Adaptive Time Windows for Real-Time Crowd Captioning

Matthew J. Murphy
ROC HCI, Computer Science
University of Rochester
Rochester, NY 14620 USA
m.murphy@rochester.edu

Walter S. Lasecki
ROC HCI, Computer Science
University of Rochester
Rochester, NY 14620 USA
wlasecki@cs.rochester.edu

Christopher D. Miller
ROC HCI, Computer Science
University of Rochester
Rochester, NY 14620 USA
c.miller@rochester.edu

Jeffrey P. Bigham
ROC HCI, Computer Science
University of Rochester
Rochester, NY 14620 USA
jbigham@cs.rochester.edu

CHI'13, April 27 – May 2, 2013, Paris, France.
ACM 978-1-4503-1952-2/13/04.

Abstract

Real-time captioning provides deaf and hard of hearing users with access to live spoken language. The most common source of real-time captions are professional stenographers, but they are expensive (up to $200/hr). Recent work shows that groups of non-experts can collectively caption speech in real-time by directing workers to different portions of the speech and automatically merging the pieces together. This work uses 'one size fits all' segment durations regardless of an individual worker's ability or preferences. In this paper, we explore the effect of adaptively scaling the amount of content presented to each worker based on their past and recent performance. For instance, giving fast typists longer segments and giving workers shorter segments as they fatigue. Studies with 24 remote crowd workers, using ground truth in segment calculations, show that this approach improves average coverage by over 54%, and F_1 score (harmonic mean) by over 44%.

Author Keywords

Real-time crowdsourcing; assistive technology

ACM Classification Keywords

H.5.m [Information interfaces and presentation]: Miscellaneous.

General Terms

Human Factors; Design; Experimentation

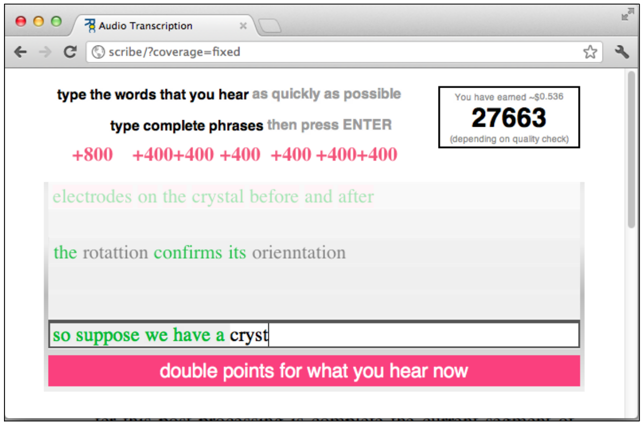
Introduction

Real-time captioning provides deaf and hard of hearing users access to mainstream classrooms, discussions with friends, and public events. The most common source of real-time captioning are professional stenographers, who are very accurate but cost up to $200/hr. Automatic speech recognition (ASR) is more affordable but produces unusable captions in real-world domains on which it has not been trained in advance [5].

Legion:Scribe [5] showed that groups of non-experts (the crowd) can collectively caption speech in real-time with high accuracy by automatically merging the partial results of multiple workers (Figure 1). This approach brings human intelligence to bear to perform much more accurately than automatic systems, while retaining low costs by lowering requirements of expertise. However, since the segment durations used are fixed, they are not always well suited to an individual worker's typing speed or preference.

In this paper, we introduce a method for adaptively scaling the duration of these time periods in order to adjust the amount of content each worker is given to better match their skill level, based on their past performance. Studies with 24 remote crowd workers show that this approach improves average coverage by 54.15%, and F_1 score, or harmonic mean (a combined measure of precision and recall), by 44.33%.

We then discuss how to extend the approach shown here to work in real-world domains with no ground truth comparison available. Finally, we conclude with a discussion of potential improvements to our approach, and methods for online scheduling of workers with dynamic segment durations.



type the words that you hear as quickly as possible

type complete phrases then press ENTER

+800 +400+400 +400 +400 +400+400

You have earned ~$0.536

27663

(depending on quality check)

electrodes on the crystal before and after

the rotattion confirms its orienntation

so suppose we have a cryst

double points for what you hear now

Figure 1: Crowd captioning worker interface. Workers are directed to type different portions of the audio via the visual indication and changes to the audio stream.

