Building a Persistent Workforce on Mechanical Turk for Multilingual Data Collection

David L. Chen
Department of Computer Science
The University of Texas at Austin
Austin, TX 78712, USA
dlcc@cs.utexas.edu

William B. Dolan
Microsoft Research
One Microsoft Way
Redmond, WA 98052, USA
billdol@microsoft.com

Abstract

Traditional methods of collecting translation and paraphrase data are prohibitively expensive, making the construction of large, new corpora difficult. While crowdsourcing offers a cheap alternative, quality control and scalability can become problematic. We discuss a novel annotation task that uses videos as the stimulus which discourages cheating. In addition, our approach requires only *monolingual* speakers, thus making it easier to scale since more workers are qualified to contribute. Finally, we employ a multi-tiered payment system that helps retain good workers over the long-term, resulting in a persistent, high-quality workforce. We present the results of one of the largest linguistic data collection efforts to date using Mechanical Turk, yielding 85K English sentences and more than 1k sentences for each of a dozen more languages.

1 Introduction

Much of the recent progress in machine translation can be attributed to the availability of large bilingual corpora. However, existing resources are limited in their domain coverage, mostly centering around government or business needs. Moreover, there is a heavy bias toward corpora containing English as one of the pair of languages. The problem is worse for machine paraphrasing, where parallel data is even more scarce, leading researchers to focus on methods leveraging more abundant bilingual data as training data. Thus, collecting new data resources is important for advancing the states of both of these fields.

Traditional methods for building such corpora are expensive and time-consuming. While popular crowdsourcing platforms such as Amazon's Mechanical Turk seem promising as a cheaper alternative for data collection, the nature of the translation and paraphrasing tasks make them difficult to pose to a large, random Internet crowd. To collect translation data, bilingual speakers in the desired language-pair are required. This is especially problematic for low-resource language-pairs. Collecting paraphrase data presents a different problem in that the annotators are invariably biased by the lexical items and word order of the original sentence. Consequently, creating natural, creative paraphrases can be a challenge. In this paper we expand on our discussion (Chen and Dolan 2011) of a novel data collection framework that addresses all these problems by using videos as the stimulus to create both translation and paraphrase data.

Instead of presenting annotators with a sentence to translate or paraphrase, we show them a short video segment and ask them to describe the video in one sentence. These video segments were carefully chosen so they clearly show a single, unambiguous action or event (e.g. a man riding a horse, a chef slicing an onion, etc.) Descriptions of the same video segment can then be used as translation data if they are in different languages and as paraphrase data if they are in the same language.

In addition to the novel data collection method, we also employed a tiered-payment system to facilitate large-scale data collection. By actively engaging in communications with the workers and setting up a system that rewards consistent, high-quality submissions, we were able to build a persistent workforce that is able to produce good annotations in large volumes. While our system is more time-consuming to set up initially, it is more efficient in the long run as good workers require little or no supervision. They may also eventually be trained to evaluate and manage new workers, thus allowing the system to scale.

With a budget of $5,000 spent over a two-month period, we collected over 122K sentences in total with 85K of them in English and the rest in more than a dozen different languages. The data has been made available to the research community as the Microsoft Research Video Description Corpus[1]. In the rest of the paper we describe the details of our data collection process and discuss the lessons we have learned from one of the largest linguistic data collection efforts to date using Mechanical Turk.

2 Related Work

Given the low-cost of crowdsourcing, there have been several attempts to collect translation and paraphrase data using Mechanical Turk. Callison-Burch (2009) conducted a pilot study translating 50 sentences into several languages. Continuing this effort, larger-scale data collections were later done to recreate reference sentences for the NIST 2009 Urdu-English test set (Bloodgood and Callison-Burch 2010;

[1] Available for download at http://research.microsoft.com/en-us/downloads/38cf15fd-b8df-477e-a4e4-a4680caa75af/

Zaidan and Callison-Burch 2011). Expanding the scale in a different direction, Irvine and Klementiev (2010) constructed lexicons for 42 rare languages, testing the wide range of language expertise available on Mechanical Turk.

