Master thesis (ICA-0357014)

Computer-generated Dialogues through Sequential Pattern Mining

Author:
E. Broersma

Supervisor:
Dr. F.P.M. Dignum

December 12, 2012

Abstract

In this thesis I attempt to answer the following question:

> How does generating artificial, goal-oriented dialogue using automated data mining weigh up against a manual approach to this problem?

I wrote the Pet Shop Game, a web application that anonymously couples players and allows them to engage in a dialogue game set in a pet shop scenario. Players can express themselves in a controlled natural language and perform relevant physical actions. The dialogue data is then mined for sequential patterns using the GSP algorithm, which also provides a taxonomy parameter that is used in this case to not only find patterns at the utterance level but also on the level of speech acts. Sequential patterns can be translated into rules suitable for use by an AI program to replace a human player in the Pet Shop Game. These data mined rules, though arguably of the same structure as rules used by hand-written chatbots, are weaker and find their strength in numbers. That is, many data mined rules combine to come to the right conclusion, whereas hand-written rules are typically much stronger. I suggest a comparative study of serious implementations of both approaches should be made in order to arrive at a more definitive judgement.

Contents

Chapter 1

Introduction

Computer games are a relatively new medium with an industry that has grown bigger than the movie industry in terms of revenue. Games are becoming more and more realistic in terms of visuals and more grown-up in terms of narrative.

Games are an inherently interactive medium, where players are encouraged to interact with a world of possibilities. To support this world of possibility computer game technology employs advanced technology to make the world as lifelike as possible. Physics engines create the laws of nature, rendering engines create beautiful pictures, and networking technology makes all of this appear on the user's computer as lagfree as possible.

An important part of a lot of game worlds are their virtual inhabitants: non-player characters or NPCs. Just as the behaviour of a falling rock in a game is governed by the physics engine, so is the behaviour of the NPCs governed by a system. This system is commonly referred to as the artificial intelligence or AI. The behaviour of NPCs ranges from realistic movement within the game world to the way they communicate their intentions. In this thesis I would like to focus on the latter aspect, and more specifically on player–NPC dialogues.

Currently, dialogues in games consist at best of a pre-written tree of dialogue options. Whenever a player selects an option, the NPC they are interacting with gives its canned response and a new set of utterances becomes available to the player. While this is a proven approach, it is prone to break the feeling of immersion or the player's identification with their in-game character. This is definitely an area where computer game AI should do better.

1.1 What this thesis is about

This thesis is about generating lifelike dialogues for computer games. Because full-blown communication is out of reach for now, at least until we solve the hard problem of AI [13], the choice will be made instead to focus on a class of dialogues with favourable regularities.

Functional conversation—unlike small talk or banter—is about achieving a concrete goal, like buying a bread at the baker's. When engaging in such a conversation we usually follow a mutually shared protocol. One first exchanges greetings with the baker, then (not necessarily in this order) communicates which type of bread one wants to buy and pays the correct amount, and finally greetings are exchanged once more. The more commonplace the conversation's goal is, the more standardized the protocol will be [2].

An AI making use of this theory of behavioural protocols will need to follow an artificial protocol modelled after the natural protocols that emerge in every day human behaviour. Artificial protocols can be hand-crafted but their pattern-like nature makes them a prime candidate for the use of pattern recognition or data mining techniques. In this research the advantages and disadvantages of both hand-written and data mined protocols will be compared.

In order to collect data for this comparision I have created the Pet Shop Game. The Pet Shop Game is a web-based collective AI experiment, aimed at finding some of the obstacles encountered when applying data mining to behaviour generation. The goal of the game is to have a well-formed functional conversation between a customer and a shopkeeper in a fictional pet shop scenario. The data collected is data mined for behaviour patterns, which will be then combined into an artificial protocol. This protocol informs an AI chatbot which imitates the shopkeeper's behaviour.

1.2 Research question

The research question that will be answered in this thesis is:

> How does generating artificial, goal-oriented dialogue using automated data mining weigh up against a manual approach to this problem?

It is composed of the following subquestions:

> How useful is sequential pattern mining as a source of patterns for generating artificial, goal-oriented dialogue?
>
> What are the obstacles specific to generating artificial, goal-oriented dialogue, and how can they be overcome?

1.3 Thesis structure

Chapter 2 provides the relevant theoretical background and a review of related research, including the main source of inspiration for this work. Chapter 3 and 4 are about the research methodology and the implementation of the experiments, respectively. In Chapter 5 I present the experimental results. Chapter 6 is used for discussion and interpretation of these results. My answer to the research question and suggestions for future research are given in Chapter 7.

Chapter 2

Literature review

Before introducing the methodology to find the answer to the research question it is necessary to give an overview of related subjects.

Section 2.1 explains the terms and concepts which are relevant to the research methodology and for which we should have a single non-ambiguous definition. Afterwards, in Section 2.2, a review is made of similar research efforts and their differences with this work.

2.1 Theoretical background

This thesis combines linguistics and data mining with the aim of generating dialogues. In this section I will explain the relevant terms from these fields, and explain how they are related to this research. Sections 2.1.1 and 2.1.2 explain the linguistical background in the form of *dialogues* and *speech acts*. Technologies to generate dialogues are discussed in sections 2.1.3 and 2.1.4 about *data mining* and *production rule systems*.

2.1.1 Dialogue

The word *dialogue* can have a number of meanings. In this thesis, unless stated otherwise, dialogue should be understood to be the interchange of thoughts and information between two or more persons using a common language subject to rules of etiquette.

Nearly synonymous to dialogue is *conversation*, although conversations that arise spontaneously, undertaken for their own sake, or to while away the time are not considered to be dialogues. Examples of conversation in this sense can be found at birthday parties or on long bus trips. Given this observation one could also define dialogue as functional or goal-oriented conversation.

There are many different reasons to have a dialogue. Walton and Krabbe define six simple types of dialogue [15]. Each type is defined by the shared goal of the dialogue's participants and has its own rules of etiquette. A seventh type is mixed dialogue, which is dialogue that consists of two or more simple types. The seven types are listed in Table 2.1 on page 4.

For this thesis the focus will be on a type of mixed dialogue: *"shop" dialogue*, which potentially contains the dialogue types *negotiation* and *information-seeking dialogue* depending on the goals of participants. (See Figure 2.1.) It is the general type of dialogue when a person enters a store to purchase a product and engages in a conversation with a store-employee in pursuit of this goal.

Type	Subtypes	Initial Situation	Main Goal	Participant's Aims	Side Benefits
I Persuasion Dialogue (Critical Discussion)	Dispute Formal Discussion Discussion of Proposals	Conflicting Points of View	Resolution of Such Conflicts by Verbal Means	Persuade the Other(s)	Develop & Reveal Positions Build Up Confidence Influence Onlookers Add to Prestige
II Negotiation	Bargaining Making a Package Deal	Conflict of Interest & Need for Cooperation	Making a Deal	Get the Best out of it for Oneself	Agreement Build Up Confidence Reveal Positions Influence Onlookers Add to Prestige
III Inquiry	Scientific Research Investigation Examination	General Ignorance	Growth of Knowledge & Agreement	Find a "Proof" or Destroy One	Add to Prestige Gain Experience Raise Funds
IV Deliberation	Means-End Discussion Discussion of Ends Board Meeting	Need for Action	Reach a Decision	Influence Outcome	Agreement Develop & Reveal Positions Add to Prestige Vent Emotions
V Information-Seeking Dialogue	Expert Consultation Didactic Dialogue Interview Interrogation	Personal Ignorance	Spreading Knowledge & Revealing Positions	Gain, Pass on, Show, or Hide Personal Knowledge	Agreement Develop Position Influence Onlookers Add to Prestige Vent Emotions
VI Eristics	Eristic Discussion Quarrel	Conflict & Antagonism	Reaching a (Provisional) Accomodation in a Relationship	Strike the Other Party & Win in the Eyes of Onlookers	Develop & Reveal Positions Add to Prestige Gain Experience Amusement Vent Emotions
VII Mixed	A. Debate (Persuasion & Eristics)	Conflicting Points of View in Front of a Third Party	Accomodating Conflicting Points of View	Persuade or Influence Each Other & a Third Party	Develop & Reveal Positions Add to Prestige Amusement
	B. Committee Meeting (Mainly Deliberation)	Conflict & Antagonism & Need for Agreement in Practical Matters	Working out a Policy & Endorsing It	Influence Outcome	Agreement Build Up Confidence Develop & Reveal Positions Air Objections
	C. Socrating Dialogue (Mainly Inquiry)	Illusion of Knowledge	Healing the Soul from This Vice to Get Ready for Real Knowledge & Virtue	Refute & Avoid Being Refuted Agreement	Develop & Reveal Positions Gain Experience Amusement

Table 2.1: Overview of the six simple dialogue types by Walton & Krabbe. Also shown is a seventh 'mixed' dialogue type which combines the simple dialogue types.

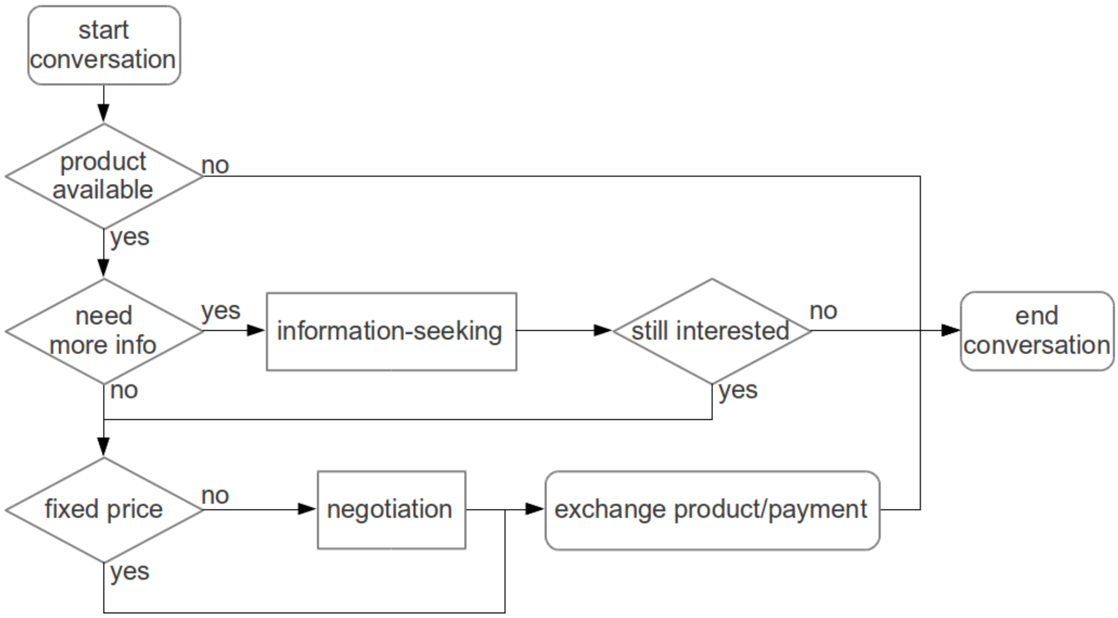

Figure 2.1: Typical flow of “shop” dialogue, showing the relationship with the simple dialogue types Information-Seeking Dialogue and Negotiation. These are represented by the straight rectangles whereas rounded rectangles represent “static” pieces of dialogue and diamonds represent choices (not a part of the actual dialogue).

Of note is the “shop” dialogue’s non-linearity—also apparent by the large number of arrows in the figure. This makes it relatively hard to capture this type of dialogue in a common script-based approach, as they tend to become quite verbose when trying to accurately represent dynamic dialogue flows.

Computer game dialogue

Computer games have a story, and a story has dialogue. This is dialogue in the literary sense of the word, as a method of exposition in a narrative. But computer games have the distinct ability, as opposed to book and film, to be interactive by bestowing the player a certain amount of agency within the game world.

This interactiveness is most commonly implemented in computer game dialogue as a *dialogue tree*[1]. (For a graphical example, see Figure 2.2.) Instead of a linear dialogue in the literary sense, a dialogue tree models a *branching dialogue*, containing multiple “paths” through a conversation with a specific NPC. This gives the player the illusion of choice, while the game’s designers keep full control over the experience. Through clever writing it is even possible to make all dialogue paths lead to a single pre-defined outcome, as if the dialogue was not branched at all.

With a dialogue tree, an in-game conversation between the player character and an NPC consists of a series of turns. On their character’s turn the player is presented with a menu containing multiple dialogue options. (For an in-game example, see Figure 2.3.) The player chooses one of the options and on their turn the NPC interlocutor will reply with the pre-written response. This continues until a node in the dialogue tree is reached where there are no more options, at which point the conversation is finished.

[1]Though in practice the term dialogue *graph* may be more appropriate.

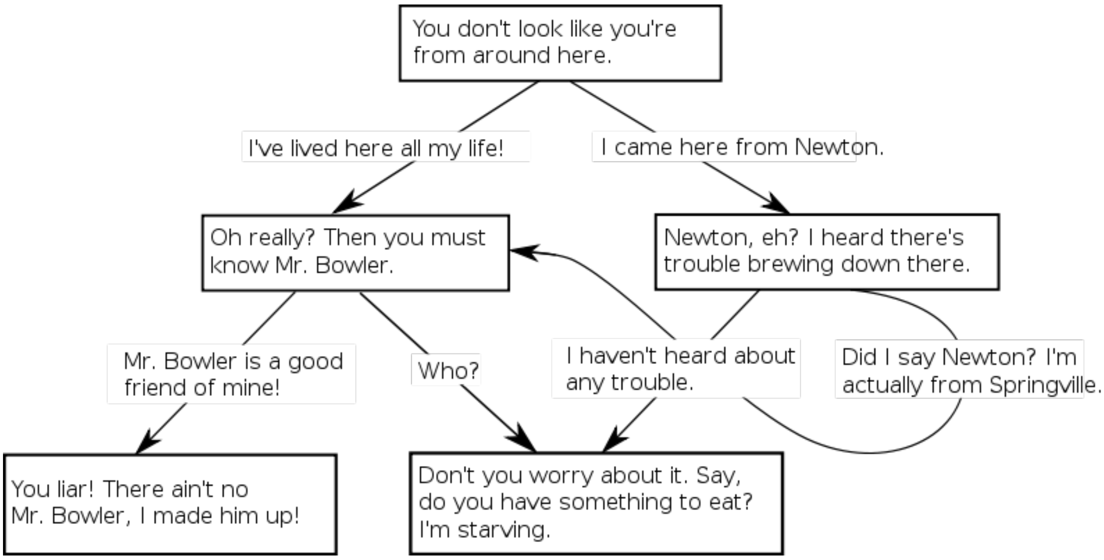

Figure 2.2: Example of a dialogue tree, representing a fictional conversation between a player character and an NPC. The arrow labels show the player's dialogue options, the NPC's pre-written responses are in the boxes. In this case there are two possible outcomes.

2.1.2 Speech acts

Dialogues are composed of utterances. For the purposes of generating goal-oriented dialogues, it is useful to view utterances as a kind of actions as described by *speech act theory*. Later on in this section the relationship between *speech acts and dialogue* is explained further.

Speech act theory

According to some early linguistic and analytic philosophers like Frege, the meaning of a sentence, like "I'll be back", should be analysed by its truth value: it can be either *true* or *false*, depending on the state of the world [4]. However, when trying to find out what it means to actually utter this sentence, the truth value alone is not enough. According to Austin [3], when that sentence is uttered it *does* something—most likely, in our earlier example, it creates a promise.

Promising is a type of act that has become known as a *speech act*. When we promise something we put an obligation on ourselves to keep our promise. In other words, apart from the physical utterance of the words, our speech act changed something in the reality. Typically this is a change in the context of a dialogue or the psychological state of its participants. Other examples of speech acts are: apologizing, declaring (marriage, war), identifying, requesting (information, action).

A classification of speech acts was given by Searle [12]. His categories include: *representatives*[2] which commit the speaker to the truth of a proposition, *directives* which try to commit the listener to a course of action, *commissives* which commit the speaker to a course of action, *expressives* which express the speaker's psychological state, and *declarations* which change reality according to a proposition. The proposition, course of action or psychological state mentioned in the previous sentence are usually referred to as an utterance's *propositional content*. This propositional content can be seen as the speech act's subject.

[2]Sometimes referred to as *assertives*.

Figure 2.3: Guybrush Threepwood (the player character) in dialogue with the Voodoo Lady in *The Curse of Monkey Island* by LucasArts. The player's current four dialogue options are shown.

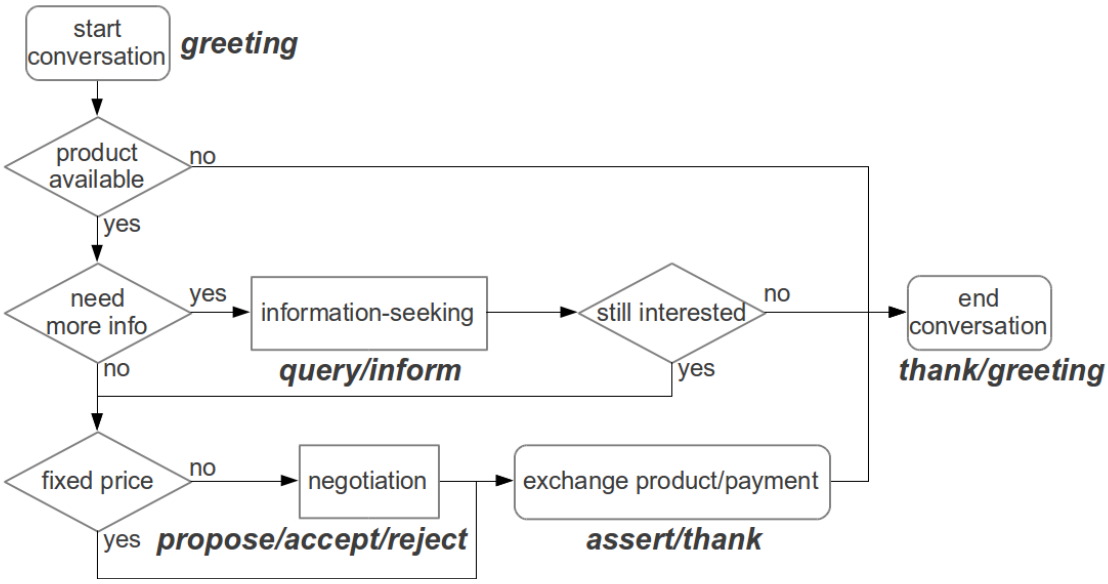

Figure 2.4: The "shop" dialogue diagram from Figure 2.1 annotated with typical speech acts for each of its parts.

