Can the Crowd be Controlled?: A Case Study on Crowd Sourcing and Automatic Validation of Completed Tasks based on User Modeling

Balamurali A R

Samsung Advanced Institute of Technology (SAIT) India
balamurali.r@samsung.com

Abstract

Annotation is an essential step in the development cycle of many Natural Language Processing (NLP) systems. Lately, crowdsourcing has been employed to facilitate large scale annotation at a reduced cost. Unfortunately, verifying the quality of the submitted annotations is a daunting task. Existing approaches address this problem either through sampling or redundancy. However, these approaches do have a cost associated with it. Based on the observation that a crowdsourcing worker returns to do a task that he has done previously, a novel framework for automatic validation of crowd-sourced task is proposed in this paper. A case study based on sentiment analysis is presented to elucidate the framework and its feasibility. The result suggests that validation of the crowd-sourced task can be automated to a certain extent.

Keywords: Crowdsourcing, Evaluation, User-modelling

Annotation is an unavoidable task for developing NLP systems. Large scale annotation projects such as (Palmer et al., 2005), (Baker et al., 1998), (Marcus et al., 1993) have shown that such tasks can initiate new research frontiers. However, annotation endeavors are extremely expensive in terms of the effort and the money spent. To mitigate the associated cost and to expedite the annotation process, cheap and large scale annotation tasks using non-expert annotators through crowd sourcing platforms are leveraged.

Crowd sourcing is the strategy that combines effort of the public to solve one problem or produce one particular thing (Snow et al., 2008; Callison-Burch, 2009; Bloodgood and Callison-Burch, 2010). Underlying assumption followed here is that workers need not be an expert to do the task provided they are properly guided. Depending on the task, the motivation for the work can be different (Wang et al., 2010). For instance, fun factor (von Ahn and Dabbish, 2008; Vickrey et al.,) can be the motivating factor for game based crowd sourcing whereas for websites like freelancer[1] or Mturk[2] it can be the profit associated that she recieves upon the task completion (Callison-Burch, 2009; Snow et al., 2008). In other instances, it can be a sense of altruism or even fame (Forte and Bruckman, 2005).

According to (Wang et al., 2010; Qui,) the other parameters which define a crowd sourced task, apart from the ways to motivate for doing the task are *annotation quality* (Sheng et al., 2008; Feng et al., 2009), *set up effort*(Zesch et al., 2007) and *human participation* (Bloodgood and Callison-Burch, 2010; Kunath and Weinberger, 2010). The focus of this paper is related to the assessment of the annotation quality. More specifically,

> *Can one model the users involved in crowd sourcing so that we can automatically evaluate their task and reward them based on the correctness of result?*

A sentiment analysis based case study is presented to answer the above question. In this study, a crowd-sourced task is floated to annotate short message texts with sentiment labels (positive/negative/neutral). To validate the annotation, different review strategies are explored. These strategies aim at understanding some unanswered questions related to large scale annotation and crowdsourcing like *whether attractive financial benefits expedite the completion of the crowd sourced task?, Does this affect the annotation quality?, Does the task easiness expedite the completion of the crowd sourced task?*

The annotated data along with the details of the user who were involved in it are obtained through this crowd sourcing endeavor. Thereafter a system is developed for automatic validation of annotation submitted by the crowd. The intuition for this is *the fact that same set of user (workers) comes back to do similar kind of task*. Different features pertaining to the user and the task complexity is extracted from the previously completed annotation tasks. Extracted features are then used to create models which can tell whether a submitted annotation by the user is valid or not. The system was able to detect valid annotation with an average accuracy of 95%. The contributions of this paper are two folds:

1. We present a framework for automatic verifying a crowd sourced task. This can save time and effort spend for validating the submitted task. Moreover, using this framework, a set of reliable worker force can selected a priori for a future task of similar nature.

2. Our results suggest that making the task easier can expedite the task completion rate when compared to increasing the monetary incentive associated with task.

The rest of the paper is organized as follows: section 1. introduces the case study done for setting up the framework mentioned. Crowdsourcing and ways to ensure annotation quality is explained in section 2.. The dataset used for the

[1] www.freelancer.com
[2] www.mturk.com/mturk/

case study is described in section 3.. The framework for automatic validation of crowd-sourced task is explain in section 5. which is based on the observation given in section 4.. Results and its discussions are presented in section 6.. Section 8. concludes the paper and points to some future research directions.

