Czech Technical University

Faculty of Nuclear Sciences and Physical Engineering

Department of Software Engineering
Field: Engineering Informatics

Navigation in virtual environment
Navigování ve virtuálním světě

MASTER THESIS

Author:	Václav Honzík
Supervisor:	Doc. Ing. Zdeněk Žabokrtský, Ph.D.
Year:	2013

Insert assignment instead of this page before handing in the thesis!!!!

Declaration

I declare, that I have developed my master thesis independently and used only the materials (literature, projects, SW etc.) listed in attached list.

In Prague on ....

....
Václav Honzík

Acknowledgment

I would like to thank Doc. Ing. Zdeněk Žabokrtský, Ph.D.for supervising and shaping my master thesis. I would like to also thank Prof. Kristina Striegnitz, Ph.D.for giving me the opportunity and helping me with my master thesis.

Václav Honzík

Title:
Navigation in virtual environment

Author: Václav Honzík

Field: Engineering Informatics
Type of thesis: Master thesis

Supervisor: Doc. Ing. Zdeněk Žabokrtský, Ph.D.
Institute of Formal and Applied Linguistic, Faculty of Mathematics and Physics, Charles University in Prague
Advisor: Prof. Kristina Striegnitz, Ph.D.
Union College, Schenectady USA

Abstract: abstract en

Key words: keywords en

Název práce:
Navigování ve virtuálním světě

Autor: Václav Honzík

Obor: Inženýrská Informatika
Typ práce: Diplomová práce

Vedoucí: Doc. Ing. Zdeněk Žabokrtský, Ph.D.
Institute of Formal and Applied Linguistic, Faculty of Mathematics and Physics, Charles University in Prague
Konzultant: Prof. Kristina Striegnitz, Ph.D.
Union College, Schenectady USA

Abstrakt: abstract cz

Klíčová slova: keywords cz

Contents

Introduction

Chapter 1

Background

INTRODUCE REF EXPRE AS RE

1.1 Related work

Ha et al. (2012) talk about an 'information gaps' caused by existence of a non-dialogue communication stream. They concluded that the posture of user, an example of implicit information from the non-dialogue streams, is a significant attribute in modeling of dialog acts. Their goal is to overcome this 'information gaps' through machine learning techniques. A shared view of the virtual world in this thesis is also a form of non-dialogue stream, with which must the navigation system work. I also try to apply machine learning to help with language generation.

Viethen et al. (2011a) compare traditional algorithmic approaches with alignment approaches based on psycho-linguistic models for the REG. They use large data-set (16,358 referring expressions) of direction giving task on a shared 2D visual scene introduced by Louwerse et al. (2007). They use three feature sets: traditional REG, alignment and independent (general information about the scene) to build decision tree models (concretely C4.5) combining these feature sets. The alignment based models outperform the traditional REG ones and the best model combines all feature sets to achieve accuracy 58.8% and DICE score 0.81. Not using traditional algorithmic REG features did not result in a significant decrease of accuracy, suggesting that the visual context doesn't play such an important role as it was believed in the REG research so far. Viethen et al. (2011b) verified this surprising conclusion by varying the visual context. They argue that the relative simplicity of visual scenes used in contemporary research might be the cause of insignificance of the visual context. I would argue that 3D virtual world explored in this thesis is more complex then theirs and therefore this paper can provide some insight into these questions.

Stoia et al. (2006a) were interested in timing of the first reference to the target in 3D virtual world. They predicted whether direction giver refers to the target or delay the reference based on the spatial data. Their attributes were angle and distance to the target, number of visible distractors (either same category as target or all of them) and whether the target is visible. The most important feature in decision tree model was number of visible distractors followed by angle and distance. They achieved 86% accuracy, compared 70% baseline. The baseline was to refer when the target is visible and to delay the reference when the target isn't visible. Part of the machine learning attempts of this thesis is to replicate first reference timing of Stoia et al. (2006a) on GIVE dataset.

Stoia et al. (2006b) developed decision trees to generate a noun phrase, specified by three slots: determiner/quantifier, pre-modifier/post-modifier and head noun. They used a data-set from 3D virtual world navigation task similar to GIVE dataset. Four categories of features were used: dialog history, spatial and visual features, relation to other objects in the world and object category. The decision trees revealed significant dependencies between the slots and importance of the spatial features. Interestingly, they used three types of system's evaluation. The exact match evaluation produced 31.2% accuracy compared to 20% most-frequent baseline. Comparison with hand-crafted Centering algorithm (Kibble and Power, 2000) ended with similar accuracy, favoring the machine-learning approach for requiring less structural analysis of the input text. Lastly, when human judged the system output, it was at least equal or preferred to original spontaneous language in 62.6% (inter-annotator reliability $\kappa = 0.51$).

Gallo et al. (2008) showed that the Fruit Carts corpus can be used in NLG by case study on message complexity and structural realizations. A logistic regression confirmed that the complexity of verb arguments affects production choice between mono-clausal or bi-clausal structure. In more general terms, the complexity of the virtual environment affects how people speak on all linguistic levels. Referring expression generation should take that into consideration.

Clark and Krych (2004) were examining speakers' monitoring of addressees in a Lego-building experiment. One participant - director - knew 10 Lego models and how to build them. The director was verbally instructing second participant - builder - to build these models. In one group the director could see the builders workspace, in second group he could not and in a third the instructions were audio-taped and simply passed onto the builder. Builders communicated with the directors on the workspace through head gestures and manipulating blocks (placing, exhibiting or poising and so on). When the workspace was blocked of, the task took much longer. In the audio-taped group the builders made many more errors. Directors often altered their utterances midcourse based on builders actions.

Koller et al. (2012) tracked hearer gaze using camera and used that information to produce feedback to correct or confirm previous referring expressions. Experiment took place in a 3D virtual world. This enhancement was compared with feedback based on virtual agent's position and system with no feedback at all. Eye-tracking enhancement significantly improved hearers understanding of the REs. Eye-tracking is therefore useful tool to improve interaction quality. This experiment also shows importance of feedback, since the system with no feedback performed worse than the two systems with feedback.

Chapter 2

GIVE Challenge

Important framework for this thesis is the GIVE Challenge (Koller et al., 2010a). The data I used to develop the hypothesis were collected using the GIVE framework (Koller et al., 2010a). I used GIVE framework to implement and test my hypothesis. Therefore, in this chapter I will describe this academic competition in detail.

The first section will answers basic questions such as what is the GIVE Challenge, why was it created and what are its interesting properties. In the next section, I will provide a brief history of the GIVE Challenge together with some of its results. In the third section, the focus will be a detailed description of the shared task and the virtual world of the GIVE Challenge.

2.1 Introduction

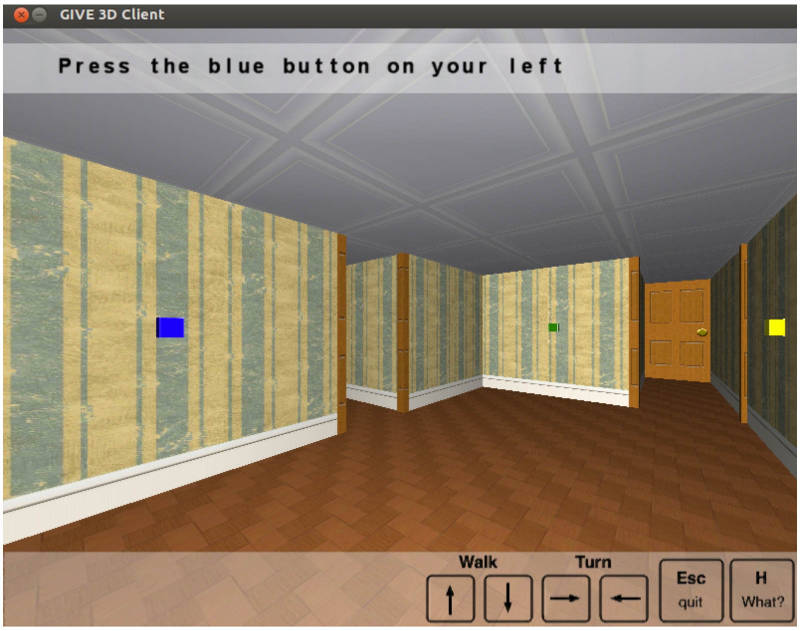
The GIVE Challenge was a series of Natural Language Generation (NLG) competitions run from November 2008 to March 2012. Participants developed NLG systems to navigate human-controlled avatars in a 3D virtual environment. The real-time navigation was realized through written instructions displayed on the screen. Goal of the navigation was to finish a treasure-hunt game. In Figure 2.1 we can see the GIVE client with virtual world and example of an instruction. A more detailed description of the task and the environment is in the Section 2.3.

Koller et al. (2010a) state that one of the goals of the GIVE Challenge was spawning interest in NLG, a subfield of computational linguistics (CL), and was inspired by other competitions in the field such as the Recognizing Textual Entailment challenge[1] and NIST machine translation competition[2].

According to Koller et al. (2010a), another important goal was to introduce and explore a new way of evaluating NLG algorithms, techniques and systems in a shared task. More specifically a shared task which was, on the one hand, complex enough to encompass multiple NLG subtasks and, on the other hand, was only concerned

[1] http://pascallin.ecs.soton.ac.uk/Challenges/

[2] http://www.itl.nist.gov/iad/mig//tests/mt/



Press the blue button on your left

Figure 2.1: A human subject is being navigated through the environment.

with NLG and not any other fields of computational linguistic.

Three basic approaches to evaluation of NLG systems are compared annotated corpora, measuring task performance in an experiment and human judges evaluation. Koller et al. (2010a) argue the advantages and disadvantages of these evaluation in more depth, therefore I will only provide a brief summary. The first approach compares output of the NLG system to an annotated corpora, also known as a gold-standard. It is fast and cheap approach, but a problem with it lies in the complexity of the natural language. We can often express concepts in many different ways and there is often no telling which way is a better one. The second approach conducts an experiment and measures task performance on human subjects. Measuring task performance avoids the problems of gold-standard, but it is expensive and time consuming. Lastly, trained human judges are used to evaluate the system. It is less demanding than the second approach, but for the cost of certainty, that the results correspond with results one might achieve with non-expert subjects.

The GIVE Challenge proposes and successfully implements a new, in a sense that it wasn't used for NLG before, approach through Internet-based evaluation. The basic premise is using a client-server software methodology. The client is a program installed on test subject computer, which is easily downloadable from a public website. The client connects through the Internet to a matchmaker server and random evaluation world is selected. Matchmaker also connects the client to a randomly selected NLG system, which itself, can be hosted on a different server. Client and NLG systems then communicate back and forth until the task is finished. Matchmaker finally logs the entire sessions to database. Figure 2.2 shows that architecture in a simple diagram.

This approach immediately presents several advantages. It does not require physical presence of the test subject in a laboratory. The subject simply downloads the client

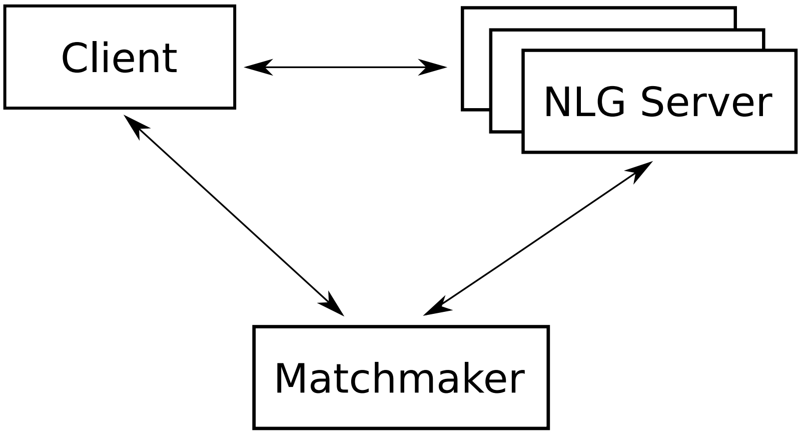


Figure 2.2: Software architecture of GIVE Challenge.

from a website and is able to do the experiment at his/her convenience. Second obvious advantage is a scalability. The number of individuals which can parallelly undertake the experiment is only limited by the servers' load. Thanks to the low costs, advertisement becomes the decisive limitation on the number of subjects. Take for example second instalment of the GIVE Challenge which had up to 1800 participants.

On the other hand, part of the control over the experiment is lost in this approach; for example the control over subject pool. Another problem which rises with this approach is that individuals can repeat the experiment.

In addition to Internet-based evaluation, the GIVE Challenge utilizes variety of evaluation measures of both objective and subjective nature. Among objective measure are task success rate, number of instructions or time required to finish the task. For subjective measures a questionnaire was used at the end of the session. The questionnaire mostly used a 5 point scale with question such as how clear where the instruction or how friendly was the system. Some of the measures intentionally collided with each other, putting emphasize on a certain characteristic of the system.

Having presented the basic concept of the GIVE Challenge and reasons for its creation, I will now move onto brief history of this competition.

2.2 History of GIVE Challenge

The first instalment of GIVE Challenge (GIVE-1) was publicized in March 2008. Koller et al. (2010a) report on this instalment and are the source of following information. For more details please refer to their paper. The data collection period was from November 2008 to February 2009. Four teams participated in this challenge, namely from these universities: University of Texas at Austin, Universidad Complutense de Madrid, University of Twente and Union College. The team from University of Twente submitted two systems, making the final number of systems five.

