PONTIFICIA UNIVERSIDAD CATOLICA DE CHILE
SCHOOL OF ENGINEERING

AN ADAPTIVE READING TUTOR USING MARKOV MODELS TO DEVELOP PHONOLOGICAL AWARENESS

ESTEBAN ANDRÉS HURTADO LEÓN

Thesis submitted to the Office of Research and Graduate Studies
in partial fulfillment of the requirements for the degree of
Master of Science in Engineering

Advisor:
ALVARO SOTO

Santiago de Chile, December 2008

PONTIFICIA UNIVERSIDAD CATOLICA DE CHILE
SCHOOL OF ENGINEERING

AN ADAPTIVE READING TUTOR USING MARKOV MODELS TO DEVELOP PHONOLOGICAL AWARENESS

ESTEBAN ANDRÉS HURTADO LEÓN

Members of the Committee:
ALVARO SOTO
DOMINGO MERY
RICARDO ROSAS
MARCELO GUARINI

Thesis submitted to the Office of Research and Graduate Studies
in partial fulfillment of the requirements for the degree of
Master of Science in Engineering

Santiago de Chile, December 2008

To the memory of Álvaro Campos

ACKNOWLEDGEMENTS

During the development of this work, I had the luck of counting on people that have been very influential for me. I would like to thank my advisor, Álvaro Soto, for providing all the guidance I needed without ever being too restrictive. I would also like to thank Ricardo Rosas, who introduced me to cognitive psychology and to research in that field some years ago. Many ideas in this work are inspired on what I learnt from him.

This work would not have been possible without the lessons and support of Agustín Ibañez; the kind help of Ana Araneda; and the great people at Colegio Calasanz, especially Marcela Gutierrez, Andrea Ferrada, and José Aravena.

I would like to express my appreciation to María Luisa Ilabaca, Gloria González, and Álvaro Navarro, for their help during the experiment; to Jacqueline Köhler and Josefina Escobar for their unconditional support; and with a very deep feeling, to my family.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

Phonological awareness, a form of sensitivity to the sound structure of the language, is a key skill in the development of reading. It has to be explicitly taught and, ideally, the needs of each child have to be addressed individually. In a classroom environment this level of individualized attention is often impractical. Therefore, an informatized tutoring solution that adapts automatically to each child would be beneficial. Traditionally, intelligent tutoring systems (ITS) have been used to teach in domains where student and/or teacher meta-cognition can be modeled. This document reports the development and testing of an adaptive intelligent tutor that can be used in domains like phonological awareness where a meta-cognitive model does not exist. The design is a natural application to education of current technology for decision taking under uncertainty. Results show that the adaptive tutor is effective in teaching phonological awareness to kindergarten children. This is not observed with a non-adaptive tutor which is otherwise similar.

Keywords: phonological awareness, ITS, intelligent tutoring system, uncertainty, MDP, POMDP, education.

RESUMEN

La conciencia fonológica, que es cierta forma de sensibilidad a la estructura sonora del lenguage, es una habilidad fundamental en el desarrollo de la lectura. Debe ser enseñada explícitamente y, de ser posible, de manera individualizada. En una sala de clase, es difícil entregar a cada niño atención personalizada, por lo cual resultaría beneficioso el contar con un sistema de enseñanza informatizado que se adapte automáticamente a cada niño. Tradicionalmente, los sistemas tutores inteligentes (ITS, por sus siglas en inglés), han sido utilizados para enseñar en dominios donde existe un modelo de la metacognición del alumno y/o del profesor. El presente documento es el reporte del desarrollo y evaluación de un tutor inteligente adaptativo que puede ser utilizado en dominios como la conciencia fonológica, donde no existe un modelo metacognitivo. El diseño utilizado es una aplicación natural a la educación de la tecnología actual para la toma de decisiones bajo incerteza. Los resultados muestran que el tutor adaptativo es efectivo en la enseñanza de conciencia fonológica a niños de kindergarten. Este resultado no se observa en un tutor no adaptivo que en sus demás aspectos es similar al anterior.

Palabras Claves: conciencia fonológica, ITS, sistema tutor inteligente, incerteza, MDP, POMDP, educación.

1. INTRODUCTION

Reading is considered to be a complex skill, since it involves different cognitive processes. A reader needs perceptual skills in order to acquire information from a visual a representation of language. Also, at some level, a sensitivity to syntactical structure (i.e. grammar) is needed and, more obscurely, processes allowing the reader to understand what he or she is reading must take place. This ability to comprehend, while not specific to written language, is necessary for successful reading.

A first step towards building tools for helping and improving the reading acquisition process in children would seem to be modeling this process so to understand how to modify it for better. Currently, it has not been possible build a computational model that accounts for the complexity of human language, especially comprehension (Kronmüller & Cornejo, 2008). This makes it difficult to explain the specific aspect of literacy and its acquisition in enough detail for building accurate simulations. Nevertheless, some well defined skills under the name of *phonological processing* have been demonstrated to be related to reading (Passenger, Stuart, & Terrell, 2002; Wagner, Torgesen, & Rashotte, 1994).

The idea has been around for some time that there is a connection between sensitivity to sound components of spoken language and the development of reading skills, for example in the work of Liberman (Liberman, 1973). Along with short and long term phonological memory, phonological awareness is a phonological processing skill which has been related to reading acquisition by several research efforts (Simmons, Singleton, & Horne, 2008; Jiménez & Venegas, 2004).

Gombert (Gombert, 1992), has defined phonological awareness (PA) as the ability to identify the phonological components of linguistic units, and operating on them at will. Being able to spell words, split them in syllables, tell whether two word do rhyme or replacing the sound of one letter for other when speaking are examples of this ability. Since such forms of segmentation do not arise naturally in spoken language, phonological awareness has to be explicitly taught to children. For instance, even though the word ‘cat’ can be split into the three different consecutive sounds ‘c’, ‘a’, and ‘t’, the acoustic stimulus of that spoken word does not provide enough information to uniquely determine such segmentation (Ball & Blachman, 1991).

It is a common opinion that PA is a prerequisite for learning to read (Juel, 1988; Tunmer, Herriman, & Nesdale, 1988). Unfortunately, it takes a whole deal of effort for children to learn it (Liberman, 1983; Liberman, Shankweiler, Fischer, & Carter, 1974). The relationship between PA and literacy is a non trivial fact. Not only children gain better PA skills as a result of learning to read (Liberman, 1983; Liberman et al., 1974; Bradley & Bryant, 1983), but also non readers will learn to read easier and better if they learn PA skills (Bradley & Bryant, 1991). Based on this discovery, several activities have been designed for helping children to learn reading, requiring no detailed knowledge on how the learning process takes place.

As it can be inferred from longitudinal studies on this topic (Bradley & Bryant, 1983; Roth, Speece, & Cooper, 2002; Storch & Whitehurst, 2002; Wagner et al., 1994), different children have different skill levels with respect phonological awareness. Therefore, any informatized alternative to human teaching of PA, must be adaptive in nature so to address the specific needs of each child. Failing to provide each child with activities of appropriate difficulty levels can lead to boredom or frustration, possibly generating a negative attitude towards literacy, and anyways not helping much with its acquisition.

In particular, the present work explores a way of using adaptive software for providing children with PA training in an informatized way. Although autonomous, it is not intended to replace teachers for this task, which would be an enormously more ambitious goal. The goal is to propose a tool that can bridge the gap that generates between a single teacher and a class consisting of children with different needs. We present an adaptive software tool that can be used as didactic material, dealing with children diversity and, hopefully, helping them develop a more even level of PA competencies.

A phonological awareness tutor was implemented in the form of a multiple choice test that teaches by asking questions. There is a finite set of questions and the tutor must “decide” at each time which will be the most beneficial at a given time. Questions of an adequate difficulty, challenging but not impossible to answer, will motivate the acquisition of new knowledge. In order to give the tutor this adaptive capability, a Markovian decision taking model was built so to make efficient use of available information. The result was a video-game that was tested on children beginning primary education.

In the following sections previous work will be reviewed and more details will be given on Markovian decision taking. Then, the design of a PA tutor software will be presented including a Markovian decision taking model. Finally, a study that assessed this tutor with children will be discussed.

2. PREVIOUS WORK

The category of software that can be used for learning is probably best represented by what has been called Intelligent Tutoring Systems (ITS), applications of artificial intelligence to education (Nwana, 1990). It is a diverse class of software, but traditionally an ITS is designed around four aspects: a domain model, a student model, a teaching model, and a user interface (Freedman, 2000). As a consequence of explicit modelling, facts about each of this four aspects are often expressed by means of a formal language.

