Real-time Captioning by Groups of Non-Experts

DRAFT – Walter Lasecki, Christopher Miller, Adam Sadilek, Andrew AbuMoussa, and Jeffrey P. Bigham – DRAF

ABSTRACT

Real-time captioning provides deaf and hard of hearing people immediate access to the spoken word and enables participation in dialogue with hearing people. Low latency is often critical because it allows speech to be accurately paired with visual cues. Currently, the only reliable source of real-time transcriptions are expensive stenographers trained to use specialized keyboards who must be recruited in advance. Automatic speech recognition (ASR) is less expensive and available on-demand, but its low accuracy, high sensitivity to noise, and need for training beforehand, render its captions 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 SCRIBE, an end-to-end system that allows deaf people to request captions at any time. The interface that we built to collect captions was designed to encourage coverage of the entire audio stream and we developed an algorithm to merge partial captions into a single output stream in real-time. Our evaluation with 20 local participants demonstrates the potential of our approach and highlights areas for future work, showing that 10 workers are able to cover over 93.5% of an audio stream with an average per-word latency of less than 2.5 seconds.

ACM Classification: H5.2 [Information interfaces and presentation]: User Interfaces. - Graphical user interfaces.

General terms: Design, Human Factors, Experimentation

Keywords: Real-time, captioning, transcription, deaf, hard of hearing, text alignment

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 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 [27]. 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 SCRIBE, an end-to-end system that we have developed to allow groups of non-experts to transcribe speech in real time. Each individual is unable to type fast enough to keep up with natural speaking rates, and so SCRIBE computationally combines these inputs into a final caption stream.

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 [27]. 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 speakers have not trained the ASR, when captioning multiple speakers, or when not using a high-quality microphone located close to the speaker [12, 8]. 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].

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, sign language interpreters are also expensive and difficult to schedule.

Non-expert captionists can be drawn from more elastic worker 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 [3, 6]. 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 [27]. As one example, SCRIBE will allow student workers to serve as non-expert captionists for $8-12 per hour (a typical work-study pay). At that rate 10

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

UIST'12, October 7-10, 2012, Cambridge, MA, USA.

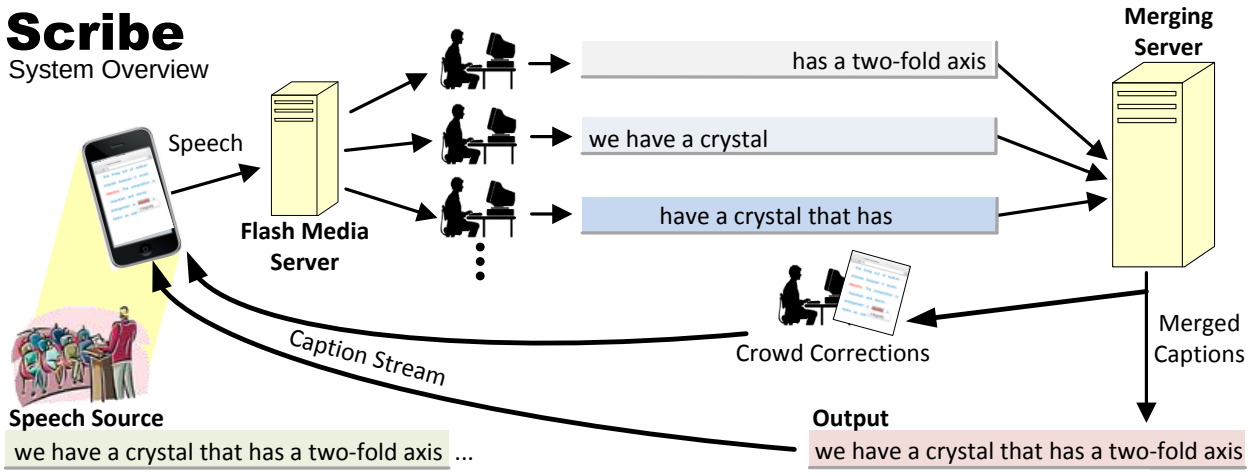

Figure 1: $\mathcal{S}$CRIBE allows users to caption audio on their mobile device. The audio is sent to multiple amateur captionists in real-time who use the $\mathcal{S}$CRIBE web-based interface to caption as much of the audio as they can. These partial captions are sent to the merging server to be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

students could be hired in place of one professional captionist, although we expect to need far fewer.

