THESIS PROPOSAL: TASK CLASSIFICATION AND RECOMMENDATION FOR CROWDWORKERS

by

Riley Miller

A thesis submitted to the Faculty and the Board of Trustees of the Colorado School of Mines in partial fulfillment of the requirements for the degree of Master of Science (Computer Science).

Golden, Colorado

Date __________________________

Signed: __________________________

Riley Miller

Signed: __________________________

Dr. Chuan Yue
Thesis Advisor

Golden, Colorado

Date __________________________

Signed: __________________________

Dr. Tracy Camp
Department Head
Department of Computer Science

ABSTRACT

TODO: Adding after draft of proposal

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

The following contains my proposal for the research I plan to conduct under the guidance of Dr. Chuan Yue in pursuit of my Masters of Computer Science degree from Colorado School of Mines. My research will be centered around the development of a machine learning approach to improve the efficiency of crowdworkers in crowdsourcing platforms.

In this section I will outline the problem that we are trying to solve, provide a high level overview of crowdsourcing platforms, ecosystems, and tools, as well as highlight traditional approaches to developing recommendation systems.

1.1 Background

Crowdsourcing platforms have become increasingly popular over the course of the last several years, introducing the concept of utilizing "crowd"' intelligence to complete work. [?] These platforms are historically architected to facilitate the relationship between requesters, who create work on the platform in the form of items commonly referred to as tasks, and crowdworkers who are real people that complete tasks that are curated by the requesters. There are a variety of different crowdsourcing platforms that have entered the crowdsourcing market such as Figure Eight, ClickWorker, Spare5, Respondent, Swagbucks, and the most popular platform: Amazon Mechanical Turk (MTURK). These platforms are used for a variety of different use cases ranging from collecting data via surveys, image/video annotation, translations, spreadsheet modifications and analysis, writing, to really anything imagineable. The typical workflow in crowdsourcing platforms starts with requesters who break work that they need to get completed down into tasks. This process usually involves writing a description of what needs to be done, how it needs to be completed, providing the input the crowdworker needs to complete the work, setting a qualification so that only certain crowdworkers may have access to the task based off their work history, setting a lifetime on the task for when the work needs to be completed, and setting a reward for successful completion of the task. All of these specifications vary by platform since a universal specification standard between different crowdsourcing platforms fails to exist. After the requesters finish a task and publish it to the crowdsourcing platform, the task becomes available to crowdworkers (depending on the platform it may only be made available to crowdworkers who match the qualification set by the requesters) who can then attempt to complete the task. After the crowdworkers are satisfied with their work on a given task they can then submit their work on a task to the requesters for approval. After this work is received by the requesters, the requesters have the ability to review the work that was completed by the crowdworker and determine whether they will accept the work, reject the work, or send the task back to the crowdworker because the quality of the work was insufficient. This workflow in crowdsourcing platforms presents several inherent issues that need to be improved in order for crowdsourcing platforms to continue to grow.

1.2 Requester Role in Crowdsourcing Platforms

Requesters often utilize crowdsourcing platforms to outsource large pieces of work to a distributed workforce to leverage the skills, time, and experience of crowdworkers. This allows requesters to offload tasks to ultimately save themselves time and allow them to focus on higher priority tasks. Some of the practical use cases that requesters offload to crowdsourcing sites are bulk tasks that require human input but take a considerable amount of time like data annotation of machine learning and computer vision data sets which requires a large amount of accurately labeled data to fuel deep learning algorithms. One of the primary benefits that crowdsourcing platforms present requesters is that requesters are able to curate a massive amount of tasks and distribute them amongst crowdworkers for a reasonable cost considering the average wage for crowdworkers lingers around $2-$5 and hour [**?** **?**]. Many of the tasks that requesters publish on crowdsourcing sites have small rewards ranging on average from a couple cents to several dollars. The low average cost of tasks in crowdsourcing platforms inherently allows researchers to publish a large number of tasks which also simultaneously benefits crowdworkers and crowdsourcing platforms by increasing the volume of work available. Some of the issues that exist for requesters in crowdsourcing platforms are the amount of time it takes to curate tasks, the amount of time it takes to review and approve tasks completed by crowdworkers, and the diversity in the quality of the work that crowdworkers perform.