The development of ITS technology begun with computer aided instruction (CAI) efforts in the 1960's, in the form of *generative* tutors that automatically generated problems for students to solve, and tabulated responses for computing performance measures (Uhr, 1969). In the 1970's the student appeared as a relevant factor in the process. Responses were not only being tabulated, but also considered for the presentation of new material. Attention shifted to the more ambitious goal of modeling human tutors and students (Corbett, Koedinger, & Anderson, 1997). This is consistent with a crisis in education and artificial intelligence occurring at the same time.

In 1975 Chomsky published his 1957 doctoral thesis (Chomsky, 1975) containing an influential and devastating criticism of behaviorism, a previously dominant school proposing, among other things, that behavior can be studied without recourse to internal descriptions of mind. By the same time Newel and Simon (Newell & Simon, 1972) established the foundations for an information processing approach to the understanding of human cognition. It was believed that intelligence and formal symbol processing were essentially the same (Newell & Simon, 1976). This motivated an optimistic research program, not only about looking into the mind, but also about modelling its processes by means of formal languages.

In the 1980's the term *student model* was explicitly used in the ITS literature, considering three aspects: an *overlay* model that represents what the student knows as a subset of expert knowledge; a *differential* model containing the difference between the overlay and expert knowledge, and a *perturbation* model that indicates student misconceptions. It was acknowledged, however, that the communication between a human tutor and his or her student is partially implicit, but this was not seen as an intrinsic limitation (Sleeman & Brown, 1982).

At present, things have not changed much for ITS. Student modelling and overlay models have remained a fundamental aspect of ITS design, giving importance to assumptions about how students think and learn (Murray, 1999; Koedinger & Corbett, 2006). However, current advances in mind sciences suggest that such approaches, rooted in classical artificial intelligence, might not point to the right direction. Today, it is accepted that the information processing approach to human cognition has not accomplished its goals and is intrinsically limited (Bruner, 1990; Kronmüller & Cornejo, 2008; Ibañez, 2008). But this is the perspective taken when an ITS models student or teacher mind processes.

Apart from that, computer science has also produced advances, particularly in the field of machine learning. Surprise has been expressed for the limited use of machine learning methods in ITS, in comparison to other fields (Hämäläinen & Vinni, 2006; Hamburger & Tecuci, 1998). There has been some research on machine learning methods for classification of educational data. In distance learning, the Naive Bayes algorithm has proven useful for predicting student dropout (Kotsiantis, Pierrakeas, & Pintelas, 2003); while the combination of multiple classifiers, and feature weighting by means of a genetic algorithm have been used as successful strategies for predicting student final grades from logged data in an educational web application (Minaei-Bidgoli, Kashy, Kortemeyer, & Punch, 2003). The big size of distance learning datasets is a key aspect for this research. Other educational data sets are typically much smaller. A study dealing with this problem concluded that the Naive Bayes approach is applicable in such cases after careful preprocessing (Hämäläinen & Vinni, 2006).

Research on classification of educational data is important for ITS, since before taking a tutoring action, the situation must be classified. The fact that the Naive Bayes classifier succeeds consistently is insightful. It derives from Bayesian Networks, a method that deals well with uncertainty because it optimizes the use of available information. At the same time, it requires less parameters than a Bayesian Network, which leads to generality because less knowledge is needed about the domain that is being modelled, and more importantly, reduces the chance of overfitting with small datasets.

Though important, this research is only about solving basic problems of ITS. Additional work is needed to integrate this solutions into working systems ready to be used at classrooms. Unfortunately, the transition from the laboratory to the classroom has been more often than not, a difficult one (Lester et al., 2001). A machine learning based ITS, capable of taking intelligent decisions in the time span of a learning session would need to deal with uncertainty in assessing student knowledge and outcomes of tutor actions. With finite knowledge states and decisions being taken regularly, this problem fits into the description of a discrete state, discrete time, stochastic process, well modeled by a partially observable Markov decision process (POMDP) (Kaelbling, Littman, & Cassandra, 1998; Cassandra, Kaelbling, & Littman, 1994).

An intelligent tutor that, at a general level, is similar to the one developed and tested as part of the present work was described by Cassandra in 1998 as a possible application of POMDP models to education (Cassandra, 1998). Nevertheless, to our knowledge it has not been taken to realization before. Current efforts in combining machine learning and intelligent tutors seem to be more focused on problems not directly related to decision taking under uncertainty, for example user interface. An innovative proposal in this direction is the LISTEN project (Mostow, 2008; Zhang et al., 2008; Aist, 2000; Mostow, Hauptmann, Chase, & Roth, 1993) which applies speech recognition technology to the design of reading tutors able to classify student speech and detect errors.

A question remains open on how to step away from outdated ideas in human cognition and at the same time apply current advances in decision taking under uncertainty to the construction of intelligent tutors. How can tutoring systems be built without recourse to computationalist models of human mind, while taking advantage of current machine learning technology? And what would be the benefit if any?

In order to explore this question, we have built a tutor for teaching phonological awareness. This would be challenging from a current ITS paradigm, since it is difficult to model the acquisition process for such a skill which is not of meta-cognitive nature. As it will be shown, if the attention is shifted to modelling the learning domain instead of the way a student or teacher thinks about it, an adaptive model can be generated for a tutoring system. Markovian decision taking provides a framework for creating models with parameters that can be deduced from the way the tutor interacts with the student, removing the need of learning parameters from a big dataset.

3. GAMING METAPHOR

The tutor was implemented as a video-game that takes a child to travel from the Earth trough the space to planet Mars. A little spaceship is presented on screen and controlled by the user with the mouse. The ship has an on-board computer that gives directions about how to play when the game starts. In a first stage, the child travels from the Earth to the space. A background slides to the left of the screen to simulate that the child's spaceship is traveling to the right. In each question asked by the tutor software, the sound of a letter is played back and a column made up of three letters appears from the right moving to the

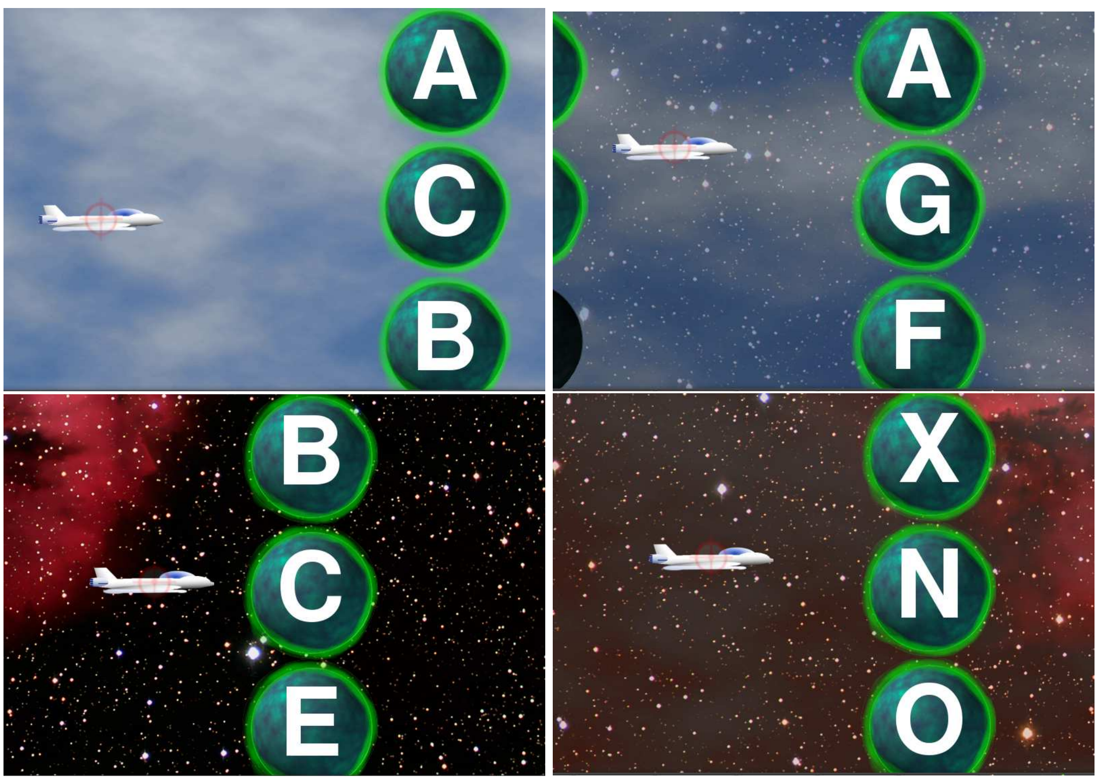


FIGURE 3.1. Screenshots from the phonological awareness game. Both variants, adaptive and non-adaptive, look the same. In the first stage the user is given the mission of travelling from Earth to the outer space. Auditive stimuli are letter sounds. The background shows clouds that slide to the left simulating a trip near the Earth surface. As the game advances, clouds begin to fade out, and stars begin to appear. During the second stage the mission is to get to Mars. Full word sounds are played instead of letters and the background simulates the outer space, with gradually appearing red clouds.

left. The ship will necessarily collide with one of the letters. If a collision occurs with a letter that does not correspond to the auditive stimulus, the ship will crash. But if the child guides the ship so that it collides with the only right letter, a short positive feedback will be displayed and no crash will occur. After three consecutive crashes the ship will not be able to keep flying. In that case, a new ship will appear after some seconds. There is an unlimited amount of new ships.