$\mathcal{S}$CRIBE may also benefit people who are not DHH as well. For example, students have more time to focus and interact in class because detailed notes are automatically generated for them. They can also perform full-text search over the transcript to quickly identify segments of the audio recording they want to listen to. Furthermore, we all are subject to a situational disability from time to time [24]. 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 real-time captioning by groups of non-experts, and develop $\mathcal{S}$CRIBE, an end-to-end system that has advantages in availability, cost and accuracy as compared to alternative solutions.
- We demonstrate the feasibility of our approach by showing that non-expert captionists can collectively cover speech at high rates.
- We show that $\mathcal{S}$CRIBE can produce transcripts that both cover more of the input signal and are more accurate than ASR and each constituent worker.

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 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 achieve the 225 words per minute (WPM) needed to consistently transcribe speech at natural speaking rates.

Non-Verbatim Systems: In response to the cost of CART, computer-based macro expansion services like C-Print were introduced [27]. 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 of 150 WPM, and cannot produce a verbatim transcript. $\mathcal{S}$CRIBE employs captionists with no training and compensates for slower typing speeds and lower accuracy by combining the efforts of multiple captionists at once.

Automated Speech Recognition: ASR works well in ideal situations with high-quality audio equipment, but degrades quickly in many real 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, 10]. ASR systems can require substantial computing power and special audio equipment to work well, which lowers availability. In our experiments, we used Dragon Naturally Speaking 11.5 for Windows.

Another approach to transcription 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 has been extensively trained for their voice [16]. Respeaking works well for offline transcription, but simultaneous speaking and listening requires professional training. By contrast, $\mathcal{S}$CRIBE enables non-experts to contribute to transcription without special training or skill.

Metrics for Evaluation

Determining the quality of captioning is difficult [28]. The most common method is the word error rate (WER). This metric performs a best-fit alignment between the test captions and the ground truth captions. The WER is then calculated as the sum of the substitutions S, the deletions D, and the insertions I 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. Humans infer meaning from context, influencing their prior probability toward those words that make sense in context. We anticipate this will make SCRIBE more usable than automated systems even when traditional metrics are similar. Figure 2 gives an example of the confusing 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 [28].

We define two other automatic methods that help to characterize the performance of real-time captioning that we believe are particularly useful in understanding potential of various approaches. 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 choose to use coverage because we augment the definition of recall a bit in calculating it. In particular, in calculating coverage we require that a word in the test signal appear no later than 15 seconds after the word in the ground truth signal to count. We define *precision* similarly. Precision is the fraction of words in the test caption stream that appear in the ground truth within a time window of 15 seconds. As compared to WER, coverage and precision are looser measures of alignment.

Finally, for real-time captioning, latency is also important. 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. In order for DHH individuals to participate in a conversation or in a lecture, captions must be provided quickly (within about 5 seconds [27]).

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 at sufficient speed and accuracy to generate an exact transcript of spoken content in real-time. Therefore, it is unlikely to get a decent coverage rate using single workers to caption extended audio clips, and instead rely on multiple workers typing different parts of the audio signal. Later, we describe how our interface encourages different workers to type different part of the signal by adjusting audio saliency.

Non-expert captionists may also be drawn from micro-task marketplaces such as Amazon's Mechanical Turk, groups of volunteers, or other sets of willing participants. Our approach only requires workers to be able to hear and type. Because the pool is dynamic, workers come and go, and no specific worker can be relied upon to be available at a given time or to continue working on a job for a set amount of time. Workers cannot be relied upon to provide high-quality work of the type one might expect from a traditional employee, due to laziness, misunderstanding of the task, or delays that are beyond their control, such as network latency.

Speakers will make mistakes, speak unclearly or away from the mic, use unfamiliar terms, and otherwise make it difficult for workers to hear certain parts of the speech. In this case, 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 specialized terms, or terms the speaker defined during the presentation, whereas people may have prior knowledge of the topic, or can learn the new term on-the-fly. In most cases workers are only delayed by their typing rate, not their comprehension, allowing overall faster responses than most ASR (as we show later in this paper), without sacrificing accuracy. ASR captions, on the other hand, often get considerably behind the speech, given sufficient speaking rates.