What is important to note about GIVE-1 is a different world representation from the following instalments. GIVE-1 used discrete square grid for player movement. Player was able to rotate only by 90° and walk forward and backwards by one square of the grid. That had a major impact on the design of NLG systems. Participating teams at least occasionally used this grid in their references (eg. *move forward three steps*). Afterwards organizers realized that the grid and the discrete movement made the task easier than intended and they were after GIVE-1 removed.

Altogether, 1143 valid games were recorded. The demographics featured a majority of males (over 80%) and wide spread over different countries in the world. For the actual results, the system from Austin significantly outperformed all other systems in task completion time. At the same time systems from Union and Madrid outperformed other systems in success rate. That shows the significance of different measures for the evaluation. Similar interesting conclusion in both objective and subjective measures can be found in previously mentioned paper. Apart from objective and subjective measures, the report examined influence of English language proficiency and differences between evaluation worlds. The English proficiency had an impact on the task success rate but solely for the least proficient category. The evaluation world also had a significant influence on the task success rate.

Finally, the first instalment also compared the Internet-based evaluation with more standard laboratory evaluation. The conclusion was that Internet-based evaluation provides meaningful results comparable and even more precise in some areas to the laboratory setting.

The second instalment (GIVE-2) run from August 2009 (data collection starting in February 2010) to May 2010 and is thoroughly described by Koller et al. (2010b). Following information are based on this paper. Biggest difference to the GIVE-1, which was mentioned previously, is that players were now able to move freely. This made the instruction generation considerably harder. Additionally, the questionnaire was revised and a few new objectives measures were introduced. Evaluation worlds used in GIVE-2 were considerably harder than in GIVE-1. Number of distracting buttons was increased and same-colored buttons were in some cases next to each other. Also number of alarm tiles was increased. Otherwise, the architecture and the rest of the details stayed the same as in GIVE-1.

This time 1825 games were played over seven NLG systems developed by six teams from: Dublin Institute of Technology, Trinity College Dublin, Universidad Complutense de Madrid, University of Heidelberg, Saarland University and INRIA Grand-Est in Nancy (2 systems).

There was a big drop in success rate, most likely linked to the free movement and the increase of difficulty in the evaluation worlds. Similarly to results in GIVE-1, there was an influence of English proficiency and game world on the task success rate. Additionally, age of the subject played a role in the time required to finish the task and number of actions to finish the task (younger subjects being faster and requiring less actions). The difference between genders in time required to finish the task disappeared in GIVE-2.

Some teams participating in the GIVE Challenge tried to use a corpora of a human to human interactions in GIVE scenario. They were learning language expression or decision-making process and applying them in their NLG systems. The teams were however relying on small self-collected datasets. In a light of this, organizers of GIVE Challenge decided they would collect and provide dataset for future use. Gargett et al. (2010) describe this dataset, which was used in the next instalment of the GIVE challenge.

Following GIVE-2 was so called Second Second instalment (GIVE-2.5), which kept almost the same settings as GIVE-2. There was just a small addition to objective measures and a reduction in the number of subjective questions. The data collection took place between July 2011 and March 2012. Striegnitz et al. (2011) report on the partial results of 536 valid games from July and August 2011, which however constitute a majority of the final number of 650 valid games.

Eight NLG systems participated from 7 teams: University of Aberdeen, University of Bremen, Universidad Nacional de Córdoba, Universidad Nacional de Córdoba and LORIA/CNRS, LORIA/CNRS, University of Potsdam (2 systems) and University of Twente. In this instalment the teams employed more broad spectrum of approaches. Team from University of Bremen used decision trees learned from GIVE-2 corpus. Universidad Nacional de Córdoba and LORIA/CNRS, LORIA/CNRS selected instructions from a corpus of human to human interactions. The teams also often included algorithms from existing NLG and CL literature.

Apart from comparing the systems through objective and subjective measures, Striegnitz et al. (2011) again examined effects of evaluation worlds and demographics factors on task success rate. The evaluation worlds and the English proficiency had an effect. Additionally computer expertise and familiarity with computer games significantly influenced the task performance. The difference between male and female subject wasn't significant.

The following section describes the shared task in more detail and lists possible contents of the GIVE virtual worlds.

2.3 Task and GIVE world

The GIVE world is a 3D virtual world. The world is an indoor environment, comprising of rooms connected by doors. It's defined in a human-readable format and stored in a text file. The following objects can be places in a world:

- Alarm tile
- Button
- Door
- Landmark
  - Bed

  - Chair
  - Couch
  - Dresser
  - Flower
  - Lamp
  - Table
  - Window
- Picture
- Safe
- Trophy
- Wall

In addition, some of these objects can have attributes, states or can operate other objects. Buttons have colors as an example of attribute. Doors and safes can be in a closed or an open state. Buttons can operate doors, safes or pictures.

Walls are actually created automatically by defining shapes of rooms. Rooms can have rectangular shape or can be defined by a polygon. I will sometimes use a term "corridor" which is a connecting room, usually not containing any button.

The landmarks serve as a decoration but they can be used in an expression generation. Picture is technically a landmark as well, but in GIVE Challenge it often serves another purpose. It covers the safe and needs to be put aside by a button press.

Figure 2.3 shows an evaluation world number one from GIVE-2.5. In top-left room we can see player starting position. Buttons are colored squares on the walls. Grey bars on the walls are closed doors. Trophy in a safe is in the middle-left room. There are also landmarks (like lamp or chair) and one big read square marking an alarm tile.

The flexibility of GIVE world creation allows relatively broad range of scenarios for the task. On the other hand, all the GIVE Challenge instalments consisted of similar sequence of steps.

The goal of all the GIVE Challenge worlds is to pick up a trophy. The trophy is hidden in a closed safe. In order to open the safe a sequence of buttons, usually counting somewhere around 6 buttons, has to be pressed. The safe can be also hidden by a picture, which needs to be put aside. The buttons in a safe-opening series are often in different rooms. Rooms can be also closed off, requiring another button press to open the door. While moving around the world, player has to avoid alarm tiles. Stepping on an activated alarm tile causes an immediate loss. Alarm tiles can also block the path and need to deactivated by a button press. Some buttons also cause an alarm and an immediate loss.

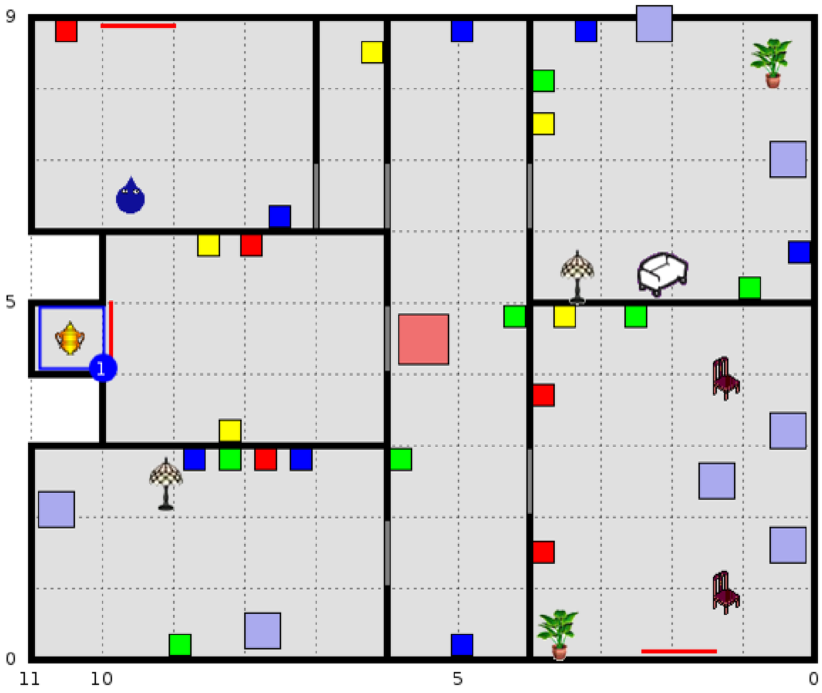


Figure 2.3: Example GIVE world viewed in GIVE map viewer utility.

Depending on the number of rooms, complexity of buttons arrangement and length of safe-opening sequence, the task can range from short and trivial problem easily handled by a few instruction templates, to a long and hard case, where it's impossible to capture every possible scenario.

To summarize, navigation system of a player in the GIVE world has to deal with following steps. Note that their order depends on the world definition and they can be thought of as layers of behaviours the system must enforce on player.

- Avoid alarm tiles
- Avoid pressing alarm-causing buttons
- Deactivate path-blocking alarm tiles by a button press
- Open closed-off rooms by pressing correct buttons
- Press a sequence of buttons to open the safe
- Reveal safe behind a picture by pressing correct button
- Take the trophy

After the safe was opened and possibly revealed from behind the picture player can pick up the trophy and therefore win the game.

Chapter 3

The S-GIVE Dataset

After GIVE 2.5 instalment presented in Chapter 2, the organizers of the GIVE Challenge became interested in spoken communication and therefore decided to collect a new dataset, called the S-GIVE Dataset. This chapter serves as an introduction and analysis of this dataset.

As a side note, Striegnitz et al. (2012) report on a smaller German dataset, which is similar to the one I will be talking about.

In the first section, I will introduce the dataset and provide technical details of how it was created. Section 3.2, will, after the fashion of GIVE Challenge look how world and demographic factors influenced the task performance. Next section analyses REs in the dataset. Last section explores a phenomenon of chains of references.

3.1 General overview

The S-GIVE dataset is different from previous GIVE Challenge experiments because the IG's instructions were of a spoken form. That changes many aspects of the discourse. For one thing, the IG knows when the IF received his instruction, which is not true for the written instructions. That promotes faster feedback and allows interrupting during an instruction. On the other hand, in some cases, the spoken word does not follow grammatical rules. Moreover, interjections are very common, as are incomplete sentences. That makes S-GIVE dataset complicated yet interesting to explore.

The data-collection started in July and finished November of 2012. Through that period 21 interactions between two human subjects were recorded. Originally, 22 pairs participated, but one of the pair failed to finish the tasks and is excluded from the dataset. The subjects were asked to bring someone they know and they were financially compensated for the effort.

The set-up for the experiment is in Figure 3.1. One human subject was an instruction giver (IG). He is on the right in Figure 3.1. His role was essentially the role of NLG system in GIVE Challenge. He/she was able to see a map of the world, which was updated in real-time and he/she got information about all necessary steps to finish the task. In addition, he was able to see the other person's client screen. He communicated with the other person through a microphone and his goal was to navigate the other person through the world and make him finish the treasure-hunt.

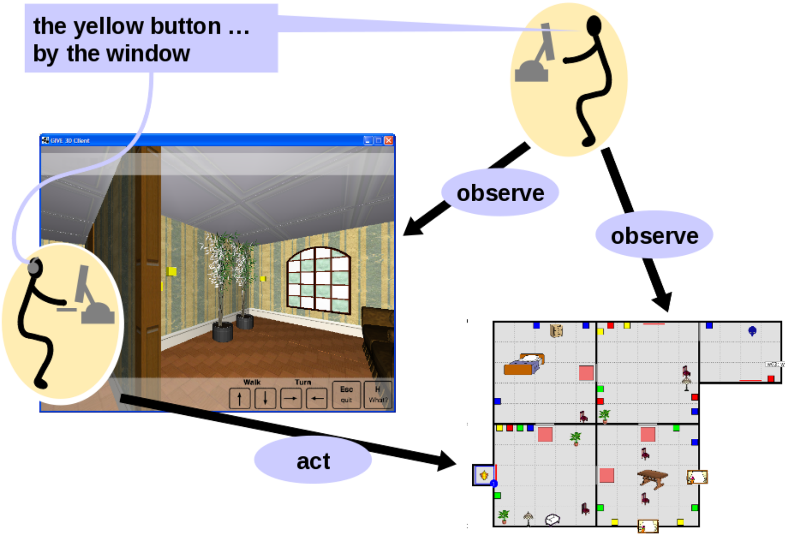


Figure 3.1: Experiment set-up of data collection

The other person was an instruction follower (IF). He/she is on the left in Figure 3.1. He/she interacted with the client and moved the avatar around the world and was able to press buttons. IF listened to IG's instructions through a headset.

Each pair did one short tutorial world. After that they switched roles of instruction giver and instruction follower. Following was one "normal" world randomly chosen from two variants, marked world 1 and world 3 in the dataset. Maps of the worlds 1 and 3 are in Figures 3.2 and 3.3 respectively. Finally they did a difficult version of the other variant (if they started with world 1 the difficult version was for world 3 and vice versa). The difficult versions are marked 1-d and 3-d in the dataset. A difficult version of the world had an increased number of distracting buttons and landmarks compared to the "normal" version, as can be seen in the map of world 1-d in figure 3.4. If not present in the report or not stated otherwise, the short tutorial worlds are normally excluded from the analysis.

Similarly to the GIVE Challenge, after all 3 rounds subjects were asked to fill in a questionnaire. Its purpose was to get demographic and other relevant information on subjects. The questionnaire can be divided into three parts. First part was only filled out by IF and rated IG and his instruction giving on a scale from 1 to 7. Complete list of its questions follows:

1. Overall, my partner gave me good instructions.
2. Interacting with my partner wasn't annoying at all.
3. My partner's instructions were clearly worded.

