Real-Time Captioning by Groups of Non-Experts

Walter S. Lasecki[1], **Christopher D. Miller**[1], **Adam Sadilek**[1], **Andrew Abumoussa**[1]
Donato Borrello[1], **Raja Kushalnagar**[2], **and Jeffrey P. Bigham**[1]

University of Rochester[1]
Computer Science, ROC HCI
Rochester, NY 14623 USA
{wlasecki,sadilek,abumouss,jbigham}@cs.rochester.edu
{c.miller,donato.borrello}@rochester.edu

Rochester Institute of Technology[2]
Computer Science and NTID
Rochester, NY 14623 USA
rskics@rit.edu

ABSTRACT

Real-time captioning provides deaf and hard of hearing people immediate access to spoken language and enables participation in dialogue with others. Low latency is critical because it allows speech to be paired with relevant visual cues. Currently, the only reliable source of real-time captions are expensive stenographers who must be recruited in advance and who are trained to use specialized keyboards. Automatic speech recognition (ASR) is less expensive and available on-demand, but its low accuracy, high noise sensitivity, and need for training beforehand render it unusable in real-world situations. In this paper, we introduce a new approach in which groups of non-expert captionists (people who can hear and type) collectively caption speech in real-time on-demand. We present LEGION:$\mathcal{S}$CRIBE, an end-to-end system that allows deaf people to request captions at any time. We introduce an algorithm for merging partial captions into a single output stream in real-time, and a captioning interface designed to encourage coverage of the entire audio stream. Evaluation with 20 local participants and 18 crowd workers shows that non-experts can provide an effective solution for captioning, accurately covering an average of 93.2% of an audio stream with only 10 workers and an average per-word latency of 2.9 seconds. More generally, our model in which multiple workers contribute partial inputs that are automatically merged in real-time may be extended to allow dynamic groups to surpass constituent individuals (even experts) on a variety of human performance tasks.

ACM Classification Keywords

H.5.2 [Information interfaces and presentation]: User Interfaces. - Graphical user interfaces.

General Terms

Design, Human Factors, Experimentation

Author Keywords

real-time; captioning; transcription; deaf; hard of hearing; text alignment; crowdsourcing

UIST'12, October 7–10, 2012, Cambridge, Massachusetts, USA.

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. 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-world situations [30]. This paper introduces a new approach of having groups of non-expert captionists (people who can hear and type, but are not trained stenographers) collectively caption speech in real-time, and explores this new approach via LEGION:$\mathcal{S}$CRIBE (henceforth $\mathcal{S}$CRIBE), our end-to-end system allowing collective instantaneous captioning for live events on-demand. Since each individual is unable to type fast enough to keep up with natural speaking rates, $\mathcal{S}$CRIBE automatically combines multiple inputs into a final caption.

While visual access to spoken material can also be achieved through sign language interpreters, many DHH people do not know sign language. This is particularly true of the large (and increasing) number of DHH people who lost their hearing later in life [15]. Captioning may also be preferred by some to sign language interpreting for technical domains because it does not involve translating from the spoken language to the sign language[1], but rather transliterating an aural representation to a written one. Finally, like captionists, sign language interpreters are also expensive and difficult to schedule.

Professional captionists (stenographers) provide the best real-time (within a few seconds) captions. Their accuracy is generally over 95%, but they must be arranged in advance for blocks of at least an hour, and cost between $120 and $200 per hour, depending on skill [30]. As a result, they cannot be used to caption a lecture or other event at the last minute, or provide access to unpredictable and ephemeral learning opportunities, such as conversations with peers after class.

Automatic speech recognition (ASR) is inexpensive and available on-demand, but its low accuracy in many real settings makes it unusable. For example, ASR accuracy drops below 50% when it is not speaker-trained, captioning

[1]Sign languages, such as American Sign Language (ASL) are not simply codes for an aural language, but rather an entirely different languages with their own vocabulary, grammar, and syntax.

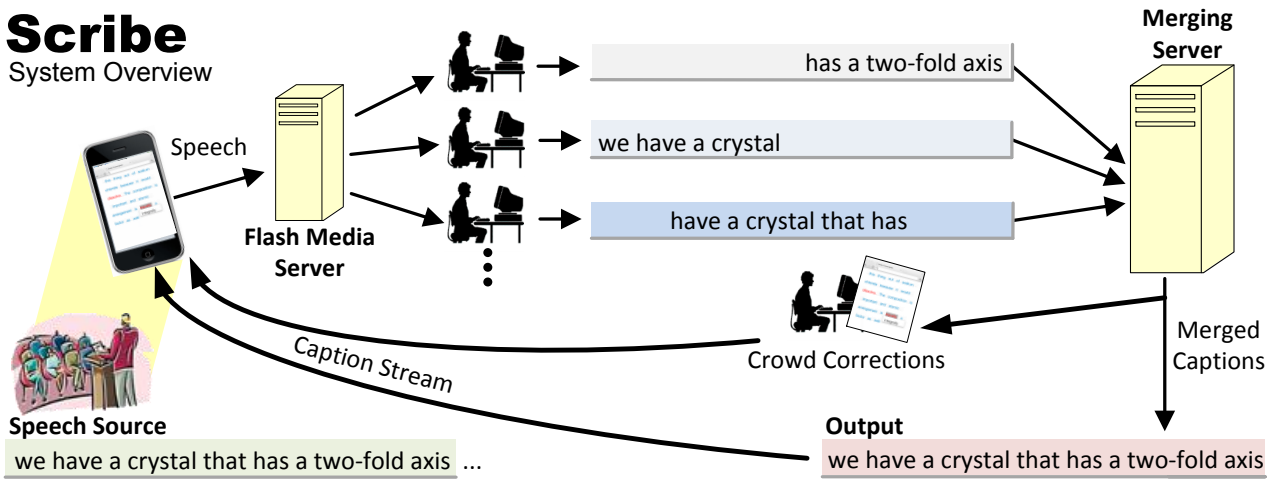

Figure 1. SCRIBE allows users to caption audio on their mobile device. The audio is sent to multiple amateur captionists who use the SCRIBE web-based interface to caption as much of the audio as they can in real-time. These partial captions are sent to our server to be merged into a final output stream, which is then forwarded back to the user's mobile device. Crowd workers are optionally recruited to edit the captions after they have been merged.

multiple speakers, or when not using a high-quality microphone located close to the speaker [12, 7]. Both ASR and the software used to assist real-time captionists often make errors that significantly distort the meaning of the original speech. As DHH people use context to compensate for errors, they often have trouble following the speaker [12].

Non-expert captionists can be drawn from more diverse labor pools than professional captionists, and so we expect captioning by groups of non-experts to be cheaper and more easily available on demand. Recent work has shown, for instance, that workers on Mechanical Turk can be recruited within a few seconds [2, 5]. Recruiting from a broader pool allows workers to be *selectively* chosen for their expertise not in captioning but in the technical areas covered in a lecture. While professional stenographers type faster and more accurately than most crowd workers, they are not necessarily experts in other fields, which often distorts the *meaning* of transcripts of technical talks [30]. SCRIBE will allow student workers to serve as non-expert captionists for $8-12 per hour (a typical work-study pay). Therefore, we could hire several students for less than the cost of one professional captionist.

