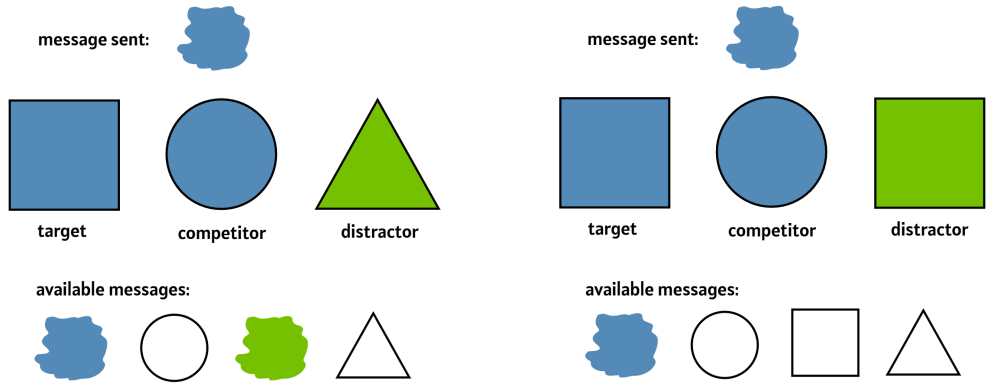
In particular, this study aims to answer the question: How do gaze patterns correlate with the accuracy and strategies used to solve specific communicative challenges in reference games? By integrating eye-tracking data with the analysis of reasoning in these games, this paper contributes to a richer understanding of the decision-making processes involved in collaborative communication and problem-solving.

Chapter 1
Related Work

1.1 Reference Games

Although, the idea of communication as a signaling game goes back to **?**, we will be focusing on a type of a signaling game called a reference game as presented in **?**. The game tries to mimic the challenges of communication that people face on daily basis, as we discussed in the **??**. At first, an instance of a reference game looked as in **??**, that is, no information about the available messages is given to the speaker and listener. The later instances, on the other hand, include this information. The examples are shown in **??**. The goal for newer version is the same, that is, to identify which object a speaker is referring to. This change allows for more controlled experiments by increasing the variability in setups. In addition, it should improve clarity, for instance, participants should not wonder, why wouldn't the speaker just utter the location of the object instead of specifying the properties.

Let's take a closer look at **??**. An uttered message is presented on the top. We will denote the object being referred to as a target, a competitor is an object that shares the message property with the target. While a distractor does not share the sent message property, but could share another property with the target depending on the difficulty of the trial. Note that, obviously, captions target, competitor and distractor are not available to the participants. The difference between the simple and complex trials in **??** mainly in how the distractor is constructed. In particular, in the simple trial it does not share any properties with the target, while in the complex it does. The simple example **??** can be solved without considering the distractor. That is, one could count the matching messages from the available ones. In this case, it would be 1 for blue square, 2 for blue circle and 2 for green triangle. Hence, the target is blue square, as "blue" is the only message that could refer to it. This way of solving is not necessarily what people tend to do, but it is one way of interpreting the difference between simple and complex trials. Because if you apply the same strategy to the complex example in **??**, both target and competitor have two matching messages. On the other hand, if you try to solve these examples yourself, you will probably end up recursively reasoning of what the



(a) simple (b) complex

Figure 1.1: Two instances of reference games with different difficulties.

speaker could have said had they had another target. The simple and complex in this case still appear to their names, you will need a more robust recursion in order to solve the complex one comparing to the simple one.

1.2 Rational Speech Act Model

Studying this phenomena needs a formalized approach. One such model, called the Rational Speech Act (RSA) was developed. It mimics how a speaker and a listener reason about each other. A detailed explanation can be found in the manuscript by **?** as well as in the article **?**. We will go through the main ideas of how listener and speaker interact with each other. Firstly, take a look at the matrix M_s from the **??**. Each columns is a one-hot encoding of an objects, in other words this matrix encodes which objects match the literal meaning of each message, this matrix is constructed for the simple example in **??**.

$$M_s = \begin{array}{c} \\ \\ \\ \\ \end{array} \begin{array}{c} \begin{array}{ccc} \square & \bigcirc & \triangle \end{array} \\ \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix} \end{array} \tag{1.1}$$

Now let us define a listener matrix, each row shows conditional probabilities for the objects given a message. Accordingly a speaker matrix has columns that depict conditional probabilities of messages given an object.

Subsequently we arrive at a literal listener **??** and speaker **??**. Simply put a literal speaker would output one of the matching messages with equal probability for the given target. For example, if green triangle is provided for the speaker, they would refer to it be uttering “green” or “triangle” with equal probability. While literal listener would interpret the ambiguous messages with equal probabilities.

$$L_0(M_s) = L(M_s) = \begin{array}{c} \\ \\ \\ \\ \end{array} \begin{array}{c} \begin{array}{ccc} \square & \bigcirc & \triangle \end{array} \\ \begin{bmatrix} 0.5 & 0.5 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix} \end{array} \tag{1.2}$$

$$S_0(M_s) = S(M_s) = \begin{array}{c} \\ \\ \\ \\ \end{array} \begin{array}{c} \begin{array}{ccc} \square & \bigcirc & \triangle \end{array} \\ \begin{bmatrix} 1 & 0.5 & 0 \\ 0 & 0.5 & 0 \\ 0 & 0 & 0.5 \\ 0 & 0 & 0.5 \end{bmatrix} \end{array} \tag{1.3}$$

One could see that such approach would not solve even a simple trial. However, if there is a completely unambiguous message, the literal listener would be able to correctly identify the target. The way we derived the two matrices is just a normalization within columns or rows, correspondingly for the listener and speaker. We can keep applying this technique recursively, to find more complex listeners and speakers. That is, a speaker would normalize within columns the matrix previously normalized within rows. In this way we can derive an L_1 listener **??**, also called a first-order cooperative listener. Note that **?** and **?** apply different strategies to construct L_1 and L_2 listeners, we will stick to the **?** variation.

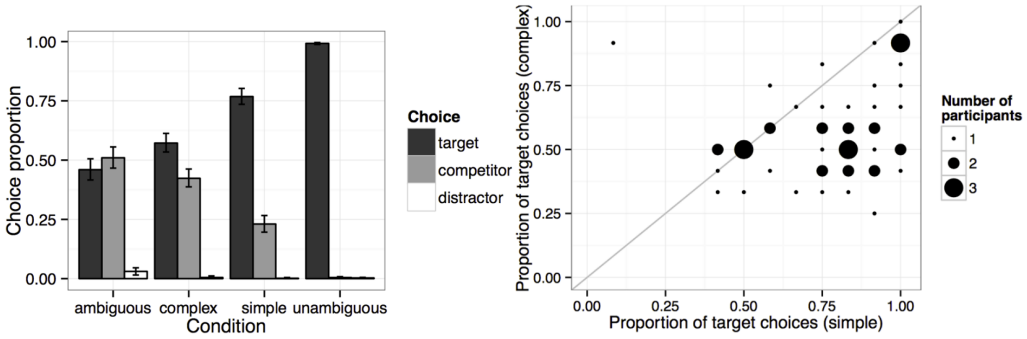
$$L_1(M_s) = L(S(M_s)) = \begin{array}{c} \\ \\ \\ \\ \end{array} \begin{array}{c} \begin{array}{ccc} \square & \bigcirc & \triangle \end{array} \\ \begin{bmatrix} 0.66 & 0.33 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix} \end{array} \tag{1.4}$$