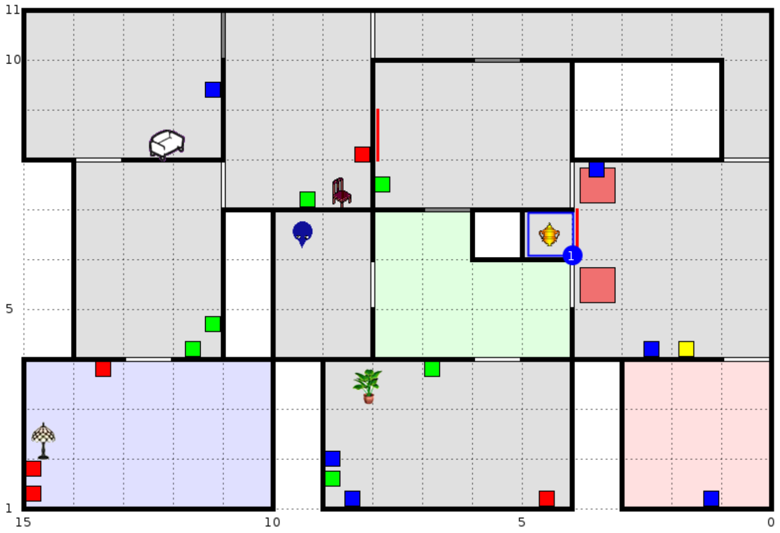


Figure 3.2: Map of the world 1 - normal version.

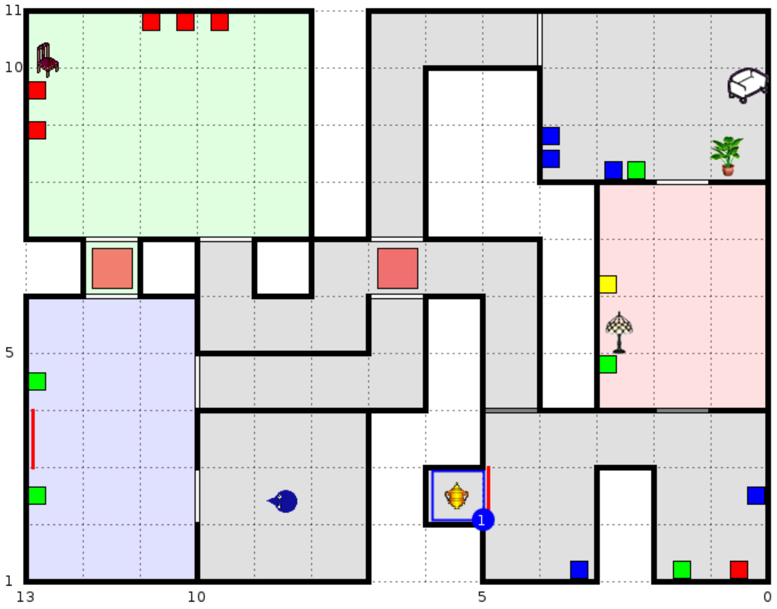


Figure 3.3: Map of the world 3 - normal version.

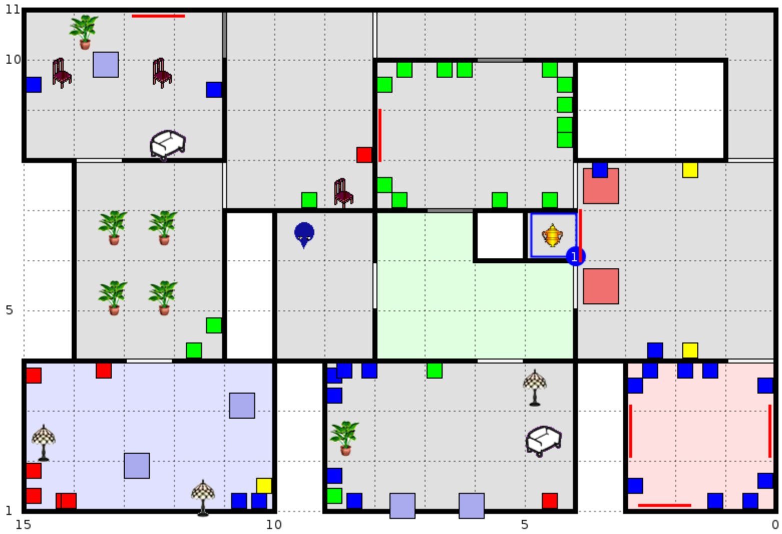


Figure 3.4: Map of the world 1 - difficult version.

4. When I had problems with the instructions, we solved them quickly.
5. I enjoyed solving the task.
6. I felt I could trust my partner's instructions.
7. I really wanted to find that trophy.
8. My partner immediately offered help when I was in trouble.
9. I would recommend this experiment to a friend.
10. My partner's instructions were not repetitive.

Second part was filled by both IF and IG and was concerned with their navigation skills. To measure the ability to navigate in the world the Santa Barbara Sense of Direction (SBSOD) scores were used (Hegarty et al., 2002). Again the questions were on a scale from 1 to 7. Note that some of the question in the following list are of positive nature (higher rating equals better navigation skill) while other are of negative nature, therefore these scores had to be normalized.

1. I am very good at giving directions.
2. I have a poor memory for where I left things.
3. I am very good at judging distances.
4. My "sense of direction" is very good. I tend to think of my environment in terms of cardinal directions (N, S, E, W).
5. I very easily get lost in a new city.
6. I enjoy reading maps.

7. I have trouble understanding directions.
8. I am very good at reading maps.
9. I don't remember routes very well while riding as a passenger in a car.
10. I don't enjoy giving directions.
11. It's not important to me to know where I am.
12. I usually let someone else do the navigational planning for long trips.
13. I can usually remember a new route after I have travelled it only once.
14. I don't have a very good "mental map" of my environment.

Finally, the last part of the questionnaire was of a demographic character. We can see questions about age, gender, language and computer expertise, 3D games experience and knowledge of the partner in following list.

1. What is your age?
2. Are you male of female?
3. What is your profession / major / favorite subject in school?
4. How would you rate your computer expertise?
5. How familiar are you with 3D computer games?
6. How many hours per week to you play 3D computer games?
7. Was there a time in your life when you played more 3D computer games? If so, how many hours did you play then?
8. What languages do you speak? Please indicate how well you speak each on a scale from 1-5, where 5 is your native language.
9. Did you already know the second participant?
10. How well do you know the second participant?
11. Have you worked collaboratively with the second participant before? (For example, when doing homework or preparing a class presentation?)
12. Have you played 3D computer games with the second participant before?

Many of these questions are explored in the next section as a potential factors influencing the task performance.

As was mentioned in Chapter 2, the entire session is logged to a database. The player's position, orientation and all visible objects are logged at a fixed rate. Moreover, other information as buttons presses or an end of the session are also stored in the logs. Because the worlds are static, distances and angles between player and other game's objects are easily computed from these logs.

Apart from logs, there are of course sound files of the IG giving directions. These were transcribed and together with some information from the logs transformed into ELAN files. ELAN is an annotation software (Sloetjes and Wittenburg, 2008). I will use the term *automatic annotations* for the ELAN files created from logs and transcribed audio.

Building on top of these automatic annotations are manual annotations. They are primarily concerned with referring expressions and also stored in ELAN format. Most referring expressions in GIVE aim to locate a button, which needs to be pressed. I will call the button, which is a goal of a referring expression, the target button. Which button is the target button of a reference is one of the layers in the manual annotations. Another layer of the annotations is some basic grouping of the references. Whether it is a reference to a single button, to a group of button, to a landmark and so on. The third layer looks deeper into the contents of the reference. It notes whether the reference contains for example the color of a button, whether distractors or landmarks are part of the reference or whether the reference explicitly points out that the button was already pressed before.

The previously mentioned logs, automatic annotations and manual annotations together form the dataset this chapter is dealing with.

An example of interaction between IF and IG is in the following text. Spatial information are transcribed in parentheses for the sake of clearness.

> (IF enters a room)
> IG: Go towards the red buttons.
> (IF turns right and start walking, but he turns too much)
> IG: No the ones next to the lamp...
> (IF corrects his direction)
> IG: Yeah that lamp. On the right.
> (IF is facing three buttons.)
> IG: Press the button on the wall you are looking at, that's far from the lamp and on the left.
> (IF goes towards the correct button and stops close to him)
> IG: Press it.

3.2 World and Demographic factors

As was noted repeatedly in Chapter 2, the world had significant influence on the task success rate in GIVE Challenge. However in the GIVE Challenge the worlds were designed to be of a different difficulty, whereas in the S-GIVE dataset they were designed to be similar in terms of the difficulty to minimize effects outside of navigation strategy. In addition, all the sessions were successful except for one which was discarded. The question about influence of worlds must be reformulated, since the task success rate no longer makes sense. Instead, I will measure task performance by time required to finish the task (duration) and also use other performance measure when appropriate.

Despite the design choices, I found out that the normal worlds 1 and 3, had a different mean duration (p-value 0.0473 for unpaired two-sample t-test). There are multiple explanation for this difference. Relatively small number of subjects is certainly one of them. We can also notice in the figure 3.3 slightly complicated system of hallways in the center of world-3. But this discovery does not have major influence on my work. Moreover, the difficult world did not have significant difference between their mean duration (p-value 0.6195 for unpaired two-sample t-test).

Another thing I was interested in was the influence of gaming experience of both participants on certain performance measures, namely on the duration, on the average speed of IF movement and the time the IF spent moving. The average speed of IF is simply a total distance the avatar controller by IF travelled in a session divided by duration. The time IF spent moving aggregates time where avatar was either motionless or only rotating in place.

I found correlations between gaming experiences and these variables. Table 3.1 shows the correlation matrix between gaming experience questions and performance measure. Not surprisingly, these correlation are especially high for the IF, since he/she is the one who is actually playing the world. The past gaming experience (questions 7 in third part of the questionnaire) is more important than contemporary playing (question 6 in the third part of the questionnaire). Most prominent are the familiarity of IF with 3D games (question 5), the IF hours per week spent playing games at the past peak gaming period (question 7), the same variable for IG and hours spent gaming per week for IF at present (question 6). For the difficult worlds some correlations change slightly. IF's gaming at past peak period has much weaker correlation with duration here. In general, individuals who are familiar with games (gamers) take less time to finish the world, they spent more time moving and they have higher speed.

The influence of SBSOD scores (second part of the questionnaire) on the task proficiency was another thing I have looked at. A correlation matrix revealed weak or almost no correlation between SBSOD scores and the time needed to complete the world, as can be seen in Table 3.2. In the difficult worlds, however, the correlation became slightly stronger.

The data suggest that there is positive correlation between male gender and task proficiency measured as duration. Table 3.3 shows these correlations. The effect of male IG diminishes in the difficult worlds but the effect of IF is even stronger in the difficult worlds. However there are several facts to take in consideration here. First of all, we don't have enough data to have a statistically significant conclusion. This correlation might have also been caused by having more male gamers than female gamers. In fact, Table 3.4 suggest so. Lastly, there has been research about influence of gender on spatial cognition and mental rotation; an example of more recent one is (Geary et al., 2000). They conclude that males are more proficient in tasks requiring mental rotation. Since the IGs have to do mental rotation while giving direction

World	Question	Duration	Speed	Time moving
Normal	IF hours/week peak gaming (7)	-0.411	0.486	0.428
	IF hours/week now (6)	-0.366	0.338	0.255
	IF familiarity 3D games (5)	-0.590	0.720	0.664
	IG hours/week peak gaming (7)	-0.359	0.450	0.435
	IG hours/week now (6)	0.079	0.155	0.180
	IG familiarity 3D games (5)	-0.230	0.221	0.224
Difficult	IF hours/week peak gaming (7)	0.111	0.199	0.135
	IF hours/week now (6)	-0.388	0.312	0.233
	IF familiarity 3D games (5)	-0.478	0.569	0.520
	IG hours/week peak gaming (7)	-0.287	0.715	0.715
	IG hours/week now (6)	0.149	0.348	0.420
	IG familiarity 3D games (5)	0.009	0.403	0.473

Table 3.1: Correlation matrix of gaming experiences and performance measures

World	Role	Duration
Normal	IF	-0.085
	IG	-0.082
Difficult	IF	0.162
	IG	-0.210

Table 3.2: Correlation between SBSOD scores and duration

in GIVE scenario, this might be a source of the correlation. Another paper worth considering on this topic is (Moffat et al., 1998), which found a gender difference in time required to finish a virtual maze.

World	Role	Duration
Normal	IF	-0.234
	IG	-0.277
Difficult	IF	-0.349
	IG	-0.106

Table 3.3: Correlation between male gender and duration

Role	Familiarity with 3D games
IF	0.341
IG	0.661

Table 3.4: Correlation between male gender and familiarity with 3D games

Table 3.5 shows correlation matrix for age. The age of IF have positive correlation with task proficiency measured in the duration and in difficult worlds this correlation is one the strongest ones. Older IF are also moving less and are generally slower. For IG the correlations have the same direction, however they are much weaker.

World	Role	Duration	Speed	Time moving
Normal	IF	0.175	-0.467	-0.448
	IG	0.275	-0.098	-0.125
Difficult	IF	0.614	-0.275	-0.196
	IG	0.016	-0.217	-0.232

Table 3.5: Correlation matrix of age and performance measures

Lastly I was interested how familiarity of participants with each other (questions 9-12 in the third part) influenced the task performance. Table 3.6 shows, that the knowledge of the partner had a positive impact on the task efficiency. The most correlated question was question 10, how well they know each other.

3.3 Referring expressions

Because REs are the main focus of my research, this section serves as an introductory analysis of REs in the S-GIVE dataset.