However, given the incentive to maximize their rewards, workers who are unqualified often cheat on these translation tasks, whether by using online translation services or by collaborating to defeat consensus filtering methods (Ambati and Vogel 2010). Methods for defeating such cheating and improving the quality of the data include posting text as images to prevent copying and pasting to online translation services, and asking other workers to edit the original translations and to rank the different translation candidates (Zaidan and Callison-Burch 2011). We circumvent these issues by using videos as the stimulus and allowing workers to type in the language of their choice.

There has also been some work on collecting paraphrase data using Mechanical Turk. Buzek et al. (2010) automatically identified problem regions in a translation task and had workers attempt to paraphrase them. Denkowski et al. (2010) used a different approach, asking workers to assess the validity of automatically extracted paraphrases. Our work is distinct from these earlier efforts both in terms of the task – attempting to collect linguistic descriptions using a visual stimulus – and the dramatically larger scale of the data collected.

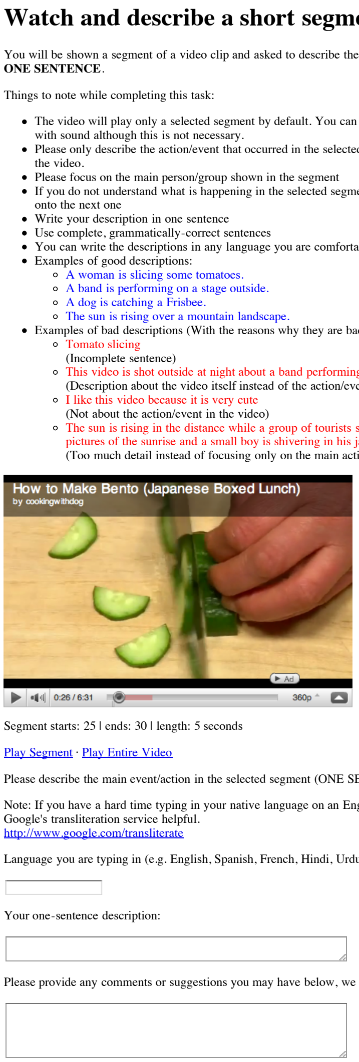
One of the attractive features of our framework is that we only require *monolingual* speakers to gather translation data. Bederson et al. (2010) utilized monolingual speakers in a different way by asking them to identify and paraphrase problem regions of automatically translated outputs. By iteratively paraphrasing and retranslating using a machine translation engine, the final output is ideally more coherent.

We use a multi-tiered payment system similar in spirt to the one used by Novotney and Callison-Burch (2010). They asked workers to first submit sample narrations of Wikipedia articles and only allowed those who qualified to complete the full narration tasks.

Finally, many of the lessons we learned are similar to those reported by Kochhar et al. in describing their human computation engine RABJ (Kochhar, Mazzocchi, and Paritosh 2010). They employ a hierarchical system that systematically promotes contracted judges to higher status and more difficult tasks. They also reported the benefits of long-standing relationships with their workers: less quality control required, ability to train the workers to improve task-specific competence, and valuable feedback from the workers to shape and improve task designs.

3 Data Collection Framework

Since our goal was to collect large numbers of translations and paraphrases quickly and inexpensively using a crowd, our framework was designed to make the tasks short, simple, easy, accessible and somewhat fun. For each task, we asked the annotators to watch a very short video clip (usually less than 10 seconds long) and describe in one sentence the main action or event that occurred. By using videos as the stimulus we avoided the need for bilingual speakers for collecting translation data. We also avoided linguistic biases caused by the source sentence in collecting paraphrase data.



Watch and describe a short segment of a video

You will be shown a segment of a video clip and asked to describe the main action/event in that segment in **ONE SENTENCE**.