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 generally cheaper to employ. 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 coverage throughout.

In order to successfully use groups of non-expert captionists to generate complete and accurate transcripts, we need to intelligently merge all of the partial and noisy inputs into a single stream. Workers have different typing speeds, captioning styles, and connection latencies, making time alone a poor signal for global 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. An alignment method must be able to robustly handle these inconsistencies, while being careful not to over-estimate the similarity of two inputs.

Using worker input exclusively fails to leverage existing knowledge of languages and common errors. We use additional information about the most likely intended input from a worker can gained from 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 conflicting worker input about ordering. To determine when words are *equivalent*, we use the Damerau-Levenshtein distance between the words, weighted using the manhattan distance between the different letters on a standard QWERTY keyboard.

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

Real-time captioning by non-experts is a type of human computation [25], which has been shown to be useful in many areas, including writing and editing [4], image description and interpretation [6, 26], and protein folding [9]. People with

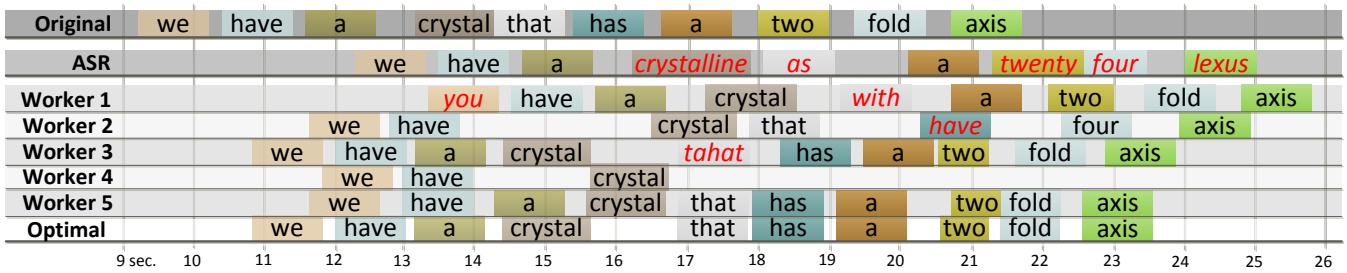

Figure 2: Captions by ASR and 5 crowd workers from Mechanical Turk for a portion of 60 seconds of speech. Results suggest that merging the input of multiple non-expert workers may result in accurate, nearly real-time captioning. The left edge of each box represents when that word was received; boxes with the same color were manually coded to represent the same word position; and italicized words in red are incorrect.

disabilities have long solved accessibility problems with the support of people in their community [5]. Increasing connectivity has made remote services possible that once required human supporters to be co-located. Existing abstractions obtain quality work by introducing redundancy and layering into tasks so that multiple workers contribute and verify results at each stage [19, 20]. For instance, the ESP Game uses answer agreement [26] and Soylent uses the multiple-step find-fix-verify pattern [4]. Because these approaches take time, they are not suitable for real-time support.

Human computation has been applied to offline transcription with great success [2], 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 [21]. 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 [6], 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 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 [26, 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 [18]. 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 could 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 who was then given direct control for small periods of time, 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 this case, the runtime is exponential in the number of workers), effective approximate solutions have been developed by the bioinformatics community. We have extended the MUSCLE package [11] in order to align English text in a meaningful way (see Fig. 3). Namely, we replaced the original mutation model for nucleotides with a spelling error model for English based on the physical layout of a keyboard. For example, when a person intends to type [a], he is more likely to mistype [q] rather than [m]. The model is further augmented with context-based features learned from spelling corrections in 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 in a principled fashion. Once the partial captions are aligned, we need to merge them into a single transcript, as shown in Fig. 3. We perform a majority vote for each column, then remove the gaps in a post-processing step.