Overall, 793 REs were annotated in the manual annotations. Apart from time interval of the reference, several other facts were annotated in the manual annotations,

World	Question	Duration
Normal	IF Co-players in past (12)	-0.038
	IF Collaborative work (11)	0.109
	IF how well know each other (10)	0.529
	IF know each other (9)	0.164
	IG Co-players in past (12)	-0.124
	IG Collaborative work (11)	0.189
	IG how well know each other (10)	0.420
	IG know each other (9)	0.177
Difficult	IF Co-players in past (12)	-0.078
	IF Collaborative work (11)	0.170
	IF how well know each other (10)	0.437
	IF know each other (9)	0.247
	IG Co-players in past (12)	0.291
	IG Collaborative work (11)	0.245
	IG how well know each other (10)	0.361
	IG know each other (9)	0.189

Table 3.6: Correlation between participants familiarity and duration

as was mentioned in Section 3.1. First of all, the target button of each RE was annotated. The count of distinct target buttons is 29.

REs were also separated into 5 high-level categories depending on what is the target of the reference. The overview of the categories is in the following list:

- Target - Referring to the target button
- Group - Referring to a group of buttons, one of which is the target button
- Landmark object - Referring to a landmark (any object or room feature, but not a button) which will then be used to locate the target button
- Landmark button - Referring to a distractor button as a landmark
- Remove button - Referring to a distractor button to exclude it

Percentage count of the categories is in Table 3.7. References to target button are a dominant category. Around 10% of references are group references. References to landmarks occupy only 6% of all references.

Another layer of manual annotations looked into the contents of the REs. It was done through annotating several semantic elements of the REs of the Target category. They are listed in following list and their usage is summarised in Table 3.8. Please note that these elements are not necessarily exclusive with each other.

- Type - RE expressed the target object as its type (“button”)

Category	Percentage (%)
Target	82.47
Group	10.34
Landmark object	5.04
Landmark button	1.39
Remove button	0.76

Table 3.7: Percentage of REs in the categories

- One - RE expressed the target object as “one”
- Pronoun - RE expressed the target object as a pronoun
- Color - RE contained color of the target object
- Button location - RE contained relative location of the target object to a distracting button
- Object location - RE contained relative location of the target object to a distracting object (not a button)
- IF location - RE contained relative location of the target object to the IF
- Room location - RE contained relative location of the target object to a room
- History - RE informed that the target object was already manipulated with

Element	Percentage (%)
Type	55.35
Color	47.09
One	26.29
Button location	18.80
Object location	16.67
IF location	11.62
Pronoun	10.70
History	6.88
Room location	5.81

Table 3.8: Percentage of REs which contained a semantic element

3.4 Chains of references

Interesting phenomenon I have noticed and further explored in the dataset are consecutive references to one button. It can be seen in following sentences: “Straight ahead of you there on the opposite wall there are two blue buttons. Press the one on the right. The one close to the picture.” The IG started of with a references to a group of two buttons; the target button being one of those two. In the second sentence he picked out the target button from the group. In the last sentence IG made another reference containing landmark, adding redundant information. Since the references are concerned with one target button and follow each other relatively fast, I have called them chains of references (chains, in short).

The chains vary in length, from short ones, consisting of only two references, up to lengthy ones with eight references following each other. The example from previous paragraph is three references long. Figure 3.5 shows histogram of the chains length.

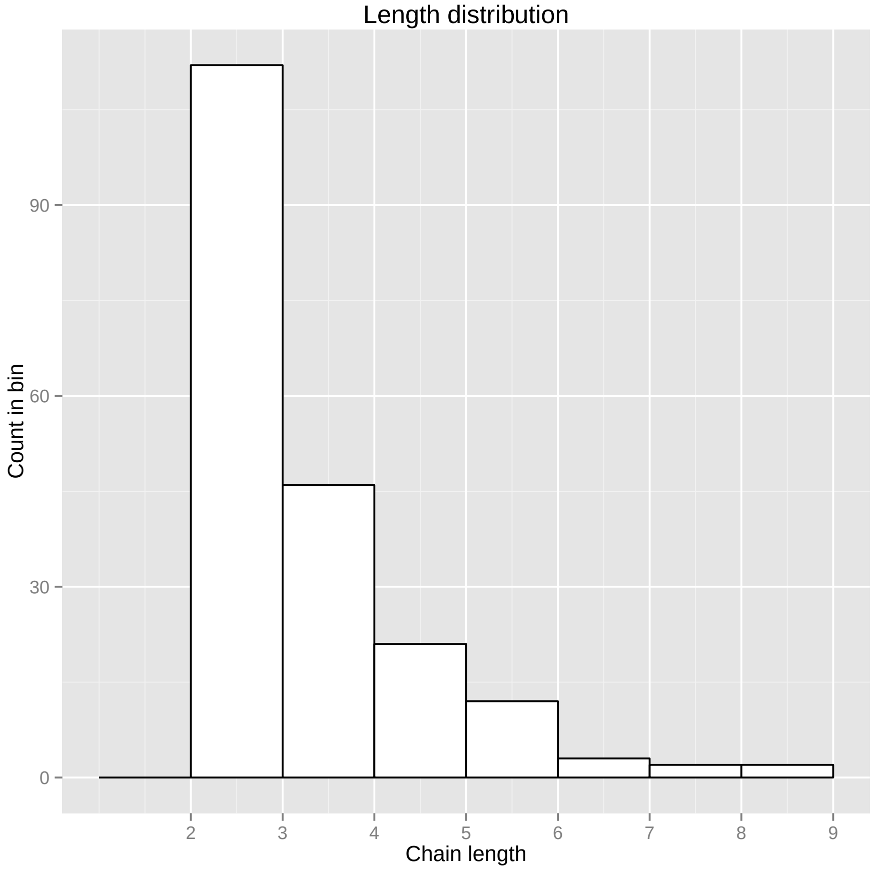


Figure 3.5: Histogram of the chain length

The chains are significant part of the discourse, in fact over 77% of all RE belong to a chain. However, the chains have multiple linguistic functions, which makes them difficult to explore. Even inside one chain, there are often combination of functions. I have manually annotated some common functions of the chains. Most common one is to inform IF that he is supposed to press the target button. A simple example of this function, which I call action, is in following discourse: “The red button in front of you. Press that one.” It may sound redundant to use action function as the experiment progresses, since the IF are not manipulating the buttons in any other way than pressing them. But I have found out that it’s often the case.

Another common function is confirmation: “That same red button we pressed before, we'll press that again. Yeah that one." The IG confirms at the end of previous utterance that IF is looking at or heading to a correct target button.

IG often utter a RE, which does not perfectly "picks out" the target button from the set of buttons in the room. IG can make up for that information deficit with confirmation function or further specify with another RE. That is a specification function, as in following two sentences: "Now that green button. You want the one closer to the lamp."

When a group of buttons is utilized in RE and the target button is part of that group, it is inevitable that IG will have to make another RE to "pick out" the target button from the group. Therefore group references imply chains of references and should be considered as one of the functions. Simple example of the group function: "Two blue buttons on the wall. Hit the blue button on the right side."

When IF has clearly chosen the wrong button, IG will try to correct that error. I call that an error function. Following extract features this function: "That blue button. No no no. The other. That one."

Summary of the functions is in Table 3.9. Please note that these functions are not exclusive. One chain can have both confirmation and specification function.

Function	Chains containing it (%)
Action	66.16
Confirmation	29.29
Specification	25.25
Group	24.74
Error	11.11

Table 3.9: Percentage of chains having specific functions

Chapter 4

Machine Learning on RE

RE are big part of language realization of a navigation system and especially so in the GIVE scenario. Part of my work was attempt to apply machine learning techniques on S-GIVE dataset with a clear goal to help navigation system with RE. This part is presented in this chapter.

In the first section I'll describe attempts at predicting timing of the first reference to a target button. Second section talks about modeling of chains of references.

4.1 Introduction

4.2 Timing of the first reference

Work of Stoia et al. (2006a) was previously mentioned in related work section of chapter 1. They applied machine learning on timing of the first reference in a 3D virtual world. The set-up of their experiment is quite similar to the GIVE's one and so I decided it would be interesting to replicate their methodology on GIVE dataset.

I defined the problem of the timing of the first reference as a classification task, as did Stoia et al. (2006a). More precisely binary classification, the two classes being either refer to the target button or delaying the reference. After extracting the first references to buttons which needed to be pressed from the dataset, I excluded plural references, because of their complexity. Some buttons were placed on top of each other and IG wasn't sure which one need to be pressed. These were excluded as well because of the unnecessary confusion. That left me with 351 first references. For each first reference I have chosen one negative example, where IG could refer to the target but chose not to. I picked negative examples randomly from interval between entering room and time of the first reference. Overall, that is 702 data-points with perfectly balanced classes.

As for features extraction, I have chosen similarly to Stoia et al. (2006a) various spatial information. For the positive examples, I averaged these spatial information over 0.6 seconds window centered on the time of the reference. Reasoning for that, is that IG take scene situation into consideration before and possibly after they start uttering the reference. For negative examples I chose not to averaged them, since they are chosen randomly. All features are listed in the following list. The list also includes figures' numbers. These figures are histograms of the attributes, separated for both classes and can be found in appendix.

- Distance to target button - Figure 3
- Absolute value of angle to target - Figure 4
- Whether target is visible (True/False) - Figure 5
- Number of distractors - Figure 6
- Number of distracting buttons - Figure 7
- Number of visible buttons with smaller angle to IF than the target button - Figure 8

Once I have extracted these features I used three machine learning techniques: C4.5 decision tree because of their easy interpretation, naive Bayes to observe effect of all attributes and Support vector machine for linear classification. I used Weka software implementation of previous algorithms (Hall et al., 2009).

For all the algorithms I used a standard ten-fold cross validation. The results can be seen in table 4.1. Pruned decision tree for timing of the first reference can be seen in figure 4.1. I used two simple baselines to compare the results with. I have perfectly balanced classes so the first baseline is 50%. Simple rule for the first reference is to refer when the target is visible, and delay it if the target is not visible. In my case that rule has an accuracy of 64.2%.

Model	Accuracy (%)
Class baseline	50.00
Visibility baseline	64.2
Naive Bayes	**64.70**
C4.5	63.31
SVM	55.42

Table 4.1: Results of first reference timing modeling

Only one algorithm was able to get over visibility baseline and not by a significant amount. These results were surprising, because Stoia et al. (2006a) had success with the same approach on a similar dataset. Reasons for this difference are probably in the differences between their experiment and GIVE set-up. First, their tasks also included different actions than pushing buttons (e.g. picking up items). Second, their worlds had higher diversity of the distractors and smaller frequency of them. With increasing number of distractors and particularly distractors of the same category, it seems that spatial features loose their power in predicting the timing of the first reference. The number of visible distractors was the best attribute in their decision tree, but in my tree (figure 4.1) it had lower information gain. Moreover, the decision tree did the split on the number of visible buttons not the number of all visible distractors.

After the timing of the first references classification proved to be more difficult than excepted, I have switched from predicting the timing of the references to predicting their content. Next, I focused on chains of references.

```
Distance to target <= 4.45
| Is target visible? = False: DELAY
| Is target visible? = True
| | Distance to target <= 2.51: REFER
| | Distance to target > 2.51
| | | Number of distracting buttons <= 4.67: REFER
| | | Number of distracting buttons > 4.67: DELAY
Distance to target > 4.45: DELAY
```

Figure 4.1: Decision tree for first reference timing

4.3 Chains of references

Section 3.4 introduced the phenomenon of chains of references. It also analysed various linguistic functions, the chains can play in REG. This section will build on top of this analysis, by employing machine learning techniques to model the chains.

Valid and important question is whether chains aren't something, which should actually be avoided, instead of modelled. That is, what is the relation between usage of chains and task performance measure, such as duration. To address this issues, I used linear regression predicting the duration of the experiment explained by the average chain length. Figure 4.2 does contain a hint of a trend, but also contains a lot of counter-examples.

When I applied linear regression the R^2 was 0.188, which means the average chain length explains very little of the duration variation. If the average chain length was to significantly influence the duration, I would except much higher R^2. Correlation between the duration and average chain length is not low: 0.433, however correlation does not imply causation. Longer chains can be caused by errors of IF or IG and the errors are also likely to increase the duration. Based on these facts, I don't believe the chains are harmful phenomenon and it makes sense to proceed with attempting to model them.

With modeling the chains of references, there are numerous problems to tackle. Simple presence of the chain can be thought of as a classification task. It can be interesting try to predict the chain length. Having established common linguistic functions of the chains, classify chains whether they would contain these function is another task to consider. I looked into each of these problem and together with ex-

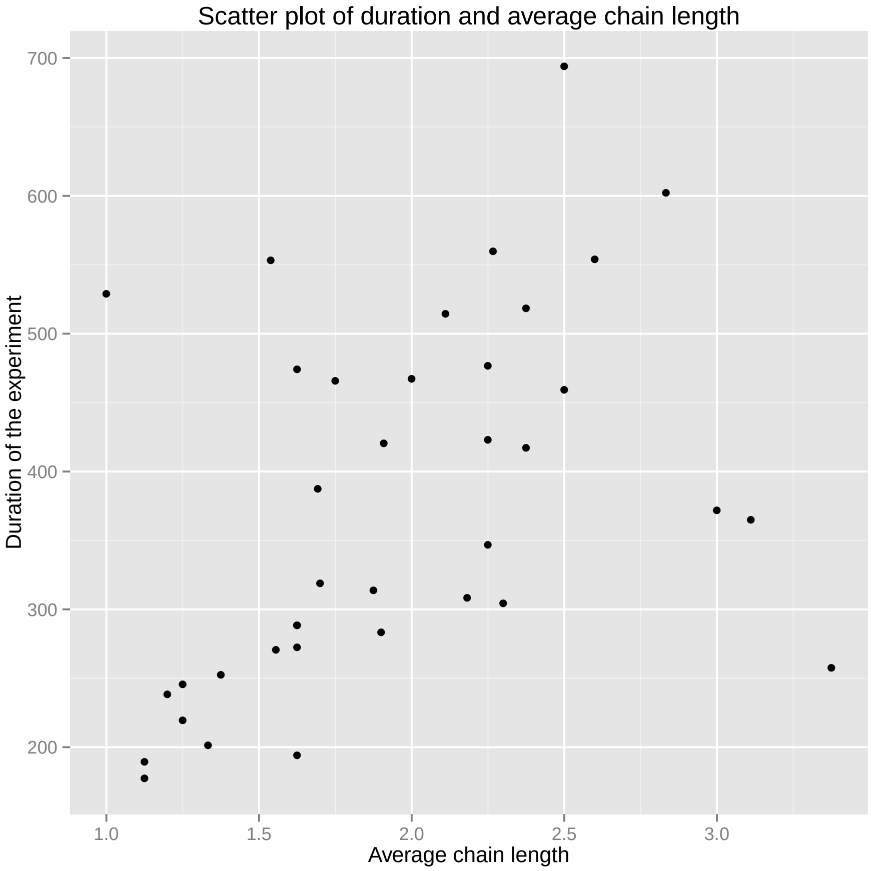


Figure 4.2: Scatter plot of duration and chain length

tracting relevant features, thoroughly explored using machine learning for modeling the chains.