SCRIBE can benefit people who are not DHH as well. For example, students can easily and affordably obtain searchable text transcripts of a lecture even before the class ends, enabling them to review earlier content they may have missed. Furthermore, we all are subject to a situational disability from time to time [27]. Even a person with excellent hearing can have trouble following a lecture when sitting too far from the speaker, when acoustics are poor, or when it is too noisy.

The key contributions of this paper are as follows:

- We introduce the idea of using non-expects to caption audio in real-time, and present SCRIBE—an end-to-end system that has advantages over current state of the art in terms of availability, cost, and accuracy.
- We show that non-experts can collectively cover speech at rates similar to or above that of a professional (over 93%).
- We demonstrate that SCRIBE can produce transcripts that both cover more of the input signal and are more accurate than either ASR or any single constituent worker.
- More generally, we introduce the idea of automatically merging the real-time inputs of dynamic groups of workers to outperform individuals on human performance tasks.

CURRENT APPROACHES FOR REAL-TIME CAPTIONING

In this section, we first overview current approaches for real-time captioning, introduce our data set, and define evaluation metrics used in this paper. Methods for producing real-time captioning services come in three main varieties: (1) verbatim computer-aided real-time translation, (2) non-verbatim systems, and (3) automatic speech recognition.

Communications Access Real-Time Translation (CART): CART is the most reliable real-time captioning service, but is also the most expensive. Trained stenographers type in shorthand on a "steno" keyboard that maps multiple key presses to phonemes that are expanded to verbatim text. Stenography requires 2-3 years of training to consistently keep up with natural speaking rates that average141 words per minute (WPM) and can reach 231 WPM [17].

Non-Verbatim Systems: In response to the cost of CART, computer-based macro expansion services like C-Print were introduced [30]. C-Print captionists need less training, and generally charge around $60 an hour. However, they normally cannot type as fast as the average speaker's pace, and cannot produce a verbatim transcript. SCRIBE employs captionists with no training and compensates for slower typing speeds and lower accuracy by combining the efforts of multiple parallel captionists.

Automated Speech Recognition: ASR works well in ideal situations with high-quality audio equipment, but degrades quickly in real-world settings. ASR is speaker-dependent, has difficulty recognizing domain-specific jargon, and adapts poorly to changes, such as when the speaker has a cold [12, 9]. ASR systems can require substantial computing power

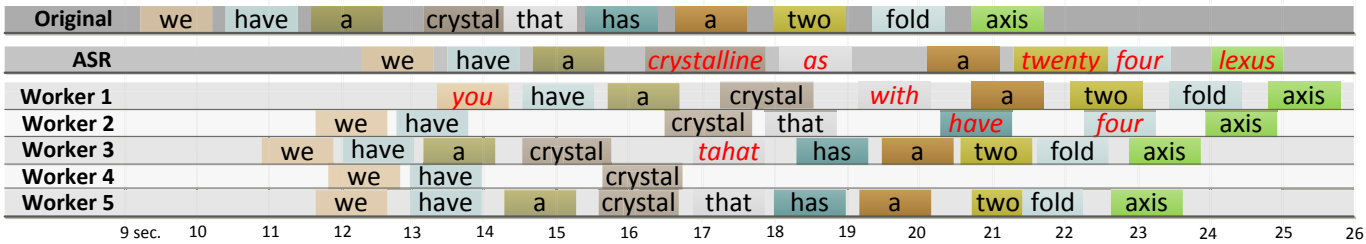

Figure 2. Captions by ASR and five Mechanical Turk workers for a segment of speech. Our results show that merging the input of multiple non-expert workers results in accurate, real-time captions. The left edge of each box represents the time when the typed word was received. Boxes are color-coded to show word position (red denotes incorrect text).

and special audio equipment to work well, which lowers availability. In our experiments, we used Dragon Naturally Speaking 11.5 for Windows.

Another approach is *respeaking*, where a person in a controlled environment is connected to a live audio feed and repeats what they hear to an ASR that is extensively trained for their voice [16]. Respeaking works well for offline transcription, but simultaneous speaking and listening requires professional training. By contrast, SCRIBE enables non-experts to contribute without any special training or skill.

REAL-TIME CAPTIONING WITH NON-EXPERTS

Non-expert captionists can be anyone who can type what they hear. People are able to understand spoken language with ease, but most lack the ability to record it with sufficient speed and accuracy to generate an exact transcript of audio in real-time. As we will see, no single person achieves an acceptable level of coverage when captioning extended audio clips. SCRIBE overcomes this problem by automatically unifying input from multiple workers. Additionally, SCRIBE's user interface encourages different workers to concentrate on different segments of the speech by adjusting audio saliency.

Our approach only requires workers to be able to hear and type. Non-expert captionists can be drawn from the general population, micro-task marketplaces (such as Amazon's Mechanical Turk), groups of volunteers, and students. The pool of workers is dynamic. As a result, no specific worker can be relied upon to be available at a given time or to continue working on a job for any amount of time. Workers cannot be relied upon to provide high-quality work of the type one might expect from a traditional employee. This stems from the lack of direct control, misunderstanding of the task, or delays that are beyond their control, such as network latency. Using multiple workers decreases the likelihood that no worker will be able to provide a correct caption quickly.

Aural speech is often noisy and ambiguous. Speakers may make mistakes, use unfamiliar terms, or have a unique accent. In these cases, people have the advantage of *understanding* the context the word was spoken in, unlike ASR. This makes people less likely to mistake a word for another that does not fit the current context. Furthermore, ASR is unlikely to recognize unusual words or terms the speaker defined during the presentation, whereas human workers may have prior knowledge of the topic, and can learn new terms on-the-fly. In most cases workers are only delayed by their typing rate, not comprehension, allowing faster responses than most ASR, without sacrificing accuracy.

We focus on two types of worker: *local* and *remote*. Local workers are able to hear the audio with no communication delay, and at the original audio quality. These workers may be more familiar with the topic being discussed, and may already be used to the style of the speaker. Remote workers are easier to recruit on-demand, and are generally cheaper. However, remote workers will not be trained on the specific speaker, and may lack the background knowledge of a local worker. These two types of workers can be mixed in order to extract the best properties of each. For instance, using local workers to take advantage of the low latency when possible, while using remote workers to maintain enough captionists to ensure consistent coverage.

In order to successfully generate complete and accurate captions, we need to intelligently merge all of the noisy partial inputs into a single stream. Workers have different typing speeds, captioning styles, and connection latencies, making time alone a poor signal for word ordering. Aligning based on word matching can be more consistent between workers, but spelling mistakes, typographical errors, and confusion on the part of workers make finding a consensus difficult. A robust alignment method must be able to handle these inconsistencies, while not overestimating the similarity of two inputs.

Using worker input exclusively fails to take advantage of existing knowledge of languages and common errors. We use additional information about the most likely intended input from a worker by making use of language and typing models. For the language model, we use bigram and trigram data from Google's publicly available N-gram corpus. This provides prior probabilities on sets of words, which we use to resolve ordering conflicts in workers' input. To determine *equivalent* words, we use the Damerau-Levenshtein distance [10] between the words, weighted using the manhattan distance between the letters on a QWERTY keyboard.

