Acquiring High Quality Non-Expert Knowledge from On-demand Workforce

Donghui Feng **Sveva Besana** **Remi Zajac**
AT&T Interactive Research
Glendale, CA, 91203
{dfeng, sbesana, rzajac}@attinteractive.com

Abstract

Being expensive and time consuming, human knowledge acquisition has consistently been a major bottleneck for solving real problems. In this paper, we present a practical framework for acquiring high quality non-expert knowledge from on-demand workforce using Amazon Mechanical Turk (MTurk). We show how to apply this framework to collect large-scale human knowledge on AOL query classification in a fast and efficient fashion. Based on extensive experiments and analysis, we demonstrate how to detect low-quality labels from massive data sets and their impact on collecting high-quality knowledge. Our experimental findings also provide insight into the best practices on balancing cost and data quality for using MTurk.

1 Introduction

Human knowledge acquisition is critical for training intelligent systems to solve real problems, both for industry applications and academic research. For example, many machine learning and natural language processing tasks require non-trivial human labeled data for supervised learning-based approaches. Traditionally this has been collected from domain experts, which we refer to as expert knowledge.

However, acquiring in-house expert knowledge is usually very expensive, time consuming, and has consistently been a major bottleneck for many research problems. For example, tremendous efforts have been put into creating TREC corpora (Voorhees, 2003).

As a result, several research projects sponsored by NSF and DARPA aim to construct valuable data resources via human labeling; these are exemplified by PennTree Bank (Marcus *et al.*, 1993), FrameNet (Baker *et al.,* 1998), and OntoNotes (Hovy *et al.*, 2006).

In addition, there are projects such as Open Mind Common Sense (OMCS) (Stork, 1999; Singh *et al.*, 2002), ISI LEARNER (Chklovski, 2003), and the Fact Entry Tool by Cycorp (Belasco *et al.*, 2002) where knowledge is gathered from volunteers.

One interesting approach followed by von Ahn and Dabbish (2004), applied to image labeling on the Web, is to collect valuable input from entertained labelers. Turning label acquisition into a computer game addresses tediousness, which is one of the main reasons that it is hard to gather large quantities of data from volunteers.

More recently researchers have begun to explore approaches for acquiring human knowledge from an on-demand workforce such as Amazon Mechanical Turk[1]. MTurk is a marketplace for jobs that require human intelligence.

There has been an increase in demand for crowdsourcing prompted by both the academic community and industry needs. For instance, Microsoft/Powerset uses MTurk for search relevance evaluation and other companies are leveraging turkers to clean their data sources.

However, while it is cheap and fast to obtain large-scale non-expert labels using MTurk, it is still unclear how to leverage its capability more efficiently and economically to obtain sufficient useful and high-quality data for solving real problems.

In this paper, we present a practical framework for acquiring high quality non-expert knowledge using MTurk. As a case study we have applied this framework to obtain human classifications on AOL queries (determining whether a query might be a local search or not). Based on extensive experiments and analysis, we show how to detect bad labelers/labels from massive data sets and how to build high-quality labeling sets. Our experiments also provide in-

[1] Amazon Mechanical Turk: http://www.mturk.com/

Proceedings of the 2009 Workshop on the People's Web Meets NLP, ACL-IJCNLP 2009, pages 51–56,
Suntec, Singapore, 7 August 2009. ©2009 ACL and AFNLP

sight into the best practices for balancing cost and data quality when using MTurk.

The remainder of this paper is organized as follows: In Section 2, we review related work using MTurk. We describe our methodology in Section 3 and in Section 4 we present our experimental results and further analysis. In Section 5 we draw conclusions and discuss our plans for future work.

2 Related Work

It is either infeasible or very time and cost consuming to acquire in-house expert human knowledge. To obtain valuable human knowledge (*e.g.*, in the format of labeled data), many research projects in the natural language community have been funded to create large-scale corpora and knowledge bases, such as PenTreeBank (Marcus *et al.*, 1993), FrameNet (Baker *et al.,* 1998), PropBank (Palmer *et al.*, 2005), and OntoNotes (Hovy *et al.*, 2006).

