



A Multimodal Platform to Teach Mathematics to Students with Vision-Impairment

Abhishek Jariwala^(✉), Daniela Marghitu^(✉), and Richard Chapman

Department of Computer Science and Software Engineering, Auburn University,
Auburn, USA

{avj0003,marghda,chapmro}@auburn.edu

Abstract. The advancements in Artificial Intelligence (AI) technologies in the last decade have led many developers, universities, and researchers to invest in the research and development of digital teaching platforms. This paper compares existing features implemented in the first prototype and provides visually impaired high-school students with a system to learn and practice mathematical problems. This project describes possibilities provided by open-source APIs such as Natural Language Toolkit (NLTK) and shows how to implement them to make education universal and accessible. We aim to achieve this by introducing an open-source platform, which will provide a set of modules for personalized speech recognition and accessible online tutoring. The system is divided into two modules: First, an intelligent conversational agent to help students learn new mathematical concepts, which opens up possibilities of providing students with a personalized response. Second, a mathematical expression parser and evaluator to help the student practice mathematical questions. The implementation and development of the system shall open doors towards self-directed learning for visually impaired students.

Keywords: Vision-impairment · Accessibility · High school mathematics · Natural Language Processing (NLP) with Python · Natural Language Toolkit (NLTK)

1 Introduction

Human-Computer Interaction (HCI) is a study of design, implementation, and evaluation of an interactive computing system for human use and studying the major phenomena surrounding them [14, 17]. The abundant information available on the Internet and recent improvements in Human-Computer Interaction (HCI) technologies have improved the way we acquire, share and test our knowledge.

High-school mathematics is a foundation for students to learn and persevere in social and professional worlds. Recent advancements in technology have provided students with many virtual tools to improve their knowledge of mathematics. Students with visual impairments are at a disadvantage due to a lack

of accessibility to the educational tools and the inability to process complex mathematical formulas and visual cues. This disadvantage leads to a significant knowledge gap between students with vision-impairment and students without disabilities. According to the U.S. Department of Education, as mentioned in [15], 15% of all visually impaired students are five or more grade levels behind their sighted peers. This knowledge gap leads to a staggering unemployment rate among disabled individuals.

The COVID-19 pandemic has affected everyone, but it has been challenging for visually impaired students [4]. While e-learning technologies have paved the way for self-learning during the COVID-19 pandemic, self-learning has not become a reality for vision-impairment people. According to the Royal National Institute of Blind People (RNIB) [1], two-thirds of visually impaired individuals have become less independent.

We reconsider the way high school students learn and practice mathematical questions. The prototype mentioned in [16] was developed with two goals: First, provide an easy-to-learn interaction mechanism to the students for clear communication with the application and integrate a highly interactive text-to-speech library that gives speech control to the student. The application uses a text-to-speech and a speech synthesizer JavaScript library that gives speech control to the user. This approach enables communication with the system but limits the responses to closed-ended questions. The developer must manually provide the questions and answers for the system to work as intended. The system intended to work for visually impaired students where clear communication is a key; this was a severe downside. To improve the human-computer interaction, we introduce an intelligent conversational bot. The paper proposes using NLTK to perform basic text operations on a given dataset to search for an answer to students' questions.

The prototype uses mXparser expression parser [18] to parse and evaluate mathematical equations. The major drawback of using the proposed library is that it lacks support for the Python programming language and provides step-by-step solutions to only limited mathematical concepts. This paper describes how an expression parser can be implemented and integrated with the system to help students practice mathematical questions.

The points mentioned above illustrate a need to improve the existing prototype of an interactive application for visually impaired students to learn high school mathematics concepts. The system is developed with the following goals:

- Improve the interaction by integrating NLTK to implement an intelligent bot for speech recognition.
- Integrate a JavaScript library to create an expression tree and implement an evaluator that solves the math problems and provides step-by-step solutions to students.

The rest of the paper is organized as follows: Sect. 2 introduces background and related research work to improve the online education experience for students with vision-impairment, Sect. 3 describes the proposed system along with the architecture. The paper is concluded in Sect. 4 with future research plans.

2 Background and Related Work

Over the years, with the technology advancements, there have been many improvements in how information is presented to students with vision-impairment, screen readers being the dominant mechanism. Many screen readers are also compatible with the websites developed under accessibility standards. However, that does not make the screen readers useful for visually impaired people because of several drawbacks. Visually impaired people often have to go through lots of irrelevant content before getting helpful information.