Metrics for Evaluation

Determining the quality of captions is difficult [31]. The most common method is word error rate (WER), which performs a best-fit alignment between the caption and the ground truth. The WER is then calculated as the sum of the substitutions S, the deletions D, and the insertions I needed to make the two transcripts match divided by the total number of words in the ground truth N, or $\frac{S+D+I}{N}$. A key advantage of human captionists over ASR is that humans tend to make more reasonable errors because they are able to infer meaning from context, influencing their prior probability toward words that make sense in context. We anticipate this will make SCRIBE more usable than automated systems even when the results of

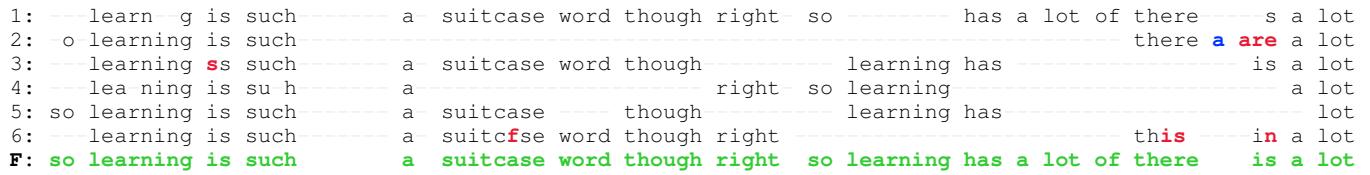

Figure 3. Example output of our MSA algorithm. Each line is a partial caption input by a worker, and the final merged caption (F). Dashes represent "gaps" inserted to attain an optimal alignment given our language model. While individual workers provide noisy and incomplete data, merging multiple captions significantly improves coverage and precision.

traditional metrics are similar. Figure 2 gives an example of the confusing errors often made by ASR, substituting "twenty four lexus" for "two-fold axis".

We define two other metrics in addition to WER to help characterize the performance of real-time captioning. We believe these metrics are particularly useful in understanding the potential of various approaches. The first is *coverage*, which represents how many of the words in the true speech signal appear in the merged caption. While similar to 'recall' in information retrieval, we choose to use 'coverage' because we augment the definition of recall in calculating coverage by requiring that a word in the caption appear no later than 10 seconds after the word in the ground truth, and not before it, to count. Similarly, *precision* is the fraction of words in the caption that appear in the ground truth within 10 seconds.

Finally, for real-time captioning, latency is also important. Calculating latency is not straightforward because workers' noisy partial captions differ from the ground truth. In this paper, we measure latency by first aligning the test captions to the ground truth using the Needleman-Wunsch sequence alignment algorithm [24], and then averaging the latency of all matched words. In order for DHH individuals to participate in a conversation or in a lecture, captions must be provided quickly (within about 5 seconds) [30].

BACKGROUND

In addition to alternative approaches to real-time captioning, SCRIBE also builds from prior work in (i) real-time human computation and (ii) multiple sequence alignment.

Real-Time Human Computation

Historically, people with disabilities have attempted to solve their accessibility problems with the support of people in their community [4]. Increasing connectivity has made remote services possible that once required human supporters to be co-located. Real-time captioning by non-experts is a type of human computation [28], which has been shown to be useful in many areas, including writing and editing [3], image description and interpretation [5, 29], 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 [22, 19]. For instance, the ESP Game uses answer agreement [29] and Soylent uses the multiple-step find-fix-verify pattern [3]. SCRIBE presents a model of crowdsourcing that uses workers in parallel, not sequentially, to improve performance on real-time tasks.

Human computation has been applied to offline transcription with great success [1], but has not been previously applied to real-time captioning. 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 [23]. SCRIBE enables real-time transcription from multiple non-experts and uses crowd agreement to ensure quality.

Real-time human computation has only started to be explored. VizWiz [5], was one of the first systems to target nearly real-time response from the crowd. It introduced a queuing model to help ensure that workers were available quickly on-demand. For SCRIBE to be available on-demand multiple users are required 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 [29, 6]. Adrenaline combines the concepts of queuing and waiting to recruit crowds (groups) in less than 2 seconds from existing sources of crowd workers [2]. SCRIBE also 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 [20]. Each crowd worker 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 could be viewed as an instance of an input mediator. A primary difference is that while Legion selected from individual user inputs, we use a synthesis of the crowd's input to create the final stream.

Multiple Sequence Alignment (MSA)

Our transcription problem is an instance of the general problem of multiple sequence alignment with an additional merging step. Much work in bioinformatics concentrates on aligning multiple related sequences of nucleotides and other chemical compounds. The main biological motivation for this process is to gain insights into the relationships between organisms based on their respective genomes. While finding the globally optimal alignment is an NP-hard problem [13] (in our case, the runtime is exponential in the number of workers), effective approximate solutions have been developed. One of our input combiners extends the MUSCLE package [11] with a language model for English in order to align partial captions in a meaningful way (Figure 3).

SCRIBE

SCRIBE gives users on-demand access to real-time captioning from groups of non-experts via their laptop or mobile devices (Figure 1). When a user starts SCRIBE, it immediately

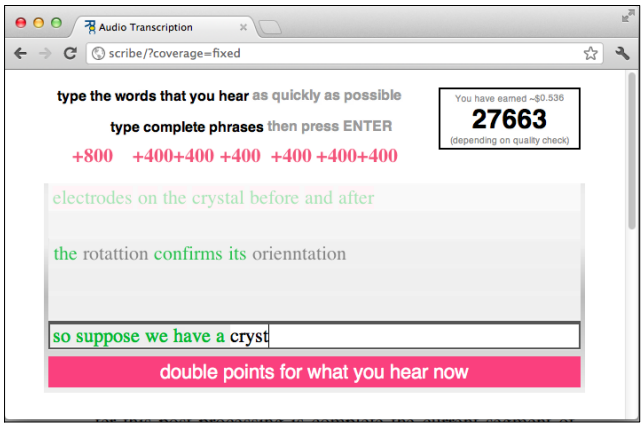

Figure 4. The worker interface encourages captionists to type audio quickly by locking in words soon after they are typed. To encourage coverage of specific segments, visual and audio cues are presented, and the volume is reduced during off periods. Rewards are increased for words typed during these segments.

begins recruiting workers for the task from Mechanical Turk, or a pool of volunteer workers, using quikTurkit [5]. When users want to begin captioning audio, they press the start button, which forwards audio to Flash Media Server (FMS) and signals the SCRIBE server to begin captioning.

Workers are presented with a text input interface designed to encourage real-time answers and increase global coverage (Figure 4). A display shows workers their rewards for contributing in the form of both money and points. In our experiments, we paid workers $0.005 for every word the system thought was correct. As workers type, their input is forwarded to an *input combiner* on the SCRIBE server. The input combiner is modular to accommodate different implementations without needing to modify SCRIBE. The combiner and interface are discussed in the next section.