MTurk has been attracting much attention within several research areas since its release. Su *et al.* (2007) use MTurk to collect large-scale review data. Kaisser and Lowe (2008) report their work on generating research collections of question-answering pairs using MTurk. Sorokin and Forsyth (2008) outsource image-labeling tasks to MTurk. Kittur *et al.* (2008) use MTurk as the paradigm for user studies. In the natural language community Snow *et al.* (2008) report their work on collecting linguistic annotation for a variety of natural language tasks including word sense disambiguation, word similarity, and textual entailment recognition.

However, most of the reported work focuses on how to apply data collected from MTurk to their applications. In our work, we concentrate on presenting a practical framework for using MTurk by separating the process into a validation phase and a large-scale submission phase.

By analyzing workers' behavior and their data quality, we investigate how to detect low-quality labels and their impact on collected human knowledge; in addition, during the validation step we study how to best use MTurk to balance payments and data quality. Although our work is based on the submission of a classification task, the framework and approaches can be adapted for other types of tasks.

In the next section, we will discuss in more detail our practical framework for using MTurk.

3 Methodology

3.1 Amazon Mechanical Turk

Amazon launched their MTurk service in 2005. This service was initially used for internal projects and eventually fulfilled the demand for using human intelligence to perform various tasks that computers currently cannot do or do very well.

MTurk users naturally fall into two roles: a requester and a turker. As a requester, you can define your Human Intelligent Tasks (HITs), design suitable templates, and submit your tasks to be completed by turkers. A turker may choose from HITs that she is eligible to work on and get paid after the requester approves her work. The work presented in this paper is mostly from the perspective of a requester.

3.2 Key Issues

While it is quite easy to start using MTurk, requesters have to confront the following: how can we obtain sufficient useful and high-quality data for solving real problems efficiently and economically?

In practice, there are three key issues to consider when answering this question.

Key Issues	Description
Data Quality	Is the labeled data good enough for practical use?
Cost	What is the sweet spot for payment?
Scale	How efficiently can MTurk be used when handling large-scale data sets? Can the submitted job be done in a timely manner?

Table 1. Key issues for using MTurk.

Requesters want to obtain high-quality data on a large scale without overpaying turkers. Our proposed framework will address these key issues.

3.3 Approaches

Since not all tasks collecting non-expert knowledge share the same characteristics and suitable applications, there is not a one-size-fits-all solution as the best practice when using MTurk.

In our approach, we divide the process into two phases:

- Validation Phase.
- Large-scale Submission Phase.

The first phase gives us information used to determine if MTurk is a valid approach for a given problem and what the optimal parameters for high quality and a short turn-around time are.

We have to determine the right cost for the task and the optimal number of labels. We empirically determine these parameters with an MTurk submission using a small amount of data. These optimal parameters are then used for the large-scale submission phase.

Most data labeling tasks require subjective judgments. One cannot expect labeling results from different labelers to always be the same. The degree of agreement among turkers varies depending on the complexity and ambiguity of individual tasks. Typically we need to obtain multiple labels for each HIT by assigning multiple turkers to the same task.

Researchers mainly use the following two quantitative measures to assess inter-agreement: observed agreement and kappa statistics.

$P(A)$ is the observed agreement among annotators. It represents the portion where annotators produce identical labels. This is very natural and straightforward. However, people argue this may not necessarily reflect the exact degree of agreement due to chance agreement.

$P(E)$ is the hypothetical probability of chance agreement. In other words, $P(E)$ represents the degree of agreement if both annotators conduct annotations randomly (according to their own prior probability).