Things to note while completing this task:

- The video will play only a selected segment by default. You can choose to watch the entire clip and/or with sound although this is not necessary.
- Please only describe the action/event that occurred in the selected segment and not any other parts of the video.
- Please focus on the main person/group shown in the segment
- If you do not understand what is happening in the selected segment, please skip this HIT and move onto the next one
- Write your description in one sentence
- Use complete, grammatically-correct sentences
- You can write the descriptions in any language you are comfortable with
- Examples of good descriptions:
  - A woman is slicing some tomatoes.
  - A band is performing on a stage outside.
  - A dog is catching a Frisbee.
  - The sun is rising over a mountain landscape.
- Examples of bad descriptions (With the reasons why they are bad in parentheses):
  - Tomato slicing
    (Incomplete sentence)
  - This video is shot outside at night about a band performing on a stage
    (Description about the video itself instead of the action/event in the video)
  - I like this video because it is very cute
    (Not about the action/event in the video)
  - The sun is rising in the distance while a group of tourists standing near some railings are taking pictures of the sunrise and a small boy is shivering in his jacket because it is really cold
    (Too much detail instead of focusing only on the main action/event)

Segment starts: 25 | ends: 30 | length: 5 seconds

Play Segment · Play Entire Video

Please describe the main event/action in the selected segment (ONE SENTENCE):

Note: If you have a hard time typing in your native language on an English keyboard, you may find Google's transliteration service helpful.
http://www.google.com/transliterate

Language you are typing in (e.g. English, Spanish, French, Hindi, Urdu, Mandarin Chinese, etc):

Your one-sentence description:

Please provide any comments or suggestions you may have below, we appreciate your input!

Figure 1: A screenshot of our annotation task as it was deployed on Mechanical Turk.

We deployed the task on Amazon's Mechanical Turk, with video segments selected from YouTube. A screenshot of our annotation task is shown in Figure 1. On average, annotators completed each task within 80 seconds, including time watching the videos. Experienced annotators were even faster, completing the task in only 20 to 25 seconds.

One interesting aspect of this framework is that each annotator approaches the task from a linguistically independent perspective, unbiased by the lexical or word order choices in a pre-existing description. This is similar in spirit to the 'Pear Stories' film, which was designed to tap into our universal experience independent of language (Chafe 1997). An important aspect of our approach is that it allows us to gather arbitrarily many of these independent descriptions for each video, capturing nearly-exhaustive coverage of how native speakers are likely to summarize a small action. It might be possible to achieve similar effects using images or panels of images as the stimulus (von Ahn and Dabbish 2004; Fei-Fei et al. 2007; Rashtchian et al. 2010), but we believed that videos would be more engaging and less ambiguous in their focus. In addition, videos have been shown to be more effective in prompting descriptions of motion and contact verbs, as well as verbs that are generally not imageable (Ma and Cook 2009).

3.1 Quality Control

One of the main problems with collecting data using a crowd is quality control. While the cost is very low compared to traditional annotation methods, workers recruited over the Internet are often unqualified for the tasks or are incentivized to cheat in order to maximize their rewards.

By using videos to collect translation data, we removed the possibility of using online translation services to cheat. Moreover, to encourage native and fluent contributions, we asked annotators to write the descriptions in the language of their choice. Since there are no benefits for writing in one language over another, this discourages workers from using a language they are not proficient in.

To ensure the quality of the annotations, we used a 2-tiered payment system to reward workers who submit good descriptions and work on our tasks consistently. While everyone had access to the Tier-1 tasks, only workers who had been manually qualified could work on the Tier-2 tasks. The tasks were identical in the two tiers except each Tier-1 task only paid 1 cent while each Tier-2 task paid 5 cents, giving the workers a strong incentive to earn the qualification.

The qualification process was performed manually by the authors. We periodically evaluated the workers who had submitted the most Tier-1 tasks (usually on the order of few hundred submissions) and granted them access to the Tier-2 tasks if they had performed well. We assessed their work mainly on the grammaticality and spelling accuracy of the submitted descriptions. Since we had hundreds of submissions on which to base our decisions, it was fairly quick and easy to identify cheaters and people with poor English skills. Workers who were rejected during this process were still allowed to work on the Tier-1 tasks. For submissions in languages other than English, we used online translation services (if available) to ensure that they were not submitting random sentences, but could not verify the quality of those sentences. In the end, we granted everyone who submitted non-English descriptions access to the Tier-2 tasks, partly to encourage more submissions in different languages. Future data collection would ideally have access to at least one native speaker who could verify the quality of the submissions.