Speech acts & dialogue

Dialogues, especially goal-oriented dialogues as described by Walton & Krabbe, require a conversation to "go somewhere". Not only does this mean that when the conversation is over something tangible needs to be accomplished, but it also means that certain "steps" need to be taken as the dialogue continues. When analysing the sentences that compose a dialogue purely by their truth-value, no progress or change in dialogue state can be discerned. However, treating utterances within a dialogue as speech acts provides a framework for the structure of a dialogue. (For an example, see Figure 2.4.)

For example, promising and requesting are speech acts which define the negotiation dialogue type, but not only their use, but also the order in which they are used is relevant to having a proper negotiation. An offer should either be followed by an utterance representing a counter-offer, an acceptance, or a backout, regardless of its propositional content. If this demand is not met the dialogue is no longer a negotiation.

It is easier to focus on a dialogue's goal when treating it on the higher level of speech acts instead of the semantics of a dialogue's concrete utterances. As such the speech act can be seen as the smallest unit of intention within a dialogue.

Furthermore, by using speech acts as an abstraction of dialogue utterance we gain a computational advantage which is favourable when trying to implement a functional dialogue system.

In the previous part of this thesis' theoretical background it was proposed to look at *dialogues* as a sequence of *speech acts*. We can assume this will put more of a focus on the goal-orientedness of dialogues and that this will make generating dialogues easier than an approach which analyses dialogue utterances at the semantic level.

The next section introduces *data mining* as a way of finding patterns in data. A specific instance of data mining, sequential pattern mining, finds patterns in sequential data. These

sequence-id	time	itemset
Customer #1	1	Braid
	2	Uplink
	15	The Witness, Prison Architect
Customer #2	1	Uplink, Braid
	20	DEFCON
	50	The Witness
Customer #3	5	Uplink, Braid
	15	Prison Architect
	20	The Witness

Table 2.2: Sequential database containing data sequences for three customers. All of them made three separate store visits.

patterns can then be turned into rules which make predictions about novel data. The theoretical background concludes with an explanation of *production rule systems*, a rule-based formalism which I will use as a benchmark for a qualitative comparison between a data-mined chatbot and a hand-written chatbot.

2.1.3 Data mining

Given a large dataset, it can be of interest to extract more knowledge about it in the form of novel patterns within the data. Examples of interesting patterns are groups of similar data, data anomalies, and dependencies between data. So-called *data mining* techniques are typically based on statistical and machine learning methods.

In the next section we will look at *sequential pattern mining*, which is used to find frequently occurring patterns in sequential data, like dialogues.

Sequential pattern mining

Sequential pattern mining is a data mining field concerned with finding *sequential patterns* in sequential itemset data. A canonical example of this type of data is store purchase data. Every customer's purchases across multiple store visits are tracked to form a data sequence consisting of multiple itemsets (i.e. a specific combination of bought products).

A *sequence* s is denoted $\langle s_1 s_2 \dots s_n \rangle$ where s_j is an itemset, which is denoted $(i_1, i_2, \dots, i_m)$ where i_j is an item. An item can occur only once in an itemset of a sequence, but can occur multiple times in different itemsets.

Sequential patterns are defined as sequences that are contained in or *supported by* a certain minimum number of data sequences. For example, in Table 2.2, the following sequences have a support count higher than one:

$$\langle(\text{Braid})(\text{The Witness})\rangle \quad \langle(\text{Uplink})(\text{The Witness})\rangle \quad \langle(\text{Uplink, Braid})(\text{The Witness})\rangle$$
$$\langle(\text{Uplink, Braid})\rangle \quad \langle(\text{Braid})(\text{Prison Architect})\rangle \quad \langle(\text{Uplink})(\text{Prison Architect})\rangle$$

It is possible to generate *sequential rules* based on these patterns. Since we are interested in predicting the future[3], only *forward rules*, rules with a conclusion in the future, are of interest.

[3]Predicting the past falls outside the scope of this thesis.

Two examples of forward sequential rules (and their confidence values) are:

$$\langle(\text{Uplink, Braid})\rangle \rightarrow \langle(\text{The Witness})\rangle \qquad (conf = 1.0)$$
$$\langle(\text{Braid})\rangle \rightarrow \langle(\text{Prison Architect})\rangle \qquad (conf = 0.66)$$

These rules say that a customer who has bought both Uplink and Braid in one store visit, it is one hundred percent certain they will buy The Witness, and when they have bought Braid they have a two out of three chance to buy Prison Architect.

A dialogue is a time-ordered sequence of dialogue turns, where each turn is based on the ones that precede it. This makes dialogue data great for sequential pattern matching to find sequential patterns of speech acts and utterances. These patterns can then be automatically transformed into forward rules for dialogue generation.

2.1.4 Production rule systems

Rules of the form $X \rightarrow Y$, like the rules found by sequential pattern mining, are called *production rules*. Production rules consist of two parts, a precondition and an action. Whenever the precondition is evaluated positively, the rule is said to be triggered and its action is executed. Production rule systems are used in general to generate behaviour, by using the rules to map beliefs to actions.

In the case of using sequential rules to generate a dialogue, the rule's precondition is evaluated by checking whether its sequence of dialogue turns is contained in the preceding dialogue, and the action should be seen as a prediction of a dialogue which should occur. We will see most chatbots can be seen as production rule systems as well, using rules to transform input sentences into replies. By treating both data mined chatbots and hand-written chatbots as production rule systems, it will be possible to make a qualitative comparison between both approaches.

2.2 Related work

This research is not the only effort towards generating artificial dialogue. *Traditional chatbots* and *QA-systems* are aimed at creating specific dialogue types, but the repertoire of an individual chatbot or QA-system is typically hand-written—a labour-intensive process. A different approach is taken by the main inspiration for this work: Jeff Orkin's *Collective Artificial Intelligence*. In this section these efforts and their differences with this thesis' approach are reviewed and explained.

2.2.1 Traditional chatbots

A chatbot is a computer program aimed at having a convincing conversation with human users. This means they have to handle dialogue that is properly open-ended, which means the search space for proper utterances given a certain dialogue state can be very large. As a way to cope, these conversations typically involve *small talk*, or conversation for the sake of conversation, as these less formal types of dialogue leave the most room for error. In this section I will mention two chatbot systems, one of the very earliest: *ELIZA*, and one of the more recent iterations: *ChatScript*.

ELIZA

Natural language processing, and by extension artificial dialogue, is said to have started as a field of research with ELIZA [16], a computer program pretending to engage in conversation with its user. A human would interact with ELIZA by typing a question or a statement, and ELIZA would offer a transformed version of the input as a reply. (For an example, see Excerpt 1.) Using a specific set of input rules called DOCTOR, it was able to parody a stereotypical psychiatrist. Curiously, even after being told how it worked, its users took ELIZA very seriously. This became known as the *ELIZA-effect*.

```
HUMAN: You are like my father in some ways.
ELIZA: What resemblance do you see?
HUMAN: You are not very aggressive but I think you don't want me to
       notice that.
ELIZA: What makes you think I am not very aggressive?
HUMAN: You don't argue with me.
ELIZA: Why do you think I don't argue with you?
HUMAN: You are afraid of me.
ELIZA: Does it please you to believe I am afraid of you?
HUMAN: My father is afraid of everybody.
ELIZA: What else comes to mind when you think of your father?
HUMAN: Bullies.
```

Excerpt 1: Excerpt from an ELIZA log.

Despite being intended as a parody, ELIZA was very influential. Both in popular culture and in the development of similar technologies, which picked up the rule-based hand-written approach. A lot of chatbots followed in its wake.

ChatScript

ChatScript is a newer generation chatbot system that won the 2010 and 2011 editions of the Loebner Prize[4] and finished second in 2012. Like ELIZA, ChatScript uses hand-written rules, but they are specified in a very concise scripting language. It employs powerful pattern matching and *concepts* to create synonyms of words based on WordNet[5] entries. On a higher level, the ChatScript engine vetoes repetitive rules from executing and has support for basic emotional states like boredom, anger, and delight which are triggered by the interlocutor's behaviour. An example of ChatScript code can be seen in Excerpt 2.

Chatbots use sequential rules to generate their utterances based on earlier dialogue turns. As opposed to the effort of this thesis, these rules are hand-written, which is a lot of work. Therefore the hand-written approach is typically employed to generate ephemeral conversation or chit-chat, where global coherence is less important and rules can be as simple as mapping the previous dialogue turn to the next, as is done in ELIZA. Data mined rules for dialogues can be a lot more complex, in the sense that it is easier and more efficient for a computer to generate rules which have preconditions that can be satisfied by dialogue turns as early as the start of the conversation.

[4] "An annual competition in artificial intelligence that awards prizes to the chatterbot considered by the judges to be the most human-like. The format of the competition is that of a standard Turing test." – `http://en.wikipedia.org/wiki/Loebner_Prize`
The competition's website can be found on `http://www.loebner.net/Prizef/loebner-prize.html`.

[5] "WordNet® is a large lexical database of English. Nouns, verbs, adjectives and adverbs are grouped into sets of cognitive synonyms (synsets), each expressing a distinct concept." – `http://wordnet.princeton.edu/`

```
concept: ~buildings [ shelter~1 living_accomodations~1 building~3 ]

s: ( I like spinach ) Are you a fan of the Popeye cartoons?
    a: ( yes ) I used to watch him as a child. Did you lust after Olive Oyl?
         b: ( no ) Me neither. She was too skinny.
         b: ( yes ) You probably like skinny models.
    a: ( no )  What cartoons do you watch?
         b: ( none ) You lead a deprived life.
         b: ( Mickey Mouse ) The Disney icon

topic: ~DEATH [dead corpse death die body]
t: I don't want to die
?: (When will you die) I don't know..
```

Excerpt 2: Excerpt from a ChatScript script showing concepts derived from WordNet, rejoinders, and a topic definition with a topic gambit and a topic-specific rule.

2.2.2 Question–answer systems

Question–answer (QA) systems focus on one specific type of dialogue: information-seeking dialogue. A user will pose a question in natural language and the system tries to find the most likely answer by querying a knowledge base or perusing natural language corpora like reference texts or web pages.

QA systems come in two main types. *Closed-domain* QA systems draw knowledge from a specific domain. Examples of this type include ELIZA's DOCTOR script, SHRDLU—which operated on virtual blocks in a toy world and is able to answer questions about the state of that world, and ANNA—a virtual assistent which is featured on the Ikea webpage[6] for several countries. The other type is *open-domain* QA systems, which show some overlap with chatbots because they deal with questions about nearly anything. These systems have to rely on general ontologies and world knowledge. Examples include the answer engine Wolfram Alpha, and, in a very rudimentary form, the Google search engine.

2.2.3 Collective Artificial Intelligence

Jeff Orkin has been applying a crowd sourced approach to the task of creating an autonomous agent for a special-purpose scenario in The Restaurant Game Project. The scenario has two roles, a customer and a waiter, and takes place in a restaurant environment containing around 40 objects. Players are able to explore and interact with the 3D restaurant environment. All of the player's actions are logged and subsequently used by Orkin to create a game-playing agent able to perform one of the roles in the scenario. [7]

In a bid to exploit the intentionality of goal-oriented restaurant dialogue, Orkin abstracts away from natural language by using a construct called a *dialogue act*. A dialogue act is a 3-tuple consisting of an utterance's "speech act" (inspired by Searle's taxonomy), "content" and "referent". Orkin refers to these dialogue acts as a "common currency" between physical actions and dialogue utterances. To recognize dialogue acts in natural language dialogue he has trained a classifier. [8]

[6] See, e.g., `http://www.ikea.com/nl/`.

We will use a *controlled natural language* instead of open-ended natural language in an attempt to side-step this difficult pre-processing step. The assumption is made that utterances in the controlled natural language can be seen as directly representing their corresponding speech act, and that small changes in utterance surface structure result in utterances that have a similar associated speech act. For example, when comparing “I would like to buy a cat” and “I would like to buy a dog.” it is merely the propositional content that changes, other speech act properties remain the same.

Two drawbacks of Orkin’s approach at this point are the absence of higher-level control and the system’s inability to produce chat text that does not appear (often) in the traces but is necessary for proper dialogue. So Orkin’s newer work ([9], [10]) focuses on enabling higher-level control, using semi-automated tagging of groups of actions to arrive at a hierarchical representation of the behaviour. These groups of actions, which represent higher-level behaviours like taking an order or cleaning the table, can be more purposely fitted into a generated conversation.

In my research sequential pattern mining is used to discover sequential patterns in behavioural data. These recurring behaviourial patterns provide the same benefits as Orkin’s hierarchical grouping of actions, but unlike Orkin’s work I generate them fully automatically.

Chapter 3

Methodology

In this chapter I will state the *research hypothesis* and give an explanation of the *experimental approach* to either confirming or denying that hypothesis. In the next chapter I will describe the implementation of the experimental approach in more detail.

3.1 Research hypothesis

To answer the research question and research subquestions posed in Chapter 1, I will compare a dialogue generation system based on data mining with a hand-written dialogue system. Specifically I will generate "shop" dialogue, because the shop scenario is commonplace and has clear goals, and sequential pattern mining, because it is well-suited to finding patterns in dialogue data. The patterns will be used to generate rules which will be qualitatively analysed for their applicability within a system which generates goal-oriented shop dialogue. Furthermore, a comparison between such a hypothetical system and a hand-written system is made.

The research hypothesis is stated as follows.

Hypothesis It is possible to generate a goal-oriented shop dialogue by using rules data mined from dialogue data using sequential pattern mining that compare well to the rules used in a hand-written dialogue generation system.

The experimental set-up to test this hypothesis will be explained in the rest of this chapter.

3.2 Experimental approach

To collect "shop" dialogue data I will implement *The Pet Shop Game*. This is an online application which allows players to take on the roles of customer and shopkeeper in a pet shop scenario. The dialogue data that is collected using The Pet Shop Game will be data mined for behavioural patterns using a sequential pattern mining algorithm. These patterns are then used to create rules for use by the *data mined chatbot*, a computer program which is able to play the role of the shopkeeper in the Pet Shop Game.

3.2.1 The Pet Shop Game

The Pet Shop Game will be an on-line application which anonymously connects two players to play a dialogue game. In the game, one player is cast in the role of a customer in a *pet shop scenario*, and the other in the role of the shopkeeper.

An overview of a Pet Shop Game session is listed below.

1. Two players are connected by the Pet Shop Game and are randomly assigned a role.
2. Both players are shown a *background story*.
3. They play a *dialogue game*.
4. The players are asked to fill out an *evaluation* about the dialogue game.
5. A player can then choose to play again (returning to Step 1) or to stop, in which case they are asked to fill out an extra evaluation about the entire session.

The pet shop scenario

The scenario tries to convey a typical Western-European pet shop. However, for the sake of simplicity, its stock only consists of a limited collection of animals: a cat, a dog, a goldfish, a hamster and a parrot. And, for the same reason, the customer is limited to buying only one animal.

Background story

A short, role-specific background story is shown to both players at the start of a game. This is to prime them for role-playing during the game, and to induce variation in the dialogue should one player choose to play multiple games.

While there is some thematic relationship between the stories, the stories are independently selected.

Customer At the start of the game, the customer is shown one of the five random background stories listed below.

- "You are a parent of twins and their 8$^{\text{th}}$ birthday is coming up. The children have been asking you for a cute pet to take care of for months now, but you are not sure if they will be able to take care of it everyday."
- "A friend of yours recently broke up with his girlfriend whom he has been with for over a year. He must feel lonely, and while in town looking for a present to cheer him up you stumbled across this pet shop."
- "Ever since you were young you dreamt of breeding animals, and despite your best efforts, keeping mice in the basement was not met with much enthusiasm by your parents. Now you have finally moved to a place of your own with plenty of space to make your wish come true. Now to find a couple of animals to breed."
- "Yesterday burglars broke into the house next door. You are not easily scared, but it would be nice to have a faithful companion to stand watch during the night, to act as a warning signal or even to deter burglars from trying to get in."

- "You had a very serious conversation with your partner last week. In the end it came down to the question when you wanted to start having children. You decided to go to the pet shop and get a cute baby animal which might temper their desires a little bit, for now."

Shopkeeper The shopkeeper is always instructed to assist the customer to the best of his or her ability, but the background story is extended with an exceptional situation. The default background story and its four possible extensions are listed below.

- "As the shopkeep, your main job is to keep the customer satisfied, and preferably get them to come back to our shop. ...
  - ... However, today you had a big delivery of hamsters. It is in your (and the animal's) best interest to sell them as quickly as possible."
  - ... All parrots were sold out this week, and no new delivery has been made. So you might have to disappoint any bird-lovers that enter the store today."
  - ... A reviewer for a big pet shop magazine is in town, be sure to treat them with respect when you see them."
  - ... Illegal breeding is on the rise. Animals are bought from pet shops and used for breeding to gain as much profit without much regard for the living conditions."

Dialogue game

After both players have seen the background story, one of them is randomly selected to be the starting player. The players then engage in a turn-based dialogue game, where each turn they can perform multiple utterances in a *controlled natural language* or perform a single *physical action*. The utterances and actions that can be performed are role-specific.

The game ends when the customer leaves the shop, which is only possible when the customer paid for an animal and the shopkeeper gave an animal to the customer or when neither of those physical actions took place.

Controlled natural language To keep things simple and to focus on more interesting parts of this problem the decision was made to steer clear of the use of free natural language input. This will be achieved by providing the players with a controlled natural language, which, while simple, should also allow the players enough freedom to communicate as much as necessary for the typical pet shop scenario.

In the Pet Shop Game players will be able to select parts of speech based on a pre-defined controlled natural language grammar to "click together" their utterances.

Physical actions On his or her turn a player will be able to perform one action instead of performing one or more utterances. The shopkeeper is able to **give** an animal to the customer and the customer can either **pay** the shopkeeper or **leave** the shop.

To keep things simple two limitations were added to the use of the physical actions: each physical action can be performed only once per game and one cannot select a specific animal to give or an amount of money to pay. In the case of the latter limitation it is assumed that players' imagination will fill in the blanks with respect to the actual dialogue context.

Player evaluation

At the end of each game the players have to fill in a short evaluation about that game, answering questions about the role-playing performance of their co-player and themselves.

Afterwards the player can choose to either play a new game with someone else, or stop playing altogether. In the latter case they are asked to answer additional evaluation questions about their entire playing session.

