Personalized Language Learning Through Human-Robot Word-Guessing Game Interaction

Jerno Beuker (s5193559), Rob Everts (s5204771)
Group 19

April 11, 2025

Contributions: The project idea and implementation were done together. For the report, Rob wrote the introduction, the basic model, and the evaluation study, and Jerno wrote the extended model and the conclusion and discussion.
Word-count: 2585

Our code can be found on our GitHub

Introduction (401 words)

Using Bayesian knowledge tracing (BKT), we implemented a personalized With Other Words (WOW) game for the Alpha Mini robot. The WOW game is a word-guessing game in which one player keeps a word in mind, and the other player has to guess it by asking yes or no questions. The aim is that second language learners (specifically those learning the English language) can improve their learning by playing the WOW game with the robot. It has been shown that personalization in educational human-robot interaction (HRI) improves the learning outcomes compared to unpersonalized educational HRI, even when the personalization is relatively simple Leyzberg et al., 2014. The way we personalized the HRI with the Alpha Mini robot while playing the WOW game, is by implementing Bayesian Knowledge Tracing (BKT). BKT is an algorithm used to track a student's mastery of a skill, in our case, proficiency in the English language. Like Schodde et al., we implemented a BKT model to represent the user's mastery of the English vocabulary in discrete levels. Schodde et al. found that people using their adaptive learning model showed significantly better progress compared to a control group 2017. The discrete levels we use to indicate vocabulary difficulty are the first five levels of the *Common European Framework of Reference for Languages* (CEFR), which are A1 through C1. These language levels represent a guideline used to describe achievements of learners of foreign languages across Europe and are widely used. When a player plays the WOW game, the robot will choose words from a CEFR level that matches the mastery of the English level of the player. By winning or failing a game, this mastery level is updated accordingly, and thus, the words get easier or harder. It is essential that the WOW game is be played with a robot, instead of through a digital medium, like a computer, because it has been shown that language processes are grounded in the brain's motor control systems Glenberg and Gallese, 2012. A language is learned and understood through interaction with the real world. This is possible when interacting with our robot, which makes gestures and motivates the user to be active as well, unlike a screen/audio-based interface. In this report, we will explain in detail how we implemented Bayesian knowledge tracing when playing the WOW game with the Alpha Mini robot and propose a study to evaluate the learning process through interaction with our robot.

Basic Model (305 words)

In assignments 1 and 2, we implemented the WOW game with beat gestures and two iconic gestures. The only difference in the basic model and the extension model (that is not the actual extension) is that we added a losing state by allowing the user to give up. The core WOW game and gestures from the first two assignments stayed intact.

Unless the user declines to play, the WOW game is started by asking the user if they want to start thinking of a word. This determines the role in the game. When this is determined, a Gemini chat is prompted with the role specific instructions on how to play the WOW gam. Then, we enter a loop where either the user or the robot starts to ask yes or no questions, which the other player answers. The loop ends once the user guesses the word or says ”stop”, in which case the robot shuts down.

To accompany the robot’s speech, we implemented nine beat gestures. The robot performs a random one of these gestures on a random next syllable. We calculated how quickly the robot speaks syllables and performs a beat gesture on a random next one so that the beat gestures always feel natural (this is better described in the previous assignments). The robot keeps performing beat gestures as long as there is time before its speech is running out. So the beat gestures themselves contain some randomness to make the gestures feel more natural.

Besides this, we also implemented two iconic gestures. The ”eureka” gesture, which shows a deep think followed by an ”aha!” movement, and the ”celebrate” gesture. The eureka gesture is played randomly (but sparsely) in the game loop when the robot has to answer/come up with a question. The celebrate gesture is played when the word is guessed.

Extended Model Design (1046 words)

Design Process

We kicked off the design phase by outlining the additional features we wanted to implement:

1. Varying difficulty levels in the WOW game.
2. A way to identify and save user profiles.
3. Integration of Bayesian Knowledge Tracing (BKT) to personalize vocabulary difficulty.

We decided to implement each of these features initially and bug-fixing them at a later point. This was possible as we had a clear overview and a good idea on what exactly needs to be done for this project. As we had this very overview, we were able to figure out potential issues before we faced them, which led us to not run into any ’failed attempts’. An example of this is figuring out a user’s name. Beforehand, we realized that users might respond with a sentence, so simple string splitting would not work. To solve this, we gave it to the LLM to have it figure out the name in the response.

Design Choices

One important design choice is that we decided to map CEFR levels to the BKT value. As mentioned before, the CEFR scale is commonly used to communicate language fluency. Therefore this is a logical and reliable scale to change the difficulty of the game. The different CEFR levels we mapped to values of the BKT. It is good to mention that these values do not necessarily correspond. We chose to map the values so that every CEFR level covers 20% of the percentage chance of one mastering the language. We chose this as we thought as long as the mapping is consistent, users will get stuck on their level of proficiency.

