Shared Task: Crowdsourced Accessibility Elicitation of Wikipedia Articles

Scott Novotney and **Chris Callison-Burch**
Center for Language and Speech Processing
Johns Hopkins University
3400 North Charles Street
Baltimore, MD, USA
snovotne@bbn.com ccb@jhu.edu

Abstract

Mechanical Turk is useful for generating complex speech resources like conversational speech transcription. In this work, we explore the next step of eliciting narrations of Wikipedia articles to improve accessibility for low-literacy users. This task proves a useful test-bed to implement qualitative vetting of workers based on difficult to define metrics like narrative quality. Working with the Mechanical Turk API, we collected sample narrations, had other Turkers rate these samples and then granted access to full narration HITs depending on aggregate quality. While narrating full articles proved too onerous a task to be viable, using other Turkers to perform vetting was very successful. Elicitation is possible on Mechanical Turk, but it should conform to suggested best practices of simple tasks that can be completed in a streamlined workflow.

1 Introduction

The rise of Mechanical Turk publications in the NLP community leaves no doubt that non-experts can provide useful annotations for low cost. Emerging best practices suggest designing short, simple tasks that require little amount of upfront effort to most effectively use Mechanical Turk's labor pool. Suitable tasks are best limited to those easily accomplished in 'short bites' requiring little context switching. For instance, most annotation tasks in prior work (Snow et al., 2008) required selection from an enumerated list, allowing for easy automated quality control and data collection.

More recent work to collect speech transcription (Novotney and Callison-Burch, 2010) or parallel text translations (Callison-Burch, 2009) demonstrated that Turkers can provide useful free-form annotation.

In this paper, we extend open ended collection even further by eliciting narrations of English Wikipedia articles. To vet prospective narrators, we use *qualitative* qualifications by aggregating the opinions of other Turkers on narrative style, thus avoiding quantification of qualitative tasks.

The Spoken Wikipedia Project[1] aims to increase the accessibility of Wikipedia by recording articles for use by blind or illiterate users. Since 2008, over 1600 English articles covering topics from art to technology have been narrated by volunteers. The charitable nature of this work should provide additional incentive for Turkers to complete this task. We use Wikipedia narrations as an initial proof-of-concept for other more challenging elicitation tasks such as spontaneous or conversational speech.

While previous work used other Turkers in second-pass filtering for quality control, we flip this process and instead require that narrators be judged favorably before working on full narration tasks. Relying on human opinion sidesteps the difficult task of automatically judging narrative quality. This requires a multi-pass workflow to manage potential narrators and grant them access to the full narration HITs through Mechanical Turk's Qualifications.

In this paper, we make the following points:

- Vetting based on qualitative criteria like narration quality can be effectively implemented through Turker-provided ratings.

[1] http://en.wikipedia.org/wiki/Wikipedia:WikiProject_Spoken_Wikipedia

Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk, pages 41–44,
Los Angeles, California, June 2010.

- Narrating full articles is too complex and time-consuming for timely task throughput - best practices are worth following.
- HITs should be streamlined as much as possible. Requiring Turkers to perform work outside of the web interface seemingly hurt task completion rate.

2 Prior Work

The research community has demonstrated that complex annotations (like speech transcription and elicitation) can be provided through Mechanical Turk.

Callison-Burch (2009) showed that Turkers could accomplish complex tasks like translating Urdu or creating reading comprehension tests.

McGraw et al. (2009) used Mechanical Turk to improve an English isolated word speech recognizer by having Turkers listen to a word and select from a list of probable words at a cost of $20 per hour of transcription.

Marge et al. (2010) collected transcriptions of clean speech and demonstrated that duplicate transcription of non-experts can match expert transcription.

Novotney and Callison-Burch (2010) collected transcriptions of conversational speech for as little as $5 / hour of transcription and demonstrated that resources are better spent annotating more data than improving data quality.

McGraw et al. (2010) elicited short snippets of English street addresses through a web interface. 103 hours were elicted in just over three days.