L_1 listener gives the highest probability to the target with message blue. Repeating this procedure further to get to deeper recursion increases the probability of target being chosen. In addition, RSA model has a greed parameter α which amplifies the probabilities. $\alpha = \infty$ would result in simply choosing the object with the highest probability. Now let's take a look at the complex case and see how it differs from the simple one. The matrix M_c is given in **??**.

$$M_c = \begin{array}{c} \\ \\ \\ \\ \end{array} \begin{array}{c} \begin{array}{ccc} \square & \bigcirc & \square \end{array} \\ \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \end{array} \tag{1.5}$$

Going through the same steps to derive the L_1 listener, we get **??**.

$$L_1(M_c) = L(S(M_c)) = \begin{array}{c} \\ \\ \\ \\ \end{array} \begin{array}{c} \begin{array}{ccc} \square & \bigcirc & \square \end{array} \\ \begin{bmatrix} 0.5 & 0.5 & 0 \\ 0 & 1 & 0 \\ 0.33 & 0 & 0.66 \\ 0 & 0 & 0 \end{bmatrix} \end{array} \tag{1.6}$$



(a) Proportions of target, competitor and distractor choices in their experiment.

(b) Proportion of target choices in simple and complex conditions by participant.

Figure 1.2: Plots from **?**.

One important difference is that depth of recursion for the L_1 listener is not enough to assign the highest probability to the target. Here the "blue" row has the same probabilities for the distractor and the target. Note that in this case the greed parameter α would not be able to help. So instead we consider a deeper level of recursion and introduce an L_2 listener **??**, also called second-order cooperative listener.

$$L_2(M_c) = L(S(L(M_c))) = \begin{array}{c} \\ \\ \\ \\ \end{array} \begin{array}{c} \begin{array}{cccc} \square & \bigcirc & \square & \triangle \end{array} \\ \begin{bmatrix} 0.6 & 0.4 & 0 \\ 0 & 1 & 0 \\ 0.33 & 0 & 0.66 \\ 0 & 0 & 0 \end{bmatrix} \end{array} \tag{1.7}$$

So as one can see the L_2 can correctly identify the target considering the highest probability. Hence, the main point to take from here is that L_1 listener can solve the simple task, but cannot solve the complex one, while the L_2 listener is able to solve both.

Further expending on this, the previous research shows, that the modeled listeners align with the empirical data. In particular, see **??** taken from **?**. **??** shows that indeed the difficulty gets harder going from unambiguous to simple and further to complex trials. On the other hand, **??** shows that there are roughly 3 clusters present depending on whether one can solve only simple, both or neither of trials. This strongly supports the alignment with L_0, L_1 and L_2 listeners. However, very important to note that we are only talking about the alignment of RSA model's accuracy with the empirical data, while the concrete strategies are not taken into account.

Now we will proceed further, and make a hypothesis about how people could be solving these problems. A key difference between simple and complex trials is the fact that solving complex trials requires one to consider the distractor as well due to the matching feature with the target, while in the simple one, the distractor can be ignored completely. This can be demonstrated by the following matrix transformations. If one does not take into account the distractor the M_s, M_c will instead look as in **??** and **??** correspondingly.

$$M'_s = \begin{array}{c} \\ \\ \\ \\ \end{array} \begin{array}{c} \begin{array}{cc} \blacksquare & \bullet \end{array} \\ \begin{bmatrix} 1 & 1 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \end{array} \tag{1.8}$$

$$M'_c = \begin{array}{c} \\ \\ \\ \\ \end{array} \begin{array}{c} \begin{array}{cc} \blacksquare & \bullet \end{array} \\ \begin{bmatrix} 1 & 1 \\ 0 & 1 \\ 1 & 0 \\ 0 & 0 \end{bmatrix} \end{array} \tag{1.9}$$

Applying L_1 transformation to the M'_s we get **??** which accomplishes the same as in **??**.

$$L_1(M'_s) = \begin{array}{c} \\ \\ \\ \\ \end{array} \begin{array}{c} \begin{array}{cc} \blacksquare & \bullet \end{array} \\ \begin{bmatrix} 0.66 & 0.33 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \end{array} \tag{1.10}$$

On the other hand, neither L_1 nor L_2 can solve the matrix M'_c. In fact no depth of recursion is helpful in this case as $L_0(M'_c) = L_1(M'_c) = L_2(M'_c)$ (**??**).

$$L(M'_c) = L(S(M'_c)) = L(S(L(M'_c))) = \begin{array}{c} \\ \\ \\ \\ \end{array} \begin{array}{c} \begin{array}{cc} \blacksquare & \bullet \end{array} \\ \begin{bmatrix} 0.5 & 0.5 \\ 0 & 1 \\ 1 & 0 \\ 0 & 0 \end{bmatrix} \end{array} \tag{1.11}$$

This leads us to a research question of whether people achieve L_1 accuracy by not considering the distractor and applying reasoning deeper than one of a literal speaker. Or they include the distractor in their reasoning but simply lack the depth of recursion therefore failing to solve the complex trials.

1.3 How Eye Tracking is Useful

We will take a look at a related field with different kind of tasks, this strategy has shown to be particularly informative and insightful there. The Raven Progressive Matrices, commonly referred to as the Raven Tests, are a set of nonverbal intelligence tests designed to measure abstract reasoning and problem-solving abilities through pattern recognition and logical inference. Such test usually contains 8 objects arranged in a 3 by 3 grid with one object missing, as well as the set of possible answers displayed below the matrix. Each matrix either has a particular rule it is constructed by or a mix of them. An example is presented in **??**.

Researches suggest that there are two main strategies for solving the Raven Tests constructive matching and response elimination (**?**) and later followed by **?**. The former is described as successively finding rules by which the matrix is constructed, until the

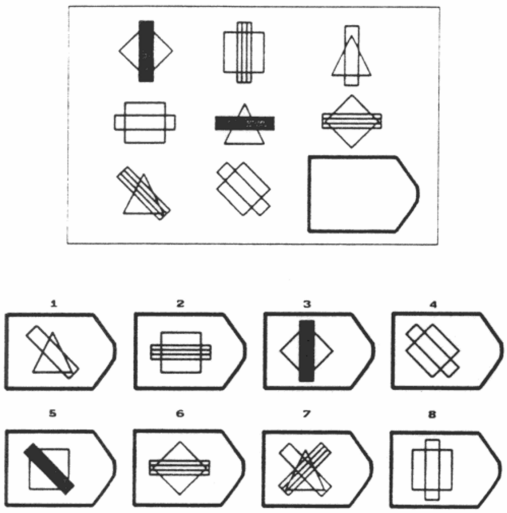


Figure 1.3: An example of Raven item. The upper part is the matrix, while the bottom is possible answers. The matrix is constructed as follows. The lines orientation is constant within rows. While the shapes and line appearances are obeying the distribution-of-three-values rule. Simply put same three values are present in each row. The correct answer is 5.

answer is fully derived. And the later means that rather than going through the matrix, one goes over the possible answers and eliminates them one by one, ending up with the correct one in the end. The less efficient of these, response elimination, seemed to be used more by lower ability subjects on more difficult items.

