Online Quality Control for Real-Time Crowd Captioning

Walter S. Lasecki and Jeffrey P. Bigham
ROC HCI, Department of Computer Science
University of Rochester
Rochester, NY 14618 USA
{wlasecki, jbigham}@cs.rochester.edu

ABSTRACT

Approaches for real-time captioning of speech are either expensive (professional stenographers) or error-prone (automatic speech recognition). As an alternative approach, we have been exploring whether groups of non-experts can collectively caption speech in real-time. In this approach, each worker types as much as they can and the partial captions are merged together in real-time automatically. This approach works best when partial captions are correct and received within a few seconds of when they were spoken, but these assumptions break down when engaging workers on-demand from existing sources of crowd work like Amazon's Mechanical Turk. In this paper, we present methods for quickly identifying workers who are producing good partial captions and estimating the quality of their input. We evaluate these methods in experiments run on Mechanical Turk in which a total of 42 workers captioned 20 minutes of audio. The methods introduced in this paper were able to raise overall accuracy from 57.8% to 81.22% while keeping coverage of the ground truth signal nearly unchanged.

Categories and Subject Descriptors

H.5.2 [**Information Interfaces and Presentation**]: User Interfaces; K.4.2 [**Social Issues**]: Assistive technologies for persons with disabilities

General Terms

Human Factors, Experimentation

Keywords

captioning, human computation, deaf, hard of hearing

1. INTRODUCTION

Real-time captioning converts aural speech to visual text to provide access to speech content for deaf and hard of hearing (DHH) people in classrooms, meetings, casual conversation, and other live events. These systems need to operate with low latency (generally under 5 seconds) so that DHH users can appropriately place the captions in context [19]. Current options are severely limited because they either require highly-skilled professional captionists whose services are expensive and not available on demand, or use automatic speech recognition (ASR) which produces unacceptable error rates in many real situations [19]. To address this problem, we previously introduced Legion:Scribe [13], a system that allows groups of non-experts to collaboratively caption audio in real time.

The main idea of Legion:Scribe is that while each non-expert worker will not be able to keep up with natural speaking rates (like a professional captionist could), they can type part of what they hear. Legion:Scribe uses new natural language processing techniques inspired by Multiple Sequence

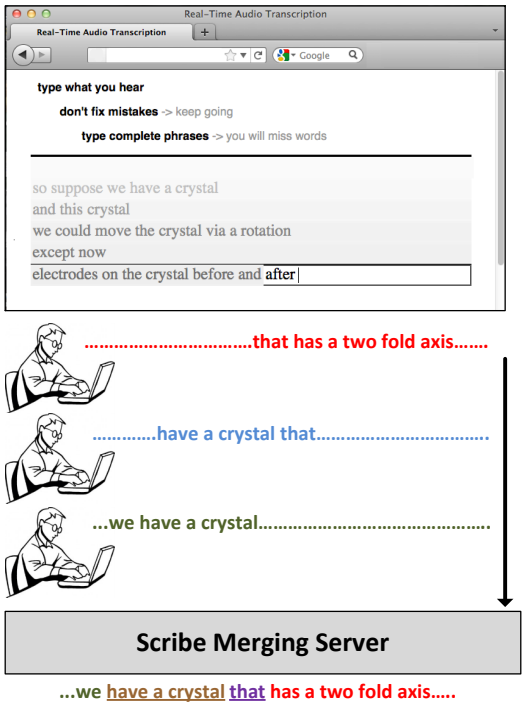

Figure 1: The idea behind Legion:Scribe is that multiple non-expert workers type as much as they can of speech that they hear in real-time, and the system merges it together into a final output stream. This paper considers how to use agreement between the input of different workers to filter this input before attempting to merge it together.

ASSETS'12, October 22–24, 2012, Boulder, Colorado, USA.

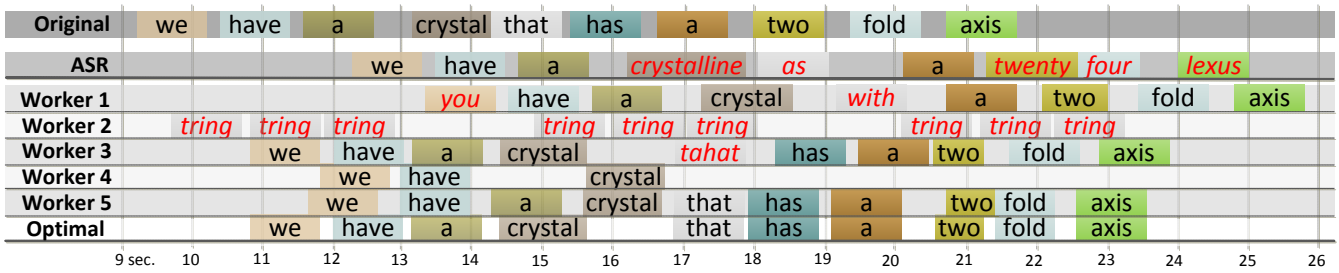

Figure 2: Example transcriptions provided by workers from Amazon's Mechanical Turk, illustrating some of the errors that we observe. Worker 2 misunderstood the task directives and typed a description of aural alerts given by the interface. Workers 1 and 3 provide mostly good input that contains some errors. The rest of the work input is accurate.