Using the tiered payment system allowed us to pay the good workers higher wages to retain them without wasting money on potentially poor workers. An alternative method would be to inspect each description individually and reject the poor ones. But this quickly becomes infeasible as the number of annotations grows. While our approach requires more manual effort initially than some other methods such as using a qualification test or automatic post-annotation filtering, it creates a much higher quality workforce that needs little or no supervision. It also evaluates the workers on their actual task competence and willingness to work on our tasks. The initial effort is amortized over time as these quality workers continued to come back and many of them annotated all the available videos we had posted.

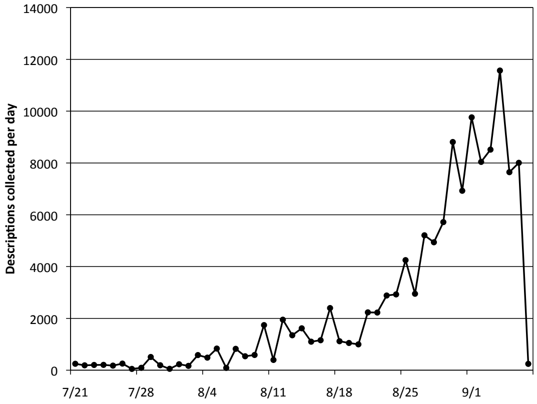


Figure 2: Number of descriptions collected per day.

3.2 Video Collection

To find suitable videos to annotate, we deployed a separate task. Workers were asked to submit short (generally 4-10 seconds) video segments depicting single, unambiguous events by specifying links to YouTube videos, along with the start and end times.

Given the goal of gathering descriptions in many languages, we aimed to collect videos that could be understood regardless of linguistic or cultural background. In order to avoid biasing lexical choices in the descriptions, we muted the audio and excluded videos that contained either subtitles or overlaid text. Finally, we manually filtered the submitted videos to ensure that each met our criteria and was free of inappropriate content before posting them for annotations.

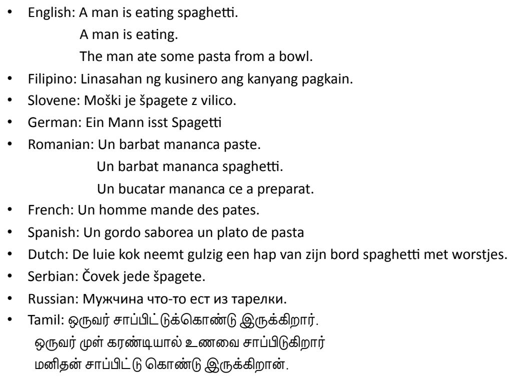
We used a 3-tiered payment system to reward and retain workers who performed well. While we paid workers for submitting any valid videos, promotions to higher tiers, with an associated increase in pay, were based on the percentage of their submissions that met our standards.

4 Data

We deployed our data collection framework on Mechanical Turk over a two-month period from July to September in 2010, collecting 2,089 video segments and 122K descriptions. Figure 2 shows the number of descriptions accepted per day during this period. The rate of data collection accelerated as we built up our workforce, topping 10K descriptions a day when we ended our data collection. Given the trajectory and the fact that data collection was often limited by how fast we could review the videos and post them, we believe an even higher annotation throughput is feasible.

English	85550	Spanish	1883
Hindi	6245	Gujarati	1437
Romanian	3998	Russian	1243
Slovene	3584	French	1226
Serbian	3420	Italian	953
Tamil	2789	Georgian	907
Dutch	2735	Polish	544
German	2326	Chinese	494
Macedonian	1915	Malayalam	394

Table 1: The number of annotations obtained for each language. Other languages that yielded at least one annotation included: Tagalog, Portuguese, Norwegian, Filipino, Estonian, Turkish, Arabic, Urdu, Hungarian, Indonesian, Malay, Bulgarian, Danish, Bosnian, Marathi, Swedish, and Albanian.