The user interface for SCRIBE presents streaming text within a collaborative editing framework (see Figure 5). SCRIBE's interface masks the staggered and delayed format of real-time captions with a more natural flow that mimics writing. In doing this, the interface presents the merged inputs from the crowd workers via a dynamically updating web page, and allows users to focus on reading, instead of tracking changes. SCRIBE also supports real-time editing by users or other crowds. The web interface visually presents relevant information, such as the confidence of each spelling and possible word and arrangement alternatives. These cues both reduce the attention that must be paid to the editing process, and encourage users to focus their efforts on specific problems in the caption. For example, conflicted words or spellings are highlighted and, when selected, alternatives are displayed and can be agreed with or new answers can be added. These updates are then forwarded back to the combiner.

When users are done, pressing the stop button will end the audio stream, but let workers complete their current transcription task. Workers are asked to continue working on other audio for a time to keep them active in order to reduce the response time if users need to resume captioning.

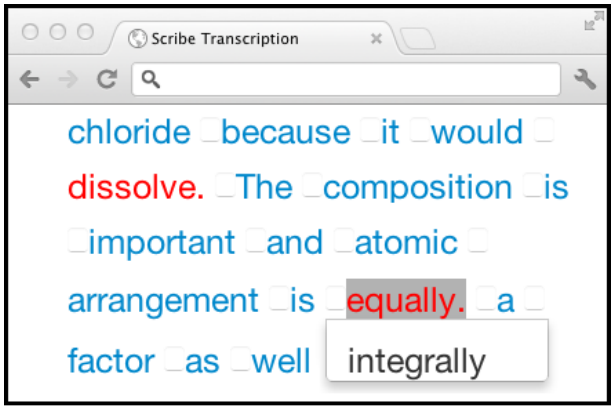

Figure 5. The web-based interface for users to see and correct the live caption stream returned by SCRIBE.

Collaborative Editing

We expect that multiple users will often want to use SCRIBE to generate captions for the same event. SCRIBE's interface supports this by allowing users to share the link to the web interface for a given session to view the generated captions. This allows more captionists from the worker pool to be used for a single task, improving performance. Additionally, the joint session acts as a collaborative editing platform. Each participant in this shared space can submit corrections to the captions, adding their individual knowledge to the system.

Adjustable Quality

SCRIBE allows for placing emphasis on either coverage or precision. 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 precision, but reduce the coverage since not all workers will agree on all words. We allow users to either let the system choose a default balance, or select their own balance of precision versus coverage by using a slider bar in the user interface. Workers can select from a continuous range of values between 'Most Accurate' and 'Most Complete' which are mapped to settings within the combiner.

Co-Evolving Systems

Our solution to the transcription problem is two-fold. First, we have designed an interface that facilitates real-time captioning by non-experts and encourages covering the entire audio signal. Second, we have developed an algorithm for merging partial captions to form one final output stream. The interface and algorithm have been developed to address these

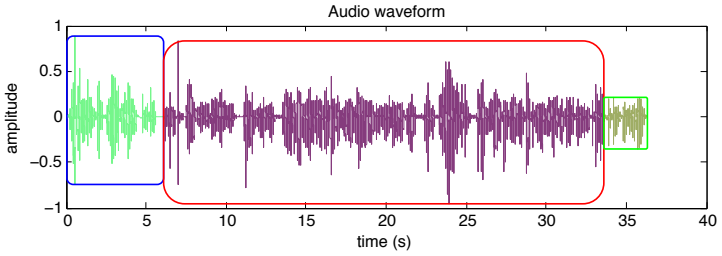

Figure 6. Audio segmented using our kernel SVM method. Each segment contains audio from one speaker, and is sent to different workers.

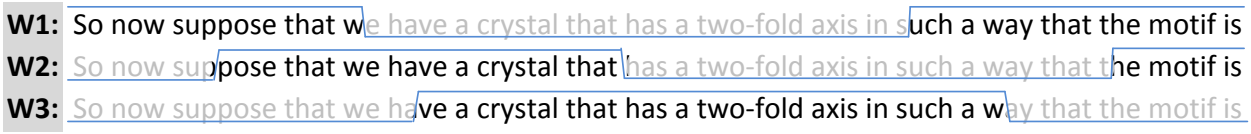

Figure 7. SCRIBE encourages workers to type different portions of the input speech by raising and lowering the volume of the audio, as depicted visually here. Artificially adjusting saliency while streaming the signal improves overall coverage.

problems jointly. For instance, determining where each word in a partial caption fits into the final transcript is difficult, so the interface was designed to encourage workers to type continuous segments then signal breaks. We detail the co-evolution of the worker interface and algorithm for merging partial captions in order to form a final transcript. By developing the interface and merging algorithm to best suit each other, we can create a system that efficiently uses the imperfect captioning abilities of workers to create transcripts.

Transcribing a Dialogue

Interleaving different speakers adds an additional layer of complexity to the transcription task. ASR attempts to adapt to a particular speaker's voice; however, if speakers constantly change, this adjustment often reduces the quality of the transcription further [7]. In order to address this problem and enable accurate transcriptions of conversations, even those between individuals with very different speaking styles, systems must be able to either dynamically adjust to the variances, or isolate the separate components of the audio.

Though this paper has focused on transcribing a single person speech so far, SCRIBE can handle dialogues using automated speaker segmentation techniques (Figure 6). We combine a standard convolution-based kernel method for identification of distinct segments in a waveform with a one-class support vector machine (SVM) classifier to each segment to assign it a speaker ID [18]. Prior work has shown such segmentation techniques to be accurate even in the presence of severe noise, such as when talking on a cell phone while driving [26, 18]. The segmentation allows us to *decompose* a dialogue in real-time, then caption each part individually.

AN INTERFACE FOR REAL-TIME CAPTIONING

The first component of SCRIBE is the interface that non-expert captionists will use to provide their captions (Figure 4). The web-based interface streams audio to the captionists who are instructed to type as much of it as they can. Furthermore, Workers are told to separate contiguous sequences of words by pressing enter. Knowing which word sequences are likely to be contiguous can help later when recombining the partial captions from multiple captionists.

To encourage real-time entry of captions, the interface "locks in" words a short time after they are typed (800 milliseconds). New words are identified when the captionist types a space after the word, and are sent to the server. The delay is added to allow workers to correct their input while adding as little additional latency as possible to it. When the captionist presses enter (or following a 2 second timeout during which they have not typed anything), the line is confirmed and animates upward. During the 10 second trip to the top of the display, words that SCRIBE determines were entered correctly (by either a spelling match or overlap with another worker) are colored green. When the line reaches the top, a point score is calculated for each word based on its length and whether it has been determined to be correct.

To recover the true speech signal, non-expert captions must *cover* all of the words in that signal. A primary reason why the partial transcriptions may not fully cover the true signal relates to *saliency*, which is defined in a linguistic context as "that quality which determines how semantic material is distributed within a sentence or discourse, in terms of the relative emphasis which is placed on its various parts." [14]. Numerous factors influence what is salient, and so it is likely to be difficult to detect automatically. Instead, we inject saliency artificially by systematically varying the volume of the audio signal that captionists hear. The web-based interface that we use is able to vary the volume over a given a period with an assigned offset. It also displays visual reminders of the period to further reinforce this notion. Figure 7 shows how the volume can be systematically varied to maximize coverage over the whole signal.