Implementation of the Extension

As mentioned before, the different levels of the WOW game are implemented by CEFR level. For in case it is the robot’s turn to think of a word, we found lists of words that correspond to a level of CEFR (`https://www.esl-lounge.com/student/reference/a1-cefr-vocabulary-word-list.php`). These lists we imported and filtered by nouns (this is because adjectives, for example, exist at a low CEFR level, like A1, but are a lot harder to guess, like lovely). From these lists we pick 5 random words and feed them to the LLM. The LLM chooses what word to think of based on the prompt we gave it;

```
"You are playing the game of taboo. Pick the simpelest noun in the following
list: {words} to keep in mind, dont say which one. I will
have to guess this word with yes or no questions. Do not explain the game.
Once I guessed the word, say: you guessed it, lets celebrate!
If I give up, say: okay we will stop. After you say what the word was."
```

We made it pick the easiest word because still not all nouns of a CEFR level are well suited for a game of WOW (like ’modification’ and ’monk’).
If the robot has to guess a word, we tell Gemini the CEFR level of the user and ask it to take that into account with formulating questions or answers.

The user profiles are saved in a dictionary in a JSON file. The keys are the names of the people, and the values are the user stats (and their name once again for ease of use in our code).

The Bayesian Knowledge Tracing is implemented by using the following formula (source: Wikipedia):

$$p(L_{t+1})_u^k = p(L_t \mid \text{obs})_u^k + (1 - p(L_t \mid \text{obs})_u^k) \cdot p(T)^k$$

Figure 1: Formula for the probability of skill mastery.

In this formula, p(L) is the probability of knowing a skill beforehand, and p(T): probability of demonstrating knowledge of a skill after an opportunity to apply it. This is for user 'u', given skillset 'k' and observation obs.

Pipeline

Below is a more detailed description of the pipeline of the extended model.

After putting our robot in a natural position and setting up the STT, we call the `game_setup()` function. This function asks whether the user wants to play, asks for their name, and lets them pick their role (as mentioned in the basic model). We ask the user for their name to make the personalization happen; a user's name is the key in a dictionary to their progress. As mentioned before, instead of just taking the raw response of the user, we pass it through a Gemini chat to extract just the name. This is necessary because people can answer with a full sentence (e.g., "My name is..."), which would confuse the system and searching/making a user with the entire sentence as the key.

After this setup, the main game loop starts (described better in the next section). Once the game ends, we use `save_player_progress()` to log results. This function checks whether the user 'won' or gave up. If the user gave up, it counts as a loss (as we assume less learning takes place if a user gives up). If the word was guessed (by either the user or robot, depending on the roles), it counts as a win.

These outcomes update the user's vocabulary mastery using our BKT implementation. BKT has two tunable parameters: `p_T_loss` (default `-0.05`) and `p_T_win` (default `0.1`). These control how much a user forgets or learns based on losing or winning. The result is a smooth curve of learning that emphasizes reward while minimizing penalty (see Figure 2).

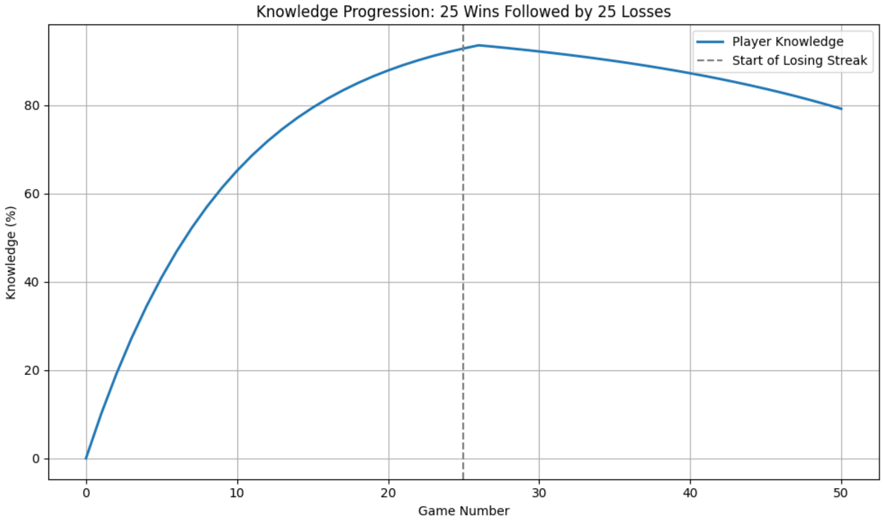


Figure 2: 50 games where a player always wins (starting with zero knowledge) for the first 25 games and always loses for the last 25 games

In this image, it can be seen that a user can level up quite quickly initially. Once the user has gained a lot of knowledge, they learn less and less per game. If a user starts to lose, this is reverse (so the user starts to 'forget' more and more per game), but this effect is halved (as the default value is half as low).

To visualize this pipeline in a simple flow chart:

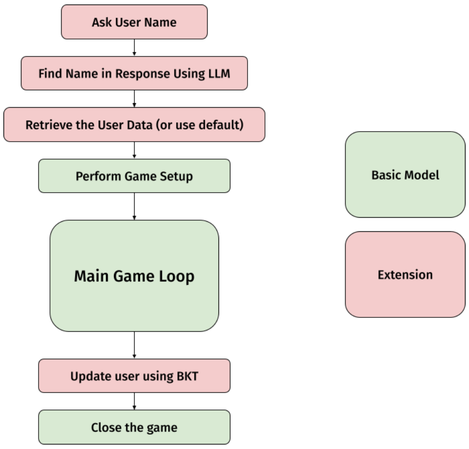


Figure 3: The game flow where the basic model is colored green and the extension is colored red

Libraries

Apart from standard libraries like `numpy`, `os`, and `random`, we made use of:

- `google.genai` – to access the Gemini API for natural language understanding.
- `dotenv` – to securely load our API key without exposing it in version control.