- English: A man is eating spaghetti.
  A man is eating.
  The man ate some pasta from a bowl.
- Filipino: Linasahan ng kusinero ang kanyang pagkain.
- Slovene: Moški je špagete z vilico.
- German: Ein Mann isst Spagetti
- Romanian: Un barbat mananca paste.
  Un barbat mananca spaghetti.
  Un bucatar mananca ce a preparat.
- French: Un homme mande des pates.
- Spanish: Un gordo saborea un plato de pasta
- Dutch: De luie kok neemt gulzig een hap van zijn bord spaghetti met worstjes.
- Serbian: Čovek jede špagete.
- Russian: Мужчина что-то ест из тарелки.
- Tamil: ஒருவர் சாப்பிட்டுக்கொண்டு இருக்கிறார்.
  ஒருவர் முள் கரண்டியால் உணவை சாப்பிடுகிறார்
  மனிதன் சாப்பிட்டு கொண்டு இருக்கிறான்.

Figure 3: Examples of descriptions collected for a particular video.

While most of the descriptions we collected were in English, there were also significant volumes of data for other languages. Table 1 shows the top languages for which we received descriptions. Examples of some of the descriptions collected are shown in Figure 3. Of the 85,550 English descriptions accepted, 33,855 were from Tier-2 tasks, meaning they were provided by workers who had been manually identified as good performers. A detailed breakdown of the statistics for Tier-1 and Tier-2 tasks is shown in Table 2[2].

Overall, 835 workers submitted at least one description with only 15 that did not have any descriptions accepted. Table 3 shows the number of descriptions contributed by each worker. The distribution follows the power law, with a small portion of the workers contributing most of the data, some annotating all the available videos we had. The largest number of descriptions submitted by a single worker was 3624[3]. Of the 835 workers, 94 were granted access to the Tier-2 tasks. The success of our data collection effort was in part due to our ability to retain these good workers, building a reliable and efficient workforce. Overall, we spent under $5,000 for the entire data collection effort, including Amazon's 10% service fees, some pilot experiments and surveys.

	Tier 1	Tier 2
pay	$0.01	$0.05
# videos	2089	2029
# workers (English)	683	50
# workers (total)	835	94
# submitted (English)	51510	33829
# submitted (total)	68578	55682
# accepted (English)	51052	33825
# accepted (total)	67968	55658

Table 2: Statistics for the two video description tasks

# descriptions	# workers
1-10	408
11-100	255
101-500	84
501-1000	35
1001-2000	30
2001-3624	8

Table 3: Number of descriptions contributed by each worker

5 Workforce

To better understand the workers who contributed significant portion of our data, we conducted a survey of all the Tier-2 workers. Out of 94 workers, 46 responded to the survey.

Table 4 shows some basic demographic information about these workers. There were roughly equal numbers of male and female workers. The workers mostly ranged from young adults to middle-aged men and women, with the average age being 34 years old. They come from many different countries, generally corresponding to the different languages used to describe the videos. Most of them are from the United States, with India being a distant second. The wide distribution of geographical areas is a positive sign, showing both that there exists a large international crowd on Mechanical Turk and that our data is likely contributed by native speakers. This is promising for future collections of translation or any type of multilingual data on Mechanical Turk.

To build a reliable workforce, we need workers who can consistently work on our tasks. Table 5 shows the reported average number of hours in a week the surveyed workers work on Mechanical Turk. A bit surprisingly, some workers spend a large amount of time doing tasks on Mechanical Turk, treating it almost like a part-time job. Some even spend the equivalent time of a full-time job doing these tasks. This shows that there exists a dedicated population on Mechanical Turk who are willing to work long hours on certain tasks.

[2] The numbers for the English data are slightly underestimated since workers sometimes incorrectly filled out the form when reporting what language they were using.

