LearnerSkill – A Serious Hex Wars Game Centered on Learner Comprehension

Alexandru Marian Grigoroiu, Irina Toma, Mihai Dascalu, Stefan Trausan-Matu
University “Politehnica” of Bucharest
313 Splaiul Independenţei, Bucureşti
grigoroiualexandru@gmail.com. irina_toma@rocketmail.com, mihai.dascalu@cs.pub.ro
stefan.trausan@cs.pub.ro

ABSTRACT

The serious games sector has shown a significant increase in the last years while relating to the overall released video games. Available serious games that provide Natural Language Processing (NLP) facilities only cover very specific tasks and, in most cases, are outdated in terms of User Interface (UI) and user experience. Thus, despite their educational purpose, this makes them less attractive and less engaging. The game introduced in this paper is a novel serious game that is based on *ReaderBench*, a powerful NLP framework, and a ranking system derived from the True Skill algorithm. Our aim consists of creating an easily extensible and highly accessible game that can accommodate multiple comprehension-centered mini-games. The UI is represented as a simple map conquest game in which learners play different mini-games that asses their reading skills. A preliminary validation was conducted on a group of 5 users that were particularly impressed by the concept of the game and the ease of adding new mini-games in the learning cycle.

Author Keywords

Serious games; Natural Language Processing; TrueSkill ranking algorithm; *ReaderBench* framework.

ACM Classification Keywords

I.2.1 [Applications and Expert Systems]: Games.

I.2.7 [Natural Language Processing]: Discourse, Language parsing and understanding, Text analysis.

General Terms

Natural Language; Web-based Interaction; Games; Language acquisition; Text analysis.

INTRODUCTION

Serious games are a rather new approach in e-Learning [10], that keeps expanding with every new game and idea implemented. The domain of serious games is wide and expands from educational contexts, like games that exercise memory or previous knowledge, to the military field, where games are used for training [7]. The advantage of using these games is that they enable situations that are hard to create in real life.

From the educational field, this article concentrates especially on serious games that enhance students’ ability to learn from a given text. These games enhance students’ reading abilities and their capacity to understand texts. Also, games should be personalized according to each student’s intrinsic abilities and qualities, such as: interest in the field, prior knowledge, innate learning capabilities, but also on external factors such as learning materials and environment.

In the classroom, teachers experiment with different learning materials to help students better understand the given information and at the same time to keep them engaged. Despite their efforts, the additional materials can be hard to use and understand, or even outdated. Even though teachers can assess the textual complexity of the taught lessons and the reading comprehension of each student, this method is daunting and prompt to errors. A solution to this problem is integrating serious games in the learning environment. These learning games use automated tools, which evaluate users based on their reading capabilities, and provide tasks to users to complete and enhance their skills.

From an educational point of view, having a strong evaluation system is enough for creating a successful serious game. But from the user perspective, if the game is not appealing or nor motivating enough, it will not provide the expected results [4].

The game introduced in this paper, *LearnerSkill*, targets all the aforementioned items. It uses the *ReaderBench* [2] framework for evaluating students’ answers to different text-based tasks and it provides a fun and competitive game environment in which users improve their skills.

LEARNERSKILL

LearnerSkill is a serious game that helps both students and teachers. On one side, students develop their reading and comprehension skills of English language by playing engaging and competitive games. On the other side, it helps tutors evaluate learners’ progress over time and decide the best suited materials for their classes relying on the reading comprehension of each learner.

Expert readers have different strategies when it comes to reading a novel text. They are able to oversee their level of understanding at any given point, and when facing a difficulty, they rely on specific procedures, called reading strategies. Reading strategies are the mental processes used by students, consciously or unconsciously [8], when trying to understand a text. [1]. While integrating advanced Natural Language Processing (NLP) techniques, *ReaderBench* is able to identify reading strategies employed by learners in their verbalizations and asses their self-explanations or collaborative contributions within chat forums [3]. In this first implementation, *LearnerSkill* uses the *ReaderBench* module centered on identifying reading strategies employed by learners while self-explaining a given text.

Game Flow

LearnerSkill supports two types of users, tutors and learners. Tutors provide reading materials and create games based on them, can distribute learners in groups, track their evolution, and also participate in certain parts of the game offering aid and support. Whilst learners can compete in games by challenging one another or playing games proposed by tutors, track their own evolution in their profile and see what players have a similar rank. Tutors can upload or register text-based materials in the game that, on the server-side, are called documents. Based on each document one or more challenges can be created. A challenge is a task that a learner has to complete and it can be either a self-explanation, a summarization or selection of the most important sentences (see Figure 1). All the latter educational tasks become different mini-games within *LearnerSkill*. In terms of game design, our aim was to ensure extensibility with regards to the ease of adding new mini-games in the hexagons, such as highlighting important sentences or providing contextual definitions.