In preliminary work, we instead divided the audio signal into segments that we gave to individual workers to transcribe. We found a number of problems with this approach. First, workers tended to take longer to provide their transcriptions as it took them a bit to get into the flow of the audio. A continuous stream avoids this problem. Second, the interface seemed to encourage workers to favor quality over speed, whereas a stream that does not stop is a reminder of the real-time nature of the transcription. The continuous interface was designed using an iterative process involving tests with 57 remote and local users with a range of backgrounds and typing abilities. These tests demonstrated that workers generally tended to provide chains of words rather than disjoint words, and that workers needed to be informed of the motivations behind aspects of the interface to use them properly.

A non-obvious question is what the period of the volume changes should be. In our experiments, we chose to play the audio at high volume for four seconds and then at a lower volume for six seconds. This seems to work well in practice, but it is likely that it is not ideal for everyone. Our experience suggested that keeping the on period short is preferable even when a particular worker was able to type more than the period because the latency of a worker's input tended to go up as they typed more consecutive words.

REAL-TIME INPUT COMBINER

The second primary component of SCRIBE is the merging server, which uses a selectable algorithm to combine partial

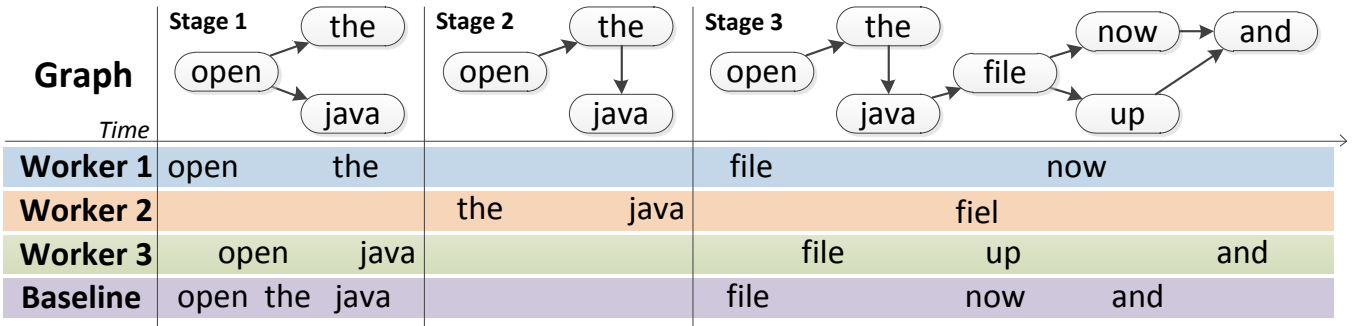

Figure 8. Graph building for three workers captioning the spoken sentence *"Open the Java code up now and . . . "*. The top section shows the current state of the graph after each stage. The bottom shows the corresponding input. Note that Worker 2 spelling 'file' incorrectly does not adversely affect the graph since a majority of the workers still spell it correctly.

captions into a single output stream. A naive approach would be to simply arrange all words in the order in which they arrive, but this approach does not handle reordering, omissions, and inaccuracy within the input captions. Instead, our algorithm combines timing information, observed overlap between the partial captions, and models of natural language to inform the construction of the final output. In this section, we first describe our straightforward adaptation of multiple sequence alignment (MSA) to the caption combination problem, and then describe the reformulation of this algorithm that enables $\mathcal{S}$CRIBE to overcome many of the run-time and scalability issues associated with MSA.

Multiple Sequence Alignment

To use MSA, we replaced the mutation model for nucleotides used in the MUSCLE bioinformatics package with a spelling error model based on the physical layout of a keyboard. For example, when a person intends to type [a], he is more likely to mistype [q] than [m]. The model is further augmented with context-based features learned from spelling corrections drawn from the revisions of Wikipedia articles.

Learning a substitution matrix for each pair of characters along with character insertion and deletion penalties allows us to run a robust optimization technique that finds a near-optimal joint alignment [11]. Even though finding the best alignment is computationally expensive, our system operates in real-time by leveraging dynamic programming and approximations. Once the partial captions are aligned, we need to merge them into a single transcript, as shown in Figure 3. We perform a majority vote for each column of aligned characters, then remove the gaps in a post-processing step. The entire computation takes only a few seconds for input several minutes in length. To apply our MSA model to longer audio signals while maintaing the real-time aspect of the system, a sliding window can be used to bound the runtime.

Online Dynamic Sequence Alignment

In order to achieve the response-time and scalability required for real-time captioning of longer sessions, we create a version of MSA that aligns input using a graphical model. Worker captions are modeled as a linked list with nodes containing equivalent words aligned based on sequence order submission time. As words are added, consistent paths arise. We maintain the longest self-consistent path between any two nodes to avoid unnecessary branching. Figure 8 shows an example of the graph building process.

Reconstructing the Stream

Using a greedy search of the graph, in which we always follow the highest weight edge (a measure of the likelihood of two words appearing in a row), we derive a transcript in real-time. The greedy search traverses the graph between inferred instances of words by favoring paths between word instances with the highest levels of confidence derived from worker input and n-gram data. Ideally, we imagine using n-gram corpora tailored to the domain of the audio clips being transcribed, either by generating them in real time along with our graph model, or by pre-processing language from similar contexts. Specific n-gram data should allow more accurate transcriptions of technical language by improving the accuracy of the model used to infer word ordering in ambiguous cases.

The greedy graph traversal favors paths through the graph with high worker confidence, and omits entirely words contained within branches of the graph that contain unique instances of words. A post-processing step augments the initial sequence by adding into it any word instances with high worker confidence that were not already included. Because the rest of the branch is not included, these words can be disconnected from words adjacent in the original audio The positioning of these words are added back into the transcript by considerng the most likely sequence given their timestamps and the bigrams and trigrams that result from their insertion into the transcription. After this post-processing is complete the current transcript is forwarded back to the user.

Run-time

Each time a worker submits new input, a node is added to the worker's input chain. A hash map containing all existing unique words spoken so far in the stream is then used to find a set of equivalent terms. The newest element can always be used since the guarantee of increasing timestamps means the the most recent occurrence will always be the best fit. The match is then checked to see if a connection between the two nodes would form a back-edge. Using this approach allows us to reduce the runtime from worst-case $O(n^k)$ to $O(n)$. We can further reduce the runtime of this algorithm by limiting the amount of data stored in the graph at any one time – since we can safely assume that the latency with which any worker submits a response is limited. In practice, a 10 second time

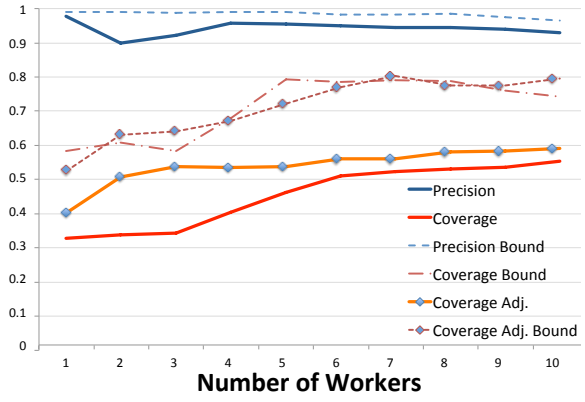

Figure 9. Precision and coverage plots for MSA with artificial saliency adjustments (Adj.) and without, and their theoretical upper bounds. We see that the adjustments significantly improve SCRIBE's coverage.