3.2.2 Data mining behavioural patterns

After the data from the Pet Shop Game is collected, it will be used as input for my own implementation of *GSP*, an algorithm for sequential pattern mining. The output of this algorithm is a set of recurring behavioural patterns in the Pet Shop Game dialogue. These patterns are converted into rules which will be used in the prediction process as described below in Section 3.2.3.

The GSP algorithm

GSP [14] or *Generalized Sequential Patterns* is a widely used sequence mining algorithm that finds patterns in sequential data. To apply this algorithm to the dialogue data that results from the Pet Shop Game, each dialogue in the dataset is transformed into a *data-sequence*—a sequence of *itemsets*, items being individual utterances. This set of data-sequences is then combined with a *taxonomy*—modeling an is-a relationship of sequence items (i.c., utterances) to higher-level items (i.c., speech acts)—to form the main input to the algorithm.

Data-sequences The input for the GSP algorithm consists of a database containing data-sequences each of which consists of an ordered set of itemsets. The algorithm finds sequences of itemsets that have *minimum support*: sequences which are "contained" by a specified minimum number of data-sequences. For now, it is sufficient to define containment as being a subsequence. A sequence s is a subsequence of a data-sequence d if all itemsets of s occur as subsets of itemsets of d, and these supersets in d occur in the same order as their corresponding subsets from s.

A collected dialogue from the Pet Shop Game consists of dialogue turns, each containing multiple utterances. Since we want to find behavioural patterns which describe relationships between utterances, we need to model an utterance as an item, a dialogue turn as an itemset, and a dialogue as a data-sequence. Because they give important clues about the state of the dialogue, the player's actions are also included in the data-sequence as separate itemsets containing the action as a single item. To differentiate between shopkeeper utterances or actions and customer utterances or actions all are marked with either S or C.

For an example of a mapping from a dialogue to a data-sequence, see Table 3.1.

Taxonomy Given a minimum support value of 2 the data-sequences in Table 3.1 support seven sequences:

$$\langle(\text{C:“Hello.”})\,(\text{S:“Yes.”})\rangle \qquad \langle(\text{C:“Do you sell goldfish?”})\,(\text{S:“Yes.”})\rangle$$
$$\langle(\text{C:“Hello.”},\text{C:“Do you sell goldfish?”})\rangle \quad \langle(\text{C:“Hello.”},\text{C:“Do you sell goldfish?”})\,(\text{S:“Yes.”})\rangle$$
$$\langle(\text{S:“Hello.”})\,(\text{C:“Hello.”})\rangle \qquad \langle(\text{S:“Hello.”})\,(\text{C:“Hello.”})\,(\text{S:“Yes.”})\rangle$$
$$\langle(\text{S:“Hello.”})\,(\text{S:“Yes.”})\rangle$$

Conv.	Role	Dialogue turn	Itemset for GSP
#1	Customer	"Hello." "Do you sell goldfish?"	(C:"Hello.", C:"Do you sell goldfish?")
	Shopkeeper	"Welcome." "Yes." "We sell goldfish."	(S:"Welcome."S:"Yes."S:"We sell goldfish.")
#2	Shopkeeper	"Hello." "How may I help you?"	(S:"Hello."S:"How may I help you?")
	Customer	"Hello." "Do you sell goldfish?"	(C:"Hello."C:"Do you sell goldfish?")
	Shopkeeper	"Yes."	(S:"Yes.")
#3	Customer	enters	(C:"enter_action")
	Shopkeeper	"Hello."	(S:"Hello.")
	Customer	"Hello." "Do you sell dog?"	(C:"Hello."C:"Do you sell dog?")
	Shopkeeper	"Yes."	(S:"Yes.")

Table 3.1: Three partial dialogues with the itemsets corresponding to the dialogue turns in the fourth column. The ordered set of itemsets for each dialogue is its corresponding data-sequence.

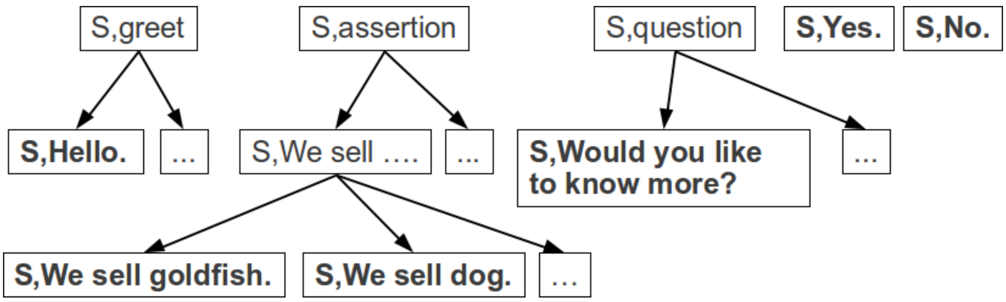

Figure 3.1: Partial taxonomy for the shopkeeper's utterances. The latter correspond to the leaves of the trees, drawn in boldface. This hierarchy is based on The Pet Shop Game's constrained natural language grammar.

This is already quite interesting but there are more patterns to be found which are not directly found by GSP. For example, the shopkeeper providing an answer to a question anytime (regardless of the question's content).

To enable the system to pick up on this question–answer pattern we use a taxonomy which maps sequence items to higher-level items according to an *is-a* relationship. In this case we create a mapping from utterances to their corresponding speech acts. This taxonomy is based on the grammar of the controlled natural language of the Pet Shop Game. A condensed example can be seen in Figure 3.1. When this taxonomy is applied to the first dialogue in Table 3.1, additional items will be added to the data-sequence's itemsets, as can be seen in Table 3.2.

The advantage of using a taxonomy in general is that more interesting patterns will be found. This also seems the case for this specific application. For example, these two additional sequential patterns can be found when a taxonomy is used for our previous example:

$$\langle(\text{C:“Do you sell}\ldots\text{?”})\,(\text{S:“Yes.”})\rangle \qquad \langle(\text{C:“Hello.”}, \text{C:“question”})\rangle$$

3.2.3 Data mined chatbot

The data mined patterns will be converted into dialogue rules of the form $A \rightarrow B$ to be used by a *data mined chatbot* to generate correct dialogue game turns for the shopkeeper role in The Pet Shop Game.

The data mined chatbot will use the dialogue rules to decide its next dialogue turn based on the current dialogue's history. While playing a game the chatbot finds all the rules of which the

Role	Dialogue turn	Extended itemset based on grammar taxonomy
Customer	"Hello." "Do you sell goldfish?"	(C: "Hello.", C: "Do you sell goldfish?", C: "greeting", C: "Do you sell ... ?", C: "question")
Shopkeeper	"Welcome." "Yes." "We sell goldfish."	(S: "Welcome." S: "Yes." S: "We sell goldfish.", S: "greeting", S: "We sell ... ?", S: "assertion")

Table 3.2: Two dialogue turns translated to input for the GSP algorithm. The third column shows the itemsets when using the constrained natural language grammar as the taxonomy-input for the algorithm.

condition matches the dialogue history, and decides what the most likely next turn is based on the matching rules' conclusion.

As an example, consider a customer who has just entered the store and said "Hello.". The data mined chatbot then tries to match its dialogue rules to this short dialogue history. Examples of matching rules could be:

$$\langle(\text{C: "Hello."})\rangle \rightarrow \langle(\text{S: "Yes."})\rangle \qquad \langle(\text{C: "Hello."})\rangle \rightarrow \langle(\text{S: "How may I help you?"})\rangle$$
$$\langle(\text{C: "Hello."})\rangle \rightarrow \langle(\text{S: "Welcome."})\rangle \qquad \langle(\text{C: "greeting"})\rangle \rightarrow \langle(\text{S: "Welcome."})\rangle$$
$$\langle(\text{C: "greeting"})\rangle \rightarrow \langle(\text{S: "greeting.", S: "How may I help you?"})\rangle$$

Based on this set of rules it is most likely that the shopkeeper's response is the dialogue turn corresponding to the itemset (S: "Welcome.", S: "How may I help you?"): "Welcome. How may I help you?"

This is a very different approach from the common way of creating a goal-oriented dialogue system. The main part of the process—finding and formalizing behavioural patterns—is completely automated, whereas mainstream methods rely heavily on hand-crafting in this respect. These methods might overlook relevant patterns despite being created by domain experts, whereas my approach leans on untrained people for generating data, which is then mined for all of its interesting patterns.

Chapter 4

Implementation

This chapter describes the implementation of the *Pet Shop Game* web application used to collect dialogue data for the pet shop scenario, the *GSP implementation* that is used to data mine the collected data for interesting patterns, and the implementation of the *data mined chatbot* which is used to evaluate the usefulness the patterns that were found by data mining.

4.1 The Pet Shop Game

The Pet Shop Game is implemented as a *web application*, where players are coupled to play one or more dialogue games using a *controlled natural language*. The Pet Shop Game's source code will be made available online via Bitbucket[1].

4.1.1 Web application

The Pet Shop Game is implemented as a PHP/MySQL/jQuery[2] web application and was designed to function in as many web browsers as possible. For full duplex client–server communication a combination of Ajax [5] and Comet [11] was used.

For the human interaction with the game a chat-room-like interface was created. It shows a dialogue history and has buttons to perform the role-specific actions and form utterances according to the controlled natural language grammar. See Figure 4.1 for a typical view.

4.1.2 Controlled natural language

The controlled natural language was modelled as a grammar as shown in Figures 4.2, 4.3, and 4.4. On a player's turn they are presented with an utterance template and given the choice to substitute it for an expression as defined by the grammar, where an expression might contain a new template for which new choices are presented. This continues until all non-terminals are substituted at which point the player may speak the utterance, or expand their turn with subsequent utterances.

As an example, on their turn a shopkeeper player might choose "greeting" (an instance of "utterance"), at which point they are given a choice between "Hello.", "Goodbye.", "Thank you.", "I am sorry.", and, from the shopkeeper-specific grammar, "Welcome", and "At your service.".

[1] `https://bitbucket.org/broersma/thesis`

[2] See `http://www.php.net/`, `http://www.mysql.com/`, and `http://www.jquery.com/` respectively.

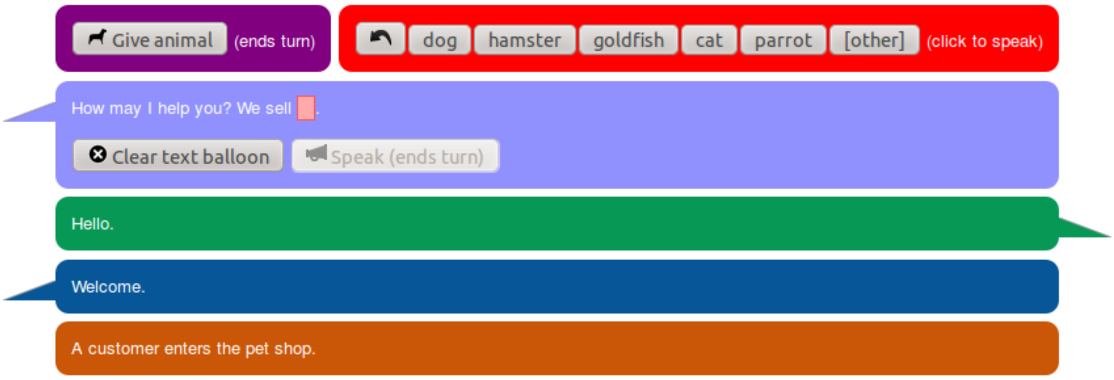

Figure 4.1: The interface of the Pet Shop Game. On top are possible actions (purple area) and dialogue options (red). The dialogue history (green, dark blue, orange) is at the bottom, with the most recent message on top. We are looking through the eyes of the shopkeeper player and it is their turn. They have just formed the utterance "How may I help you?" (light blue area) and are about to finish the utterance "We sell ...."

Choosing one of these six options enables the "Speak"-button, as no more non-terminals remain, and the "Add utterance"-button, which, if clicked, adds another utterance template.

To prevent players from having difficulty with expressing what they want to express, the "other" option was implemented for certain non-terminals. This allows players to supply a value of their own choosing instead of one of the pre-written ones.

4.2 GSP implementation

To perform the sequential pattern generation I have implemented the GSP algorithm [14] in Python[3]. The GSP implementation's source code will be made available online via Bitbucket[4].

The GSP algorithm's input parameters and their experimental values are as follows:

- A database of data sequences—adapted from the Pet Shop Game's collected dialogues according to the method described in Section 3.2.2.
- A taxonomy—a directed acyclic graph based on the Pet Shop Game's controlled natural language grammar.
- A minimum support count—varied for fixed intervals between 50% to 10% of the number of data-sequences.
- A maximum gap size—varied between 1, 3, and 5, representing the possibility to skip a maximum of 0, 2 and 4 dialogue turns, respectively, between itemsets in a sequential pattern.

4.3 Sequential rules

Sequential dialogue rules are generated from the sequential patterns found by the GSP algorithm. These dialogue rules predict the next dialogue turn for a dialogue based on its preceding history.

[3] http://www.python.org/
[4] https://bitbucket.org/broersma/thesis

$\langle$utterance$\rangle$ $\models$ $\langle$assertion$\rangle$ | $\langle$question$\rangle$ | $\langle$greeting$\rangle$ | Yes. | No. |
Yes, $\langle$reason_to_say_yes$\rangle$. | No, $\langle$reason_to_say_no$\rangle$.
$\langle$assertion$\rangle$ $\models$ That's good. | I'm not sure. | I don't understand.
$\langle$question$\rangle$ $\models$ What do you mean?
$\langle$greeting$\rangle$ $\models$ Hello. | Goodbye. | Thank you. | I am sorry.
$\langle$reason_to_say_yes$\rangle$ $\models$ that's possible | that is okay
$\langle$reason_to_say_no$\rangle$ $\models$ that's not possible | that is not okay
$\langle$type_of_animal$\rangle$ $\models$ cat | dog | goldfish | hamster | parrot
$\langle$sum_of_money$\rangle$ $\models$ 5 | 10 | 15 | 20
$\langle$pet_property$\rangle$ $\models$ big | small | friendly | alert | cheap in maintenance |
faithful | talkative | pettable | a good breed

Figure 4.2: General grammar of the controlled natural language. This part of the grammar is shared between both roles.

$\langle$assertion$\rangle$ $\models$ That is too expensive.
$\langle$question$\rangle$ $\models$ What pets do you sell? |
Can you tell me if a $\langle$type_of_animal$\rangle$ is $\langle$pet_property$\rangle$? |
Do you sell $\langle$type_of_animal$\rangle$? |
How much does a $\langle$type_of_animal$\rangle$ cost? |
What can you tell me about $\langle$type_of_animal$\rangle$? |
I would like to buy a $\langle$type_of_animal$\rangle$. |
Which pet is $\langle$pet_property$\rangle$? |
Give me the $\langle$type_of_animal$\rangle$, please.
$\langle$reason_to_say_yes$\rangle$ $\models$ I need to know more
$\langle$reason_to_say_no$\rangle$ $\models$ I know enough

Figure 4.3: Customer grammar of the controlled natural language. This grammar extends the general grammar from Figure 4.2.

$$
\begin{array}{rcl}
\langle \text{assertion} \rangle & \models & \text{We sell } \langle \text{type\_of\_animal} \rangle. \mid \text{We don't have } \langle \text{type\_of\_animal} \rangle \text{ in stock.} \mid \\
& & \text{A } \langle \text{type\_of\_animal} \rangle \text{ costs } \langle \text{sum\_of\_money} \rangle \text{ euro.} \mid \\
& & \text{A } \langle \text{type\_of\_animal} \rangle \text{ is a pet that is } \langle \text{pet\_property} \rangle. \\
\langle \text{question} \rangle & \models & \text{How may I help you?} \mid \text{Would you like to know more?} \\
\langle \text{greeting} \rangle & \models & \text{Welcome.} \mid \text{At your service.}
\end{array}
$$

Figure 4.4: Shopkeeper grammar of the controlled natural language. This grammar extends the general grammar from Figure 4.2.

The sequential rule generation method is based on *association rule generation* [1] where rules are filtered based on a confidence measure, with the addition that we are only interested in "forward rules": rules that draw conclusions about the future.

4.3.1 Sequential rule generation

In order to generate sequential rules we modify an existing method used in association rule learning. Association rules represent co-occurrence patterns of items in *frequent itemsets* within a database. Frequent itemsets are itemsets that meet a minimum support value (*min_sup*), like the sequential patterns found in sequential pattern mining.

The method for association rule generation is as follows. From a frequent itemset A and all its non-empty strict subsets we get the set of potential rules $\{A \setminus B \to B : B \subset A \wedge B \neq \emptyset\}$ and this set is filtered based on a minimum confidence value (min_conf). The confidence of a rule $X \to Y$ is defined as:

$$
conf(X \to Y) = \frac{supp(X \cup Y)}{supp(X)}
$$

Unlike sequential pattern mining the ordering between items is irrelevant for association rule learning. To preserve ordering when generating sequential rules from sequential patterns the algorithm above is modified as follows.

Given a sequence s denoted $\langle s_1 s_2 \ldots s_n \rangle$ where s_j is an itemset, and a minimum confidence value min_conf, we define the set of sequential rules as:

$$
\left\{ \langle s_1 \ldots s_x \rangle \to \langle s_{x+1} \ldots s_n \rangle : x \in \{1 \ldots n-1\} \wedge \frac{supp(\langle s_1 \ldots s_x \rangle \sqcup \langle s_{x+1} \ldots s_n \rangle)}{supp(\langle s_1 \ldots s_x \rangle)} \geq min\_conf \right\}
$$

The $\sqcup$ denotes the concatenation operator for sequences. Given two sequences s and t, $s \sqcup t$ is equal to the sequence $\langle s_1 \ldots s_n t_1 \ldots t_n \rangle$.

For our shopkeeper bot we are looking for specific rules, namely ones that have are useful for a bot which takes over the role of a shopkeeper in the Pet Shop Game. So in this experiment we will not only filter the rules based on their confidence value, but also whether their conclusion consists of a concrete dialogue turn for the shopkeeper role. This means the dialogue turn in the conclusion of the rule should only contain utterances which are leaves of the taxonomy, and those utterances should be part of the shopkeeper grammar.

4.4 Data mined chatbot prediction procedure

The data mined chatbot will be based on a procedure which predicts, based on the sequential rules found in the Pet Shop Game dataset, how a certain dialogue history should continue. The chatbot can then perform further filtering to come to a decision of the action to apply.

The pseudocode for this procedure is given in Procedure 1.

Procedure 1 The *prediction* procedure.

Input: A data-sequence *history*, a rule-set *rules*, a max gap size *max_gap*, a flag *use_specificity*.

