To appear in EMNLP-2008. Accepted draft as of Aug. 2008. This is **NOT THE FINAL VERSION.**

Cheap and Fast — But is it Good? Evaluating Non-Expert Annotations for Natural Language Tasks

Rion Snow
Computer Science Dept.
Stanford University
Stanford, CA 94305
rion@cs.stanford.edu

Brendan O'Connor
Dolores Labs, Inc.
832 Capp St.
San Francisco, CA 94110
brendano@doloreslabs.com

Daniel Jurafsky
Linguistics Dept.
Stanford University
Stanford, CA 94305
jurafsky@stanford.edu

Andrew Y. Ng
Computer Science Dept.
Stanford University
Stanford, CA 94305
ang@cs.stanford.edu

Abstract

Human linguistic annotation is crucial for many natural language processing tasks but can be expensive and time-consuming. We explore the use of Amazon's Mechanical Turk system, a significantly cheaper and faster method for collecting annotations from a broad base of paid non-expert contributors over the Web. We investigate five tasks: recognizing textual entailment, affect recognition, word similarity, event temporal ordering, and word sense disambiguation. For all five, we show high agreement between Mechanical Turk non-expert annotations and existing gold standard labels provided by expert labelers. For the task of affect recognition, we also show that using non-expert labels for training machine learning algorithms can be as effective as using gold standard annotations from experts. We propose a technique for bias correction that significantly improves annotation quality on two tasks. We conclude that many large labeling tasks can be effectively designed and carried out in this method at a fraction of the usual expense.

1 Introduction

Large scale annotation projects such as TreeBank (Marcus et al., 1993), PropBank (Palmer et al., 2005), TimeBank (Pustejovsky et al., 2003), FrameNet (Baker et al., 1998), SemCor (Miller et al., 1993), and others play an important role in natural language processing, encouraging the development of novel ideas, tasks, and algorithms. The construction of these datasets, however, is extremely expensive in annotator-hours as well as money. Since the performance of many natural language processing tasks is limited by the amount and quality of data available to them (Banko and Brill, 2001), a promising alternative, at least for some tasks, is collecting annotations from non-expert volunteers.

In this work we explore such a system, Amazon Mechanical Turk[1] (AMT), to study whether non-expert volunteers on the web can provide reliable natural language annotations. We chose five natural language understanding tasks that we felt would be sufficiently natural and learnable for non-experts, and for which we had gold standard labels from expert labelers, as well as (in some cases) human labeler agreement information. The tasks are: affect recognition, word similarity, recognizing textual entailment, event temporal ordering, and word sense disambiguation. For each task, we used AMT to annotate data and measured the quality of the annotations by comparing them with the gold standard (expert) labels on the same data, and by using the AMT annotations to train machine learning classifiers.

In the next sections of the paper we introduce the five tasks and the evaluation metrics, and offer methodological insights, including a new technique for bias correction that improves annotation quality.

2 Related Work

The idea of collecting annotations from volunteer contributors has been used for a variety of tasks. Luis von Ahn pioneered the collection of data via

[1] Amazon Mechanical Turk may be found online at http://mturk.com.

online annotation tasks in the form of games, including the ESPGame for labeling images (von Ahn and Dabbish, 2004) and Verbosity for annotating word relations (von Ahn et al., 2006). The Open Mind Initiative (Stork, 1999) has taken a similar approach, attempting to make such tasks as annotating word sense (Chklovski and Mihalcea, 2002) and commonsense word relations (Singh, 2002) sufficiently "easy and fun" to entice users into freely labeling data.

There have been an increasing number of experiments using Mechanical Turk for annotation. In (Su et al., 2007) workers provided annotations for the tasks of hotel name entity resolution and attribute extraction of age, product brand, and product model, and were found to have high accuracy compared to gold-standard labels. In (Nakov, 2008) workers generated paraphrases of 250 noun-noun compounds which were then used as the gold standard dataset for evaluating an automatic method of noun compound paraphrasing (though no external gold standard was compared). Kaisser and Lowe (2008) use AMT to help build a dataset for question answering, annotating the answers to 8107 questions with the *sentence* containing the answer. Kaisser et al. (2008) examines the task of customizing the summary length of QA output; non-experts from AMT chose a summary length that suited their information needs for varying query types. Zaenen (2008) studied the agreement of annotators on the problem of recognizing textual entailment (a similar task and dataset is explained in more detail in Section 4).

3 Task Design

In this section we describe Amazon Mechanical Turk and the general design of our experiments.