After creating challenges, tutors can group them into fixtures. A fixture can be made of one or more unique challenges. On the client side, a fixture corresponds to a game, which is composed of one or more rounds (corresponding to a server-side challenge). This grouping flexibility offers tutors endless options and the possibility to define tasks in coordination with the class curriculum. It can also let them test different scenarios for their chosen reading materials in order to find the best suited combination for their students' level of comprehension. This flow is explained in Figure 1.

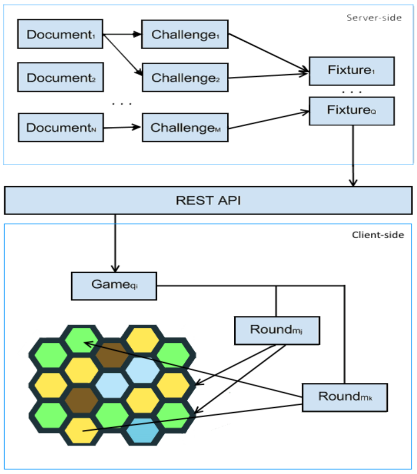

Figure 1. *LearnerSkill* game creation workflow.

This approach also offers the opportunity for tutors to track the answers submitted for a certain task. The capability to mix tasks that are based on different documents as part of the same fixture enables teachers to accurately evaluate how students act when faced with different combination of reading materials. In addition, the possibility of mixing tasks is not bound to materials covering a certain topic or subject. Depending on the materials teachers provide, any combination can be made, opening new perspectives on how students cope with reading on from different topics and of different difficulty levels, in various sequences.

Ranking Algorithm

LearnerSkill was meant to create a competitive environment in which players not only compete against each other in a random fashion, but play with other students that match their skill. However, players are deliberately capable of choosing to compete against a certain user, if the tutor enables this facility.

TrueSkill [6] is the model chosen to rank users based on their skills as it is stable, feature rich, and thoroughly tested. The model is implemented in the JSkills library (available at https://github.com/goochjs/JSkills), which was afterwards adapted to fit our particular requirements.

The first difference is that *LearnerSkill* supports only head to head competitions and tasks, without any team based game modes. The second change addresses the limitation of the original TrueSkill implementation which accepts only an ordered hierarchy of players, i.e. it does not account for one's score, just the ordering within the final standings. Given the fact that *ReaderBench* offers a numerical value that evaluates the overall performance of a learner's submission, a simple strategy was devised to adapt *ReaderBench* evaluations to the TrueSkill implementation: in head to head mode, the evaluation for each player is compared and the best score wins.

Unfortunately, this simplistic approach makes it very difficult for draw situations to develop, as two players are highly unlikely to present the exact same level of comprehension for a given text. Having only win or loss outcomes may not necessarily represent an inconvenient; however, in time this may decrease the accuracy of the estimation of each learner's skill. Therefore, in order to produce match outcomes as draws, we decided to set a margin relative to the average score for each specific task. If the scores of both players fall within this margin, the game ends in a draw; otherwise one wins, whilst the other loses. Depending on the overall performance of players, these margins may be modified accordingly in order to offer the best game experience.

As *LearnerSkill* consists of games which are split into rounds, the learner who wins most rounds, also wins the game. It is important to note that the round or the game outcome does not necessarily matter in the economy of a learner's skill. The purpose of the game is not to rate how well learners play it, but how well they understand texts, thus their comprehension skill. The outcome of the game is merely a motivational stimulus to continue playing the game while improve your personal skill.

The most important feature of the TrueSkill algorithm is represented by the ability to compute the draw margin for any given pair of players. This draw margin (or match quality) is computed using Formula 1:

$$\text{draw margin}(\beta^2, \mu_i, \mu_j, \sigma_i, \sigma_j) = \sqrt{\frac{2\beta^2}{2\beta^2+\sigma_i^2+\sigma_j^2}}\, e^{-\frac{(\mu_i+\mu_j)^2}{2(2\beta^2+\sigma_i^2+\sigma_j^2)}} \quad (1)$$

where μ_i and μ_j are the average value of skill, σ_i and σ_j are the standard deviations for two given players; β^2 is the variance of performance and is given by the formula $\beta^2 = (\sigma_0/2)^2$ [5]. For this algorithm, the original scale is kept given by a prior $\mu_0 = 25$ and ${\sigma_0}^2 = (25/3)^2$.