The two strategies can be identified by the patterns of one's attention, hence, eye gaze. The constructive matching being focused on the matrix and systematically going through rows and columns of it. **?** expand on the eye tracking experiments in this research question by recording eye gaze as well as the verbal comments during the process of solving the tasks. A very detailed sequence of actions is acquired therefore giving an insight into how one uses the constructive matching strategy to solve Raven Progressive Matrices. On the other hand, the response elimination involves a lot of toggling between the possible answers and the matrix. In order to deepen the understanding in this problem **?** develop a set of features to encode ones attention. Such features include for example Time on Matrix, Time on Alternatives (possible answers) or Number of Toggles between the possible answers and the matrix. The authors proceed to report the correlation between the features and the percent of ones correct answers. Indeed, the results show statistically significant negative correlation of Time on Alternatives and Number of Toggles with overall score. These findings further supports theory about the difference in effectiveness in the two strategies.

One can see based on these studies why and how the eye tracking is useful in the reference games. In our case as discussed in the end of **??** there are multiple ways people could reach L_1 accuracy. Therefore suggesting that the two potential strategies would be distinguished by the use of distractor. At the same time, there are still L_0 and L_2 listeners present in the experiment which makes the distinction more difficult. On the other hand, as there is no previous work on eye tracking reference games, the study is also highly exploratory.

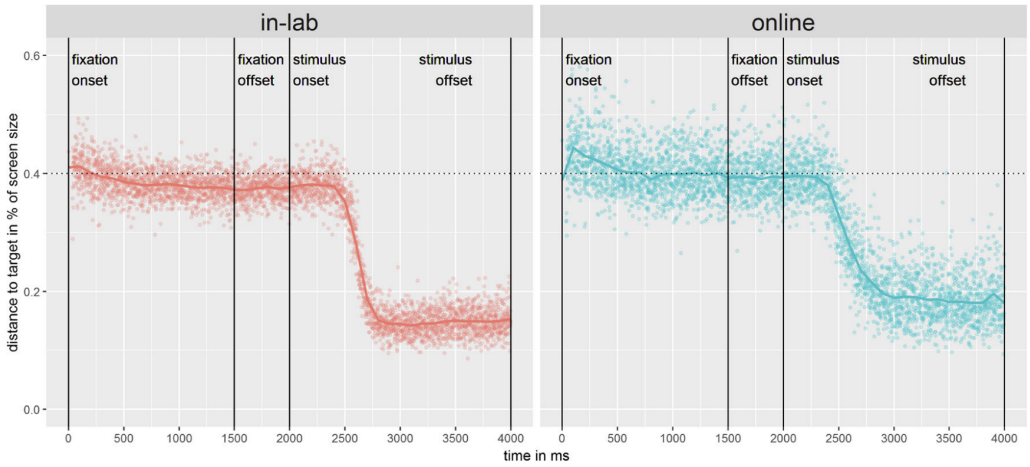


Figure 1.4: Figure from **?**. Fixation task results. Each dot denotes a single recorded data point in distance to a target in percentage of screen size over time.

1.4 Out of Lab Eye Tracking

Up until now most of the eye tracking studies have been using the laboratory equipment in order to conduct the experiments. This is very important as a reliable and precise method is needed for such experiments. On the other hand, this approach requires people to be physically present in the laboratory, which makes the experiment far more difficult to conduct in comparison to participants answering a series of questions on their laptops. Hence, a different approach was chosen. This experiment will incorporate participants' webcams to get collect the eye tracking part of the data. In particular a library called WebGazer is used **?**. Details about the implementation will be discussed in the following sections.

Although, on the first glance, the effectiveness of such approach can be debatable, there is work in favor of the method. Starting from the article **?** where they take a look into online webcam-based eye tracking comparing it to a respective in-lab experiment. Along with a more fresh research article which also makes this comparison (**?**). Both of them conclude that while WebGazer is still inferior to the lab equipment in terms of precision, the measurements are reasonably accurate. In particular, taking a look at the results of **?** shown in **??**. This figure depicts a particular fixation on the target which was shown after 2000 ms. It takes some time for one to react and for the software to capture the eye movement. Then we observe the saccade in both settings, on average the saccade took 450 ms (750 ms in the online case). The accuracy was 171 px (207 px online), which translates to about 3.94° visual angle in the in-lab setting. In addition, it is visible that the online setting has higher variance.

Taking into account the fact that each problem statement by itself consists of multiple objects located on one page, it is not hard to setup them far apart to mitigate the decline in precision. The shorter saccade length is not of the essence in our case.

Chapter 2
Concept

Previous work on individual differences **?** has focused on differences in underlying pragmatic reasoning tendencies. However participants with the same underlying pragmatic tendencies may underperform based on the information in the problem that they pay attention to. Take for example the hypothesis we saw in **??**, there not paying attention to the distractor would mean that the problem simply becomes unsolvable. Therefore, cognitive abilities such as attention and working memory are most likely not the only features contributing to the performance of participants in these tasks.

This thesis aims to shed light on the realm of where the attention is spent during the pragmatic reasoning problem-solving task. And the eye tracking data will be used to investigate this.

2.1 Research Questions

2.1.1 Estimating the Posterior

At first, we are interested in predicting the posterior, that is the probability of a participant to solve a problem give the eye gaze features as well as the general information about the trial.

Research Hypothesis 1

As we discussed in **??**, the distractor in the hypothesis is a crucial part of the problem. Therefore, it is important to understand how participants interact with the distractor. The first research hypothesis is H1: Proportional time on distractor is positively associated with accuracy on Complex trials.

Research Hypothesis 2

The second research question is about the messages that are available to the participants. The second research hypothesis is H2: Proportional time on available messages is positively associated with accuracy on Simple trials. This hypothesis is mainly based on the idea that while the unambiguous trials can be solved without considering the available messages, the simple ones require one to consider the available messages. In addition, the Complex trials can be solved without looking at the available messages.

2.1.2 Estimating the Likelihood

Because of the same patterns of information, we expect that skilled participants should in general have a different attention profile in Simple trials than in Complex trials. Therefore, we would like to do a slightly alternative analysis, estimating the likelihood directly instead of trying to estimate the posterior. In order to realize this, we will only take the correctly solved trials into account and predict the probability of a each fixation to be on the area of interest.

Research Hypothesis 3

The third research hypothesis is H3: on correctly solved trials, Complex trial condition is positively associated with the probability of fixation being on the distractor.

Research Hypothesis 4

The fourth research hypothesis is H4: on correctly solved trials, Simple trial condition is positively associated with the probability of fixation being on the available messages.