1.3 Crowdworker Role in Crowdsourcing Platforms

While requesters supply crowdsourcing platforms with the volume of work, crowdworkers supply the labor. The primary motivation of crowdworkers discovered empirically as apart of a study in 2018 by a team of crowdsourcing researchers was to earn money [**?**]. The distributed workforce of crowdworkers complete tasks all over the world with a wide range of skillsets and experience while providing human intelligence to complete work published by requesters. The researchers who performed the survey on crowdsourcing workers in Amazon MTURK discovered that over half (61.7%) of crowdworkers were employed fulltime and 50.2% of workers had recieved a four year education.[**?**] This data shows that many of the crowdworkers don't rely on their crowdsourcing work as a primary source of income and that many crowdworkers are formally educated. However, since many crowdworkers aren't reliant on crowdsourcing platforms as their primary source of income this suggests that quality will be a lower priority for crowdworkers in exchange for efficiency as the repercussions for low quality work have much less severe implications than they would for low quality work with their full-time employers. With the primary motivation of crowdworkers regarding the wage they receive from crowdsourcing platforms, efficiency is imperative and many of the crowdworkers' interactions with crowdsourcing platforms will be guided by the overarching goal of making as much money as possible as efficiently as possible. To help optimize their time during crowdsourcing working sessions, crowdworkers use a variety of different tools to help improve their efficiency while completing tasks.

1.4 Requester vs. Crowdworker Imbalance

The two differing motives of both requesters and workers develops inherent tension between the primary relationship in crowdsourcing platforms. Requesters desire high quality work whereas crowdworkers desire efficient work resulting in two states in the crowdsourcing platform that are relatively mutally exclusive. However, the power in this relationship leans heavily towards requesters who are able to dominate the dynamic using several features that are common amongst crowdsourcing platforms such as: the ability to specify certain qualifications that workers must meet to access certain tasks and having ultimate authority over accepting or rejecting the work that crowdworkers complete based on their own personal, subjective view on quality. [**?** **?**] This imbalance places workers at a severe disadvantage in crowdsourcing platforms in addition to flaws in the platforms themselves. Some of the disadvantages that crowdsourcing platforms inflict on workers include: naive search functionality for surfacing tasks, lack of metrics for how long it will take to complete a task, stated feasibility of tasks, and a lack of estimated wage value for tasks [**?**]. Crowdworkers are at an inherent disadvantage in crowdsourcing platforms which harms worker participation, a fundamental requirement for crowdsourcing platforms to be successful.

1.5 Crowdworker Difficulties

More specifically, a survey on crowdworkers showed that the biggest painpoints for crowdworkers in the Amazon MTURK crowdsourcing platform are loss of compensation on rejected or returned tasks, data on whether or not a given task is even completeable (often times a task may not even be completeable because requesters may not have provided enough information for workers to successfully complete the task), and the amount of time it takes to find a task or switch context between different types of tasks [**?**]. The two pain points that I plan to explore as apart of my research are decreasing the number of returned or rejected tasks by providing data upfront to workers regarding the feasibility of a task as well as decreasing the amount of time spent searching for tasks. Another interesting data point the researchers

(Kaplan et al) collected from their survey on crowdworkers was that 30% of respondents said that finding a task was reported “4 - Very“ or “5 - Extremely“ difficult, but probably even more intriguing was the data that the most pertinent reason for ending a session was that they were unable to find a task worth doing (48% said this ranked as an “5 - Extremely Important“ reason in terminating a crowdsourcing session). [**?**] This dissatisfaction of crowdworkers shows that the current working model for crowdsourcing platforms needs to be improved to improve the user experience of crowdworkers. Improving the user experience of crowdworkers is imperative to the continued adoption of crowdsourcing platforms by new crowdworkers and has a lot of opportunity for research to improve the experience of crowdworkers in these platforms including the use of machine learning to aid crowdworkers in making intelligent decisions regarding task selection and to intelligently surface relevant tasks that crowdworkers believe are worth doing.

1.6 Goal

1.7 Approach

After digging into crowdsourcing platforms, I will be focusing my research on improving the efficiency of crowdworkers in crowdsourcing platforms. I believe this is an important facet of the crowdsourcing ecosystem which would greatly benefit crowdsourcing platforms due to the lack of tooling surrounding task recomendation for crowdworkers, the poor user experience of crowdworkers in the current state of crowdsourcing platforms, and the disadvantage of crowdworkers in crowdsourcing platforms.

CHAPTER 2

RELATED WORK

2.1 Existing Tools

Crowdworkers have created a variety of user plugins and browser extensions to help out the community of crowdworkers to try and improve crowdworker efficiency. Some of the more prominent plugins focus on batching similar tasks to reduce the time users spend switching context between dissimilar tasks and plugins for workers to rate requesters based off their interactions directly and indirectly with how the tasks are structured. I've curated a list of tools used by crowdworkers from my own research and from the results of a survey of crowdworkers [**?**].

