Annotating Large Email Datasets for Named Entity Recognition with Mechanical Turk

Nolan Lawson, Kevin Eustice, Mike Perkowitz
Kiha Software
100 South King Street, Suite 320
Seattle, WA 98104
{nolan,kevin,mikep}@kiha.com

Meliha Yetisgen Yildiz
Medical and Health Informatics
University of Washington
Seattle, WA 98101
melihay@u.washington.edu

Abstract

Amazon's Mechanical Turk service has been successfully applied to many natural language processing tasks. However, the task of named entity recognition presents unique challenges. In a large annotation task involving over 20,000 emails, we demonstrate that a competitive bonus system and inter-annotator agreement can be used to improve the quality of named entity annotations from Mechanical Turk. We also build several statistical named entity recognition models trained with these annotations, which compare favorably to similar models trained on expert annotations.

1 Introduction

It is well known that the performance of many machine learning systems is heavily determined by the size and quality of the data used as input to the training algorithms. Additionally, for certain applications in natural language processing (NLP), it has been noted that the particular algorithms or feature sets used tend to become irrelevant as the size of the corpus increases (Banko and Brill 2001). It is therefore not surprising that obtaining large annotated datasets is an issue of great practical importance for the working researcher. Traditionally, annotated training data have been provided by experts in the field or the researchers themselves, often at great costs in terms of time and money. Recently, however, attempts have been made to leverage non-expert annotations provided by Amazon's Mechanical Turk (AMT) service to create large training corpora at a fraction of the usual costs (Snow *et al.* 2008). The initial results seem promising, and a new avenue for enhancing existing sources of annotated data appears to have been opened.

Named entity recognition (NER) is one of the many fields of NLP that rely on machine learning methods, and therefore large training corpora. Indeed, it is a field where more is almost always better, as indicated by the traditional use of named entity gazetteers (often culled from external sources) to simulate data that would have been inferred from a larger training set (Minkov *et al.* 2005; Mikheev *et al.* 1999). Therefore, it appears to be a field that could profit from the enormous bargain-price workforce available through AMT.

It is not immediately obvious, though, that AMT is well-suited for the task of NER annotation. Commonly, AMT has been used for the classification task (Snow *et al.* 2008) or for straightforward data entry. However, NER does not fit well into either of these formats. As pointed out by Kozareva (2006), NER can be thought of as a composition of two subtasks: 1) determining the start and end boundaries of a textual entity, and 2) determining the label of the identified span. The second task is the well-understood classification task, but the first task presents subtler problems. One is that AMT's form-based user interface is inappropriate for the task of identifying textual spans. Another problem is that AMT's fixed-fee payment system encourages low recall on the part of the annotators, since they receive the same pay no matter how many entities they identify.

This paper addresses both of these problems by describing a custom user interface and competitive payment system that together create a fluid user experience while encouraging high-quality annotations. Further, we demonstrate that AMT successfully scales to the task of annotating a very large set of documents (over 20,000), with each document annotated by multiple workers. We also present a system for resolving inter-annotator conflicts to create the final training corpus, and determine the ideal agreement threshold to maximize precision and recall of a statistical named entity recognition model. Finally, we demonstrate that a model trained on our corpus is on par with one trained from expert annotations, when applied to a labeled test set.

2 Related Work

AMT is a virtual market in which any requester can post tasks that are simple for humans but difficult for computers. AMT has been adopted for a variety of uses both in industry and academia from user studies (Kittur *et al.* 2008) to image labeling (Sorokin and Forsyth 2008). In March 2007, Amazon claimed the user base of AMT consisted of over 100,000 users from 100 countries (Pontin 2007).

In the scope of this paper, we examine the feasibility of AMT in creating large-scale corpora for training statistical named entity recognition models. However, our work was not the first application of AMT in the NLP domain. Snow *et al.* (2008) examined the quality of labels created by AMT workers for various NLP tasks including word sense disambiguation, word similarity, text entailment, and temporal ordering. Since the publication of Snow *et al.*'s paper, AMT has become increasingly popular as an annotation tool for NLP research. Examples include Nakov's work on creating a manually annotated resource for noun-noun compound interpretation based on paraphrasing verbs by AMT (Nakov 2008) and Callison-Burch's machine translation evaluation study with AMT (Callison-Burch 2009). In contrast to the existing research, we both evaluated the quality of corpora generated by AMT in different named entity recognition tasks and explored ways to motivate the workers to do higher quality work. We believe the experiences we present in this paper will contribute greatly to other researchers as they design similar large-scale annotation tasks.