[3] This number exceeds the total number of videos because the worker completed both Tier-1 and Tier-2 tasks for the same videos

Sex			
Male	24	Female	22

Age	
18-25	13
26-35	13
36-45	12
46-55	6
56 and above	2

Country			
United States	21	Colombia	1
India	6	Lithuania	1
Romania	2	Austria	1
Philippines	2	Brazil	1
Slovenia	1	Canada	1
Serbia	1	Italy	1
Holland	1	Georgia	1
Germany	1	Poland	1
Macedonia	1	Norway	1
Mexico	1		

Table 4: Demographic information about the Tier-2 workers who responded to our survey.

Hours spent per week	# workers
1-10	20
11-19	6
20-39	15
40 and above	3

Table 5: Reported numbers of hours working on Mechanical Turk in an average week.

This is useful for tasks that require some training. Instead of getting a random crowd to perform the task, it is possible to recruit dedicated workers to continuously work on the task until they are proficient.

The long hours spent by some of these workers suggest that their motivations are beyond simple curiosity or boredom. Table 6 lists some of the reasons these workers choose to spend their time working on Mechanical Turk. While most of them treat these tasks as a rewarding experience in and of itself or as a way to get some extra income, there also exists a large number of workers who depend on this income for living expenses such as paying bills or buying groceries. This provides further evidence for our observation that some workers are essentially treating Mechanical Turk tasks as a part-time or even a full-time job.

Reason to work	# workers
I depend on this income for living (e.g. pay bills, buy groceries, etc)	17 (37%)
I use this income as extra spending money (e.g. support hobbies)	32 (70%)
I enjoy doing the tasks on Mechanical Turk	29 (63%)
I do the tasks to pass the time	15 (33%)

Table 6: Reported reasons for working on Mechanical Turk (multiple reasons allowed)

6 Discussion and Future Work

Having established the existence of serious Mechanical Turk workers who could form the basis of a persistent, high-quality workforce, we need to learn how to attract them and retain them over time. Based on our experience conducting this data collection and comments submitted by the Tier-2 workers through their surveys, we organized the lessons we learned into the following principles.

Design the tasks well As stated in Section 3, we aimed to make the tasks short, simple, easy, accessible, and fun. Several workers commented that our task was easy for them because they could write the descriptions in their native language. Others enjoyed watching short video clips and found them to be entertaining. Many people also commented that the task was quick to complete, which kept them engaged. While it is not possible to design every human computation task to satisfy all these criteria, spending some time to test-run the tasks could greatly improve the worker experience. For example, improving the layout of the task could reduce the amount of scrolling required, and writing clear instructions makes it easier for the workers to understand exactly what is being asked for.

Learn from worker responses Regardless of the amount of effort spent in designing the tasks, unforeseen problems are bound to arise. For example, some workers suggested that we should cache the language they use so they do not have to type it repeatedly. This would also avoid typos when inputting that information, eliminating the need for manual cleanup later. Another problem that we faced was that some YouTube videos were restricted in certain countries, or they were removed after we posted the tasks. A way to check the availability of videos at annotation time would have prevented workers from submitting empty descriptions due to their inability to watch the videos.

Compensate the workers fairly As one worker commented, fair rewards result in worker loyalty, which is vital in retaining good workers. Part of the equation of determining proper pay has to do with the time required to perform the task. A task that can be completed quickly will require less payment. Thus, improving the design of the task could also save money if workers need less time to complete them. Our multi-tiered system allowed us to pay good workers fairly, while eliminating the need to waste money on potentially poor work.

Communicate One often-overlooked aspect of the human computation process is the need to actually communicate with the workers. Many workers cited our quick task approval as the reason they chose to work on our task. Timely responses gave them confidence that they would get paid for their work. Providing feedback in the form of bonuses, or rejections along with an explanation, are useful in training the workers to improve their work. Responding to workers' comments or emails also makes them feel more engaged and willing to overcome any initial difficulties they might have in doing the tasks. Finally, making sure the tasks are available on a regular basis for each worker gives them incentive to check back often.

While we have conducted one of the largest linguistic data collection efforts to date on Mechanical Turk, the amount of data collected is still very small compared to standard translation datasets. In order to train broad-domain paraphrase or machine translation engines, we would need to extend our data collection to a much larger scale.