Thus, our aim of ensuring an equitable game emerges in which leaners do not become stressed when facing only more experienced players or, in contrary, do not become demotivated when encountering only lower ranked learners. Moreover, our approach encourages active engagement while learners strive to achieve higher rankings that reflect more than basic cumulative scores.

Graphical User Interface

The GUI is a simple web interface that makes use of the REST web services exposed by the *ReaderBench* framework. The current implementation targets the learner interface, while the tutor interface was limited to inserting the challenges directly in the database. After learners successfully sign in, they are presented with a hexagon grid, inspired from the Hex Wars games (available at https://www.hexwar.com), which represents the game map. The signed in learner is placed in the center tile of the grid. Other tiles of the map are occupied by other learners who participate in placed at a distance relative to their personal ranking. The closer the rank the other learners have, the closer they are placed to the center of the map.

In the current version of the UI, all learners start with a standard skill value at the game debut. In this situation, our system does not have any information on which learners could be best matched. An example of this situation can be seen in Figure 2. The signed-in user plays a game with another user (north-west position), both their skills are updated while the match quality for these two users changes. As the second user supposedly loses the game, his position changes and the corresponding modification is visible in Figure 2.b. As learners continue playing, their skills are updated accordingly, concurrently with their positions on the map.

From this map, the learner can choose an opponent to compete against. By hovering on the tile corresponding to another user, a tooltip element shows more information about that certain player, such as their score. By clicking on the other user's tile, the learner transitions to another map, the *Conquest Map*, in which players are shown the rounds of a particular game, as set by the teacher. Once again, each round occupies its own tile. Completed rounds can be distinguished from available ones by tile coloring.

When selecting a round from a map, the learner is presented with the target text and a text box in which he can input his self-explanation. After users decide that the self-explanation is complete, they can submit it for evaluation to the *ReaderBench* service. Upon the completion of the automated analysis, *ReaderBench* responds with a score for each particular self-explanation and the round is marked as done.

The final result for the given round is not decided until both players have submitted their answers. In addition, learners are not required to be online at the same time in order to play. Mini-games can feature a limited time for the user to complete them, but this is optional and mostly dependent on tutor decisions. Skill updates occur only after the entire game has been decided and the average score from each round is computed. Based on this average, the outcome of a game can be either win, draw or loss. The result is fed to the ranking algorithm which computes the new ratings and updates the learners' skills. These changes are afterward reflected in the *Ranking Map* from Figure 2.b, as described in the earlier paragraphs.

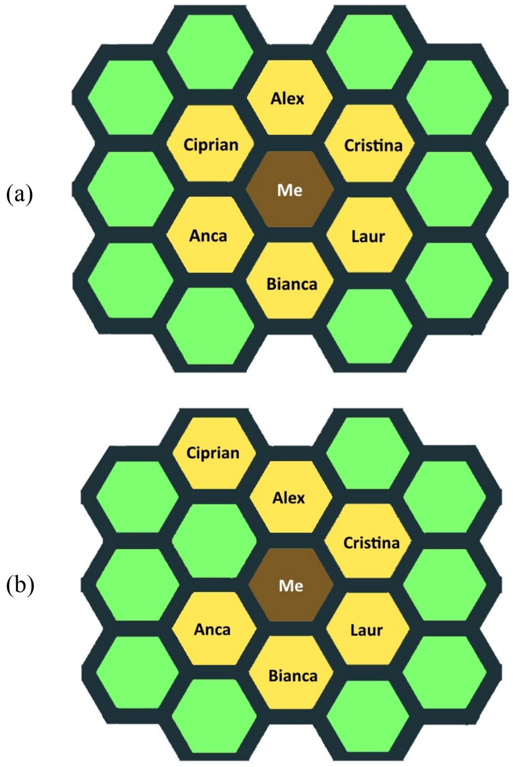

Figure 2. Ranking Map examples: a) at the start of the game; b) after skill update

RESULTS

A preliminary validation was performed on a group of 5 students that were asked to play the game and provide feedback. They had to answer a 13 questions survey with ratings on a 5-point Likert scale (1–completely disagree; 5–completely agree). The questions were separated into three subject groups: a) general perspective, b) map design and usability and c) game interaction and flow. 90% of the users enjoyed playing the game and 80% considered it was clear that, the closer players are on the *Ranking Map*, the more competitive the game will be. Only 12% of the users did not understand that their skill is based on their performance in the task and not on the game outcome.

