Creating Speech and Language Data With Amazon's Mechanical Turk

Chris Callison-Burch and **Mark Dredze**
Human Language Technology Center of Excellence
& Center for Language and Speech Processing
Johns Hopkins University
ccb,mdredze@cs.jhu.edu

Abstract

In this paper we give an introduction to using Amazon's Mechanical Turk crowdsourcing platform for the purpose of collecting data for human language technologies. We survey the papers published in the NAACL-2010 Workshop. 24 researchers participated in the workshop's shared task to create data for speech and language applications with $100.

1 Introduction

This paper gives an overview of the NAACL-2010 Workshop on Creating Speech and Language Data With Amazon's Mechanical Turk. A number of recent papers have evaluated the effectiveness of using Mechanical Turk to create annotated data for natural language processing applications. The low cost, scalable workforce available through Mechanical Turk (MTurk) and other crowdsourcing sites opens new possibilities for annotating speech and text, and has the potential to dramatically change how we create data for human language technologies. Open questions include: What kind of research is possible when the cost of creating annotated training data is dramatically reduced? What new tasks should we try to solve if we do not limit ourselves to reusing existing training and test sets? Can complex annotation be done by untrained annotators? How can we ensure high quality annotations from crowdsourced contributors?

To begin addressing these questions, we organized an open-ended $100 shared task. Researchers were given $100 of credit on Amazon Mechanical Turk to spend on an annotation task of their choosing. They were required to write a short paper describing their experience, and to distribute the data that they created. They were encouraged to address the following questions: How did you convey the task in terms that were simple enough for non-experts to understand? Were non-experts as good as experts? What did you do to ensure quality? How quickly did the data get annotated? What is the cost per label? Researchers submitted a 1 page proposal to the workshop organizers that described their intended experiments and expected outcomes. The organizers selected proposals based on merit, and awarded $100 credits that were generously provided by Amazon Mechanical Turk. In total, 35 credits were awarded to researchers.

Shared task participants were given 10 days to run experiments between the distribution of the credit and the initial submission deadline. 30 papers were submitted to the shared task track, of which 24 were accepted. 14 papers were submitted to the general track of which 10 were accepted, giving a 77% acceptance rate and a total of 34 papers. Shared task participants were required to provide the data collected as part of their experiments. All of the shared task data is available on the workshop website.

2 Mechanical Turk

Amazon's Mechanical Turk[1] is an online marketplace for work. Amazon's tag line for Mechanical Turk is *artificial* artificial intelligence, and the name refers to a historical hoax from the 18th cen-

[1] http://www.mturk.com/

Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk, pages 1–12,
Los Angeles, California, June 2010.

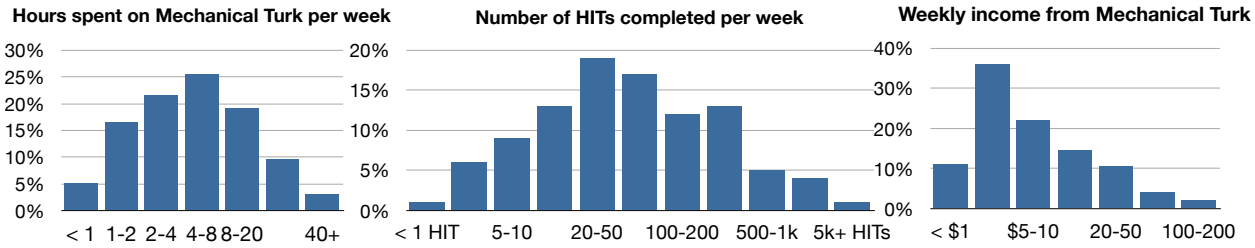

Figure 1: Time spent, HITs completed, and amount earned from a survey of 1,000 Turkers by Ipeirotis (2010).

tury where a chess-playing automaton appeared to be able to beat human opponents using a mechanism, but was, in fact, controlled by a person hiding inside the machine. These hint at the the primary focus of the web service, which is to get people to perform tasks that are simple for humans but difficult for computers. The basic unit of work on MTurk is even called a Human Intelligence Task (HIT).

Amazon's web service provides an easy way to pay people small amounts of money to perform HITs. Anyone with an Amazon account can either submit HITs or work on HITs that were submitted by others. Workers are referred to as "Turkers" and people designing the HITs are called "Requesters." Requesters set the amount that they will pay for each item that is completed. Payments are frequently as low as $0.01. Turkers are free to select whichever HITs interest them.], and to disregard HITs that they find uninteresting or which they deem pay too little.

Because of its focus on tasks requiring human intelligence, Mechanical Turk is obviously applicable to the field of natural language processing. Snow et al. (2008) used Mechanical Turk to inexpensively collect labels for several NLP tasks including word sense disambiguation, word similarity, textual entailment, and temporal ordering of events. Snow et al. had two exciting findings. First, they showed that a strong correlation between non-expert and expert annotators can be achieved by combining the judgments of multiple non-experts, for instance by voting on each label using 10 different Turkers. Correlation and accuracy of labeling could be further improved by weighting each Turker's vote by calibrating them on a small amount of gold standard data created by expert annotators. Second, they collected a staggering number of labels for a very small amount of money. They collected 21,000 labels for just over $25. Turkers put in over 140+ hours worth of human effort to generate the labels. The amount of participation is surprisingly high, given the small payment.