As a side note, during exploring chains of references I switched from Weka implementation of machine learning techniques (Hall et al., 2009) to scikit learn package for Python (Pedregosa et al., 2011). The main reason being easier automation of the entire pipeline, therefore speeding-up of the whole process and more control over whole process.

4.3.1 Presence of the chains

From Section 3.4, the functions the chains can play in a discourse are known. However, what role does play the scene complexity and particular situations in the presence of the chains? Will more complex scene spawn more chains or are the chains too complex to be captured by simply looking at where they are created? To answer these question, I extracted scene complexity features and the target button for all references and classified whether the chains was present or not.

The features I extracted from the dataset are in the following list:

- Target button
- Number of objects In the room - Figure
- Number of buttons in the room - Figure
- Number of landmarks in the room - Figure
- Number of very close buttons to the target button (closer than 0.3m) - Figure
- Number of buttons on the same wall as target button - Figure
- Number of close buttons to the target button (closer than 1m) - Figure
- Number of far buttons (farther than 1.5m) - Figure

The target button is a categorical feature, so I used DictVectorizer class in scikit to transform it into multiple numerical features. I further tried to use all features, select 8 best features using SelectKBest class from scikit with χ^2 statistic as a scoring function and select 16 best features using the same strategy. For algorithms I used Decision Tree, Naive Bayes, Support Vector Machines (SVM), One-Nearest Neighbour (1-NN), Two-Nearest Neighbours (2-NN) and Random Forest. This selection employs various approaches for classification task, each of the algorithms having strong and weak points. For evaluation, I used standard ten-fold cross validation and compared the accuracy of the classifiers with majority class baseline. From now on, I will also include double the standard deviation (std). The idea behind the double is that if the features follow normal distribution and the three sigma rule is

Features	Model	Mean accuracy (%)	2× Std
Majority baseline		55.2	
All	Decision Tree	55.5	13.5
	Naive Bayes	55.8	6.6
	SVM	56.1	11.2
	1-NN	56.3	9.3
	2-NN	53.6	12.1
	Random Forest	55.0	13.6
8 best	Decision Tree	59.7	14.6
	Naive Bayes	56.4	2.9
	SVM	58.3	13.1
	1-NN	58.8	11.1
	2-NN	56.1	14.4
	Random Forest	**60.0**	16.6
16 best	Decision Tree	55.8	13.5
	Naive Bayes	56.4	2.9
	SVM	58.3	13.9
	1-NN	56.9	9.1
	2-NN	53.3	12.8
	Random Forest	57.2	12.8

Table 4.2: Results of chains presence modeling

applied, the chance of the accuracy being in interval of ± standard deviation times two is 95.45% . The results are summarized in Table 4.2.

The best mean accuracy had the Random Forest for 8. However, it wasn't significantly better than majority class baseline. Take into account the standard deviation, none of the classifier and features combination outperformed the baseline. From these results, I conclude that the chains presence is not dependent only on the scene complexity and specific scenarios. IG strategies for referencing, IF behaviour and other circumstances play role in creation of the chains.

4.3.2 Chain length prediction

After not being able to predict presence of the chains based on spatial information and target button, I was interested if I could predict the chain length based. I used same attributes as in previous classification of the chains' presence, but added two more features concerned with IF movement behaviour. Namely, the ratio of the time IF spent not moving at all through the chain duration and ratio of the time IF spent only rotating in place through the chain duration. The idea behind these features is that IF who is not moving or just looking around is an indication of him/her being confused, which then should produce more referring expression for the chain.

For reference, I list all features and add figures numbers for the two new attributes:

- Target button
- Number of objects In the room
- Number of buttons in the room
- Number of landmarks in the room
- Number of very close buttons to the target button (closer than 0.3m)
- Number of buttons on the same wall as target button
- Number of close buttons to the target button (closer than 1m)
- Number of far buttons (farther than 1.5m)
- Ratio of time IF spent not moving - Figure
- Ratio of time IF spent rotating - Figure

I applied linear regression, predicting chain length based on three groups of mentioned attributes - target, spatial features and IF movement behaviour features. I evaluated the regressions by looking at R^2. The results are in Table 4.3.

None of the regressions from Table 4.3 are particularly good at predicting chain length. Combining spatial and IF movement features does increase the percentage of chain length variation explained by the model, suggesting that these features have, however small, effect on the chain length. Once again it shows that chains are complex phenomenon, influenced by IF, IG and scene variables.

Features	R^2
Target button	0.154
Spatial	0.146
IF movement	0.126
Spatial ad IF movement	0.262

Table 4.3: Results of chains length modeling

4.3.3 Closer look at chains' content

Despite not being able to predict the chains presence and length, I was still interested in chains and decided to look closely into chains content. First, I took advantage of having annotated the functions in the chains, as introduced in Section 3.4. I focused on specification and group function and tried to predict whether the chain will contain these functions. Second, I tried to predict whether the chain provide new information about the target button after its first reference. For example IG could add RE about position of the target button relative to a landmark in the third reference of the chain. Reasoning behind this classification is trying to predict when IG add information to the chains.

For all these classification task, I used the same features as in the chain length prediction, that is: the target button, spatial features, IF movement behaviour features and combination of the previous two. Again, I used ten-fold cross validation for evaluation and compared that to majority class baseline. The results for specification function are in Table 4.4, for group function in Table 4.5 and for predicting new information in Table 4.6. For algorithms, I maintained broad range of algorithms, similarly to previous machine learning attempts.

The best classifiers for these 3 tasks can be found in respective results tables, marked by bold font. Similarly to previous attempts, all combinations of attributes and algorithms had the accuracy around the baseline. The last section of this chapter will address the negative results to more depth.

4.4 Room memory

Last machine learning problem I tackled was using memory of the previously visited rooms in RE for room switching. When IF has to return to a recently visited room, IG sometimes employ RE such as: "Go to the room you were just in." Deciding whether to use the memory in room switching RE can be thought of as a classification task.

I extracted all moments where IF went between rooms and determined whether the room memory was used from what was IG saying before going into the new room. I also extracted 3 features:

- Was IF in the new room before?
- How many seconds before, he/she was there?

Features	Model	Mean accuracy (%)	2× Std (%)
Majority baseline		74.7	
Target button	Decision Tree	73.3	6.0
	Naive Bayes	38.0	22.4
	SVM	74.7	5.3
	1-NN	70.7	14.8
	Random Forest	74.0	7.2
IF movement	Decision Tree	64.0	21.7
	Naive Bayes	72.7	11.1
	SVM	74.7	5.3
	1-NN	**77.3**	12.2
	Random Forest	71.3	16.9
Spatial	Decision Tree	74.7	16.7
	Naive Bayes	66.7	20.7
	SVM	71.3	14.7
	1-NN	68.7	23.9
	Random Forest	74.7	18.7
Spatial and movement	Decision Tree	66.7	23.1
	Naive Bayes	66.7	20.7
	SVM	71.3	18.9
	1-NN	76.7	8.9
	Random Forest	68.0	16.7

Table 4.4: Results of specification function in chains modeling

Features	Model	Mean accuracy (%)	2× Std (%)
Majority baseline		75.4	
Target button	Decision Tree	81.3	14.4
	Naive Bayes	38.0	14.7
	SVM	75.3	6.1
	1-NN	79.3	13.9
	Random Forest	79.3	17.3
IF movement	Decision Tree	61.3	23.7
	Naive Bayes	75.3	6.1
	SVM	75.3	6.1
	1-NN	73.3	15.8
	Random Forest	68.0	16.7
Spatial	Decision Tree	80.0	10.3
	Naive Bayes	77.3	18.1
	SVM	77.3	8.8
	1-NN	77.3	13.6
	Random Forest	**82.0**	13.4
Spatial and movement	Decision Tree	71.3	16.9
	Naive Bayes	78.0	15.8
	SVM	77.3	8.8
	1-NN	79.3	9.3
	Random Forest	74.0	18.3

Table 4.5: Results of group function in chains modeling

Features	Model	Mean accuracy (%)	2× Std (%)
Majority baseline		67.3	
Target button	Decision Tree	68.3	19.6
	Naive Bayes	38.0	15.8
	SVM	67.3	4.0
	1-NN	56.0	31.1
	Random Forest	68.0	19.6
IF movement	Decision Tree	58.7	8.0
	Naive Bayes	68.0	11.6
	SVM	67.3	4.0
	1-NN	41.3	20.5
	Random Forest	57.3	12.2
Spatial	Decision Tree	68.0	16.7
	Naive Bayes	60.7	28.3
	SVM	71.3	8.5
	1-NN	59.3	21.9
	Random Forest	67.3	17.3
Spatial and movement	Decision Tree	66.0	26.3
	Naive Bayes	62.7	26.1
	SVM	**72.7**	9.3
	1-NN	59.3	19.3
	Random Forest	64.7	18.9

Table 4.6: Results of new information in chains modeling

- How many rooms before, he/she was there?

The last feature being of a categorical character, once again DictVectorizer was used for transformation and from the transformed numerical values I selected 3 best with SelectKBest and χ^2 statistic. The results are in Table 4.7.

Model	Mean accuracy (%)	2× Std (%)
Majority baseline	78.5	
Decision Tree	78.0	7.1
Naive Bayes	68.3	7.3
SVM	82.2	6.2
1-NN	80.8	6.6
Random Forest	**84.4**	6.6

Table 4.7: Results of room memory modeling

Despite not being able to significantly outperform the baseline using machine learning, I used the knowledge gained in the room memory modeling attempt. In the systems I developed for the experiment, which will be discussed in the last chapter, I created a simple rule for using room memory and enhanced the REG process.

4.5 Conclusion

Conclusion

Bibliography

Herbert H Clark and Meredyth A Krych. Speaking while monitoring addressees for understanding. *Journal of Memory and Language*, 50(1):62–81, 2004.

Carlos Gómez Gallo, T Florian Jaeger, James F Allen, and Mary D Swift. Production in a multimodal corpus: how speakers communicate complex actions. In *LREC*, 2008.

Andrew Gargett, Konstantina Garoufi, Alexander Koller, and Kristina Striegnitz. The give-2 corpus of giving instructions in virtual environments. In *LREC*, 2010.

David C Geary, Scott J Saults, Fan Liu, and Mary K Hoard. Sex differences in spatial cognition, computational fluency, and arithmetical reasoning. *Journal of Experimental Child Psychology*, 77(4):337–353, 2000.

Eun Young Ha, Joseph F Grafsgaard, Christopher M Mitchell, Kristy Elizabeth Boyer, and James C Lester. Combining verbal and nonverbal features to overcome the 'information gap' in task-oriented dialogue. In *Proceedings of the 13th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 247–256. Association for Computational Linguistics, 2012.

Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H Witten. The weka data mining software: an update. *ACM SIGKDD Explorations Newsletter*, 11(1):10–18, 2009.

Mary Hegarty, Anthony E Richardson, Daniel R Montello, Kristin Lovelace, and Ilavanil Subbiah. Development of a self-report measure of environmental spatial ability. *Intelligence*, 30(5):425–447, 2002.

Rodger Kibble and Richard Power. An integrated framework for text planning and pronominalisation. In *Proceedings of the first international conference on Natural language generation-Volume 14*, pages 77–84. Association for Computational Linguistics, 2000.

Alexander Koller, Kristina Striegnitz, Donna Byron, Justine Cassell, Robert Dale, Johanna Moore, and Jon Oberlander. The first challenge on generating instructions in virtual environments. In *Empirical Methods in Natural Language Generation*, pages 328–352. Springer, 2010a.

Alexander Koller, Kristina Striegnitz, Andrew Gargett, Donna Byron, Justine Cassell, Robert Dale, Johanna Moore, and Jon Oberlander. Report on the second nlg challenge on generating instructions in virtual environments (give-2). In *Proceedings of the 6th international natural language generation conference*, pages 243–250. Association for Computational Linguistics, 2010b.

Alexander Koller, Maria Staudte, Konstantina Garoufi, and Matthew Crocker. Enhancing referential success by tracking hearer gaze. In *Proceedings of the 13th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 30–39. Association for Computational Linguistics, 2012.

Max M Louwerse, Nick Benesh, Mohammed E Hoque, Patrick Jeuniaux, Gwyneth Lewis, Jie Wu, and Megan Zirnstein. Multimodal communication in face-to-face computer-mediated conversations. In *Proceedings of the 28th Annual Conference of the Cognitive Science Society*, pages 1235–1240, 2007.