Output: A set of *rule-actions* sorted by their maximum confidence value, or, if *use_specificity* = *true*, first by their specificity (i.e., $length(rule_{condition})$) and then their maximum confidence value.

$rules \leftarrow matching\_rules(history, rules, max\_gap)$ ▷ See Procedure 2.
if *use_specificity* **then**
  yield all $rule_{action} \in sort(rules$ **by** $length(rule_{condition})$ **then** $rule_{confidence})$
else
  yield all $rule_{action} \in sort(rules$ **by** $rule_{confidence})$

Procedure 2 The *matching_rules* procedure.

Input: A data-sequence *history*, a rule-set *rules*, a max gap size *max_gap*.

Output: A subset of *rules* matching *history* given *max_gap*.

for all $rule \in rules$ **do**
  $history\_suffix \leftarrow suffix(history, length(rule_{condition}) \cdot max\_gap)$
  if $check\_candidates(history\_suffix, rule_{condition}, max\_gap)$ **then**
    yield *rule*

Given a certain dialogue history the procedure first matches rules according to GSP's *check_candidates* procedure [14], keeping in account the maximum gap size parameter used to generate the rules. The actions of the matching rules are sorted by their confidence value and returned as a list, with the first action being the one generated by the rule with the highest confidence value. If the *use_specificity* flag is set, the actions will be first ordered by the specificity of their rule (the length of its condition) and then by the rule's confidence value.

In the next chapter this algorithm is used to apply rulesets found by varying parameter combinations to the Pet Shop Game dataset to measure the effectiveness of the rules.

Chapter 5

Results

In this chapter we will take a look at the results of the experiments described in the previous chapters. These results consist of the dialogue data set collected by the Pet Shop Game experiment (Section 5.1), the rules generated from the sequential patterns mined from the dialogue data (Section 5.2) and the data derived from the generated rules (Sections 5.2.2 and 5.2.3). The chapter is concluded in Section 5.3 with a qualitative analysis of all of these results in the context of a data mined chatbot.

5.1 Dialogue data set

In this section I will describe the dialogue data set which will be used to generate rules for the data mined chatbot and provide utterance usage statistics which are of interest for evaluating the controlled natural language.

5.1.1 Dialogues

From the 45 dialogues that were logged in the database, only 28 resulted in the dialogue being completed (i.e., the customer leaving the store through their own action). For the other 17 dialogues either no player turns were logged (6 dialogues) or the dialogue was interrupted (11 dialogues). These incomplete dialogues have either been caused by a player closing their web browser mid-game or by an inadvertent software error in the Pet Shop Game application.

The 28 dialogues that were completed constitute the Pet Shop Game data set, which is used in the remainder of this chapter. Appendix B has a listing of all the dialogues in the data set. An example of a dialogue from this data set is shown in Excerpt 3.

5.1.2 Utterances

Given the controlled natural language and excluding the "other" option, shopkeeper players could use 93 possible utterances and customer players could use 97 possible utterances. Across both roles players had the possibility to produce 176 unique utterances.

Of these 176 possible utterances, 108 were actually used in the Pet Shop Game dialogues. When utterances containing a user-specified input for the "other" option are included there are 294 utterance instances in the data set. The frequencies of the main utterance subtypes as defined by the controlled natural language are listed in Table 5.1. Examples of concrete utterances grouped by their number of occurrences are shown in Table 5.2.

```
* A customer enters the pet shop.
Customer: Hello.
Shopkeep: Hello. How may I help you?
Customer: What pets do you sell?
Shopkeep: We sell cat. We sell dog. We sell goldfish. We sell
          hamster.
Customer: Which pet is easy?
Shopkeep: What do you mean? A goldfish is a pet that is cheap
          in maintenance.
Customer: Give me the goldfish, please.
Shopkeep: Yes, that's possible. A goldfish costs 5 euro. Would
          you like to know more?
Customer: No, I know enough. Thank you.
* The shopkeep gives the animal to the customer.
Customer: Thank you. Goodbye.
Shopkeep: Yes, that'll be 5 euros.
* The customer pays the shopkeep for the animal.
Shopkeep: Thank you. Goodbye.
Customer: Goodbye.
Shopkeep: At your service. Goodbye.
* The customer leaves the store.
```

Excerpt 3: One of the dialogues collected by the Pet Shop Game experiment. Parts of utterances written in cursive type are not part of the controlled natural language grammar, but user-specified input for the "other" option. Lines starting with an asterisk are player actions.

	Excl. "other" option.			**Incl. "other" option.**		
Utterance type	**Cust.**	**Shop.**	**Total**	**Cust.**	**Shop.**	**Total**
assertion	42	102	144	42	159	201
question	108	37	145	150	37	187
greeting	88	78	166	88	78	166
Yes.	3	10	13	3	10	13
Yes, *reason_to_say_yes*.	11	19	30	32	69	101
No.	1	2	3	1	2	3
No, *reason_to_say_no*.	16	5	21	25	20	45

Table 5.1: Breakdown of the occurrences of the direct subtypes of utterance in the controlled natural language.

#	**Examples**
51	Thank you.
41	Goodbye.
39	Hello.
36	That's good.
28	**leave** / **enter**
25	**give** / **pay**
17	Would you like to know more?
16	How may I help you?
15	Welcome.
14	Yes, that's possible.
13	Yes. / At your service.
12	What pets do you sell?
11	I'm not sure.
9	Yes, that is okay.
8	What do you mean?
7	No, that's not possible. / I am sorry. / Yes, I need to know more. *(3 more)*
6	How much does a cat cost? / That is too expensive. / A hamster is a pet that is cheap in maintenance.
5	I don't understand. / We sell goldfish. / I would like to buy a dog. *(1 more)*
4	I would like to buy a hamster. / Give me the hamster, please. / How much does a hamster cost? *(4 more)*
3	A cat costs 50 euro. / Which pet is faithful? / How much does a goldfish cost? *(13 more)*
2	Give me the *boss*, please. / Which pet is big? / A dog is a pet that is faithful. *(22 more)*
1	Yes, *have fun with your guinea pigs.* / No, *I'll get one at the bookshop.* / A cat is a pet that is *armed with claws and teeth.* *(212 more)*

Table 5.2: Breakdown of the utterances found in the Pet Shop Game data set, grouped by their number of occurrence. Parts of utterances written in cursive type are not part of the controlled natural language grammar, but user-specified input for the "other" option.

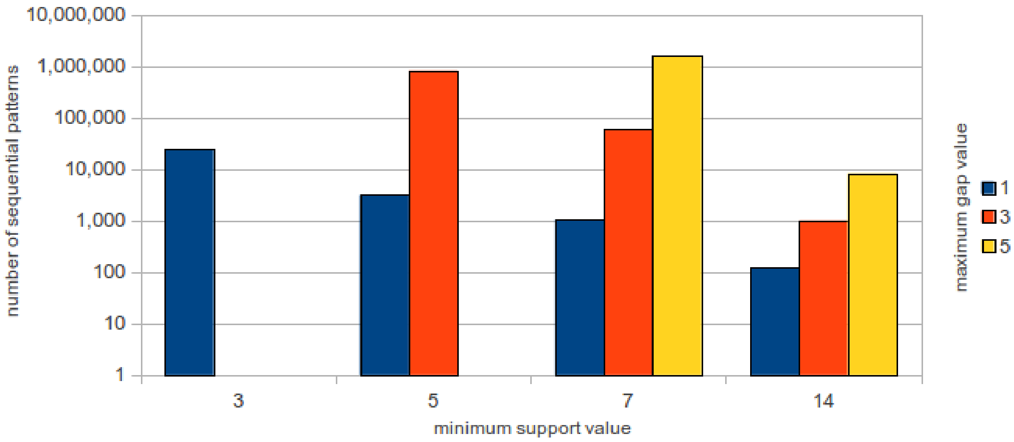

Figure 5.1: The number of sequential patterns generated given varying minimum support count and maximum gap size values.

5.2 Generated rules

The GSP algorithm was run on the Pet Shop Game dataset. The input for the taxonomy parameter was generated using the controlled natural language grammar. Multiple runs were made with differing values for the GSP algorithm parameters maximum gap size (max_gap) and minimum support count (min_sup). The minimum confidence value for generating sequential rules (min_conf) was set to 0.0 for all runs.

In Section 5.2.1 the quantitative results will be shown in the form of number generated sequential patterns and rules. Because quantity does not imply quality, we will see the results of applying these rules to the Pet Shop Game data set itself in Section 5.2.2. Finally, three random sample sets of the generated rules are listed in Section 5.2.3.

5.2.1 Number of patterns and rules

The number of sequential patterns and sequential rules that were generated are visualised in the bar charts shown in Figures 5.1 and 5.2 respectively.

Because certain combinations of parameters caused the number of sequential patterns generated to outgrow the available memory of the data mining hardware, there are no results for $min\_sup = 5$ and $max\_gap = 3$, $min\_sup = 3$ and $max\_gap = 3$, and $min\_sup = 3$ and $max\_gap = 5$.

5.2.2 Testing the rules

The effectiveness of the generated sequential rules was tested by applying them to the Pet Shop Game data set. Within the data set there are 260 instances of the shopkeeper speaking or performing a physical action. Each set of rules was applied to each of these instances' preceding dialogue history using the data mined chatbot algorithm described in Section 4.4. The algorithm's predictions of the shopkeeper's next dialogue turn are then compared to the dialogue turn that occurred in the data set.

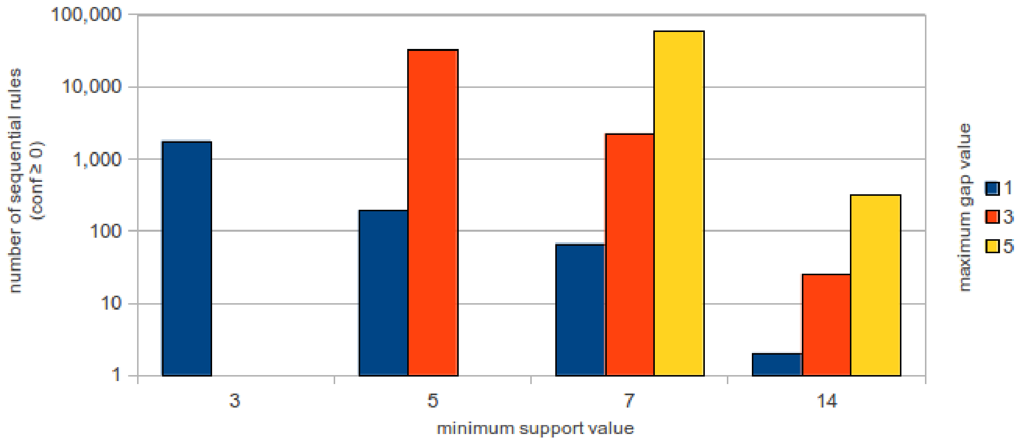

Figure 5.2: The number of sequential rules generated given varying minimum support count and maximum gap size values and a minimum confidence value of 0.0.

For this comparison three different correctness measures were used:

- **conf** – the highest confidence prediction matches the expected dialogue turn
- **spec, conf** – the most specific prediction with the highest confidence matches the expected dialogue turn
- **not none** – there are at least some predictions (i.e., the shopkeeper will not seem to be lost for words)

The aggregated results can be seen in Table 5.3. Figures 5.3 and 5.4 show the number of correct predictions as a percentage of the total number of 260 shopkeeper dialogue turns.

min. supp.	**max. gap**	**seq's**	**rules**	**correct predictions**		
				conf.	**spec., conf.**	**not none**
3	1	23697	1754	65	58	260
5	1	3102	191	51	50	246
5	3	799152	32567	45	47	260
7	1	1049	66	39	41	226
7	3	58076	2233	42	37	260
7	5	1537450	58508	36	32	260
14	1	121	2	7	7	71
14	3	991	25	33	26	257
14	5	8068	319	26	25	260

Table 5.3: The results of applying the rules generated by different minimum support count and maximum gap size parameter values to the Pet Shop Game data set using the data mined chatbot algorithm.

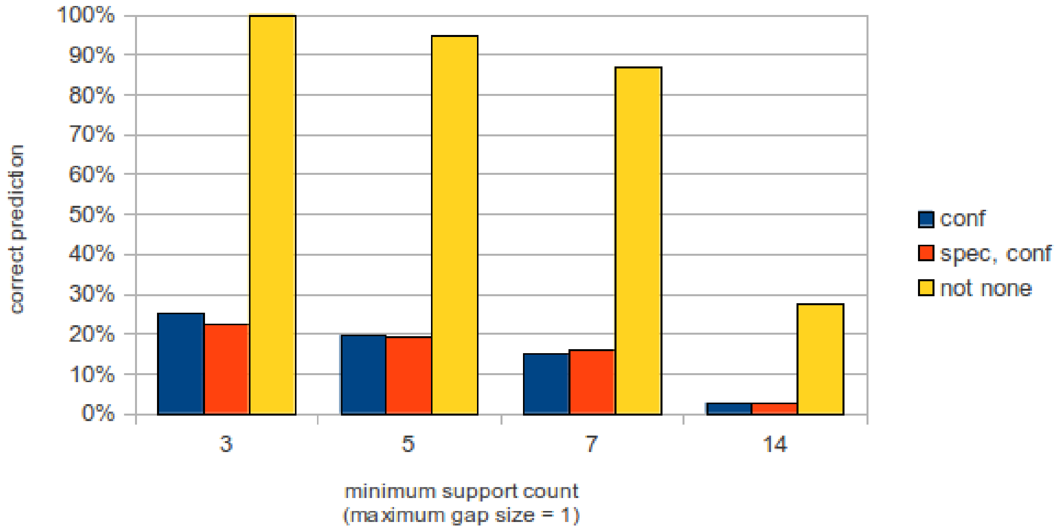

Figure 5.3: The percentage of correct predictions according to three different correctness measures (only match the highest confidence prediction, match to any prediction, and lastly, only null-predictions are incorrect) given rules generated by variable minimum support counts and a maximum gap size of 1.

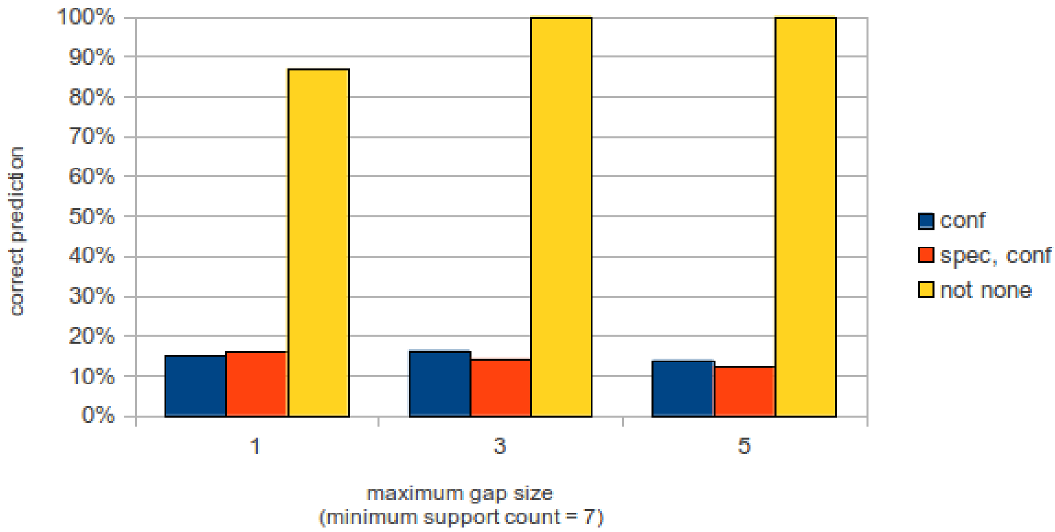

Figure 5.4: The percentage of correct predictions according to three different correctness measures (only match the highest confidence prediction, match to any prediction, and lastly, only null-predictions are incorrect) given rules generated by variable maximum gap sizes and a minimum support count of 7.

5.2.3 Samples of generated rules

In this section a number of sets of results are shown with different parameters. From each of these sets a small random sample is drawn which will be used later on to make a qualitative analysis of the sequential rules that are generated and the effects of the different parameters.

Baseline As a baseline set we will use the following values: $min\_sup = 7$, $max\_gap = 1$, and $min\_conf = 0.0$. This resulted in 1,049 sequences being generated, which translated into 39 sequential rules. A random sample of these sequential rules is listed in Table 5.4.

Minimum support count Keeping the other parameters in the baseline set equal and decreasing min_sup to 5 leads to 3,102 sequential patterns being generated, which translated into 125 rules. A random sample of these sequential rules is listed in Table 5.5.

Maximum gap size On the other hand, keeping the other parameters in the baseline set equal and increasing max_gap to 3 results in 58,076 sequential patterns being generated, which translated into 2,233 rules. A random sample of these sequential rules is listed in Table 5.6.

The sample rules are of the general $A \rightarrow B$ format, where A is the rule's condition, and B is the rule's action. For these dialogue rules, A consists of a sequence $\langle s_1 s_2 \ldots s_n \rangle$ of itemsets (i.e., dialogue turns) $(i_1, i_2, \ldots, i_m)$ where the items i_j are either concrete utterances (e.g., C: "Hello.") or ancestors of utterances (e.g., C: "greeting") according to the taxonomy based on the constrained natural language grammar. B consists of a sequence containing a single itemset containing only concrete utterances (e.g., $\langle$(S: "Welcome.", S: "How may I help you?")$\rangle$).

5.3 Analysis of results

I will analyse the various findings and derived results described above. The conclusions of this analysis will be used to support the answer to the research question in the following, concluding chapter.

In Section 5.3.1 I will analyse the quantitive results shown in the previous sections to find out the influence of the different parameter values on sequential rule generation. In Section 5.3.2 I will make a qualitative analysis of some of the rules generated based on the Pet Shop Game data set. Lastly in Section 5.3.3 I will give an analysis of the controlled natural language used in the Pet Shop Game based on how it was used by the players.

5.3.1 Quantitive rule generation analysis

Certain parameter combinations yielded a lot of sequential patterns and consequently a lot of sequential rules were generated. In this section I will analyse the quantitive results shown in Sections 5.2.1 and 5.2.2 to find out the influence of the different parameter values used to generate the sequential patterns. This analysis will be supported by findings from the randomly sampled sequential rules listed in Tables 5.4, 5.5, and 5.6 mentioned in Section 5.2.3.