Alignment (MSA) to then stitch the partial captions back together into a single stream. Its worker interface encourages different workers to cover different portions of the input speech by systematically adjusting their saliency. Because non-expert workers form the base of Legion:Scribe, it has the potential to be cheaper and more readily available than a professional captionist, while maintaining the advantages of human captionists.

The current approach is not sufficiently robust to errant input from the crowd, whether it be accidental, malicious, or simply a typo. In prior experiments, locally-recruited workers of known quality were shown to be able to enter partial captions with high enough accuracy that Legion:Scribe could merge them back together without errors substantially impacting the final stream. However, to fully leverage the elastic, on-demand nature of the crowd, we would like to be able to recruit workers and volunteers without the need to vet them prior to their participation. This paper considers how to deal with the errors that will inevitably arise when using unvetted workers.

In this paper, we present techniques to dynamically rate workers and the input they produce in an online fashion. The key insight is that workers whose input matches that of other workers is more likely to be correct. By estimating quality in an online (real-time) way, our methods may be able to run seamlessly within Legion:Scribe to provide real-time captions more reliably.

The contributions of this paper are the following:

- We introduce the problem of online quality estimation in real-time crowd captioning.
- We introduce methods for both worker quality estimation and word quality estimation that use overlap between workers.
- We demonstrate the efficacy of these methods in an experiment with 42 Amazon Mechanical Turk workers.

1.1 Quality Metrics

The most common method for determining the quality of a caption is the word error rate (WER), which performs a best-fit alignment between the test captions and the ground truth captions. The WER is then calculated as the sum of the number of substitutions (S), deletions (D), and insertions (I) divided by the total number of words in the ground truth caption (N), or $\frac{S+D+I}{N}$. However, determining the quality of captioning is difficult [20]. A key advantage of human captionists over ASR is that humans tend to make more reasonable errors. Humans infer meaning from context, influencing their prior probability toward those words that make sense in context. We anticipate this will make Legion:Scribe more usable than automated systems even when traditional metrics are similar. Figure 2 shows an example of the errors often made by ASR - in that case substituting, "twenty four lexus" for, "two-fold axis". Such problems with relying solely on WER have been noted before [20].

We define two additional metrics that can be automatically calculated to help characterize the performance of real-time captioning. The first is *coverage*, which represents how many of the words in the true speech signal appear in the union of the partial captions that are received. While similar to recall in information retrieval, we use coverage in this work because we alter the definition of recall slightly in calculating it by requiring that a word in the test signal appear no later than 10 seconds after the word in the ground truth signal to count. We define *precision* similarly, as the fraction of words in the test caption stream that appear in the ground truth within a 10 second time window. Compared to WER, coverage and precision are looser measures of alignment, but we believe they are particularly useful in understanding the potential of our approaches.

In terms of latency, in order for DHH individuals to participate in a conversation or in a lecture, it is important that captions are provided quickly (within about 5 seconds [19]). Calculating latency is not straightforward because captions being tested are not the same as the ground truth. In this paper, we measure latency by first aligning the test captions to the ground truth, and then averaging the latency of all matched words. We also investigate the causes of latency due to the worker (as opposed to network connection or processing delay).

2. RELATED WORK

Crowd captioning builds from work in (i) real-time captioning, and (ii) real-time human computation.

2.1 Real-Time Captioning Systems

Automated speech recognition (ASR) attempts to provide text-to-speech services without human intervention. Currently, ASR is capable of working well in ideal situations with high-quality audio signal and equipment, but degrades quickly in many real settings since it is speaker-dependent, has difficulty recognizing domain-specific jargon, and adapts poorly to vocal changes, such as when the speaker has is sick [10, 9]. ASR systems require substantial computing power and special audio equipment to work well, which lowers availability. To account for these causes of error *respeaking* was created. In this approach, a person located in a