3 General Problem Definition

Named entity recognition (NER) is a well-known subtask of information extraction. Traditionally, the task has been based on identifying words and phrases that refer to various entities of interest, including persons, locations, and organizations, (Nadeau and Sekine 2007). The problem is usually posed as a sequence labeling task similar to the part-of-speech (POS) tagging or phrase-chunking tasks, where each token in the input text corresponds to a label in the output, and is solved with sequential classification algorithms (such as CRF, SVMCMM, or MEMM).

Previous works have tackled NER within the biomedical domain (Settles 2004), newswire domain (Grishman and Sundheim 1996), and email domain (Minkov *et al.* 2005). In this paper, we focus on extracting entities from email text.

It should be noted that email text has many distinctive features that create a unique challenge when applying NER. For one, email text tends to be more informal than either newswire or biomedical text, which reduces the usefulness of learned features that depend on patterns of capitalization and spelling. Also, the choice of corpora in email text is particularly important. As email corpora tend to come from either a single company (e.g., the Enron Email Dataset[1]) or a small group of people (e.g., the Sarah Palin email set[2]), it is easy to build a classifier that overfits the data. For instance, a classifier trained to extract personal names from Enron emails might show an especially high preference to words such as "White," "Lay," and "Germany," because they correspond to the names of Enron employees.

Within the newswire and biomedical domains, such overfitting may be benign or actually beneficial, since documents in those domains tend to deal with a relatively small and pre-determined set of named entities (e.g., politicians and large corporations for newswire text, gene and protein names for biomedical text). For NER in the email domain, however, such overfitting is unacceptable. The personal nature of emails ensures that they will almost always contain references to people, places, and organizations not covered by the training data. Therefore, for the classifier to be useful on any spontaneous piece of email text, a large, heterogeneous training set is desired.

To achieve this effect, we chose four different sources of unlabeled email text to be annotated by the Mechanical Turk workers for input into the training algorithms:

1. The Enron Email Dataset.
2. The 2005 TREC Public Spam Corpus (non-spam only).[3]
3. The 20 Newsgroups Dataset.[4]
4. A private mailing list for synthesizer aficionados called "Analogue Heaven."

4 Mechanical Turk for NER

As described previously, AMT is not explicitly designed for NER tasks. Because of this, we decided to build a custom user interface and bonus payment system that largely circumvents the default AMT web interface and instead performs its operations through the AMT Command Line Tools.[5] Additionally, we built a separate set of

[1] http://www.cs.cmu.edu/~enron/
[2] http://palinemail.crivellawest.net/
[3] http://plg.uwaterloo.ca/~gvcormac/treccorpus/
[4] http://people.csail.mit.edu/jrennie/20Newsgroups/
[5] http://mturkclt.sourceforge.net

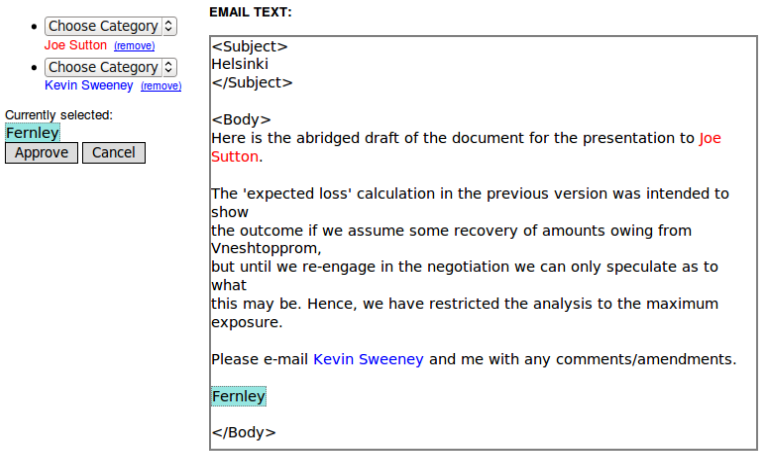

Fig. 1: Sample of the interface presented to workers.

tools designed to determine the ideal number of workers to assign per email.

