Preliminary Experience with Amazon's Mechanical Turk for Annotating Medical Named Entities

Meliha Yetisgen-Yildiz, Imre Solti
Biomedical & Health Informatics
University of Washington
Seattle, WA 98195, USA
{melihay,solti}@uw.edu

Fei Xia, Scott Russell Halgrim
Department of Linguistics
University of Washington
Seattle, WA 98195, USA
{fxia,captnpi}@uw.edu

Abstract

Amazon's Mechanical Turk (MTurk) service is becoming increasingly popular in Natural Language Processing (NLP) research. In this paper, we report our findings in using MTurk to annotate medical text extracted from clinical trial descriptions with three entity types: medical condition, medication, and laboratory test. We compared MTurk annotations with a gold standard manually created by a domain expert. Based on the good performance results, we conclude that MTurk is a very promising tool for annotating large-scale corpora for biomedical NLP tasks.

1 Introduction

The manual construction of annotated corpora is extremely expensive both in terms of time and money. Snow et al. (2008) demonstrated the potential power of Amazon's Mechanical Turk (MTurk) service in annotating large corpora for natural language tasks cheaply and quickly. We are working on a Natural Language Processing (NLP) project to automate the clinical trial eligibility screening of patients. This project involves building statistical models for medical named entity recognition which requires a large-scale annotated corpus for training. As part of corpus development, we tested the feasibility of using MTurk for the annotation of medical named entities in biomedical text and we report our findings in this paper.

In the following sections we describe how we used MTurk to annotate the biomedical corpus created from publicly available clinical trial announcements. The main goal of our study was to understand how well non-experts perform compared to medical expert in annotating the biomedical text.

2 Related Work

MTurk[1] is an online micro-task market that allows requesters to distribute work to a large number of workers from all over the world. The inspiration of the system was to have human workers complete simple tasks that would otherwise be extremely difficult for computers to perform (Kittur et al., 2008). A complex task is broken down into simple, one-time tasks called Human Intelligence Tasks (HITs). Requesters post their HITs on the MTurk marketplace by specifying the amount paid for the completion of each task, and the workers select from the available HITs the ones that they would like to work on. In 2007, Amazon claimed that the user base of MTurk consisted of over 100,000 users from 100 countries[2].

MTurk has been adopted for a variety of uses both in industry and academia, ranging from user studies (Kittur et al., 2008) to image labeling (Sorokin and Forsyth, 2008). Snow et al. (2008) examined the quality of labels created by MTurk 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, MTurk has become increasingly popular as an annotation tool for NLP research. Nakov (2008) used MTurk to create a manually annotated resource for noun-noun compound interpretation based on paraphrasing verbs. In a different NLP task, Callison-Burch (2009) used MTurk to evaluate machine translation quality. With a budget of only $10, Callison-Burch demonstrated the feasibility of performing manual evaluations of machine translation quality by recreating judgments from a WMT08 translation task.

In our pilot study we used MTurk to annotate entities in the biomedical text. To our knowledge, this is the first study that investigates the feasibility of MTurk for biomedical named entity annotation.

3 Annotation Task Description

In this section we will describe the types of entities in our annotation task and the details of our corpus creation process.

[1] https://www.MTurk.com/MTurk/welcome

[2] Source: New York Times article "Artificial Intelligence, With Help from the Humans" , Available at: http://www.nytimes.com/2007/03/25/business/yourmoney/25Stream.html

Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk, pages 180–183,
Los Angeles, California, June 2010.

3.1 Entity Types

We used MTurk to annotate the biomedical text for the following three entity types:

- Medical Conditions
  Example: First-degree relative who developed *<Medical_Condition>**breast cancer*** *</Medical_Condition>* at ≤ 50 years of age.
- Medications
  Example: Previous treatment with an *<Medication>**anthracycline**</Medication>* in the metastatic breast cancer setting.
- Laboratory Test
  Example: *<Laboratory_Test>**Platelet count >=100,000 cells/mL**</Laboratory_Test>*.