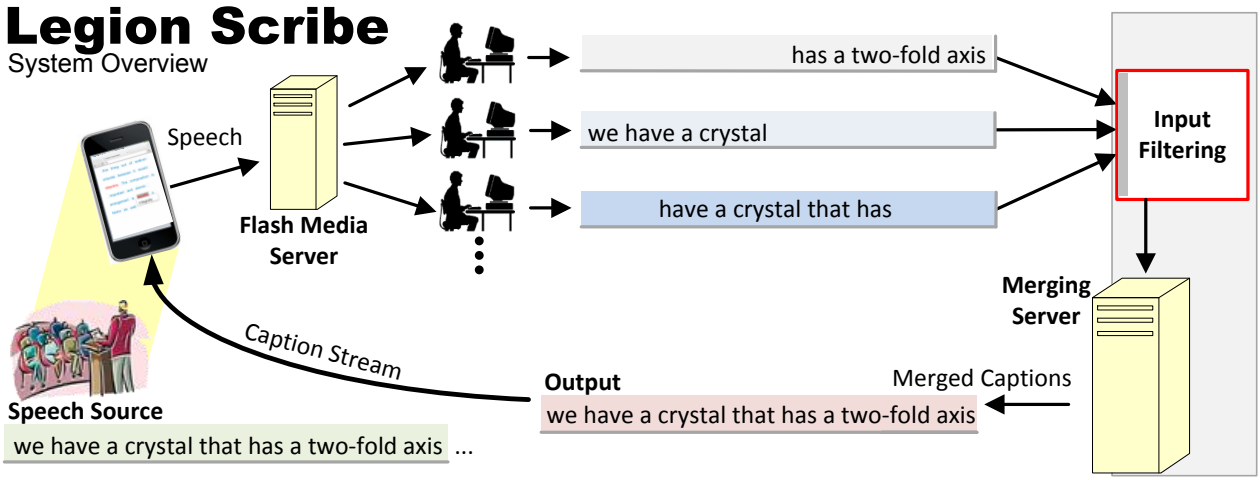

Figure 3: Legion:Scribe allows users to caption audio on their mobile device. The audio is sent to multiple non-expert captionists in realtime who use our web-based interface to caption as much of the audio as they can. These partial captions are sent to the merging server to be merged into a final output caption stream, which is then forwarded back to the user's mobile device. This paper considers the "Input Filtering" stage above, which uses agreement between workers to estimate the quality of both workers and their inputs.

controlled environment is connected to a live audio feed and repeats what they hear to an ASR that has been extensively trained for their voice [11]. Respeaking works well for offline transcription, but simultaneously speaking and listening requires professional training. Crowd captioning aims to allow people without special training to help generate transcripts.

Communications Access Real-Time Translation (CART) is the most reliable captioning service, but is also the most expensive. Professional stenographers type in shorthand on a "steno" keyboard that maps multiple key presses to phonemes that are then automatically expanded to verbatim text. Stenography typically requires 2-3 years of training to achieve the 225 words per minute (WPM) needed to consistently caption speech at natural speaking rates.

Non-verbatim systems attempt to reduce the cost of professional captioning systems such as CART by using macro expansion of customizable abbreviations. For example, C-Print captionists need less training, and generally charge around $60 an hour [19]. However, they normally cannot type as fast as the average speaker's pace of 150 WPM, and thus cannot produce a verbatim transcript. Crowd captioning employs captionists with no training and compensates for slower typing speeds and lower accuracy by combining the efforts of multiple individuals.

2.2 Real-Time Human Computation

People with disabilities have long solved accessibility problems with the support of people in their community [6]. Increasing connectivity has made remote services possible that once required human supporters to be co-located. Real-time captioning by non-experts leverages human computation [17], which has been shown to be useful in many areas, including writing and editing [4], image description and interpretation [5, 18], and protein folding [8]. Existing abstractions obtain quality work by introducing redundancy and layering into tasks so that multiple workers contribute and verify results at each stage [15, 12]. For instance, the ESP Game uses answer agreement [18] and Soylent uses the multiple-step find-fix-verify pattern [4]. Because these approaches take time, they are well suited for real-time support. Crowd captioning enables real-time transcriptions from multiple non-experts to be used to find crowd agreement as a means of ensuring quality.

Human computation has been applied to offline transcription with great success [2]. Scribe4Me allowed deaf and hard of hearing people to receive a transcript of a short sound sequence in a few minutes, but was not able to produce verbatim captions over long periods [16].

Real-time human computation has recently started to be explored by systems such as VizWiz [5], which was one of the first to target nearly real-time responses from the crowd. It introduced a queuing model to help ensure that workers were available quickly on-demand. For Crowd captioning to be available on-demand requires multiple users to be available at the same time so that multiple workers can collectively contribute. Prior systems have shown that multiple workers can be recruited for collaboration by having workers wait until enough workers have arrived [18, 7]. Adrenaline combines the concepts of queuing and waiting to recruit crowds (groups) in less than 2 seconds from existing sources of crowd workers [3]. Real-time captioning by non-experts similarly uses the input of multiple workers, but differs because it engages workers for longer continuous tasks.

Legion enables real-time control of an existing user interface by allowing the crowd to collectively act as a single operator [14]. Each crowd workers submits input independently of other workers, then the system uses an *input mediator* to combine the input into a single control stream. Our input combination approach can be viewed as an instance of an input mediator. A primary difference is that while Legion was shown effective using a mediator in which the crowd's input was used to elect a representative leader to be given direct control for small periods of time, we use a synthesis of the crowd's input to create the final stream.

3. LEGION SCRIBE

Legion:Scribe is a system that provides users with on-demand access to real-time captions from groups of non-experts from their laptop or mobile devices (Figure 3). When the Legion:Scribe app is started, it immediately begins recruiting workers from a set of volunteer workers using quikTurkit [5]. Previous experiments have shown that Mechanical Turk workers can provide useful input in terms of coverage, but the signal was too noisy to use reliably, do to the high number of low-quality workers [13]. When users are ready to begin captioning they press the start button, which then begins forwarding audio to Flash Media Server (FMS) and signals the Legion:Scribe server to begin captioning. We use FFMPEG to stream audio from the user to FMS using the RTMP protocol for real-time audio streaming.

Once connected, workers are presented with a text input interface designed to encourage real-time answers and designed to encourage global coverage (shown in Figure 4). Legion:Scribe rewards workers with points that can optionally correspond to money depending on the crowd. In our experiments, we paid workers $0.005 for every word the system thought was correct. This interface is discussed further in the next section.

As workers type, their input is forwarded to an input combiner running on the Legion:Scribe server. The input combiner is discussed in the next section and is modular to accommodate different implementations without needing to modify the rest of the Legion:Scribe system. Once the inputs have been merged, we present users with the current transcrtip on a dynamically updating web page.