We can also use the kappa coefficient as a quantitative measure of inter-person agreement. It is a commonly used measure to remove the effect of chance agreement. It was first introduced in statistics (Cohen, 1960) and has been widely used in the language technology community, especially for corpus-driven approaches (Carletta, 1996; Krippendorf, 1980). Kappa is defined with the following equation:

$$kappa = \frac{P(A) - P(E)}{1 - P(E)}$$

Generally it is viewed more robust than observed agreement $P(A)$ because it removes chance agreement $P(E)$.

```
DetectOutlier( $P$ )
  for each turker $p \in P$
    collect the label set $L$ from $p$
    for each label $l \in L$
      /* compared with others' majority voting */
      compute its agreement with others
    compute $P(A)_p$ (or $kappa_p$)
  analyze the distribution of $P(A)$
  return outlier turkers
```

Figure 1. Outlier detection algorithm.

We use these measures to automatically detect outlier turkers producing low-quality results. Figure 1 shows our algorithm for automatically detecting outlier turkers.

4 Experiments

Based on our proposed framework and approaches, as a case study we conducted experiments on a classification task using MTurk.

The classification task requires the turker to determine whether a web query is a local search or not. For example, is the user typing this query looking for a local business or not? The labeled data set can be used to train a query classifier for a web search system.

This capability will make search systems able to distinguish local search queries from other types of queries and to apply specific search algorithms and data resources to better serve users' information needs.

For example, if a person types "largest biomed company in San Diego" and the web search systems can recognize this query as a local search query, it will apply local search algorithms on listing data instead of or as well as generating a general web search request.

4.1 Validation Phase

We downloaded the publicly available AOL query log[2] and used this as our corpus. We first scanned all queries with geographic locations (including states, cities, and neighborhoods) and then randomly selected a set of queries for our experiments.

For the validation phase, 700 queries were first labeled in-house by domain experts and we refer to this set as expert labels. To obtain the optimal parameters including the desired number of labels and payment price, we designed our HITs and experiments in the following way:

We put 10 queries into one HIT, requested 15 labels for each query/HIT, and varied payment for each HIT in four separate runs. Our payments include \$0.01, \$0.02, \$0.05, and \$0.10 per HIT. The goal is to have HITs completed in a timely fashion and have them yield high-quality data.

We submitted our HITs to MTurk in four different runs with the following prices: \$0.01, \$0.02, \$0.03, and \$0.10. According to our predefined evaluation measures and our outlier detection algorithm, we investigated how to obtain the optimal parameters. Figure 2. shows the task completion statistics for the four different runs.

[2] AOL Log Data: http://www.gregsadetsky.com/aol-data/

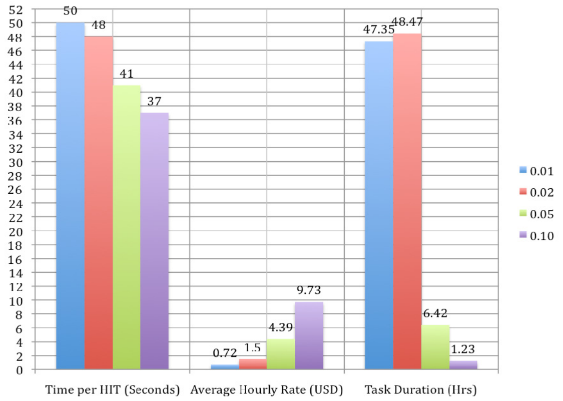

Figure 2. Task completion statistics.

As shown in Figure 2, with the increase of payments, the average hourly rate increases from $0.72 to $9.73 and the total turn-around time dramatically decreases from more than 47 hours to about 1.5 hours. In the meantime, people tend to become more focused on the tasks and spend less time per HIT.

In addition, as we increase payment, more people tend to stay with the task and take it more seriously as evidenced by the quality of the labeled data. This results in fewer numbers of workers overall as well as fewer outliers as shown in Figure 3.

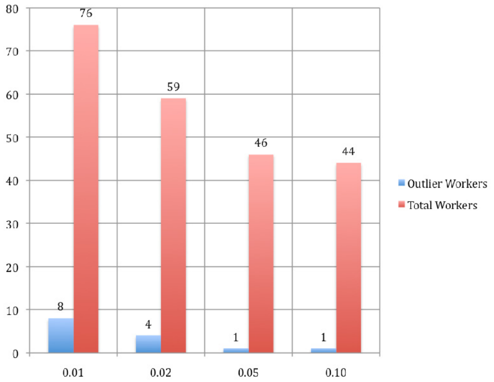

Figure 3. Total number of workers and outliers.

