Buying Time: Enabling Learners to become Earners with a Real-World Paid Task Recommender System

Guanliang Chen*
Delft University of Technology
Delft, the Netherlands
guanliang.chen@tudelft.nl

Dan Davis†
Delft University of Technology
Delft, the Netherlands
d.j.davis@tudelft.nl

Markus Krause
UC Berkeley ICSI
Berkeley, California USA
markus@icsi.berkeley.edu

Claudia Hauff
Delft University of Technology
Delft, the Netherlands
c.hauff@tudelft.nl

Geert-Jan Houben
Delft University of Technology
Delft, the Netherlands
g.j.p.m.houben@tudelft.nl

ABSTRACT

Massive Open Online Courses (MOOCs) aim to educate the world, especially learners from developing countries. While MOOCs are certainly *available* to the masses, they are not yet fully *accessible*. Although all course content is just clicks away, deeply engaging with a MOOC requires a substantial time commitment, which frequently becomes a barrier to success. To mitigate the time required to learn from a MOOC, we here introduce a design that enables learners to earn money by applying what they learn in the course to real-world marketplace tasks. We present a *Paid Task **Rec**ommender **Sys**tem* (`Rec-$ys`), which automatically recommends course-relevant tasks to learners as drawn from online freelance platforms. `Rec-$ys` has been deployed into a data analysis MOOC and is currently under evaluation.

CCS Concepts

•Applied computing → Learning management systems;

Keywords

Learning Analytics, Learning Design, MOOCs

1. INTRODUCTION

To alleviate the challenge of learners devoting time to engage with MOOCs, we have developed a system that try to enable learners to earn money while taking MOOCs, thus "buying time." But how can we enable hundreds of thousands of learners to earn money all at once? To achieve this, we propose that online freelance platforms like UpWork[1] can be included to formulate the solution. If we can automatically provide learners with recommendations of freelance paid tasks which are relevant to the course content, learners can compete for solving these tasks as a means to earn money and thus better justify persisting through the course. As a foundation, our previous study [1] demonstrated that learners can solve these real-world paid tasks in high quality. In this poster, we advance the research by implementing a *Paid Task **Rec**ommender **Sys**tem* (`Rec-$ys`), which automatically collects course-related tasks from UpWork and recommends them to learners. The system has been deployed into the *EX101x Data Analysis: Take It to theMAX()*, which is currently running on the edX platform. The course aims to teach introductory data analysis skills using spreadsheet.

We realize that whether learners will be selected to complete a task depends on not only their knowledge but also their experience. In other words, these learners will probably not be selected or earn money. But even so, we are still interested what effects these financial incentives of real-world tasks would have on learners and what potential strategies could be adopted to turn learners into earners. We hypothesize that, by realizing the financial benefits to be gained from the external freelance platform, learners will exhibit higher engagement and completion rates in the course. By deploying `Rec-$ys` in an experimental setup, we will investigate the following questions:

- What are the effects of (continuously) presenting real-world paid tasks relevant to the course?
- Can learners benefit from real-world paid tasks, e.g., earning money or fostering their interests on the MOOC subject?
- How does the payoff of the task affect learner engagement? For example, are easy tasks with low payment more attractive than difficult tasks with high payment?

2. SYSTEM ARCHITECTURE

To make `Rec-$ys` easily reusable in a variety of MOOCs as well as online freelance platforms, we adopt a modular structure in the development process, as depicted in Figure 1. We briefly describe the structure layer by layer:

MOOC: The MOOC layer serves as the playground for learners to interact with course components as well as `Rec-$ys`. The course is hosted by edX, which allows us to deploy the system as an iFrame component. By doing this, learners can interact with `Rec-$ys` in the same way they interact with other course material. Based on the available course material (e.g., lecture video transcripts,

*The author's research is supported by the *Extension School* of the Delft University of Technology.
†The author's research is supported by the *Leiden-Delft-Erasmus Centre for Education and Learning.*
[1]https://www.upwork.com/