In a second stage, the child will go from the space to planet Mars. Everything is very similar to the first stage, except for the fact that instead of letter sounds, sounds of whole words are played back, emphasizing the sound of the first letter.

4. PHONOLOGICAL AWARENESS ADAPTIVE TUTOR DESIGN

In consequence with what was expressed in the previous work section, this is not about modeling the mind of a child or a teacher. We want to explore an approach to adaptive ITS that does not require analytic mind models, but simple assumptions about the domain that is being taught.

Phonological awareness training usually takes the form of games in which a tutor asks questions to a child. We decided to use the task of asking for the first letter of a spoken word. As it was mentioned before, it is beneficial to train this skill because it cannot be learnt without supervision. Also, it is easy to implement, and has the right difficulty level for children who cannot read yet. This cannot be said of rhyme games which have reading as a prerequisite.

Before a child can solve the task of identifying the letter corresponding to the initial sound in a word, he or she most likely needs to know how each letter sounds. Because of this, it was decided that questions teaching this would precede questions asking for the initial letter of a spoken word.

With respect to the previous section, the test can be described more schematically as a series of multiple selection questions in which the computer system plays back a sound and displays three letters on screen, one of which corresponds to the auditive stimulus. A child has to answer to each question by selecting which of the three letters corresponds to the sound that was played back. During a first stage of the test, auditive stimuli are letter sounds, and at each trial the child has to answer which of the three displayed letters corresponds to the sound that was played back. In a second stage, auditive stimuli are whole words, and the child has to select which of the three displayed letters corresponds to the initial sound of the word.

At each trial, the software must decide which question to ask. Both a target letter and two distractors must be chosen. The objective is to ask questions that are neither two easy nor too difficult. Easy questions are boring and do not lead to learning anything new, while too difficult questions are frustrating and do not encourage learning either. By knowing which letters the child already knows well and recognizes as initials, a question can be built that brings the right amount of challenge. For example, a question having a target that the child does not know well, along with distractors that he recognizes well should be both challenging and not too difficult.

Unfortunately, we cannot know for sure which letters a child has mastered. All we can have is an estimation on how well a child will perform when asked about a certain letter, based on past answers. By the time we have gathered enough data to make reliable estimations, we will have asked about each content so many times that the child would have learnt everything (if he or she does not get extremely bored before, which is the most likely scenario). So decisions must be taken on incomplete information. Here is where the Bayesian approach found in a Markovian model shows its potential. It would be inefficient to first assess a child's knowledge and then act. But it is possible to act before full knowledge is present, even before any knowledge about the situation has been captured. The strategy is to take, at all times, what seems to be the best decision based on knowledge that has been accumulated. Here, the best decision is, in terms of probability, the one expected to lead to the biggest amount of learning, once again based on available knowledge about the child. Certainly, taking decisions based on incomplete information can lead to suboptimal results. But this is better than trying to act on full information which is not available.

4.1. Markov decision processes

By the end of section 2 it was argued that POMDP is an appropriate model for an intelligent tutor dealing with uncertainty. This is an extension to Markov decision processes (MDP) (Puterman, 1994; Bertsekas, 1995), a discrete stochastic model that assumes fully observable states along with uncertainty in the outcome of actions. POMDP generalizes MDP by allowing partially observable states. It does so by replacing states by state probability distributions, called beliefs. The set of all possible beliefs forms the belief space.

Solving the action policy for a POMDP model is computationally expensive and often can only be done approximately. The main problem is that the belief space has as many dimensions as the size of the set of states. There is no problem when the number of states and the number of different observations that can be drawn about the current state are each no more than a dozen; problems having a about a hundred states has been solved by an approximate method (Kaelbling et al., 1998). Problems with hundreds or even thousands of states have been successfully solved using point-based approaches (Kurniawati, Hsu, & Lee, 2008), currently the most promising method for solving POMDPs with large state sets. A key idea of this methods is to work with an approximation of the belief space by sampling points of it.

However, in an adaptive intelligent tutor, it makes a lot of sense to model states as boolean vectors having one entry for each concept or item that is being taught (Cassandra, 1998). This leads to a count of 2^n states, where n is the number of concepts. The phonological awareness tutor needs 24 concepts (the 27 letters in the Spanish alphabet minus 3 unused letters), leading to a state set size of almost 17 million. Is is impractical to solve a POMDP for a problem where the belief space has millions of dimensions. Instead, we start with an MDP, replacing states by a succinct state-belief representation so to deal with state uncertainty. Then we argue that there is no need to solve an action policy for this model. A greedy assumption allows to take a decision whenever it is needed by simulating one step ahead of the current tutor state and picking the action that leads to the best outcome.

4.2. Modeling the knowledge domain

Questions are built by choosing a target and two distractors from the Spanish alphabet, which has 27 letters. Letters ‘h’, ‘ñ’, and ‘w’ are not considered because they lead to problems: ‘h’ has no sound in Spanish, ‘ñ’ gets confused with ‘n’ and ‘w’ is not originally used in Spanish, so it’s sound depends on the origin of the word being pronounced. Therefore, the following are the symbols from where questions are chosen.

$$\Sigma = \{\textbf{a, b, c, ..., z}\} - \{\textbf{h, ñ, w}\}$$

It is worth mentioning that a restriction was implemented disallowing questions having two letters of similar or same sound. For instance, letters ‘t’ and ‘d’ could not appear in the same question.

In order to specify a MDP, the following is needed

- a state space S,
- an action space A,
- transition probabilities $P_a(s, s')$ (probability of going from state s to state s' when taking action a),
- and a reward function $R_a(s, s')$ (estimation of the immediate reward of going from state s to state s' when taking action a)

Child knowledge is represented by stating whether he knows each of the letters in Σ. During the first stage of the test it is considered that a child “knows” a letter if he or she will successfully recognize its symbol after its sound is played back by the computer. Then, in the second stage, a letter is considered to be known if it will be recognized as the initial of a spoken word. Since $|\Sigma| = 24$, there are are 2^{24} possible *knowledge states*. A representation of a probability distribution for the current knowledge state will need to always available in order for this model to work, requiring the storage 2^{24} numbers in memory. Fortunately, if we assume that knowing a letter does not imply knowing any other, this independence allows a more succinct representation. Even if knowledge of different letters is associated in some way before a child uses the tutor software (e.g. vocals tend to be learnt first), in the time scale of a single session with the software, changes in the knowledge of one letter are not expected to modify knowledge about other letters. Therefore, an independence assumption seems a reasonable design decision. By keeping track of the probability associated with knowing each letter, a probability value can be computed for any possible knowledge state. Consider this representation.

$$S = \{(p_a, p_b, p_c, ..., p_z) | p_* in [0..1]\} \tag{4.1}$$

Please note that each state is a probability vector. Instead of representing a state as a vector of truth values indicating whether each letter is currently known by the user, the representation in equation (4.1) allows to compute the probability any vector of truth values. Therefore a state is equivalent to a probability distribution over all possible actual states. In this sense, this is an information state MDP. Such is a more accurate model when states are partially observable, compared to a normal MDP where the current state is known at all times..