1. Our Case Study: Crowd Sourcing for Sentiment Analysis

Sentiment analysis (SA) deals with automatically tagging text as positive, negative or neutral with respect to a topic from the perspective of the speaker/writer (Pang and Lee, 2008). It is popularily referred as sentiment classification . Thus, a sentiment classifier tags the sentence '*The swimming pool is definitely worth a visit- serious fun for people of all ages!*' in a travel review as *positive*. On the other hand, a sentence '*There was weird nauseating stench inside the pool.*' is labeled as *negative*. Finally, '*The swimmng pool has two diving boards; one at 8 ft and another at 15 ft.*' is labeled as neutral. For the purpose of this work, we consider output labels as positive and negative according to the definition by (Pang et al., 2002) & (Turney, 2002).

2. Crowdsourcing and Annotation Quality

The Amazon Mechanical Turk (MTurk) is a crowdsourcing internet platform that enables individuals or businesses (known as Requesters) to co-ordinate the use of human intelligence to perform tasks that computers are currently unable to do. The Requesters have to post tasks known as HITs (Human intelligence tasks), such as choosing the right label for a picture. In this, case each HIT contain a set of SMSes to be annotated with sentiment labels. Workers (or called providers or Turkers) can then browse through the floated HITS and complete them for monetary payment fixed by the Requester. This HIT fee includes the fee to be paid for the worker and the Amazons service charge. The reward is paid only after the approval of the HIT by the Requester.

There can be different strategies for approving HITs. Two of the commonly followed strategies are:

1. **Redundancy:** Use of multiple workers for the same HIT and selecting the valid HIT by some voting mechanism. For instance if 4 out of 5 workers tag a SMS as positive then by *majority voting principle*, it is tagged as positive and all the 4 workers are paid.

2. **Sampling:** Insertion of gold samples into each HIT and only upon successfully annotating the gold samples would the annotator be paid. For example, in each HIT contains 5 SMSes to be tagged. The Requestor has added 1 gold sample for validation. The worker will be paid only if the gold sample is correctly tagged.

If the annotation is of a large scale, the first option is very expensive. Further, it may take more time to complete. However, the strategy can be effective if a high quality crowd-sourced data is required. Depending on the number of gold samples inserted, the reviewer can dictate the expected annotation quality.

2.1. Different Strategies for Quality Control

For this case study, the annotation quality is controlled through validation of inserted gold samples. Further, cost of each HIT and number of SMSes to be annotated in a HIT are varied to have an additional control over the annotation quality. By changing these parameters, it is possible to control which worker will take up the annotation task. Another objective for changing these parameters is to see if there is any correlation between them and the time required to complete the task. Table 1 shows different settings followed in this case study to control the annotation quality and the speed of annotation. Each row in the table shows an experimental annotation setup. For instance, under *varying cost of HIT*, the first row shows that there is an experimental setup to annotated 1000 SMS wherein each HIT contains 10 SMS out of which 2 are gold samples and it costs 0.05$ to float this HIT on MTurk.

Varying cost of HIT
10 SMS/HIT floated at a fee of 0.05$ and 2 gold samples
10 SMS/HIT floated at a fee of 0.1$ and 2 gold samples
10 SMS/HIT floated at a fee of 0.2$ and 2 gold samples
Varying number of gold samples per HIT
10 SMS/HIT to be floated at fee of 0.07$ and 2 gold samples
10 SMS/HIT to be floated at fee of 0.07$ and 4 gold samples
10 SMS/HIT to be floated at fee of 0.07$ and 1 gold samples
Varying number of SMS per HIT
5 SMS/HIT to be floated at fee of 0.07$ and 2 gold samples
7 SMS/HIT to be floated at fee of 0.07$ and 2 gold samples
12 SMS/HIT to be floated at fee of 0.07$ and 2 gold samples

Table 1: Different annotation strategies for quality control

Each of these experiments contain 1000 SMSes to be annotated. Each experiment was given a maximum duration of time to complete and were independently floated. If all the HITs under that set are not completed in this time, the experiment is deemed invalid and it was assumed that the parameter being tested in the experiment has no correlation with the task attractiviness. Once submitted by the worker, HIT was validated based on the inserted gold samples.