Why do you complete tasks in MTurk?	US	India
To spend free time fruitfully and get cash (e.g., instead of watching TV)	70%	60%
For "primary" income purposes (e.g., gas, bills, groceries, credit cards)	15%	27%
For "secondary" income purposes, pocket change (for hobbies, gadgets)	60%	37%
To kill time	33%	5%
The tasks are fun	40%	20%
Currently unemployed or part time work	30%	27%

Table 1: Motivations for participating on Mechanical Turk from a survey of 1,000 Turkers by Ipeirotis (2010).

Turker demographics

Given the amount of work that can get done for so little, it is natural to ask: who would contribute so much work for so little pay, and why? The answers to these questions are often mysterious because Amazon does not provide any personal information about Turkers (each Turker is identifiable only through a serial number like A23KO2TP7I4KK2). Ipeirotis (2010) elucidates some of the reasons by presenting a demographic analysis of Turkers. He built a profile of 1000 Turkers by posting a survey to MTurk and paying $0.10 for people to answer questions about their reasons for participating on Mechanical Turk, the amount that they earn each week, and how much time they spend, as well as demographic information like country of origin, gender, age, education level, and household income.

One suspicion that people often have when they first hear about MTurk is that it is some sort of digital sweatshop that exploits workers in third world countries. However, Ipeirotis reports that nearly half

(47%) of the Turkers who answered his survey were from the United States, with the next largest group (34%) coming from India, and the remaining 19% spread between 66 other countries.

Table 1 gives the survey results for questions relating to why people participate on Mechanical Turk. It shows that most US-based workers use Mechanical Turk for secondary income purposes (to have spending money for hobbies or going out), but that the overwhelming majority of them use it to spend their time more fruitfully (i.e., instead of watching TV). The economic downturn may have increased participation, with 30% of the US-based Turkers reporting that they are unemployed or underemployed. The public radio show Marketplace recently interviewed unemployed Turkers (Rose, 2010). It reports that they earn a little income, but that they do not earn enough to make a living. Figure 1 confirms this, giving a break down of how much time people spend on Mechanical Turk each week, how many HITs they complete, and how much money they earn. Most Turkers spend less than 8 hours per week on Mechanical Turk, and earn less than $10 per week through the site.

3 Quality Control

Ipeirotis (2010) reports that just over half of Turkers have a college education. Despite being reasonably well educated, it is important to keep in mind that Turkers do not have training in specialized subjects like NLP. Because the Turkers are non-experts, and because the payments are generally so low, quality control is an important consideration when creating data with MTurk.

Amazon provides three mechanisms to help ensure quality:

- Requesters have the option of rejecting the work of individual Turkers, in which case they are not paid.[2] Turkers can also be blocked from doing future work for a requester.
- Requesters can specify that each HIT should be redundantly completed by several different Turkers. This allows higher quality labels to be selected, for instance, by taking the majority label.
- Requesters can require that all workers meet a particular set of qualifications, such as sufficient accuracy on a small test set or a minimum percentage of previously accepted submissions.

Amazon provides two qualifications that a Requester can use by default. These are past HIT Approval Rate and Location. The location qualification allows the Requester to have HITs done only by residents of a certain country (or to exclude Turkers from certain regions). Additionally, Requesters can design custom Qualification Tests that Turkers must complete before working on a particular HIT. These can be created through the MTurk API, and can either be graded manually or automatically. An important qualification that isn't among Amazon's default qualifications is language skills. One might design a qualification test to determine a Turker's ability to speak Arabic or Farsi before allowing them to do part of speech tagging in those languages, for instance.

There are several reasons that poor quality data might be generated. The task may be too complex or the instructions might not be clear enough for Turkers to follow. The financial incentives may be too low for Turkers to act conscientiously, and certain HIT designs may allow them to simply randomly click instead of thinking about the task. Mason and Watts (2009) present a study of financial incentives on Mechanical Turk and find, counterintuitively, that increasing the amount of compensation for a particular task does not tend to improve the quality of the results. Anecdotally, we have observed that sometimes there is an inverse relationship between the amount of payment and the quality of work, because it is more tempting to cheat on high-paying HITs if you don't have the skills to complete them. For example, a number of Turkers tried to cheat on an Urdu to English translation HIT by cutting-and-pasting the Urdu text into an online machine translation system (expressly forbidden in the instructions) because we were paying the comparatively high amount of $1.

[2] Since the results are downloadable even if they are rejected, this could allow unscrupulous Requesters to abuse Turkers by rejecting all of their work, even if it was done well. Turkers have message boards at `http://www.turkernation.com/`, where they discuss Requesters. They even have a Firefox plugin called Turkopticon that lets them see ratings of how good the Requesters are in terms of communicating with Turkers, being generous and fair, and paying promptly.

3.1 Designing HITs for quality control

We suggest designing your HITs in a way that will deter cheating or that will make cheating obvious. HIT design is part of the art of using MTurk. It can't be easily quantified, but it has a large impact on the outcome. For instance, we reduced cheating on our translation HIT by changing the design so that we displayed images of the Urdu sentences instead of text, which made it impossible to copy-and-paste into an MT system for anyone who could not type in Arabic script.