3.1 Amazon Mechanical Turk

We employ the Amazon Mechanical Turk system in order to elicit annotations from non-expert labelers. AMT is an online labor market where workers are paid small amounts of money to complete small tasks. The design of the system is as follows: one is required to have an Amazon account to either submit tasks for annotations or to annotate submitted tasks. These Amazon accounts are anonymous, but are referenced by a unique Amazon ID. A *Requester* can create a *group* of *Human Intelligence Tasks* (or *HIT*s), each of which is a form composed of an arbitrary number of questions. The user requesting annotations for the group of HITs can specify the number of unique annotations per HIT they are willing to pay for, as well as the reward payment for each individual HIT. While this does not guarantee that unique people will annotate the task (since a single person could conceivably annotate tasks using multiple accounts, in violation of the user agreement), this does guarantee that annotations will be collected from unique accounts. AMT also allows a requester to restrict which workers are allowed to annotate a task by requiring that all workers have a particular set of qualifications, such as sufficient accuracy on a small test set or a minimum percentage of previously accepted submissions. Annotators (variously referred to as *Workers* or *Turkers*) may then annotate the tasks of their choosing. Finally, after each HIT has been annotated, the Requester has the option of approving the work and optionally giving a bonus to individual workers. There is a two-way communication channel between the task designer and the workers mediated by Amazon, and Amazon handles all financial transactions.

3.2 Task Design

In general we follow a few simple design principles: we attempt to keep our task descriptions as succinct as possible, and we attempt to give demonstrative examples for each class wherever possible. We do not include our task instructions here for lack of space; we have published our full experimental design and the data we have collected online[2]. We have restricted our study to tasks where we require only a multiple-choice response or numeric input within a fixed range. For every task we collect ten independent annotations for each unique item; this redundancy allows us to perform an in-depth study of how data quality improves with the number of independent annotations.

4 Annotation Tasks

We analyze the quality of non-expert annotations on five tasks: affect recognition, word similarity, recognizing textual entailment, temporal event recognition, and word sense disambiguation. In this section

[2]All tasks and collected data are available at `http://ai.stanford.edu/~rion/annotations/`.

we define each annotation task and the parameters of the annotations we request using AMT. Additionally we give an initial analysis of the task results, and summarize the cost of the experiments.

4.1 Affective Text Analysis

This experiment is based on the affective text annotation task proposed in Strapparava and Mihalcea (2007), wherein each annotator is presented with a list of short headlines, and is asked to give numeric judgments in the interval [0,100] rating the headline for six emotions: anger, disgust, fear, joy, sadness, and surprise, and a single numeric rating in the interval [-100,100] to denote the overall positive or negative *valence* of the emotional content of the headline, as in this sample headline-annotation pair:

Outcry at N Korea 'nuclear test'

(*Anger, 30*), (*Disgust,30*), (*Fear,30*), (*Joy,0*), (*Sadness,20*), (*Surprise,40*), (*Valence,-50*).

For our experiment we select a 100-headline sample from the original Semeval test set, and collect 10 affect annotations for each of the seven label types, for a total of 7000 affect labels.

We then performed two comparisons to evaluate the quality of the AMT annotations. First, we asked how well the non-experts agreed with the experts. We did this by comparing the interannotator agreement (ITA) of individual expert annotations to that of single non-expert and averaged non-expert annotations. In the original experiment ITA is measured by calculating the Pearson correlation of one annotator's labels with the average of the labels of the other five annotators. For each expert labeler, we computed this ITA score of the expert against the other five; we then average these ITA scores across all expert annotators to compute the average expert ITA (reported in Table 1 as "E vs. E". We then do the same for individual non-expert annotations, averaging Pearson correlation across all sets of the five expert labelers ("NE vs. E"). We then calculate the ITA for each expert vs. the averaged labels from all other experts and non-experts (marked as "E vs. All") and for each non-expert vs. the pool of other non-experts and all experts ("NE vs. All"). We compute these ITA scores for each emotion task separately, averaging the six emotion tasks as "Avg. Emo" and the average of all tasks as "Avg. All".