Given a child's knowledge state $(k_a, k_b, k_c, ..., k_z)$, where k_α is a binary truth value (0 or 1) specifying whether letter α is known, the probability for that state can be computed as the joint probability for all the letters having the corresponding truth value:

$$\prod_{\alpha in \Sigma} (p_a)^{k_a} (1 - p_a)^{(1-k_a)} \tag{4.2}$$

An MDP needs a set of actions from where to chose. Here an action a question. It is represented as a triple where the first element is the target letter and the other elements are distractors.

$$A = \{(t, x, y) | t, x, y \in \Sigma\} \tag{4.3}$$

Also the probability of going from any state to any other when performing any given action needs to be specified. It is interesting to note that when giving a wrong answer, choosing any of the distractors produces the same effect since, as it will be seen later, the probability update function is the same for both distractors. Therefore, even though tree different answers are possible, there are only two possible results: right or wrong. Then, from any state, there are only two possible next states. Transition probabilities need only be specified for those two states.

$$P_a(s,s') = \left| \begin{array}{lll} P_a^s(\text{correct}) & \text{if} & s' = C_a(s) \\ P_a^s(\text{incorrect}) & \text{if} & s' = I_a(s) \\ 0 & & \text{otherwise} \end{array} \right. \tag{4.4}$$

Here $P_a^s(\text{correct})$ is the probability of receiving a correct answer after asking a in state s, and $P_a^s(\text{incorrect})$ is the same for incorrect answers. The first probability is the probability of going to state s' as long as s' is the state that results from updating probabilities for the case of a correct answer: $C_a(s)$. You probably guessed that $I_a(s)$ is the state that results from updating when the answer is incorrect. A procedure for computing this states will be given in the following section.

And finally a reward needs to be assigned at least to each state. The reward value must be representative of how good is it to be at the given state. One alternative is to take the lower probability from the state S, so that a state will be as good as the letter less likely to be known. Unfortunately, when several letters have the same probability (for example in the beginning of a session) many actions will lead to states having the same reward, which does not help in choosing the best one, specially considering the greedy approach that is being used here. Summing all probabilities together was tested. But better results where obtained when using the following reward definition.

$$R_a(s,s') = R(s') = \sum_{\alpha \in \Sigma} \sqrt{p_\alpha} \tag{4.5}$$

Here p_α is the belief associated to α in s'. Since $\partial R(s)/\partial p_\alpha = \frac{1}{2}\sqrt{p_\alpha}$ is bigger for lower values of p_α, improving one of the smallest probabilities is more rewarding than improving one of the biggest, so the tutor focuses on teaching the letters which have shown problems.

4.3. Updating states

Suppose that the software is being used by a child, and s is the current information state. An action (t, d_1, d_2) was performed (t is the target letter, and d_1, d_2 are distractors), which is the same as saying that an auditive stimulus was played back for letter t and the three letters t, d_1 and d_2 are being displayed on the screen so that the child can choose an answer.

Should the child choose letter t, chances are that he or she knew the correct answer. But it is also possible that a correct answer was given by chance or that a different answer was intended and the correct one was given by mistake. Also it is possible that the child knew the two distractors d_1 and d_2 well enough to increase the chance of a correct answer. Two conclusions can be extracted from this: (1) a correct answer should increase the belief on that the child knows letter t but not make it 1.0, because there is always some uncertainty, and (2) beliefs for d_1 and d_2 should also be updated because it is possible that the child knew one or both of that letters well.

In the case of an incorrect answer, one possibility is that the child did not know t, but it is also possible that he or she intended to answer t and chose d_1 or d_2 by mistake. In any case, capturing a wrong answer should make the software “believe” that not only the target, but also one or both of the distractors are not well known by the child, since their presence did not help to discard the incorrect alternatives. Once again, just like when the answer is correct, beliefs must changed for the three letters involved in the question, but not dramatically because of the uncertainty.

First the case of a correct answer will be studied. Based on the Bayes rule, we compute a new belief value for letter t given the newly captured answer (evidence). Let r be the captured answer, k_t be the proposition that the child knows letter t well, and a be the action (question) that was chosen.

$$P_t \leftarrow P(k_t|r = t, a) = \frac{P(r = t|k_t, a)P_t}{P(r = t|a)} \tag{4.6}$$

The probability that the child knows letter t is assigned a new value proportional to the probability of capturing a correct answer (evidence) given that the child knows t and the question was a, multiplied by the previous belief value for t. The same argument supports the following as an update rule for a distractor d. Let k_d be the proposition stating that the child knows letter d.

$$P_d \leftarrow P(k_d|r = t, a) = \frac{P(r = t|k_d, a)P_d}{P(r = t|a)} \tag{4.7}$$

In the case of an incorrect answer, the update rules become as follows.

$$P_t \leftarrow P(k_t|r \neq t, a) = \frac{P(r \neq t|k_t, a)P_t}{P(r \neq t|a)} \tag{4.8}$$

$$P_d \leftarrow P(k_d|r \neq t, a) = \frac{P(r \neq t|k_d, a)P_d}{P(r \neq t|a)} \tag{4.9}$$

In order to compute a value for equations (4.6) and (4.8), a value is needed for $P(r = t|k_t, a)$ and $P(r \neq t|k_t, a)$. Previous to the study that will be detailed later in this text, a trial was performed with a small group of pre-reader children having little to no experience with computers. Even though it is difficult to tell when a child answers wrong by mistake (e.g. because of an unintended movement of the mouse), observation suggested that this happened in 1 out of 4 or 1 out of 5 trials. Because of methodological difficulties an exact measure could not be extracted, so based just on the previous observation, it was decided that

$$P(r = t|k_t, a) = P(r = t|k_t) = 0.75 \tag{4.10}$$

Therefore,

$$P_a(r \neq t|k_t, a) = 0.25 \tag{4.11}$$

Despite the guesswork, this value worked very well in practice. A little bit more involved is the calculation of $P(r = t|k_d, a)$, necessary for computing (4.7). Since d is one of the distractors, let e be the other. Therefore, $a = (t, d, e)$. Please, consider the following expression

$$\begin{aligned} P(r = t|k_d, a) &= P(r = t|k_t \wedge k_d \wedge k_e)p_t p_e \\ &+ P(r = t|k_t \wedge k_d \wedge \neg k_e)p_t(1 - p_e) \\ &+ P(r = t|\neg k_t \wedge k_d \wedge k_e)(1 - p_t)p_e \\ &+ P(r = t|\neg k_t \wedge k_d \wedge \neg k_e)(1 - p_t)(1 - p_e) \end{aligned} \tag{4.12}$$

Since three letters are involved, the consideration that each one can be either known or not known by the child gives place to $2^3 = 8$ possible cases. But since we are assuming that the distractor d is known, only the four cases that include k_d are possible and correspond to the four rows of (4.12). For each of the four cases, the probability of obtaining a correct answer in that case is multiplied by the probability that this is the case, and the four terms are summed, having (4.12) as result.

Because of (4.10) $P(r = t) = 0.75$ when t is known. Not knowing t but knowing both distractors is equivalent, yielding the same probability. But when only d is known, the child has to guess between to alternatives, so in this case $P(r = t) = 0.5 \cdot 0.75 = 0.375$. In consequence (4.12) becomes

$$\begin{aligned} P(r = t|k_d, a) &= 0.75 p_t p_e \\ &+ 0.75 p_t(1 - p_e) \\ &+ 0.75(1 - p_t)p_e \\ &+ 0.375(1 - p_t)(1 - p_e) \end{aligned} \tag{4.13}$$

The same argument applied to computing the probability of an incorrect answer leads to

$$P(r \neq t|k_d, a) = 0.25p_tp_e + 0.25p_t(1-p_e) + 0.25(1-p_t)p_e + 0.625(1-p_t)(1-p_e) \tag{4.14}$$

Now it is possible to compute numerators for the four belief update equations (4.6), (4.7), (4.8), and (4.9). For computing denominators, with $a = (t, d, e)$ we have

$$\begin{aligned} P(r = t|a) &= P(r = t|k_t)p_t \\ &+ P(r = t|\neg k_t \wedge k_d \wedge k_e)(1-p_t)p_dp_e \\ &+ P(r = t|\neg k_t \wedge k_d \wedge \neg k_e)(1-p_t)p_d(1-p_e) \\ &+ P(r = t|\neg k_t \wedge \neg k_d \wedge k_e)(1-p_t)(1-p_d)p_e \\ &+ P(r = t|\neg k_t \wedge \neg k_d \wedge \neg k_e)(1-p_t)(1-p_d)(1-p_e) \end{aligned} \tag{4.15}$$

Which evaluates to

$$\begin{aligned} P(r = t|a) &= 0.75p_t \\ &+ 0.75(1-p_t)p_dp_e \\ &+ 0.375(1-p_t)p_d(1-p_e) \\ &+ 0.375(1-p_t)(1-p_d)p_e \\ &+ 0.225(1-p_t)(1-p_d)(1-p_e) \end{aligned} \tag{4.16}$$

Then

$$\begin{aligned} P(r \neq t|a) &= 0.25p_t \\ &+ 0.25(1-p_t)p_dp_e \\ &+ 0.625(1-p_t)p_d(1-p_e) \\ &+ 0.625(1-p_t)(1-p_d)p_e \\ &+ 0.775(1-p_t)(1-p_d)(1-p_e) \end{aligned} \tag{4.17}$$

In short, the update of belief values for target and distractor letters equations can be obtained with the expressions (4.6), and (4.7) in the case of a correct answer. Expressions (4.8), and (4.9) can be used for wrong answers. Each element of those four equations can be computed using the other expressions in this section.