We investigate two types of agreements, inter-turker agreement and agreement between turkers and our in-house experts. For inter non-expert agreements, we compute each turker's agreement with all others' majority voting results.

Payment (USD)	0.01	0.02	0.05	0.10
Median of inter-turker agreement	0.8074	0.8583	0.9346	0.9028

Table 2. Median of inter-turker agreements.

As in our outlier detection algorithm, we analyzed the distribution of inter-turker agreements. Table 2 shows the median values of inter-turker agreement as we vary the payment prices. The median value keeps on increasing when the price increases from $0.01, to $0.02 and $0.05. However, it drops as the price increases from $0.05 to $0.10. This implies that turkers do not necessarily improve their work quality as they get paid more. One of the possible explanations for this phenomenon is that when the reward is high people tend to work towards completing the task as fast as possible instead of focusing on submitting high-quality data. This trend may be intrinsic to the task we have submitted and further experiments will show if this turker behavior is task-independent.

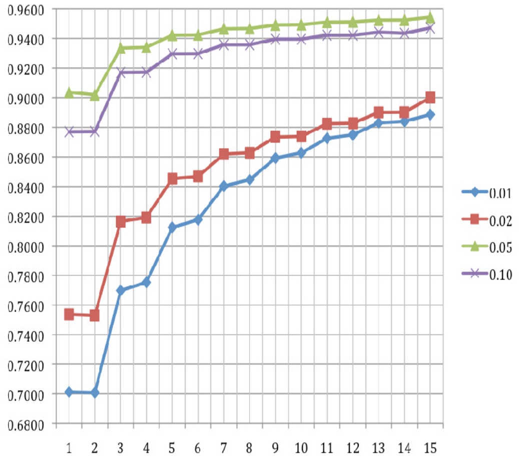

Figure 4. Agreement with experts.

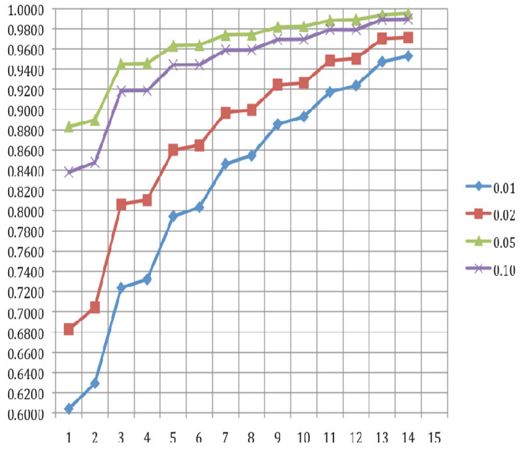

Figure 5. Inter non-expert agreement.

We also analyzed agreement between non-experts and experts. Figure 4 depicts the trend of the agreement scores with the increase of number of labels and payments. For example, given seven labels per query, in the experiment with the $0.05 payment, the majority voting of non-expert labels has an agreement of 0.9465 with expert labeling. As explained earlier we do not necessarily obtain the best data quality/agreement with the $0.10 payment. Instead, we get the highest agreement with the $0.05 payment. We have determined this rate to be the sweet spot in terms of cost. Also, seven labels per query produce a very high agreement with no further significant improvement when we increase the number of labels.

For inter non-expert agreements, we found similar trends in terms of different payments and number of labels as shown in Figure 5.

As mentioned above, our algorithm is able to detect turkers producing low-quality data. One natural question is: how will their labels affect the overall data quality?