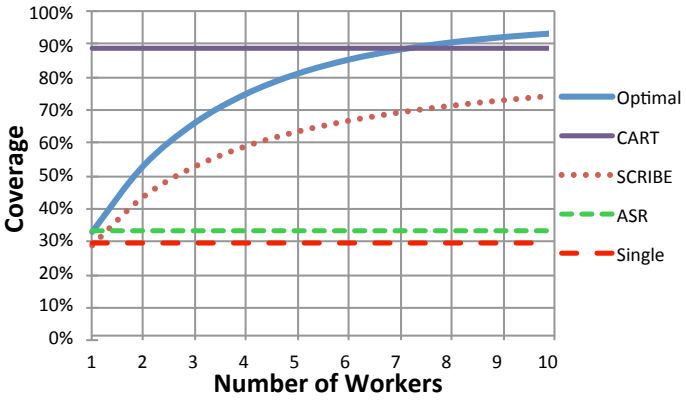

Figure 10. Optimal coverage reaches nearly 80% when combining the input of four workers, and nearly 95% with all 10 workers. This demonstrates that captioning audio in real-time with non-experts is feasible.

window is effective, though SCRIBE was able to incrementally build the graph and generate output within a few milliseconds for time windows beyond 5 minutes.

Our approach does not have the optimality guarantees of offline MSA; however, we show that this approach is effective in the real-time captioning domain, due to properties such as the relatively low frequency of repeated words. In the future, we will extend this model to handle general online sequence alignment, given statistical information about the domain.

EXPERIMENTS

We ran experiments to test the ability of non-expert captionists drawn from both local and remote crowds to provide captions that cover speech, and then evaluate our approaches for merging the input from these captionists into a final real-time transcription stream. We collected a data set of speech selected from freely available lectures on MIT OpenCourseWare (http://ocw.mit.edu/courses/). These lectures were chosen because one of the main goals of SCRIBE is to provide captions for classroom activities, and because the recording of the lectures roughly matches our target as well – there is a microphone in the room that often captures 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, and corrected these to obtain a completely accurate baseline.

Our study used 20 local participants. Each participant captioned 23 minutes of aural speech over a period of approximately 30 minutes. Participants first took a standard typing test and averaged a typing rate of 77.0 WPM (*SD*=15.8) with 2.05% average error (*SD*=2.31%). We then introduced participants to the real-time captioning interface, and had them caption a 3-minute clip using it. Participants were then asked to caption the four 5-minute clips, two of which were selected to contain saliency adjustments.

One key question as to the effectiveness of our approach is whether or not groups of non-experts can effectively *cover* the speech signal. If some part of the speech signal is never typed then it will never appear in the final output, regardless of merging effectiveness. We also measure the precision and WER of the captions, which influence the overall readability.

Multiple Sequence Alignment

Figure 9 shows precision and coverage plots for the MSA algorithm using multiple workers. The metrics are calculated at a character level and averaged over all subsets of workers and over all audio clips. We see that precision is not significantly impacted by the number of workers, whereas coverage significantly improves as we use additional captionists. The theoretical upper bounds in Figure 9 for both precision and coverage shows what can be attained by a fully informed MSA with access to ground truth. This shows that SCRIBE is extracting nearly all of the information it can in terms of precision, but could improvement in terms of coverage. Narrowing this gap is a key direction of our future work.

Note that adjusting the saliency dramatically improves coverage, as compared to no adjustments (Figure 9). For example, only 2 workers are needed to achieve 50% coverage when using adjustments, while 6 workers are required to produce the same level with no adjustments. We discuss this in more in more detail later in this section.

Real-time Combiner

In our tests, an average worker achieved 29.0% coverage, ASR achieved 32.3% coverage, CART achieved 88.5% coverage and SCRIBE reached 74% out of a possible 93.2% coverage using 10 workers (Figure 10). Groups of workers also had an average latency of 2.89 seconds (including network latency), significantly improving on CART's latency of 4.38 seconds. Results show that SCRIBE is easily able to outperform the coverage of both ASR and lone workers. Additionally, the number and types of errors in ASR captions make it far less useful to a user trying to comprehend the content of the audio. Our naive approach (optimal coverage curve), which combines input based only on timestamp, shows that multiple workers have the potential to surpass CART with respect to coverage. Latency is also improved since multiple concurrent workers will naturally interleave answers, so that no one worker falling behind will delay the whole system, unlike a single-captionist approach such as CART.

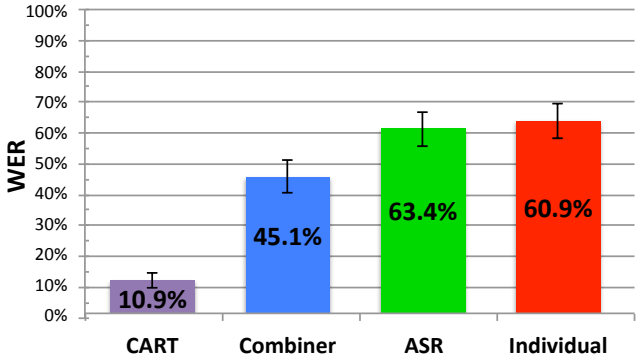

Figure 11. Word error rate (WER) results for the captioning systems. Our combiner outperforms both individual workers and ASR.

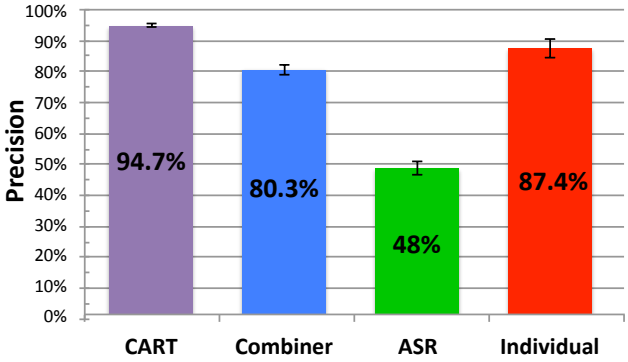

Figure 12. Precision results for the captioning systems.

Figure 11 shows WER results from the captioning systems. For this example, we tuned our combiner to balance coverage and precision, getting an average of 66% and 80.3% respectively. As expected, CART outperforms the other approaches. However, our combiner presents a clear improvement over both ASR and a single worker. The precision of each system is shown in Figure 12. Because the input of multiple workers is merged, our combiner occasionally includes repeated words, which are marked as incorrect by our metrics. The difference is that SCRIBE can be tuned to reach arbitrarily high accuracies, at the expense of coverage. This tradeoff is discussed in the next section.

Adjusting Tradeoffs

The input combiner is parameterized and allows users to actively adjust the tradeoff between improving coverage and improving precision while they are viewing the captions. To increase coverage, the combiner reduces the number of workers required to agree on a word before including it in the final caption. To increase accuracy, the combiner increases the required agreement. Figure 13 shows tradeoffs users can make by adjusting these settings.

Saliency Adjustment

We also tested the interface changes designed to encourage workers to type different parts of the audio signal. For all participants, the interface indicated that they should be certain to type words appearing during a four second period followed by six seconds in which they could type if they wanted to. The 10 participants who typed using the modified version of the interface for each 5-minute file were assigned offsets ranging from 0 to 9 seconds.