4.1 User Interface

In order to adapt the task of NER annotation to the Mechanical Turk format, we developed a web-based graphical user interface using JavaScript that allowed the user to select a span of text with the mouse cursor and choose different categories of entities from a dropdown menu. The interface also used simple tokenization heuristics to divide the text into highlightable spans and resolve partial overlaps or double-clicks into the next largest span. For instance, highlighting the word "Mary" from "M" to "r" would result in the entire word being selected.

Each Human Intelligence Task (or HIT, a unit of payable work in the Mechanical Turk system) presented the entire subject and body of an email from one of the four corpora. To keep the HITs at a reasonable size, emails with bodies having less than 60 characters or more than 900 characters were omitted. The average email length, including both subject and body, was 405.39 characters.

For the labeling task, we chose three distinct entity types to identify: PERSON, LOCATION, and ORGANIZATION**.** To reduce potential worker confusion and make the task size smaller, we also broke up each individual HIT by entity type, so the user only had to concentrate on one at a time.

For the PERSON and LOCATION entity types, we noticed during initial tests that there was a user tendency to conflate unnamed references (such as "my mom" and "your house") with true named references. Because NER is intended to be limited only to named entities (i.e., references that contain proper nouns), we asked the users to distinguish between "named" and "unnamed" persons and locations, and to tag both separately. The inclusion of unnamed entities was intended to keep their named counterparts pure and undiluted; the unnamed entities were discarded after the annotation process was complete. The same mechanism could have been used for the ORGANIZATION entity type, but the risk of unnamed references seemed smaller.

Initially, we ran a small trial with a base rate of $0.03 for each HIT. However, after compiling the results we noticed that there was a general tendency for the workers to under-tag the entities. Besides outright freeloaders (i.e., workers who simply clicked "no entities" each time), there were also many who would highlight the first one or two entities, and then ignore the rest of the email.

This may have been due to a misunderstanding of the HIT instructions, but we conjectured that a deeper reason was that we were paying a base rate regardless of the number of entities identified. Ideally, a HIT with many entities to highlight should pay more than a HIT with fewer. However, the default fixed-rate system was paying the same for both, and the workers were responding to such an inflexible incentive system accordingly. To remedy this situation, we set about to create a payment system that would motivate higher recall on entity-rich emails, while still discouraging the opposite extreme of random over-tagging.

4.2 Bonus Payment System

Mechanical Turk provides two methods for paying workers: fixed rates on each HIT and bonuses to individual workers for especially good work. We chose to leverage these bonuses to form the core of our payment system. Each HIT would pay a base rate of $0.01, but each tagged entity

could elicit a bonus of $0.01-$0.02. PERSON entities paid $0.01, while LOCATION and ORGANIZATION entities paid $0.02 (since they were rarer).

To ensure quality and discourage over-tagging, bonuses for each highlighted entity were limited based on an agreement threshold with other workers. This threshold was usually set such that a majority agreement was required, which was an arbitrary decision we made in order to control costs. The terms of the bonuses were explained in detail in the instructions page for each HIT.

Additionally, we decided to leverage this bonus system to encourage improvements in worker performance over time. Since the agreed-upon spans that elicited bonuses were assumed to be mostly correct, we realized we could give feedback to the workers on these entities to encourage similar performance in the future.

In general, worker bonuses are a mysterious and poorly understood motivational mechanism. Our feedback system attempted to make these bonuses more predictable and transparent. The system we built uses Amazon's "NotifyWorkers" REST API to send messages directly to the workers' email accounts. Bonuses were batched on a daily basis, and the notification emails gave a summary description of the day's bonuses.

```
In recognition of your performance, you
were awarded a bonus of $0.5 ($0.02x25)
for catching the following span(s): ['ve-
gas', 'Mt. Hood', 'Holland', [...]
```

Fig. 2: Example bonus notification.

Both the UI and the bonus/notification system were works in progress that were continually refined based on comments from the worker community. We were pleasantly surprised to find that, throughout the annotation process, the Mechanical Turk workers were generally enthusiastic about the HITs, and also interested in improving the quality of their annotations. Out of 169,156 total HITs, we received 702 comments from 140 different workers, as well as over 50 email responses and a dedicated thread at Turker-Nation.com[6]. Most of the feedback was positive, and negative feedback was almost solely directed at the UI. Based on their comments, we continually tweaked and debugged the UI and HIT instructions, but kept the basic structure of the bonus system.