3 Narration Task

Using a python library for parsing Wikipedia[2], we extracted all text under the `<p>` tag as a heuristic for readable content. We ignored all other content like lists, info boxes or headings. Since we wanted to preserve narrative flow, each article was posted as one HIT, paying $0.05 per paragraph. Articles averaged 40 paragraphs, so each HIT averaged $2 in payment - some as little as $0.25.

We provided instructions for using recording software and asked Turkers to record one paragraph at a time. Using Mechanical Turk's API, we generated an XML template for each paragraph and let the Turker upload a file through the `FileUploadAnswer` form. The API supports constraints on file extensions, so we were able to require that all files be in mp3 format before the Turker could submit the work.

Mechanical Turk's API supports file requests through the `GetFileUploadURL` call. A URL is dynamically generated on Amazon's servers which stays active for one minute. We then fetched each audio file and stored them locally on our own servers for later processing.

Since these narrations are meant for public consumption and are difficult to quality control, we required prospective Turkers first qualify.

4 Granting Qualitative Qualifications

Qualifications are prerequisites that limit which Turkers can work on a HIT. A common qualification provided by Mechanical Turk is a minimum approval rating for a Turker, indicating what percentage of submitted work was approved. We created a qualification for our narration tasks since we wanted to ensure only those turkers with a good speaking voice would complete our tasks.

However, the definition of a "good speaking voice" is not easy to quantify. Luckily, this task is well suited to Mechanical Turk's concept of *artificial* artificial intelligence. Humans can easily decide a narrator's quality while automatic methods would be impractical. Additionally, we never define what a 'good' narration voice is, relying instead on public opinion.

4.1 Workflow

We implemented the qualification ratings using the API with three different steps. Turkers who wish to complete the full narration HITs are first directed to a 'qualification' HIT with one sample paragraph paying $0.05. We then use other Turkers to rate the quality of the narrator, asking them to judge based on speaking style, audio clarity and pronunciation.

Post Qualification The narration qualification and full narration HITs are posted.

Sample HIT A prospective narrator uploads a recording of a sample paragraph earning $0.05.

[2] `http://github.com/j2labs/wikipydia`

The audio is downloaded and hosted on our web host.

Rating HIT A HIT is created to be completed ten times. Turkers make a binary decision as to whether they would listen to a full article by the narrator and optionally suggest feedback.

Grant Qualification The ten ratings are collected and if five or more are positive we grant the qualification. The narrator is then automatically contacted with the decision and provided with any feedback from the rating Turkers.

Although not straightforward, the API made it possible to dynamically create HITs, approve assignments, sync audio files and ratings,notify workers and grant qualifications. It does not, however, manage state across HITs, requiring us to implement our own control logic for associating workers with narration and rating HITs. Once implemented, managing the process was as simple as invoking three perl scripts a few times a day. These could easily be rolled into one background process automatically controlling the entire workflow.

4.2 Effectiveness of Turker Ratings

Thirteen Turkers submitted sample audio files over the course of a week. Collecting the ten ratings took a few hours per Turker. The average rating for the narrators was 7.5, with three of the thirteen being rejected for having a score less than 5. The authors agreed with the sentiment of the raters and feel that the qualification process correctly filtered out the poor narrators.

Below is a sample of the comments for an approved narrator and a rejected narrator.

This Turker was approved with 9/10 votes.

- The narration was very easy to understand. The speaker's tone was even, well-paced, and clear. Great narration.
- Very good voice, good pace and modulation.
- Very nice voice and pleasant to listen to. I would have guessed that this was a professional voice actor.

This Turker was rejected with 3/10 votes.

- Monotone voice, uninterested and barely literate. I would never listen to this voice for any length of time.
- muddy audio quality; narrator has a tired and a very low tone quality.
- Very solemn voice - didn't like listening to it.