We studied this problem in two different ways. We evaluated the data quality by removing either all polluted queries or only outliers' labels. Here polluted queries refer to those queries receiving at least one label from outliers. By removing polluted queries, we only investigate the clean data set without any outlier labels. The other alternative is to only remove outliers' labels for specific queries but others' labels for those queries will be kept. Both the agreement between experts and non-experts and inter-non-experts agreement show similar trends: data quality without outliers' labels is slightly better since there is less noise. However, as outliers' labels may span a large number of queries, it may not be feasible to remove all polluted queries. For example, in one of our experiments, outliers' labels pollute more than half of all the records. We cannot simply remove all the queries with outliers' labels due to consideration of cost.

On the other hand, the effect of outliers' labels is not that significant if a certain number of requested labels per query are collected. As shown in Figure 6, noisy data from outliers can be overridden by assigning more labelers.

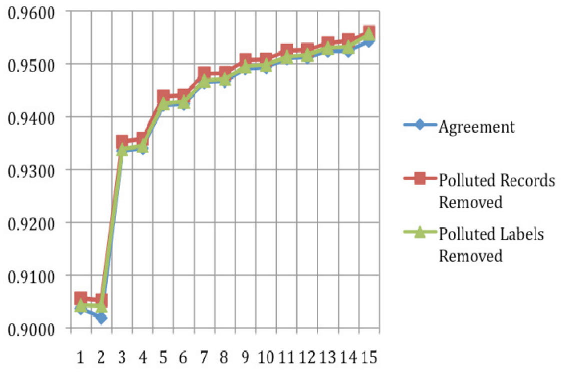

Figure 6. Agreement with Experts (removing outliers' labels (payment = $0.05)).

From the validation phase of the query classification task, we determine that the optimal parameters are paying $0.05 per HIT and requesting seven labels per query. Given this number of labels, the effect of outliers' labels can be overridden for the final result.

4.2 Large-scale Submission Phase

Having obtained the optimal parameters from the validation phase, we are then ready to make a large-scale submission.

For this phase, we paid $0.05 per HIT and requested seven labels per query/HIT. Following similar filtering and sampling approaches as in the validation phase, we selected 22.5k queries from the AOL search log. Table 3 shows the detected outliers for this large-scale submission.

Total Number of Turkers	228
Number of Outlier Turkers	23
Outlier Ratio	10.09%

Table 3. Number of turkers and outliers.

Based on the distribution of inter-turker agreement, any turkers with agreement less than 0.6501 are recognized as outliers. For a total number of 15,750 HITs, 228 turkers contributed to the labeling effort and 10.09% of them were recognized as outliers.

Table 4 shows the number of labels from the outliers and the approval ratio of collected data. About 10.08% of labels are from outlier turkers and rejected.

Total Number of Labels	157,500
Number of Outlier Labels	15,870
Approval Ratio	89.92%

Table 4. Total number of labels.

We have experimented using MTurk for a web query classification task. With learned optimal parameters from the validation phase, we collected large-scale high-quality non-expert labels in a fast and economical way. These data will be used to train query classifiers to enhance web search systems handling local search queries.

5 Conclusions and Future Work

In this paper, we presented a practical framework for acquiring high quality non-expert knowledge from an on-demand and scalable workforce. Using Amazon Mechanical Turk, we collected large-scale human classification knowledge on web search queries.

To learn the best practices when using MTurk, we presented a two-phase approach, a validation phase and a large-scale submission phase. We conducted extensive experiments to obtain the optimal parameters on the number of labelers and payments in the validation phase. We also presented an algorithm to automatically detect outlier turkers based on the agreement analysis and investigated the effect of removing an inaccurately labeled set.

Acquiring high-quality human knowledge will remain a major concern and a bottleneck for industry applications and academic problems. Unlike traditional ways of collecting in-house human knowledge, MTurk provides an alternative way to acquire non-expert knowledge. As shown in our experiments, given appropriate quality control, we have been able to acquire high-quality data in a very fast and efficient way. We believe MTurk will attract more attention and usage in broader areas.

In the future, we are planning to investigate how this framework can be applied to different types of human knowledge acquisition tasks and how to leverage large-scale labeled data sets for solving natural language processing problems.