Although a digital copy of a text is useful for learning for visually impaired students, creating a digital copy of complex mathematical formulae is relatively difficult, as mentioned in [7,10]. When a complex mathematical formula, as shown in Eq. 1, is presented in digital format, many screen readers can interpret it in multiple ways. The digital copy of mathematical formulae often fails to convey the correct information. The authors in [21] introduced a non-ambiguous language MathSpeak, that can easily translate STEM materials into high-quality computer-synthesized voice.

$$3 + 2 + \frac{1}{x} \quad (1)$$

In paper [10], the authors evaluated the performance of the digital textbook vs. the traditional textbook while accessing algebra. The results demanded further research in the implementation of digital text in mathematics for visually impaired students. Regec [20] showed that the use of self-training tools was beneficial for teaching visually impaired students. The authors in separate studies [8,12] also showed the advantages of audio-tactile devices and speaking systems. Audio-tactile systems also increased motivation and curiosity for visually-impaired students.

Another useful approach, Process-Driven Math [13] was introduced to help blind students to be successful in college mathematics. Process-Driven Math is an auditory method that frees up students' working memory while solving equations by hiding complex numbers and symbols behind mathematical vocabulary layers. For example, the mathematical formula, " $x^2 + 2x + 1$ " will be presented to the student as "term + term + constant", freeing students' working memory and preparing the student to listen to the "elements" one by one. In this approach, the student is highly dependent on the trained reader-scribe for the information, which can be eliminated by developing an application that acts as a reader-scribe.

Davide and Volker [11] presented novel accessibility features of MathJax by generating tree representation of mathematical expressions. This extension mainly focuses on embedding complex mathematical expressions on the web and offering a similar user experience across all web browsers. This assistive technology extension also provides speech and tactile outputs of mathematical formulae.

In summary, the literature introducing new self-learning tools designed to teach visually impaired students illustrates the effectiveness of combining spoken systems with smart computing and the need for further research in this area.

3 System Architecture

The proposed system provides a platform for visually impaired students to learn and practice mathematical concepts. There are two objectives: First, creating an intelligent conversational agent that can teach new mathematical concepts and respond to students' questions. We use the NLTK library for data pre-processing and training the model. The decision to use NLTK was made because of its wide range of API features and precise documentation.

Second, providing step-by-step solutions to students to help them practice mathematical problems while providing hints whenever needed. This step is divided further into three steps: We use open-source JavaScript library math.js [2] to build an expression tree. It provides a flexible expression parser with support for symbolic computation and a large set of built-in functions and constants. Once we have an expression tree, we traverse the tree using inorder traversal and flatten the tree by simplifying math expressions to generate steps.

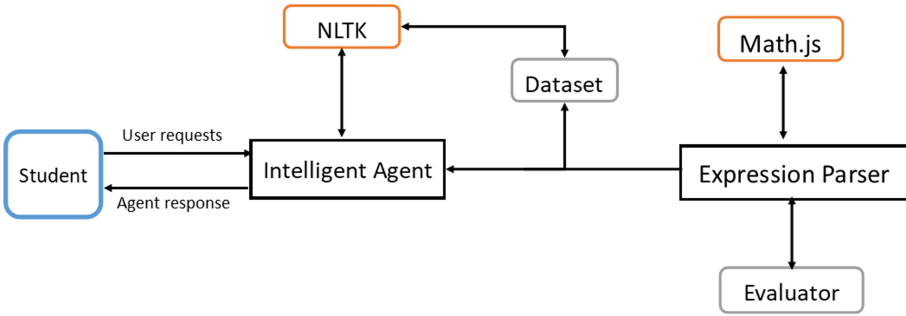


Fig. 1. System architecture.

Figure 1 shows the system architecture above. The system is divided into two steps:

3.1 An Intelligent Agent

The study of Natural language Processing (NLP) is studying the interaction between human language and computers. By utilizing NLP, researchers can organize and structure datasets to perform various automatic summarization, translation, and speech recognition tasks [19]. NLTK (Natural Language Toolkit) [9] is a leading platform for building Python programs to work with human language data.