Minimum support count Decreasing the minimum support count allows less frequent and thus more sequential patterns to be found, this in turn leads to more sequential rules. Examples can be seen in Table 5.5. All rules but 2, 4, and 8 are part of this ruleset by virtue of the lower

Table 5.4: Random sample of sequential rules (min_sup = 7, max_gap = 1)

#	**Sequential rule**	**conf.**	**supp.**
1	⟨(S:“assertion”) (C:“question”) (S:“assertion”) (C:“action”)⟩ → ⟨(S:“give_action”)⟩	1.00	7
2	⟨(S:“assertion”) (C:“question”) (S:“assertion”) (C:“question”) (S:“assertion”) (C:“pay_action”)⟩ → ⟨(S:“give_action”)⟩	1.00	7
3	⟨(C:“question”) (S:“assertion”) (C:“question”) (S:“assertion”) (C:“action”)⟩ → ⟨(S:“give_action”)⟩	1.00	7
4	⟨(C:“question”) (S:“assertion”) (C:“action”)⟩ → ⟨(S:“give_action”)⟩	0.88	7
5	⟨(S:“assertion”) (C:“action”)⟩ → ⟨(S:“give_action”)⟩	0.78	7
6	⟨(C:“action”) (S:“give_action”) (C:“greeting”)⟩ → ⟨(S:“At your service.”)⟩	0.64	7
7	⟨(C:“Goodbye.”, C:“Thank you.”)⟩ → ⟨(S:“Goodbye.”)⟩	0.62	10
8	⟨(C:“greeting”)⟩ → ⟨(S:“At your service.”)⟩	0.48	13
9	⟨(S:“Yes, $reason_to_say_yes.”) (C:“question”)⟩ → ⟨(S:“Yes, #other_reason_to_say_yes.”)⟩	0.47	8
10	⟨(C:“question”)⟩ → ⟨(S:“Yes.”)⟩	0.25	7

Given a minimum support value of 7 and a maximum gap size of 1, 1,049 sequential patterns were found. From these patterns 66 sequential rules were generated. This is a random sample of 10 rules ordered by their confidence value.

Table 5.5: Random sample of sequential rules (min_sup = 5, max_gap = 1)

#	**Sequential rule**	**conf.**	**supp.**
1	⟨(C:“question”) (S:“assertion”) (C:“question”) (S:“assertion”) (C:“question”) (S:“assertion”) (C:“pay_action”)⟩ → ⟨(S:“give_action”)⟩	1.00	6
2	⟨(C:“action”) (S:“greeting”) (C:“Goodbye.”)⟩ → ⟨(S:“Goodbye.”)⟩	0.89	8
3	⟨(S:“action”) (C:“pay_action”) (S:“greeting”) (C:“Goodbye.”, C:“Thank you.”)⟩ → ⟨(S:“Goodbye.”)⟩	0.83	5
4	⟨(S:“assertion”) (C:“action”)⟩ → ⟨(S:“give_action”)⟩	0.78	7
5	⟨(C:“enter_action”) (C:“Hello.”)⟩ → ⟨(S:“Welcome.”)⟩	0.75	6
6	⟨(C:“action”) (S:“give_action”) (C:“Goodbye.”)⟩ → ⟨(S:“At your service.”)⟩	0.71	5
7	⟨(S:“assertion”) (C:“Can you tell me if a $type_of_animal is $pet_property?”)⟩ → ⟨(S:“Yes.”)⟩	0.36	5
8	⟨(C:“greeting”)⟩ → ⟨(S:“How may I help you?”)⟩	0.30	8
9	⟨(C:“question”)⟩ → ⟨(S:“A dog costs #other_sum_of_money euro.”)⟩	0.21	6
10	⟨(C:“greeting”)⟩ → ⟨(S:“Thank you.”)⟩	0.19	5

Given a minimum support value of 5 and a maximum gap size of 1, 3,102 sequential patterns were found. From these patterns 191 sequential rules were generated. This is a random sample of 10 rules ordered by their confidence value.

Table 5.6: Random sample of sequential rules (min_sup = 7, max_gap = 3)

#	**Sequential rule**	**conf.**	**supp.**
1	⟨(S:“assertion”) (C:“question”) (S:“action”) (C:“action”)⟩ → ⟨(S:“Thank you.”)⟩	1.00	7
2	⟨(S:“A $type_of_animal costs $sum_of_money euro.”) (C:“question”) (S:“assertion”) (C:“pay_action”)⟩ → ⟨(S:“give_action”)⟩	0.82	9
3	⟨(C:“question”) (S:“assertion”) (S:“assertion”) (C:“pay_action”)⟩ → ⟨(S:“give_action”)⟩	0.77	10
4	⟨(C:“greeting”) (S:“assertion”) (C:“question”) (S:“assertion”) (S:“assertion”) (C:“question”)⟩ → ⟨(S:“give_action”)⟩	0.58	7
5	⟨(C:“greeting”) (C:“question”) (S:“assertion”) (C:“question”) (C:“question”)⟩ → ⟨(S:“give_action”)⟩	0.57	8
6	⟨(S:“greeting”) (C:“question”) (S:“A $type_of_animal is a pet that is $pet_property.”) (C:“question”) (C:“question”)⟩ → ⟨(S:“give_action”)⟩	0.50	7
7	⟨(C:“action”) (S:“How may I help you?”) (C:“question”) (C:“question”)⟩ → ⟨(S:“Yes, #other_reason_to_say_yes.”)⟩	0.50	7
8	⟨(S:“assertion”) (C:“assertion”, C:“question”)⟩ → ⟨(S:“give_action”)⟩	0.44	8
9	⟨(S:“assertion”) (C:“question”) (S:“assertion”) (C:“question”)⟩ → ⟨(S:“Yes, #other_reason_to_say_yes.”)⟩	0.41	9
10	⟨(C:“question”)⟩ → ⟨(S:“Yes, that’s possible.”)⟩	0.36	10

Given a minimum support value of 7 and a maximum gap size of 3, 58,076 sequential patterns were found. From these patterns 2,233 sequential rules were generated. This is a random sample of 10 rules ordered by their confidence value.

minimum support count. The lower support count also does not imply these rules are of a lower quality per se, as can be seen by their confidence value.

As can be seen in Figure 5.3, the increase of the minimum support count decreases the amount of rules that match the expected dialogue turn. Only for a minimum support count of 3 there is always at least one rule that returns a prediction. More rules due to a lower minimum support count increases the usefulness of the summed confidence measure in determining the best conclusion, as the lowest minimum support count yields the best results when trying to match the highest summed confidence conclusion to the expected dialogue turn.

Maximum gap size A gap size of 1 causes the alternating pattern of customer and shopkeeper dialogue turns within the sequential patterns in Tables 5.4 and 5.5. Increasing the maximum gap size from 1 to 3 increases the number of sequential patterns as we now accept gaps within the sequence. The itemsets within a sequential pattern model dialogue turns, so increasing this parameter by 2 means the algorithm may now skip 2 dialogue turns between any single dialogue turn of a dialogue. This means any question—answer interjunctions may now occur within a larger pattern without preventing the algorithm from finding that encompassing pattern.

Another effect of increasing the gap size is that sequential patterns on average become longer. Longer patterns means that generated rules have preconditions that "look further back" in a dialogue before matching.

As can be seen based on Figure 5.4, the increase of the maximum gap size increases the amount of rules that match the expected dialogue turn. For maximum gap sizes of 3 and 5 there is always at least one rule that returns a prediction, and the bigger gap size also increases the chance a conclusion that matches the expected dialogue turn comes up. However, the quality of highest confidence prediction goes down. As more rules will match the same dialogue history there is more competition for that highest confidence spot, decreasing the likelihood the right prediction is made.

5.3.2 Qualitative rule generation analysis

In this section I make a qualitative analysis of the types of rules generated by the system. This analysis will be based on the 25 sequential rules generated by $min\_sup = 14$ and $max\_gap = 3$ and the 66 sequential rules generated by $min\_sup = 7$ and $max\_gap = 1$. These rules are listed in Appendix A.

A good set of dialogue rules should possess the following marks of quality:

1. each conclusion has a rule – to ensure the system is *expressively complete*
2. some rules have general conditions – to ensure there is always at least one rule that matches
3. little overlap between rules – to minimize the amount of matching rules and improve the prediction

In the following I will apply these marks of quality to the rule sets in Appendix A.

Each conclusion has a rule According to the pigeon hole principle neither of the investigated rule sets can conclude all of the 93 single-utterance dialogue turns the shopkeeper is able to utter[1]. The total number of possible dialogue turns is even larger than this number, because players are able to use multiple utterances per dialogue turn.

[1]Which is not counting the shopkeeper's actions and the "other" option for specific utterances (see also Section 5.1.2).

conclusion (dialogue turn)	rules
⟨(S:"give_action")⟩	14
⟨(S:"Goodbye.")⟩	4
⟨(S:"How may I help you?")⟩	2
⟨(S:"Welcome.")⟩	2
⟨(S:"Hello.")⟩	2
⟨(S:"Yes, #other_reason_to_say_yes.")⟩	1

Table 5.7: Unique conclusions for the rule set generated from $min\_sup = 14$ and $max\_gap = 3$.

conclusion (dialogue turn)	rules
⟨(S:"Goodbye.")⟩	16
⟨(S:"give_action")⟩	14
⟨(S:"At your service.")⟩	9
⟨(S:"Yes, #other_reason_to_say_yes.")⟩	8
⟨(S:"Thank you.")⟩	6
⟨(S:"How may I help you?")⟩	3
⟨(S:"Would you like to know more?")⟩	3
⟨(S:"Yes.")⟩	2
⟨(S:"Welcome.")⟩	2
⟨(S:"Hello.")⟩	2
⟨(S:"Yes, that's possible.")⟩	1

Table 5.8: Unique conclusions for the rule set generated from $min\_sup = 7$ and $max\_gap = 1$.

In fact, the $min\_sup = 14$ and $max\_gap = 3$ rule set shown in Table A.1 only has 6 distinct conclusions. About half of the rules (14 to be precise) for this rule set conclude ⟨(S:"give_action")⟩. For the $min\_sup = 7$ and $max\_gap = 1$ rule set (Table A.2) the results are not much better, despite having more than two times as much rules—66 as opposed to 25—the rule set is able to conclude 11 dialogue turns. Tables 5.7 and 5.8 list the counts for these dialogue turn conclusions for both rule sets.

Some rules have general conditions It is important for the shopkeeper chatbot to keep the dialogue flowing. This means it should always be able to match a rule to the current dialogue history. To see whether the rule sets have a sufficient number of rules with a general condition, I will consider rules with the most general condition possible: a condition that consists of a sequence containing one dialogue turn with one utterance. According to this definition, the most general rules in the two rule sets are shown in Tables 5.9 and 5.10.

The main utterance subtypes are "assertion", "question", "greeting", "Yes.", "No.", "Yes, $reason_to_say_yes." and "No, $reason_to_say_no." and the customer is able to perform an "action". Both rule sets have general rules to deal with "question", "greeting" and "action".

There are no general rules that deal with an "assertion", "Yes.", "No.", "Yes, $reason_to_say_yes." or "No, $reason_to_say_no." uttered by the customer. This is presumably due to their relatively low occurrence in the data set as can be seen in Table 5.1.

Curiously, however, in the case of $min\_sup = 14$ and $max\_gap = 3$ a rule was found that has the shopkeeper give a pet to the customer whenever he had uttered an "assertion" on his previous dialogue turn. This can be explained by the increase in maximum gap size, which allows more patterns to be found by skipping dialogue turns in the data-sequence.

condition	rules	highest conf. conclusion
⟨(C:“action”)⟩	3	⟨(S:“Hello.”)⟩
⟨(C:“enter_action”)⟩	3	⟨(S:“Hello.”)⟩
⟨(C:“question”)⟩	2	⟨(S:“give_action”)⟩
⟨(S:“assertion”)⟩	1	⟨(S:“give_action”)⟩
⟨(C:“greeting”)⟩	1	⟨(S:“Goodbye.”)⟩
⟨(C:“Goodbye.”)⟩	1	⟨(S:“Goodbye.”)⟩

Table 5.9: General conditions (i.c., containing a single utterance) for the rule set generated from $min\_sup = 14$ and $max\_gap = 3$. The third column contains the conclusion with the highest confidence for rules with these conditions.

condition	rules	highest conf. conclusion
⟨(C:“action”)⟩	5	⟨(S:“give_action”)⟩
⟨(C:“question”)⟩	4	⟨(S:“Yes, #other_reason_to_say_yes.”)⟩
⟨(C:“greeting”)⟩	4	⟨(S:“Goodbye.”)⟩
⟨(C:“enter_action”)⟩	3	⟨(S:“Hello.”)⟩
⟨(C:“pay_action”)⟩	2	⟨(S:“give_action”)⟩
⟨(C:“Thank you.”)⟩	2	⟨(S:“Goodbye.”)⟩
⟨(C:“Goodbye.”)⟩	2	⟨(S:“Goodbye.”)⟩
⟨(C:“Can you tell me if a $type_of_animal is $pet_property?”)⟩	1	⟨(S:“Yes, #other_reason_to_say_yes.”)⟩
⟨(C:“Goodbye.”, C:“Thank you.”)⟩	1	⟨(S:“Goodbye.”)⟩

Table 5.10: General conditions (i.c., containing a single utterance) for the rule set generated from $min\_sup = 7$ and $max\_gap = 1$. The third column contains the conclusion with the highest confidence for rules with these conditions.

Little overlap between rules One of the main things that catches the eye in Table A.2 is the similarity of the dialogue rules. Rules 1 through 6 all contain the subsequence ⟨(S:"assertion") (C:"question") (S:"assertion" in their condition, and conclude ⟨(S:"give_action")⟩. Something similar happens with ⟨(S:"greeting") (C:"greeting")⟩ and ⟨(S:"Goodbye.")⟩ for rules 7 through 12.

This is caused by basing the taxonomy on the controlled natural language grammar. Because most concrete utterances have either an assertion or a question as their ancestor (cf. Table 5.1), a lot of patterns containing chains of questions and assertions will be found.

While these patterns obviously occur within the dialogues in the Pet Shop Game data set, it also seems unlikely that the (S:"assertion") and (C:"question") dialogue turns are really as relevant for the (S:"give_action") as rule 1 in Table 5.4 with its confidence value of 1.0 seems to imply.

Neither of the analysed rule sets possesses the marks of quality. It is possible that the first two marks ("Each conclusion has a rule", "Some rules have general conditions") can be fixed by increasing the data set size or lowering the minimum support value, but it is also likely that the problematic overlap between rules remains together with its detrimental effect on predictive ability.

Nevertheless, based on the conclusions shown in Table 5.8 it seems like a minimal "shop" dialogue is supported when using the rules generated from $min\_sup = 7$ and $max\_gap = 1$, as long as the human customer does not stray too far from the core dialogue.

5.3.3 Controlled natural language analysis

The use of the controlled natural language was analysed by looking at all the dialogues listed in Appendix B.

It seems players used the "other" option relatively often to enter a turn of phrase that was not supported by the system. The controlled natural language grammar nodes "Yes, ..." and "No, ..." were the main targets of this behaviour. Using the "other" option in this way means that the meaning of those speech acts does not have anything to do anymore with saying yes for some reason, but is totally dominated by whatever the player substituted for a reason to say yes, e.g., "Yes, have fun with your guinea pigs." says more about guinea pigs than it is saying "Yes." supplemented with some reason.

Based on the number of utterances occurring in the data set (Section 5.1.2) it appears the controlled natural language was *sound* (i.e., a significant number of utterances that players were able to produce, were produced), but in this same sense it was, naturally, not *complete* (i.e., players could produce all utterances they needed to produce), as exemplified by the fact people needed to overload the meaning of "Yes, ..." and "No, ..." using the "other" option.

Even though the Pet Shop Game data set is very small, a large number of rules can be generated given the right parameters. However, more rules does not mean the quality of the system improves. The best results for the Pet Shop Game data set seem to result from a GSP parameter setting of $min\_sup = 3$ and $max\_gap = 1$. The quality of the GSP taxonomy parameter also plays a role in providing a manageable number of relevant rules. Making the taxonomy too expressive makes the number of found patterns explode which leads to the creation of spurious dialogue rules.

These spurious rules mean additional competition for the "right" prediction, because many rules might match a given dialogue history. Finding the "right" rule then entails more tweaking of the prediction algorithm, which is something that might not be possible to solve in the general case. In that case the problem of manually finding the right dialogue rules has been replaced by an orthogonal problem of (semi-)manually finding the right rules for making a subselection of dialogue rules.

The inclusion of the "other" option in the controlled natural language was not a good idea, as players used it to overload the meaning of certain speech acts within utterances. This blurred the meaning of these utterances and consequently the patterns in which they were found are less informative.

Chapter 6

Discussion

In this chapter I will discuss the successes and shortcomings of the methodology as put forward in Chapter 3 and how it influenced the results shown in the previous chapter. This discussion is split into two parts. I will first talk about the theoretical aspects of the methodology, and subsequently more practical considerations are put forward.

6.1 Theoretical discussion

6.1.1 Experimental set-up

A fair judgement of the data mining approach to creating chatbots would, first and foremost, need to collect a large amount of data to get statistically significant results from the data mining algorithm. Secondly, a comparable hand-written chatbot would also need to be made by a third party. The time investment required for these prerequisites fell outside the scope of this thesis project. This means it was not possible to fairly test the hypothesis beyond the qualitative sample-based analysis of the previous chapter.

6.1.2 Comparing rules

A lot of sequential dialogue rules can be generated using the data mining method, but quantity is no measure of quality so further qualitative analysis was performed in the previous chapter. However, it should be mentioned that these data mined rules are of a type that specifically plays to the strength of the data mined chatbot algorithm. The algorithm works by aggregating a large number of weak and easily triggered rules to come to a conclusion. A hand-written chatbot on the other hand, uses a smaller, more manageable number of stronger and more specific rules to operate. This difference in rule "strength" means it is hard to compare the data mined rules head-to-head with hand-written rules.

6.2 Practical discussion

6.2.1 Controlled natural language

The design of the controlled natural language provided restricted language user input for the Pet Shop Game, which was convenient for keeping the search space small for the GSP algorithm.

However, it is also apparent from the players' use of the language that they were not able to express themselves completely using the available utterances.

There are two possible directions for a solution to this problem. The natural language input can be controlled using an improved grammar that is based on trial runs to find out the player's desired level of expressiveness for the language, or the grammar should be foregone in favour of free natural language input combined with an extra processing step which automatically classifies the free natural language utterances.

A mix between these two could be the former, using a restrictive grammar that includes the "other" option as provided in the Pet Shop Game. However, the free user input to the "other" option will need to be classified using natural language processing so as to keep the problem space small.