Emotion	E vs. E	E vs. All	NE vs. E	NE vs. All
Anger	0.459	0.503	0.444	0.573
Disgust	0.583	0.594	0.537	0.647
Fear	0.711	0.683	0.418	0.498
Joy	0.596	0.585	0.340	0.421
Sadness	0.645	0.650	0.563	0.651
Surprise	0.464	0.463	0.201	0.225
Valence	0.759	0.767	0.530	0.554
Avg. Emo	0.576	0.603	0.417	0.503
Avg. All	0.580	0.607	0.433	0.510

Table 1: Average expert and non-expert ITA on test-set

The results in Table 1 conform to the expectation that experts are better labelers: experts agree with experts more than non-experts agree with experts, although the ITAs are in many cases quite close. But we also found that adding non-experts to the gold standard ("E vs. All") improves agreement, suggesting that non-expert annotations are good enough to increase the overall quality of the gold labels. Our first comparison showed that individual experts were better than individual non-experts. In our next comparison we ask how many averaged non-experts it would take to rival the performance of a single expert. We did this by averaging the labels of each possible subset of n non-expert annotations, for value of n in $\{1, 2, \ldots, 10\}$. We then treat this average as though it is the output of a single 'meta-labeler', and compute the ITA with respect to each subset of five of the six expert annotators. We then average the results of these studies across each subset size; the results of this experiment are given in Table 2 and in figure 1. In addition to the single meta-labeler, we ask: what is the minimum number of non-expert annotations k from which we can create a meta-labeler that has equal or better ITA than an expert annotator? In Table 2 we give the minimum k for each emotion, and the averaged ITA for that meta-labeler consisting of k non-experts (marked "k-NE"). In Figure 1 we plot the expert ITA correlation as the horizontal dashed line.

These results show that for all tasks except "Fear" we are able to achieve expert-level ITA with the held-out set of experts within 9 labelers, and frequently within only 2 labelers. On average it requires only 4 non-expert annotations per example to achieve the equivalent ITA as a single expert annotator. Thus, given that we paid US$2.00 in order to

Emotion	1-Expert	10-NE	k	k-NE
Anger	0.459	0.675	2	0.536
Disgust	0.583	0.746	2	0.627
Fear	0.711	0.689	–	–
Joy	0.596	0.632	7	0.600
Sadness	0.645	0.776	2	0.656
Surprise	0.464	0.496	9	0.481
Valence	0.759	0.844	5	0.803
Avg. Emo.	0.576	0.669	4	0.589
Avg. All	0.603	0.694	4	0.613

Table 2: Average expert and averaged correlation over 10 non-experts on test-set. k is the minimum number of non-experts needed to beat an average expert.

collect the 7000 non-expert annotations, we may interpret our rate of 3500 non-expert labels per USD on this task as at least 875 expert-equivalent labels per USD.

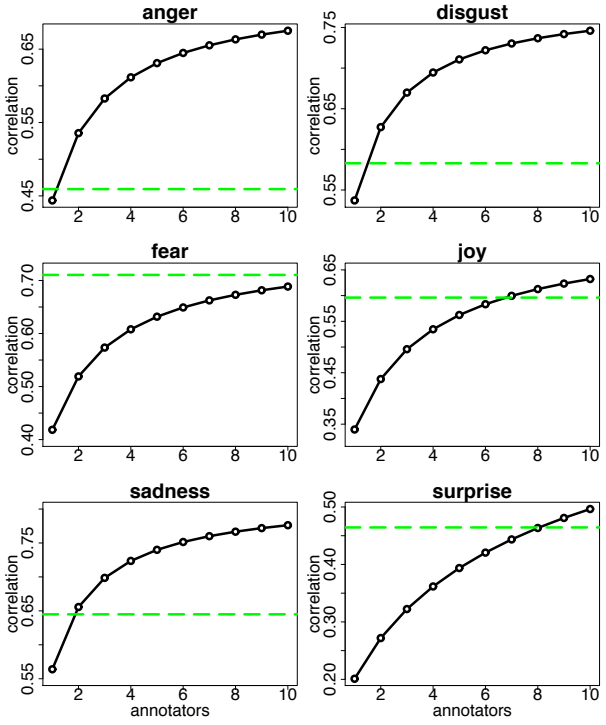

Figure 1: Non-expert correlation for affect recognition

4.2 Word Similarity

This task replicates the word similarity task used in (Miller and Charles, 1991), following a previous task initially proposed by (Rubenstein and Goodenough, 1965). Specifically, we ask for numeric judgments of word similarity for 30 word pairs on a scale of [0,10], allowing fractional responses[3]. Numerous expert and non-expert studies have shown that this is task tends to yield very high interannotator agreement as measured by Pearson correlation; (Miller and Charles, 1991) found a 0.97 correlation of the annotations of 38 subjects with the annotations given by 51 subjects in (Rubenstein and Goodenough, 1965), and a following study (Resnik, 1999) with 10 subjects found a 0.958 correlation with (Miller and Charles, 1991).