- **HIT Scraper:** A web scraper that helps provide additional search filters not offered as apart of the native offering for Amazon MTurk.
- **MTurk Suite:** A browser extension used to combine a plethora of other crowdworker tooling.
- **Turkopticon:** A web tool that allows crowdworkers to rate requesters and tasks.
- **Greasemonkey/Tampermonkey:** A browser extension that allows crowdworkers to run custom scripts to help boost efficiency in crowdsourcing platforms.
- **Panda Crazy:** A tool used by crowdworkers to batch similar tasks together.
- **Turkmaster:** A script that monitors search pages, requesters, and can automatically accepts tasks on Amazon MTurk.
- **Block Requesters:** Allows users to block and ignore requesters from search results, useful if crowdworkers have a bad experience with a requester and wish to avoid future interactions.

- **Pending Earning:** Allows crowdworkers to view pending earnings for tasks that have been completed and submitted but not approved.
- **MTurk HIT DataBase:** Improved interface for searching tasks that you have worked on previously, Amazon MTurk.
- **MTurk Worst Case Scenario Calculator:** Tool to calculate approval rate and how many rejections it would take to drop your approval percentage to a certain threshold.
- **MTurk Dashboard HIT Status links:** A tool which provides quick access to rejected and pending tasks, Amazon MTurk.
- **MTurk Engine:** A browser extension that combines additional search filters with batching, as well as automated task watching for Amazon MTurk. This tool also includes a dashboard to track earnings.

These tools show the desire for improvement in the crowdworker user experience from the native crowdsourcing platform and a high level of community involvement and support for crowdworkers. Although there is existing tooling for batching similar tasks using keywords and search filters there still lacks tooling for common painpoints highlighted in the survey results collected by (Kaplan et al). Some of the main areas of the crowdworker user experience that still need to be addressed are: surfacing useful recommendations of tasks that are curated based off worker history, expertise, and preferences, content-based analysis of the feasibility of a task, and intelligent automation based on manual user search for tasks.

2.2 Existing Recommendation Systems

CHAPTER 3

PROPOSED WORK

3.1 Potential Approaches

Recommendation systems have been around since the early days of the internet and several of the techniques used early on are still heavily relied on today and in some cases their general principles are utilized to develop intelligent approaches in conjunction with traditional methods. The two primary methods that the majority of recommendation systems are based off of are collaborative filtering and content based approaches. [**?**]

3.1.1 Collaborative Filtering

Collaborative filtering is the most popular recommendation system in use historically and today. This approach pertains to considering user data when processing information for recommendations, or in simpler terms, making predictions based off of other users. This approach has been in use since the earlier days of the internet, getting traction around the advent of e-commerce websites. An example of how this approach is applied is that for an e-commerce site, after items on the site are purchased by customers, many times the customer who purchased the item will be prompted to rate the product or to leave a review. You probably recognize this pattern from any time you buy something off Amazon and you get an email a couple of days after your package arrives asking you how you liked the product or if you'd be willing to leave a review. Anyways, once customers start interacting with items in the e-commerce store, collaborative filtering will use this data to recommend items to users who are similar to the people who originally bought the item. One of the primary stumbling blocks with this approach is the cold-start problem where the recommendation system isn't able to surface relevant recommendations at the inception of the system since there are a limited number of user interactions with items and thus the quality and relevancy of the recommendations will be poor. [**?**] Collaborative filtering is also only effective on data that has already existed on the service and is unable to make recommendations for new items until the new items are interacted with. One thing that is interesting to note is that this idea of collaborative filtering can be used in conjunction with deep learning techniques to create intelligent hybrid recommendation systems based off of basic collaborative filtering principles. One example of this is in a study in which researchers created a model using a technique that they called collaborative deep learning (CDL) which leveraged a hierarchical Bayesian model which combined collaborative filtering and the content based approach using deep learning to create better recommendations . [**?**]

3.1.2 Content Based

The content based approach is less commonly used than collaborative filtering but it is still a prevalent approach for building recommendation systems. Content based recommender systems are based on similarity of items in the recommender system. These systems leverage the matching of similar items to a user profile. [**?**]. Content based approaches inherently don't have to deal with the cold-start problem that is common in collaborative filtering recommendation systems since they don't bother with the behavior of similar users but of similar items within the system. These recommendation systems are commonly used in systems where content has frequent turnover, or where data is sparse [**?**].

3.1.3 Text Classification

Text classification is the process of determining similarity from free text. This is a common area of natural language processing research and pertains to a wide array of different applications. Text classification can leverage either supervised or unsupervised techniques, providing a lot of flexibility for differnet approaches. Supervised text classification techniques use labeled training data to develop models that can then be used to categorize unlabeled free text. The most important piece of the supervised approach to text classification is possesing a large, annotated corpus to train accurate models. Alternative to the supervised approach, unsupervised learning can also be applied to text classification problems. Unsupervised learning relies on discovering similarities within a dataset and grouping together similar data, this approach is typically referred to as clustering. One of the beautiful things about unsupervised learning is that it doesn't rely on annotated training data, it will just discover similarities from the data provided. These two approaches can be applied to improving efficiency of crowdworkers.