Another suggestion is to include information within the data that you upload to MTurk that will not be displayed to the Turkers, but will be useful to you when reviewing the HITs. For example, we include machine translation output along with the source sentences. Although this is not displayed to Turkers, when we review the Turkers' translations we compare them to the MT output. This allows us to reject translations that are identical to the MT, or which are just random sentences that are unrelated to the original Urdu. We also use a javascript[3] to gather the IP addresses of the Turkers and do geolocation to look up their location. Turkers in Pakistan require less careful scrutiny since they are more likely to be bilingual Urdu speakers than those in Romania, for instance.

CrowdFlower[4] provides an interface for designing HITs that includes a phase for the Requester to input gold standard data with known labels. Inserting items with known labels alongside items which need labels allows a Requester to see which Turkers are correctly replicating the gold standard labels and which are not. This is an excellent idea. If it is possible to include positive and negative controls in your HITs, then do so. Turkers who fail the controls can be blocked and their labels can be excluded from the final data set. CrowdFlower-generated HITs even display a score to the Turkers to give them feedback on how well they are doing. This provides training for Turkers, and discourages cheating.

[3] http://wiki.github.com/callison-burch/mechanical_turk_workshop/geolocation

[4] http://crowdflower.com/

3.2 Iterative improvements on MTurk

Another class of quality control on Mechanical Turk is through iterative HITs that build on the output of previous HITs. This could be used to have Turkers judge whether the results from a previous HIT conformed to the instructions, and whether it is of high quality. Alternately, the second set of Turkers could be used to improve the quality of what the first Turkers created. For instance, in a translation task, a second set of US-based Turkers could edit the English produced by non-native speakers.

CastingWords,[5] a transcription company that uses Turker labor, employs this strategy by having a first-pass transcription graded and iteratively improved in subsequent passes. Little et al. (2009) even designed an API specifically for running iterative tasks on MTurk.[6]

4 Recommended Practices

Although it is hard to define a set of "best practices" that applies to all HITs, or even to all NLP HITs, we recommend the following guidelines to Requesters. First and foremost, it is critical to convey instructions appropriately for non-experts. The instructions should be clear and concise. To calibrate whether the HIT is doable, you should first try the task yourself, and then have a friend from outside the field try it. This will help to ensure that the instructions are clear, and to calibrate how long each HIT will take (which ought to allow you to price the HITs fairly).

If possible, you should insert positive and negative controls so that you can quickly screen out bad Turkers. This is especially important for HITs that only require clicking buttons to complete. If possible, you should include a small amount of gold standard data in each HIT. This will allow you to determine which Turkers are good, but will also allow you weight the Turkers if you are combining the judgments of multiple Turkers. If you are having Turkers evaluate the output of systems, then randomize the order that the systems are shown in.

When publishing papers that use Mechanical Turk as a source of training data or to evaluate the output of an NLP system, report how you ensured the quality of your data. You can do this by measuring the

[5] http://castingwords.com/

[6] http://groups.csail.mit.edu/uid/turkit/

inter-annotator agreement of the Turkers against experts on small amounts of gold standard data, or by stating what controls you used and what criteria you used to block bad Turkers. Finally, whenever possible you should publish the data that you generate on Mechanical Turk (and your analysis scripts and HIT templates) alongside your paper so that other people can verify it.

5 Related work

In the past two years, several papers have published about applying Mechanical Turk to a diverse set of natural language processing tasks, including: creating question-answer sentence pairs (Kaisser and Lowe, 2008), evaluating machine translation quality and crowdsouring translations (Callison-Burch, 2009), paraphrasing noun-noun compouds for SemEval (Butnariu et al., 2009), human evaluation of topic models (Chang et al., 2009), and speech transcription (McGraw et al., 2010; Marge et al., 2010a; Novotney and Callison-Burch, 2010a). Others have used MTurk for novel research directions like non-simulated active learning for NLP tasks such as sentiment classification (Hsueh et al., 2009) or doing quixotic things like doing human-in-the-loop minimum error rate training for machine translation (Zaidan and Callison-Burch, 2009).

Some projects have demonstrated the super-scalability of crowdsourced efforts. Deng et al. (2009) used MTurk to construct ImageNet, an annotated image database containing 3.2 million that are hierarchically categorized using the WordNet ontology (Fellbaum, 1998). Because Mechanical Turk allows researchers to experiment with crowdsourcing by providing small incentives to Turkers, other successful crowdsourcing efforts like Wikipedia or Games with a Purpose (von Ahn and Dabbish, 2008) also share something in common with MTurk.

6 Shared Task

The workshop included a shared task in which participants were provided with $100 to spend on Mechanical Turk experiments. Participants submitted a 1 page proposal in advance describing their intended use of the funds. Selected proposals were provided $100 seed money, to which many participants added their own funds. As part of their participation, each team submitted a workshop paper describing their experiments as well as the data collected and described in the paper. Data for the shared papers is available at the workshop website.[7]

This section describes the variety of data types explored and collected in the shared task. Of the 24 participating teams, most did not exceed the $100 that they were awarded by a significant amount. Therefore, the variety and extent of data described in this section is the result of a minimal $2,400 investment. This achievement demonstrates the potential for MTurk's impact on the creation and curation of speech and language corpora.

6.1 Traditional NLP Tasks

An established core set of computational linguistic tasks have received considerable attention in the natural language processing community. These include knowledge extraction, textual entailment and word sense disambiguation. Each of these tasks requires a large and carefully curated annotated corpus to train and evaluate statistical models. Many of the shared task teams attempted to create new corpora for these tasks at substantially reduced costs using MTurk.