References

Baker, C.F., Fillmore, C.J., and Lowe, J.B. 1998. The Berkeley FrameNet Project. In *Proc. of COLING-ACL-1998.*

Belasco, A., Curtis, J., Kahlert, R., Klein, C., Mayans, C., and Reagan, P. 2002. Representing Knowledge Gaps Effectively. In *Practical Aspects of Knowledge Management*, (*PAKM*).

Carletta, J. 1996. Assessing agreement on classification tasks: the kappa statistic. *Computational Linguistics*. 22(2):249–254.

Chklovski, T. 2003. LEARNER: A System for Acquiring Commonsense Knowledge by Analogy. In *Proc. of Second International Conference on Knowledge Capture (KCAP 2003).*

Cohen, J. 1960. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement.* Vol.20, No.1, pp.37-46.

Colowick, S.M. and Pool, J. 2007. Disambiguating for the web: a test of two methods. In *Proc. of the 4th international Conference on Knowledge Capture (K-CAP 2007).*

Hovy, E., Marcus, M., Palmer, M., Ramshaw, L., and Weischedel, R. 2006. OntoNotes: The 90% Solution. In *Proc. of HLT-NAACL-2006.*

Kaisser, M. and Lowe, J.B. 2008. Creating a Research Collection of Question Answer Sentence Pairs with Amazon's Mechanical Turk. In *Proc. of the Fifth International Conference on Language Resources and Evaluation (LREC-2008).*

Kittur, A., Chi, E. H., and Suh, B. 2008. Crowdsourcing user studies with Mechanical Turk. In *Proc. of the 26th Annual ACM Conference on Human Factors in Computing Systems (CHI-2008).*

Krippendorf, K. 1980. *Content Analysis: An introduction to its methodology*. Sage Publications.

Marcus, M., Marcinkiewicz, M.A., and Santorini, B. 1993. Building a large annotated corpus of English: the Penn Treebank. *Computational Linguistics.* 19:2, June 1993.

Nakov, P. 2008. Paraphrasing Verbs for Noun Compound Interpretation. In *Proc. of the Workshop on Multiword Expressions (MWE-2008).*

Palmer, M., Gildea, D., and Kingsbury, P. 2005. The Proposition Bank: A Corpus Annotated with Semantic Roles. *Computational Linguistics.* 31:1.

Sheng, V.S., Provost, F., and Ipeirotis, P.G. 2008. Get another label? improving data quality and data mining using multiple, noisy labelers. In *Proc. of the 14th ACM SIGKDD international Conference on Knowledge Discovery and Data Mining (KDD-2008).*

Singh, P., Lin, T., Mueller, E., Lim, G., Perkins, T., and Zhu, W. 2002. Open Mind Common Sense: Knowledge acquisition from the general public. In Meersman, R. and Tari, Z. (Eds.), LNCS: Vol. 2519. *On the Move to Meaningful Internet Systems: DOA/CoopIS/ODBASE* (pp. 1223-1237). Springer-Verlag.

Snow, R., O'Connor, B., Jurafsky, D., and Ng, A.Y. 2008. Cheap and Fast – But is it Good? Evaluating Non-Expert Annotations for Natural Language Tasks . In *Proc. of EMNLP-2008.*

Sorokin, A. and Forsyth, D. 2008. Utility data annotation with Amazon Mechanical Turk. In *Proc. of the First IEEE Workshop on Internet Vision at CVPR-2008.*

Stork, D.G. 1999. The Open Mind Initiative. *IEEE Expert Systems and Their Applications.* pp. 16-20, May/June 1999.

Su, Q., Pavlov, D., Chow, J., and Baker, W.C. 2007. Internet-scale collection of human-reviewed data. In *Proc. of the 16th international Conference on World Wide Web (WWW-2007).*

Von Ahn, L. and Dabbish, L. 2004. Labeling Images with a Computer Game. In *Proc. of ACM Conference on Human Factors in Computing Systmes (CHI)*. pp. 319-326.

Voorhees, E.M. 2003. Overview of TREC 2003. In *Proc. of TREC-2003.*