[6] http://turkers.proboards.com/index.cgi?action=display&board=everyoneelse&thread=3177

4.3 Worker Distribution

With the bonus system in place, it was still necessary to determine the ideal number of workers to assign per email. Previously, Snow *et al.* (2008) used expert annotations to find how many Mechanical Turk workers could "equal" an expert in terms of annotation quality. Because we lacked expert annotations, we developed an alternative system to determine the ideal number of workers based purely on inter-annotator agreement.

As described in the previous section, the most significant problem faced with our HITs was that of low recall. Low precision was generally not considered to be a problem, since, with enough annotators, inter-annotator agreement could always be set arbitrarily high in order to weed out false positives. Recall, on the other hand, could be consistently expected to improve as more annotators were added to the worker pool. Therefore, the only problem that remained was to calculate the marginal utility (in terms of recall) of each additional annotator assigned to an email.

In order to estimate this marginal recall gain for each entity type, we first ran small initial tests with a relatively large number of workers. From these results, we took all the entities identified by at least two workers and set those aside as the gold standard annotations; any overlapping annotations were collapsed into the larger one. Next, for each *n* number of workers between 2 and the size of the entire worker pool, we randomly sampled *n* workers from the pool, re-calculated the entities based on agreement from at least two workers within that group, and calculated the recall relative to the gold standard annotation. The threshold of 2 was chosen arbitrarily for the purpose of this experiment.

From this data we generated a marginal recall curve for each entity type, which roughly approximates how many workers are required per email before recall starts to drop off significantly. As expected, each graph shows a plateau-like behavior as the number of workers increases, but some entity types reach their plateau earlier than others. Most saliently, Person entities seem to require only a few workers to reach a relatively high recall, compared to LOCATION or ORGANIZATION entities.

Based on the expected diminishing returns for each entity type, we determined some number of workers to assign per email that we felt would maximize entity recall while staying within our budgetary limits. After some tinkering and experimentation with marginal recall curves, we ul-

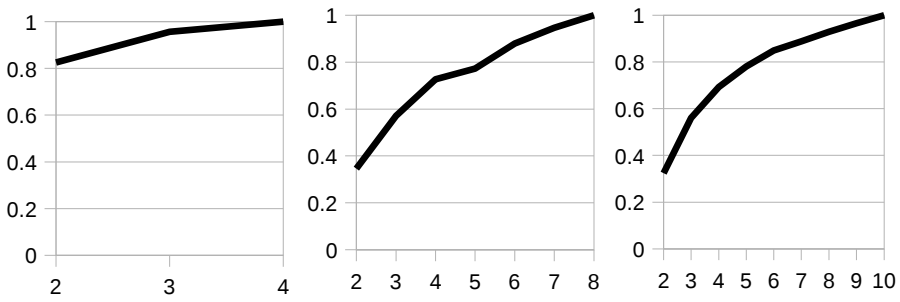

Fig. 3: Marginal recall curves for PERSON, LOCATION, and ORGANIZATION entity types, from a trial run of 900-1,000 emails. Recall is plotted on the y-axis, the number of annotators on the x-axis.

timately settled on 4 assignments for PERSON entities, 6 for LOCATION entities, and 7 for ORGANIZATION entities.

5 Corpora and Experiments

We ran our Mechanical Turk tasks over a period of about three months, from August 2008 to November 2008. We typically processed 500-1,500 documents per day. In the end, the workers annotated 20,609 unique emails which totaled 7.9 megabytes, including both subject and body.

All in all, we were pleasantly surprised by the speed at which each HIT series was completed. Out of 39 total HIT series, the average completion time (i.e. from when the HITs were first posted to MTurk.com until the last HIT was completed) was 3.13 hours, with an average of 715.34 emails per HIT series. The fastest completion time per number of emails was 1.9 hours for a 1,000-email task, and the slowest was 5.13 hours for a 100-email task. We noticed, that, paradoxically, larger HIT series were often completed more quickly – most likely because Amazon promotes the larger tasks to the front page.

5.1 Corpora Annotation

In Table 1, we present several statistics regarding the annotation tasks, grouped by corpus and entity type. Here, "Cost" is the sum of all bonuses and base rates for the HITs, "Avg. Cost" is the average amount we paid in bonuses and base rates per email, "Avg. # Workers" is the average number of workers assigned per email, "Avg. Bonus" is the average bonus per HIT, "Avg. # Spans" is the average number of entities highlighted per HIT, and "Avg. Time" is the average time of completion per HIT in seconds. Precision and recall are reported relative to the "gold standards" determined by the bonus agreement thresholds. None of the reported costs include fees paid to Amazon, which varied based on how the bonuses were batched.