Parent and Eskenazi (2010) produce new corpora for the task of word sense disambiguation. The study used MTurk to create unique word definitions for 50 words, which Turkers then also mapped onto existing definitions. Sentences containing these 50 words were then assigned to unique definitions according to word sense.

Madnani and Boyd-Graber (2010) measured the concept of transitivity of verbs in the style of Hopper and Thompson (1980), a theory that goes beyond simple grammatical transitivity – whether verbs take objects (transitive) or not – to capture the amount of action indicated by a sentence. Videos that portrayed verbs were shown to Turkers who described the actions shown in the video. Additionally, sentences containing the verbs were rated for aspect, affirmation, benefit, harm, kinesis, punctuality, and volition. The authors investigated several approaches for eliciting descriptions of transitivity from Turkers.

Two teams explored textual entailment tasks. Wang and Callison-Burch (2010) created data for

[7] http://sites.google.com/site/amtworkshop2010/

recognizing textual entailment (RTE). They submitted 600 text segments and asked Turkers to identify facts and counter-facts (unsupported facts and contradictions) given the provided text. The resulting collection includes 790 facts and 203 counter-facts. Negri and Mehdad (2010) created a bi-lingual entailment corpus using English and Spanish entailment pairs, where the hypothesis and text come from different languages. The authors took a publicly available English RTE data set (the PASCAL-RTE3 dataset1) and created an English-Spanish equivalent by having Turkers translating the hypotheses into Spanish. The authors include a timeline of their progress, complete with total cost over the 10 days that they ran the experiments.

In the area of natural language generation, Heilman and Smith (2010) explored the potential of MTurk for ranking of computer generated questions about provided texts. These questions can be used to test reading comprehension and understanding. 60 Wikipedia articles were selected, for each of which 20 questions were generated. Turkers provided 5 ratings for each of the 1,200 questions, creating a significant corpus of scored questions.

Finally, Gordon et al. (2010) relied on MTurk to evaluate the quality and accuracy of automatically extracted common sense knowledge (factoids) from news and Wikipedia articles. Factoids were provided by the KNEXT knowledge extraction system.

6.2 Speech and Vision

While MTurk naturally lends itself to text tasks, several teams explored annotation and collection of speech and image data. We note that one of the papers in the main track described tools for collecting such data (Lane et al., 2010).

Two teams used MTurk to collect text annotations on speech data. Marge et al. (2010b) identified easy and hard sections of meeting speech to transcribe and focused data collection on difficult segments. Transcripts were collected on 48 audio clips from 4 different speakers, as well as other types of annotations. Kunath and Weinberger (2010) collected ratings of accented English speech, in which non-native speakers were rated as either Arabic, Mandarin or Russian native speakers. The authors obtained multiple annotations for each speech sample, and tracked the native language of each annotator, allowing for an analysis of rating accuracy between native English and non-native English annotators.

Novotney and Callison-Burch (2010b) used MTurk to elicit new speech samples. As part of an effort to increase the accessibility of public knowledge, such as Wikipedia, the team prompted Turkers to narrate Wikipedia articles. This required Turkers to record audio files and upload them. An additional HIT was used to evaluate the quality of the narrations.

A particularly creative data collection approach asked Turkers to create handwriting samples and then to submit images of their writing (Tong et al., 2010). Turkers were asked to submit handwritten shopping lists (large vocabulary) or weather descriptions (small vocabulary) in either Arabic or Spanish. Subsequent Turkers provided a transcription and a translation. The team collected 18 images per language, 2 transcripts per image and 1 translation per transcript.

6.3 Sentiment, Polarity and Bias

Two papers investigated the topics of sentiment, polarity and bias. Mellebeek et al. (2010) used several methods to obtain polarity scores for Spanish sentences expressing opinions about automative topics. They evaluated three HITs for collecting such data and compared results for quality and expressiveness. Yano et al. (2010) evaluated the political bias of blog posts. Annotators labeled 1000 sentences to determine biased phrases in political blogs from the 2008 election season. Knowledge of the annotators own biases allowed the authors to study how bias differs on the different ends of the political spectrum.

6.4 Information Retrieval

Large scale evaluations requiring significant human labor for evaluation have a long history in the information retrieval community (TREC). Grady and Lease (2010) study four factors that influence Turker performance on a document relevance search task. The authors present some negative results on how these factors influence data collection. For further work on MTurk and information retrieval, readers are encouraged to see the SIGIR 2010 Workshop on Crowdsourcing for Search Evaluation.[8]

[8] http://www.ischool.utexas.edu/~cse2010/call.htm

6.5 Information Extraction

Information extraction (IE) seeks to identify specific types of information in natural languages. The IE papers in the shared tasks focused on new domains and genres as well as new relation types.

The goal of relation extraction is to identify relations between entities or terms in a sentence, such as *born in* or *religion*. Gormley et al. (2010) automatically generate potential relation pairs in sentences by finding relation pairs appearing in news articles as given by a knowledge base. They ask Turkers if a sentence supports a relation, does not support a relation, or whether the relation makes sense. They collected close to 2500 annotations for 17 different person relation types.

The other IE papers explored new genres and domains. Finin et al. (2010) obtained named entity annotations (person, organization, geopolitical entity) for several hundred Twitter messages. They conducted experiments using both MTurk and CrowdFlower. Yetisgen-Yildiz et al. (2010) explored medical named entity recognition. They selected 100 clinical trial announcements from ClinicalTrials.gov. 4 annotators for each of the 100 announcements identified 3 types of medical entities: medical conditions, medications, and laboratory test.