Merging partial captions allows for either an emphasis on coverage or accuracy. However, these two properties are at odds: using more of the worker input will increase coverage, but maintain more of the individual worker error, while requiring more agreement on individual words will increase accuracy, but reduce the coverage since not all workers will agree on all words. Legion:Scribe allows users to either let the system choose a default balance between the two, or select their own balance using a slider bar in the that allows them to select from values that range from 'Most Accurate' to 'Most Complete'.

When users are done captioning, they can stop or pause the application to terminate the audio stream. This will let workers complete their current transcription task and ask them to continue captioning other audio for a time in case the users needs to resume captioning quickly.

Legion:Scribe is also able to forward the live output to a second group of workers who are asked to use an editing interface to correct the final stream. While this is optional, it can help correct many of the easily identifiable small errors made by workers and the input combiner. Additionally, users themselves have the option of making corrections to the final stream for errors such as out-of-order words, or a term known to them that remote workers may have missed. The user interface allows users to edit, add or delete words within the transcription, in realtime. As the transcription is generated, the meta information is visually presented to assist the user with the edits. Legion:Scribe returns information such as the confidence of each spelling, possible alternative words and arrangements.

The interface can be shared by other people on different computers, affording for a collaborative environment where interested groups are able to curate a transcript, fixing any

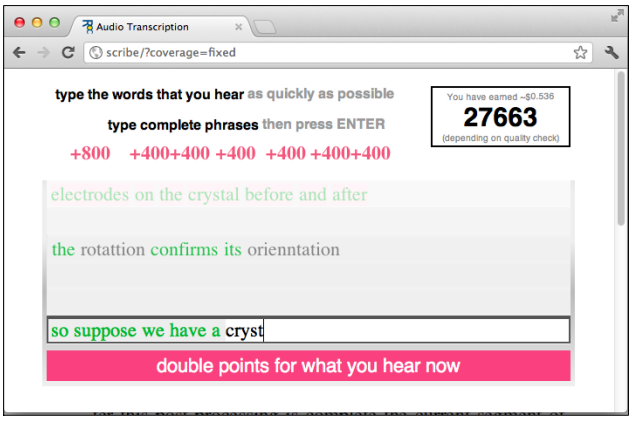

Figure 4: The captioning interface shown to workers by Legion:Scribe. It encourages workers to cover specific portions of the audio using both audio and visual cues during 'bonus rounds', (periods in which workers are incentivized to participation).

collisions within the graph. Many of the common edits are abstracted by the interface to allow for interactions such as a two click replacement for typos or word replacement by alternatives, visual contrast to draw attention to low confidence outputs and transitions to confirm a change made by other collaborators.

4. ONLINE QUALITY ESTIMATION

To estimate the quality of workers and the captions they produce in real-time, we primarily consider agreement between the captions that different workers produce. The idea is that workers whose captions overlap the most with other workers are likely to be the best, whereas workers who rarely overlap with other workers are likely to be the worst. This matches both our intuition that it should be difficult to guess the input that another worker will provide, and follows from prior work in achieving quality work from the crowd, e.g. the ESP Game [18].

In our case, input that is provided by more than one worker is likely to be correct. In practice, estimating words that are contributed by more than one worker is not as simple as it first appears due to alignment. How do we know that a worker's mention of word w is really a match of another worker's mention of w? Legion:Scribe aligns the partial phrases contributed by each worker to form a final output stream, but is often strict resulting in low coverage. For the filtering step described in this paper, we use a relaxed notion of agreement that says two workers agree on a word if each says it within t seconds of one another, where t is a parameter that can be tuned that we set to be 10 seconds.

4.1 Per Worker Quality

Our first approach is to use word-level agreement to dynamically determine if a particular worker is producing high-quality input. The idea is to again look at word-level agreement over the sliding time window, but to use agreement to assign a quality score to the worker, instead of using it to select whether a word is passed on. The per worker quality score is simply the fraction of words produced by a worker within the time window that have also be contributed by another worker during the time window. Using agreement to

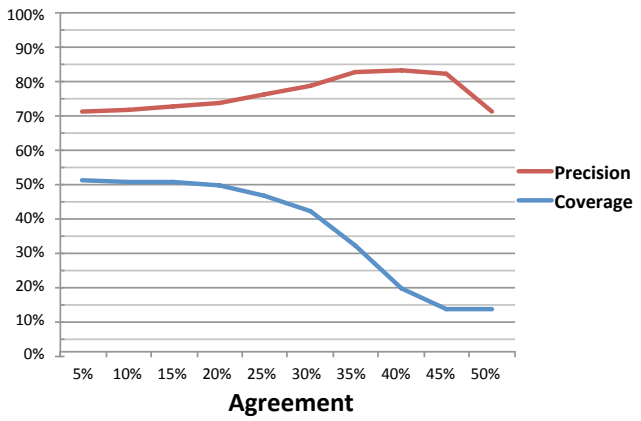

Figure 5: Graph showing an example in which workers captioning a clip actually did worse overall as our per-worker agreement threshold was increased beyond the optimal point.