These libraries were crucial in making the interaction both flexible and secure.

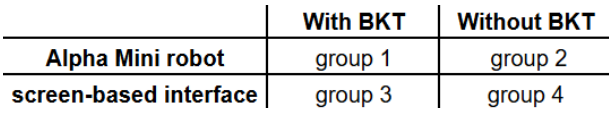
Evaluation Study (415 words)

If we were to evaluate our model, we would have to perform an evaluation study. We would determine the effects of HRI and BKT on second language learning through playing the WOW game with an Alpha Mini robot equipped with our model. To do this, we would answer the research question *Do human-robot interaction and Bayesian knowledge tracing improve the second language learning process when learning by playing the with-other-words game?*

By "improving the second language learning process", we mean two things: Does the user have a better experience when learning? And does the user measurably perform better compared to other methods? Answering these questions allows us to investigate if personalization in educational HRI improves learning outcomes (Leyzberg et al., 2014) and if BKT boosts learning progress, although it might not result in better results, post-learning (Schodde et al., 2017).

To answer our research question, we would gather enough participants to obtain significant results, who are learning English as their second language and are currently below a c1 CEFR level. This allows us to observe the learning progress and derive conclusions. The participants will be divided into four experimental setups (Figure 4). All groups will learn English vocabulary by playing the WOW game. The groups will be split by playing with the robot or a screen-based interface and by playing with or without BKT. This allows us to test the differences in second language learning through HRI and the differences through personalization by BKT. We would also see if the combined factors of HRI and BKT result in greater learning performance than simply the sum of their improvements combined. All groups will have multiple learning sessions of one hour, spread over a month (preferably one session every day). This allows for a lot of training by which the participants could improve significantly, and the differences in their improvements from the different setups become apparent.

To determine the two parts of the improved learning process, we would evaluate them in two different ways. The user experience will be measured through a survey. We will ask the participants to rate different parts of their learning experience, such as how motivated they were, how enjoyable they perceived it, how engaging they found it, etc.. To measure the actual improvement of their English vocabulary, we will let them take a test before and after the training process to measure their advancement. The results of these evaluations combined will allow us



	With BKT	Without BKT
Alpha Mini robot	group 1	group 2
screen-based interface	group 3	group 4

Figure 4: The different experimental setups in our proposed evaluation study, where group 1 learns with our model.

to determine the effectiveness of HRI and BKT on the second language learning process when playing the WOW game.

Conclusion and Discussion (305 words)

In this project, we extended the model for With Other Words (WOW) game for second language learning through interaction with the Alpha Mini robot by adding personalization. By integrating Bayesian Knowledge Tracing (BKT), the robot dynamically adjusted vocabulary difficulty based on a user's ongoing performance. This resulted in a system that not only made the learning process more engaging but also laid the foundation for studying the impact of personalization in educational human-robot interaction.

There are still a few limitations in our current system. A notable issue is that we do not currently distinguish between users with the same name. Since user profiles are stored using names as keys, having two individuals with identical names may result in their learning histories overlapping or being incorrectly updated. Another key challenge lies in the performance reliability of the technologies we rely on—specifically, the large language model (Gemini) and the speech-to-text system. While the LLM generally performs well, its consistency and precision in generating game-relevant responses can vary. Likewise, although the STT generally works quite well, recognition errors can happen. However, now that we have access to the confidence intervals from STT output, there is potential to improve its reliability by filtering or requesting clarifications on low-confidence responses.

Looking ahead, there are a couple of things next research could do. One thing is parameter tuning for the BKT model, optimizing parameters may yield improved personalization and learning outcomes. Moreover, adding hints to support players (particularly at lower proficiency levels) could enhance both motivation and vocabulary acquisition. This idea could be tested in a controlled experiment comparing the effects of hint-enhanced gameplay versus the current version. In summary, our prototype demonstrates the feasibility and potential of adaptive language learning via robot interaction, and future iterations could build on this foundation to deliver more accurate, supportive, and individualized educational experiences.

References

Glenberg, A. M., & Gallese, V. (2012). Action-based language: A theory of language acquisition, comprehension, and production. *cortex*, *48*(7), 905–922.

Leyzberg, D., Spaulding, S., & Scassellati, B. (2014). Personalizing robot tutors to individuals' learning differences. *Proceedings of the 2014 ACM/IEEE international conference on Human-robot interaction*, 423–430.

Schodde, T., Bergmann, K., & Kopp, S. (2017). Adaptive robot language tutoring based on bayesian knowledge tracing and predictive decision-making. *Proceedings of the 2017 ACM/IEEE international conference on human-robot interaction*, 128–136.