6.6 Machine Translation

The most popular shared task topic was Machine Translation (MT). MT is a data hungry task that relies on huge corpora of parallel texts between two languages. Performance of MT systems depends on the size of training corpora, so there is a constant search for new and larger data sets. Such data sets are traditionally expensive to produce, requiring skilled translators. One of the advantages to MTurk is the diversity of the Turker population, making it an especially attractive source of MT data. Shared task papers in MT explored the full range of MT tasks, including alignments, parallel corpus creation, paraphrases and bilingual lexicons.

Gao and Vogel (2010) create alignments in a 300 sentence Chinese-English corpus (Chinese aligned to English). Both Ambati and Vogel (2010) and Bloodgood and Callison-Burch (2010) explore the potential of MTurk in the creation of MT parallel corpora for evaluation and training. Bloodgood and Callison-Burch replicate the NIST 2009 Urdu-English test set of 1792 sentences, paying only $0.10 a sentence, a substantially reduced price than the typical annotator cost. The result is a data set that is still effective for comparing MT systems in an evaluation. Ambati and Vogel create corpora with 100 sentences and 3 translations per sentence for all the language pairs between English, Spanish, Urdu and Telugu. This demonstrates the feasibility of creating cheap corpora for high and low resource languages.

Two papers focused on the creation and evaluation of paraphrases. Denkowski et al. (2010) generated and evaluated 728 paraphrases for Arabic-English translation. MTurk was used to identify correct and fix incorrect paraphrases. Over 1200 high quality paraphrases were created. Buzek et al. (2010) evaluated error driven paraphrases for MT. In this setting, paraphrases are used to simplify potentially difficult to translate segments of text. Turkers identified 1780 error regions in 1006 English/Chinese sentences. Turkers provided 4821 paraphrases for these regions.

External resources can be an important part of an MT system. Irvine and Klementiev (2010) created lexicons for low resource languages. They evaluated translation candidates for 100 English words in 32 languages and solicited translations for 10 additional languages. Higgins et al. (2010) expanded name lists in Arabic by soliciting common Arabic nicknames. The 332 collected nicknames were primarily provided by Turkers in Arab speaking countries (35%), India (46%), and the United States (13%).

Finally, Zaidan and Ganitkevitch (2010) explored how MTurk could be used to directly improve an MT grammar. Each rule in an Urdu to English translation system was characterized by 12 features. Turkers were provided examples for which their feedback was used to rescore grammar productions directly. This approach shows the potential of fine tuning an MT system with targeted feedback from annotators.

7 Future Directions

Looking ahead, we can't help but wonder what impact MTurk and crowdsourcing will have on the speech and language research community. Keeping in mind Niels Bohr's famous exhortation "Pre-

diction is very difficult, especially if it's about the future," we attempt to draw some conclusions and predict future directions and impact on the field.

Some have predicted that access to low cost, highly scalable methods for creating language and speech annotations means the end of work on unsupervised learning. Many a researcher has advocated his or her unsupervised learning approach because of annotation costs. However, if 100 examples for any task are obtainable for less than $100, why spend the time and effort developing often inferior unsupervised methods? Such a radical change is highly debatable, in fact, one of this paper's authors is a strong advocate of such a position while the other disagrees, perhaps because he himself works on unsupervised methods. Certainly, we can agree that the potential exists for a change in focus in a number of ways.

In natural language processing, data drives research. The introduction of new large and widely accessible data sets creates whole new areas of research. There are many examples of such impact, the most famous of which is the Penn Treebank (Marcus. et al., 1994), which has 2910 citations in Google scholar and is the single most cited paper on the ACL anthology network (Radev et al., 2009). Other examples include the CoNLL named entity corpus (Sang and Meulder (2003) with 348 citations on Google Scholar), the IMDB movie reviews sentiment data (Pang et al. (2002) with 894 citations) and the Amazon sentiment multi-domain data (Blitzer et al. (2007) with 109 citations) . MTurk means that creating similar data sets is now much cheaper and easier than ever before. It is highly likely that new MTurk produced data sets will achieve prominence and have significant impact. Additionally, the creation of shared data means more comparison and evaluation against previous work. Progress is made when it can be demonstrated against previous approaches on the same data. The reduction of data cost and the rise of independent corpus producers likely means more accessible data.

More than a new source for *cheap* data, MTurk is a source for *new types* of data. Several of the papers in this workshop collected information about the annotators in addition to their annotations. This creates potential for studying how different user demographics understand language and allow for targeting specific demographics in data creation. Beyond efficiencies in cost, MTurk provides access to a global user population far more diverse than those provided by more professional annotation settings. This will have a significant impact on low resource languages as corpora can be cheaply built for a much wider array of languages. As one example, Irvine and Klementiev (2010) collected data for 42 languages without worrying about how to find speakers of such a wide variety of languages. Additionally, the collection of Arabic nicknames requires a diverse and numerous Arabic speaking population (Higgins et al., 2010). In addition to extending into new languages, MTurk also allows for the creation of evaluation sets in new genres and domains, which was the focus of two papers in this workshop (Finin et al., 2010; Yetisgen-Yildiz et al., 2010). We expect to see new research emphasis on low resource languages and new domains and genres.