Now, how do we compute $C_a(s)$ and $I_a(s)$ in equation (4.4)? Remember that $C_a(s)$ is the information state that results when updating state s after a correct answer to a. And $I_a(s)$ is the same for an incorrect answer. For computing $C_a(s)$, begin with s and update beliefs for the three letters in a using (4.6) and (4.7). For $I_a(s)$ do the same using (4.8), and (4.9).

4.4. Decision taking policy

For so many states, attempting a closed solution to the MDP is not sensible. Fortunately, an approximation can be found by considering an important fact. The expression for reward is a measure of how much a child seems to know. States that yield the bigger reward are actually those with more value. This can be understood by realizing that the more letters the child has mastered, not only the more has the goal been achieved but also there is more knowledge to be used in questions to support the acquisition of new knowledge.

Instead of solving the MDP, an approach has been taken of considering the rewards of immediate states as a greedy approximation to their value. As a result, we have the following policy π for deciding which action to take for any given information state.

$$\pi(s) = \mathtt{argmax}_a \sum_{s'} P_a(s, s')R(s') \tag{4.18}$$

4.5. Initial conditions and termination criterion

If a standard population of children who would use the tutor software were known, it could be possible to test them on the knowledge they have of each letter with regard to the two tasks that have been implemented. This would allow to initialize beliefs to average values for that population. However, because of the adaptive nature of the software it desirable to use it with children from different cultural contexts and of different development stages. This made it sensible to initialize belief values in a neutral way which is a commonly used criterion when designing and MDP.

The activity that was developed here has two parts. First a child is asked about the association between letter sounds and symbols. After this stage is finished, he or she is asked about the initial sound of words. Each stage has the following termination criterion. If the minimum current belief value is bigger than the maximum initial belief, the game ends. In other words, this happens when belief values for all letters have been increased above the biggest initial belief. Let s_0 be the initial state and s the current state. Formally the termination criterion is,

$$\min s > \max s_0 \tag{4.19}$$

As it was mentioned, s_0 is initialized neutrally.

$$s_0 = (p_a = 0.5, p_b = 0.5, p_c = 0.5, ..., p_z = 0.5) \tag{4.20}$$

But for the second stage, the initial state s_0 is initialized as the last information state of the previous stage. This was decided because the more a child knows the sound of a letter the better it is expected he or she can recognize it as the initial sound in a word. Termination criterion is the same as before.

5. EMERGENT PROPERTIES

An interesting aspect that appears when observing the behavior of the game is that when a wrong answer is given, the same question will be repeated several times. If then, a child starts to give correct answers to the question, it still will be repeated some more times. The more times a wrong answer was given, the more times the question will be asked. Eventually, a new question will appear, but soon the question that was answered wrong will appear once again.

In practice, the amount of repetitions was never excessive. This observed behavior was quite “intelligent” since while the computer kept repeating the same question, a child would eventually discover the correct answer by trial and error and then have to answer it some more times so he or she would remember better. And some trials later, he or she would be tested again to see whether the correct association was learnt.

This behavior was never deliberately programmed. It emerged from the execution of the decision taking model. This repetitions can be explained by the fact that a wrong answer to letter α strongly reduce the belief p_α that the letter is well known by the child. In order to get to a more rewarding state, that belief needs to be increased, which requires getting correct answers from the child. And the only way to get such answers is to ask first. The more times the question was answered wrong, the more the belief would reduce and then more correct answers would be needed to achieve an equally rewarding state.

The fact that the same question is repeated consecutively can be seen as a consequence of the square root used for computing the reward of an information state. This makes the letter with lower belief to be more “attractive” to ask, because increasing its low belief would be more rewarding then increasing a higher belief.

As a result of the termination criterion in equation (4.19), it emerges in practice that the duration of the game is very sensitive to the performance of each child, as it will be shown in a next section. If all questions are being answered correctly, the termination criterion is matched very soon. Nevertheless, a few wrong answers make belief values go low and it gets more difficult to end the game. In that case letters that got wrong answers need to get at least a few correct answers before the termination criterion can reached. Because of the reward function, those are actually the letters that the software will want to ask more about. Two adaptive behaviors emerge here. First, the software "wants" to spend more time with children that have more difficulties, and second, it insists on training letters that previously got wrong answers. The inclusion of equation (4.10) in the model helps with correctly managing the case when a wrong answer is given by mistake.

6. STUDY: BENEFITS OF THE ADAPTIVE TUTOR

In order to determine whether the adaptive design of the tutor yields any benefits with respect to a more traditionally designed software, a second, non-adaptive game was developed, looking exactly the same as the adaptive game but with a different decision taking policy. The non-adaptive game keeps track for each letter of how many times consecutively a correct answer has been given to questions having that letter as a target. At each trial, to decide which question to ask, the lower counter value is found, and a target is chosen from all letters that have that count. This way, a question is generated about a letter having less consecutive correct answers than other letters. Then two targets are chosen randomly. As it was mentioned before, a restriction exists in the adaptive game that does not allow a question to include two letters with the same or similar sound. The same restrictions apply for this second non adaptive game.

Since no belief values are present in the non-adaptive game, a different termination criterion was needed. Whenever the last two answers for a letter are correct, it is considered to be done. The game ends when all letters are done.

Details follow on a study that was done to compare the adaptive game to the non-adaptive one in terms of the learning they produced in children.

6.1. Sample and methodology

Children for the study were taken from the Chilean school *Colegio Calasanz*, property of a Catholic religious congregation. In 2007, *SIMCE*, a study that assesses quality of

	Adaptive game	Non-adaptive game	Total
Kindergarten group	56	56	112
First grade group	57	55	112
Total	113	111	224

TABLE 6.1. Experiment group sizes. The table shows how many kindergarten and first grade children participated in the experiment, and how many were randomly assigned to play the adaptive and the non-adaptive game in each level.

education in Chilean schools, classified this school as belonging to a high socio-economic level. This considers the education an income level of parents, and also implies the absence of students in social vulnerability. SIMCE 2007 locates this school above the national average in all measured student performance scales (language, mathematics, and natural science).

A sample of 224 children participated in this study (47% male, 53% female), 112 from Kindergarten (ages: 5-6 years old) and 112 primary students from first grade (ages: 6-7 years old). Teachers verbally report that, despite individual differences, the development of all participants is within an expected range, informally but explicitly ruling out the presence of illnesses that could compromise learning.

Children from both levels where randomly assigned to an adaptive and a non-adaptive group (relative to the game that they would play) as shown in Table 6.1. Each child played only one of the two games, either once or consecutively several times at will. Children were explicitly told to only play if they wanted to and as many times as they wished. Other activities had been prepared for children not willing to play any more. In the same room groups of about 40 children played the game at the same time. Headphones were used by the participants so that each one will only listen sounds from his or her computer.

For each trial the software recorded the question and the given answer. This allowed using each game as a phonological awareness test. Answers having a reaction time lower than 300 msec. were excluded from analysis, because it is not likely for a student to have perceived and processed a question so fast. Answers from trials where the mouse pointer remained static where also removed since during the experiment some children would occasionally stand up from the computer. This to two criteria caused 0.7345% of the answers to be excluded.

