Human Computation
AAAI Technical Report WS-12-08

Machine-Learning for Spammer Detection in Crowd-Sourcing

Harry Halpin
World Wide Web Consortium
MIT, Boston, USA
hhalpin@w3.org

Roi Blanco
Barcelona, Spain
Yahoo! Research
roi@yahoo-inc.com

Abstract

Over a series of evaluation experiments conducted using naive judges recruited and managed via Amazon's Mechanical Turk facility using a task from information retrieval (IR), we show that a SVM shows itself to have a very high accuracy when the machine-learner is trained and tested on a single task and that the method was portable from more complex tasks to simpler tasks, but not vice versa.

Introduction

The central problem of crowd-sourcing is eliciting quality work from a possibly anonymous 'crowd' of wed-mediated 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*). There are two fairly independent sources of poor quality work: **Bad faith** judges make no attempt to actual perform the task but rather select answers uniformly or randomly as quickly as possible to get paid and **unsuitable** judges that make every effort to perform the task but either do not understand or lack the necessary abilities to carry out the task. While it may be possible to distinguish the two categories, both are called *spammers* as both are sources of error and are generally not paid. There has been surprisingly little research on detecting spammers automatically. Research has described the statistical facets of spammers rather than using machine-learning to predict spammers (Kern, Thies, and Satzger 2010), although similar work focused on detecting spammers albeit on simulated data (Vuurens, Vries, and Eickhoff 2011). The most attention has been paid to using expectation-maximization (EM) to detect spammers after the data has been collected (Ipeirotis, Provost, and Wang 2010), but this approach focuses on normalizing the scores in compensating for task difficulty rather than the detection of spammers per se. The use of machine-learning to detect spammers has been explored in terms of optimal labeling using Naive Bayes machine-learning in IR tasks (Kumar and Lease 2011), but this work has not explored a range of machine-learners over tasks of varying complexity.

Experimental Tasks

In our experiment, tasks consisted of judging the list of records from a Semantic Web data-set for relevance given a particular natural language term query. The task used a ordinal three-point scale of relevant, unknown, and irrelevant. For example, the query 'iowa energy' found records describing both the company Iowa Energy and simply a list of the states in the USA, of which the former would be relevant and the latter irrelevant. We ran two distinct tasks: the first task, called the *entity task*, had for queries only simple natural language descriptions of entities such as 'Hugh Downs', while 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' of which 'Neil Armstrong' would count as relevant. We used a publicly available data-set [1] for our experiment. For the entity task, a total of 100 queries were ran by different search systems, while 50 queries were ran by the complex task. The results of the search algorithms were pooled and then divided into HITs, where each HIT consisted of a natural language term and 12 returned results plus 3 *gold-standard* result in each task, where each gold-standard result is defined as either a 'known-good' or 'known-bad' result where the correct answer was known beforehand as they were judged manually using a pool of known human experts. These HITs were then presented to crowd-sourced judges using Amazon Mechanical Turk. A total of 9673 results were judged in the simple task and a total of 5675 results were done in the complex task.

In order to create a more reliable set of spammer classifications, each judge in the entire data-set was independently labeled by three experts as either a spammer or not with the assumption that spammers could only be reliably identified if they did more than 3 HITs. The experts had a Fleiss' κ of .803 for the entity task and a κ of .704 for the complex task. The difference in κ values shows that the complex task is harder even for experts to determine whether or not a particular judge is a spammer, with the final score created via voting amongst the three expert judgments. This resulted in a total of 32 spammers identified (of a total of 242 judges) for the the entity task and a total of 25 spammers identified (of a total of 89 judges) for the the complex task, so the data-

[1] Available at *http://semsearch.yahoo.com*

Train-Test	SVM Acc.	DT Acc.
Entity-Entity	97.92%	93.92%
Complex-Complex	96.51%	93.05%
Entity-Complex	82.56%	70.01%
Complex-Entity	88.75%	79.59%

Table 1: Accuracy (Acc.) of spammer identification using classifiers (SVM and DT)

set having 57 spammers out of 331 workers. Null hypothesis accuracy (classifying everyone as non-spammers) is 87% for the entity task and 72% for the complex task.