Another factor is the change of data type and its impact on machine learning algorithms. With professional annotators, great time and care are paid to annotation guidelines and annotator training. These are difficult tasks with MTurk, which favors simple intuitive annotations and little training. Many papers applied creative methods of using simpler annotation tasks to create more complex data sets. This process can impact machine learning in a number of ways. Rather than a single gold standard, annotations are now available for many users. Learning across multiple annotations may improve systems (Dredze et al., 2009). Additionally, even with efforts to clean up MTurk annotations, we can expect an increase in noisy examples in data. This will push for new more robust learning algorithms that are less sensitive to noise. If we increase the size of the data ten-fold but also increase the noise, can learning still be successful? Another learning area of great interest is active learning, which has long relied on simulated user experiments. New work evaluated active learning methods with real users using MTurk (Baker et al., 2009; Ambati et al., 2010; Hsueh et al., 2009; **?**). Finally, the composition of complex data set annotations from simple user inputs can transform the method by which we learn complex outputs. Current approaches expect examples of labels that exactly match the expectation of the system. Can we instead provide lower level sim-

pler user annotations and teach systems how to learn from these to construct complex output? This would open more complex annotation tasks to MTurk.

A general trend in research is that good ideas come from unexpected places. Major transformations in the field have come from creative new approaches. Consider the Penn Treebank, an ambitious and difficult project of unknown potential. Such large changes can be uncommon since they are often associated with high cost, as was the Penn Treebank. However, MTurk greatly reduces these costs, encouraging researchers to try creative new tasks. For example, in this workshop Tong et al. (2010) collected handwriting samples in multiple languages. Their creative data collection may or may not have a significant impact, but it is unlikely that it would have been tried had the cost been very high.

Finally, while obtaining new data annotations from MTurk is cheap, it is not trivial. Workshop participants struggled with how to attract Turkers, how to price HITs, HIT design, instructions, cheating detection, etc. No doubt that as work progresses, so will a communal knowledge and experience of how to use MTurk. There can be great benefit in new toolkits for collecting language data using MTurk, and indeed some of these have already started to emerge (Lane et al., 2010)[9].

Acknowledgements

Thanks to Sharon Chiarella of Amazon's Mechanical Turk for providing $100 credits for the shared task, and to CrowdFlower for allowing free use of their tool to workshop participants.

Research funding was provided by the NSF under grant IIS-0713448, by the European Commission through the EuroMatrixPlus project, and by the DARPA GALE program under Contract No. HR0011-06-2-0001. The views and findings are the authors' alone.

References

Cem Akkaya, Alexander Conrad, Janyce Wiebe, and Rada Mihalcea. 2010. Amazon Mechanical Turk for subjectivity word sense disambiguation. In *NAACL Workshop on Creating Speech and Language Data With Amazon's Mechanical Turk*.

Vamshi Ambati and Stephan Vogel. 2010. Can crowds build parallel corpora for machine translation systems? In *NAACL Workshop on Creating Speech and Language Data With Amazon's Mechanical Turk*.

Vamshi Ambati, Stephan Vogel, and Jamie Carbonell. 2010. Active learning and crowd-sourcing for machine translation. *Language Resources and Evaluation (LREC)*.

Kathy Baker, Steven Bethard, Michael Bloodgood, Ralf Brown, Chris Callison-Burch, Glen Coppersmith, Bonnie Dorr, Wes Filardo, Kendall Giles, Ann Irvine, Mike Kayser, Lori Levin, Justin Martineau, Jim Mayfield, Scott Miller, Aaron Phillips, Andrew Philpot, Christine Piatko, Lane Schwartz, and David Zajic. 2009. Semantically-informed machine translation. Technical Report 002, Johns Hopkins Human Language Technology Center of Excellence, Summer Camp for Applied Language Exploration, Johns Hopkins University, Baltimore, MD.

John Blitzer, Mark Dredze, and Fernando Pereira. 2007. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Association for Computational Linguistics (ACL)*.

Michael Bloodgood and Chris Callison-Burch. 2010a. Bucking the trend: Large-scale cost-focused active learning for statistical machine translation. In *48th Annual Meeting of the Association for Computational Linguistics*, Uppsala, Sweden.

Michael Bloodgood and Chris Callison-Burch. 2010b. Using Mechanical Turk to build machine translation evaluation sets. In *NAACL Workshop on Creating Speech and Language Data With Amazon's Mechanical Turk*.

Cristina Butnariu, Su Nam Kim, Preslav Nakov, Diarmuid Ó Séaghdha, Stan Szpakowicz, and Tony Veale. 2009. Semeval-2010 Task 9: The interpretation of noun compounds using paraphrasing verbs and prepositions. In *Workshop on Semantic Evaluations*.

Olivia Buzek, Philip Resnik, and Ben Bederson. 2010. Error driven paraphrase annotation using Mechanical Turk. In *NAACL Workshop on Creating Speech and Language Data With Amazon's Mechanical Turk*.

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 (EMNLP-2009)*, Singapore.

Jonathan Chang, Jordan Boyd-Graber, Chong Wang, Sean Gerrish, and David M. Blei. 2009. Reading tea leaves: How humans interpret topic models. In *Neural Information Processing Systems*.