From what was observed during the experiment, it was not uncommon that a child would not want to pay attention to questions but would like to keep answering erratically in order to see what happened. Also, despite being tested before each session, headphones would sometimes get disconnected because of children activity. It was noted that some

Factor	Df	Sum Sq	Mean Sq	F value	Pr(>F)
level	1	13.545	13.545	13.0186	0.0003847
game	1	2.444	2.444	2.3487	0.1268796
level:game	1	0.319	0.319	0.3071	0.5800679
Residuals	212	220.575	1.040		

TABLE 6.2. ANOVA table for children performance while playing with the phonological awareness tutor software. First row (*level*) corresponds to the main effect of children level (kindergarten vs first grade). Second row (*game*) corresponds to the effect of the type of game (adaptive vs non-adaptive) on top of the effect of *level*. The third row corresponds to the interaction effect of the two previous factors. Only the *level* factor produces a statistically significant effect.

children liked to play and see the ship crashing even when they could not hear sounds through the headphones. In order to avoid the inclusion of erratic data in the analysis, Sessions having an overall proportion of correct answers less or equal than $1/3$ (expected by chance) were excluded. As a result 14.1975% of the sessions were deleted.

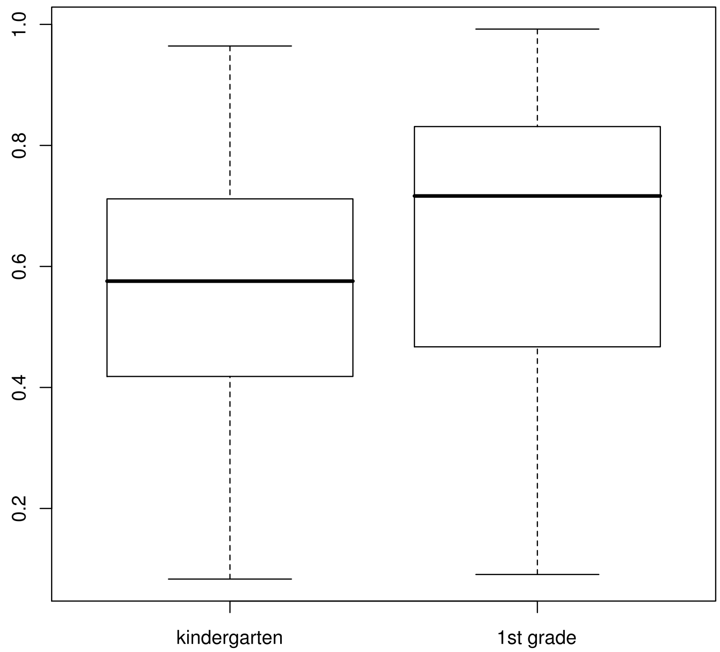
6.2. Results

Children performance in both games

A performance measure was computed for the first time each child played the game, as the proportion of correct answers from the total number of trials. As expected, children from first grade performed better both in the adaptive and non-adaptive game compared to kindergarten children (see Figure 6.1). After applying a logit transformation ($y = \log(x/(1-x))$) to each child's measure for normalization, a two-way ANOVA was performed as shown in Table 6.2, yielding a significant effect of children level (first grade vs. kindergarten, $F(1,167) = 13545, p = 0.0004$) but not of the type of game ($F(1,167) = 2.3487, p = 0.1269$). There was no significant interaction effect ($F(1,167) = 0.3071, p = 0.5801$).

Session lengths

The length of a session was measured as the number of questions that the child was asked. Only the first session was considered for each subject. It would not be surprising

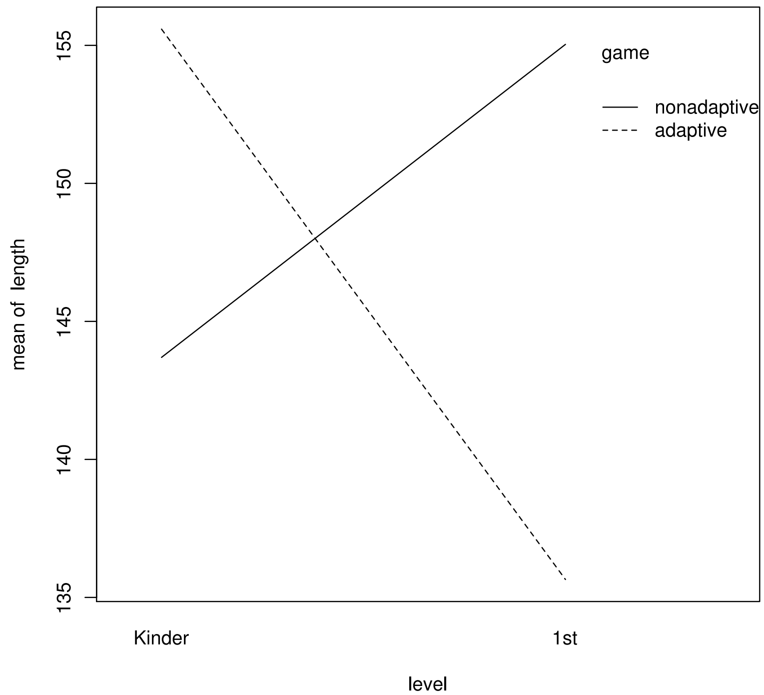


Level	Mean	Std. dev.
kindergarten	0.5622773	0.1779617
1st grade	0.6536510	0.2197492

FIGURE 6.1. Children performance while playing with the phonological awareness tutor software. Results are shown by level (kindergarten vs first grade) which was the only factor that showed a statistically significant effect for this measure ($F(1, 212) = 13.0186$, $p = 0.0004$).

Factor	Df	Sum Sq	Mean Sq	F value	Pr(>F)
level	1	1121	1121	0.1446	0.7041
game	1	785	785	0.1013	0.7505
level:game	1	13696	13696	1.7679	0.1850
Residuals	220	1704370	7747		

TABLE 6.3. ANOVA table for phonological awareness tutor session lengths. First row (*level*) corresponds to the main effect of children level (kindergarten vs first grade). Second row (*game*) corresponds to the effect of the type of game (adaptive vs non-adaptive) on top of the effect of *level*. The third row corresponds to the interaction effect of the two previous factors. No statistically significant effects were found.



Level	Game	Mean	Std. dev.
Kindergarten	adaptive	155.5893	83.01999
Kindergarten	non-adaptive	143.6964	88.57414
1st grade	adaptive	135.6491	95.21732
1st grade	non-adaptive	155.0364	84.55569

FIGURE 6.2. Interaction plot for phonological awareness tutor session lengths. Results are shown by game type (adaptive vs. non-adaptive) and level (kindergarten vs first grade). An ANOVA did not yield statistically significant effects either main or interactive.

if the two games had different average session lengths during the test because they have different termination criteria. It is, however, interesting to compare the adaptive game to the non-adaptive in how different the session lengths were for kindergarten and first grade children. Figure 6.2 shows that the adaptive tutor tends to ask more questions to kindergarten children than first grade. This is reasonable, since kindergarten children need more instruction in phonological awareness. The non-adaptive tutor shows an opposite tendency: it asks more questions to first grade children. Anyways, it must be noted that an ANOVA did not find significant main or interaction effects (see Table 6.3). So, even

Factor	Df	Sum Sq	Mean Sq	F value	Pr(>F)
level	1	0.03346	0.03346	2.0318	0.16051
game	1	0.03507	0.03507	2.1292	0.15103
level:game	1	0.07223	0.07223	4.3857	0.04155
Residuals	48	0.79058	0.01647		

TABLE 6.4. ANOVA table for accuracy change from first to second gaming session. First row (*level*) corresponds to the main effect of children level (kindergarten vs first grade). Second row (*game*) shows the effect of the type of game (adaptive vs non-adaptive) on top of the effect of *level*. The third row corresponds to the interaction effect of the two previous factor, and is the only effect for which statistical significance was found.