Scott D Moffat, Elizabeth Hampson, and Maria Hatzipantelis. Navigation in a "virtual" maze: Sex differences and correlation with psychometric measures of spatial ability in humans. *Evolution and Human Behavior*, 19(2):73–87, 1998.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Han Sloetjes and Peter Wittenburg. Annotation by category: Elan and iso dcr. In *LREC*, 2008.

Laura Stoia, Donna K Byron, Darla Magdalene Shockley, and Eric Fosler-Lussier. Sentence planning for realtime navigational instructions. In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, pages 157–160. Association for Computational Linguistics, 2006a.

Laura Stoia, Darla Magdalene Shockley, Donna K Byron, and Eric Fosler-Lussier. Noun phrase generation for situated dialogs. In *Proceedings of the fourth international natural language generation conference*, pages 81–88. Association for Computational Linguistics, 2006b.

Kristina Striegnitz, Alexandre Denis, Andrew Gargett, Konstantina Garoufi, Alexander Koller, and Mariët Theune. Report on the second second challenge on generating instructions in virtual environments (give-2.5). In *Proceedings of the 13th European Workshop on Natural Language Generation*, pages 270–279. Association for Computational Linguistics, 2011.

Kristina Striegnitz, Hendrik Buschmeier, and Stefan Kopp. Referring in installments: a corpus study of spoken object references in an interactive virtual environment. In *Proceedings of the Seventh International Natural Language Generation Conference*, pages 12–16. Association for Computational Linguistics, 2012.

Jette Viethen, Robert Dale, and Markus Guhe. Generating subsequent reference in shared visual scenes: Computation vs. re-use. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1158–1167. Association for Computational Linguistics, 2011a.

Jette Viethen, Robert Dale, and Markus Guhe. The impact of visual context on the content of referring expressions. In *Proceedings of the 13th European Workshop on Natural Language Generation*, pages 44–52. Association for Computational Linguistics, 2011b.

Appendix

.1 Histograms for first reference ML

This section contains attributes' histograms for timing of the first reference ML.