Chapter 3
Method

3.1 Data Collection

The experiment was hosted through an open-source crowdsourcing client/system server system LingoTurk **?**. While the participants were recruited via Prolific **?** - a subject pool for online experiments. The following criteria were used to filter the participants: native English speaker located in the UK, age in range 18-40, minimal approval rate of 95%, number of previous submissions must be ar least 20 studies, participants cannot have taken part in any of the related studies our group has conducted before. The estimated length of the experiment was 22 minutes, while the actual median was 19 minutes 30 seconds. The participants were paid 3.89 pounds which is equivalent to minimal hour wage in Germany. The participants were paid only in case of successful completion of the full experiment. The experiment was conducted in a web browser and the participants were asked to use a computer with functional webcam.

3.1.1 Reference Games

There are 3 conditions of the reference games: Simple, Complex and Unambiguous. In the **??** we mainly talked about the Simple and Complex conditions. The Unambiguous condition suits two main purposes. First of all, it acts as a sort of filler, so that participants do not get used to the same type of problems. Second of all, it acts as a control check, that is, participants who do not reach 75% accuracy on the Unambiguous trials are excluded from the analysis.

The trials were generated using 3 colors: red, green and blue, and 3 shapes: square, circle and triangle. In total there are 72 combinations of unique trials. The full list of trials can be found **TODO: insert link to apendix**. The code used to generate trials can be found at **TODO: insert link to the source code** For each condition every unique sent message was repeated exactly twice. Which results in 12 trials of every condition. Furthermore one simple and one complex trial was picked to be repeated once again in the very end of the experiment. These were so called strategy trials where we would ask the participants to

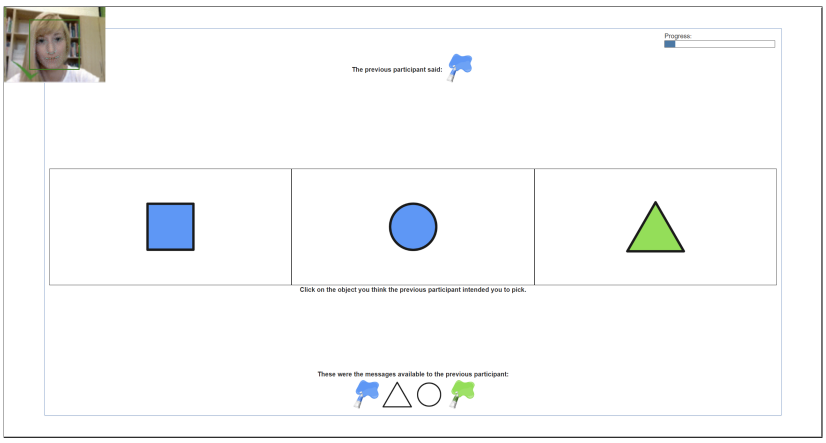


Figure 3.1: Example trial

explain their reasoning behind the choice of the object. All trials were randomly shuffled before the experiment, except for the strategy trials which were always at the end of the experiment.

Before the main part of the experiment with 36 + 2 trials, the participants were asked to do a speaker's job in reference games with 3 Unambiguous trials and 1 completely ambiguous one. The participants were asked to describe the object in the ambiguous trial in a way that the listener would be able to pick the correct object. Later they were told that a previous participant had already done the speaker's job and they were to do the listener's job, which was the main part of the experiment with 36 + 2 trials. An example trial can be seen in **??**. Depending on the zoom and resolution on the page the sizes and absolute positions of the images would vary a lot. Hence, the absolute positions in pixels as well as sizes of the images were saved during the experiment.

3.1.2 Eye Tracking

The eye tracking was done via library WebGazer **?**. The library was used to track participants' gaze on the screen. There was a calibration in the beginning of the experiment after the practice trials where participants did the speaker's job, this allowed to put the calibration as close to the main experiment as possible. The calibration was adapted from the one used in the demo of WebGazer **?**. Although, we included 11 points instead of 9, the additional points were put on the objects' places. Each point had to be clicked 5 times. The setup can be seen in **??**.In addition, the calibration accuracy assessment in the end was done not with 1 point but 3: middle, left and right. Where left and right again correspond to the positions of the main objects on the screen. Furthermore, an in-between trial calibration was incorporated to ensure that the calibration was still accurate. The number of points were reduced to 5 each point corresponding to one of the areas of interest. No accuracy assessment was done during the in-between trial calibration. The setup can be seen in **??**. In order to make the first fixation after the in-between calibration more predictable, the top point, that corresponds to the sent message, was not available until all the other points were clicked on. This way participants would enter the trial looking at the sent message which is we expected to be the first point of interest. The sent message gives crucial information about the trial without which it is impossible to solve the task.

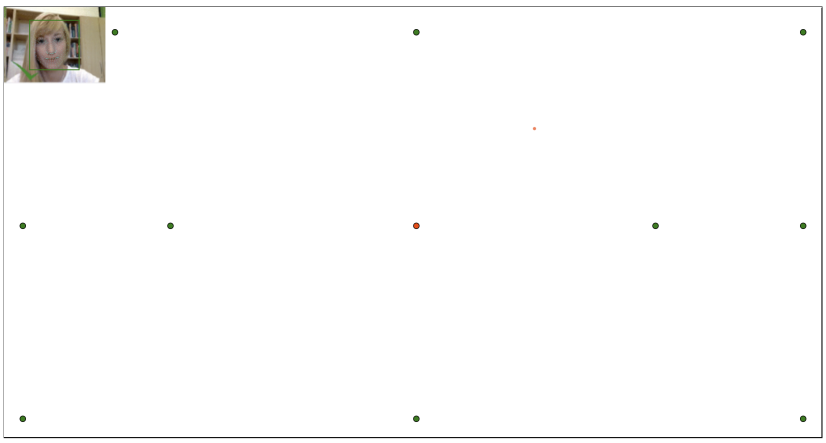


Figure 3.2: Calibration setup

In order to successfully pass the calibration assessment, the participants must reach at least 65% accuracy on each of the three calibration points. However, during the testing phase we noticed that the calibration assessment was too strict and difficult to pass. Therefore, it was decided to make the left and right points easier to calibrate via adjusting the calculation of the accuracy. A weighted accuracy calibration procedure was implemented. While the distance between for the accuracy of the middle point was calculated via euclidean distance, the left and right points were calculated via the following formula: $\sqrt{(w_x \cdot (calib\_point\_x - gaze\_x))^2 + (w_y \cdot (calib\_point\_y - gaze\_y))^2}$. Where w_x and w_y are the coefficients that were adjusted during the testing phase. The final values were $w_x = 1$ and $w_y = 0.5$. The values were chosen based on the fact that left and right objects do not have any other objects on the vertical axes this can be seen in **??**. Therefore a slightly inaccurate result on the vertical axes would be relatively easy to correct during the analysis.

The images were located on the screen as far as possible to reduce the errors as much as possible. For the same reason, the sent message were kept as a single block instead of being spread further apart.