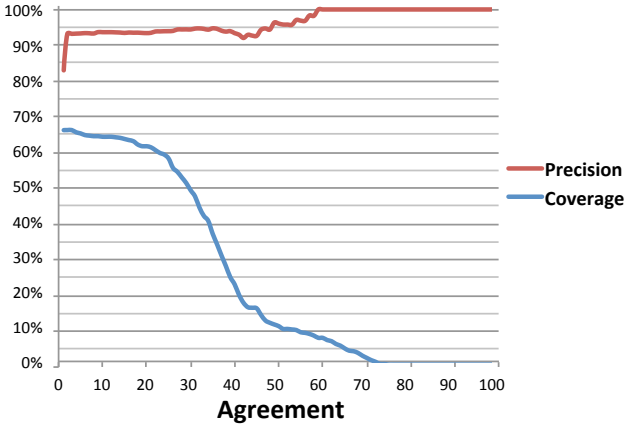

Figure 6: The tradeoff in terms of coverage and precision that we experience when requiring increasing required worker agreement. Even minimal overlap with other workers (0.1) improves precision. Requiring too much agreement negatively affects coverage, which eventually goes to zero.

judge the quality of workers and then forwarding the input of "good" workers on immediately has been used before in real-time crowdsourcing systems [14], but not for real-time captioning or other natural language tasks.

One of the main drawbacks of this system is that there is minimal fine-grained control over what input is accepted. Instead, we rely on trusted workers to continue providing valid input. This reliance results in two problems: first, if a previously reliable worker begins to input poor quality captions, we will still accept the input immediately, even if it's clear they are now an outlier. Second, increasing the threshold for reliability does not result in a completely monotonic change in accuracy because invalid input from users who have not yet been rated, or good workers who make mistakes are included into a smaller get of correct answers, and their contributors then down-weighted, preventing possible good input from being added by the worker.

Figure 5 shows an example of such a situation, in which fewer number of bad inputs are forwarded to the system, but those included represent an increasing proportion of the inputs as even good workers are barred from contributing due to such low tolerances for bad workers by the system. One way to avoid this is to start workers below the minimal thresholding value, requiring them to "prove" themselves before accepting their input. However, this potentially reduces coverage too much at the beginning of a session or at any point of particularly high turnover in the crowd.

4.2 Word-by-Word Quality

Our second approach seeks to filter out words that are unlikely to appear in the true signal because too few workers agree on the word. This filtering step receives each word in real-time and looks back to see if it appears at least k times in the past t seconds. The effect is that at least k workers need to contribute the word before it will be passed through this filtering stage. Many crowd algorithms are based on redundancy like this; however, in a real-time system like Legion:Scribe, the benefit of the added confidence achieved through redundancy comes at the cost of both latency and coverage. Since words will not be passed through to the user until they are input by at least k workers, latency is increase by the time provided for this agreement to occur. Furthermore, correct words that are contributed may not be passed through at all if they are covered by too few workers within the timespan t, even if they are eventually said by enough other workers. This creates a tradeoff in the selection of t that balances response time, with giving workers sufficient chance to implicitly agree on content.

5. EXPERIMENTS

In order to test our quality estimation methods, we conducted experiments with workers recruited through the Amazon Mechanical Turk microtask marketplace [1]. For Legion:Scribe to scale, we believe it will be beneficial to be able to recruit workers online from elastic marketplaces like this one. Mechanical Turk provides a valuable testbed for this paper because workers vary substantially in their reliability and in the quality of work that they provide. A number of other research projects have used it as a way to quickly and easily recruit crowd workers to test various crowd algorithms intended to improve worker reliability [12, 4, 5].

5.1 Data Collection

We collected a data set of speech selected from freely available lectures on MIT OpenCourseWare[1]. These lectures were chosen because a primary goal of Legion:Scribe is to provide captions for classroom activities, and because the recording of the lectures roughly matches our target as well – the clips generally consist of continuous speech captured by a microphone in the room. There are often multiple speakers, e.g. students asking questions. We chose four 5-minute segments that contained speech from courses in electrical engineering and chemistry, and had them professionally transcribed at a cost of $1.75 per minute. Despite the high cost, we found a number of errors and omissions, which we manually fixed to ensure no errors were observed. This data set is described in more detail in [13].

To collect data on Mechanical Turk, we modified the base captioning interface (Figure 4) in two ways. First, we introduced a 45 second video that turkers were required to

[1]http://ocw.mit.edu/courses/

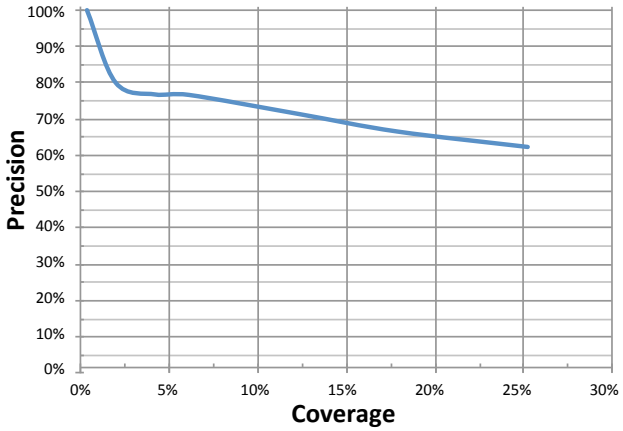

Figure 7: The tradeoff between coverage and precision encountered when increasing the required redundancy. Although dependent on the number of workers, results showed that requiring redundancy increased precision at the cost of coverage.