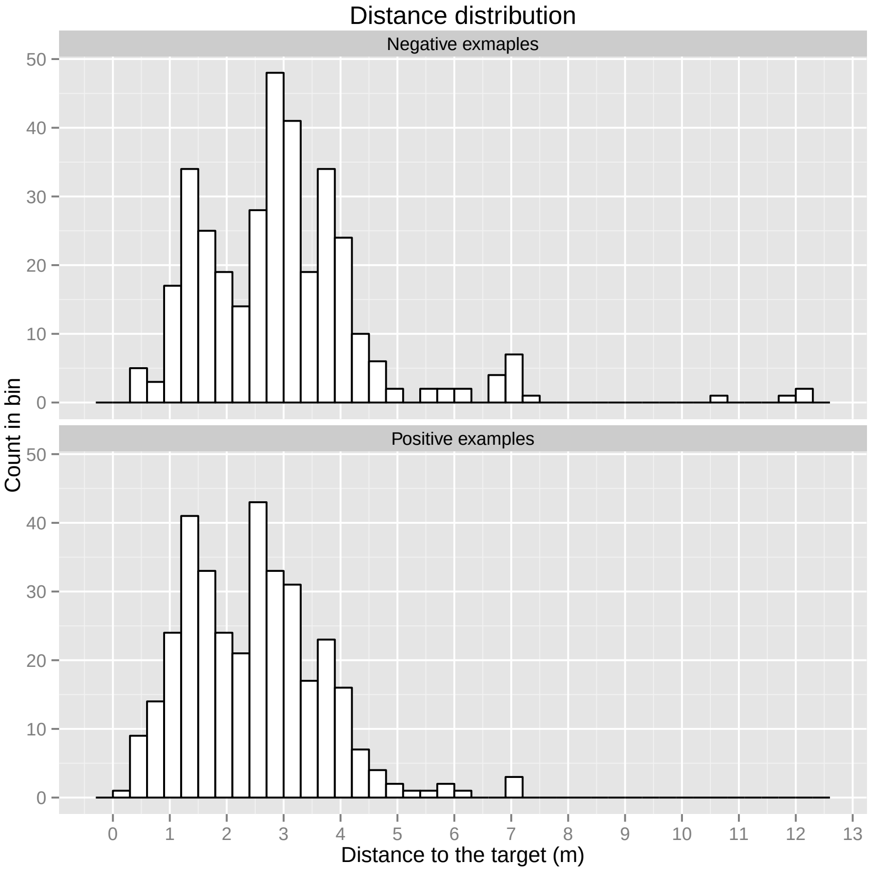


Figure 3: Histogram of attribute distance to the target

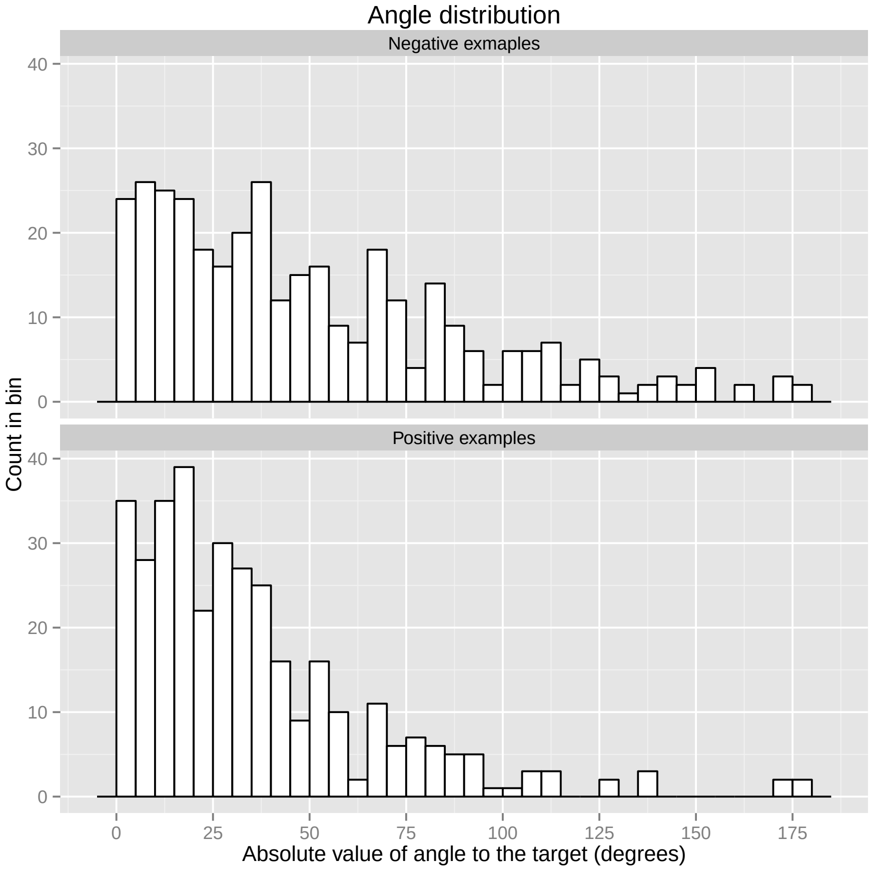


Figure 4: Histogram of attribute angle to the target

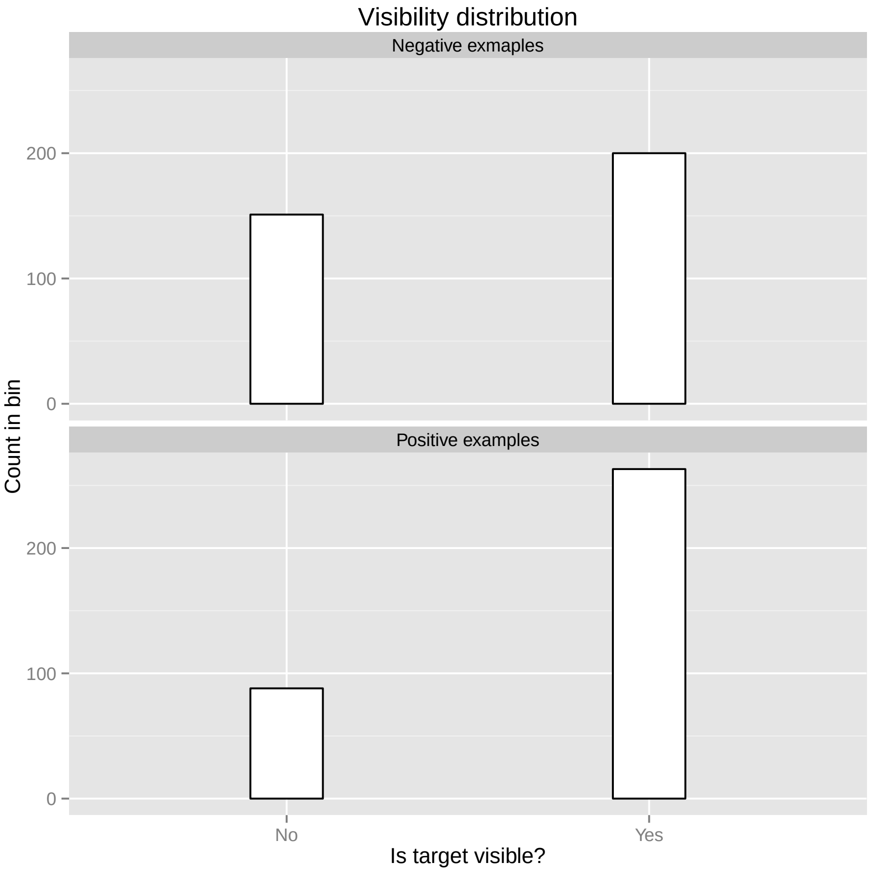


Figure 5: Histogram of attribute whether the target is visible

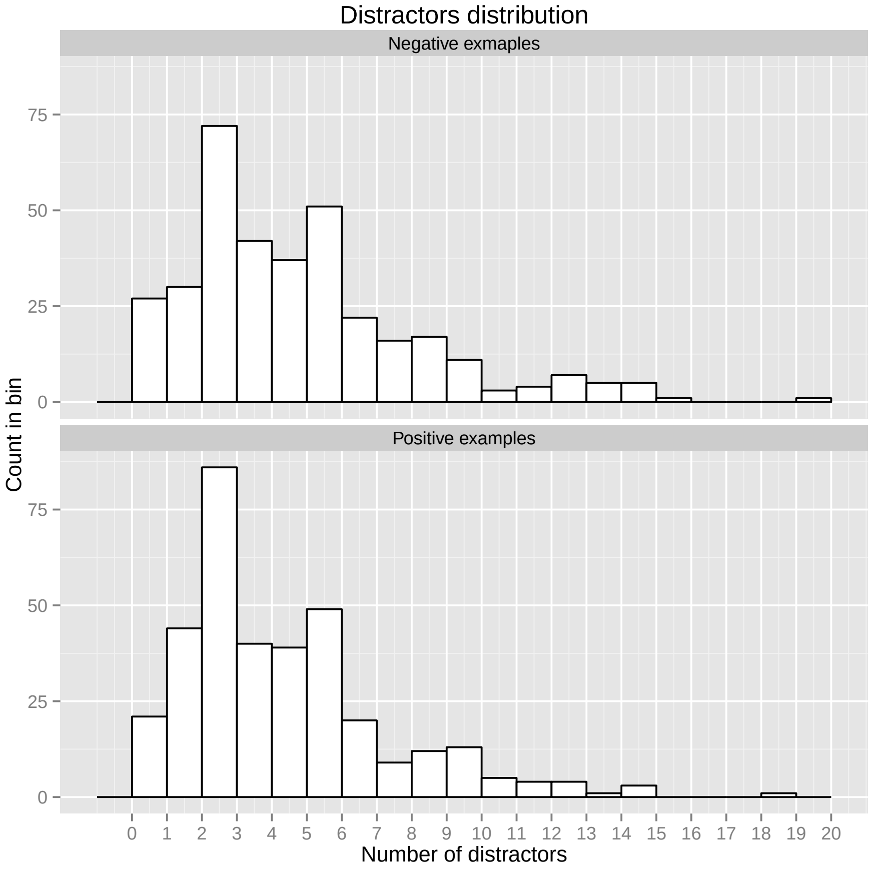


Figure 6: Histogram of attribute number of distractors

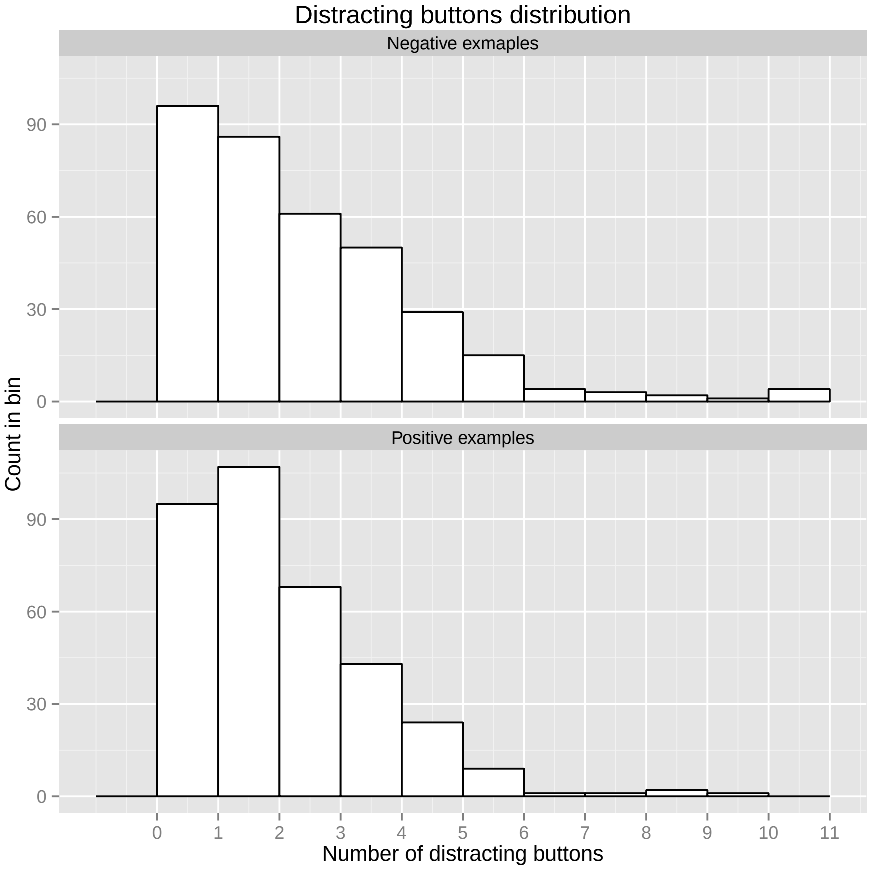


Figure 7: Histogram of attribute number of distracting buttons

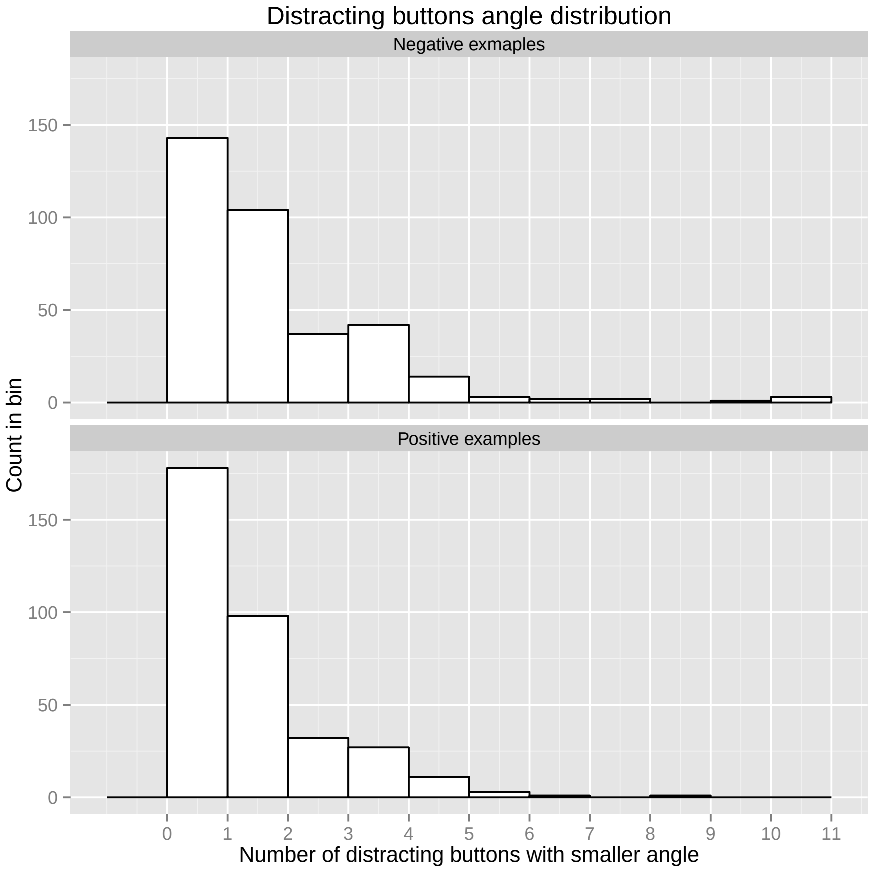


Figure 8: Histogram of attribute number of distracting buttons with lesser angle

.2 Histograms for chains ML

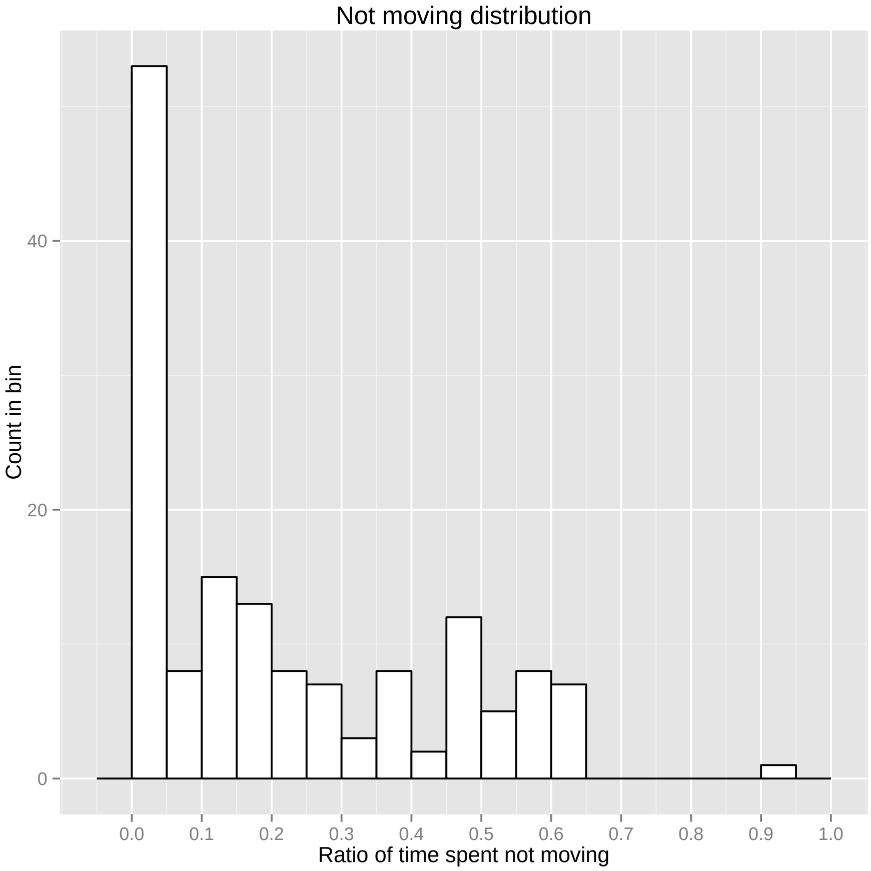


Figure 9: Histogram of attribute time spent not moving

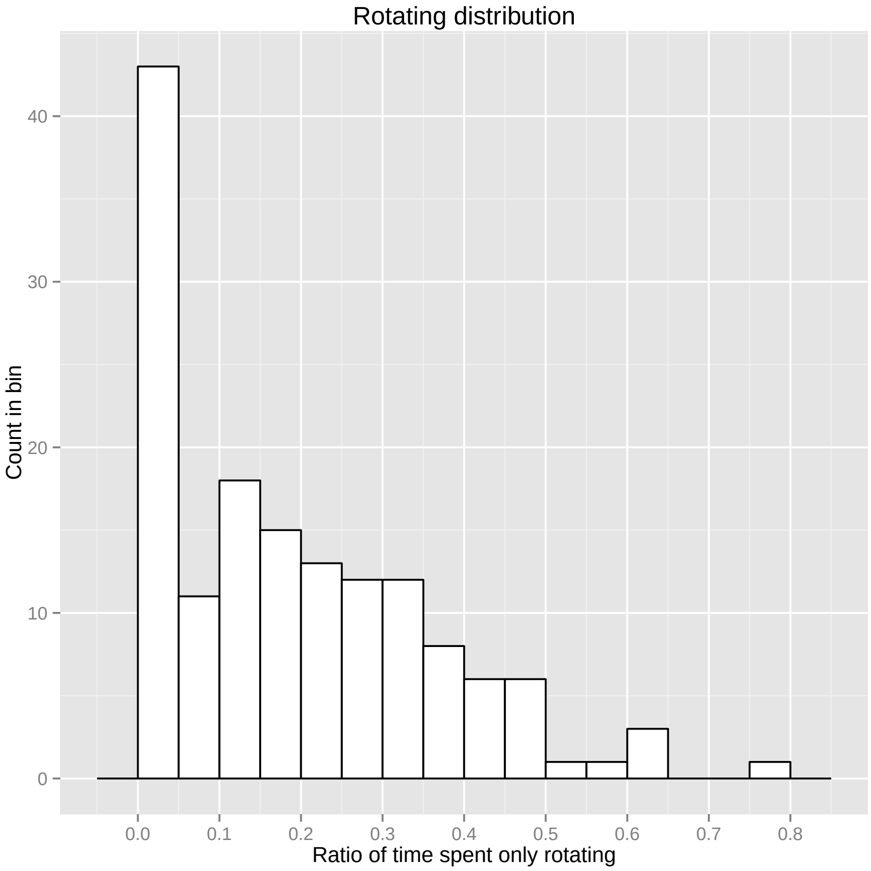