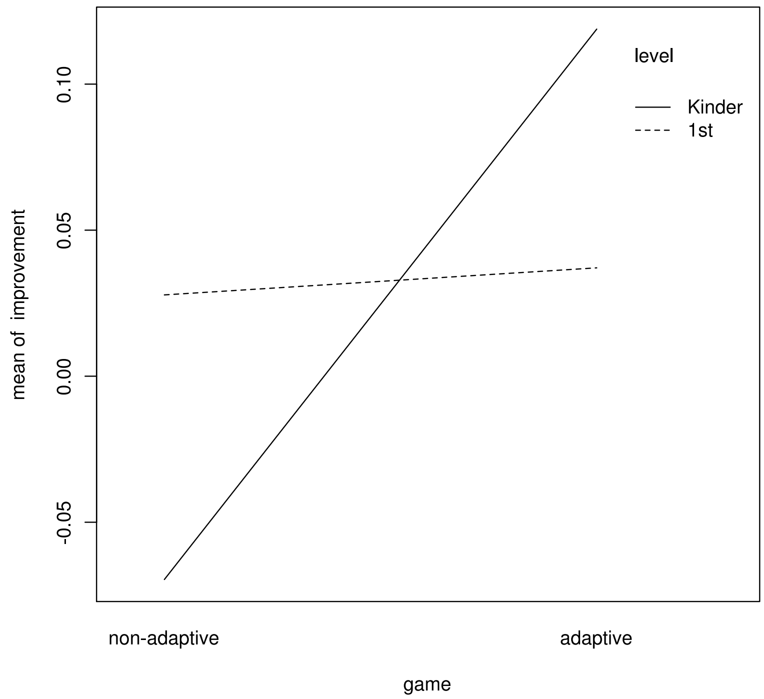
Participant level	t	df	p value
Kindergarten	2.4432	8.464	0.0194
First grade	0.2578	33.936	0.3991

TABLE 6.5. One sided t-tests for the adaptive game being better than non-adaptive (performance difference between second and first session being bigger for adaptive game). A statistically significant difference is found in kindergarten children but not in first grade.

though the adaptive game produces more reasonable session lengths than the non-adaptive, it cannot be discarded that this measure was obtained by chance.

Learning

Learning that occurred during the time span of a playing session was assessed by considering subjects who played their game twice or more. Only the two first playing sessions where analyzed. For each subject, only letters that where asked in both sessions were considered. An accuracy measure was computed for first (acc_1) and second (acc_2) sessions, as the proportion of correct answers from all trials. Then for each child an improvement measure was computed as the difference in accuracy from first to second session ($acc_2 - acc1$). A two way ANOVA was performed for this improvement measure, with children level and type of game as factors. Results are shown in Table 6.4. No main effects were found, but the interaction effect was significant ($F(1, 167) = 4.3857, p = 0.0416$). In order to further



Level	Game	Mean	Std. dev.
kindergarten	adaptive	0.11883260	0.1139982
kindergarten	non-adaptive	-0.06960801	0.1799970
first grade	adaptive	0.03709857	0.1101324
first grade	non-adaptive	0.02783356	0.1054427

FIGURE 6.3. Interaction plot for accuracy improvement from first to second gaming session. Improvement rate is similar for both games when used by first grade children. For kindergarten children, however, the adaptive game shows a performance increase while the non-adaptive game shows a decrease. Tiredness is a likely cause for this decrease with the non-adaptive game. Despite this, an increase in performance is still observed with the adaptive game.

investigate this effect, two t-tests comparing adaptive to non adaptive game, one for kindergarten participants and one for first grade were performed (Table 6.5). With kindergarten children, improvement from first to second session was significantly higher for the adaptive game compared to non-adaptive ($t = 2.4432$, $p = 0.0194$). In contrast, no statistically significant difference was found in first grade ($t = 0.2578$, $p = 0.3991$). In some way, this was expected because first grade children can read, and are therefore expected to already have phonological awareness skills. For them, there is not too much room for improvement in this task. This results are summarized in Figure 6.3. The figure shows one negative improvement value, which corresponds to kindergarten participants playing the non-adaptive game. This should not be interpreted as a proof of dislearning, since tiredness is a more plausible cause.

7. DISCUSSION

7.1. Main findings

An analysis of children accuracy in answering tutor questions, particularly how it improved form the first to the second gaming session, allowed to assess both games in terms of the learning that they produce. It was found that improvement was subtle for first grade children, being almost the same for both games. Together with the finding that those children performed better than the kindergarten group, this suggests that first grade children cannot benefit too much from training in the task that was designed. However, in children from kindergarten it was found that the adaptive game produced an improvement while a negative improvement value was measured with the non-adaptive game. It is unlikely that children would dislearn by practicing. The fact that the kindergarten group is younger and found the task more difficult (judging by their performance), suggests that a more reasonable explanation is tiredness. Since two evaluations are being performed, one immediately after the other, it is expected that each subject will face the second evaluation a little bit more tired. Other possible explanation is boredom. Nevertheless, in that case this effect should be expected in a higher degree with first grade children, but it was not found. In any case, results show that with the adaptive game the effect is either absent or has been overcome.

Perhaps the main finding was that for kindergarten children, the learning measure was significantly different between the adaptive and non-adaptive game. The fact that the adaptive game performed better strongly suggests that there is a benefit in exploring this approach to educational software. Both software pieces have the same looks, make the same kind of questions, and are equally unaware of what is going on inside the student's mind. All that is known to both games is the answers, and they are only used to assess performance. What makes the adaptive game different is that it attempts to use efficient use of this little available information in order to decide which questions are expected to produce an increase in knowledge over time. Here the Markovian approach is a key factor allowing the efficient use of available information in a model that can handle dynamicly changing states that are not fully observable.

7.2. Applicability to other domains

Although inspired on activities that help children with phonological awareness, the decision taking model is not specific to that domain. From the state update equations (see section 4.3) it can be seen that no domain specific probabilities appear in the model. Transition probabilities depend on the number of distractors and belief values which result from registering a subject's answers. This generality is a consequence of avoiding student modelling. Considering the mind of the student in the design would require a model that in some way encodes facts about how a student reasons in the specific domain.

Think for instance that instead of teaching sound/symbol associations in written language, you want to teach geographic associations. A country capital city could be displayed in a question showing three or more possible countries to choose. Cognitive processes related to this learning must be very different to those involved in learning to read sounds and words. Nevertheless, because of the naive approach taken here it is possible to apply exactly the same equations shown in this text to the design of a software that can help with learning country capitals.

In general, whenever a set of related knowledge items can be known more or less independently one of the other (e.g. letters or country capitals), and their number and nature makes it sensible to think that the right multiple selection questions will help with learning, then the approach taken in this work can be applied. First, when the subject has no knowledge about the topic, some trial and error applied to the first questions will teach some basic facts. Then, if known items appear as distractors, this will help with giving correct answers for questions about unknown targets. By repeatedly giving correct answers to the item it will become known. And this repetition of questions about items that are not well known yet, is an adaptive behavior that emerges naturally from the design that was proposed here.

Teaching multiplication tables is an interesting possibility because it has a slightly different structure. Choose an exercise taking the form $xy =?$. You would be making a question about the multiplication tables of x and y. Regardless of which alternatives are chosen to go with the question, two different items are being asked at the same time. By carefully rethinking the equations, a model can be built for this new scenario of teaching multiplication tables, updating for two targets at once.

A general property of the model that has been proposed here is that it is never necessary to directly address the question of how fast a subject learns: how many times does he or she need to answer questions about some item in order to learn it? Instead of that, here the software tries to detect learning when it has occurred, regardless of how long it took. It is evidence that drives belief updates, making no previous assumptions on when learning is expected to occur. By measuring how well things are going (through the correctness of answers) and updating belief values accordingly, the software adapts to what the subject knows, including what he or she has recently learnt. So despite making no assumptions on how the subject's mind will work during the task, the software is still sensitive to what the subject is learning. The advantage of making less assumptions lies in that this model can be applied to teaching about different learning domains and different kinds of subjects without having to adjust for difficulty, age, mental health, etc.

7.3. Future research

It was found that the proposed model, when applied to the design of an intelligent tutor, can produce learning, at least in the domain of phonological awareness. Even though there seem to be no theoretical obstacles to applying this model to other domains with little modification, it remains to be seen if result will be good in practice. A lot of simplifying assumptions are being made in this design. A relevant one to consider is that knowing all distractors and not the target is equivalent to knowing the target. In practice, deducing the target from distractors should be more difficult and prone to error than knowing the target directly. Also, such deduction requires a process of logical thinking, be it explicit or not, that will probably be more developed on some children than others. Generally speaking, this shows the limitations of a naive approach. Explicit consideration of how much more difficult it is to deduce the target than answering directly is an example of an issue that would require studying how the mind of a subject works in the context of a particular task. This limitation did not show in practice for this study. The adaptive capabilities of the software seemed to "absorb" quite well any problems of this nature. Further research is needed in order to see whether good results will also be observed in other applications.