3. Dataset

For the experiments, SMS corpus[3] by (Chen and Kan, 2013) was used. The corpus contains 41,790 SMSes collected from various sources and through various means. For further information please refer the original paper. 9,000 SMSes were randomly selected to be floated for crowdsourcing experiments. Based on the different strategies mentioned in table 1 experiments were floated on MTurk. Instructions with example annotation were also presented to the workers for their assistance. Upon the completion of the task, the reviewer would be furnished with a HIT level report. This includes information pertaining to the approval

[3] http://wing.comp.nus.edu.sg/SMSCorpus/

Varying cost of HIT				
Task	**Number of HITs**	**Number of workers Attempted the task**	**Average Time per Assignment**	**Annotation Accuracy**
10 SMS/HIT floated at a fee of 0.05$ and 2 gold samples	97 (970 SMS)	3	1 hr 7 min	93.81%
10 SMS/HIT floated at a fee of 0.1$ and 2 gold samples	98 (980 SMS)	2	59 sec	97.95%
10 SMS/HIT floated at a fee of 0.2$ and 2 gold samples	100 (1000 SMS)	5	1 min 33 sec	99%
Varying number of gold samples per HIT				
Task	**Number of HITs**	**Number of workers Attempted the task**	**Average Time per Assignment**	**Annotation Accuracy**
10 SMS/HIT to be floated at fee of 0.07$ and 1 gold samples	100 (1000 SMS)	2	3 min 24 sec	92%
10 SMS/HIT to be floated at fee of 0.07$ and 3 gold samples	100 (1000 SMS)	2	1 min 49 sec	100%
10 SMS/HIT to be floated at fee of 0.07$ and 7 gold samples	75 (750 SMS)	5	5 min 44 sec	100%
Varying number of SMS per HIT				
Task	**Number of HITs**	**Number of workers Attempted the task**	**Average Time per Assignment**	**Annotation Accuracy**
5 SMS/HIT to be floated at fee of 0.07$ and 2 gold samples	150 (750SMS)	3	40 sec	90.00%
7 SMS/HIT to be floated at fee of 0.07$ and 2 gold samples	132 (924SMS)	3	2 min 27 sec	97.72%
12 SMS/HIT to be floated at fee of 0.07$ and 2 gold samples	83 (996SMS)	3	2 min 37 sec	96.38%

Table 2: Annotation statistics for different crowd-sourced experiment sets

rate, time spend on each HIT *etc*. The requestor approves or rejects the HIT submitted by the worker based on his validation of the gold samples.

3.1. Annotation Statistics

A major part of the selected SMS for crowd sourcing were confirmations or acknowledgments like“ok” ,“yes”,“sure” *etc*. The duplicates of the same were removed before floating the experiments. Table 2 shows the annotation result statistics for different experimental annotation setups. It includes the details about the experiment, the actual number of HITs floated per experiment after removing the duplicate SMSes, the number of workers who attempted the task, the average time per assignment and the annotation accuracy based on the gold sample verification process.

For experiments pertaining to varying cost of the HIT, it is seen that increasing the HIT reward attracts more workers. The task wherein each HIT costs 0.05$ for 10 SMS took more average time per assignment. This was due to the fact that this was the first batch floated and the workers were not ready to accept the job as they were not sure about the approval rate of the requestor. Usually it is seen that if the approval rate of the reviewer is high then workers have high confidence in her and will readily accept the floated task by her. The details about the evaluation were included in the HIT information page. This was done so as to make the worker aware of the evaluation criteria. For instance, all the information about the number of gold samples inserted in each HIT was mentioned before the acceptance of the task. It is seen that, in general, this does not have an effect on the attractive quotient of floated task. However, the task reward has an implication on the task attractiveness. As the task reward was increased from 0.05$/HIT to 0.2$/HIT the number of workers who attempted increased from 3 to 5.

It was assumed that if the task difficulty was increased (by making the evaluation criteria more stringent); it would adversely affect the task attractiveness. However this belief was thwarted as our experiments show that as the number of gold samples were increased the number of workers who attempted it also increased. This suggests that stringent evaluation norms does not deter the workers from accepting the job which is easy and has a high acceptance rate. Moreover, this increment did not come at the expense of task quality as well.

From the experiments listed in Table 1 and the annotation statistics of Table 2, it can be safely assumed that a good balance of quality annotation and task attractiveness can be attained by making the task simple. For instance, the best average time per assignment was obtained for the experimental setup where each HIT contained just 5 SMSes.