Figure 10: Histogram of attribute time spent rotating

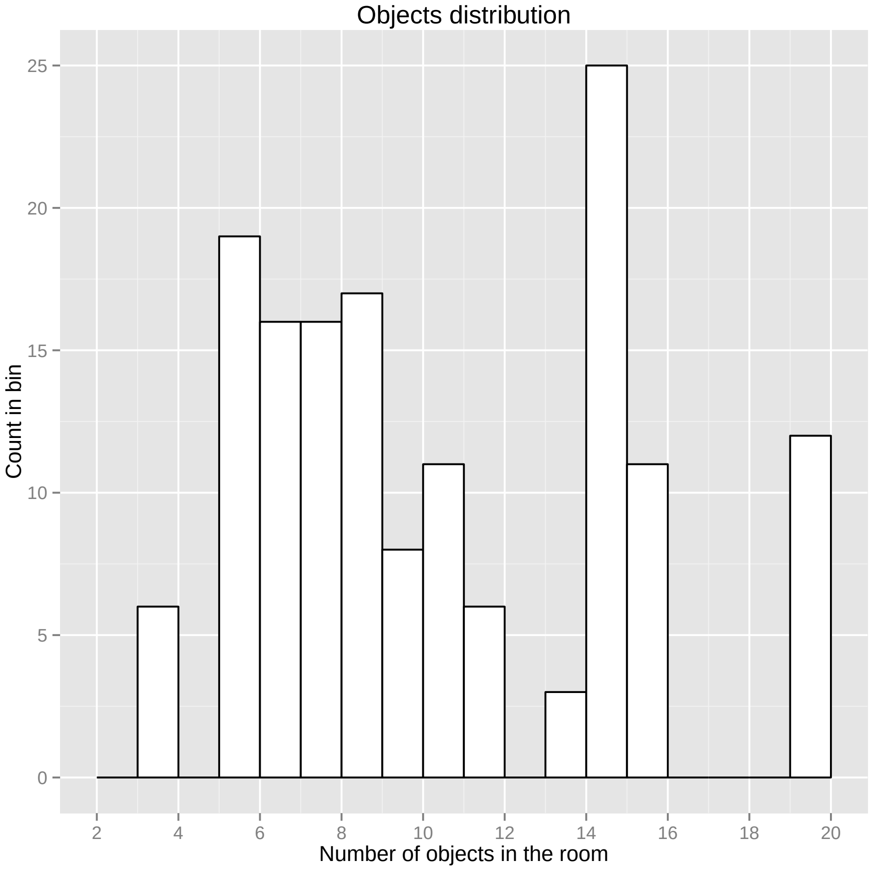


Figure 11: Histogram of attribute objects in the room

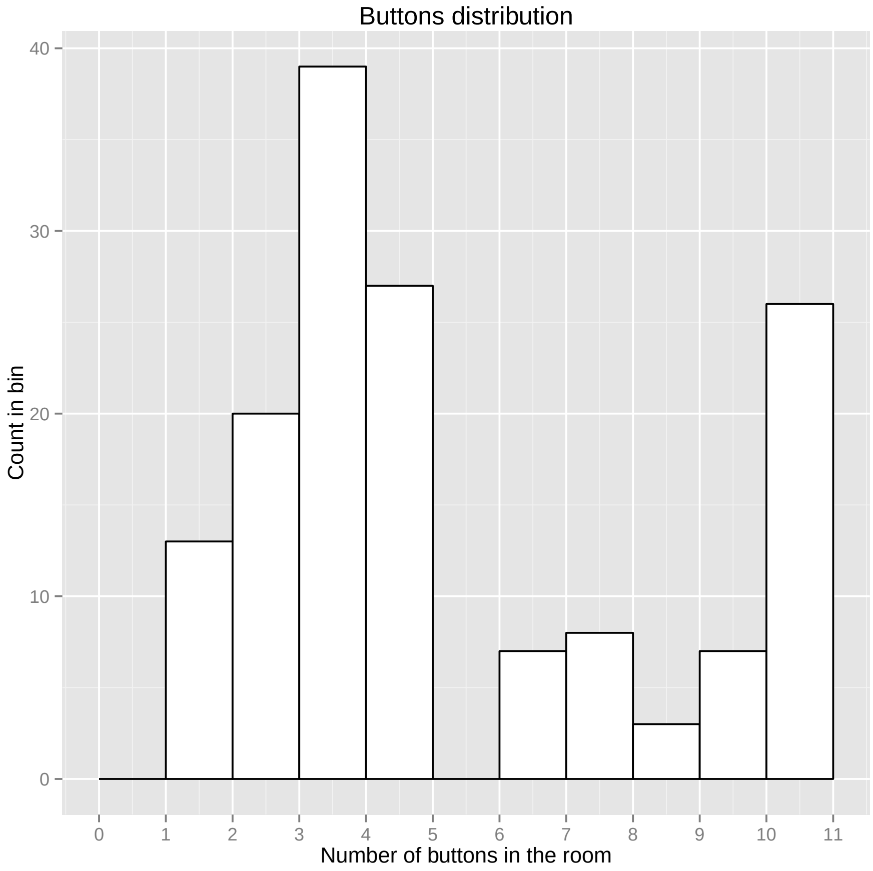


Figure 12: Histogram of attribute buttons in the room

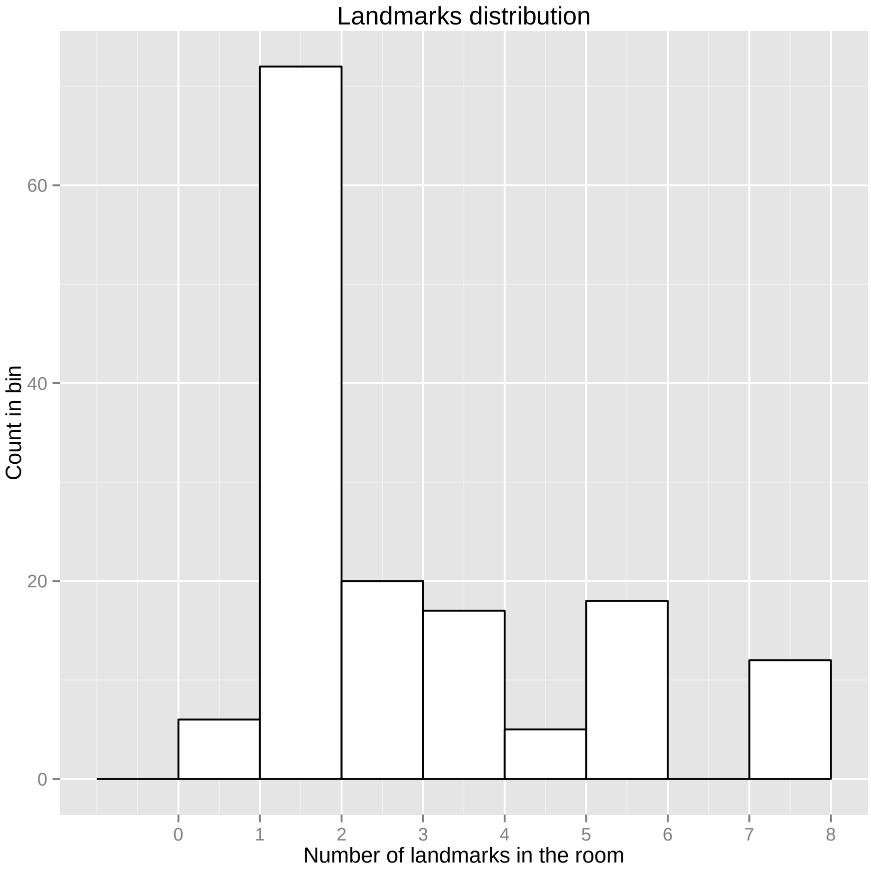


Figure 13: Histogram of attribute landmarks in the room

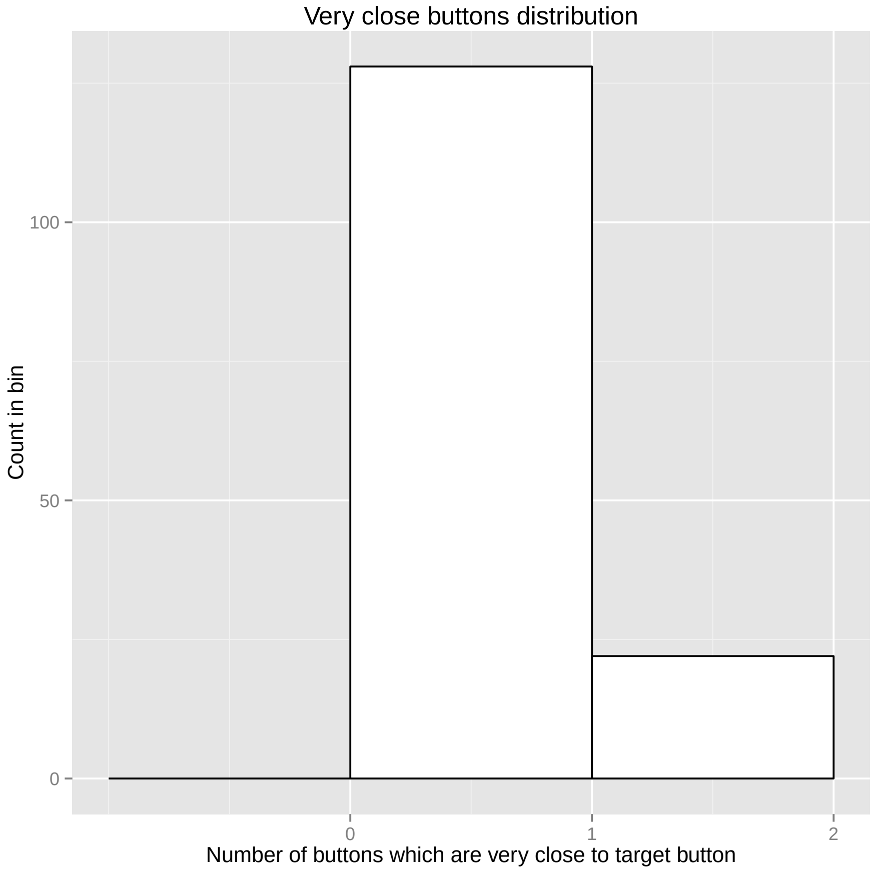


Figure 14: Histogram of attribute very close buttons to the target

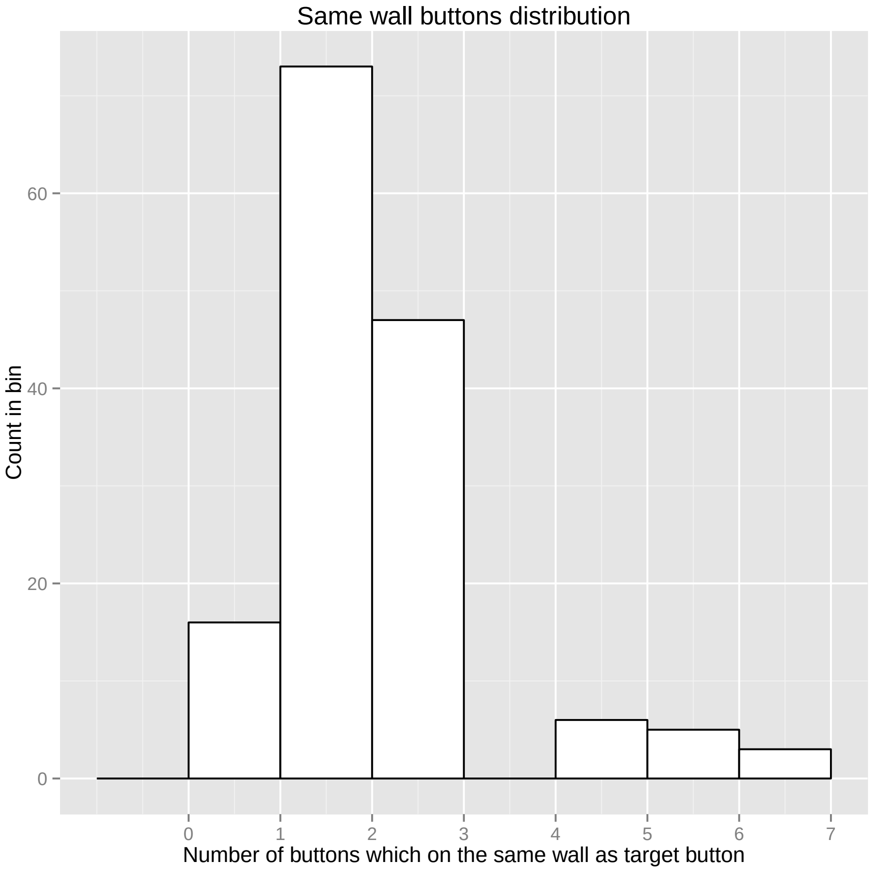


Figure 15: Histogram of attribute buttons on the same wall as the target

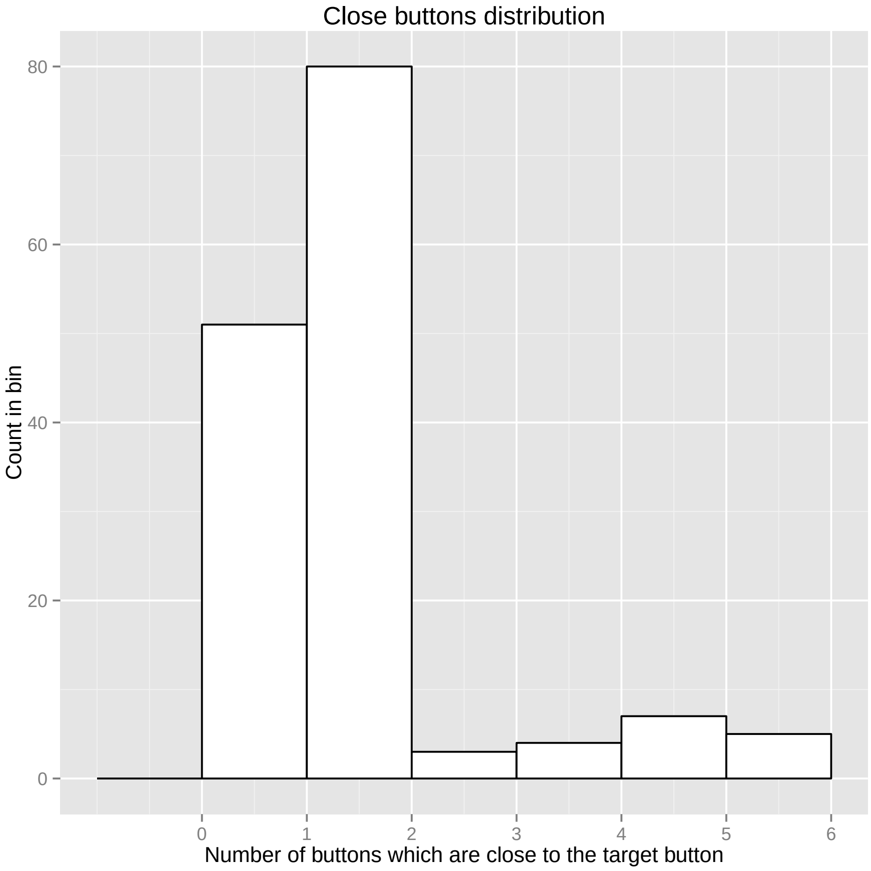


Figure 16: Histogram of attribute close buttons to the target

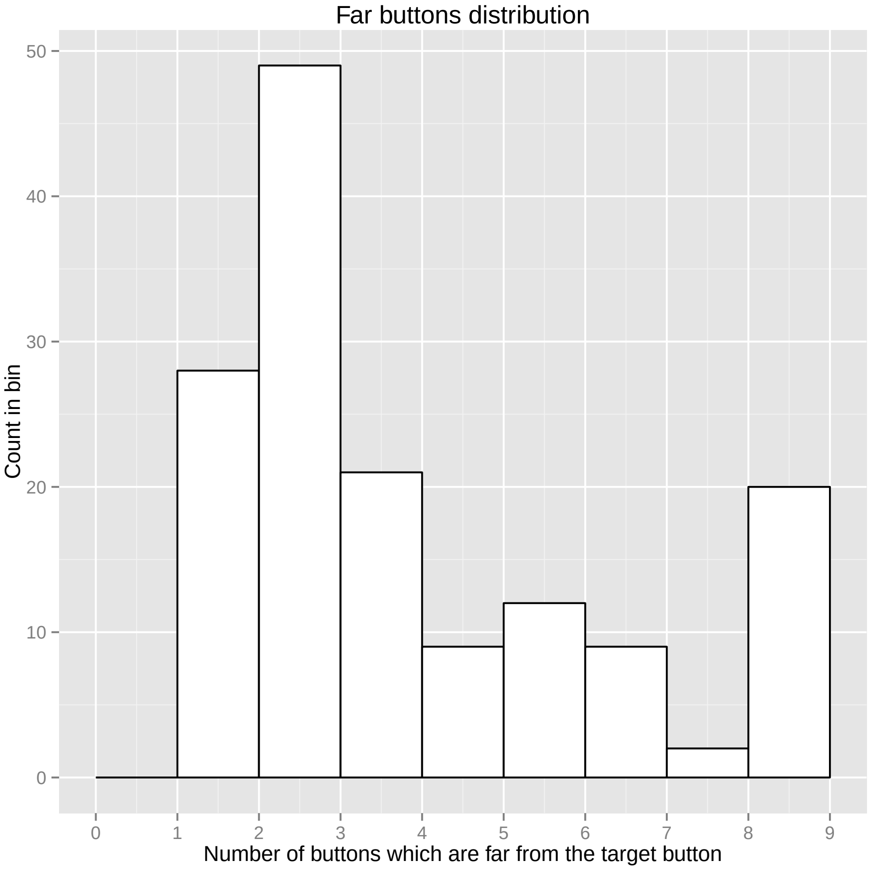


Figure 17: Histogram of attribute far buttons from the target