3.2 Corpus

Our corpus came from the publicly available clinical trial announcements available at the ClinicalTrials.gov website. This website is a registry of federally and privately supported clinical trials conducted in the United States and around the world. The objectives and procedures of each clinical trial are explained in detail along with participant selection criteria and logistical information such as locations and contact information.

For this task we selected 50,109 announcements from the roughly 85,000 announcements posted on the ClinicalTrials.gov site. For selection criteria we relied on the following keywords: "heart | cancer | tumor | influenza | alzheimer | parkinson | malignant | stroke | respiratory | diabetes | pneumonia | nephritis | nephrotic | nephrosis | septicemia | liver | cirrhosis | hypertension | renal | neoplasm". We chose these keywords because they were part of the phrases of diagnoses for the top 12 leading causes of death excluding suicide, homicide and accidents (Heron et al., 2009). We limited the selection to trials for "Adult" or "Senior" patients.

After downloading the corpus of XML files we converted them to ANSI text using ABC Amber XML Converter[3]. 49,794 files successfully converted to ANSI text format. Using a simple regular expression search we selected documents that had both the "Inclusion Criteria" and "Exclusion Criteria" phrases. The final selection process resulted in 35,385 files. From this latest set we randomly selected 100 files to build the corpus for our pilot study. One of the authors, who has medical training, then manually annotated the three entity types in those selected files. We used this annotated set as the gold standard to measure the quality of the MTurk workers' annotations.

4 HIT Design

Biomedical text is full of jargon, and finding the three entity types in such text can be difficult for non-expert annotators. To make the annotation task more convenient for the MTurk workers, we used a customized user interface and provided detailed annotation guidelines. We also tested the bonus system available in the MTurk environment and evaluated the performance of the workers.

4.1 User Interface

In order to adapt the task of entity annotation to the MTurk format, we used an in-house web-based graphical user interface that allows the worker to select a span of text with the mouse cursor. The interface also uses simple tokenization heuristics to divide the text into highlightable spans and resolve partial token highlights or double-clicks into the next largest span. For instance, highlighting the word "cancer" from the second "c" to "e" will result in the entire span "cancer" being highlighted.

4.2 Annotation Guidelines

We created three separate annotation tasks, one for each entity type. For each task, we wrote annotation guidelines that explained the task and showed examples of entities that should be tagged and the ones that should not.

4.3 Bonus System

MTurk provides two methods for paying workers – fixed rates on each document and bonuses to workers for especially good work. In this study, we experimented with the bonus system to see its effect on performance and annotation time. Annotating a document would receive a base rate of $0.01-$0.05, but each tagged entity span could elicit a bonus of $0.01. The base rate would cover the case where the document truly contained no entities, but the bonus amount could potentially be much larger than the base rate if the document was entity-rich. Bonuses for each tagged entity span were awarded based on an agreement threshold with peer workers. In this study, each document was annotated by four workers and we granted bonuses for entity spans that were agreed upon by at least three workers.

4.4 Performance Monitoring

We monitored a worker's performance by comparing the worker's annotations with his/her peer workers' annotations. After we posted the HITs, we continuously monitored the workers' performance and rejected the annotations from the ones who tried to cheat the system by either not doing any annotations (e.g., immediately submitting the document after accepting it) or con-

[3] ABC Amber XML Converter. Available at: http://www.processtext.com/abcxml.html.

Table 1. Cost analysis of annotation experiments ("File" in this table means the annotation of a document. There are 100 documents, and each document is annotated by four workers.)