One of the bottlenecks in our data collection framework involves finding appropriate video segments. While crowdsourcing the video searching task greatly improved the rate of video collection, manually filtering the submitted videos remained a time-consuming process. One possible solution might be to promote the best workers at finding appropriate videos to perform the filtering task.

In general, we could extend our hierarchical system in many directions, promoting our best workers to higher tiers where they can perform other administrative tasks such as training new workers or approving new Tier-2 workers. Given enough time to build up our hierarchy, we could crowdsource the entire process, thus speeding up the rate of both video and description collections.

The data we collected is unique in its high parallelism in many different languages. This characteristic allows our data to be particularly useful as an evaluation dataset for machine paraphrasing (Chen and Dolan 2011). The large number of parallel sentences capture a wide space of semantically-equivalent sentences that vary lexically and syntactically. Other uses may include training and testing machine translation systems with large number of reference sentences and building computer vision systems that can generate natural language descriptions of video content.

7 Conclusion

Traditional methods for collecting translation and paraphrase data require a highly skilled workforce and are extremely expensive. Even with the rise in popularity of crowdsourcing methods for data collection, it remains difficult to create quality translation data as the number of qualified workers is limited and the incentive to cheat is high. By using videos to elicit linguistic annotations, we avoid the need for bilingual annotators to create translation data. Moreover, our method removes any incentive to cheat by asking workers to annotate in the language of their choice. With a budget of $5000, we were able to collect over 122K annotations in less than 2 months. Part of our success in collecting this data is due to our ability to train and retain good workers over the long-term. We used a tiered-payment system to properly compensate good workers. We also responded to workers quickly, engaging them and giving them confidence they would get paid. These features made our tasks attractive to serious workers who take pride in their work and are willing to work on tasks for long hours.

References

Ambati, V., and Vogel, S. 2010. Can crowds build parallel corpora for machine translation systems? In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*.

Bederson, B. B.; Hu, C.; and Resnik, P. 2010. Translation by iterative collaboration between monolingual users. In *Proceedings of the 36th Graphics Interface conference(GI-2010)*.

Bloodgood, M., and Callison-Burch, C. 2010. Using Mechanical Turk to build machine translation evaluation sets. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*.

Buzek, O.; Resnik, P.; and Bederson, B. B. 2010. Error driven paraphrase annotation using Mechanical Turk. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*.

Callison-Burch, C. 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)*.

Chafe, W. L. 1997. *The Pear Stories: Cognitive, Cultural and Linguistic Aspects of Narrative Production*. Norwood, NJ: Ablex.

Chen, D. L., and Dolan, W. B. 2011. Collecting highly parallel data for paraphrase evaluation. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL-2011)*.

Denkowski, M.; Al-Haj, H.; and Lavie, A. 2010. Turker-assisted paraphrasing for English-Arabic machine translation. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*.

Fei-Fei, L.; Iyer, A.; Koch, C.; and Perona, P. 2007. What do we perceive in a glance of a real-world scene? *Journal of Vision* 7(1):1–29.

Irvine, A., and Klementiev, A. 2010. Using Mechanical Turk to annotate lexicons for less commonly used languages. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*.

Kochhar, S.; Mazzocchi, S.; and Paritosh, P. 2010. The anatomy of a large-scale human computation engine. In *Proceedings of the 2nd Human Computation Workshop (KDD-HCOMP-10)*.

Ma, X., and Cook, P. R. 2009. How well do visual verbs work in daily communication for young and old adults. In *Proceedings of ACM CHI 2009 Conference on Human Factors in Computing Systems*.

Novotney, S., and Callison-Burch, C. 2010. Crowdsourced accessibility elicitation of wikipedia articles. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*.

Rashtchian, C.; Young, P.; Hodosh, M.; and Hockenmaier, J. 2010. Collecting image annotations using Amazon's Mechanical Turk. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*.

von Ahn, L., and Dabbish, L. 2004. Labeling images with a computer game. In *Proceedings of ACM CHI 2004 Conference on Human Factors in Computing Systems*.

Zaidan, O. F., and Callison-Burch, C. 2011. Crowdsourcing translation: Professional quality from non-professionals. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL-2011)*.