A few interesting observations emerge from these data. For one, the average bonus was usually a bit more than the base rate of $0.01. The implication is that bonuses actually comprised the majority of the compensation, somewhat calling into question their role as a "bonus."

Also noteworthy is that ORGANIZATION entities took less time per identified span to complete than either location or person entities. However, we suspect that this is due to the fact that we ran the ORGANIZATION tasks last (after PERSON and LOCATION), and by that time we had ironed out several bugs in the UI, and our workers had become more adept at using it.

5.2 Worker Performance

In the end, we had 798 unique workers complete 169,156 total HITs. The average number of HITs per worker was 211.97, but the median was only 30. Ten workers who tagged no entities were blocked, and the 1,029 HITs they completed were rejected without payment.

For the most part, a small number of dedicated workers completed the majority of the tasks. The top 10 most prolific workers who were not blocked completed 22.51% of the tasks, the top 25 completed 38.59%, the top 50 completed 55.39%, and the top 100 completed 74.93%.

Callison-Burch (2009) found in their own Mechanical Turk system that the workers who contributed more tended to show lower quality, as measured by agreement with an expert. We had hoped that our bonus system, by rewarding quality work with higher pay, would yield the opposite effect, and in practice, our most prolific workers did indeed tend to show the highest entity recall.

Figure 3 shows how each of the non-rejected workers fared in terms of entity recall (relative to

Corpus	Entity	Cost	#Emails	Avg. Cost	Avg. #Workers	Avg. Bonus	Avg. #Spans	Avg. Precision	Avg. Recall	Avg. Time
20N.	Loc.	315.68	1999	0.1579	6	0.0163	1.6885	0.5036	0.7993	144.34
A.H.	Loc.	412.2	2500	0.1649	6.4	0.0158	1.1924	0.6881	0.8092	105.34
Enron	Loc.	323.54	3000	0.1078	6.23	0.0073	1.0832	0.3813	0.7889	105.25
TREC	Loc.	274.88	2500	0.1100	6	0.0083	1.1847	0.3794	0.7864	122.97
20N.	Org.	438.44	3500	0.1253	7	0.0079	1.2396	0.3274	0.6277	105.68
A.H.	Org.	396.48	2500	0.1586	7	0.0127	1.2769	0.4997	0.7062	92.01
Enron	Org.	539.19	2500	0.2157	8.6	0.0151	1.3454	0.5590	0.7415	80.55
TREC	Org.	179.94	1500	0.1200	7	0.0071	0.8923	0.4414	0.6992	84.23
20N.	Per.	282.51	2500	0.1130	4	0.0183	2.8693	0.7267	0.9297	152.77
A.H.	Per.	208.78	2500	0.0835	4	0.0109	1.6529	0.7459	0.9308	112.4
Enron	Per.	54.11	400	0.1353	6.14	0.0120	2.7360	0.8343	0.8841	111.23
TREC	Per.	214.37	2500	0.0857	4	0.0114	1.5918	0.7950	0.9406	103.73

Table 1: Statistics by corpus and entity type (omits rejected workers).

the "gold standard" determined by the bonus agreement threshold), compared to the number of HITs completed. As the chart shows, out of the 10 most productive workers, only one had an average recall score below 60%, and the rest all had scores above 80%. While there are still quite a few underperforming workers within the core group of high-throughput annotators, the general trend seemed to be that the more HITs a worker completes, the more likely he/she is to agree with the other annotators. This chart may be directly compared to a similar one in Callison-Burch (2009), where the curve takes largely the opposite shape. One interpretation of this is that our bonus system had the desired effect on annotator quality.