In our experiment we ask for 10 annotations each of the full 30 word pairs, at an offered price of $0.02 for each set of 30 annotations (or, equivalently, at the rate of 1500 annotations per USD). The most surprising aspect of this study was the speed with which it was completed; the task of 300 annotations was completed by 10 annotators in less than 11 minutes from the time of submission of our task to AMT, at the rate of 1724 annotations / hour.

As in the previous task we evaluate our non-expert annotations by averaging the numeric responses from each possible subset of n annotators and computing the interannotator agreement with respect to the gold scores reported in (Miller and Charles, 1991). Our results are displayed in Figure 2, with Resnik's 0.958 correlation plotted as the horizontal line; we find that at 10 annotators we achieve a correlation of 0.952, well within the range of other studies of expert and non-expert annotations.

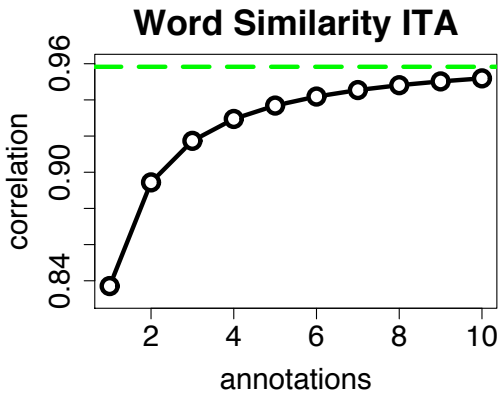

Figure 2: ITA for word similarity experiment

[3](Miller and Charles, 1991) and others originally used a numerical score of [0,4].

4.3 Recognizing Textual Entailment

This task replicates the recognizing textual entailment task originally proposed in the PASCAL Recognizing Textual Entailment task (Dagan et al., 2006); here for each question the annotator is presented with two sentences and asked whether the second sentence can be inferred from the first. We gather 10 annotations each for all 800 sentence pairs in the PASCAL RTE-1 dataset. For this dataset expert interannotator agreement studies have been reported as achieving 91% and 96% agreement over various subsections of the corpus. For greater than 1 annotation we employ a simple voting mechanism; further, we break ties randomly and average our performance over all possible ways to break ties. We collect 10 annotations for each of 100 RTE sentence pairs; as displayed in Figure 3, we achieve a maximum accuracy of 89.7%, averaging over the annotations of 10 workers.

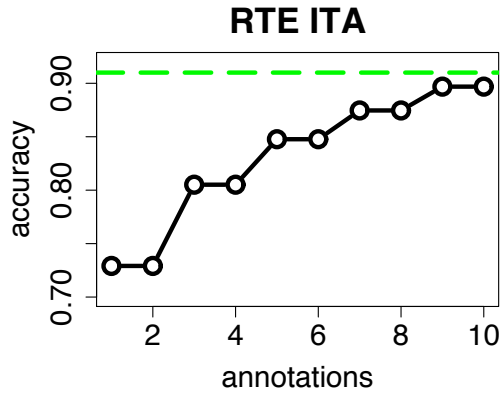

Figure 3: Inter-annotator agreement for RTE experiment

4.4 Event Annotation

This task is inspired by the TimeBank corpus (Pustejovsky et al., 2003), which includes among its annotations a label for event-pairs that represents the temporal relation between them, from a set of fourteen relations (*before*, *after*, *during*, *includes*, etc.).

We implement *temporal ordering* as a simplified version of the TimeBank event temporal annotation task: rather than annotating all fourteen event types, we restrict our consideration to the two simplest labels: "strictly before" and "strictly after". Furthermore, rather than marking both nouns and verbs in the text as possible events, we only consider possible verb events. We extract the 462 verb event pairs labeled as "strictly before" or "strictly after" in the TimeBank corpus, and we present these pairs to annotators with a forced binary choice on whether the first verb event in the pair occurs *before* or *after* one another. The results of this task are presented in Figure 4. We achieve high agreement for this task, at a rate of 0.94 with simple voting over 10 annotators (4620 total annotations). While an expert ITA of 0.77 was reported for the more general task involving all fourteen labels on both noun and verb events, no expert ITA numbers have been reported for this simplified temporal ordering task.