6.2.2 Scenario

The pet shop scenario ties into the problem of player freedom as well. The scenario was setup to minimize the player's expectations: a pet shop is assumed to have commonplace associations for people and the shop's assortment of products was described to the players as limited. Despite this, and partly due to the role-playing nature of the experiment, the scenario still invited people to come up with guinea pigs for sale and bosses that could be spoken to.

6.2.3 Taxonomy

We used the controlled natural language to create the input taxonomy for the GSP algorithm. The controlled natural language was designed for ease of use by the players, but not with the GSP taxonomy in mind. This meant certain relationships between utterances did not make sense as far as the data mining algorithm was concerned.

To summarise, there are two theoretical obstacles in creating a data mined chatbot, namely getting enough data to mine and creating a fair judgement of its ability. A larger experimental set-up would need to be employed than was used in this research. Practical problems that became apparent after the experiment concluded are the design of the controlled natural language and the pet shop scenario. Both were meant to reduce player freedom in a way that did not feel too restrictive, but it seems in both cases that the right balance was not struck. Lastly, the taxonomy input for the GSP algorithm could have been better tailored to representing speech acts instead of using the controlled natural grammar as a template.

Chapter 7

Conclusion

In this concluding chapter I will answer the research questions posed in Chapter 1 by summarising the observations made in the last two chapters. I will also point out avenues for future research, including ways to improve upon the methodology used in this research.

7.1 Research question

The subquestions to the research question were:

> How useful is sequential pattern mining as a source of patterns for generating artificial, goal-oriented dialogue?
>
> What are the obstacles specific to generating artificial, goal-oriented dialogue, and how can they be overcome?

The first question can be answered positively, the temporal yet turn-based structure of behaviour in dialogues is quite naturally represented in sequential patterns. It should be noted that sequential pattern mining, like all data mining techniques, is tuned for large amounts of data to make sure statistically relevant patterns are found.

The second question is answered by summarising the obstacles found:

Reasoning vs. planning Generating a goal-oriented dialogue is both about goal-oriented "reasoning" (topic-tracking, using world knowledge, natural language processing) and dialogue "planning" (conversation flow, matching appropriate responses). These two aspects of generating behaviour are ideally modelled in a single system, but this is quite hard, which is why this research focuses purely on the planning part of dialogue generation.

Non-frequent yet interesting patterns Patterns exist which are very relevant in specific dialogue contexts, but which are not found because the dialogue data containing these patterns is too sparse. This might be due to too little data in general, but it could also mean that the method of data generation insufficiently saturizes all parts of the specific dialogue search space. The first problem can be solved by generating more data or by employing techniques to find interesting yet non-frequent patterns (see also Section 7.2). The second problem can be solved by introducing more variation into the data generation environment, e.g., what has been done in the Pet Shop Game by priming the players with small background stories.

The research question was:

> How does generating artificial, goal-oriented dialogue using automated data mining weigh up against a manual approach to this problem?

The main research question can only be answered by saying that it depends. Both approaches suffer from the obstacles mentioned in the second subquestion. But when reducing the approaches to methods of generating rules, the data mining approach has more favorable scaling properties. Given enough data to work with, mining patterns should result in more variation and a lower "error rate" both of which might result from a cognitive bias in the hand-written apparoach (involving human domain experts). Once a data mining system is in place, including the facilities to generate data cheap and fast, having it generate rules is more cost-effective. However, there is an investment cost here that must be paid, and before this leap is made, the hand-written approach is the way to go.

7.2 Future research

In this section I make suggestions for further research. These suggestions are comprised of both improvements to the methodology of the Pet Shop Game experiment and insights related to dialogue data mining that might prove interesting research subjects.

Improved experimental set-up To really compare the data mining of dialogue rules to a hand-written approach it is necessary to collect way more data. Furthermore, a comparative analysis of a full-featured implementation of the data mined chatbot and a hand-written chatbot for the same goal-oriented scenario, the latter being created by a third party.

Better controlled natural language Increase the production capabilities of the controlled natural language without giving up all favourable properties for language/behaviour data mining purposes. Or: What are ways to limit open-endedness while staying true to the structure of natural language dialogue?

Other types of dialogue There exist a lot more goal-oriented dialogue types, as described amongst others by Walton & Krabbe. Dialogue types like interrogation, containing orders or requests for action, making promises, but also multi-way dialogue or dialogue that is suddenly interrupted by an external event all qualify for further experimentation because they pose new ways in which the dialogue state is structured.

Other types of taxonomy Next to the is-a hierarchy, mapping utterances to speech acts, it might be interesting to also have a has-a hierarchy, mapping utterances to their content. GSP supports multiple inheritance taxonomies by design. It is interesting to see if it is possible to also data mine semantic patterns this way (e.g., we are talking about dogs, and dogs are associated with security, so we will not suddenly be talking about goldfish).

"Unexpected" sequential patterns Recently, a method has been proposed to find interesting sequential patterns, even if they are not abundant within the data [6]. The method involves finding rules which contradict expert beliefs. This might be a way to get more rules from a limited set, a way to find relatively rare "boundary behaviour" which can still be valid in some circumstances, or a method for creating a hybrid system that combines the manual creation of dialogue rules with data mining to find these additional interesting patterns.

Implications of emergence Consider two data-mined chatbots, one of them has a one in a thousand chance to make a catastrophic error (like an extremely insulting remark[1]), the other has a one in ten chance of making minor mistakes. The former is basically useless, the latter might be acceptable. What causes this dichotomy and how can this be dealt with.

[1] For example, in 2007 Microsoft pulled the plug from its Santa Claus chatbot because it said things like "I think you're a dirty bastard." to teenagers.

Appendix A

Rules

This chapter lists the rule sets generated by the system for two different parametrisations, namely $min\_sup = 14$ and $max\_gap = 3$ (Table A.1), and $min\_sup = 7$ and $max\_gap = 1$ (Table A.2).

Table A.1: Sequential rules (min_sup = 14, max_gap = 3)

#	Sequential rule	conf.	supp.
1	⟨(C:“Goodbye.”)⟩ → ⟨(S:“Goodbye.”)⟩	0.71	15
2	⟨(C:“action”)(C:“greeting”)⟩ → ⟨(S:“Goodbye.”)⟩	0.70	14
3	⟨(C:“pay_action”)(C:“greeting”)⟩ → ⟨(S:“Goodbye.”)⟩	0.70	14
4	⟨(S:“assertion”)(C:“question”)(S:“assertion”)(C:“question”)⟩ → ⟨(S:“give_action”)⟩	0.68	15
5	⟨(S:“assertion”)(S:“assertion”)(C:“question”)⟩ → ⟨(S:“give_action”)⟩	0.67	14
6	⟨(C:“question”)(S:“assertion”)(C:“question”)(S:“assertion”)(C:“question”)⟩ → ⟨(S:“give_action”)⟩	0.67	14
7	⟨(S:“assertion”)(C:“question”)(C:“question”)⟩ → ⟨(S:“give_action”)⟩	0.67	16
8	⟨(C:“question”)(S:“assertion”)(C:“question”)(C:“question”)⟩ → ⟨(S:“give_action”)⟩	0.65	15
9	⟨(C:“question”)(C:“question”)⟩ → ⟨(S:“give_action”)⟩	0.63	17
10	⟨(C:“question”)(C:“question”)(S:“assertion”)(C:“question”)⟩ → ⟨(S:“give_action”)⟩	0.62	15
11	⟨(C:“question”)⟩ → ⟨(S:“give_action”)⟩	0.61	17
12	⟨(C:“question”)(C:“question”)(C:“question”)⟩ → ⟨(S:“give_action”)⟩	0.60	15
13	⟨(C:“question”)(S:“assertion”)(C:“question”)⟩ → ⟨(S:“give_action”)⟩	0.59	16
14	⟨(S:“assertion”)(C:“question”)⟩ → ⟨(S:“give_action”)⟩	0.59	16
15	⟨(C:“action”)⟩ → ⟨(S:“Hello.”)⟩	0.57	16
16	⟨(C:“enter_action”)⟩ → ⟨(S:“Hello.”)⟩	0.57	16
17	⟨(S:“assertion”)(C:“question”)(S:“assertion”)⟩ → ⟨(S:“give_action”)⟩	0.56	14
18	⟨(C:“greeting”)⟩ → ⟨(S:“Goodbye.”)⟩	0.56	15
19	⟨(C:“question”)(S:“assertion”)⟩ → ⟨(S:“give_action”)⟩	0.54	15
20	⟨(S:“assertion”)⟩ → ⟨(S:“give_action”)⟩	0.54	15
21	⟨(C:“action”)⟩ → ⟨(S:“How may I help you?”)⟩	0.54	15
22	⟨(C:“enter_action”)⟩ → ⟨(S:“Welcome.”)⟩	0.54	15
23	⟨(C:“action”)⟩ → ⟨(S:“Welcome.”)⟩	0.54	15
24	⟨(C:“enter_action”)⟩ → ⟨(S:“How may I help you?”)⟩	0.54	15

Continued on next page

Table A.1 – *Continued from previous page*

#	**Sequential rule**	**conf.**	**supp.**
25	⟨(C:"question")⟩ → ⟨(S:"Yes, #other_reason_to_say_yes.")⟩	0.50	14

Table A.2: Sequential rules (min_sup = 7, max_gap = 1)

#	**Sequential rule**	**conf.**	**supp.**
1	⟨(S:"assertion")(C:"question")(S:"assertion")(C:"question") (S:"assertion")(C:"pay_action")⟩ → ⟨(S:"give_action")⟩	1.00	7
2	⟨(C:"question")(S:"assertion")(C:"question")(S:"assertion") (C:"pay_action")⟩ → ⟨(S:"give_action")⟩	1.00	7
3	⟨(S:"assertion")(C:"question")(S:"assertion")(C:"pay_action")⟩ → ⟨(S:"give_action")⟩	1.00	7
4	⟨(S:"assertion")(C:"question")(S:"assertion")(C:"action")⟩ → ⟨(S:"give_action")⟩	1.00	7
5	⟨(C:"question")(S:"assertion")(C:"question")(S:"assertion") (C:"action")⟩ → ⟨(S:"give_action")⟩	1.00	7
6	⟨(S:"assertion")(C:"question")(S:"assertion")(C:"question") (S:"assertion")(C:"action")⟩ → ⟨(S:"give_action")⟩	1.00	7
7	⟨(S:"greeting")(C:"Goodbye.")⟩ → ⟨(S:"Goodbye.")⟩	0.90	9
8	⟨(C:"pay_action")(S:"greeting")(C:"Goodbye.")⟩ → ⟨(S:"Goodbye.")⟩	0.89	8
9	⟨(C:"action")(S:"greeting")(C:"greeting")⟩ → ⟨(S:"Goodbye.")⟩	0.89	8
10	⟨(C:"action")(S:"greeting")(C:"Goodbye.")⟩ → ⟨(S:"Goodbye.")⟩	0.89	8
11	⟨(C:"pay_action")(S:"greeting")(C:"greeting")⟩ → ⟨(S:"Goodbye.")⟩	0.89	8
12	⟨(C:"pay_action")(S:"Thank you.")(C:"greeting")⟩ → ⟨(S:"Goodbye.")⟩	0.88	7
13	⟨(S:"Yes, $reason_to_say_yes.")(C:"pay_action")⟩ → ⟨(S:"give_action")⟩	0.88	7
14	⟨(S:"action")(C:"action")⟩ → ⟨(S:"Thank you.")⟩	0.88	7
15	⟨(S:"give_action")(C:"action")⟩ → ⟨(S:"Thank you.")⟩	0.88	7
16	⟨(S:"give_action")(C:"pay_action")⟩ → ⟨(S:"Thank you.")⟩	0.88	7
17	⟨(S:"Yes, $reason_to_say_yes.")(C:"action")⟩ → ⟨(S:"give_action")⟩	0.88	7
18	⟨(S:"Thank you.")(C:"greeting")⟩ → ⟨(S:"Goodbye.")⟩	0.88	7
19	⟨(C:"pay_action")(S:"Thank you.")(C:"Goodbye.")⟩ → ⟨(S:"Goodbye.")⟩	0.88	7
20	⟨(C:"action")(S:"Thank you.")(C:"greeting")⟩ → ⟨(S:"Goodbye.")⟩	0.88	7
21	⟨(S:"Thank you.")(C:"Goodbye.")⟩ → ⟨(S:"Goodbye.")⟩	0.88	7
22	⟨(C:"question")(S:"assertion")(C:"pay_action")⟩ → ⟨(S:"give_action")⟩	0.88	7
23	⟨(C:"question")(S:"assertion")(C:"action")⟩ → ⟨(S:"give_action")⟩	0.88	7
24	⟨(C:"action")(S:"Thank you.")(C:"Goodbye.")⟩ → ⟨(S:"Goodbye.")⟩	0.88	7
25	⟨(S:"action")(C:"pay_action")⟩ → ⟨(S:"Thank you.")⟩	0.88	7
26	⟨(S:"assertion")(C:"action")⟩ → ⟨(S:"give_action")⟩	0.78	7
27	⟨(S:"assertion")(C:"pay_action")⟩ → ⟨(S:"give_action")⟩	0.78	7
28	⟨(C:"Goodbye.")⟩ → ⟨(S:"Goodbye.")⟩	0.71	15
29	⟨(C:"action")(S:"give_action")(C:"greeting")⟩ → ⟨(S:"At your service.")⟩	0.64	7

Continued on next page

Table A.2 – *Continued from previous page*

#	**Sequential rule**	**conf.**	**supp.**
30	⟨(C:“pay_action”)(S:“action”)(C:“greeting”)⟩ → ⟨(S:“At your service.”)⟩	0.64	7
31	⟨(C:“action”)(S:“action”)(C:“greeting”)⟩ → ⟨(S:“At your service.”)⟩	0.64	7
32	⟨(C:“pay_action”)(S:“give_action”)(C:“greeting”)⟩ → ⟨(S:“At your service.”)⟩	0.64	7
33	⟨(C:“Goodbye.”, C:“Thank you.”)⟩ → ⟨(S:“Goodbye.”)⟩	0.62	10
34	⟨(S:“action”)(C:“greeting”)⟩ → ⟨(S:“At your service.”)⟩	0.62	8
35	⟨(S:“give_action”)(C:“greeting”)⟩ → ⟨(S:“At your service.”)⟩	0.62	8
36	⟨(C:“greeting”)⟩ → ⟨(S:“Goodbye.”)⟩	0.56	15
37	⟨(S:“Yes, #other_reason_to_say_yes.”)(C:“question”)⟩ → ⟨(S:“Yes, #other_reason_to_say_yes.”)⟩	0.54	7
38	⟨(C:“action”)⟩ → ⟨(S:“give_action”)⟩	0.52	13
39	⟨(C:“pay_action”)⟩ → ⟨(S:“give_action”)⟩	0.52	13
40	⟨(C:“greeting”)⟩ → ⟨(S:“At your service.”)⟩	0.48	13
41	⟨(C:“Goodbye.”)⟩ → ⟨(S:“At your service.”)⟩	0.48	10
42	⟨(S:“Yes, $reason_to_say_yes.”)(C:“question”)⟩ → ⟨(S:“Yes, #other_reason_to_say_yes.”)⟩	0.47	8
43	⟨(C:“question”)(S:“Yes, $reason_to_say_yes.”)(C:“question”)⟩ → ⟨(S:“Yes, #other_reason_to_say_yes.”)⟩	0.47	7
44	⟨(C:“action”)⟩ → ⟨(S:“Hello.”)⟩	0.46	13
45	⟨(C:“question”)⟩ → ⟨(S:“Yes, #other_reason_to_say_yes.”)⟩	0.46	13
46	⟨(C:“enter_action”)⟩ → ⟨(S:“Hello.”)⟩	0.46	13
47	⟨(S:“greeting”)(C:“greeting”)⟩ → ⟨(S:“Goodbye.”)⟩	0.45	9
48	⟨(C:“Thank you.”)⟩ → ⟨(S:“Goodbye.”)⟩	0.43	10
49	⟨(C:“Thank you.”)⟩ → ⟨(S:“At your service.”)⟩	0.39	9
50	⟨(C:“Can you tell me if a $type_of_animal is $pet_property?”)⟩ → ⟨(S:“Yes, #other_reason_to_say_yes.”)⟩	0.39	7
51	⟨(C:“action”)⟩ → ⟨(S:“Thank you.”)⟩	0.36	9
52	⟨(C:“pay_action”)⟩ → ⟨(S:“Thank you.”)⟩	0.36	9
53	⟨(C:“question”)⟩ → ⟨(S:“Would you like to know more?”)⟩	0.36	10
54	⟨(C:“enter_action”)⟩ → ⟨(S:“Welcome.”)⟩	0.32	9
55	⟨(C:“action”)⟩ → ⟨(S:“Welcome.”)⟩	0.32	9
56	⟨(S:“assertion”)(C:“question”)⟩ → ⟨(S:“Yes, #other_reason_to_say_yes.”)⟩	0.30	8
57	⟨(C:“greeting”)⟩ → ⟨(S:“How may I help you?”)⟩	0.30	8
58	⟨(C:“greeting”)⟩ → ⟨(S:“Yes, #other_reason_to_say_yes.”)⟩	0.30	8
59	⟨(C:“enter_action”)⟩ → ⟨(S:“How may I help you?”)⟩	0.29	8
60	⟨(C:“action”)⟩ → ⟨(S:“How may I help you?”)⟩	0.29	8
61	⟨(C:“question”)(S:“assertion”)(C:“question”)⟩ → ⟨(S:“Would you like to know more?”)⟩	0.27	7
62	⟨(C:“question”)(S:“assertion”)(C:“question”)⟩ → ⟨(S:“Yes, #other_reason_to_say_yes.”)⟩	0.27	7
63	⟨(S:“assertion”)(C:“question”)⟩ → ⟨(S:“Would you like to know more?”)⟩	0.26	7
64	⟨(S:“assertion”)(C:“question”)⟩ → ⟨(S:“Yes.”)⟩	0.26	7
65	⟨(C:“question”)⟩ → ⟨(S:“Yes, that’s possible.”)⟩	0.25	7

Continued on next page

Table A.2 – *Continued from previous page*

#	**Sequential rule**	**conf.**	**supp.**
66	$\langle$(C:“question”)$\rangle \rightarrow \langle$(S:“Yes.”)$\rangle$	0.25	7

Appendix B

Dialogues

All the dialogues in the Pet Shop Game data set are listed here. They are formatted as they were used as input to the GSP algorithm. This means actions and utterances are flagged with C and S for customer and shopkeeper respectively and that all uses of the "other" option are replaced by the string "#other_*variable*" to reduce the search space.