[9] http://wiki.github.com/callison-burch/mechanical_turk_workshop/

Jonathan Chang. 2010. Not-so-Latent Dirichlet Allocation: Collapsed Gibbs sampling using human judgments. In *NAACL Workshop on Creating Speech and Language Data With Amazon's Mechanical Turk*.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR*, Miami Beach, Floriday.

Michael Denkowski and Alon Lavie. 2010. Exploring normalization techniques for human judgments of machine translation adequacy collected using Amazon Mechanical Turk. In *NAACL Workshop on Creating Speech and Language Data With Amazon's Mechanical Turk*.

Michael Denkowski, Hassan Al-Haj, and Alon Lavie. 2010. Turker-assisted paraphrasing for English-Arabic machine translation. In *NAACL Workshop on Creating Speech and Language Data With Amazon's Mechanical Turk*.

Mark Dredze, Partha Pratim Talukdar, and Koby Crammer. 2009. Sequence learning from data with multiple labels. In *ECML/PKDD Workshop on Learning from Multi-Label Data (MLD)*.

Keelan Evanini, Derrick Higgins, and Klaus Zechner. 2010. Using Amazon Mechanical Turk for transcription of non-native speech. In *NAACL Workshop on Creating Speech and Language Data With Amazon's Mechanical Turk*.

Christiane Fellbaum. 1998. *WordNet: An Electronic Lexical Database*. Bradford Books.

Tim Finin, William Murnane, Anand Karandikar, Nicholas Keller, Justin Martineau, and Mark Dredze. 2010. Touring PER, ORG and LOC on $100 a day. In *NAACL Workshop on Creating Speech and Language Data With Amazon's Mechanical Turk*.

Qin Gao and Stephan Vogel. 2010. Semi-supervised word alignment with Mechanical Turk. In *NAACL Workshop on Creating Speech and Language Data With Amazon's Mechanical Turk*.

Dan Gillick and Yang Liu. 2010. Non-expert evaluation of summarization systems is risky. In *NAACL Workshop on Creating Speech and Language Data With Amazon's Mechanical Turk*.

Jonathan Gordon, Benjamin Van Durme, and Lenhart Schubert. 2010. Evaluation of commonsense knowledge with Mechanical Turk. In *NAACL Workshop on Creating Speech and Language Data With Amazon's Mechanical Turk*.

Matthew R. Gormley, Adam Gerber, Mary Harper, and Mark Dredze. 2010. Non-expert correction of automatically generated relation annotations. In *NAACL Workshop on Creating Speech and Language Data With Amazon's Mechanical Turk*.

Catherine Grady and Matthew Lease. 2010. Crowdsourcing document relevance assessment with Mechanical Turk. In *NAACL Workshop on Creating Speech and Language Data With Amazon's Mechanical Turk*.

Michael Heilman and Noah A. Smith. 2010. Rating computer-generated questions with Mechanical Turk. In *NAACL Workshop on Creating Speech and Language Data With Amazon's Mechanical Turk*.

Chiara Higgins, Elizabeth McGrath, and Laila Moretto. 2010. AMT crowdsourcing: A viable method for rapid discovery of Arabic nicknames? In *NAACL Workshop on Creating Speech and Language Data With Amazon's Mechanical Turk*.

Paul J. Hopper and Sandra A. Thompson. 1980. Transitivity in grammar and discourse. *Language*, 56:251–299.

Pei-Yun Hsueh, Prem Melville, and Vikas Sindhwani. 2009. Data quality from crowdsourcing: A study of annotation selection criteria. In *Proceedings of the NAACL HLT 2009 Workshop on Active Learning for Natural Language Processing*, pages 27–35, Boulder, Colorado, June. Association for Computational Linguistics.

Panos Ipeirotis. 2010. New demographics of Mechanical Turk. `http://behind-the-enemy-lines.blogspot.com/2010/03/new-demographics-of-mechanical-turk.html`.

Ann Irvine and Alexandre Klementiev. 2010. Using Mechanical Turk to annotate lexicons for less commonly used languages. In *NAACL Workshop on Creating Speech and Language Data With Amazon's Mechanical Turk*.

Mukund Jha, Jacob Andreas, Kapil Thadani, Sara Rosenthal, and Kathleen McKeown. 2010. Corpus creation for new genres: A crowdsourced approach to PP attachment. In *NAACL Workshop on Creating Speech and Language Data With Amazon's Mechanical Turk*.

Michael Kaisser and John Lowe. 2008. Creating a research collection of question answer sentence pairs with Amazons Mechanical Turk. In *Proceedings of the Sixth International Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco.

Stephen Kunath and Steven Weinberger. 2010. The wisdom of the crowd's ear: Speech accent rating and annotation with Amazon Mechanical Turk. In *NAACL Workshop on Creating Speech and Language Data With Amazon's Mechanical Turk*.

Ian Lane, Matthias Eck, Kay Rottmann, and Alex Waibel. 2010. Tools for collecting speech corpora via Mechanical-Turk. In *NAACL Workshop on Creating Speech and Language Data With Amazon's Mechanical Turk*.

Nolan Lawson, Kevin Eustice, Mike Perkowitz, and Meliha Yetisgen Yildiz. 2010. Annotating large email datasets for named entity recognition with Mechanical Turk. In *NAACL Workshop on Creating Speech and Language Data With Amazon's Mechanical Turk*.

Greg Little, Lydia B. Chilton, Rob Miller, and Max Goldman. 2009. Turkit: Tools for iterative tasks on mechanical turk. In *Proceedings of the Workshop on Human Computation at the International Conference on Knowledge Discovery and Data Mining (KDD-HCOMP '09)*, Paris.