4. An Observation

Figure 1 shows the individual users (workers) who participated in the experiment and the number of HITs they submitted. It was observed that many workers do come back to take up similar jobs that they have attempted in the past. The fact that they return to the similar task floated by the same requestor suggests that they have confidence in the rate at which the reviewer approves the task. This also suggests that these workers are reliable and they do not have any malicious intent.

As the same workers return to attempt the similar task, it seems redundant to insert the gold samples to validate their work. Based on the task difficulty and prior behavioral pat-

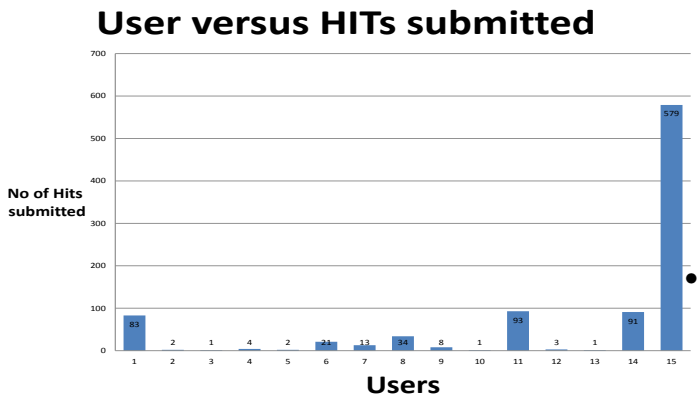

Figure 1: Number of HIT attempted by Workers

terns, the validation part of the crowdsourcing could be automated. In the next section, the framework and feasibility of such a setup is explored.

5. User Modeling for Automatic HIT Validation

Same worker returns to attempt similar task floated by the same requestor. Based on this premise, *is it possible to model users activity for automatic validation of similar tasks he might do in the future*?
To answer this question, it is imperative to know when a worker will make a mistake. It is assumed that worker inherently does not want to invalidate the task. Upon the analysis of the annotated data for all the users who had attempted more than 90 HITs, it was seen that there exists a correlation between the task difficulty and the tendency to commit mistake. Tasks that require automatic validation need to capture these parameters. Thereafter, these parameters are translated into features for developing user models which can predict whether a HIT can be accepted or not.
If such a framework can be developed and its feasibility verified, the evaluation cost associated with the repetitive crowd-sourced tasks can be reduced. Currently there is an overhead of adding gold samples to each HIT for task approval. This cost can be reduced if each user who comes back for the task can be modeled based on his/her prior annotation pattern.

5.1. The framework for automatic validation

Figure 2 shows the general framework for automatic validation of crowd-sourced task. For most of the repeatable crowd-sourced tasks such as the one that is dealt in this paper, all modules, barring two, mentioned in the framework remain same. They are:

- **Task parameters**: These are the parameters which distinguish each instances of the task. For exmaple, in the SMS annotation task, difficulty index pertaining to annotating each SMS is one of the task parameter. User may take different time to annotate each SMS based on the inherent difficulty level. Capturing these parameters correctly is necessary to model the annotation process. The way to define these parameters depends on the task itself. For the SMS annotation scenario, it deals with the syntactic and semantic complexity associated with each textual unit of the annotation. For some other tasks, say for instance, *image tagging*, the task parameter may be how the colour contour changes over the segments, which in turn is some form of syntactic complexity element related to vision.

- **Behavioral parameters**: These are the parameters which underline a users behavioral traits which in turn is depended on the task. This can change from task to task. An example of such parameter is the average time a user spends on each SMS

5.2. Feature Engineering to Model the User:

Based on the task and behavioral parameters that are deemed important, features were extracted from various experiments floated for SMS annotation. MTurk gives the annotation related statistics at the HIT level. Therefore, all features were calculated at the HIT level rather than at the SMS level. However, the features were so engineered that it indirectly captures information at the SMS level.

5.2.1. Features based on Task Parameters:

To capture the difficulty involved in annotating an SMS with polarity labels, lexical, syntactical and semantic complexity was analyzed. To do so, NLPCore[4] was used. Given a sentence for processing, NLPCore gives word lemmas, their parts of speech (POS), whether they are names of companies, people, *etc.*, thereafter normalize dates, times, and numeric quantities, and mark up the structure of sentences in terms of word dependences and phrases, and indicate which noun phrases refer to the same entities. Table 3 shows the features based on the output of NLPcore, for each SMS (aggregated at HIT level).