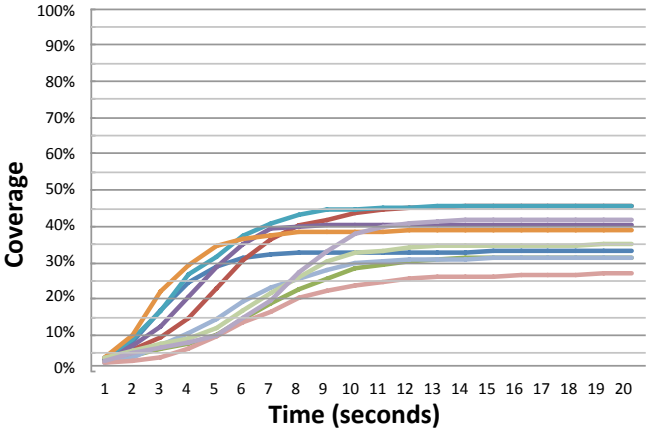

Figure 8: Graph showing the varying coverage rates of workers as acceptable latency is increased. The plots of each worker's coverage level off prior to 10 seconds, with very little additional coverage gained after that point. Thus, we choose 10 seconds as our window time for comparing answers.

play (and presumably watch) that described the captioning process before they were allowed to caption. Workers were paid $0.02 for watching this video, but could not collect any payment until they had captioned at least $0.05 of work. Second, we modified the interface so that in addition to showing points achieved, it also showed an amount of money that these points would be worth when redeemed. Our exchange rate for points was approximately 500 points per cent USD. This works out to an achievable pay rate of approximately $20.00 per hour depending on the skill of the worker and the speech content of what they are captioning, which is a very good rate for the Mechanical Turk marketplace. We expect that workers will initially receive less, then over time will be able to achieve this rate.

Workers were recruited using the quikTurkit real-time recruitment tool [5]. Throughout the experiment, the number of workers actively engaged varied, but never dropped below four. A total of 42 workers contributed to the task over a 20-minute time period, which cost a total of $9.55 USD (a rate of just under $30.00 per hour). We maintained a fairly constant worker pool with 14, 16, 19, 17 workers potentially contributing to each of the four clips, respectively.

Workers seemed to enjoy the task. Four of the workers wrote to us after the task remarking positively about the work[2]. For instance, one worker wrote "I was curious if you had any plans to schedule these HITs in the future. I find them fascinating and fun and would like to look out for them." We have not yet tried to optimize cost, but the positive reaction to the task suggests that either a paid or volunteer model may be appropriate for attracting workers for Legion:Scribe.

We manually looked over the results from each worker to understand the types of errors made. Most workers gave what appeared to be reasonable captions, although we noticeably more spelling errors than in the tests with local workers presented in [13]. Approximately a quarter of the workers gave clearly bad input, most often because they did not understand the instructions or because they were unable to hear the sound for some reason. One of the audio clues given to workers by the interface is a beeping sound before the "bonus period" starts. Two workers typed these beeps, one as "beep" and the other as "tring." Another worker typed 65 words of the form, "I cannot hear the sound I don't know what I'm supposed to do." Although these examples are relatively easy for people to spot manually, Legion:Scribe previously had no way to filter them out automatically and would have likely included them in the final caption stream.

5.2 Quality Estimation Results

We analyzed both per worker and word-by-word quality estimation methods on the data collected to explore how they affect the three evaluation metrics that we introduced previously (Section 1.1). We focused on precision and coverage because WER is highly dependent on the method used to merge inputs together. Estimating worker quality focuses on improving the input to the merge step, thus our goal is to improve precision without substantially lowering coverage.

5.2.1 Per Worker Quality Estimation

We also considered worker-level quality estimation. Figure 6 shows the effect of increasing the level of agreement required between workers to include a given word. Requiring even modest agreement (of just 10%) can result in substantially higher precision (82.9% to 93.2%) with no change in overall coverage. This is due to the fact that input with no agreement whatsoever is almost always errant (typically from workers who misunderstood the task and were, for example, captioning non-speech sounds instead).

5.2.2 Word-by-Word Quality Estimation

Figure 7 shows the tradeoff seen when applying our word-by-word quality estimation. As expected, requiring increased redundancy amongst worker input before accepting a word improved precision, but also decreased coverage. With no redundancy requirement, precision was 57.9%, but rose to 81.2% when requiring redundancy of just 2 workers. Requir-

[2]It is not particularly common to receive feedback from workers, and even less common to receive positive feedback

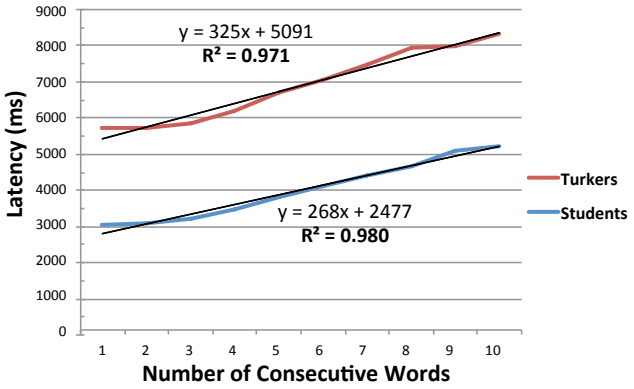

Figure 9: Graph showing the latency plots for student volunteers and Mechanical Turk workers.

ing redundancy at levels of more than a small proportion of workers caused coverage to drop severely. Although this is heavily dependent on the number of actively participating workers, it seems that substantial benefit can be achieved by only requiring low levels of redundancy. This eliminates only very rare input, which is often indicative of a worker-specific erroneous input. Otherwise, workers must consistently caption highly overlapping segments, hurting the overall transcript generated and requiring greater numbers of workers (which increases costs).