Furthermore, a performance issue arised during the pilot phase. The issues was that the eye tracking became very slow and laggy towards the the second half of the experiment. The more clicks were made, the worse the performance became. Due to a drastic drop in sampling rate and increase in response time, the pilot data was clearly unacceptable. The issue was resolved by reducing the DataWindow size from 700 to 50 in source code and recreating the WebGazer source file afterwards. The issue was resolved and the performance was stable throughout the experiment.

The calibration is highly dependent on the setup. Hence, the following pieces of advice were given to the participants to increase chances of successful calibration: keep the laptop on charging during the whole experiment (to make sure the eye tracking does not suffer from battery saving features); choose a quiet, well-lit room with minimal distractions and use a stable chair; place your laptop on a stable surface, screen directly in front of you. Later during the calibration, ensure the webcam is centered with your face; keep your head sas still as possible during the experiment; make this window full screen size if not already. In addition, right before the calibration the participants were shown their webcam feed to make sure they are centered and the lighting is good as well as some visual advice on how to improve the calibration. The visuals can be seen

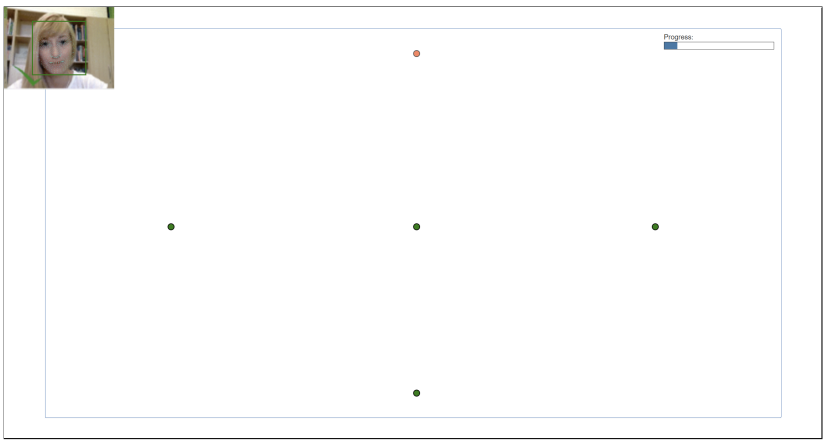


Figure 3.3: In-between trial calibration

Figure 3.4: Calibration instructions taken from **?**

in **??**. They were taken from a different experiment **?** that was conducted with use of WebGazer **?**.

Even with all the optimizations in place, the calibration procedure was still relatively difficult to pass. The participants were not restricted in number of attempts to calibrate. However, the participants were informed that the approval is only possible after successfully completing the calibration and the rest of the experiment.

3.1.3 Consent

The participants were told about the eye tracking right in the beginning of the experiment. They were informed that no video of them will be stored at any point of time. The participants were informed that they can stop the experiment at any point by exiting the website and none of their data will be saved. The exact formulation was:

> This experiment is being conducted as part of ongoing research at Saarland University. If you have any questions or comments about the study, please contact us. You must be at least 18 years old to participate. Your participation in this research is voluntary. There are no risks or benefits to participating in this study. You may decline to answer any or all of the following questions. You may decline further participation, at any time, without adverse consequences. Part of the data collections involves using your webcam to estimate your eye gaze. No video or audio is stored at any point during the experiment. We only use the video to store the estimated position of your eye gaze, as well as estimated size of your pupil. All data will be anonymized prior to analysis.

If you agree to participate, please read the below instructions before proceeding.

3.2 Analysis

3.2.1 Features

Considering the related eye tracking study **?**, the analysis will be around the defined eye tracking features. **?** defined mainly 5 types of features: absolute time on an area of interest, proportional time on an area of interest, toggling (between the Raven's matrix and the available messages), item latency (time to complete the trial) and latency to first toggle and matrix distribution index (how equally the attention was spread across the matrix items). In this study on the other hand, we are mainly focused on the proportional time on the areas of interest. The decision was made for multiple reasons. First of all the sampling rate of WebGazer is highly dependent on the participants' computer, making the absolute time on the areas of interest and latency to first fixation not comparable between participants. Second of all, we do not have such a strong hypothesis about what exactly people do during the task solving process as **?** had. We are rather interested in the general profile of attention, hence, no toggling features were included in the analysis.

In addition to the eye tracking features, we included some general features about the trial. The features are: trial number, condition (Simple, Complex, Unambiguous), type of sent message feature (shape or color), correct (whether the trial was solved correctly or not) and target position (left, center or right). The condition and correct features play crucial role in the analysis. While the other features such as trial number, type of sent message and target position were included due to the fact that they were shown to sometimes have a significant effect on the participants' performance in the previous studies **??**.

3.2.2 Data Preprocessing

The data preprocessing was done in Python using the pandas library **?**. Firstly, due to how LingoTurk **?** is implemented, we had to parse the data from string into the dictionary and create a data frame from it. As was mentioned before, the WebGazer **?** library was used to track the eye gaze. The WebGazer predicts the x and y coordinates of the current gaze point on the screen. Hence, we had to define a certain boundaries for the areas of interest in order to assign the predicted gaze points to them. Taking into account the weighted calibration **??** implemented for the left and right objects we had to make the margin a little larger for the left and right objects comparing to the center one. We settled on defining four point polygons for each of the interest areas.

The polygons were defined using the following variables:

- x_{12} – x coordinate equally distant between the left and the center images of the objects.
- x_{23} – x coordinate equally distant between the center and the right images of the objects.
- y_{12} – y coordinate equally distant between the sent message image and the center object image.

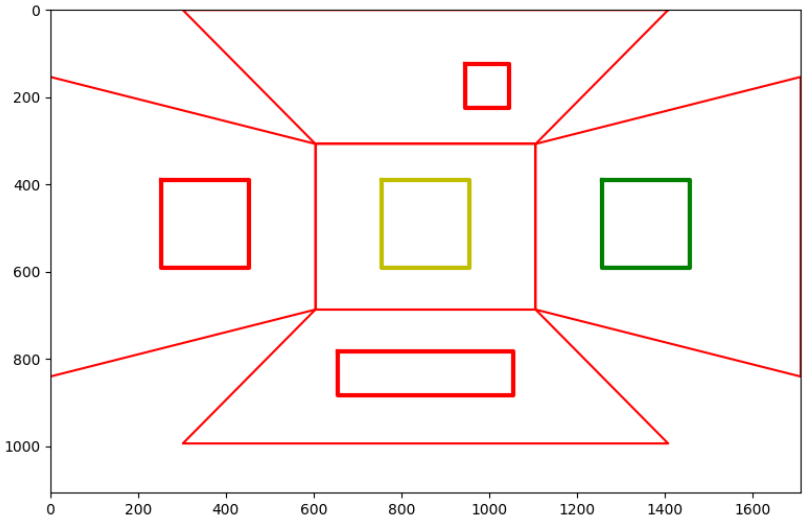


Figure 3.5: Polygons for the areas of interest. The inner rectangles show the exact positions of the images on the participant's screen. The outer rectangles show the polygons used to define the areas of interest.

- y_{23} – y coordinate equally distant between the center object image and the available messages images.