Experiment Label	File Count		Total Worker Count	MONETARY COST				TIME COST	
				Pay Rate ($)		Total Cost ($)		Completion Time	
	Total	Completed		File	Bonus	File	Bonus	Per file (seconds)	Total (hours)
MedicalCondition-I	400	272	45	0.01	0	2.72	0	156.09	71.16
MedicalCondition-II	400	400	30	0.05	0.01	20	22.61	162.66	7.28
Medication-I	400	400	45	0.01	0.01	4	4.43	87.96	31.65
Medication-II	400	400	17	0.05	0.01	20	6.11	89.06	4.36
Laboratory Test	400	400	26	0.05	0.01	20	1.49	75.61	24.41

stantly doing wrong annotations (e.g., always annotating the first word of the text). Those rejected documents were automatically re-posted on the MTurk so other workers could work on them. In this pilot study, performance monitoring was done mainly manually. As future work, we plan to automate the process in order to scale it for larger annotation tasks.

4.5 Communication with Workers

The workers could send us their questions and comments about the individual documents or the general annotation task through a text box in the interface. During this study we received more than 100 messages from the workers. The majority of the messages were positive messages ("*thank you*", "*easy hit!*"). However, some of the comments included questions such as: "*Is pregnancy a medical condition?*" or "*Text doesn't mention the type of insulin but I highlighted it because insulin is a medication!*". We responded to the questions in a timely manner to increase the quality of annotations.

5 Annotation Experiments

In our annotation experiments, each of 100 documents in our corpus was annotated by four workers, resulting in 100×4=400 files per experiment. We experimented with different pay scales to understand how they affect the quality and speed of the annotations.

5.1 Cost of Annotations

We investigated the cost of annotations both in terms of money and time. The summary of the results is in Table 1. We ran five different MTurk annotation experiments for our corpus of 100 documents. A total of 139 workers were involved in our experiments, and we identified eight of those workers as cheaters and rejected their annotation. The remaining workers spent 138.86 hours to complete 1872 files. The slowest experiment was MedicalCondition-I, in which we paid a base document rate of $0.01 without any bonuses. With this pay scale, it took 71.16 hours for workers to annotate 272 out of 400 files. We suspected we could not attract enough workers to finish the annotation task on time so we stopped the experiment before all 400 files were completed. When we compiled the results, we noticed that there was a general tendency for the workers to tag the first one or two entities and then ignore the rest of the document. Based on this observation, we decided to add bonuses to motivate the workers to read through the whole document. We ran the same annotation task, MedicalCondition-II, with a higher base document rate of $0.05 and a bonus rate of $0.01. With this new payment scale the annotation task was fully completed in 7.28 hours.

We also compared the effect of base rates when the bonus amounts were kept the same. For medication annotations, increasing the base document rate from $0.01 to $0.05 decreased the total amount of annotation time from 31.65 hours to 4.36 hours and also decreased the number of workers from 45 to 17. We ordered the workers based on the number files they annotated. The top ranked 5 workers in Medication-I annotated 187 files (46%) and the top ranked 5 workers in Medication-II annotated 313 files (78%). The difference between those two values was interesting since it indicated that by increasing the base rate, we managed to attract workers who worked on more documents.

The average amount of time workers spent per document varied based on entity type. They spent the longest amount of time for medical condition and shortest amount of time for laboratory test. This can be explained by the richness of documents in terms of entities. In the manually created gold standard there were 1159 mentions of medical condition, 518 mentions of medication, and 249 mentions of laboratory tests. Another observation was that the change in pay scales did not affect the average annotation time per document.

5.2 Quality of Annotations

We measured the quality of the MTurk annotations at different inter-annotator agreement levels by comparing the agreed entity spans with the spans in the gold standard.