5.2.3 *Latency*

Although we did not explicitly consider latency in our measures of error or in our metrics for estimating quality, it was clearly a factor because the latency with which words are receives directly impacts whether they will be in the time window or not used for agreement. How latency changes as additional workers are added plays a significant role in the ability to use crowds of captionists. Unsurprisingly, the trend is that with more workers, average latency goes down, from a single worker average of 4.4 seconds, to a group average of 2.6 seconds. We expect this because each new worker may type a particular word faster than the rest either by chance or because that word appeared nearer to the beginning of the partial caption they contributed. Figure 8 shows the coverage over time of individual workers. Importantly, the latency graph helps to justify the 10 second window use for the rest of the tests because it shows that by 10 seconds most words that will be received have been received.

We also investigated the types and causes of latency seen in workers. There are 2 main types: initial delay and progressive delay. We compared a group of 20 student workers, to a group of 21 turkers and found that the initial delay was significantly different between the two groups - 2477ms on average for student workers, and 5091ms for turkers. Interestingly, this shift was the only major difference between the groups. We measure the delay based on the position in the current chain of words being typed. The additional latency incurred by each word was lower for students, but closely mirrored turkers, both showing a linear trend with $R^2 = 0.98$ and $R^2 = 0.97$ respectively. There was an average of a 268ms additional latency per word for students, and 397ms for turkers (as shown in Figure 9). Based on this, we want to encourage workers to type shorter segments when possible in order to decrease latency.

6. DISCUSSION

This paper has demonstrated that the quality of captions and their providers can be determined in an online fashion as they are received. It is clear that even crowd workers drawn from Amazon's Mechanical Turk can caption real-time audio, which helps to validate the approach used by Legion:Scribe. Not only were workers able to collectively cover the input speech, but they also seemed to enjoy the task based on the feedback we received. The challenge going forward is to remove the noise from the captions they provide and merge it into a usable output stream.

The methods introduced for quality estimation successfully allowed precision and coverage to be tuned, although both had tradeoffs. The word-by-word method improved accuracy, but at the cost of both latency and coverage. The per-worker quality metric had a more interesting response. At lower levels of agreement (10%-30%), it caused precision to rise but at the cost of coverage, which is what we expected. However, at higher levels of agreement, precision actually went down due to instability caused by very few workers being selected at any given time. This effect would likely be mitigated by larger numbers of workers, but for crowd captioning to be effective the number of workers needed should be small. Each method currently requires parameters to be tuned, although it seems likely that both receive the most benefit when using relatively low agreement (2 word agreement for per-word, and 10% agreement for per-worker), as coverage decreases at a higher rate than the accuracy increases after low levels of agreement.

We saw no instances in our data set of workers who changed dramatically in quality over the course of the study. Workers started with low or high quality and seemed to stay consistent. If this trend holds over longer trials, then it would be possible to block bad workers entirely, or to give good workers more leeway when they disagree with the others.

Finally, we observed few examples of crowd workers be outright malicious. Instead, workers identified as being low quality via this method generally experienced an error with the caption input page (e.g., no sound played), or misunderstood the task they were to do (e.g., described background noise in the sound clip instead of typing words). Therefore, estimating the quality of workers may allow us to identify usability problems with future systems that may not be detected as quickly or reliably using other means.

7. FUTURE WORK

Real-time crowdsourcing has the potential to dramatically lower the cost of real-time captioning and dramatically increase its availability. By using crowd workers available from many existing sources (such as Amazon's Mechanical Turk), instead of relying solely on volunteers, crowd captioning can be made scalable enough for real-world deployment.

An important step for engaging the crowd in real-time captioning is determining what input is good and what is not. Classifying input in this way enables systems to pay workers appropriately for their input, encouraging workers to provide high-quality input. This paper has introduced methods for doing so at the level of workers and individual words, and suggests a number of opportunities for future work. Future work may explore building reputation over time in order to avoid bootstrapping models of workers dynamically during each session, perhaps allowing workers who

have demonstrated they consistently contribute high-quality inputs to override the crowd decision.

The final goal of this system is to provide high-quality real-time captions for deaf and hard of hearing people using less reliable sources of labor such as general crowds. Legion:Scribe is a complex system with many components, and research thus far has primarily gone to demonstrating the feasibility of the approach. In this paper in particular, we have worked with the assumption that removing errant input early will make later merging stages easier, but it may be that later approaches may benefit from considering all of the input at once even if some of it is incorrect. We also assume that filtering this bad input will improve the usability of the captions for deaf and hard of hearing people; investigating the tradeoff between removing errant words and readability is a promising area for future work.