Table B.1: Dialogue #1

#	**Dialogue turn**
1	C:"enter_action"
2	C:"Hello."
3	S:"Welcome."
4	C:"Do you sell dog?"
5	S:"Yes, #other_reason_to_say_yes."
6	C:"That's good.", C:"What can you tell me about dog?"
7	S:"A dog is a pet that is #other_pet_property."
8	C:"How much does a dog cost?"
9	S:"Would you like to know more?", S:"A dog costs #other_sum_of_money euro."
10	C:"No, I know enough.", C:"Give me the dog, please."
11	S:"That's good."
12	C:"pay_action"
13	S:"give_action"
14	C:"Goodbye.", C:"Thank you."
15	S:"Goodbye.", S:"At your service."
16	C:"leave_action"

Table B.2: Dialogue #2

#	**Dialogue turn**
1	C:"enter_action"
2	S:"Welcome.", S:"Hello."
3	C:"I would like to buy a dog.", C:"Hello."
4	S:"A dog costs #other_sum_of_money euro."
5	C:"No, that is not okay."

Continued on next page

Table B.2 – *Continued from previous page*

#	**Dialogue turn**
6	S:"We sell cat.", S:"A cat costs #other_sum_of_money euro."
7	C:"What can you tell me about cat?"
8	S:"A cat is a pet that is #other_pet_property.", S:"I'm not sure."
9	C:"Can you tell me if a cat is alert?", C:"That's good."
10	S:"Would you like to know more?", S:"Yes.", S:"A cat is a pet that is alert."
11	C:"Can you tell me if a cat is cheap in maintenance?", C:"Yes, I need to know more."
12	S:"A cat is a pet that is #other_pet_property.", S:"I'm not sure."
13	C:"pay_action"
14	S:"give_action"
15	C:"leave_action"

Table B.3: Dialogue #3

#	**Dialogue turn**
1	C:"enter_action"
2	S:"How may I help you?"
3	C:"I would like to buy a #other_type_of_animal."
4	S:"That's good."
5	C:"What pets do you sell?"
6	S:"We sell hamster."
7	C:"Can you tell me if a hamster is friendly?"
8	S:"Yes, #other_reason_to_say_yes."
9	C:"How much does a hamster cost?"
10	S:"A hamster costs 5 euro."
11	C:"That's good."
12	S:"give_action"
13	C:"Yes, #other_reason_to_say_yes."
14	S:"Would you like to know more?"
15	C:"No, I know enough."
16	S:"That's good."
17	C:"I would like to buy a hamster."
18	S:"That's good."
19	C:"Give me the hamster, please."
20	S:"Yes, #other_reason_to_say_yes."
21	C:"Yes, #other_reason_to_say_yes."
22	S:"Yes."
23	C:"pay_action"
24	S:"Would you like to know more?", S:"Thank you."
25	C:"Goodbye.", C:"No.", C:"Thank you."
26	S:"Goodbye."
27	C:"leave_action"

Continued on next page

Table B.4 – *Continued from previous page*

#	**Dialogue turn**

Table B.4: Dialogue #4

#	**Dialogue turn**
1	C:“enter_action”
2	S:“Welcome.”
3	C:“Can you tell me if a goldfish is talkative?”, C:“Hello.”
4	S:“No, #other_reason_to_say_no.”
5	C:“I am sorry.”, C:“Which pet is #other_pet_property?”
6	S:“A hamster is a pet that is cheap in maintenance.”, S:“A hamster costs 5 euro.”
7	C:“That’s good.”, C:“Give me the hamster, please.”
8	S:“give_action”
9	C:“pay_action”
10	S:“Thank you.”
11	C:“Goodbye.”, C:“Thank you.”
12	S:“Goodbye.”
13	C:“leave_action”

Table B.5: Dialogue #5

#	**Dialogue turn**
1	C:“enter_action”
2	S:“Welcome.”, S:“Hello.”
3	C:“Hello.”, C:“Which pet is alert?”
4	S:“Would you like to know more?”, S:“A dog is a pet that is alert.”
5	C:“Yes, I need to know more.”, C:“How much does a dog cost?”
6	S:“A dog costs #other_sum_of_money euro.”
7	C:“I am sorry.”, C:“That is too expensive.”
8	S:“Yes, that’s possible.”, S:“I am sorry.”
9	C:“Can you tell me if a dog is cheap in maintenance?”
10	S:“No, #other_reason_to_say_no.”
11	C:“What can you tell me about parrot?”
12	S:“A parrot is a pet that is cheap in maintenance.”, S:“A parrot is a pet that is talkative.”
13	C:“Can you tell me if a parrot is alert?”
14	S:“A parrot is a pet that is alert.”, S:“Yes.”
15	C:“How much does a parrot cost?”
16	S:“A parrot costs #other_sum_of_money euro.”
17	C:“pay_action”
18	S:“give_action”
19	C:“Thank you.”
20	S:“At your service.”, S:“Thank you.”
21	C:“leave_action”

Table B.6: Dialogue #6

#	**Dialogue turn**
1	C:“enter_action”
2	S:“Hello.”
3	C:“Hello.”
4	S:“How may I help you?”
5	C:“What pets do you sell?”
6	S:“We sell hamster.”
7	C:“What can you tell me about hamster?”
8	S:“A hamster is a pet that is cheap in maintenance.”, S:“Yes.”
9	C:“Can you tell me if a hamster is friendly?”
10	S:“A hamster is a pet that is friendly.”
11	C:“That’s good.”
12	S:“Would you like to know more?”
13	C:“Yes, I need to know more.”, C:“How much does a hamster cost?”
14	S:“A hamster is a pet that is cheap in maintenance.”, S:“A hamster costs 15 euro.”
15	C:“That’s good.”, C:“I would like to buy a hamster.”
16	S:“No, that’s not possible.”
17	C:“What do you mean?”
18	S:“A hamster costs 20 euro.”, S:“Yes.”, S:“Yes, that’s possible.”
19	C:“No, #other_reason_to_say_no.”
20	S:“Yes, that’s possible.”
21	C:“pay_action”
22	S:“give_action”
23	C:“Goodbye.”, C:“Thank you.”
24	S:“At your service.”, S:“Thank you.”
25	C:“leave_action”

Table B.7: Dialogue #7

#	**Dialogue turn**
1	C:“enter_action”
2	S:“Welcome.”, S:“Hello.”
3	C:“Hello.”, C:“What pets do you sell?”
4	S:“We sell cat.”, S:“We sell parrot.”, S:“We sell hamster.”
5	C:“How much does a parrot cost?”
6	S:“A parrot costs #other_sum_of_money euro.”, S:“A hamster costs 20 euro.”
7	C:“Can you tell me if a cat is friendly?”, C:“That is too expensive.”
8	S:“A cat is a pet that is friendly.”, S:“A cat costs #other_sum_of_money euro.”
9	C:“Can you tell me if a cat is cheap in maintenance?”, C:“I’m not sure.”
10	S:“A cat is a pet that is #other_pet_property.”
11	C:“Yes.”, C:“I would like to buy a hamster.”
12	S:“give_action”
13	C:“pay_action”
14	S:“Thank you.”
15	C:“Goodbye.”, C:“Thank you.”
16	S:“Goodbye.”

Continued on next page

Table B.7 – *Continued from previous page*

#	**Dialogue turn**
17	C:“leave_action”

Table B.8: Dialogue #8

#	**Dialogue turn**
1	C:“enter_action”
2	S:“Hello.”
3	C:“Hello.”, C:“What pets do you sell?”
4	S:“A cat is a pet that is faithful.”
5	C:“Can you tell me if a cat is friendly?”
6	S:“Yes.”
7	C:“That’s good.”, C:“How much does a cat cost?”
8	S:“A cat costs 20 euro.”
9	C:“No, that is not okay.”
10	S:“What do you mean?”
11	C:“That is too expensive.”
12	S:“No, #other_reason_to_say_no.”
13	C:“Yes.”, C:“I would like to buy a parrot.”
14	S:“Yes, #other_reason_to_say_yes.”
15	C:“How much does a parrot cost?”
16	S:“A parrot costs 15 euro.”
17	C:“pay_action”
18	S:“We sell #other_type_of_animal.”
19	C:“I don’t understand.”
20	S:“No, #other_reason_to_say_no.”
21	C:“No, #other_reason_to_say_no.”
22	S:“Yes, #other_reason_to_say_yes.”
23	C:“That’s good.”
24	S:“give_action”
25	C:“Goodbye.”
26	S:“Would you like to know more?”, S:“At your service.”
27	C:“leave_action”

Table B.9: Dialogue #9

#	**Dialogue turn**
1	C:“enter_action”
2	C:“Which pet is a good breed?”, C:“Hello.”
3	S:“Hello.”, S:“A #other_type_of_animal is a pet that is #other_pet_property.”
4	C:“Can you tell me if a #other_type_of_animal is cheap in maintenance?”
5	S:“No, #other_reason_to_say_no.”, S:“A #other_type_of_animal is a pet that is #other_pet_property.”
6	C:“Which pet is cheap in maintenance?”
7	S:“A goldfish is a pet that is #other_pet_property.”
8	C:“Can you tell me if a goldfish is #other_pet_property?”

Continued on next page

Table B.9 – *Continued from previous page*

#	**Dialogue turn**
9	S:“What do you mean?”
10	C:“Which pet is #other_pet_property?”
11	S:“A hamster is a pet that is #other_pet_property.”, S:“Yes, #other_reason_to_say_yes.”
12	C:“Yes, #other_reason_to_say_yes.”, C:“How much does a hamster cost?”
13	S:“A hamster costs 15 euro.”
14	C:“Can you tell me if a hamster is cheap in maintenance?”
15	S:“A hamster is a pet that is #other_pet_property.”, S:“Yes, #other_reason_to_say_yes.”
16	C:“That’s good.”, C:“Thank you.”, C:“Give me the #other_type_of_animal, please.”
17	S:“I am sorry.”, S:“Yes, #other_reason_to_say_yes.”, S:“We don’t have #other_type_of_animal in stock.”
18	C:“That’s good.”, C:“Thank you.”
19	S:“At your service.”
20	C:“pay_action”
21	S:“give_action”
22	C:“What can you tell me about #other_type_of_animal?”, C:“Thank you.”
23	S:“Yes, #other_reason_to_say_yes.”
24	C:“That’s good.”, C:“Goodbye.”, C:“Thank you.”
25	S:“Goodbye.”, S:“Thank you.”
26	C:“leave_action”

Table B.10: Dialogue #10

#	**Dialogue turn**
1	C:“enter_action”
2	S:“Hello.”
3	C:“Hello.”
4	S:“How may I help you?”
5	C:“What pets do you sell?”
6	S:“We sell hamster.”
7	C:“No, #other_reason_to_say_no.”, C:“Do you sell #other_type_of_animal?”
8	S:“No.”
9	C:“What pets do you sell?”
10	S:“We sell cat.”
11	C:“How much does a cat cost?”
12	S:“A cat costs 20 euro.”
13	C:“That’s good.”, C:“I would like to buy a cat.”
14	S:“That’s good.”
15	C:“pay_action”
16	S:“give_action”
17	C:“Goodbye.”, C:“Thank you.”
18	S:“At your service.”
19	C:“leave_action”

Table B.11: Dialogue #11

#	**Dialogue turn**
1	C:"enter_action"
2	C:"Hello."
3	S:"Welcome."
4	C:"Which pet is pettable?"
5	S:"A dog is a pet that is pettable.", S:"A cat is a pet that is pettable."
6	C:"How much does a cat cost?"
7	S:"No, that's not possible.", S:"We don't have cat in stock."
8	C:"What pets do you sell?"
9	S:"We sell goldfish."
10	C:"What do you mean?"
11	S:"We sell goldfish.", S:"A goldfish is a pet that is faithful."
12	C:"No, that is not okay."
13	S:"Would you like to know more?", S:"A goldfish is a pet that is small."
14	C:"No, I know enough.", C:"Goodbye."
15	S:"At your service."
16	C:"leave_action"

Table B.12: Dialogue #12

#	**Dialogue turn**
1	C:"enter_action"
2	C:"Hello."
3	S:"Hello.", S:"How may I help you?"
4	C:"What pets do you sell?"
5	S:"We sell cat.", S:"We sell goldfish.", S:"We sell dog.", S:"We sell hamster."
6	C:"Which pet is #other_pet_property?"
7	S:"A goldfish is a pet that is cheap in maintenance.", S:"What do you mean?"
8	C:"Give me the goldfish, please."
9	S:"Would you like to know more?", S:"Yes, that's possible.", S:"A goldfish costs 5 euro."
10	C:"No, I know enough.", C:"Thank you."
11	S:"give_action"
12	C:"Goodbye.", C:"Thank you."
13	S:"Yes, #other_reason_to_say_yes."
14	C:"pay_action"
15	S:"Goodbye.", S:"Thank you."
16	C:"Goodbye."
17	S:"Goodbye.", S:"At your service."
18	C:"leave_action"

Table B.13: Dialogue #13

#	**Dialogue turn**
1	C:"enter_action"
2	S:"Hello.", S:"How may I help you?"

Continued on next page

Table B.13 – *Continued from previous page*

#	**Dialogue turn**
3	C:“Hello.”, C:“What pets do you sell?”
4	S:“We sell cat.”, S:“We sell parrot.”, S:“We sell goldfish.”, S:“We sell dog.”, S:“We sell hamster.”
5	C:“That’s good.”, C:“Which pet is pettable?”
6	S:“A dog is a pet that is pettable.”, S:“A cat is a pet that is pettable.”
7	C:“Can you tell me if a dog is cheap in maintenance?”
8	S:“Would you like to know more?”, S:“Yes, #other_reason_to_say_yes.”, S:“A dog costs 5 euro.”
9	C:“What can you tell me about dog?”, C:“Yes, I need to know more.”
10	S:“A dog is a pet that is a good breed.”, S:“A dog is a pet that is friendly.”
11	C:“That’s good.”, C:“I would like to buy a dog.”, C:“Thank you.”
12	S:“A dog costs #other_sum_of_money euro.”, S:“Yes, that’s possible.”
13	C:“How much does a cat cost?”, C:“I don’t understand.”, C:“That is too expensive.”
14	S:“A cat costs #other_sum_of_money euro.”, S:“A cat is a pet that is cheap in maintenance.”
15	C:“Can you tell me if a cat is friendly?”
16	S:“A cat is a pet that is friendly.”, S:“Yes, that’s possible.”
17	C:“That’s good.”, C:“I would like to buy a cat.”, C:“Thank you.”
18	S:“Yes, #other_reason_to_say_yes.”, S:“Yes, that’s possible.”
19	C:“pay_action”
20	S:“give_action”
21	C:“Goodbye.”, C:“Thank you.”
22	S:“At your service.”
23	C:“leave_action”

Table B.14: Dialogue #14

#	**Dialogue turn**
1	C:“enter_action”
2	S:“Welcome.”, S:“How may I help you?”
3	C:“Hello.”, C:“What can you tell me about cat?”
4	S:“A cat is a pet that is #other_pet_property.”
5	C:“Can you tell me if a cat is cheap in maintenance?”
6	S:“A hamster is a pet that is cheap in maintenance.”, S:“A cat is a pet that is #other_pet_property.”, S:“A hamster costs 10 euro.”
7	C:“That’s good.”, C:“Give me the hamster, please.”
8	S:“give_action”
9	C:“pay_action”
10	S:“Yes, #other_reason_to_say_yes.”, S:“Thank you.”
11	C:“Goodbye.”, C:“Thank you.”
12	S:“Goodbye.”, S:“At your service.”
13	C:“leave_action”

Table B.15: Dialogue #15

#	**Dialogue turn**
1	C:“enter_action”
2	C:“Hello.”

Continued on next page

Table B.15 – *Continued from previous page*

#	**Dialogue turn**
3	S:“Welcome.”, S:“How may I help you?”
4	C:“I would like to buy a #other_type_of_animal.”, C:“Do you sell #other_type_of_animal?”
5	S:“No, that’s not possible.”, S:“We don’t have #other_type_of_animal in stock.”
6	C:“Do you sell #other_type_of_animal?”
7	S:“Would you like to know more?”, S:“A #other_type_of_animal is a pet that is pettable.”, S:“Yes.”
8	C:“How much does a #other_type_of_animal cost?”
9	S:“A #other_type_of_animal costs 10 euro.”
10	C:“That’s good.”, C:“I would like to buy a #other_type_of_animal.”
11	S:“Yes, that is okay.”, S:“A #other_type_of_animal costs 10 euro.”
12	C:“pay_action”
13	S:“give_action”
14	C:“Thank you.”
15	S:“At your service.”
16	C:“Goodbye.”
17	S:“Goodbye.”
18	C:“leave_action”

Table B.16: Dialogue #16

#	**Dialogue turn**
1	C:“enter_action”
2	C:“Can you tell me if a dog is friendly?”
3	S:“Yes, #other_reason_to_say_yes.”
4	C:“Yes, I need to know more.”, C:“Which pet is #other_pet_property?”
5	S:“A hamster is a pet that is cheap in maintenance.”
6	C:“Give me the hamster, please.”, C:“Thank you.”
7	S:“Would you like to know more?”
8	C:“No, I know enough.”, C:“Thank you.”
9	S:“A hamster costs 20 euro.”
10	C:“pay_action”
11	S:“Yes, #other_reason_to_say_yes.”
12	C:“Yes, #other_reason_to_say_yes.”
13	S:“Yes, #other_reason_to_say_yes.”
14	C:“No, #other_reason_to_say_no.”
15	S:“Yes, #other_reason_to_say_yes.”
16	C:“Yes.”, C:“Thank you.”
17	S:“give_action”
18	C:“leave_action”

Table B.17: Dialogue #17

#	**Dialogue turn**
1	C:“enter_action”
2	S:“Welcome.”, S:“Hello.”

Continued on next page

Table B.17 – *Continued from previous page*

#	**Dialogue turn**
3	C:“Which pet is friendly?”, C:“I would like to buy a #other_type_of_animal.”, C:“Hello.”, C:“Which pet is #other_pet_property?”
4	S:“A hamster is a pet that is #other_pet_property.”
5	C:“No, #other_reason_to_say_no.”, C:“I’m not sure.”
6	S:“A dog is a pet that is #other_pet_property.”, S:“Yes, that’s possible.”
7	C:“Can you tell me if a dog is pettable?”
8	S:“A dog is a pet that is pettable.”, S:“Yes.”
9	C:“That’s good.”, C:“I would like to buy a dog.”
10	S:“A dog costs #other_sum_of_money euro.”
11	C:“Yes, that is okay.”, C:“Give me the dog, please.”
12	S:“give_action”
13	C:“pay_action”
14	S:“Thank you.”
15	C:“leave_action”

Table B.18: Dialogue #18

#	**Dialogue turn**
1	C:“enter_action”
2	S:“Hello.”, S:“How may I help you?”
3	C:“What can you tell me about goldfish?”
4	S:“A goldfish costs 10 euro.”, S:“A goldfish is a pet that is faithful.”
5	C:“Can you tell me if a goldfish is talkative?”
6	S:“A goldfish is a pet that is talkative.”, S:“Yes, that’s possible.”
7	C:“Yes, #other_reason_to_say_yes.”, C:“Give me the goldfish, please.”
8	S:“Yes, that is okay.”, S:“A goldfish costs #other_sum_of_money euro.”
9	C:“pay_action”
10	S:“give_action”
11	C:“leave_action”

Table B.19: Dialogue #19

#	**Dialogue turn**
1	C:“enter_action”
2	C:“Hello.”, C:“Do you sell goldfish?”
3	S:“Welcome.”, S:“Would you like to know more?”, S:“Yes, that’s possible.”
4	C:“Yes, #other_reason_to_say_yes.”
5	S:“I don’t understand.”, S:“What do you mean?”
6	C:“Yes, #other_reason_to_say_yes.”
7	S:“A goldfish is a pet that is friendly.”, S:“Yes.”
8	C:“How much does a goldfish cost?”
9	S:“A goldfish costs 5 euro.”
10	C:“Yes, #other_reason_to_say_yes.”
11	S:“Yes, that’s possible.”, S:“A #other_type_of_animal costs 10 euro.”, S:“A #other_type_of_animal costs 20 euro.”

