THESIS: TOPIC MODELLING OF AMAZON MTURK

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

<Insert abstraction upon completion of the document>

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 crowd workers 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 [1]. 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 crowd workers 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, CrowdFlower, 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 theoretically anything. The typical workflow in crowdsourcing platforms starts with requesters who breakdown work that they need to get completed into smaller chunks called tasks. This process usually involves writing a description of what needs to be done, how it needs to be completed, providing the input the crowd worker needs to complete the work, setting a qualification so that only certain crowd workers 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. [2] After the requesters finish a task and publish it to the crowdsourcing platform, the task becomes available to crowd workers (depending on the platform it may only be made available to crowd workers who match the qualification set by the requesters) who can then attempt to complete the task. After the crowd workers 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 crowd worker and determine whether they will accept the work, reject the work, or send the task back to the crowd worker 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 crowd workers. [1] 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 datasets 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 crowd workers for a reasonable cost considering the average wage for crowd workers lingers around $2-$5 and hour [3, 4]. In fact, many of the tasks that requesters publish on crowdsourcing sites have small rewards ranging on average from only 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 crowd workers 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 crowd workers, and the diversity in the quality of work that crowd workers perform.

1.3 Crowd Worker Role in Crowdsourcing Platforms

While requesters supply crowdsourcing platforms with the volume of work, crowd workers supply the labor. The primary motivation of crowd workers discovered empirically as apart of a study in 2018 by a team of crowdsourcing researchers was simple: earn money [3]. The distributed workforce of crowd workers 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 crowd workers were employed fulltime and 50.2% of workers had recieved a four year education.[3] This data shows that many of the crowd workers don't rely on their crowdsourcing work as a primary source of income and that many crowd workers are formally educated. However, since many crowd workers aren't reliant on crowdsourcing platforms as their primary source of income this suggests that quality will be a lower priority for crowd workers 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 crowd workers regarding the wage they receive from crowdsourcing platforms, efficiency is imperative and many of the crowd workers' 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, crowd workers use a variety of different tools to help improve their efficiency while completing tasks.

1.4 Requester vs. Crowd Worker 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, crowd workers desire efficient work resulting in two states in the crowdsourcing platform that are relatively mutally exclusive. However, unfortunately for crowd workers, 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 crowd workers complete based on their own personal, subjective view on quality. [2, 3] 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 [3]. Crowd workers are at an inherent disadvantage in crowdsourcing platforms which harms worker participation, a fundamental requirement for crowdsourcing platforms to be successful.

1.5 Crowd Worker Difficulties

More specifically, a survey on crowd workers showed that the biggest painpoints for crowd workers 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 [3]. 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 crowd workers 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). [3] This dissatisfaction of crowd workers shows that the current working model for crowdsourcing platforms needs to be improved to improve the user experience of crowd workers. The crowdsourcing community has a vested interest to improve the user experience of crowd workers in order to continue to facilitate the adoption of crowdsourcing platforms by new crowd workers. This area of crowdsourcing has a lot of opportunity for research and is relatively sparse compared to other fields of research on crowdsourcing platforms.

1.6 Goal

To analyze common topics and trends found in crowdsourcing tasks on Amazon MTURK to improve crowdworker efficiency.

1.7 Approach

I will be focusing my research on improving the efficiency of crowd workers in crowdsourcing platforms through the analysis of crowdsource tasks to determine common topics and trends found in crowdsourcing tasks. Specifically I will focus on applying unsupervised learning techniques in the form of topic modelling to improve the understanding of crowdsourcing tasks curated by researchers as well as how this information can be leveraged to optimize crowdworker efficiency.

CHAPTER 2

RELATED WORK

2.1 Topic Modeling of Crowdsourcing Tasks

Add stuff here on the topic modeling of crowdsourcing tasks

2.2 Crowdworker Efficiency

2.2.1 Existing Tools

Crowd workers have created a variety of user plugins and browser extensions to help out the community of crowd workers to try and improve crowd worker 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 crowd workers from my own research and from the results of a survey of crowd workers [3].

- **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 crowd worker tooling.
- **Turkopticon:** A web tool that allows crowd workers to rate requesters and tasks.
- **Greasemonkey/Tampermonkey:** A browser extension that allows crowd workers to run custom scripts to help boost efficiency in crowdsourcing platforms.
- **Panda Crazy:** A tool used by crowd workers 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 crowd workers have a bad experience with a requester and wish to avoid future interactions.
- **Pending Earning:** Allows crowd workers 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 crowd worker user experience from the native crowdsourcing platform and a high level of community involvement and support for crowd workers. 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 crowd worker 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 Batching to reduce time spent switching context.