For this research, we gathered over 1000 commonly asked questions that visually impaired students may ask while interacting with the system. The dataset includes questions and answers related to knowing more about the system, learning mathematical concepts, and asking for help. Figure 2 shows the structure of the data.

```

{"tag": "start_conversation",
 "patterns": ["Hi there", "Is anyone there?", "hey", "Hello", "Good day", "Hi"],
 "responses": ["Hello", "Happy to have you here", "Good to see you."],
 "context": [""]}

```

Fig. 2. Dataset structure.

The data is structured into tags, patterns, responses, and context.

- Tags: Categories that shows students’ intention
- Patterns: Possible questions that students may ask
- Responses: Possible responses to the questions mentioned in patterns
- Context: Contextual words relating to a tag for better classification

Trying to build an intelligent agent is a complex task as the only knowledge the agent has access to is the information it has learned itself [6]. The dataset is stored in a JSON file. Since machine learning algorithms require data to be in a numerical feature vector rather than text, the dataset must be filtered carefully using statistical and numerical means. NLTK provides various text processing methods to clean the data:

- Conversion into lowercase or uppercase: In this step, the words are uniformly converted to lowercase or uppercase, so the algorithm does not treat the same words differently.
- Removing noise: This includes removing punctuation and special characters. This step also includes removing stopwords. Stopwords are the most common words in the text and removing those does not change the sentence’s meaning. For example, “about”, “me”, “something”, etc.
- Tokenization: In tokenization, sentences are broken into a list of words, i.e., tokens.
- Stemming or Lemmatizing: Stemming or Lemmatizing is the process of reducing tokens to their root. The slight difference between stemming and lemmatizing is that the stemming can often create non-existent words.

To train a model, we build a neural network with three layers. The first layer contains 128 neurons, the second layer 64 neurons, and the third output layer contains the number of neurons equal to the number of intents to predict the response. We trained the model with 200 epochs and a batch size of five. Figures 3 and 4 show the graphs of training accuracy vs. validation accuracy and loss, respectively.

After the initial processing phase, we need to transform the text into meaningful feature vectors. This process is called feature extraction. Bag-of-words is a simple and popular feature extraction method. In this method, the structure of words in a document is discarded and the model is only concerned with words in the document.

Once the model is trained and saved, the intelligent agent responds to students’ questions with 99% accuracy. The example conversation with an intelligent agent is shown in Table 1.

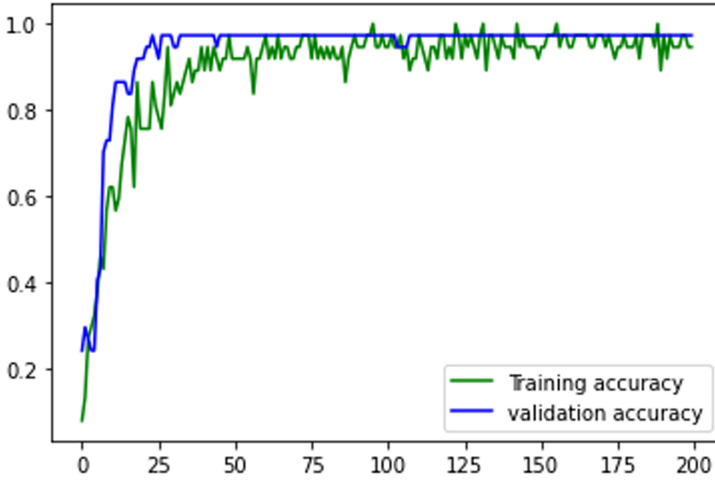


Fig. 3. Training accuracy vs. Validation accuracy.