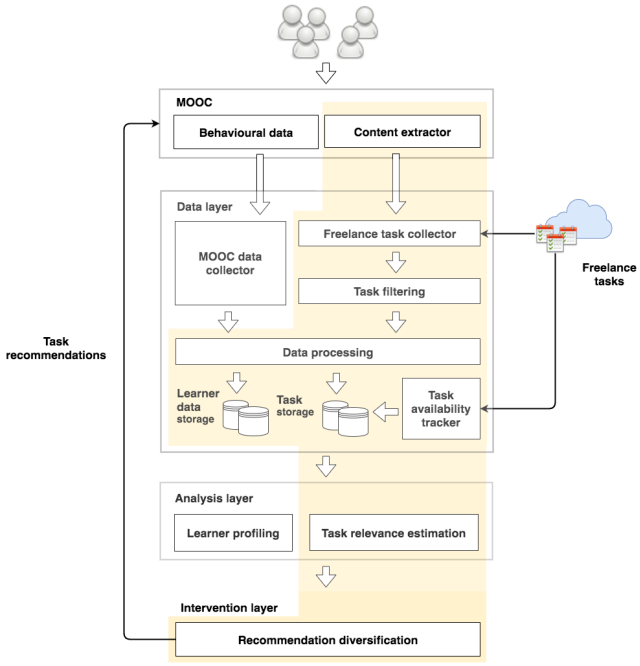

Figure 1: `Rec-$ys` architecture. The yellow overlay indicates which modules have been constructed or are being constructed.

course description), the module *Content extractor* identifies two types of keywords: 1) *general keywords* describing the MOOC subject and 2) *content keywords* specifying the detailed course content. For the data analysis MOOC, these keywords are identified by first extracting the top-k most frequent terms from the lecture video transcripts and then filtering out irrelevant ones by hand. A more automatic and advanced extraction method should be included so as to process courses with different subjects.

Data layer: This layer is responsible for: 1) keeping track of learners' activities and 2) collecting paid tasks from online freelance platforms. To be specific, the *MOOC data collector* collects all of the data generated during learners' interaction with the course material (e.g., watching lecture videos, answering quiz questions) and `Rec-$ys` (e.g., requesting more tasks, clicking task links and jumping to freelance platforms, submitting feedback about the recommended task). On the other hand, the module *Freelance task collector* takes the general keywords generated by the module *Content extractor* as input and search for course-relevant tasks from freelance platforms like UpWork, Witmart, Guru, etc. Currently, we only collect tasks from UpWork. As some retrieved tasks are not relevant to the course (e.g., those with high budget and require much more advanced skills/knowledge to solve), the module *Task filtering* filters out tasks which meet the following criteria: 1) no longer available; 2) without a fixed payment (i.e., hourly job) or the payment exceeds $250. After that, the *Data processing* module translates the collected data into a queryable format and stores them in database. In addition, as there are likely multiple students (plus the freelancers in freelance platforms) competing for solving the same task, the module *Task availability tracker* regularly checks the status of tasks and updates them in the database so that the system will only recommend learners with tasks that are still available.

Analysis layer: This layer aims to analyzing the relevance of tasks for learners based on their interaction with `Rec-$ys`. In the long run, we expect that such relevance (so as the recommendations) can be computed in a individually-personalized manner. However, as we currently have little knowledge about what learner features should be considered when calculating this relevance, the module *Task relevance estimation* only uses the content keywords generated by the module *Content extractor* as input and calculate the relevance score as these keywords' occurrence in the task title, description and required skills stated by the task publisher.

Intervention layer: To avoid a learner keeps receiving the same task or tens of thousands of learners compete for the same task, this layer dedicates to diversifying tasks recommended to different learners. At present, we use a randomization method to achieve. For a learner, we first retrieve the top-k most relevant tasks from the database; then, two out of these tasks are randomly selected as the returned recommendations.

3. ONGOING WORK

`Rec-$ys` has been deployed in a MOOC which runs from November 22, 2016 to May 23, 2018 in a self-paced mode. This enables us to continuously collect tasks from UpWork and recommend them to learners while observing how learners interact with these tasks over an extended period of time. In the next stage, we plan to: 1) analyze learner activity data in answering our research questions; 2) improve the measurement of task relevance for the course; 3) explore additional strategies to diversify recommendation results; and 4) explore methods about how our learners should be mentored so that they are likely to win task bidding in freelance platforms [2].

4. REFERENCES

[1] G. Chen, D. Davis, M. Krause, E. Aivaloglou, C. Hauff, and G. J. Houben. Can learners be earners? investigating a design to enable mooc learners to apply their skills and earn money in an online market place. *IEEE Transactions on Learning Technologies*, pages 1–12, 2016.

[2] R. Suzuki, N. Salehi, M. S. Lam, J. C. Marroquin, and M. S. Bernstein. Atelier: Repurposing expert crowdsourcing tasks as micro-internships. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, pages 2645–2656, New York, NY, USA, 2016. ACM.