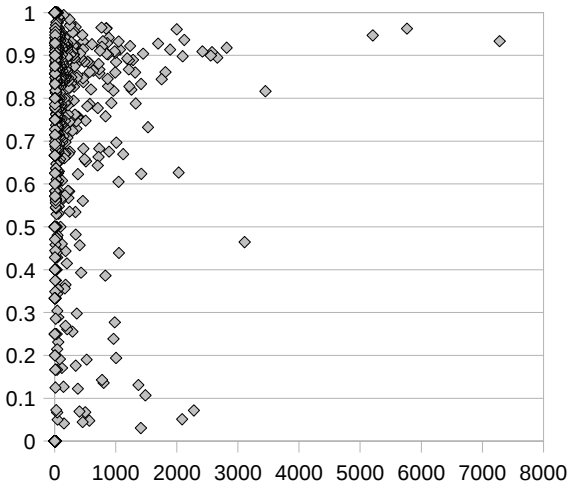

Fig. 3: # HITs Completed vs. Recall

5.3 Annotation Quality Experiments

To evaluate the quality of the worker annotations, one would ideally like to have at least a subset annotated by an expert, and then compare the expert's judgments with the Mechanical Turk workers'. However, in our case we lacked expert annotations for any of the annotated emails. Thus, we devised an alternative method to evaluate the annotation quality, using the NER system built into the open-source MinorThird toolkit.[7]

MinorThird is a popular machine learning and natural language processing library that has previously been applied to the problem of NER with some success (Downey *et al.* 2007). For our purposes, we wanted to minimize the irregularity introduced by deviating from the core features and algorithms available in MinorThird, and therefore did not apply any feature selection or feature engineering in our experiments. We chose to use MinorThird's default "CRFLearner," which is a module that learns feature weights using the IITB CRF library[8] and then applies them to a conditional Markov model-based extractor. All of the parameters were set to their default value, including the built-in "TokenFE" feature extractor, which extracts features for the lowercase value of each token and its capitalization pattern. The version of MinorThird used was 13.7.10.8.

In order to convert the Mechanical Turk annotations to a format that could be input as training into the NER system, we had to resolve the conflicting annotations of the multiple workers into a unified set of labeled documents. To achieve this, we generated four different corpora for each entity type, using a threshold of inter-annotator agreement between 1 and 4. For the PERSON corpora this meant a relatively stricter threshold than for the LOCATION or ORGANIZATION corpora, since the PERSON corpora typically had only 4 annotations per document. Mail subjects and bodies were split into separate files.

Four separate experiments were run with these corpora. The first was a 5-fold cross-evaluation (i.e., a 80%/20% split) train/test experiment on each of the twelve corpora. Because this test did

[7] http://minorthird.sourceforge.net
[8] http://crf.sourceforge.net

not rely on any expert annotations in the gold standard, our goal here was only to roughly measure the "cohesiveness" of the corpus. Low precision and recall scores should indicate a messy corpus, where annotations in the training portion do not necessarily help the extractor to discover annotations in the test portion. Conversely, high precision and recall scores should indicate a more cohesive corpus – one that is at least somewhat internally consistent across the training and test portions.

The second test was another train/test experiment, but with the entire Mechanical Turk corpus as training data, and with a small set of 182 emails, of which 99 were from the W3C Email Corpus[9] and 83 were from emails belonging to various Kiha Software employees, as test data. These 182 test emails were hand-annotated for the three entity types by the authors. Although this test data was small, our goal here was to demonstrate how well the trained extractors could fare against email text from a completely different source than the training data.

The third test was similar to the second, but used as its test data 3,116 Enron emails annotated for PERSON entities.[10] The labels were manually corrected by the authors before testing. The goal here was the same as with the second test, although it must be acknowledged that the PERSON training data did make use of 400 Enron emails, and therefore the test data was not from a completely separate domain.

The fourth test was intended to increase the comparability of our own results with those that others have shown in NER on email text. For the test data, we chose two subsets of the Enron Email Corpus used in Minkov *et al.* (2005).[11] The first, "Enron-Meetings," contains 244 training documents, 242 tuning documents, and 247 test documents. The second, "Enron-Random," contains 89 training documents, 82 tuning documents, and 83 test documents. For each, we tested our statistical recognizers against all three divisions combined as well as the test set alone.

6 Results

The results from these four tests are presented in Tables 2-5. In these tables, "Agr." refers to inter-annotator agreement, "TP" refers to token precision, "SR" to span recall, "TF" to token F-measure, etc.

Entity	Agr.	TP	TR	TF
Loc.	1	60.07%	54.65%	57.23%
	2	**75.47%**	**70.51%**	**72.90%**
	3	71.59%	60.99%	65.86%
	4	59.50%	41.40%	48.83%
Org.	1	70.79%	49.34%	58.15%
	2	**77.98%**	55.97%	**65.16%**
	3	38.96%	**57.87%**	46.57%
	4	64.68%	50.19%	56.52%
Per.	1	86.67%	68.27%	76.38%
	2	**89.97%**	**77.36%**	**83.19%**
	3	87.58%	76.19%	81.49%
	4	75.19%	63.76%	69.00%

Table 2: Cross-validation test results.