Machine Learning Experiment

Our hypothesis is that spammers should be detected by a machine-learning classifier. In particular, we are also interested in how portable the results of a classifier are across different tasks and what types of classifiers perform better than others. A small number of features were employed to detect spammers (the same used by the human judges in our manual identification of spammers), namely:

- **Number**: The total number of HITs completed per judge. In general, it is difficult to detect a spammer by how many HITs they 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 too 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-bad' 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-bad' 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.

For the particular problem of detecting spammers in our data-set, we experimented with two different classifiers, decision trees (DT) and support vector machines (SVMs). In all experiments, 10-fold cross validation was used to prevent over-fitting. Table 1 presents the performance of the SVM and the DT on the data-set described above. The column *Train-Test* indicate the data-set employed for learning and evaluating the model respectively. The best results were found using SVMs with relaxed constraints using slack variable, with training was performed using radial basis kernel.

The results are definitely acceptable and far better than the null hypothesis (in the high 90% for both kinds of task) when an SVM is trained on data from the same task. However, there is considerable degradation (reducing accuracy to in the 80% range, near the null hypothesis) for using as training data data from a crowd-sourcing task different than the task at hand (although such degradation is worse for DTs rather than SVMs), leading one to believe that features (and thus spammer classification models) are non-portable amongst tasks. However, spammers identified in a task that is more complex than the task at hand can be used to train the machine-learner to identify spammers in a simpler task above baseline, but that performance degrades to baseline when trying to use spammers identified in a less complex task as training data to identify spammers in a more complex task. SVMs consistently outperform DTs, showing that easily-transferable 'rule of thumb' baselines such as 'avg. rating known bad/avg rating is 1' are not sufficient, and in-line with previous work that shows that EM performs similar if not worse than voting (Vuurens, Vries, and Eickhoff 2011) and worse than SVMs.

Conclusions and Future Work

Overall, we have successfully shown that machine-learning based approaches can detect spammers with a very high degree of accuracy even using fairly small training and test data-sets, and that some very simple features such as number of tasks completed and average time to complete, are likely task invariant in detecting spammers. Our future work in spammer identification should look at more widely disparate kinds of tasks and continue to study the portability of models between tasks, using a larger number of different tasks and involving more raters to further refine the validity of the conclusions. As opposed to the current off-line methodology, ideally one could determine online and dynamically per task how much training data is needed to reliably detect spammers. Our early initial results point to the possibility that the automatic identification of spammers by machine-learning algorithms could take much of the pain out of crowd-sourcing tasks.

References

Ipeirotis, P. G.; Provost, F.; and Wang, J. 2010. Quality management on amazon mechanical turk. In *Human Computation Workshop (KDD-HCOMP 10)*.

Kern, R.; Thies, H.; and Satzger, G. 2010. *Statistical Quality Control for Human-Based Electronic Services*, volume 6470/2010 of *Lecture Notes in Computer Science*. Springer Verlag. 243–257. DOI: 10.1007/978-3-642-17358-5_17, http://www.springerlink.com/content/c718300217237723/.

Kumar, A., and Lease, M. 2011. Modeling annotator accuracies for supervised learning. In *Proceedings of the Workshop on Crowdsourcing for Search and Data Mining (CSDM) at the Fourth ACM International Conference on Web Search and Data Mining (WSDM)*, 19–22.

Vuurens, J.; Vries, A. P. D.; and Eickhoff, C. 2011. How Much Spam Can You Take? An Analysis of Crowdsourcing Results to Increase Accuracy. In Lease, M.; Hester, V.; Sorokin, A.; and Yilmaz, E., eds., *Proceedings of the ACM SIGIR 2011 Workshop on Crowdsourcing for Information Retrieval (CIR 2011)*, 48–55.