Background

Real-time captioning is supported by professional captionists, automatic speech recognition (ASR), or more recently, crowd captioning systems such as Legion: Scribe [5]. Each of these approaches has limitations. Professional captionists are the most reliable option, but are not available on demand, can cost over $150/hr, and can only be scheduled in blocks of an hour. ASR is relatively cheap and available on-demand, but often provides extremely low-quality results in realistic settings. Crowd captioning uses multiple non-expert human captionists each submitting partial input which is then automatically recombined to generate a caption in real-time. This approach is more robust in real-world situations, but requires that workers contribute useful partial captions. Using time warps, we are able to improve individual worker performance on captioning tasks by giving workers a more manageable task, while still providing answers in real-time.

Real-Time Human Computation

Crowd captioning is a type of real-time human computation. Real-time human computation has been explored in systems like VizWiz [2], which was one of the first applications to target nearly real-time responses from the crowd, and Adrenaline [1], which uses a retainer model to reduce response time to less than two seconds. Legion introduced the idea of engaging a synchronous crowd in a continuous real-time task, using the crowd to collectively control existing user interfaces as if they were a single individual [4]. Each workers submits input independently of other workers, then the system uses an *input mediator* to combine the input into a single control stream.

Legion:Scribe [5, 3] extends the idea of using a continuous stream of input from workers to real-time captioning, generating transcripts by combining multiple workers' partial captions into a single final caption stream. Legion:Scribe allows deaf users to stream content from the mobile devices to a server, which forwards it to multiple workers. Multiple partial captions from workers are sent to the server where they are merged into a single, more reliable transcript, and then forwarded back to the user. Our adaptive time window approach represents a way to tailor the task to each worker, and helps to bridge the gap between what the system expects and what the worker is capable of. One of the ways we measure workers' performance is using the *coverage* metric used in [5]. This measures the amount of content that workers were able to capture from the original audio within a certain amount of time. This is similar to recall, but with the added requirement that the word be submitted within some fixed window of time (in our experiments we use a 10 second window). To compare the overall quality of different sets of captions, we also use the F measure (harmonic mean), a measure common in information retrieval research.

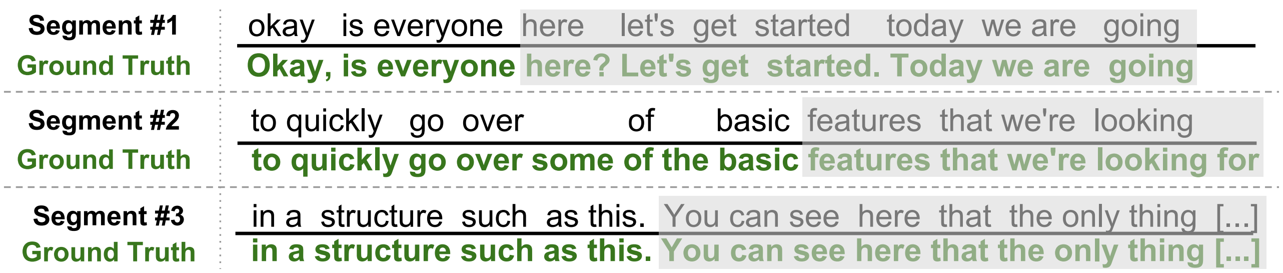
System

The web-based Scribe interface is able to systematically vary the volume of the audio that captionists hear in order to artificially introduce saliency. To facilitate this, each captionist is assigned an in-period and an out-period. The in-period is the length of time that the captionist hears audio at a louder volume, and the out-period is the length of time after the in-period that the captionist hears audio at a quieter volume. For example, if the in-period is 4 seconds and the out-period is 6 seconds, the captionist would hear 4 seconds of louder audio, followed by 6 seconds of quieter audio, after which the cycle would immediately repeat until the task is complete. Workers are instructed to transcribe only the audio they hear during the in-periods, and are given extra compensation for correct words occurring during in-periods.