Two very important assumptions guided the decisions made for the model and could be put to the test in future research. First, knowledge about different items is considered to change independently during a gaming session. In other words, the fact that a student learns letter x during the game is assumed not to imply that knowledge about letter y was produced (unless it was explicitly present as a target or distractor). This allows to update belief probabilities by applying a procedure that only involves items present in each question. It also allows a succinct representation of the probability distribution of states. One natural extension would be to build and assess a model that can deal with domains where this assumption is not reasonable. It also remains to be tested how bad will the independence assumption work in domains where it is clear that it does not hold true.

A second key assumption was that, within the proposed model, the present value of a state is a good estimator of its future value, thus allowing to compute an action policy on demand by simulating a single step ahead whenever a decision was needed. In other words, a single step planning horizon is being used. This rises a question on which methods could be used for efficiently computing action policies that consider a wider planning horizon, and whether or when this will be a valuable addition.

REFERENCES

Aist, G. (2000). *Helping children learn vocabulary during computer-assisted oral reading* (Tech. Rep.). Computer Assisted Oral Reading.

Ball, E. W., & Blachman, B. A. (1991). Does phoneme awareness training in kindergarten make a difference in early word recognition and developmental spelling? *Reading Research Quarterly*, *26*(1).

Bertsekas, D. P. (1995). *Dynamic programming and optimal control*. Athena Scientific.

Bradley, L., & Bryant, P. (1991). Phonological skills before and after learning to read. In *Phonological processes in literacy* (chap. 3).

Bradley, L., & Bryant, P. E. (1983). Categorizing sounds and learning to read: A causal connection. *Nature, 30*.

Bruner, J. S. (1990). *Acts of meaning*. Cambridge: Hardvard University Press.

Cassandra, A. (1998). *A survey of POMDP applications.* (Presented at the AAAI Fall Symposium)

Cassandra, A., Kaelbling, L. P., & Littman, M. L. (1994). Acting optimally in partially observable stochastic domains. In *Proceedings of the twelfth national conference on artificial intelligence aaai-94* (Vol. 2). Seattle, Washington, USA: AAAI Press - MIT Press.

Chomsky, N. (1975). *The logical structure of linguistic theory*. University of Chicago Press.

Corbett, A. T., Koedinger, K. R., & Anderson, J. R. (1997). Intelligent tutoring systems. In M. Helander, T. K. Landauer, & P. Prabhu (Eds.), *Handbook of human-computer interaction* (2nd ed., chap. 37). Elsevier Science B. V.

Freedman, R. (2000). What is an intelligent tutoring system? *Intelligence*, *11*(3).

Gombert, J. E. (1992). Activité de lecture et activités associées. In *Psychologie cognitive de la lecture* (chap. 3).

Hämäläinen, W., & Vinni, M. (2006). Comparison of machine learning methods for intelligent tutoring systems. In M. Ikeda, K. D. Ashley, & T.-W. Chan (Eds.), *Intelligent tutoring systems* (Vol. 4053, pp. 525–534). Springer.

Hamburger, H., & Tecuci, G. (1998). Architecture of a pedagogical agent for human-computer learning and tutoring. In *Proceedings of the its-98 workshop on pedagogical agents* (Vol. 2).

Ibañez, A. (2008). *Dinámica de la cognición*. J. C. Saez.

Jiménez, J. E., & Venegas, E. (2004). Defining phonological awareness and its relationship to reading skills in low-literacy adults. *Journal of Educational Psychology*, *96*(4).

Juel, C. (1988). Learning to read and write: A longitudinal study of 54 children from first through foruth grades. *Journal of Educational Psychology, 80*(4).

Kaelbling, L. P., Littman, L. M., & Cassandra, A. R. (1998). Planning and acting in partially observable stochastic domains. *Artificial Intelligence*(101), 1–2.

Koedinger, K. R., & Corbett, A. (2006). Cognitive tutors: Technology bringing learning science to the classroom. In R. K. Sawyer (Ed.), *The cambridge handbook of the learning sciences.* Cambridge.

Kotsiantis, S. B., Pierrakeas, C., & Pintelas, P. E. (2003). Preventing student dropout in distance learning using machine learning techniques. In V. Palade, R. J. Howlett, & L. C. Jain (Eds.), *Kes* (Vol. 2774, pp. 267–274). Springer.

Kronmüller, E., & Cornejo, C. (2008). La pregunta por la mente: las ciencias cognitivas después del cognitivismo. In E. Kronmüller & C. Cornejo (Eds.), *Ciencias de la mente: aproximaciones desde latinoamérica.* J. C. Saez.

Kurniawati, H., Hsu, D., & Lee, W. S. (2008). SARSOP: Efficient point-based POMDP planning by approximating optimally reachable belief spaces. In *Proc. robotics: Science and systems.*

Lester, J. C., Callaway, C. B., Gregoire, J. P., Stelling, G. D., Towns, S. G., & Zettlemoyer, L. S. (2001). Animated pedagogical agents in knowledge-based learning environments. In *Smart machines in education* (pp. 269–298). AAAI Press.

Liberman, I. Y. (1973). Segmentation of the spoken word and reading acquisition. *Bulletin of the Orton Society*, *23*.

Liberman, I. Y. (1983). A language-oriented view of reading and its disabilities. In H. M. Myklebust (Ed.), *Progress in learning disabilities.*

Liberman, I. Y., Shankweiler, D., Fischer, W., & Carter, B. (1974). Explicit syllable and phoneme segmentation in the young child.

Minaei-Bidgoli, B., Kashy, D. A., Kortemeyer, G., & Punch, W. F. (2003). Predicting student performance: an application of data mining methods with an educational web-based system. In *In proceedings of 33rd frontiers in education conference* (pp. 2–13). IEEE Press.

Mostow, J. (2008). Experience from a reading tutor that listens: Evaluation purposes, excuses, and methods. In C. K. Kinzer & L. Verhoeven (Eds.), *Interactive literacy education: Facilitating literacy environments through technology.* New York: Lawrence Erlbaum Associates, Taylor & Francis Group.

Mostow, J., Hauptmann, A., Chase, L. L., & Roth, S. (1993). Towards a reading coach that listens: Automatic detection of oral reading errors. In *In proceedings of the eleventh national conference on artificial intelligence (aaai-93), 392-397. washington dc: American association for artificial intelligence* (pp. 392–397). AAAI Press.

Murray, T. (1999). Authoring intelligent tutoring systems: An analysis of the state of the art. *International Journal of Artificial Intelligence in Education*, *10*.

Newell, A., & Simon, H. (1976). Computer science as an empirical enquiry. In J. Garfield (Ed.), *Foundations of cognitive science.* Parangon House.

Newell, A., & Simon, H. A. (1972). *Human problem solving*. Prentice Hall.

Nwana, H. S. (1990). Intelligent tutoring systems: an overview. *Artificial Intelligence Review*, *4*(4).

Passenger, T., Stuart, M., & Terrell, C. (2002). Phonological processing and early literacy. *Journal of Research in Reading*, *23*(1).

Puterman, M. L. (1994). *Markov decision processes: Discrete stochastic dynamic programming*. Wiley-Interscience.

Roth, F. P., Speece, D. L., & Cooper, D. H. (2002). The name assigned to the document by the author. this field may also contain sub-titles, series names, and report numbers. A longitudinal analysis of the connection between oral language and early reading. *Journal of Educational Research, 95*(5).

Simmons, F., Singleton, C., & Horne, J. (2008). Phonological awareness and visual-spatial sketchpad functioning predict early arithmetic attainment: Evidence from a longitudinal study. *European Journal of Cognitive Psychology, 20*(4).

Sleeman, D., & Brown, J. (1982). *Intelligent tutoring systems*. Academia Press.

Storch, S. A., & Whitehurst, G. J. (2002). Oral language and code-related precursors to reading: Evidence from a longitudinal structural model. *Developmental Psychology, 38*.

Tunmer, W. E., Herriman, M. L., & Nesdale, A. R. (1988). Metalinguistic abilities and begining reading. *Reading Research Quarterly, 23*.

Uhr, L. (1969). Teaching machine programs that generate problems as a function of interaction with students. In *Proceedings of the 1969 24th national conference.*

Wagner, R. K., Torgesen, J. K., & Rashotte, C. A. (1994). Development of reading-related phonological processing abilities: New evidence of bidirectional causality from a latent variable longitudinal study. *Developmental Psychology, 30*(1).

Zhang, X., Mostow, J., Duke, N. K., Trotochaud, C., Valeri, J., & Corbett, A. (2008). Mining free-form spoken responses to tutor prompts. In *Proceedings of the first international conference on educational data mining, montreal.*