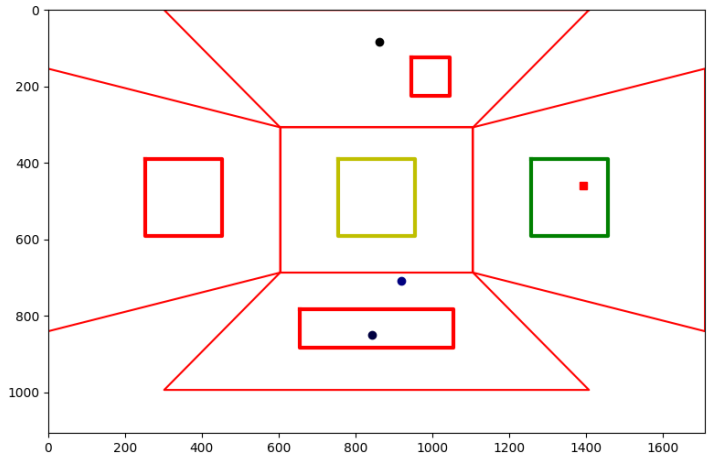
It is worth to note that the values of the variables can be easily computed using the coordinates of left top corner of an image and its' width and height. Finally, the polygons were defined via four points as follows:

- sent message – $((x_{23} + \frac{x_{12}}{2}, 0), (\frac{x_{12}}{2}, 0), (x_{12}, y_{12}), (x_{23}, y_{12}))$
- left object – $((0, \frac{y_{12}}{2}), (x_{12}, y_{12}), (x_{12}, y_{23}), (0, y_{23} + \frac{y_{12}}{2}))$
- center object – $((x_{12}, y_{12}), (x_{23}, y_{12}), (x_{23}, y_{23}), (x_{12}, y_{23}))$
- right object – $((x_{23}, y_{12}), (x_{12} + x_{23}, \frac{y_{12}}{2}), (x_{12} + x_{23}, y_{23} + \frac{y_{12}}{2}), (x_{23}, y_{23}))$
- available messages – $((x_{12}, y_{23}), (x_{23}, y_{23}), (x_{23} + \frac{x_{12}}{2}, y_{12} + y_{23}), (\frac{x_{12}}{2}, y_{12} + y_{23}))$

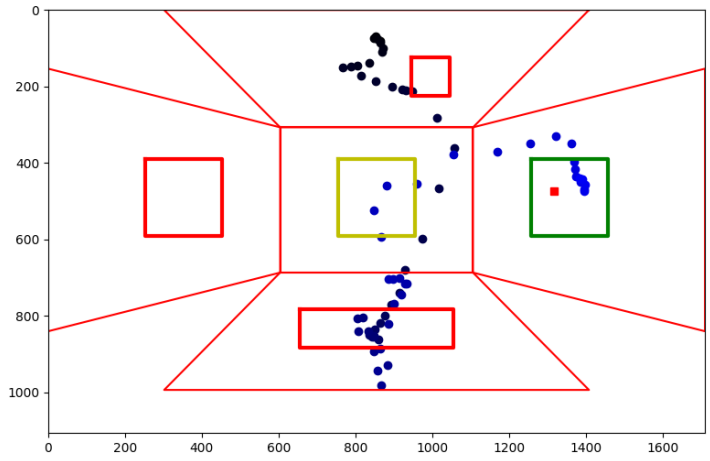
The visualization is shown in **??**.

3.2.3 Eye Tracking Features

Based on whether a point was inside a certain polygon or none of them it was assigned to the corresponding area of interest or to the non-area of interest category. Each point was not represented by the time between the current and the previous point as the rate of sampling was not the same across participants. In order to mitigate this issue a fixation detection algorithm was tried out. The algorithm is implemented in R **?**. The algorithm is based on the fact that due to a fixation present some points would be close to each other while the rest would be far away. The algorithm states to work well even for a low quality data with sampling rate less than 100 Hz. However, in our case an average sampling rate amounted to 17.65 Hz. Even with the lowest tolerance, the algorithm



(a) Fixations



(b) Scanpath

Figure 3.6: Comparison of the raw scanpath (b) and the detected fixations (a). The hue of the points corresponds to the order of the points. Black being the first point and the pure point being the last. In addition, the very last point is marked with a red square.

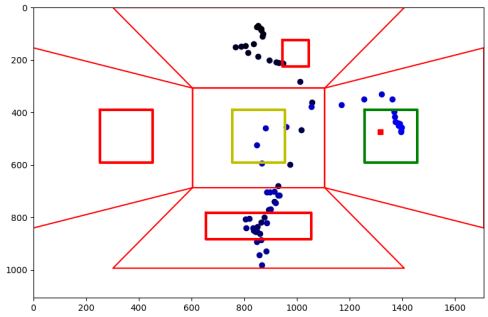


Figure 3.7: Scanpath of a simple trial, and the layout of areas of interest. The colors of the main objects are used as follows: green – target, yellow – competitor and red – distractor.

Feature	Value
TimeOnSentMsg	18
TimeOnAvailableMsgs	27
TimeOnTrgt	14
TimeOnDist	0
TimeOnComp	9
TimeOnNonAOI	0
PropTimeOnSentMsg	0.26
PropTimeOnAvailableMsgs	0.4
PropTimeOnTrgt	0.21
PropTimeOnDist	0.0
PropTimeOnComp	0.13
PropTimeOnNonAOI	0.0

Table 3.1: Features and their corresponding values. The features starting from 'time' corresponds to amount of points in the area of interest.

was not able to detect all fixations accurately due to the low sampling rate. An example of fixation detection and the original scanpath can be seen in **??**. Here, the algorithm correctly detected 4 fixations, they are: on the sent message, twice on the bank of available messages and the last one being on the right object. While the second fixations of the available messages is debatable, clearly the algorithm was not able to detect the quick glance on the middle object before the last fixation on the right object. Clearly, the sampling rate was too low in order to determine the fixations accurately. Hence, the algorithm was not used in the analysis.

Instead of using the algorithm, we decided to use the raw scanpath and derive the features from it. We assign each predicted gaze point to the corresponding area of interest. Then count the amount of them in each area of interest. This way we derive absolute time on area of interest. Again, it is worth mentioning that we did not use the actual timings of the predictions because the sampling rate varied a lot across the participants. Hence, we used the amount of points in each area of interest as a proxy for the time spent on it. Furthermore, we calculated the proportional time on area of interest as the amount of points in the area of interest divided by the total amount of points in the trial, including the ones that missed the areas of interest. The proportional time on area of interest is the main feature used in this thesis. An example of calculated features for the scanpath in **??** can be seen in **??**.

3.2.4 Pairwise Correlations

Similarly to **?**, we will report the pairwise correlations of eye tracking features and the accuracy based on condition. This should give us some general idea about what features are correlated with the accuracy. In addition, the correlation will test our hypothesis.