Nitin Madnani and Jordan Boyd-Graber. 2010. Measuring transitivity using untrained annotators. In *NAACL Workshop on Creating Speech and Language Data With Amazon's Mechanical Turk*.

Mitch Marcus., Beatrice Santorini, and Mary Ann Marcinkiewicz. 1994. Building a large annotated corpus of english: The penn treebank. *Computational linguistics*, 19(2):313–330.

Matthew Marge, Satanjeev Banerjee, and Alexander Rudnicky. 2010a. Using the Amazon Mechanical Turk for transcription of spoken language. *ICASSP*, March.

Matthew Marge, Satanjeev Banerjee, and Alexander Rudnicky. 2010b. Using the Amazon Mechanical Turk to transcribe and annotate meeting speech for extractive summarization. In *NAACL Workshop on Creating Speech and Language Data With Amazon's Mechanical Turk*.

Winter Mason and Duncan J. Watts. 2009. Financial incentives and the "performance of crowds". In *Proceedings of the Workshop on Human Computation at the International Conference on Knowledge Discovery and Data Mining (KDD-HCOMP '09)*, Paris.

Ian McGraw, Chia ying Lee, Lee Hetherington, and Jim Glass. 2010. Collecting voices from the crowd. *LREC*, May.

Bart Mellebeek, Francesc Benavent, Jens Grivolla, Joan Codina, Marta R. Costa-Jussà, and Rafael Banchs. 2010. Opinion mining of spanish customer comments with non-expert annotations on Mechanical Turk. In *NAACL Workshop on Creating Speech and Language Data With Amazon's Mechanical Turk*.

Robert Munro, Steven Bethard, Victor Kuperman, Vicky Tzuyin Lai, Robin Melnick, Christopher Potts, Tyler Schnoebelen, and Harry Tily. 2010. Crowdsourcing and language studies: the new generation of linguistic data. In *NAACL Workshop on Creating Speech and Language Data With Amazon's Mechanical Turk*.

Matteo Negri and Yashar Mehdad. 2010. Creating a bi-lingual entailment corpus through translations with Mechanical Turk: $100 for a 10 days rush. In *NAACL Workshop on Creating Speech and Language Data With Amazon's Mechanical Turk*.

Scott Novotney and Chris Callison-Burch. 2010a. Cheap, fast and good enough: Automatic speech recognition with non-expert transcription. *NAACL*, June.

Scott Novotney and Chris Callison-Burch. 2010b. Crowdsourced accessibility: Elicitation of Wikipedia articles. In *NAACL Workshop on Creating Speech and Language Data With Amazon's Mechanical Turk*.

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up?: sentiment classification using machine learning techniques. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Gabriel Parent and Maxine Eskenazi. 2010. Clustering dictionary definitions using Amazon Mechanical Turk. In *NAACL Workshop on Creating Speech and Language Data With Amazon's Mechanical Turk*.

Dragomir R. Radev, Pradeep Muthukrishnan, and Vahed Qazvinian. 2009. The acl anthology network. In *Proceedings of the 2009 Workshop on Text and Citation Analysis for Scholarly Digital Libraries*, pages 54–61, Suntec City, Singapore, August. Association for Computational Linguistics.

Cyrus Rashtchian, Peter Young, Micah Hodosh, and Julia Hockenmaier. 2010. Collecting image annotations using Amazon's Mechanical Turk. In *NAACL Workshop on Creating Speech and Language Data With Amazon's Mechanical Turk*.

Joel Rose. 2010. Some turn to 'Mechanical' job search. `http://marketplace.publicradio.org/display/web/2009/06/30/pm_turking/`. Marketplace public radio. Air date: June 30, 2009.

Erik Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *CoNLL-2003*.

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 2008 Conference on Empirical Methods in Natural Language Processing (EMNLP-2008)*, Honolulu, Hawaii.

Audrey Tong, Jerome Ajot, Mark Przybocki, and Stephanie Strassel. 2010. Document image collection using Amazon's Mechanical Turk. In *NAACL Workshop on Creating Speech and Language Data With Amazon's Mechanical Turk*.

Luis von Ahn and Laura Dabbish. 2008. General techniques for designing games with a purpose. *Communications of the ACM*.

Rui Wang and Chris Callison-Burch. 2010. Cheap facts and counter-facts. In *NAACL Workshop on Creating*

Speech and Language Data With Amazon's Mechanical Turk.

Tae Yano, Philip Resnik, and Noah A Smith. 2010. Shedding (a thousand points of) light on biased language. In *NAACL Workshop on Creating Speech and Language Data With Amazon's Mechanical Turk*.

Meliha Yetisgen-Yildiz, Imre Solti, Scott Halgrim, and Fei Xia. 2010. Preliminary experiments with Amazon's Mechanical Turk for annotating medical named entities. In *NAACL Workshop on Creating Speech and Language Data With Amazon's Mechanical Turk*.

Omar F. Zaidan and Chris Callison-Burch. 2009. Feasibility of human-in-the-loop minimum error rate training. In *Proceedings of EMNLP 2009*, pages 52–61, August.

Omar Zaidan and Juri Ganitkevitch. 2010. An enriched MT grammar for under $100. In *NAACL Workshop on Creating Speech and Language Data With Amazon's Mechanical Turk*.