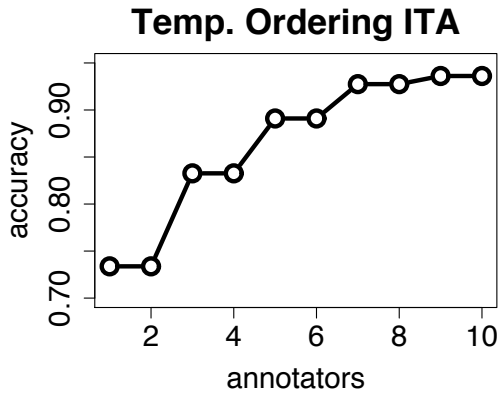

Figure 4: ITA for temporal ordering experiment

4.5 Word Sense Disambiguation

In this task we consider a very easy problem on which machine learning algorithms have been shown to produce extremely good results; here we annotate part of the Semeval Word Sense Disambiguation Lexical Sample task (Pradhan et al., 2007); specifically, we collect 10 annotations for each of 177 examples of the noun "president" for the three senses given in Semeval. As shown in Figure 5, performing simple voting over annotators results in a rapid accuracy plateau at a very high rate of 0.994 accuracy. In fact, further analysis reveals that there was only a single disagreement between the averaged nonexpert-annotator vote and the gold standard; on inspection it was observed that the annotators voted strongly against the original gold label (9-to-1 against), and that it was in fact revealed to

be an error in the original gold standard annotation.[4] After correcting this error, the non-expert accuracy rate is 100% on the 177 examples in this task. This is a specific example where non-expert annotations can be used to correct expert annotations.

Since expert ITA was not reported per word on this dataset, we compare instead to the performance of the best automatic system performance for disambiguating "president" in Semeval 17 (Cai et al., 2007), with an accuracy of 0.98.

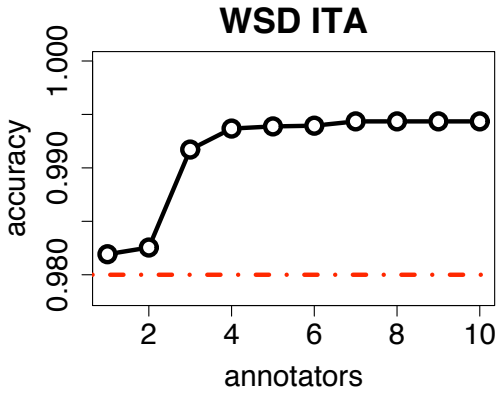

Figure 5: Inter-annotator agreement for WSD experiment

4.6 Summary

Task	Labels	Cost (USD)	Time (hrs)	Labels per USD	Labels per hr
Affect	7000	$2.00	5.93	3500	1180.4
WSim	300	$0.20	0.174	1500	1724.1
RTE	8000	$8.00	89.3	1000	89.59
Event	4620	$13.86	39.9	333.3	115.85
WSD	1770	$1.76	8.59	1005.7	206.1
Total	21690	25.82	143.9	840.0	150.7

Table 3: Summary of costs for non-expert labels

In Table 3 we give a summary of the costs associated with obtaining the non-expert annotations for each of our 5 tasks. Here *Time* is given as the total amount of time in hours elapsed from submitting the group of HITs to AMT until the last assignment is submitted by the last worker.

[4]The example sentence began "*The Egyptian president said he would visit Libya today...*" and was mistakenly marked as the "head of a company" sense in the gold annotation (example id 24:0@24@wsj/23/wsj_2381@wsj@en@on).

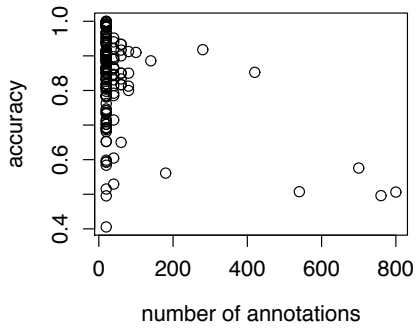

Figure 6: Worker accuracies on RTE task. Each point is one worker. Vertical jitter has been added to points on the left to show the large number of workers who did the minimum amount of work (20 examples).

5 Bias correction for non-expert annotators

The reliability of individual workers varies. Some are very accurate, while others are more careless and make mistakes; and a small few give very noisy responses. Furthermore, for most AMT data collection experiments, a relatively small number of workers do a large portion of the task, since workers may do as much or as little as they please. Figure 6 shows accuracy rates for individual workers on one task. Both the overall variability, as well as the prospect of identifying high-volume but low-quality workers, suggest that for controlling individual worker quality could yield higher quality overall judgments.

In general, there are at least three ways to enhance quality in the face of worker error. More workers can be used, as described in previous sections. Another method is to use Amazon's compensation mechanisms to give monetary bonuses to highly-performing workers and deny payments to unreliable ones; this is useful, but beyond the scope of this paper. In this section we explore a third alternative, to model the reliability of individual workers and correct for their biases.