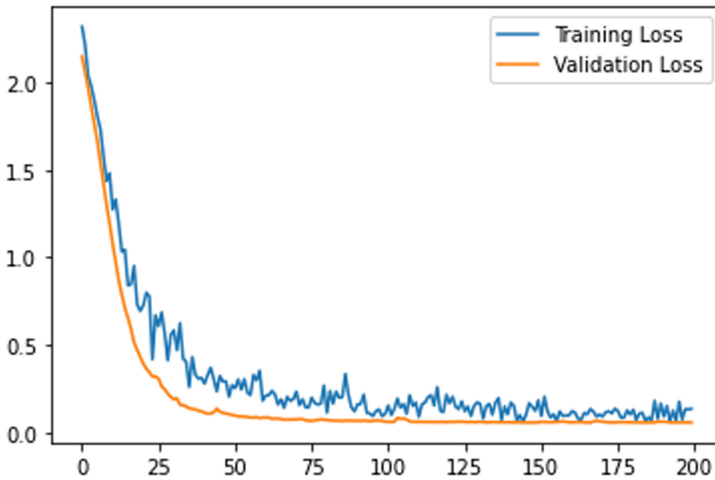


Fig. 4. Training loss vs. Validation loss

3.2 Math Expression Parser and Evaluator

We implemented an expression parser and evaluator that gives students options to practice mathematical concepts and ask for help along the way. The system shows the steps to solve a mathematical question as a tutor would. This idea is based on an open-source project, Socratic by Google [3,5].

There are three main parts in building expression parser and evaluator:

$$3 + 2 * x \quad (2)$$

Table 1. Example conversation with an intelligent agent

| Entity | Communication |
|---------|---|
| Student | Hi there |
| Agent | Hello there. Happy to have you |
| Student | Who are you? |
| Agent | I am Accessible Math. I can help you learn mathematical concepts |
| Student | What do you know about mathematics? |
| Agent | Ask me a question about linear equation and I will answer to the best of my knowledge |

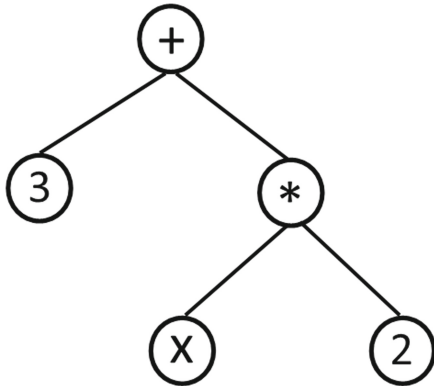


Fig. 5. Expression tree for Eq. 1

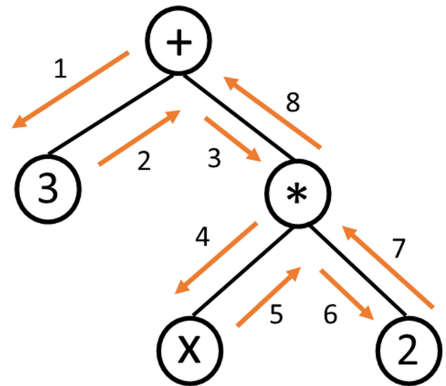


Fig. 6. Inorder tree traversal for Eq. 1

- Creating an expression Tree: Math.js is a powerful and open-source library that parses the mathematical equation string, generates an expression tree, and returns the tree’s root node. An expression tree can be used to analyze, manipulate, and evaluate expressions. The example of a generated expression tree for the Eq. (1) is shown in Fig. 5.
- Flattening the tree: Once the expression tree is parsed, we use inorder traversal to traverse and simplify the expression to generate steps. The tree traversal is shown in Fig. 6.
- Simplifying expressions: Once we traverse through the tree, we find the best nodes to apply simplification rules and record it as a “step”. This step provides necessary information such as operands, the operator, and the applied simplification rule. We store this information and wait for the student to practice the step. We only provide the information in speech format if students ask for hints.

4 Future Work and Conclusion

An accessible self-directed learning tool for visually impaired students is essential in the modern era with the increasing popularity of e-learning systems. It is necessary to integrate assistive technologies with e-learning platforms and designing educational content following accessibility guidelines. With our prototype's development and by introducing further improvements, our emphasis is to promote self-directed learning tools, especially for mathematics for visually impaired high school students.

The improvements presented in this paper are currently in beta testing but will be available as an open-source web application. The following improvements could be added in the future release of the work:

- We aim to assess the interaction with an intelligent agent for visually impaired students.
- Upon release, the application will be open-source and free to use to learn and understand the mathematical concepts.
- The current version only supports English as a primary language. In the future release, the application will be improved to support multiple languages.