3.2.5 Mixed Effects Logistic Regressions

The main analysis will be done using mixed effects logistic regressions. The models are designed to correspond to our hypothesis presented in **??**. For the hypothesis 1 and 2, the following model was preregistered:

```
Correct ~ Condition + TrgtPos + Trial +
PropTimeOnTrgt + PropTimeOnComp + PropTimeOnDist +
PropTimeOnSentMsg + PropTimeOnAvailableMsgs +
PropTimeOnNonAOI + MsgType +
(1 + Condition + TrgtPos + Trial + PropTimeOnTrgt +
PropTimeOnComp + PropTimeOnDist + PropTimeOnSentMsg +
PropTimeOnAvailableMsgs + MsgType | Subject)
```

However, we quickly realized that the model has a few issues. First of all, the model is missing interaction between the different conditions and the eye tracking features. Therefore, the interaction terms were added to the model. The second issue is that the feature 'PropTimeOnNonAOI' can be derived from the other eye tracking features making it redundant and unnecessary to include in the model, hence, it was removed. In addition, we added the 'AnswerTime' feature to the model. The 'AnswerTime' feature is the time between the start of the trial and the moment when the participant clicked on the answer. It was added to account for the fact that some participants might have been faster than others. The final starting model looks as follows:

```
Correct ~ Condition + TrgtPos + Trial +
PropTimeOnTrgt + PropTimeOnComp + PropTimeOnDist +
PropTimeOnSentMsg + PropTimeOnAvailableMsgs + MsgType +
AnswerTime +
Condition:PropTimeOnTrgt + Condition:PropTimeOnComp +
Condition:PropTimeOnDist + Condition:PropTimeOnSentMsg +
Condition:PropTimeOnAvailableMsgs + Condition:AnswerTime +
(1 + Condition + TrgtPos + Trial + PropTimeOnTrgt +
PropTimeOnComp + PropTimeOnDist + PropTimeOnSentMsg +
PropTimeOnAvailableMsgs + MsgType | Subject)
```

Through this model we will be able to test the first two hypothesis. Mainly whether the distractor is positively associated with the accuracy on the complex trials. And, for the second hypothesis, whether the available messages become important on the simple trial comparing to unambiguous and maybe complex as well. The model will be fit on the trial level and predicts whether the answer was correct or not. The 'Trial' and 'AnswerTime' features was rescaled to be between -1 and 1, in order to match the scale of other features. All the eye tracking features were centered to have mean 0 but not scaled. This approach was chosen because it would make the interpretability of the models easier.

As for the third and fourth hypothesis, each of them will be tested using a separate model. Both of them predict whether a gaze point was on a certain area of interest. Hence, the data to fit them is made on the fixation level. Only the gaze points from the correctly solved trials were added to the data to match the hypothesis. The third hypothesis will be tested using the following model:

```
OnDist ~ Condition + Trial + MsgType + TrgtPos +
(1 + Condition + Trial + MsgType + TrgtPos | Subject)
```

It predicts whether a gaze point was on the distractor or not based on the general information about the trial.

The fourth hypothesis will be tested using the following model:

```
OnAvMsgs ~ Condition + Trial + MsgType + TrgtPos +
(1 + Condition + Trial + MsgType + TrgtPos | Subject)
```

It predicts whether a gaze point was on the bank of available messages or not based on the general information about the trial.

3.2.6 Exploratory Analysis

CNNs

Chapter 4
Results

4.1 General Information

In total there were 120 participants in the study who made a submission according to Prolific. 12 were removed due to technical issues with the submissions, either their submission were not received due to technical issues or they actually did not fully complete the experiment. Another 3 were removed due to having missing data in some of the trials and another 4 were removed due to having accuracy on unambiguous trials below 75% according to our preregistration. This left us with 101 participants for the analysis.

It is worth noting that due to the nature of the experiment, the calibration was difficult to pass, making the experiment difficult to complete. 120 participants completed the experiment, however, the were around 150 submission attempts that were not completed. This mainly happened because of the calibration difficulties according to some of the participants feedback. This information gives us an idea about the completion rate of the experiment, which amounted at around 30%. However, this allowed us to collect a large amount of good quality data, which is the most important aspect of the experiment.

4.2 Pairwise Correlations

4.3 Mixed Effects Logistic Regressions

Due to the complexity of the models, the models had issues converging. Therefore, the random effects were removed one by one from the models based on the least variance among the random effects. The process was repeated until the models converged. In the case of predicting accuracy, the random effects were removed from the model entirely.

4.3.1 Predicting Accuracy

The final formula for the model predicting accuracy was:

```
Correct ~ Condition + TrgtPos + Trial + PropTimeOnTrgt +
PropTimeOnComp + PropTimeOnDist + PropTimeOnSentMsg +
PropTimeOnAvailableMsgs + MsgType + AnswerTime +
Condition:PropTimeOnTrgt + Condition:PropTimeOnComp +
Condition:PropTimeOnDist + Condition:PropTimeOnSentMsg +
Condition:PropTimeOnAvailableMsgs + Condition:AnswerTime
```

The model had the following encodings for the categorical variables **??**, **??** and **??**. The target position can be interpreted as comparing left to the center and right to the center for features 'TrgtPos2' and 'TrgtPos3' respectively. The condition can be interpreted as comparing the simple condition to the complex condition and the unambiguous condition to the simple and complex conditions together. The model was trained using the `lme4` package in R. The model was trained using the `glm` function with the following parameters: `family = binomial(link = "logit")`. The resulting coefficients can be seen at **??**.

Message Type	Encoding
Shape	-1
Color	1

Table 4.1: Encoding of the message type categorical variable.

Target Position	TrgtPos2	TrgtPos3
0	1	0
1	0	0
2	0	1

Table 4.2: Encoding of the target position categorical variable.

Condition	Condition1	Condition2
Complex	-1	-1
Simple	1	-1
Unambiguous	0	2

Table 4.3: Encoding of the condition categorical variable.

The first hypothesis we wanted to test was that Proportional time on distractor is positively associated with accuracy on Complex trials **??**. Even though the coefficient for the interaction term 'Condition1:PropTimeOnDist' is not significant, due to how interaction terms work, we can still interpret how model prediction changes based on the value of the interaction term. First of all, we can take a look at tab:proptimeondist$_t rends which shows the trends of the Proportiona$