Continued on next page

Table B.19 – *Continued from previous page*

#	**Dialogue turn**
12	C: "Yes, #other_reason_to_say_yes."
13	S: "Yes, #other_reason_to_say_yes."
14	C: "Yes, #other_reason_to_say_yes."
15	S: "A goldfish is a pet that is cheap in maintenance.", S: "A #other_type_of_animal costs 5 euro."
16	C: "Yes, #other_reason_to_say_yes."
17	S: "That's good.", S: "Yes, #other_reason_to_say_yes."
18	C: "Yes, #other_reason_to_say_yes.", C: "Thank you."
19	S: "Yes, #other_reason_to_say_yes."
20	C: "Thank you."
21	S: "Yes, #other_reason_to_say_yes."
22	C: "pay_action"
23	S: "give_action"
24	C: "Goodbye."
25	S: "Goodbye.", S: "At your service.", S: "Yes, #other_reason_to_say_yes."
26	C: "Yes, #other_reason_to_say_yes."
27	S: "That's good."
28	C: "leave_action"

Table B.20: Dialogue #20

#	**Dialogue turn**
1	C: "enter_action"
2	C: "Hello."
3	S: "Welcome.", S: "How may I help you?", S: "We sell hamster."
4	C: "How much does a hamster cost?"
5	S: "A hamster is a pet that is cheap in maintenance.", S: "A hamster costs 5 euro."
6	C: "Do you sell cat?", C: "Do you sell parrot?", C: "Do you sell dog?", C: "I'm not sure."
7	S: "A hamster is a pet that is small.", S: "Yes, that's possible.", S: "A hamster is a pet that is friendly."
8	C: "How much does a cat cost?", C: "Thank you."
9	S: "A cat costs 20 euro.", S: "A hamster costs #other_sum_of_money euro.", S: "A hamster is a pet that is pettable."
10	C: "leave_action"

Table B.21: Dialogue #21

#	**Dialogue turn**
1	C: "enter_action"
2	S: "Welcome.", S: "Hello."
3	C: "Thank you."
4	S: "How may I help you?"
5	C: "Can you tell me if a goldfish is talkative?"
6	S: "No, #other_reason_to_say_no.", S: "A parrot is a pet that is talkative."
7	C: "Thank you."
8	S: "Would you like to know more?"

Continued on next page

Table B.21 – *Continued from previous page*

#	**Dialogue turn**
9	C:“Yes, I need to know more.”
10	S:“A goldfish is a pet that is friendly.”, S:“A goldfish is a pet that is cheap in maintenance.”
11	C:“How much does a goldfish cost?”
12	S:“A goldfish costs 10 euro.”
13	C:“I would like to buy a goldfish.”
14	S:“give_action”
15	C:“pay_action”
16	S:“Thank you.”
17	C:“Goodbye.”
18	S:“Goodbye.”
19	C:“leave_action”

Table B.22: Dialogue #22

#	**Dialogue turn**
1	C:“enter_action”
2	S:“How may I help you?”
3	C:“Hello.”, C:“Do you sell dog?”
4	S:“Yes, #other_reason_to_say_yes.”
5	C:“That’s good.”, C:“I would like to buy a dog.”
6	S:“Yes, #other_reason_to_say_yes.”
7	C:“Yes, that is okay.”
8	S:“A dog costs 15 euro.”
9	C:“I’m not sure.”
10	S:“A dog is a pet that is faithful.”
11	C:“That’s good.”
12	S:“Would you like to know more?”
13	C:“No, I know enough.”, C:“Thank you.”
14	S:“How may I help you?”
15	C:“pay_action”
16	S:“give_action”
17	C:“Thank you.”
18	S:“Thank you.”
19	C:“leave_action”

Table B.23: Dialogue #23

#	**Dialogue turn**
1	C:“enter_action”
2	S:“Hello.”
3	C:“Hello.”
4	S:“How may I help you?”
5	C:“Which pet is faithful?”, C:“Which pet is big?”, C:“Which pet is alert?”, C:“Do you sell dog?”
6	S:“We don’t have dog in stock.”
7	C:“Do you sell #other_type_of_animal?”

Continued on next page

Table B.23 – *Continued from previous page*

#	**Dialogue turn**
8	S:“No, #other_reason_to_say_no.”
9	C:“Which pet is faithful?”, C:“Which pet is big?”, C:“Which pet is alert?”, C:“What pets do you sell?”
10	S:“We sell goldfish.”, S:“A goldfish costs 5 euro.”
11	C:“I'm not sure.”, C:“I would like to buy a #other_type_of_animal.”, C:“Do you sell #other_type_of_animal?”
12	S:“No, #other_reason_to_say_no.”
13	C:“Thank you.”, C:“I am sorry.”, C:“Give me the #other_type_of_animal, please.”
14	S:“Would you like to know more?”, S:“That's good.”
15	C:“Can you tell me if a #other_type_of_animal is alert?”
16	S:“Yes, #other_reason_to_say_yes.”
17	C:“Can you tell me if a #other_type_of_animal is friendly?”
18	S:“Yes, #other_reason_to_say_yes.”
19	C:“Give me the #other_type_of_animal, please.”
20	S:“Yes, #other_reason_to_say_yes.”
21	C:“Goodbye.”, C:“Thank you.”
22	S:“At your service.”
23	C:“leave_action”

Table B.24: Dialogue #24

#	**Dialogue turn**
1	C:“enter_action”
2	S:“Hello.”, S:“How may I help you?”
3	C:“Hello.”, C:“What pets do you sell?”
4	S:“Yes, #other_reason_to_say_yes.”, S:“We sell #other_type_of_animal.”
5	C:“What can you tell me about goldfish?”
6	S:“A goldfish is a pet that is #other_pet_property.”, S:“Yes, #other_reason_to_say_yes.”
7	C:“That's good.”, C:“How much does a goldfish cost?”
8	S:“A goldfish costs #other_sum_of_money euro.”
9	C:“Yes, #other_reason_to_say_yes.”, C:“That's good.”
10	S:“Yes, #other_reason_to_say_yes.”
11	C:“pay_action”
12	S:“give_action”
13	C:“Goodbye.”, C:“Thank you.”
14	S:“Goodbye.”, S:“Thank you.”
15	C:“leave_action”

Table B.25: Dialogue #25

#	**Dialogue turn**
1	C:“enter_action”
2	C:“Hello.”
3	S:“Welcome.”, S:“Hello.”
4	C:“Which pet is faithful?”, C:“Thank you.”

Continued on next page

Table B.25 – *Continued from previous page*

#	**Dialogue turn**
5	S:“A cat is a pet that is #other_pet_property.”, S:“A dog is a pet that is faithful.”
6	C:“Can you tell me if a cat is alert?”
7	S:“Yes.”, S:“A cat is a pet that is alert.”
8	C:“I'm not sure.”, C:“Can you tell me if a cat is #other_pet_property?”
9	S:“That's good.”, S:“Yes, #other_reason_to_say_yes.”
10	C:“What can you tell me about #other_type_of_animal?”
11	S:“A cat is a pet that is #other_pet_property.”
12	C:“Yes, #other_reason_to_say_yes.”, C:“That's good.”, C:“How much does a cat cost?”
13	S:“A cat costs #other_sum_of_money euro.”
14	C:“No, #other_reason_to_say_no.”, C:“No, that's not possible.”, C:“That is too expensive.”
15	S:“No.”, S:“No, that's not possible.”, S:“No, that is not okay.”, S:“I don't understand.”, S:“A cat costs #other_sum_of_money euro.”
16	C:“No, #other_reason_to_say_no.”, C:“I am sorry.”
17	S:“A cat costs #other_sum_of_money euro.”, S:“No, #other_reason_to_say_no.”, S:“A hamster costs 10 euro.”
18	C:“Can you tell me if a hamster is faithful?”, C:“Can you tell me if a hamster is #other_pet_property?”
19	S:“A hamster is a pet that is faithful.”, S:“A hamster is a pet that is #other_pet_property.”, S:“Yes, #other_reason_to_say_yes.”
20	C:“Can you tell me if a hamster is alert?”, C:“I'm not sure.”
21	S:“A hamster is a pet that is #other_pet_property.”, S:“Yes, #other_reason_to_say_yes.”, S:“A #other_type_of_animal is a pet that is faithful.”, S:“A #other_type_of_animal is a pet that is #other_pet_property.”
22	C:“Can you tell me if a #other_type_of_animal is #other_pet_property?”, C:“I'm not sure.”
23	S:“Yes, #other_reason_to_say_yes.”
24	C:“Yes, #other_reason_to_say_yes.”, C:“How much does a #other_type_of_animal cost?”
25	S:“A #other_type_of_animal costs 15 euro.”, S:“Yes, #other_reason_to_say_yes.”
26	C:“Yes, #other_reason_to_say_yes.”, C:“Yes, that is okay.”, C:“Give me the #other_type_of_animal, please.”
27	S:“Yes, that is okay.”, S:“Yes, that's possible.”
28	C:“Yes, #other_reason_to_say_yes.”, C:“Thank you.”
29	S:“give_action”
30	C:“pay_action”
31	S:“Yes, #other_reason_to_say_yes.”, S:“Thank you.”
32	C:“Goodbye.”, C:“Thank you.”
33	S:“Goodbye.”, S:“No, #other_reason_to_say_no.”
34	C:“leave_action”

Table B.26: Dialogue #26

#	**Dialogue turn**
1	C:“enter_action”
2	S:“Hello.”
3	C:“Hello.”, C:“Do you sell #other_type_of_animal?”
4	S:“Yes, #other_reason_to_say_yes.”
5	C:“Which pet is #other_pet_property?”

Continued on next page

Table B.26 – *Continued from previous page*

#	**Dialogue turn**
6	S:“Yes, #other_reason_to_say_yes.”
7	C:“How much does a dog cost?”
8	S:“A dog costs 20 euro.”
9	C:“That’s good.”, C:“I would like to buy a dog.”
10	S:“Would you like to know more?”
11	C:“No, I know enough.”, C:“Give me the dog, please.”
12	S:“Yes, #other_reason_to_say_yes.”
13	C:“I am sorry.”, C:“What do you mean?”, C:“I don’t understand.”
14	S:“Yes, #other_reason_to_say_yes.”
15	C:“Yes, that is okay.”, C:“Give me the dog, please.”
16	S:“give_action”
17	C:“pay_action”
18	S:“Goodbye.”, S:“Yes, #other_reason_to_say_yes.”
19	C:“Goodbye.”, C:“Thank you.”
20	S:“Goodbye.”
21	C:“leave_action”

Table B.27: Dialogue #27

#	**Dialogue turn**
1	C:“enter_action”
2	S:“Welcome.”, S:“How may I help you?”
3	C:“What pets do you sell?”
4	S:“We sell #other_type_of_animal.”
5	C:“What can you tell me about #other_type_of_animal?”
6	S:“A dog is a pet that is #other_pet_property.”
7	C:“How much does a dog cost?”
8	S:“A dog costs #other_sum_of_money euro.”
9	C:“No, that’s not possible.”
10	S:“A dog costs #other_sum_of_money euro.”, S:“Yes, #other_reason_to_say_yes.”, S:“Yes, that’s possible.”
11	C:“Can you tell me if a cat is #other_pet_property?”
12	S:“A cat costs #other_sum_of_money euro.”, S:“Yes, #other_reason_to_say_yes.”
13	C:“I would like to buy a hamster.”, C:“That is too expensive.”
14	S:“Yes, that is okay.”, S:“A hamster costs 10 euro.”
15	C:“pay_action”
16	S:“give_action”
17	C:“Goodbye.”, C:“Thank you.”
18	S:“Goodbye.”
19	C:“leave_action”

Table B.28: Dialogue #28

#	**Dialogue turn**
1	C:“enter_action”

Continued on next page

Table B.28 – *Continued from previous page*

#	**Dialogue turn**
2	S:“Welcome.”, S:“How may I help you?”
3	C:“Hello.”, C:“Which pet is #other_pet_property?”
4	S:“A goldfish is a pet that is #other_pet_property.”, S:“I’m not sure.”
5	C:“No, that is not okay.”, C:“I would like to buy a #other_type_of_animal.”, C:“What do you mean?”
6	S:“No, #other_reason_to_say_no.”
7	C:“Do you sell #other_type_of_animal?”
8	S:“Yes, #other_reason_to_say_yes.”
9	C:“How much does a #other_type_of_animal cost?”
10	S:“A #other_type_of_animal costs #other_sum_of_money euro.”
11	C:“How much does a #other_type_of_animal cost?”
12	S:“Yes, #other_reason_to_say_yes.”
13	C:“No, #other_reason_to_say_no.”
14	S:“A #other_type_of_animal costs 20 euro.”
15	C:“Yes, #other_reason_to_say_yes.”, C:“No, that is not okay.”
16	S:“Yes, that is okay.”
17	C:“That’s good.”, C:“Give me the #other_type_of_animal, please.”
18	S:“No, #other_reason_to_say_no.”
19	C:“Yes, #other_reason_to_say_yes.”
20	S:“That’s good.”
21	C:“Yes, #other_reason_to_say_yes.”
22	S:“give_action”
23	C:“pay_action”
24	S:“Goodbye.”, S:“Thank you.”
25	C:“Goodbye.”, C:“Thank you.”, C:“Can you tell me if a #other_type_of_animal is #other_pet_property?”
26	S:“No, #other_reason_to_say_no.”
27	C:“How much does a #other_type_of_animal cost?”, C:“Yes, I need to know more.”, C:“No, that’s not possible.”
28	S:“No, #other_reason_to_say_no.”, S:“A #other_type_of_animal costs 20 euro.”
29	C:“No, that is not okay.”, C:“No, #other_reason_to_say_no.”, C:“Goodbye.”
30	S:“Goodbye.”, S:“Yes, #other_reason_to_say_yes.”
31	C:“leave_action”

Bibliography

[1] Rakesh Agrawal, Tomasz Imieliński, and Arun Swami. Mining association rules between sets of items in large databases. *SIGMOD Rec.*, 22(2):207–216, June 1993.

[2] James Allen, Donna Byron, Myroslava Dzikovska, George Ferguson, Lucian Galescu, and Amanda Stent. An architecture for a generic dialogue shell, 2000.

[3] J.L. Austin. *How to do things with words.* Oxford University Press, 1962.

[4] G. Frege. Über Sinn und Bedeutung. In Mark Textor, editor, *Funktion - Begriff - Bedeutung*, volume 4 of *Sammlung Philosophie*. Vandenhoeck & Ruprecht, Göttingen, 1892.

[5] Jesse James Garrett. Ajax: A new approach to web applications. `http://www.adaptivepath.com/ideas/ajax-new-approach-web-applications`, 2005. [Online; accessed 4-August-2012].

[6] Dong (Haoyuan) Li, Anne Laurent, and Pascal Poncelet. Mining unexpected sequential patterns and implication rules. In *Rare Association Rule Mining and Knowledge Discovery: Technologies for Infrequent and Critical Event Detection*, pages 150–167. Kluwer, 2009.

[7] Jeff Orkin and Deb Roy. Automatic Learning and Generation of Social Behavior from Collective Human Gameplay, 2009.

[8] Jeff Orkin and Deb Roy. Semi-Automated Dialogue Act Classification for Situated Social Agents in Games, 2010.

[9] Jeff Orkin, Tynan Smith, Hilke Reckman, and Deb Roy. Semi-Automatic Task Recognition for Interactive Narratives with EAT & RUN, 2010.

[10] Jeff Orkin, Tynan Smith, and Deb Roy. Behavior Compilation for AI in Games, 2010.

[11] Alex Russell. Comet: Low latency data for the browser. `http://infrequently.org/2006/03/comet-low-latency-data-for-the-browser/`, 2006. [Online; accessed 4-August-2012].

[12] John R. Searle. A taxonomy of illocutionary acts. In Keith Gunderson, editor, *Language, Mind and Knowledge, Minnesota Studies in the Philosophy of Science*, volume VII, pages 344–369. University of Minnesota Press, Amsterdam, 1975. Also appears in John R. Searle, *Expression and Meaning: Studies in the Theory of Speech Acts*, Cambridge University Press, 1979.

[13] Ben Shneiderman and Catherine Plaisant. *Designing the User Interface: Strategies for Effective Human-Computer Interaction (4th Edition)*. Pearson Addison Wesley, 2004.

[14] Ramakrishnan Srikant and Rakesh Agrawal. Mining sequential patterns: Generalizations and performance improvements. In Peter M. G. Apers, Mokrane Bouzeghoub, and Georges Gardarin, editors, *EDBT*, volume 1057 of *Lecture Notes in Computer Science*, pages 3–17. Springer, 1996.

[15] D. Walton and E. Krabbe. *Commitment in Dialogue: Basic concept of interpersonal reasoning*. State University of New York Press, Albany NY, 1995.

[16] Joseph Weizenbaum. ELIZA—a computer program for the study of natural language communication between man and machine. *Commun. ACM*, 9(1):36–45, January 1966.