Entity	Agr.	TP	TR	TF
Loc.	1	65.90%	37.52%	47.82%
	2	**83.33%**	**56.28%**	**67.19%**
	3	84.05%	48.12%	61.20%
	4	84.21%	26.10%	39.85%
Org.	1	41.03%	35.54%	38.09%
	2	62.89%	**30.77%**	**41.32%**
	3	**66.00%**	15.23%	24.75%
	4	84.21%	9.85%	17.63%
Per.	1	85.48%	**70.81%**	**77.45%**
	2	69.93%	69.72%	69.83%
	3	86.95%	64.40%	73.99%
	4	**95.02%**	43.29%	59.49%

Table 3: Results from the second test.

The cross-validation test results seem to indicate that, in general, an inter-annotator agreement threshold of 2 produces the most cohesive corpora regardless of the number of workers assigned per email. In all cases, the F-measure peaks at 2 and then begins to drop afterwards.

The results from the second test, using the W3C and Kiha emails as test data, tell a slightly different story, however. One predictable observation from these data is that precision tends to increase as more inter-annotator agreement is required, while recall decreases. We believe that this is due to the fact that entities that were confirmed by more workers tended to be less controversial or ambiguous than those confirmed by fewer. Most surprising about these results is that, although F-measure peaks with the 2-agreement corpora for both LOCATION and ORGANIZATION entities, PERSON entities actually show the worst precision when using the 2-agreement corpus. In the case of PERSON entities, the corpus generated using no inter-annotator agreement at all, i.e., anno-

[9] http://tides.umiacs.umd.edu/webtrec/trecent/parsed_w3c_corpus.html

[10] http://www.cs.cmu.edu/~wcohen/repository.tgz and http://www.cs.cmu.edu/~einat/datasets.html.

[11] http://www.cs.cmu.edu/~einat/datasets.html.

tator agreement of 1, actually performs the best in terms of F-measure.

Agr.	TP	TR	TF
1	80.56%	62.55%	70.42%
2	85.08%	**67.66%**	**75.37%**
3	93.25%	57.13%	70.86%
4	**95.61%**	39.67%	56.08%

Table 4: Results from the third test.

Data	Agr.	TP	TR	TF	SP	SR	SF
E-M	1	**100%**	57.16%	72.74%	**100%**	50.10%	66.75%
(All)	2	**100%**	**64.31%**	**78.28%**	**100%**	**56.11%**	**71.88%**
	3	**100%**	50.44%	67.06%	**100%**	45.11%	62.18%
	4	**100%**	31.41%	47.81%	**100%**	27.91%	43.64%
E-M	1	**100%**	62.17%	76.68%	**100%**	51.30%	67.81%
(Test)	2	**100%**	**66.36%**	**79.78%**	**100%**	**54.28%**	**70.36%**
	3	**100%**	55.72%	71.56%	**100%**	45.72%	62.76%
	4	**100%**	42.24%	59.39%	**100%**	36.06%	53.01%
E-R	1	36.36%	59.91%	45.25%	40.30%	53.75%	46.07%
(All)	2	70.83%	**65.32%**	67.96%	67.64%	**57.68%**	62.26%
	3	88.69%	58.63%	**70.60%**	82.93%	54.38%	**65.68%**
	4	**93.59%**	43.68%	59.56%	**89.33%**	41.22%	56.41%
E-R	1	**100%**	60.87%	75.68%	**100%**	54.82%	70.82%
(Test)	2	**100%**	**64.70%**	**78.56%**	**100%**	**59.05%**	**74.26%**
	3	**100%**	63.06%	77.34%	**100%**	58.38%	73.72%
	4	**100%**	43.04%	60.18%	**100%**	40.10%	57.25%

Table 5: Results from the fourth test.

With the third test, however, the results are more in line with those from the cross-validation tests: F-measure peaks with the 2-agreement corpus and drops off as the threshold increases. Most likely these results can be considered more significant than those from the second test, since this test corpus contains almost 20 times the number of documents.

For the fourth test, we report both token-level statistics and span-level statistics (i.e., where credit for partially correct entity boundaries is not awarded) in order to increase comparability with Minkov *et al*. (2005). With one exception, these tests seem to show again that the highest F-measure comes from the annotator created using an agreement level of 2, confirming results from the first and third tests.