References

1. The effect of lockdown and social distancing on blind and partially sighted people. <https://www.rnib.org.uk/campaigning/priority-campaigns/inclusive-journeys/eff-ect-lockdown-and-social-distancing-blind-and-partially-sighted-people>. Accessed 05 Feb 2021
2. An extensive math library for JavaScript and Node.js. <https://mathjs.org/>. Accessed 05 Feb 2021
3. Get unstuck: learn better. <https://socratic.org/>. Accessed 05 Feb 2021
4. The pandemic is undermining visually impaired people's independence. <https://theconversation.com/the-pandemic-is-undermining-visually-impaired-peoples-independence-heres-how-to-fix-this-142209>. Accessed 05 Feb 2021
5. Stepping into math: open-sourcing our step-by-step solver. <https://blog.socratic.org/stepping-into-math-open-sourcing-our-step-by-step-solver-9b5da066ae36>. Accessed 05 Feb 2021
6. Abdul-Kader, S.A., Woods, J.: Question answer system for online feedable new born Chatbot. In: 2017 Intelligent Systems Conference (IntelliSys), pp. 863–869 (2017). <https://doi.org/10.1109/IntelliSys.2017.8324231>
7. Alajarmeh, N., Pontelli, E., Son, T.: From “reading” math to “doing” math: a new direction in non-visual math accessibility. In: Stephanidis, C. (ed.) UAHCI 2011. LNCS, vol. 6768, pp. 501–510. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-21657-2_54
8. Arlinwibowo, J., Retnawati, H.: Developing audio tactile for visually impaired students. *Int. J. New Trends Educ. Implic.* **6**(4), 78–90 (2015)
9. Bird, S., Klein, E., Loper, E.: *Natural Language Processing with Python*. O'Reilly Media, Sebastopol (2009)
10. Bouck, E.C., Weng, P.L.: Reading math: a comparison of reading and listening to algebraic problems. *J. Spec. Educ. Technol.* **29**(4), 1–13 (2014). <https://doi.org/10.1177/016264341402900401>

11. Cervone, D., Sorge, V.: Adaptable accessibility features for mathematics on the web. In: Proceedings of the 16th Web for All 2019 Personalization - Personalizing the Web, W4A 2019. Association for Computing Machinery, New York (2019). <https://doi.org/10.1145/3315002.3317567>
12. DePountis, V.M., Pogrund, R.L., Griffin-Shirley, N., Lan, W.Y.: Technologies used in the study of advanced mathematics by students who are visually impaired in classrooms: teachers' perspectives. *J. Vis. Impair. Blind.* **109**(4), 265–278 (2015). <https://doi.org/10.1177/0145482X1510900403>
13. Gulley, A.P., Smith, L.A.C., Price, J.A., Prickett, L.C., Ragland, M.F.: Process-driven math: an auditory method of mathematics instruction and assessment for students who are blind or have low vision (2017)
14. Hewett, T.T., et al.: ACM SIGCHI curricula for human-computer interaction. Technical report, New York, NY, USA (1992)
15. Blackorby, J., Chorost, M., Garza, N., Guzman, A.: The academic performance of secondary school students with disabilities (2003)
16. Jariwala, A., Marghitu, D., Chapman, R.: MyA+ Math: teaching math to students with vision impairment. In: Antona, M., Stephanidis, C. (eds.) HCII 2020. LNCS, vol. 12189, pp. 200–211. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-49108-6_15
17. Lawrence, D.O., Ashleigh, M.: Impact of Human-Computer Interaction (HCI) on users in higher educational system: Southampton University as a case study. *Int. J. Manag. Technol.* **6**(3), 1–12 (2019)
18. mxparser - math expressions parser for java android `c#.net/mono/xamarin - mathematical formula parser/evaluator library`. <http://mathparser.org/>. Accessed 05 Feb 2021
19. Create your chatbot using Python NLTK. <https://medium.com/@ritidass29/create-your-chatbot-using-python-nltk-88809fa621d1>. Accessed 05 Feb 2021
20. Regec, V.: Mathematics in inclusive education of blind students in secondary schools in the Czech Republic. *Procedia Soc. Behav. Sci.* **174**, 3933–3939 (2015). <https://doi.org/10.1016/j.sbspro.2015.01.1136>. <https://www.sciencedirect.com/science/article/pii/S1877042815011957>. International Conference on New Horizons in Education, INTE 2014, Paris, France, 25–27 June 2014
21. Sheikh, W., Schleppenbach, D., Leas, D.: MathSpeak: a non-ambiguous language for audio rendering of MathML. *Int. J. Learn. Technol.* **13**(1), 3–25 (2018). <https://doi.org/10.1504/IJLT.2018.091609>