2.2.2 Existing Recommendation Systems

There are several different approaches to developing recommendation systems, the two most common are collaborative filtering and content based approaches. Collaborative filtering is generally more accurate than content based approaches, however, collaborative filtering struggles with recommending new items, a characteristic of tasks in crowdsourcing platforms. Alternatively, content based approaches are based on determining the similarity between items and how they associate with users in the platform, crowd workers in our use case.

Often times in platforms that are trying to develop a recommendation system where new data is constantly entering the platform and old data is constantly becoming outdated, the platform will apply a content based approach instead of collaborative filtering to address the cold start problem and the sparseness of interaction of similar users on similar tasks. One example of this use case to a similar problem is the use of a content based approach to surface relevant news articles to content reviewers for media corporations like Buzzfeed. The researchers (Wang et al.) of the study leverage a character level neual network langauge model (a CNN) to perform low-level textual feature learning [5]. This is a very applicable approach to my research since tasks in crowdsourcing platforms are constantly getting outdated after a worker completes a task, the specific task is resolved and will not be reused, making any type of collaborative filtering approach forseeably ineffective.

Researchers (Yuen et al.) developed a matrix factorization method of recommending tasks in Amazon MTurk. [6] Their approach was predicated on using matrix factorization on crowd worker performance history as well as their task searching history to surface relevant reccomendations to crowd workers. Another set of researchers applied two different techniques based on implicit modeling of user history leveraging a Bag-of-words Approach and a classification approach [7]. One of the downfalls of their research was that they only used 24 users to evaluate their approach.

From my initial investigation, I haven't found any research that is directly related to my approach of extracting necessary skills from tasks using text classification. The primary targeted application of my research is to eventually create a content based recommendation system that will match a user to a task in a crowdsourcing platform based on implicit skills the user has obtained (based on previously completed tasks and user provided characteristics) and the required skills that are needed to complete a task.

2.2.3 Text Classification of Crowdsourcing Tasks

Interesting related work in the field of text classification of tasks in crowdsourcing platforms can be found in an analysis of the dynamics of crowdsourcing performed by (Difallah et al) in which they used supervised learning to classify types of tasks in Amazon MTurk [8]. This research is extremely useful for my approach and will allow me to use the common categories that they defined as apart of their research to gather data from crowd workers as apart of my research on the **skills** that are required to complete crowdsource tasks.

CHAPTER 3

TOPIC MODELING APPROACH AND FOCUS

3.1 Dataset

As explained in the "Research Focus" section above, Dr. Yue has created a team of researchers who will be handling the development of a client side browser extension to web scrape crowdsourcing platforms for task data as well as to track user profile information and user interactions with the crowdsourcing sites. We will be able to quickly web scrape crowdsourcing sites to gather enough data for the text classification I plan to perform on crowdsource tasks. However, we are envisioning that it will take considerably more time to gather enough user data to develop a machine learning recommendation system which utilizes content based recommendations.

3.2 Evaluation

In this section I will outline some of the experiments I plan to conduct to develop an effective algorithm which will be able to accurately extract skills that are required to complete tasks.

3.2.1 Metrics

To determine the success of different approaches I will allocate the dataset that we collect from scraping the crowdsourcing site into an 80/20 split where 80% of my data set will be used for training the machine learning algorithms while reserving the remaining 20% of the data set to test the accuracy of the models and to evaluate which approach is most effective for determining which skills are necessary to perform tasks. Due to the timeline of the grant and time constraint of my graduation date, I will be leveraging offline testing to perform my analysis. Ideally we would be able integrate the models into the browser extension and perform A/B testing in live environments to test the performance of the different techniques but that isn't a viable option given the timeline of the project.

3.2.2 Experiments

The two different types of experiments that I plan to conduct can be broken down into the two high-level categories of supervised and unsupervised learning.

CHAPTER 4

WHAT ARE THE PRIMARY TOPICS OF AMAZON MTURK HITS?

Answer the question

CHAPTER 5

ANALYSIS

Additional analysis and questions

CHAPTER 6

CONCLUSION

REFERENCES CITED

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

[2] 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.

[3] 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.

[4] 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.

[5] 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.

[6] Man-Ching Yuen, Irwin King, and Kwong-Sak Leung. Task recommendation in crowdsourcing systems. In *Proceedings of the first international workshop on crowdsourcing and data mining*, pages 22–26. ACM, 2012.

[7] Vamsi Ambati, Stephan Vogel, and Jaime Carbonell. Towards task recommendation in micro-task markets. In *Workshops at the Twenty-Fifth AAAI Conference on Artificial Intelligence*, 2011.

[8] Djellel Eddine Difallah, Michele Catasta, Gianluca Demartini, Panagiotis G Ipeirotis, and Philippe Cudré-Mauroux. The dynamics of micro-task crowdsourcing: The case of amazon mturk. In *Proceedings of the 24th international conference on world wide web*, pages 238–247. International World Wide Web Conferences Steering Committee, 2015.