Average no of words	Average no of Adverbs
Average no of Noun Phrases	Average no of Adjectives
Average no of verb Phrases	Average no of Named Entities
Average no of Nouns	Average no of Sub-sentences
Average no of Verbs	Maximum dependency graph depth of the sentence

Table 3: Features based on Task Parameters

5.2.2. Features based on Behavioral Parameters

To capture the behavioural pattern pertaining to annotation task, temporal features related to the annotation were devised. MTurk give the time spend by the worker on each HIT. This parameter was utilizedd to create most of the behavioural parameters based features. These are shown in Table 4.

[4] http://nlp.stanford.edu/software/corenlp.shtml

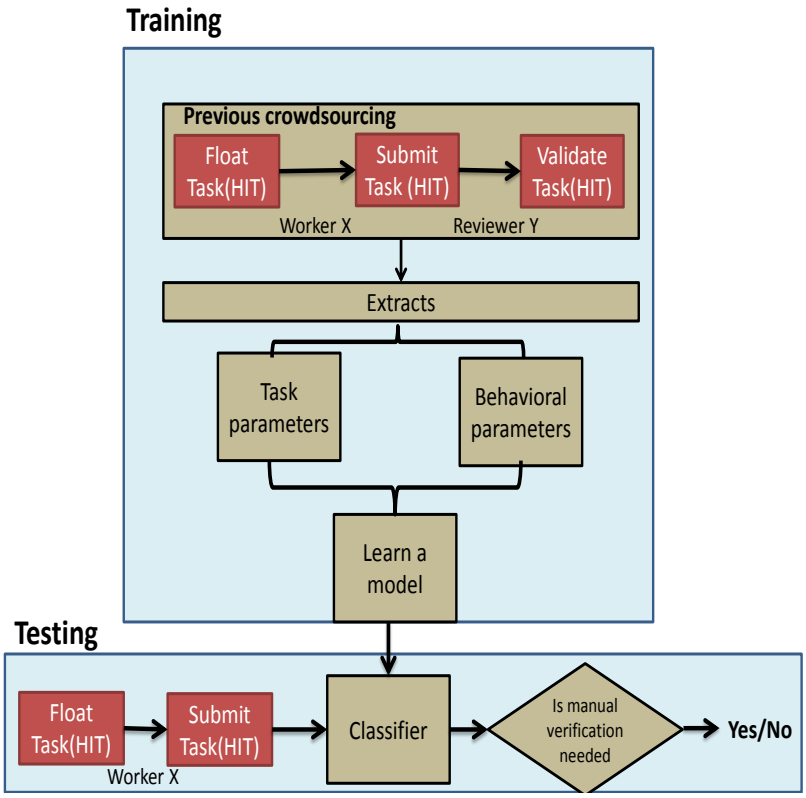

Figure 2: General framework for automatic validation of submitted HIT incase of repeatable crowd-sourced task

Time to complete the HIT
Time to complete the previous HIT
Average time to complete previous 5 HIT
Approval rate of the User in last 7 days
Approval rate of the user in last 15 days
Approval rate of user in last 30 days

Table 4: Features based on Behavioral Parameters

5.3. Experimental Setup

Out of all the workers who participated in the annotation experiments, only those who have submitted more than 90 HITs were selected for automatic validation experiment. Two set of experiments were done. In the first experiment, a single user data was selected and the prediction capability of the model was tested on changing the training and test data. For instance, in one case the training set used was based on annotation samples got from HITs containing 5 SMSes whereas the test set was based on HITs containing 10 SMSes. In the next set of experiment, the entire set of HITs, irrespective of the number SMSes each contain, were used to report cross validation prediction accuracy. For computing the accuracy of the system, it was assumed that HIT can be approved if the gold sample is correctly annotated and the correctness of the annotation at the SMS level was not done.

Whether to accept the annotation or not was modeled as a one-class problem. SVM[5] was used as the learner. The default parameter(ν) was used for the experiments. The prediction accuracy reported for the first set of experiment is on independent training and test set whereas for the second case, a 10 fold cross validation was used to report the results.

