Micro-Task Skill Inference for Crowd Workers

Thesis Proposal

Riley Miller

Background

Crowd Worker Difficulties

- Rejected Tasks
- Whether a task is feasible or not
- Time spent searching for tasks

[3]

Goal

To improve crowd worker efficiency through skill inference of crowdsourcing tasks.

Approach

- **Text classification of tasks to determine the necessary skills to complete a given task.**
- Potential Applications:
  - Intelligent Batching: Clustering of tasks based on required skills, reduce time spent switching between different types of tasks. (Similar to PANDA, Preview and Accept but with using machine learning to surface similar tasks vs. any task) Integrated into plugin.
  - Content-Based Recommendation: If we obtain enough data of users interacting with tasks, can create a content-based recommendation system based off of skills that the users have and the skills that are required to complete a task.

Related Work

Existing Tools

- HIT Scraper: A web scraper that helps provide additional search filters not offered as apart of the native offering for Amazon MTurk.
- Panda Crazy: A tool used by crowd workers to organize tasks together.
- Turkopticon: A web tool that allows crowd workers to rate requesters and tasks.
- Pending Earning: Allows crowd workers to view pending earnings for tasks that have been completed and submitted but not approved.

[3]

Existing Recommendation Systems

- Researchers developed a task recommendation system in Amazon MTurk using matrix factorization of worker performance history and worker task searching history. [6]
- Researchers used Bag of Words and Classification on worker history to provide recommendations. [7]

Text Classification of Crowdsourcing Tasks

- Researchers used text classification (supervised learning) to classify different types of tasks. [8]

Proposed Work

Collaborative Filtering

- Recommendations systems built on providing recommendations based on the behavior of other users.

Issues:

- Sparse datasets [5]
- Cold-start problem, only able to make recommendations on existing items [5]

Content-Based

- Recommendation system built on the similarity of other items in the platform.
- Matches items based on user profile. [11]

Issues:

- Not going to have user interaction data
- Not as accurate as collaborative filtering [7]

Text Classification

- Extracting similarity from free-text.
- Commonly used with supervised learning techniques.

Research Focus

The extraction of skills from tasks in crowdsourcing platforms using the dataset collected by Dr. Yue's team.

Potential Algorithms

- Decision Trees
- Rule-based Classifiers
- SVM Classifiers
- Bayesian Classifiers
- Neural Network Classifiers (LSTM RNNs and CNNs)
- Nearest Neighbor Classifiers

[13]

Experiments

Supervised Learning

Survey: Will administer a survey on MTurk to collect data on common skills that are required to complete tasks from the platform. Will focus on identifying skills for tasks in the following areas:

- Information Finding
- Verification and Validation
- Interpretation
- Content Creation
- Surveys and Questionaires
- Content Access

Annotation: Will have crowd workers in MTurk annotate a dataset of HITs collected by Dr. Yue's team by labelling *n* skills from a pool of skills curated from the survey to determine the skills required to complete a task.

Algorithms: Will then test different algorithms on the annotated dataset to develop a machine-learning model for extracting the necessary skills from a task given only metadata and raw text.

Unsupervised Learning

Will apply nearest neighbor and clustering algorithms on dataset of HITs to identify common skills from the dataset for comparison's sake. (May not be valuable to spend time doing this?)

Project Plan

- **Today: Thesis Proposal**
- **December 9:** Address Thesis Proposal feedback, receive data set from Dr. Yue, Create survey for surveying workers on common skills for common tasks and file for IRB Approval for both survey and annotation work (Will file for IRB approval ASAP).
- **December 23:** Received IRB approval for survey and published survey on MTURK (May be sooner depending on IRB approval).
- **January 6:** Approve survey tasks, analyze and store survey data. Received IRB Approval for annotation of HIT data, publish tasks for annotating HIT skill inference dataset to MTurk.

January 20: Approve, store, and analyze completed annotation data. Begin applying shallow learning techniques to annotated dataset.
January 27: Developed and tested shallow learning techniques (Decision Trees, Rule- based, SVM, Bayesian) Begin applying Deep Learning techniques to dataset.
February 10: Developed and test deep learning techniques (LSTM RNN and CNN approaches), begin writing paper for conference submission.
February 24: Algorithm iteration and improvement, continue writing paper for conference submission, start drafting thesis.

Project Plan Cont'd

March 9: Develop visualization and analysis of test results, further iteration and improvement of algorithms, continue writing paper for conference submission, begin writing thesis.
March 23: Continue algorithm analysis, continue writing paper for conference submission, continue writing thesis.
April 6: Defend Thesis
April 13: Complete graduation checkout course, submit signed thesis defense form, upload content approved thesis to ProQuest.
April 17: Thesis formatting approval by 1:00pm.
April 17 - May 7: Continue drafting paper and submit to conferences.
May 7: Graduate

References

[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 crowd- sourcing 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 Con- ference on Knowledge Discovery and Data Mining, pages 2051–2059. ACM, 2017.
[6] Man-Ching Yuen, Irwin King, and Kwong-Sak Leung. Task recommendation in crowd- sourcing 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 Cudr´e-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.
[9] Ivens Portugal, Paulo Alencar, and Donald Cowan. The use of machine learning algo- rithms in recommender systems: A systematic review. Expert Systems with Applications, 97:205–227, 2018.
[10] H. Wang N. Wang and D.-Y Yeung. Collaborative deep learning for recommender systems. Proc. KDD, pages 1235–1244, 2015.
[11] Michael J Pazzani and Daniel Billsus. Content-based recommendation systems. In The adaptive web, pages 325–341. Springer, 2007.
[12] 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.
[13] Charu C Aggarwal and ChengXiang Zhai. A survey of text classification algorithms. In Mining text data, pages 163–222. Springer, 2012.