In our experiments, we found that the participants consistently typed a greater fraction of the text that appeared in the periods in which the interface indicated that they should. For the electrical engineering clip, the difference was 54.7% (*SD*=9.4%) for words in the selected periods as compared to only 23.3% (*SD*=6.8%) for word outside of those periods. For the chemistry clips, the difference was 50.4% (*SD*=9.2%) of words appearing inside the highlighted period as compared to 15.4% (*SD*=4.3%) of words outside of the period.

Mechanical Turk

We were curious to see if the interface and captioning task would make sense to workers on Mechanical Turk since we would not be able to provide directions in person. We used quikTurkit to recruit a crowd of workers to caption the four clips (20 minutes of speech). Our HITs (Human Intelligence Tasks) paid $0.05 and workers could make an additional $0.002 bonus per word. We asked workers to first watch a 40-second video in which we describe the task. In total, 18 workers participated, at a cost of $13.84 ($36.10 per hour).

Workers collectively achieved a 78.0% coverage of the audio signal. The average coverage over just three workers was 59.7% (*SD*=10.9%), suggesting we could be conservative in recruiting workers and cover much of the input signal. Participating workers generally provided high-quality captions, although some had difficulty hearing the audio. Prior work has shown that workers remember the content of prior tasks, meaning that as more tasks are generated, we expect the size of the trained pool of workers available on Mechanical Turk will increase [21]. The high cost of alternatives means that we can pay workers well and still provide a cheaper solution.

DISCUSSION AND FUTURE WORK

Our results show that groups of non-experts have the ability to achieve better coverage and less latency than a professional captionist. Furthermore, we can encourage workers to focus

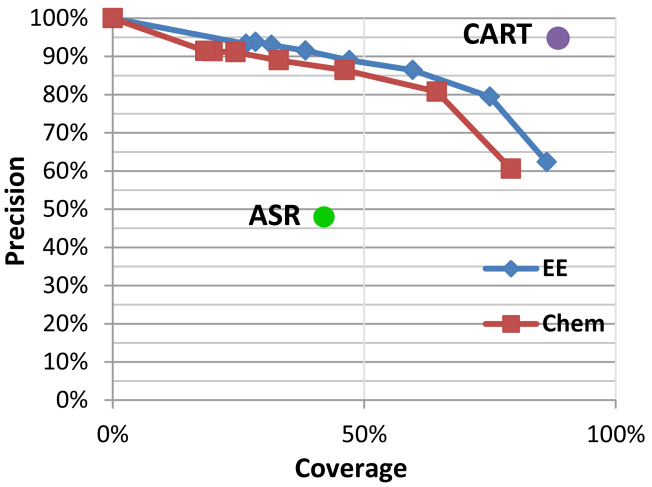

Figure 13. Precision-coverage curves for the electrical engineering (EE) and chemistry (Chem) lectures using different combiner parameters with 10 workers. In general, increasing coverage reduces accuracy.

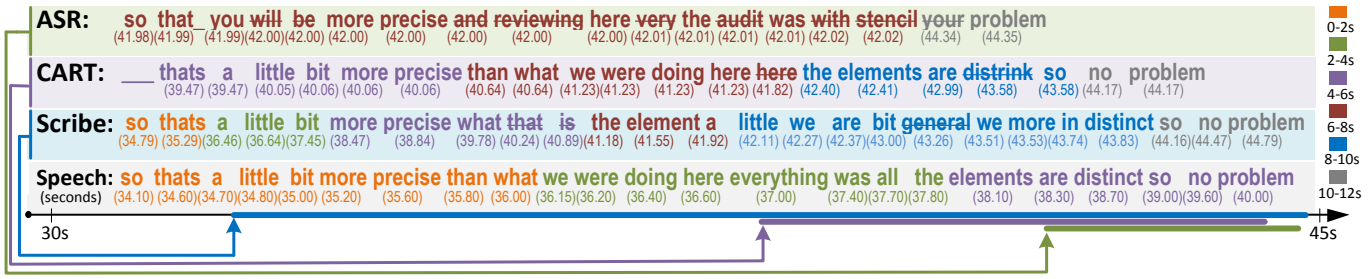

Figure 14. Captions by CART, SCRIBE, and ASR illustrating common errors and tradeoffs. Words are colored based on 2-second time spans. ASR and (to a lesser extent) CART captions arrive in bursts, disrupting the 'flow'. The timeline shows the duration of output from each system. SCRIBE repeats some content, but creates a significantly more accurate and meaningful transcript than ASR.

on specific portions of the speech to improve global coverage, and it is possible to recombine partial captions while effectively balancing coverage and precision. This presents opportunities for future work that aims to improve the quality, availability, and cost of real-time captioning.

Covering the Input Signal

For our approach to work, the captions provided by workers must fully cover the original content. We have shown that a single worker is unable to cover even one third of the speech, but groups can collectively cover over 93%. Many of the remaining gaps measured are due to differences in phrasing (e.g. "anybody" vs. "anyone") that do not affect the usability of the system, meaning our combiner will likely perform better in real settings than the results indicate.

We have also shown that we can effectively encourage workers to cover portions of speech that we want, and future work will seek to further improve the interface and worker feedback to increase coverage even more with fewer workers. In particular, our current approach does not reward workers sufficiently for typing long or complicated words, and so these are often missed. For instance, the word "non-aqueous" was used in a clip about chemistry but no workers typed it.

Our saliency adjustment is currently defined to be the same for all workers.Personalizing the on and off periods for each worker could improve results while requiring fewer workers by letting more experienced and skilled workers be used to their full potential, while not overwhelming newer workers. Several workers (both local and on Mechanical Turk), made a point of mentioning that they enjoyed the captioning task and were interested in continued participation as a SCRIBE captionist. Reporting scores, most common mistakes, and other information back to workers may help them improve their typing ability, resulting in a more skilled set of available worker in future tasks. Adding multiple classes of continuous workers will make some alternate approaches more viable. For example, using a separate group of workers to provide information such as the number of words said reduces the amount of uncertainty that needs to be handled by the model.

We expect that leveraging a more rich probabilistic language model will help improve SCRIBE's performance. Currently, the input combination algorithm is agnostic to part-of-speech tags, sentence structure, and other linguistic features. A unified conditional random field model [25] that finds the most likely sequence of words given partial worker captions *along with* language-based features, semantics, and context, will not only yield better accuracy, but may even produce a transcript that is more comprehensible than verbatim.

Leveraging Hybrid Workforces

ASR is not reliable on its own, but may be useful as a supplement to (or eventually replacement of) the input of human workers by including ASR systems as contributors. This is especially useful with small groups of workers. For example, supplementing the captions of a single worker with ASR brings the optimal coverage up from 28.5% to 55.3%, and for 10 workers, from 93.2% to 95.1%. Additionally, our tests found that the errors made by ASR differ from those made by humans: ASR tends to replace words with others that are phonetically similar, but differ in meaning, while humans tend to replace words with ones that have the same meaning, but may sound different. Thus, we expect that using a hybrid approach will be more effective than either humans or ASR alone by using these difference to increase coverage while maintaining accuracy even with lower agreement. For example, if a human worker and ASR both agree on a word and its position, it is more likely to be correct than if two workers (who make similar mistakes) agree on it. As ASR improves, SCRIBE can reduce its reliance on human contributors, and transition towards a fully automatic system that is more robust and faster than any single ASR.

