

WolfTutor

A system to enable peer tutoring built on Slack

Monica Metro
NC State University
mgmetro@ncsu.edu

Zachery DeLong
NC State University
zpdelong@ncsu.edu

Zhangqi Zha
NC State University
zzha@ncsu.edu

ABSTRACT

In this abstract, we need to preview our experiment and our results

1. INTRODUCTION

1.1 WolfTutor

WolfTutor is a system that seeks to enable students to tutor other students in a course-setting. It is a slack-based chat app that attempts to connect potential tutors in given subjects with students who need help with those subjects. Put another way, WolfTutor is an application that focuses entirely on enabling peer tutoring. At first blush, it seems like the app is an app to actually facilitate peer tutoring, which is not entirely accurate. WolfTutor has functions to register tutors and students and to schedule tutoring sessions. It does not have functionality to actually perform the tutoring itself, but is a logistic tool to enable the coordination required to schedule tutoring sessions.

WolfTutor is also gamified. It rewards tutors who are highly rated with a points system which can be implemented in a number of different contexts to help incentivize students to tutor other students.

There are a number of things that WolfTutor does extremely well, chief among them its novel interface. Being a chatbot makes the application easy to port to new languages and new devices, since its entire UI consists of simple english sentences and some rudimentary web inputs like text boxes and drop-downs. That being said, there are a few potential areas for improvement, chief among them scheduling and tutor-student matching.

1.2 Tutor Matching

WolfTutor's existing mechanisms for matching students to tutors are fairly rudimentary. When tutors register for the application, they are asked to give a set of subjects they are comfortable helping on (the list of which is determined and maintained by system administrators) and a list of days and

times they are available to instruct.

While immediately useful, this is not nearly as far as the system could go in terms of matching students with tutors. The team has built out a mechanism for enabling the matching of tutors on multiple criteria and has created an easy pathway to add or change the criteria easily.

One major concern for the team during development was the choice of a recommendation mechanism. While the team knew immediately that adding some kind of recommendation algorithm was going to be necessary, the actual mechanism is something that was debated significantly. In the end, the team opted for Occam's Razor: the simplest algorithm possible to create the most value possible, which in the future could easily be replaced or enhanced to compare performance.

The tool the team chose was a simple weighted average. The process is fairly simple: several criteria were chosen to each generate a respective "score". That score is then assigned a weight, and each tutor's score is averaged together with these weights, which can then be normalized and used to rank tutors. In this way, the problem of recommendation becomes a sorting problem, which can easily be solved using a number of very fast algorithms.

This approach also has the advantage of leaving each tutor with an individual score which can be used as input to a number of other possible algorithms, which will be discussed in section 4.

1.3 Literature Review

While the merits of peer tutoring are not the major focus of this paper, it is prudent to give a brief overview of the topic. Peer tutoring is a type of tutoring where a student seeks out instruction from another student who has already studied the subject that the student is interested in learning more on. There are a number of terms for the tutor in this situation such as "mentor" and "proctor" but for the purpose of this paper we will simply use tutor to refer to the student-tutor and student to refer to the student seeking help. [1]

There are a number of different well-documented benefits to peer tutoring, and they are not limited simply to the students being tutored. While the intention is to improve the students first, the process of teaching another student has many well-studied benefits for the tutor as well. [1]

While it is absolutely true that tutors benefit significantly from teaching their students, the obvious goal is to transfer knowledge from the tutor to the student. It is well documented that students tend to feel more engaged during one-on-one peer tutoring and that the peer tutoring can give more feedback than lecture-style learning which can in turn

help reduce student anxiety and improve learning outcomes.
[2]

1.4 System

WolfTutor can be broken down into two types of users: tutors and students. After enrolling in the system, the user's name, email, and phone number will be taken from slack and stored if available.

1.4.1 Student Use Cases

Students can:

1. Find a tutor by from a complete list of all available tutors after selecting a subject from an existing list
2. See the reviews for at tutor
3. Book a tutor by choosing one of WolfTutor's defined 30 minute slots created from the availability given by the tutor
4. Review a tutor from their last session (students have until the end of the day to review their tutor)
5. View reservations and reward point balance (points are used to schedule sessions; 100 points are given at signup)

1.4.2 Tutor Use Cases

Tutors can:

1. Become a tutor. WolfTutor only allows one major and one degree per tutor, but a tutor can represent multiple subjects. Tutor self-determines points pay rate and availability. Both apply to all subjects.
2. View subjects and availability given to WolfTutor
3. View reward point balance that can be used commercially

2. DESIGN

2.1 Enhancement

In this section we will design our goals for the system and outline what we wanted to accomplish.

2.2 Bugs

In this section we will talk about the issues we had to fix in the system before working and how we helped avoid them in our new code

2.3 Use Cases

testing

2.4 Architecture

testing

2.5 Infrastructure

testing

2.5.1 Slack

2.5.2 NodeJS

2.5.3 Heroku

3. EVALUATION

4. CONCLUSION

4.1 Results

4.2 Future Work

5. REFERENCES

- [1] M. M. Kim. Peer tutoring at colleges and universities. *College and University*, 90(4):2–7, Summer 2015. Copyright - Copyright American Association of Collegiate Registrars and Admissions Officers Summer 2015; Document feature - ; Last updated - 2017-11-22; CODEN - COLUBY; SubjectsTermNotLitGenreText - United States-US.
- [2] K. J. Topping. The effectiveness of peer tutoring in further and higher education: A typology and review of the literature. *Higher Education*, 32(3):321–345, 1996.