Our current approach only uses agreement with other human workers to estimate quality, meaning we require more workers than is necessary to cover the input speech because some are providing redundant inputs. Future work therefore may look at other signals of quality - for instance, spelling, grammar, or agreement with ASR - that may be more robust. More robust models may be possible by using these signals in conjunction with crowd agreement.

8. CONCLUSIONS

In this paper we have explored real-time quality control in real-time captioning in order to improve the quality of transcripts generated from crowd workers of initially unknown quality. We do this by introducing methods that can estimate the quality of workers and each word they contribute. We demonstrated the utility of these methods, through our experiments using workers from Mechanical Turk, by showing they can increase the resulting accuracy of captions while keeping the coverage of the speech signal nearly constant.

9. ACKNOWLEDGEMENTS

This work has been supported by Google and NSF Awards IIS-1149709 and IIS-1116051.

10. REFERENCES

[1] Amazon's mechanical turk. http://www.mturk.com.

[2] Y. C. Beatrice Liem, H. Zhang. An iterative dual pathway structure for speech-to-text transcription. In *Proceedings of the 3rd Workshop on Human Computation (HCOMP '11)*, HCOMP '11, 2011.

[3] M. S. Bernstein, J. R. Brandt, R. C. Miller, and D. R. Karger. Crowds in two seconds: Enabling realtime crowd-powered interfaces. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, UIST '11, p 33–42, 2011. ACM.

[4] M. S. Bernstein, G. Little, R. C. Miller, B. Hartmann, M. S. Ackerman, D. R. Karger, D. Crowell, and K. Panovich. Soylent: a word processor with a crowd inside. In *Proceedings of the 23nd annual ACM symposium on User interface software and technology*, UIST '10, p 313–322, 2010. ACM.

[5] J. P. Bigham, C. Jayant, H. Ji, G. Little, A. Miller, R. C. Miller, R. Miller, A. Tatarowicz, B. White, S. White, and T. Yeh. Vizwiz: nearly real-time answers to visual questions. In *Proceedings of the 23nd annual ACM symposium on User interface software and technology*, UIST '10, p 333–342, 2010. ACM.

[6] J. P. Bigham, R. E. Ladner, and Y. Borodin. The design of human-powered access technology. In *Proceedings of the 2011 SIGACCESS Conference on Computers and Accessibility (ASSETS 2011)*, ASSETS 2011, p 3–10, 2011. ACM.

[7] L. Chilton. Seaweed: A web application for designing economic games. Master's thesis, MIT, 2009.

[8] S. Cooper, F. Khatib, A. Treuille, J. Barbero, J. Lee, M. Beenen, A. Leaver-Fay, D. Baker, Z. Popovic, and F. Players. Predicting protein structures with a multiplayer online game. *Nature*, 466(7307):756–760, 2010.

[9] X. Cui, L. Gu, B. Xiang, W. Zhang, and Y. Gao. Developing high performance asr in the ibm multilingual speech-to-speech translation system. In *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*, p 5121–5124, 2008.

[10] L. B. Elliot, M. S. Stinson, D. Easton, and J. Bourgeois. College Students Learning With C-Print's Education Software and Automatic Speech Recognition. In *American Educational Research Association Annual Meeting*, New York, NY, 2008.

[11] T. Imai, A. Matsui, S. Homma, T. Kobayakawa, K. Onoe, S. Sato, and A. Ando. Speech recognition with a re-speak method for subtitling live broadcasts. In *ICSLP-2002*, p 1757–1760, 2002.

[12] A. Kittur, B. Smus, and R. Kraut. Crowdforge: Crowdsourcing complex work. Technical Report CMUHCII-11-100, Carnegie Mellon University, 2011.

[13] W. S. Lasecki, C. D. Miller, A. Sadilek, A. AbuMoussa, and J. P. Bigham. Real-time captioning by groups of non-experts. In *Proceedings of the 25rd annual ACM symposium on User interface software and technology*, UIST '12. *To Appear*, 2012.

[14] W. S. Lasecki, K. I. Murray, S. White, R. C. Miller, and J. P. Bigham. Real-time crowd control of existing interfaces. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, UIST '11, page 23–32, 2011. ACM.

[15] G. Little, L. B. Chilton, M. Goldman, and R. C. Miller. Turkit: human computation algorithms on mechanical turk. In *Proceedings of the 23rd annual ACM symposium on User interface software and technology*, UIST '10, p 57–66, 2010. ACM.

[16] T. Matthews, S. Carter, C. Pai, J. Fong, and J. Mankoff. Scribe4me: evaluating a mobile sound transcription tool for the deaf. In *Proceedings of the 8th international conference on Ubiquitous Computing*, UbiComp'06, p 159–176, 2006. Springer-Verlag.

[17] L. von Ahn. *Human Computation*. PhD thesis, Carnegie Mellon University, Pittsburgh, PA, 2005.

[18] L. von Ahn and L. Dabbish. Labeling images with a computer game. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, CHI '04, p 319–326, 2004. ACM.

[19] M. Wald. Creating accessible educational multimedia through editing automatic speech recognition captioning in real time. *Interactive Technology and Smart Education*, 3(2):131–141, 2006.

[20] A. A. Ye-Yi Wang and C. Chelba. Is word error rate a good indicator for spoken language understanding accuracy. In *IEEE Workshop on Automatic Speech Recognition and Understanding*, 2003.