We investigated two different methods of assigning in- and out-periods to workers. The first, and most basic, is a fixed set of periods. In this configuration, the system simply assigns a constant in-period and out-period to the worker. However, in most cases, a constant set of periods is not ideal for a worker, due largely to the wide variation of speaking rates, even within the same piece of audio. To remedy this, we tested an adaptive method for determining in- and out-periods. In this configuration, the system starts each worker with a pre-determined fixed period, and then uses a weight-learning algorithm to constantly adapt and modify the worker's periods based on their performance. Once a worker completes a segment of audio, the system calculates a weight for the worker, and the in- and out-periods are updated accordingly.

Weight Learning

To determine the periods, the dynamic method calculates a weight for each worker after each segment. The weight



Segment #1: okay is everyone here let's get started today we are going

Ground Truth: Okay, is everyone here? Let's get started. Today we are going

Segment #2: to quickly go over of basic features that we're looking

Ground Truth: to quickly go over some of the basic features that we're looking for

Segment #3: in a structure such as this. You can see here that the only thing [...]

Ground Truth: in a structure such as this. You can see here that the only thing [...]

Figure 2: An example of adaptive segment duration adjustment over time. The unobscured text shows what a worker was able to cover in the segment they were asked to caption. The greyed out portion is what the worker was *not* asked to caption (the rest of the crowd was responsible for those parts). Since the worker captioned all of the words in segment 1, the system automatically increases the load for the next period. When content is missed in segment 2, the load is reduced. Eventually, in segment 3, the worker is covering as much as they can do well, increasing the amount of coverage seen from a single worker, without introducing additional inaccuracies.

of a worker could be seen as a type of "net words-per-minute" calculation, where a higher weight indicates a faster and more accurate typist. The weight of a worker is calculated according to the following formula:

$$w_i = \alpha w_{i-1} + (1-\alpha)p \qquad (1)$$

Where w_i is the current weight, w_{i-1} is the previous weight, and p is the performance of the worker in the most recent segment of audio. α is a discount factor which is selected such that $0 < \alpha < 1$. Its effect is that a worker's weight is determined more by recent typing performance. The performance of a worker during the previous segment, p, is computed according to the following formula:

$$p = \frac{n-(n-c)d}{t} \qquad (2)$$

Where n is the number of total words the worker typed, t is the number of minutes that the worker typed (usually a fraction), c is the number of correct words the worker typed, and d is the error index. The error index is the penalty given to incorrect words, such that if the error index is 1, the equation deducts 1 correct word from the performance calculation. In our tests, we determined the number of correct words by matching words to a baseline file, containing a full transcription of the audio. While a baseline will not be available in a real-world scenario, our goal is to prove that adaptive durations are beneficial. We discuss how performance can be measured in the absence of a baseline in the Discussion section.

Once the weight is determined, it is used to calculate the final period times. For the sake of simplicity, the sum of the in-period and the out-period is set to a constant value, and the worker's weight is used to determine an optimal ratio between the two. The Legion:Scribe system supports a variable sum of periods, but a constant value was chosen to make calculations more straightforward. The in-period is determined according to the following formula:

$$r = T\frac{w_i}{s} \qquad (3)$$

Where r is the in-period, T is the constant total segment time (in-period plus out-period), w_i is the current weight, and s is the approximate speaking rate of the audio segment in words per minute.

Experiments

We recruited a total of 24 crowd workers from Mechanical Turk, 12 for both the fixed and adaptive segment. Our task paid $0.05 and workers could make an additional $0.002 bonus per word. Trials were randomized and workers were not able to repeat the task. Each trial consisted of captioning a 2:40 minute audio clip. Each segment consisted of only a few seconds of content to caption, so our clip was long enough to learn improved segment durations and test workers' abilities.

Results

Using adaptive segments lead to a significant increase of 54.15% in the overall coverage, from 14.76% to 22.76% ($p < 0.05$), and of 44.33% in F_1 score, from 0.242 to 0.349 ($p < 0.05$). Accuracy fell slightly from 84.33% to 80.11%, and latency improved from 5.05 seconds to 4.98 seconds, but these changes were not significant.

While even the improved coverage seems low upon initial inspection, it is important to note that the default task assumes that a worker with perfect accuracy and ability to cover all of the content assigned to them will achieve a coverage of approximately 25% (depending on speaker speed fluctuations). Therefore, by increasing coverage from 14.76% to 22.76% coverage, we have essentially improved from 59.04% of this goal to 91.04%.