Table 2. Quality measurement of MTurk annotations (k: Agreement level, P: Precision, R: Recall, F: F-measure; the highest value for each column is in boldface)

k	Medical Condition-II						Medication-II						Laboratory Test					
	Exact			Overlap			Exact			Overlap			Exact			Overlap		
	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F
1	0.51	**0.66**	0.58	0.70	**0.99**	0.79	0.43	**0.73**	0.54	0.50	**0.84**	0.62	0.30	**0.52**	0.38	0.42	**0.73**	0.53
2	**0.64**	**0.66**	**0.65**	0.84	0.87	**0.86**	0.71	0.66	**0.68**	0.79	0.73	**0.76**	**0.47**	0.43	**0.45**	0.72	0.65	**0.68**
3	0.63	0.52	0.57	0.89	0.73	0.80	**0.78**	0.38	0.51	**0.93**	0.45	0.61	0.29	0.13	0.18	0.86	0.40	0.54
4	0.60	0.31	0.41	**0.93**	0.48	0.63	0.76	0.10	0.18	0.89	0.12	0.21	0.05	0.00	0.01	**1.00**	0.08	0.14

Given a document annotated by multiple workers and an agreement level *k*, there are different ways of creating a new span file that includes only the spans that are agreed by at least k workers. One method is to go over each span in each annotation and output only the spans that are marked by at least k workers. This method does not work well when the spans are long and the workers could disagree on the boundary. We used an alternative method which first goes over each word position in the document and marks the positions that are part of spans in at least k annotations, and then outputs the spans that cover those marked positions. We call the new span file *agreement-k* file.

Once we have created agreement-k file, we compare it with the gold standard to calculate precision, recall, and F-measure. A span in agreement-k file and a span in the gold standard are called an *exact match* if they are identical and are called an *overlap match* if they overlap (exact match is a special case of overlap match). Table 2 shows the performance for the MedicalCondition-II, Medication-II, and LaboratoryTest experiments at different agreement levels (k). As can be seen from the table, as the value of k increased, the precision values increased and the recall values decreased. For all of the experiments, the best F-Score was achieved at agreement-level 2.

Of the three entity types, laboratory test was the hardest partly because laboratory test entities tend to be longer (the average length for entities in gold standard was 5.25 words, compared to 1.84 words for medication and 3.18 words for medical condition), making the exact boundary harder to define. The results for MedicalCondition-II and Medication-II were higher than LaboratoryTest. In addition, accuracy for Medication-I (not shown here due to space limit) and Medication-II were similar, indicating that pay rate did not affect accuracy much in our experiments. In the future, we plan to increase the number of annotations for each document, which we believe could further improve the performance.

6 Conclusion

Human annotation is crucial for many NLP tasks. In this paper, we demonstrated the potential of using MTurk for annotating medical text. By continuously monitoring the workers' performance and using the bonus system, we acquired high quality annotations from non-expert MTurk workers with limited time and budget.

As future work, we plan to analyze the MTurk annotations in detail in order to understand the problematic areas. Based on our observations, we will redesign our annotation tasks and continue our experiments with MTurk to create large-scale annotated corpora to be used in biomedical NLP projects.

Acknowledgement

This project was supported in part by NIH Grants 1K99LM010227-0110 and 5 U54 LM008748.

References

[1] Chris Callison-Burch. 2009. Fast, Cheap, and Creative: Evaluating Translation Quality Using Amazon's Mechanical Turk. In Proceedings of EMNLP'09.

[2] Melonie Heron, Donna L. Hoyert, Sherry L. Murphy, Jiaquan Xu, Kenneth D. Kochanek, and Betzaida Tejada-Vera. 2009. Deaths: Final data for 2006. National Vital Statistics Reports, 57:14.

[3] Aniket Kittur, Ed H. Chi, and Bongwon Suh. 2008. Crowdsourcing User Studies with Mechanical Turk. In Proceedings of CHI'08.

[4] 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.

[5] Philip V. Ogren. 2006. Knowtator: a protégé plug-in for annotated corpus construction. In Proceedings NAACL HLT'06, 273-275.

[6] 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 EMNLP'08, 254-263.

[7] Alexander Sorokin and David Forsyth. Utility data annotation with Amazon Mechanical Turk. In Proceedings of Computer Vision and Pattern Recognition Workshop at CVPR'08.