3.2 Research Focus

3.3 Potential Algorithms

3.4 Dataset

Explain that Dr. Yue will be collecting the data needed for recommendation system.

3.5 Evaluation

explain metrics and experiments

3.5.1 Metrics

3.5.2 Experiments

CHAPTER 4

PROJECT PLAN

4.1 Outline of Project Timeline

- Dates
- Dates
- Dates
- Dates
- Dates

REFERENCES CITED

[] H. Shah H. Cheng L. Koc J. Harmsen T. Shaked T. Chandra H. Aradhye G. Anderson G. Corrado W. Chai M. Ispir R. Anil Z. Haque L. Hong V. Jain X. Liu. Wide & deep learning for recommender systems. In *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems*, September 2016.

[] Chong Sun, Narasimhan Rampalli, Frank Yang, and AnHai Doan. Chimera: Large-scale classification using machine learning, rules, and crowdsourcing. *Proceedings of the VLDB Endowment*, 7(13):1529–1540, 2014.

[] Mohammad Allahbakhsh, Boualem Benatallah, Aleksandar Ignjatovic, Hamid Reza Motahari-Nezhad, Elisa Bertino, and Schahram Dustdar. Quality control in crowdsourcing systems: Issues and directions. *IEEE Internet Computing*, 17(2):76–81, 2013.

[] Ali Mamdouh Elkahky, Yang Song, and Xiaodong He. A multi-view deep learning approach for cross domain user modeling in recommendation systems. In *Proceedings of the 24th International Conference on World Wide Web*, pages 278–288. International World Wide Web Conferences Steering Committee, 2015.

[] Amirata Ghorbani and James Y. Zou. Data shapley: Equitable valuation of data for machine learning. In *ICML*, 2019.

[] Ivens Portugal, Paulo Alencar, and Donald Cowan. The use of machine learning algorithms in recommender systems: A systematic review. *Expert Systems with Applications*, 97:205–227, 2018.

[] Marco De Gemmis, Leo Iaquinta, Pasquale Lops, Cataldo Musto, Fedelucio Narducci, and Giovanni Semeraro. Preference learning in recommender systems. *Preference Learning*, 41:41–55, 2009.

[] H. Wang N. Wang and D.-Y Yeung. Collaborative deep learning for recommender systems. *Proc. KDD*, pages 1235–1244, 2015.

[] Francois Chollet. *Deep Learning With Python.* Manning, Shelter Island, NY, 2018.

[] Greg Linden, Brent Smith, and Jeremy York. Amazon. com recommendations: Item-to-item collaborative filtering. *IEEE Internet computing*, (1):76–80, 2003.

[] Shumpei Okura, Yukihiro Tagami, Shingo Ono, and Akira Tajima. Embedding-based news recommendation for millions of users. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1933–1942. ACM, 2017.

[] Xuejian Wang, Lantao Yu, Kan Ren, Guanyu Tao, Weinan Zhang, Yong Yu, and Jun Wang. Dynamic attention deep model for article recommendation by learning human editors' demonstration. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 2051–2059. ACM, 2017.

[] Sai Wu, Weichao Ren, Chengchao Yu, Gang Chen, Dongxiang Zhang, and Jingbo Zhu. Personal recommendation using deep recurrent neural networks in netease. In *2016 IEEE 32nd International Conference on Data Engineering (ICDE)*, pages 1218–1229. IEEE, 2016.

[] Toni Kaplan, Susumu Saito, Kotaro Hara, and Jeffrey P Bigham. Striving to earn more: a survey of work strategies and tool use among crowd workers. 2018.

[] Sihem Amer-Yahia, Eric Gaussier, Vincent Leroy, Julien Pilourdault, Ria Borromeo, and Motomichi Toyama. Task composition in crowdsourcing. 2016.

[] Michael J Pazzani and Daniel Billsus. Content-based recommendation systems. In *The adaptive web*, pages 325–341. Springer, 2007.

[] A. G. Reference, Magic Man, and Cool Cat. The Title of a GOOD Reference. *The Journal of Referency-Goodness*, pages 1–1000, 2009.

[] Kotaro Hara, Abigail Adams, Kristy Milland, Saiph Savage, Chris Callison-Burch, and Jeffrey P Bigham. A data-driven analysis of workers' earnings on amazon mechanical turk. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, page 449. ACM, 2018.

[] Siou Chew Kuek, Cecilia Paradi-Guilford, Toks Fayomi, Saori Imaizumi, Panos Ipeirotis, Patricia Pina, and Manpreet Singh. The global opportunity in online outsourcing. 2015.