Overview
Over a series of evaluation experiments conducted using naive judges recruited and managed via Amazon’s Mechanical Turk using a task from Information Retrieval [8], we show that a SVM demonstrates high accuracy when the machine-learner is trained and tested (10-fold validation) on data from a single kind of task and that the method was portable from more complex tasks to simpler tasks, but not vice versa.

Hypothesis
The central problem of crowd-sourcing is eliciting quality work from a possibly anonymous crowd of self-motivated workers often paid on a piece-work basis to solve a number of tasks. The reliability of the results of this kind of evaluation depends on the quality of the workers (hereinafter referred to as judges), some of whom are spammers that operate in bad faith or cannot complete the task. Our hypothesis is that spammers should be detected by a machine-learning classifier, and that the results of a spammer-detection should be portable to the results of a classifier are across different tasks.

Features

Number: The total number of HITs completed per judge. In general, it is difficult to distinguish between real human HITs and those completed, and judges who did less than 3 HITs were not considered to be spammers due to simple lack of data.

Average Time: The average time it took the judge to complete a HIT over all HITs done by the judge. In general, spammers complete HITs with a lower average time than non-spammer judges.

Known Bad: The average score of a judge on a “known-lad” gold-standard HIT. Judges that tended to judge known irrelevant results as either unknown or relevant could be considered spammers.

Known Good: The average score of a judge on a “known-lad” gold-standard HIT. Judges that tended to judge known relevant results as either unknown or irrelevant could be considered spammers.

Average Score: The average score of a judge across all HITs. Spammers that uniformly judged all HITs as relevant would get a strangely uniform score in comparison to non-spammer judges.

Experimental Design
Previous work has predicted using simulated data [3] and Naïve Bayes over a single task [2], while in contrast this work engages multiple tasks and machine-learners using a real-world information retrieval task, described more detail in [1]. Tasks consisted of judging the list of records from a Semantic Web data set for relevance (relevant, irrelevant, unknown) given a particular natural language term query. The first task, called the entity task, had only simple natural language descriptions of entities such as ‘Hugh Downs’ as queries. In the second task, called the complex task, the queries consisted of questions describing a list of one or more entities such as ‘astronauts who walked on the Moon’. The entity task is considered a sub-task of the complex task. A total of 9673 results were given in the simple task (32 spammers of 242 judges, $\kappa$ of .803) and 9675 results in the complex task, with HITs given in sets of 12 with 3 gold-standard (25 spammers of 89 judges, $\kappa$ = .704). Experts had a Fleiss’ $\kappa$ of .803 for the entity task and a $\kappa$ of .704 for the complex task. Null hypothesis accuracy is 87% for the entity task and 72% for the complex task. Judges were given the same features as the machine-learner.

Results and Conclusion
Two different classifiers were used: decision trees (DT) and support vector machines (SVMs), with 10-fold cross validation used to prevent over-fitting. Best results were found using SVMs with reduced constraints using slack variable, with training was performed using radial basis kernel. Results in terms of accuracy below have the null hypothesis as yellow, DTs as red, and SVMs as blue.

References

Entity-Entity: SVM: 97.92% DT: 93.92%
Complex-Complex: SVM: 96.51% DT: 93.05%
Entity-Complex: SVM: 82.95% DT: 70.01%
Complex-Entity: SVM: 88.75% DT: 70.59%

The results are better than the null hypothesis (over 90%) when an SVM is trained on data from the same task but accuracy experiences degrade for as training data the entity task on the complex task, but maintained reasonable accuracy when the complex task was used as training for the simple task. SVMs consistently outperform DTs, showing that easily-transferable ‘rule of thumb’ baselines are not transferable across tasks. Machine-learning based approaches can detect spammers with a very high degree of accuracy even using small training and test data-sets with simple features such as numbers of tasks completed and average time to complete, which are likely task invariant in detecting spammers.