Discussion

Our results show that tailoring the captioning task to workers can significantly improve their performance on a task. Here, workers were able to caption closer to their full capability, instead of higher skilled workers being restricted. Furthermore, allowing the task to change over time means that if workers tire, get distracted, or the speaker changes pace, the system can compensate.

Deploying Adaptive Segments

One critical component to using these dynamic times in a live system is being able to correctly schedule when workers' segments occur. With fixed windows, scheduling is trivial and can be done a priori, however, when segment lengths are unknown and not required to complement each other, the problem becomes more difficult. While dynamic segment lengths allow each worker individually to perform better than static segment lengths would allow, a scheduling mechanism that both covers the entire audio signal while allowing workers to use their dynamically determined best-performance typing lengths will need to be developed. Such a scheduler would have to take into account that at any given point in time each worker in will have a maximally bounded input period length, as well as minimally bounded rest period length, both of which may change at a moment's notice, which makes it somewhat difficult to continually arrange and rearrange the set of workers so as to cover 100 percent of the signal without prior knowledge of the incoming audio stream.

Another issue that would need to be addressed with a real world implementation of this system would be that in its current form it requires a comparison with a precompiled baseline transcript to gauge worker performance. In a real-time setting such a baseline would not exist, therefore an alternative way of gauging individual worker performance would have to be used, the most likely choice being worker agreement. Legion:Scribe [5] showed that 10 workers can accurately cover an average of 93.2% of an audio stream with an average per-word latency of 2.9 seconds. The resulting captions could easily be used to infer the rate of speech, as well as each worker's performance, by comparing each individual worker's captions to the crowd's result. Such a system would be expected to yield very similar results. However, our goal was to test the effect of adaptive windows on individual workers, and such a system would only be feasible with multiple simultaneous workers.

Future Work

In addition to the improvements we observed, future work will aim to explore tweaks that may further improve both the worker experience on the task, and task efficiency (coverage per dollar). The first of these extensions will be to test different weight adjustment parameters to see if gradual or rapid changes in segment duration work best when finding the optimal. The next is seeing whether or not maintaining weights for individual workers between sessions allows us to increase our accuracy, or if there is too much session-to-session variation to warrant maintaining such information. Another option is to test a combination of more than one method presented here. For instance, allowing workers to pick their initial segment durations may be more accurate of a baseline than using pre-determined durations, meaning that the time until the optimal is found will be reduced.

Another factor that we will explore is the amount of strain placed on workers. Prior work has shown that by using segments instead of continuous audio, workers find the task to be less stressful because workers feel as though they have performed better [4]. We expect that adaptive time windows will enable a similar gain, but may require finding a value below the maximum efficient work load for the worker. For example, we could use a time window 10% below their maximum ability in order to reduce the amount of strain associate with performing a task at the limit of their abilities.

Conclusion

In this paper, we have presented a method for finding more effective segment durations for use in a real-time captioning task. By using adaptive segments, workers were able to increase the amount of words that they could caption reliably. This approach promises to increase the quality of the resulting collective captions, while also reducing the cost of the task and the stress on workers. More generally, this work suggests that adaptively delivering work to individuals in the crowd may help improve worker utilization, and thereby improve the quantity of tasks completed per worker and the pay received by high-performing workers.

References

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

[2] Bigham, J. P., Jayant, C., Ji, H., Little, G., Miller, A., Miller, R. C., Miller, R., Tatarowicz, A., White, B., White, S., and Yeh, T. Vizwiz: nearly real-time answers to visual questions. In *Proc. of the ACM symposium on User interface software and technology*, (UIST 2010), 333–342.

[3] Lasecki, W., and Bigham, J. Online quality control for real-time crowd captioning. In *Proc. of the ACM SIGACCESS Conference on Computers & Accessibility*, ASSETS 2012.

[4] Lasecki, W., Murray, K., White, S., Miller, R. C., and Bigham, J. P. Real-time crowd control of existing interfaces. In *Proc. of the ACM symposium on User interface software and technology*, (UIST 2011), 23–32.

[5] Lasecki, W. S., Miller, C. D., Sadilek, A., Abumoussa, A., Borrello, D., Kushalnagar, R., and Bigham, J. P. Real-time captioning by groups of non-experts. In *In Proc. of the ACM Symposium on User Interface Software and Technology (UIST 2012)*.