[5]LIBSVM by (Chang and Lin, 2011)

6. Result and Discussion

Train\Test	5 SMS/HIT	10 SMS/HIT	12 SMS/HIT
5 SMS/HIT	**90%**	96.70%	92%
10 SMS/HIT	90%	**96.73%**	95%
12 SMS/HIT	91%	96%	**95.65**

Table 5: Automatic validation results for a Worker

	User	User 11	User 14	User 15
Classification accuracy		54.87%	98.88%	96.73%

Table 6: User modelling statistics for different users (all who has done more than 90 hits for the task of annotating 10SMS/HIT)

Table 6. shows the approval accuracy for HIT submitted by a worker (user 15). The table depicts different training and test scenarios based on the different variants of HITs he/she submitted. The framework is able to predict when the HIT should be approved with high accuracy if the train and test distribution is the same. Based on the decision given by the framework, the requester can take the call whether it should be manually verified or not. As the prediction accuracy is high, it also increases the requestors confidence in the annotation quality.

Whenever the test and the train distribution is different, the framework is able to predict annotations which could be approved with high accuracy. This suggests that the number of atomic tasks (number of annotation instances present) incorporated in each HIT has no bearing on the task and user modeling for automatic validation.

Table 6 shows the approval accuracy for HIT submitted by workers who have submitted more than 80 HITs. Approval accuracy obtained for user 11 is low compared to user 14 and 15. The reason for this deviation is the fact that he belongs to the first experiment that was conducted. In the first experiment, as evident from Table 2, there were deviations due to poor credibility of the workers on the reviewers approval rate. This suggests that confidence of the worker on the reviewers approval rate can adversely affect the profiling aspect of the workers for automatic validation of submitted HITS.

7. Related Work

Crowdsourcing is no longer a new term in the domain of Computational Linguistics and Datamining research (Anoop et al., 2013; Snow et al., 2008; Callison-Burch, 2009; Bloodgood and Callison-Burch, 2010). In this paper, focus is given to quality aspect of a crowd-sourced task.

Requesters are prone to quality risks while floating annotation work on crowdsourcing platforms. A way to ensure quality is through redundancy (Ipeirotis et al., 2010; Ambati and Vogel, 2010). However, redundancy can adversely effect the cost of the crowdsourcing task, sometimes making it less profitable than in-house annotation solutions. To avert this, (Dawid AP, 1979) suggested controlling annotation quality through controlling who will annotate. They proposed a solution, based on an expectation maximization algorithm. The EM algorithm contains two steps: (1) estimates the correct answer for each task, using labels assigned by multiple workers, accounting for the quality of each worker; and (2) re-estimates the quality of the workers by comparing the submitted answers to the inferred correct answers. Through series of matrix computation, (Dawid AP, 1979) was able to come up with scalar value as the quality score for each worker. The error rate of each user was used to model the quality associated with the worker. However, the workers could be prone to their bias. (Ipeirotis et al., 2010) presents an algorithm to separate this systemic bias and then model the quality of the worker based on his true error rate. Controlling quality of the annotation through the quality of the worker is considered to be an efficient strategy, even Mturk advocates it. In MTurk, there are two class of worker: 1) master workers, who have a proven track record about the quality of their work. They constitute 2% of the total worker population on Mturk 2) the rest. This is a good strategy, however there still might be more workers in that 80% who will be new and good. Through the framework mentioned in this paper, even a new worker, who is a quality annotator, has got a chance to participate in a new crowdsourcing task.

In this paper, each worker is modeled based on his previous annotation history. The annotation history is validated through sampling (Zaidan and Callison-Burch, 2011). This was done to reduce the cost. Compared to redundancy based annotation quality control, sampling based technique can be done at a lower cost. However, even this can be considered as an additional cost. In crowdsourcing markets, every time a worker is tested, an opportunity to get some work done is lost. This is analogous to the concept of inspection cost in manufacturing process. A number of studies have been carried out on this front (Forte and Bruckman, 2005; Wetherill and WK, 1975). An optimal number of historical data can assure better confidence levels about the quality of the worker. In this work, a fixed number of samples are selected to model the worker.

8. Conclusion and Future Directions

Different strategies for controlling annotation quality are explored in this paper. Validating the submitted task is expensive and time consuming. It is observed that in case of simple annotation tasks, where the tasks are generally repetitive, workers often return to the same reviewer if the approval rate is high. Based on this premise, a framework to validate the tasks submitted by the workers in the crowdsourcing market is proposed. As a case study, an SMS annotation task based on crowdsourcing was performed. Annotated data was validated and collected from various workers. The framework was developed through modeling the task parameters and user parameters associated with the floated tasks. The proposed framework was highly efficient in capturing the true HITs that needs to be accepted.