During the development phase, user feedback was also collected. Users suggested having a limited time for completing and submitting an answer, once the learner decides to answer a given task. Furthermore, answers should be checked against plagiarism, or, at least, users should not be able to paste text in the response box.

CONCLUSIONS AND FURTHER DEVELOPMENT

LearnerSkill is an educational serious game that provides players with a competitive and entertaining environment for learning and assessing their comprehension skills. For teachers, it provides powerful tools to evaluate student performances in tight correlation with the provided learning materials. Moreover, teachers can further improve the quality of the learning materials, adapting them to better suit their students' level of comprehension.

The UI consists of a map conquest game in which players offer self-explanations. The submitted self-explanations are evaluated by *ReaderBench* and based on their score, the users may win, draw or lose, followed by a rating update.

In contrast to the previously developed serious games in our research group [9], *LearnerSkill* was specifically designed to accommodate extensibility in terms of rapid integration of new language mini-games, such as highlighting important phrases, extracting keywords, or providing contextual definitions similar to TOEFL testing. The potential mix of different tasks and learning activities creates the opportunity for greatly varying the educational scenarios. Moreover, the ranking algorithm can be applied on scores originating from any competitive game, thus ensuring a wide range of activities harmoniously integrated.

As further development, more mini-games will be included in the *LearnerSkill* engine, user feedback will be addressed and the overall user experience in the UI will be improved. In addition, we will include the possibility of a single player game mode in which students can complete tasks assigned by tutors, not necessarily in a competitive manner, but as homework.

ACKNOWLEDGMENT

This work was partially funded by the 644187 EC H2020 *Realising an Applied Gaming Eco-system* (RAGE).

REFERENCES

1. Dascalu, M., Dessus, P., Bianco, M., Trausan-Matu, S., and Nardy, A., 2014. Mining texts, learner productions and strategies with ReaderBench. In *Educational Data Mining: Applications and Trends*, A. Peña-Ayala Ed. Springer, Cham, Switzerland, 345–377.
2. Dascalu, M., Dessus, P., Trausan-Matu, S., Bianco, M., and Nardy, A., 2013. ReaderBench, an environment for analyzing text complexity and reading strategies. In *16th Int. Conf. on Artificial Intelligence in Education (AIED 2013)*, H.C. Lane, K. Yacef, J. Mostow and P. Pavlik Eds. Springer, Memphis, USA, 379–388.
3. Dascalu, M., Stavarache, L.L., Dessus, P., Trausan-Matu, S., McNamara, D.S., and Bianco, M., 2015. ReaderBench: An Integrated Cohesion-Centered Framework. In *10th European Conf. on Technology Enhanced Learning*, G. Conole, T. Klobucar, C. Rensing, J. Konert and É. Lavoué Eds. Springer, Toledo, Spain, 505–508.
4. Dempsey, K.B., Brunelle, J.F., Jackson, G.T., Boonthum, C., Levinstein, I.B., and McNAMARA, D.S., 2010. Miboard: Multiplayer interactive board game. *arXiv preprint arXiv:1009.2206*.
5. Herbrich, R., Minka, T., and Graepel, T., 2006. TrueSkill™: a Bayesian skill rating system. In *Proceedings of the 19th International Conference on Neural Information Processing Systems* MIT Press, 569-576.
6. Minka, T., Graepel, T.K., and Herbrich, R., 2013. Determining relative skills of players Google Patents.
7. Ritterfeld, U., Cody, M., and Vorderer, P., 2009. Serious games: Mechanisms and effects Routledge, New York, NY & London, UK.
8. Singhal, M., 2001. Reading proficiency, reading strategies, metacognitive awareness and L2 readers. *The Reading Matrix 1*, 1.
9. Toma, I., Andrioaie, I., Dascalu, M., and Trausan-Matu, S., 2016. Reading Space Secrets – A Serious Game Centered on Reading Strategies. *Revista Romana de Interactiune Om-Calculator 9*, 4, 269–284.
10. Westera, W., Nadolski, R.J., Hummel, H.G., and Wopereis, I.G., 2008. Serious games for higher education: a framework for reducing design complexity. *Journal of Computer Assisted Learning 24*, 5, 420–432.