5 Data Analysis

Of the thirteen qualified Turkers, only two went on to complete full narrations. This happened only after we shortened the articles to the initial five paragraphs and raised payment to $0.25 per paragraph. While the audio was clear, both authors exhibited mispronunciations of domain-specific terms. For instance, one author narrating Isaac Newton mispronounced *Principia* with a soft *c* (/prɪn**s**ɪpiə/) instead of a hard *c* (/prɪn**k**ɪpiə/) and *indices* as /ɪndə͡ɪsɛz/. Since the text is known ahead of time, one could include a pronunciation guide for rare words to assist the narrator.

The more disapointing result, however, is the very slow return of the narration task. Contrasting with the successful elicitation of (McGraw et al., 2010), two reasons clearly stand out.

First, these tasks were much too long in length. This was due to constraints we placed on collection to improve data quality. We assumed that multiple narrators for a single article would ruin the narrative flow. Since few workers were willing to complete five recordings, future work could chop each article into smaller chunks to be completed by multiple narrators. In contrast, eliciting spoken addresses has no need for continuity across samples, thus the individual HITs in (McGraw et al., 2010) could be much smaller.

Second, and more importantly, our HITs required much more effort on the part of the Turker. We chose to fully use Mechanical Turk's API to manage data and did not implement audio recording or data transmission through the browser. Turkers were required to record audio in a separate program and then upload the files. We thought the added ability to re-record and review audio would be a plus compared to in-browser recording. In contrast, (McGraw et al., 2010) used a javascript package to record narrations directly in the browser window. While it was simple to use the API, it raised too much of a barrier for Turkers to complete the task.

5.1 Feasability for Full Narration

Regardless of the task effectiveness, it is not clear that Mechanical Turk is cost effective for large scale narration. A reasonable first task would be to narrate the 2500 featured articles on Wikipedia's home page. They average 44 paragraphs in length with around 4311 words per article. Narrating this corpus would cost $5500 at the rate of $0.05 per paragraph - if workers would be willing to complete at that rate.

6 Conclusion

Our experiments with Mechanical Turk attempted to find the limits of data collection and nebulous task definitions. Long-form narration was unsuccessful due to the length of the tasks and the lack of a streamlined workflow for the Turkers. However, assigning qualifications based upon aggregating qualitative opinions was very successful. This task exploited the strenghts of Mechanical Turk by quickly gathering judgements that are easy for humans to make but near impossible to reliably automate.

The contrast between the failure of this narration task and the success of previous elicitation is due to the nature of the underlying task. Our desire to have one narrator per article prevented elicitation in short bites of a few seconds long. Additionally, our efforts to solely use Mechanical Turk's API limited the simplicity of the workflow. While our backend work was greatly simplified since we relied on existing data management code, the lack of in-browser recording placed too much burden on the Turkers.

We would make the following changes if we were to reimplement this task:

1. Integrate the workflow into the browser.

2. Perform post-process quality control to block bad narrators from completing more HITs.

3. Drop the requirement of one narrator per article. A successful compromise might be one section, averaging around five paragraphs.

4. Only narrate the lead in to an article (first pargagraph) first. If a user requests a full narration, then seek out the rest of the article.

5. Place qualification as a much larger set of assignments. Turkers often sort HITs by available assignments, so the qualification HIT was rarely seen.

References

Chris Callison-Burch. 2009. Fast, Cheap, and Creative: Evaluating Translation Quality Using Amazons Mechanical Turk. *EMNLP*.

Matthew Marge, Satanjeev Banerjee, and Alexander Rudnicky. 2010. Using the amazon mechanical turk for transcription of spoken language. *ICASSP*, March.

Ian McGraw, Alexander Gruenstein, and Andrew Sutherland. 2009. A self-labeling speech corpus: Collecting spoken words with an online educational game. In *INTERSPEECH*.

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

Scott Novotney and Chris Callison-Burch. 2010. Cheap, Fast and Good Enough: Automatic Speech Recognition with Non-Expert Transcription . *NAACL*, June.

Rion Snow, Brendan O'Connor, Daniel Jurafsky, and Andrew Y. Ng. 2008. Cheap and fast—but is it good?: evaluating non-expert annotations for natural language tasks. In *EMNLP*.