A wide number of methods have been explored to correct for the bias of annotators. Dawid and Skene (1979) consider the case of having multiple annotators per example but unknown true labels. They in-

troduce an EM algorithm to simultaneously estimate annotator biases and latent label classes. Wiebe et al. (1999) analyze annotator agreement statistics to find bias, and use a similar model to correct labels.

Here we consider the slightly different problem of using a small amount of expert-labeled training data in order to correct for the individual biases of different non-expert annotators. The idea is to recalibrate worker's responses to more closely match expert behavior. We restrict our attention to categorical examples, though in principle a similar method could be used with numeric data.

5.1 Bias correction in categorical data

Following Dawid and Skeene, we model labels and workers with a multinomial model similar to Naive Bayes. Every example i has a true label x_i. For simplicity, assume two labels $\{Y, N\}$. Several different workers give labels $y_{i1}, y_{i2}, \ldots y_{iW}$. A worker's conditional probability of response is modeled as multinomial. Each worker's judgment is conditionally independent of other workers given the true label x_i, i.e.:

$$P(y_{i1}, \ldots, y_{iW}, x_i) = \left(\prod_w P(y_{iw}|x_i) \right) p(x_i)$$

To infer the posterior probability of the true label for a new example, worker judgments are integrated via Bayes rule, yielding the posterior log-odds:

$$\begin{aligned} & \log \frac{P(x_i = Y|y_{i1} \ldots y_{iW})}{P(x_i = N|y_{i1} \ldots y_{iW})} \\ = & \sum_w \log \frac{P(y_{iw}|x_i = Y)}{P(y_{iw}|x_i = N)} + \log \frac{P(x_i = Y)}{P(x_i = N)} \end{aligned}$$

The worker response likelihoods $P(y_w|x = Y)$ and $P(y_w|x = N)$ can be directly estimated from frequencies of worker performance on gold standard examples. (If we used maximum likelihood estimation with no Laplace smoothing, then each $y_w|x$ is just the worker's empirical confusion matrix.) For MAP label estimation, the above equation describes a weighted voting rule: each worker's vote is weighted by their log likelihood ratio for their given response. Intuitively, workers who are more than 50% accurate have positive votes; workers whose judgments are pure noise have zero votes; and anticorrelated workers have negative votes. (A simpler form of the model only considers accuracy rates, thus weighting worker votes by $\log \frac{\text{acc}_w}{1-\text{acc}_w}$. But we use the full unconstrained multinomial model here.)

5.1.1 Example tasks: RTE-1 and event annotation

We used this model to improve accuracy on the RTE-1 and event annotation tasks. (The other categorical task, word sense disambiguation, could not be improved because it already had maximum accuracy.) First we took a sample of annotations giving k responses per example. Within this sample, we trained and tested via 20-fold cross-validation across examples. Worker models were fit using Laplace smoothing of 1 pseudocount; label priors were uniform, which was reasonably similar to the empirical distribution for both tasks.

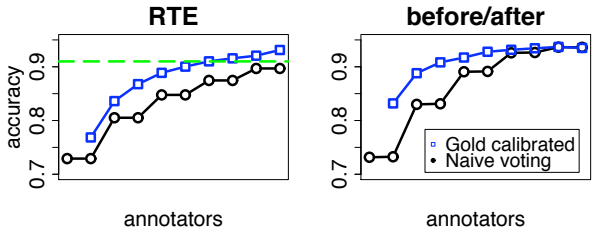

Figure 7: Gold-calibrated labels versus raw labels

Figure 7 shows improved accuracy at different numbers of annotators. The lowest line is for the naive 50% majority voting rule. (This is equivalent to the model under uniform priors and equal accuracies across workers and labels.) Each point is the data set's accuracy against the gold labels, averaged across resamplings each of which obtains k annotations per example. RTE has an average +4.0% accuracy increase, averaged across 2 through 10 annotators. We find a +3.4% gain on event annotation. Finally, we experimented with a similar calibration method for numeric data, using a Gaussian noise model for each worker: $y_w|x \sim N(x + \mu_w, \sigma_w)$. On the affect task, this yielded a small but consistent increases in Pearson correlation at all numbers of annotators, averaging a +0.6% gain.

6 Training a system with non-expert annotations

In this section we train a supervised affect recognition system with expert vs. non-expert annotations.