The framework mentioned in this paper could be used to create pool of high quality annotators. These annotators could in turn be used for future annotation tasks. Even though, high annotation validation accuracy was obtained, it needs further improvement. A transistion based model will be explored to develop more features for classification.

Acknowledgement

We thank Pratibha Moogi from SAIT India for her timely support in preparing this manuscript.

9. References

Vamshi Ambati and Stephan Vogel. 2010. Can crowds build parallel corpora for machine translation systems? In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, CSLDAMT '10, pages 62–65.

K Anoop, Chatterjee Rajen, Roy Shourya, Mishra Abhijit, and Pushpak Bhattacharyya. 2013. Transdoop: A map-reduce based crowdsourced translation for complex domains. In *Proceedings of the ACL 2013 System demo*.

Collin F. Baker, Charles J. Fillmore, and John B. Lowe. 1998. The berkeley framenet project. In *Proceedings of the 17th international conference on Computational linguistics - Volume 1*, COLING '98, pages 86–90.

Michael Bloodgood and Chris Callison-Burch. 2010. Using mechanical turk to build machine translation evaluation sets. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, CSLDAMT '10, pages 208–211.

Chris Callison-Burch. 2009. Fast, cheap, and creative: evaluating translation quality using amazon's mechanical turk. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1 - Volume 1*, EMNLP '09, pages 286–295.

Chih-Chung Chang and Chih-Jen Lin. 2011. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

Tao Chen and Min-Yen Kan. 2013. Creating a live, public short message service corpus: the nus sms corpus. *Language Resources and Evaluation*, 47(2):299–335.

Skene AM Dawid AP. 1979. Maximum likelihood estimation of observer error-rates using the em algorithm.

Donghui Feng, Sveva Besana, and Remi Zajac. 2009. Acquiring high quality non-expert knowledge from on-demand workforce.

Andrea Forte and Amy Bruckman. 2005. Why do People Write for Wikipedia? Incentives to Contribute to Open-Content Publishing.

Panagiotis G. Ipeirotis, Foster Provost, and Jing Wang. 2010. Quality management on amazon mechanical turk. In *Proceedings of the ACM SIGKDD Workshop on Human Computation*, HCOMP '10, pages 64–67.

Stephen A. Kunath and Steven H. Weinberger. 2010. The wisdom of the crowd's ear: speech accent rating and annotation with amazon mechanical turk. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, CSLDAMT '10, pages 168–171.

Mitchell P. Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. 1993. Building a large annotated corpus of english: the penn treebank. *Comput. Linguist.*, 19(2):313–330, June.

Martha Palmer, Daniel Gildea, and Paul Kingsbury. 2005. The proposition bank: An annotated corpus of semantic roles. *Comput. Linguist.*, 31(1):71–106, March.

Bo Pang and Lillian Lee. 2008. Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, pages pp. 1–135.

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up? sentiment classification using machine learning techniques. volume 10, pages 79–86. Association for Computational Linguistics.

A Taxonomy of Distributed Human Computation.

Victor S. Sheng, Foster Provost, and Panagiotis G. Ipeirotis. 2008. Get another label? improving data quality and data mining using multiple, noisy labelers. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '08, pages 614–622.

Rion Snow, Brendan O'Connor, Daniel Jurafsky, and Andrew Y. Ng. 2008. Cheap and fast—but is it good?: evaluating non-expert annotations for natural language tasks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '08, pages 254–263.

Peter Turney. 2002. Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews. In *Proceedings of ACL-02*, pages 417–424, Philadelphia, US.

David Vickrey, Aaron Bronzan, William Choi, Aman Kumar, Jason Turner-maier, Arthur Wang, and Daphne Koller. Online word games for semantic data collection.

Luis von Ahn and Laura Dabbish. 2008. Designing games with a purpose. *Commun. ACM*, 51(8):58–67, August.

Aobo Wang, Cong Duy Vu Hoang, and Min-Yen Kan. 2010. Perspectives on crowdsourcing annotations for natural language processing.

GB Wetherill and Chiu WK. 1975. A review of acceptance sampling schemes with emphasis on the economic aspect.

Omar F. Zaidan and Chris Callison-Burch. 2011. Crowdsourcing translation: professional quality from non-professionals. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 1220–1229.

Torsten Zesch, Iryna Gurevych, and Max. 2007. Analyzing and Accessing Wikipedia as a Lexical Semantic Resource. In *Biannual Conference of the Society for Computational Linguistics and Language Technology*.