Coefficient	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	1.23265	0.18460	6.677	2.43e-11 ***
Condition1	0.22087	0.05499	4.017	5.91e-05 ***
Condition2	1.10938	0.14302	7.757	8.69e-15 ***
TrgtPos2	1.92305	0.21183	9.078	< 2e-16 ***
TrgtPos3	1.69844	0.20450	8.305	< 2e-16 ***
Trial	0.22635	0.04825	4.691	2.72e-06 ***
PropTimeOnTrgt	-6.09992	6.15675	-0.991	0.3218
PropTimeOnComp	-12.92581	6.16769	-2.096	0.0361 *
PropTimeOnDist	-11.95582	6.14733	-1.945	0.0518 .
PropTimeOnSentMsg	-9.95567	6.06806	-1.641	0.1009
PropTimeOnAvailableMsgs	-8.69726	6.27784	-1.385	0.1659
MsgType1	-0.11506	0.04888	-2.354	0.0186 *
AnswerTime	-0.12537	0.05186	-2.418	0.0156 *
Condition1:PropTimeOnTrgt	-1.93730	1.09696	-1.766	0.0774 .
Condition2:PropTimeOnTrgt	-12.13074	6.12982	-1.979	0.0478 *
Condition1:PropTimeOnComp	-1.30711	1.08177	-1.208	0.2269
Condition2:PropTimeOnComp	-11.30869	6.12510	-1.846	0.0649 .
Condition1:PropTimeOnDist	-1.13168	1.08619	-1.042	0.2975
Condition2:PropTimeOnDist	-12.16270	6.10435	-1.992	0.0463 *
Condition1:PropTimeOnSentMsg	-0.97657	1.09161	-0.895	0.3710
Condition2:PropTimeOnSentMsg	-10.26290	6.03306	-1.701	0.0889 .
Condition1:PropTimeOnAvailableMsgs	0.70963	1.13710	0.624	0.5326
Condition2:PropTimeOnAvailableMsgs	-12.08512	6.24131	-1.936	0.0528 .
Condition1:AnswerTime	-0.11566	0.05769	-2.005	0.0450 *
Condition2:AnswerTime	-0.02166	0.03973	-0.545	0.5856

Table 4.4: Summary of the trained model coefficients. Significance codes: 0 '***', 0.001 '**', 0.01 '*', 0.05 '.', 0.1 ' ', 1.

Condition	PropTimeOnDist.trend	SE	asymp.LCL	asymp.UCL
Complex	0.227	0.248	-0.260	0.714
Simple	-0.132	0.233	-0.588	0.324
Unambiguous	-0.496	0.189	-0.866	-0.125

Table 4.5: Summary of PropTimeOnDist trends by condition.

Chapter 5
Conclusion and General Discussion

5.1 General Discussion

The way we assigned the fixations to the stimuli is not optimal as we clearly missed the timings of the gaze predictions as well as added some of the gaze points that are from saccades rather than from fixations. Both of the problems could have been solved with a fixation detection algorithm. A more advanced fixation detection algorithm could be implemented in the future to improve the preprocessing part of the data analysis. Currently as was discussed in **??** the fixation detection algorithm falls short in the case of low sampling rates that WebGazer demonstrates. The algorithm is not able to detect some of the fixations in the data, which would lead to a loss of information. Therefore, clearly the approach chosen by us in the end is not optimal and could be improved in the future.

References

Bethell-Fox, C. E., Lohman, D. F., & Snow, R. E. (1984). Adaptive reasoning: Componential and eye movement analysis of geometric analogy performance. *Intelligence*, *8*(3), 205-238. Retrieved from https://www.sciencedirect.com/science/article/pii/0160289684900096 doi: https://doi.org/10.1016/0160-2896(84)90009-6

Carpenter, P. A., Just, M. A., & Shell, P. (1990). What one intelligence test measures: A theoretical account of the processing in the Raven progressive matrices test. *Psychological Review*, *97*, 404-431. doi: 10.1037/0033-295x.97.3.404

Frank, M. C., Emilsson, A. G., Peloquin, B., Goodman, N. D., & Potts, C. (2016). *Rational speech act models of pragmatic reasoning in reference games.* PsyArXiv. Retrieved from osf.io/preprints/psyarxiv/f9y6b doi: 10.31234/osf.io/f9y6b

Frank, M. C., & Goodman, N. D. (2012). Predicting pragmatic reasoning in language games. *Science*, *336*(6084), 998-998. Retrieved from https://www.science.org/doi/abs/10.1126/science.1218633 doi: 10.1126/science.1218633

Franke, M., & Degen, J. (2016). Reasoning in reference games: Individual- vs. population-level probabilistic modeling. *PLOS ONE*, *11*(5), 1-25. Retrieved from https://doi.org/10.1371/journal.pone.0154854 doi: 10.1371/journal.pone.0154854

Grice, H. P. (1975). Logic and conversation. *Syntax and semantics*, *3*, 43–58.

Lewis, D. K. (1969). *Convention: A philosophical study*. Cambridge, MA, USA: Wiley-Blackwell.

Mayn, A., & Demberg, V. (2023, 06). High performance on a pragmatic task may not be the result of successful reasoning: On the importance of eliciting participants' reasoning strategies. *Open Mind*, *7*, 156-178. Retrieved from https://doi.org/10.1162/opmi_a_00077 doi: 10.1162/opmi_a_00077

Mayn, A., Loy, J. E., & Demberg, V. (2025, 01). Beliefs about the speaker's reasoning ability influence pragmatic interpretation: Children and adults as speakers. *Open Mind*, *9*, 89-120. Retrieved from https://doi.org/10.1162/opmi_a_00180 doi: 10.1162/opmi_a_00180

Palan, S., & Schitter, C. (2018). Prolific.ac—a subject pool for online experiments. *Journal of Behavioral and Experimental Finance*, *17*, 22-27. Retrieved from https://www.sciencedirect.com/science/article/pii/S2214635017300989 doi: https://doi.org/10.1016/j.jbef.2017.12.004

Papoutsaki, A., Sangkloy, P., Laskey, J., Daskalova, N., Huang, J., & Hays, J. (2016). Webgazer: Scalable webcam eye tracking using user interactions. In *Proceedings of the 25th international joint conference on artificial intelligence (ijcai)* (pp. 3839–3845).

Pusse, F., Sayeed, A., & Demberg, V. (2016). Lingoturk: managing crowdsourced tasks for psycholinguistics. In *Proceedings of the 2016 conference of the north american chapter of the association for computational linguistics: Demonstrations* (pp. 57–61).

Reback, J., McKinney, W., Van Den Bossche, J., Augspurger, T., Cloud, P., Klein, A., ... others (2020). pandas-dev/pandas: Pandas 1.0. 5. *Zenodo*.

Semmelmann, K., & Weigelt, S. (2018). Online webcam-based eye tracking in cognitive science: A first look. *Behavior Research Methods*, *50*(2), 451–465. Retrieved from https://doi.org/10.3758/s13428-017-0913-7 doi: 10.3758/s13428-017-0913-7

Vigneau, F., Caissie, A. F., & Bors, D. A. (2006). Eye-movement analysis demonstrates strategic influences on intelligence. *Intelligence*, *34*(3), 261-272. Retrieved from https://www.sciencedirect.com/science/article/pii/S0160289605001248 doi: https://doi.org/10.1016/j.intell.2005.11.003

von der Malsburg, T. (2015, oct). *Saccades: An r package for detecting fixations in raw eye tracking data.* Zenodo. Retrieved from https://doi.org/10.5281/zenodo.31799 doi: 10.5281/zenodo.31799

Wisiecka, K., Krejtz, K., Krejtz, I., Sromek, D., Cellary, A., Lewandowska, B., & Duchowski, A. (2022). Comparison of webcam and remote eye tracking. In *2022 symposium on eye tracking research and applications.* New York, NY, USA: Association for Computing Machinery. Retrieved from https://doi.org/10.1145/3517031.3529615 doi: 10.1145/3517031.3529615