6.1 Experimental Design

For the purpose of this experiment we create a simple bag-of-words unigram model for predicting affect and valence, similar to the SWAT system (Katz et al., 2007), one of the top-performing systems on the SemEval Affective Text task.[5] For each token t in our training set, we assign t a weight for each emotion e equal to the average emotion score observed in each headline H that t participates in. i.e., if $\mathbf{H}_t$ is the set of headlines containing the token t, then:

$$Score(e,t) = \frac{\sum_{H \in \mathbf{H}_t} Score(e,H)}{|\mathbf{H}_t|}$$

With these weights of the individual tokens we may then compute the score for an emotion e of a new headline H as the average score over the set of tokens $t \in H$ that we've observed in the training set (ignoring those tokens not in the training set), i.e.:

$$Score(e,H) = \sum_{t \in H} \frac{Score(e,t)}{|H|}$$

Where $|H|$ is simply the number of tokens in headline H, ignoring tokens not observed in the training set.

6.2 Experiments

We use 100 headlines as a training set (examples 500-599 from the test set of Semeval Task 14), and we use the 900 headlines as our test set. Since we are fortunate to have the six separate expert annotations in this task, we can perform an extended systematic comparison of the performance of the classifier trained with expert vs. non-expert data.

For this evaluation we compare the performance of systems trained on expert and non-expert annotations. For each expert annotator we train a system using only the judgments provided by that annotator, and then create a gold standard test set using the average of the responses of the remaining five labelers on that set. In this way we create six independent expert-trained systems and compute the average across their performance, calculated as Pearson correlation to the gold standard; this is reported in the "1-Expert" column of Table 4.

Emotion	1-Expert	10-NE	k	k-NE
Anger	0.084	0.233	1	0.172
Disgust	0.130	0.231	1	0.185
Fear	0.159	0.247	1	0.176
Joy	0.130	0.125	–	–
Sadness	0.127	0.174	1	0.141
Surprise	0.060	0.101	1	0.061
Valence	0.159	0.229	2	0.146
Avg. Emo	0.116	0.185	1	0.135
Avg. All	0.122	0.191	1	0.137

Table 4: Performance of expert-trained and non-expert-trained classifiers on test-set. k is the minimum number of non-experts needed to beat an average expert.

Next we train systems using non-expert labels; for each possible subset of n annotators, for $n \in \{1, 2, \ldots, 10\}$ we train a system, and evaluate by calculating Pearson correlation with the same set of gold standard datasets used in the expert-trained system evaluation. Averaging the results of these studies yields the results in Table 4.

As in Table 2 we calculate the minimum number of non-expert annotations per example k required on average to achieve similar performance to the expert annotations; surprisingly we find that for five of the seven tasks, the average system trained with a single set of non-expert annotations outperforms the average system trained with the labels from a single expert. One possible hypothesis for the cause of this non-intuitive result is that individual labelers (including experts) tend to have a strong bias, and since multiple non-expert labelers may contribute to a single set of non-expert annotations, the annotator diversity within the single set of labels may have the effect of reducing annotator bias and thus increasing system performance.

7 Conclusion

We demonstrate the effectiveness of using Amazon Mechanical Turk for a variety of natural language

[5] Unlike the SWAT system we perform no lemmatization, synonym expansion, or any other preprocessing of the tokens; we simply use whitespace-separated tokens within each headline.

annotation tasks. Our in-depth evaluation of labeler data vs. expert annotations for five tasks found that for many tasks only a small number of annotations per item are necessary to equal the performance of an expert annotator. In a detailed study of expert and non-expert agreement for an affect recognition task we find that we require an average of 4 non-expert labels per item in order to emulate expert-level label quality. Finally, we demonstrate significant improvement by controlling for labeler bias.

References

Collin F. Baker, Charles J. Fillmore and John B. Lowe. 1998. The Berkeley FrameNet project. In *Proc. of COLING-ACL 1998*.

Michele Banko and Eric Brill. 2001. Scaling to Very Very Large Corpora for Natural Language Disambiguation. In *Proc. of ACL-2001*.

Junfu Cai, Wee Sun Lee and Yee Whye Teh. 2007. Improving Word Sense Disambiguation Using Topic Features. In *Proc. of EMNLP-2007* .

Timothy Chklovski and Rada Mihalcea. 2002. Building a sense tagged corpus with Open Mind Word Expert. In *Proc. of the Workshop on "Word Sense Disambiguation: Recent Successes and Future Directions", ACL 2002*.