The fourth test may also be directly compared to the results in Minkov *et al*. (2005), which report span F-measure scores of 59.0% on Enron-Meetings and 68.1% on Enron-Random, for a CRF-based recognizer using the "Basic" feature set (which is identical to ours) and using the "train" division for training and the "test" division for testing. In both cases, our best-performing annotators exceed these scores – an 11.5% improvement on Enron-Meetings and a 6.16% improvement on Enron-Random. This is an encouraging result, given that our training data largely come from a different source than the test data, and that the labels come from non-experts. We see this as confirmation that very large corpora annotated by Mechanical Turk workers can surpass the quality of smaller corpora annotated by experts.

7 Conclusion

In order to quickly and economically build a large annotated dataset for NER, we leveraged Amazon's Mechanical Turk. AMT allowed us to build a dataset of 20,609 unique emails with 169,156 total annotations in less than four months. The AMT worker population responded well to NER tasks, and in particular responded well to the bonus and feedback scheme we put into place to improve annotation quality. The bonus feedback system was designed to improve the transparency of the compensation system and motivate higher quality work over time. Encouragingly, our results indicate that the workers who completed the most documents also had consistently high entity recall, i.e., agreement with other workers, indicating that the system achieved the desired effect.

Given a large body of AMT annotated documents, we were able to leverage inter-annotator agreement to control the precision and recall of a CRF-based recognizer trained on the data. Importantly, we also showed that inter-annotator agreement can be used to predict the appropriate number of workers to assign to a given email in order to maximize entity recall and reduce costs.

Finally, a direct comparison of the entity recognizers generated from AMT annotations to those generated from expert annotations was very promising, suggesting that Mechanical Turk is appropriate for NER annotation tasks, when care is taken to manage annotator error.

Acknowledgments

Thanks to Dolores Labs for the initial version of the UI, and thanks to Amazon and the 798 Mechanical Turk workers for making this work possible.

References

Michele Banko and Eric Brill. 2001. Scaling to very very large corpora for natural language disambiguation. In *ACL '01*:26-33.

Chris Callison-Burch. 2009. Fast, cheap, and creative: evaluating translation quality using Amazon's Mechanical Turk. In *EMNLP '09:*286-295.

Doug Downey, Matthew Broadhead, and Oren Etzioni. 2007. Locating complex named entities in web text. In *IJCAI '07*.

Ralph Grishman and Beth Sundheim. 1996. Message Understanding Conference-6: a brief history. In *Proceedings of the 16th conference on Computational Linguistics*:466-471.

Aniket Kittur, Ed H. Chi, and Bongwon Suh. 2008. Crowdsourcing user studies with Mechanical Turk. In *Proceedings of CHI 2008*.

Zornitsa Kozareva. 2006. Bootstrapping named entity recognition with automatically generated gazetteer lists. In *Proceedings of the Eleventh Conference of the European Chapter of the Association for Computational Linguistics*:15-21.

Andrei Mikheev, Marc Moens, and Claire Grover. 1999. Named entity recognition without gazetteers. In *Proceedings of the Ninth Conference of the European chapter of the Association for Computational Linguistics*:1-8.

Einat Minkov, Richard C. Wang, and William W. Cohen. 2005. Extracting personal names from email: applying named entity recognition to informal text. In *HTL '05*:443-450.

David Nadeau and Satoshi Sekine. 2007. A survey of named entity recognition and classification. *Linguisticae Investigationes*, 30(1):3-26.

Preslav Nakov. 2008. Noun compound interpretation using paraphrasing verbs: Feasibility study. In *Proceedings of the 13th international conference on Artificial Intelligence: Methodology, Systems and Applications* (*AIMSA 2008*):103–117.

Jason Pontin. 2007. Artificial Intelligence, With Help From the Humans. In *New York Times* (March 25, 2007).

Burr Settles. 2004. Biomedical named entity recognition using conditional random fields and rich feature sets. In *Proceedings of the COLING 2004 International Joint Workshop on Natural Language Processing in Biomedicine and its Applications*.

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 *EMNLP '08*:254-263.

Alexander Sorokin and David Forsyth. Utility data annotation with Amazon MTurk. In *Proceedings of Computer Vision and Pattern Recognition Workshop at CVPR'08*.