Real-World User Testing

The utility of captioning in real-world settings is dependent on both the information content retained and the ease by which the captions can be read. Spoken language often flows differently than written text. Speakers pause, change subjects, and repeat phrases – all of which can make exact transcripts difficult to read (Figure 14). A language model may help make SCRIBE captions more readable. Captioning methods such as C-Print paraphrase speech to keep up, often making them easier to read but also leaving out content. ASR often produces nonsensical errors, which is likely to confuse users, even though ASR can appear competitive on automatic metrics. Models could be individually customized to a user's preferred style. We plan to test these tradeoffs in real settings with DHH and hearing students.

Extending to New Problems

SCRIBE uses a new model of human computation to allow groups to collectively out-perform individuals on a difficult human performance task (real-time captioning). The general

idea of combining partial inputs automatically may be extended to new domains and new problems thus far limited by individuals' abilities. Groups composed of multiple contributors collectively possess superior motor and cognitive abilities, but how to effectively harness those abilities in general remains an open research question. We believe the model introduced in this paper is a valuable first step.

CONCLUSIONS

In this paper, we have introduced a new model of crowdsourcing in which multiple workers provide simultaneous input that is then combined into a final answer. This approach enables workers to collectively complete tasks that they may otherwise be unable to perform individually. As a specific instance of this framework, we presented LEGION:SCRIBE, an end-to-end system enabling real-time captioning by non-experts. We showed that groups of workers can outperform both individuals and ASR in terms of coverage, precision, and latency, and introduced a new algorithm for aligning and merging partial text captions as they arrive. We have furthermore demonstrated that groups of non-experts can achieve better coverage and latency than a professional captionist, and that we can encourage them to focus on specific portions of the speech to improve global coverage. Finally, we have shown that it is possible to recombine partial captions and effectively tradeoff coverage and precision. Our results demonstrate the feasibility of this approach and open a number of interesting opportunities for future research.

ACKNOWLEDGMENTS

This work was supported by National Science Foundation grants #IIS-1149709 and #IIS-1016486, and by Google.

REFERENCES

1. Y. C. Beatrice Liem, H. Zhang. An iterative dual pathway structure for speech-to-text transcription. In *Proc. of the 3rd Workshop on Human Computation*, HCOMP 2011. 2011.
2. M. S. Bernstein, J. R. Brandt, R. C. Miller, and D. R. Karger. Crowds in two seconds: Enabling realtime crowd-powered interfaces. In *Proc. of the 24th annual ACM Symp. on User Interface Software and Technology*, UIST '11, p33–42. 2011.
3. 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 *Proc. of the 23rd Annual ACM Symp. on User Interface Software and Technology*, UIST '10, p313–322. 2010.
4. J. P. Bigham, R. E. Ladner, and Y. Borodin. The Design of the Human-Backed Access Technology *Conf. on Computers and Accessibility*, ASSETS 2011, p3–10. 2011.
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 *Proc. of the 23rd Annual ACM Symp. on User Interface Software and Technology*, UIST '10, p333–342. 2010.
6. L. Chilton. Seaweed: A web application for designing economic games. Master's thesis, MIT, 2009.
7. M. Cooke, P. Green, L. Josifovski, and A. Vizinho. Robust automatic speech recognition with missing and unreliable acoustic data. *Speech Comm.*, 34(3):267–285, 2001.
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 *IEEE Intl. Conf. on Acoustics, Speech and Signal Processing*, ICASSP 2008, p5121–5124. 2008.
10. F. J. Damerau. A technique for computer detection and correction of spelling errors. In *Commun. ACM.*, 7(3):171–176. March 1964.
11. R. Edgar. Muscle: multiple sequence alignment with high accuracy and high throughput. *Nucleic acids research*, 32(5):1792–1797. 2004.
12. 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 Ed. Research Assoc. Annual Meeting*, 2008.
13. J. Felsenstein. Inferring phytogenies. Sinauer Associates, Sunderland, Massachusetts. 2004.
14. J. L. Flowerdew. Salience in the performance of one speech act: the case of definitions. *Discource Processes*, 15(2):165–181. April-June 1992.
15. J. Holt, S. Hotto, and K. Cole. Demographic Aspects of Hearing Impairment: Questions and Answers. 1994. http://research.gallaudet.edu/Demographics/factsheet.php.
16. 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 *Intl. Conf. on Spoken Lang. Processing*, ICSLP-2002, p1757–1760. 2002.
17. C. Jensema, R. McCann, S. Ramsey. Closed-captioned television presentation speed and vocabulary. In *Am Ann Deaf.* 141(4):284–92. October 1996.
18. H. Kadri, M. Davy, A. Rabaoui, Z. Lachiri, N. Ellouze, et al. Robust audio speaker segmentation using one class SVMs. In *Proc of the European Signal Processing Conf.*, EUSIPCO 2008. 2008.
19. A. Kittur, B. Smus, S. Khamkar and R. E. Kraut. Crowdforge: Crowdsourcing complex work. In *Proc. of the 24th Symp. on User Interface Software and Technology*, UIST '11, p43–52. 2011.
20. 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 ACM Symp. on User Interface Software and Technology*, UIST '11, p23–32. 2011.
21. W. S. Lasecki, S. White, K. I. Murray, and J. P. Bigham. Crowd memory: Learning in the collective. In *Proc. of Collective Intelligence 2012*, CI 2012. 2012.
22. G. Little, L. B. Chilton, M. Goldman, and R. C. Miller. Turkit: human computation algorithms on mechanical turk. In *Proc. of the 23rd ACM Symp. on User interface software and technology*, UIST '10, p57–66. 2010.
23. T. Matthews, S. Carter, C. Pai, J. Fong, and J. Mankoff. Scribe4me: evaluating a mobile sound transcription tool for the deaf. In *Proc. of the 8th Intl. Conf. on Ubiquitous Computing*, UbiComp '06, p159–176. 2006.
24. S. B. Needleman and C. D. Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. In *Journal of Molecular Biology*. 48 (3):443-53. 1970.
25. C. Sutton and A. McCallum. *An introduction to conditional random fields for relational learning*. Introduction to statistical relational learning. MIT Press, 2006.
26. A. Tritschler and R. Gopinath. Improved speaker segmentation and segments clustering using the bayesian information criterion. In *Sixth European Conf. on Speech Communication and Technology*, 1999.
27. C. Van Den Brink, M. Tijhuis, G. Van Den Bos, S. Giampaoli, P. Kivinen, A. Nissinen, and D. Kromhout. Effect of widowhood on disability onset in elderly men from three european countries. *Journal of the American Geriatrics Society*, 52(3):353–358. 2004.
28. L. von Ahn. Human Computation. Ph.D. Thesis. 2005.
29. L. von Ahn and L. Dabbish. Labeling images with a computer game. In *Proc. of the Conf. on Human Factors in Computing Systems*, CHI '04, p319–326. 2004.
30. 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.
31. 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.