Timothy Chklovski and Yolanda Gil. 2005. Towards Managing Knowledge Collection from Volunteer Contributors. Proceedings of AAAI Spring Symposium on Knowledge Collection from Volunteer Contributors (KCVC05).

Ido Dagan, Oren Glickman and Bernardo Magnini. 2006. The PASCAL Recognising Textual Entailment Challenge. Machine Learning Challenges. Lecture Notes in Computer Science, Vol. 3944, pp. 177-190, Springer, 2006.

A. P. Dawid and A. M. Skene. 1979. Maximum Likelihood Estimation of Observer Error-Rates Using the EM Algorithm. Applied Statistics, Vol. 28, No. 1 (1979), pp. 20-28.

Michael Kaisser and John B. Lowe. 2008. A Research Collection of QuestionAnswer Sentence Pairs. In *Proc. of LREC-2008*.

Michael Kaisser, Marti Hearst, and John B. Lowe. 2008. Evidence for Varying Search Results Summary Lengths. In *Proc. of ACL-2008*.

Phil Katz, Matthew Singleton, Richard Wicentowski. 2007. SWAT-MP: The SemEval-2007 Systems for Task 5 and Task 14. In *Proc. of SemEval-2007*.

Mitchell P. Marcus , Mary Ann Marcinkiewicz , and Beatrice Santorini. 1993. Building a large annotated corpus of English: the Penn Treebank. Computational Linguistics, v.19 n.2, June 1993.

George A. Miller and William G. Charles. 1991. Contextual Correlates of Semantic Similarity. Language and Cognitive Processes, vol. 6, no. 1, pp. 1-28, 1991.

George A. Miller, Claudia Leacock, Randee Tengi, and Ross T. Bunke. 1993. A semantic concordance. In *Proc. of HLT-1993*.

Preslav Nakov. 2008. Paraphrasing Verbs for Noun Compound Interpretation. In *Proc. of the Workshop on Multiword Expressions, LREC-2008*.

Martha Palmer, Dan Gildea, and Paul Kingsbury. 2005. The Proposition Bank: A Corpus Annotated with Semantic Roles. Computational Linguistics Journal, 31:1, 2005.

Sameer Pradhan, Edward Loper, Dmitriy Dligach and Martha Palmer. 2007. SemEval-2007 Task-17: English Lexical Sample, SRL and All Words. In *Proc. of the Fourth International Workshop on Semantic Evaluations (SemEval-2007), Association for Computational Linguistics, 2007*.

James Pustejovsky, Patrick Hanks, Roser Saur, Andrew See, Robert Gaizauskas, Andrea Setzer, Dragomir Radev, Beth Sundheim, David Day, Lisa Ferro and Marcia Lazo. 2003. The TIMEBANK Corpus. In *Proc. of Corpus Linguistics 2003, 647-656*.

Philip Resnik. 1999. Semantic Similarity in a Taxonomy: An Information-Based Measure and its Application to Problems of Ambiguity in Natural Language. In JAIR, Volume 11, pages 95-130.

Herbert Rubenstein and John B. Goodenough. 1965. Contextual Correlates of Synonymy. Communications of the ACM, 8(10):627–633.

Push Singh. 2002. The public acquisition of commonsense knowledge. In *Proc. of AAAI Spring Symposium on Acquiring (and Using) Linguistic (and World) Knowledge for Information Access, 2002*.

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

Carlo Strapparava and Rada Mihalcea. 2007. SemEval-2007 Task 14: Affective Text In *Proc. of SemEval 2007*.

Qi Su, Dmitry Pavlov, Jyh-Herng Chow, and Wendell C. Baker. 2007. Internet-Scale Collection of Human-Reviewed Data. In *Proc. of WWW-2007*.

Luis von Ahn and Laura Dabbish. 2004. Labeling Images with a Computer Game. In ACM Conference on Human Factors in Computing Systems, CHI 2004. Pages 319-326.

Luis von Ahn, Mihir Kedia and Manuel Blum. 2006. Verbosity: A Game for Collecting Common-Sense Knowledge. In ACM Conference on Human Factors in Computing Systems, CHI Notes 2006. Pages 75-78.

Ellen Voorhees and Hoa Trang Dang. 2006. Overview of the TREC 2005 question answering track. In *Proc. of TREC-2005*.

Janyce M. Wiebe, Rebecca F. Bruce and Thomas P. O'Hara. 1999. Development and use of a gold-standard data set for subjectivity classifications. In *Proc. of ACL-1999*.

Annie Zaenen. Submitted. Do give a penny for their thoughts. International Journal of Natural Language Engineering (submitted).