SCRIBE

SCRIBE gives users on-demand access to real-time captioning from groups of non-experts via their laptop or mobile devices (Figure 1). A user starts SCRIBE by opening the application on their laptop or smartphone. SCRIBE immediately begins recruiting workers for the task from Mechanical Turk, or a pool of volunteer workers. We currently use quikTurkit to recruit workers [6]. Workers can be rewarded using either money or points. When users are ready to begin captioning audio, they press the start button, which then begins forwarding audio to Flash Media Server (FMS) and signals to the SCRIBE server to be captioning. We use FFMPEG running on the mobile device or laptop to stream audio to FMS usign the RTMP protocol for real-time audio streaming.

```
1: ---learn--g is such------- a- suitcase word though right- so -------- has a lot of there-----s a lot
2: -o-learning is such------------------------------------------------------------ there a are a lot
3: ---learning ss such------- a- suitcase word though---------- learning has ----------------- is a lot
4: ---lea-ning is su-h------- a---------------------- right- so learning------------------------ a lot
5: so learning is such------- a- suitcase ---- though---------- learning has---------------------- lot
6: ---learning is such------- a- suitcfse word though right ------------------------ this ----in a lot
f: so learning is such        a  suitcase word though right  so learning has a lot of there    is a lot
```

Figure 3: Example output of our multiple sequence alignment algorithm when run on input from worker ins transcribing "Learning is such a suitcase word though. Right, so there h [. . .]" Each line shows a partial transcript provided by a worker, and the final merged caption (f). Dashes represent "gaps" inserted by the system in order to attain an optimal alignment given our language model. While individual workers provide noisy and incomplete data, merging multiple transcripts significantly improves recall as well as precision.

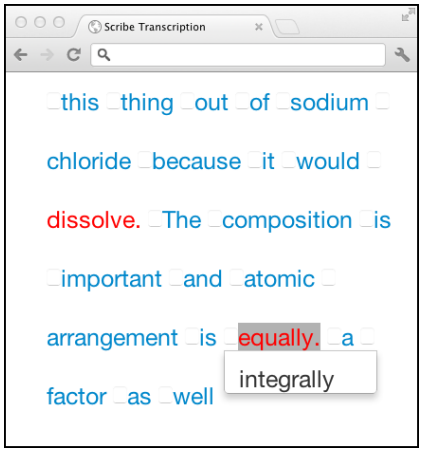

Figure 4: The web-based interface for users to see and correct the live caption stream.

Workers are presented with a text input interface designed to encourage real-time answers and designed to encourage global coverage (Figure 5). In our experiments, we paid workers $0.005 for every word the system thought was correct. This interface is discussed more in the next section.

As workers type, their input is forwarded to an input combiner running on the 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 SCRIBE system.

Once the inputs have been merged into a single transcript, we present users with the system's current best guess on a dynamically updating web page. As new information arrives from workers, this output can be modified and improved. The user interface is shown in Figure 4. Re-ordering and insertions in the transcript are animated in order to smooth the transition and make reading the text easier for users. Updates to spelling are momentarily highlighted to allow users to track changes in content they have already read.

Our approach 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. We allow users to either let the system choose a default balance between the two, or select their own balance of precision versus recall by using a slider bar in the user interface. Workers can select from a continuous range of values between 'Most Accurate' and 'Most Complete'. These values are then mapped on to settings within the combiner itself.

When users are done, pressing the application's stop/pause button will terminate the audio stream, but let workers complete their current transcription task. The workers are asked to continue captioning other audio for a time in case captioning needs to resume quickly.

In addition to the end-user, we also allow the option to forward the live output to a second group of workers recruited to correct the final stream using an editing interface. This step is optional, but can help to correct many of the easily identifiable small errors made by workers and the input combiner. Additionally, users themselves have the option of editing the final stream based on corrections they can identify, such as out-of-order words, or a term known to them that remote workers may have missed. Existing transcription services may also opt to provide professional services for lower prices to work with the crowd input

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. SCRIBE returns information such as the confidence of each spelling, possible alternative words and arrangements. 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. The interface as such 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 collisions within the graph.

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 jointly to allow difficult problems in one to be addressd in the other. For instance, determining where each word in a partial caption fits into the final output stream is difficult, and so the interface was designed to encourage captionists to type continuous sequences and signal breaks.

We detail the co-evolution of the worker interface and al-

gorithm for merging partial captions to form a final unified transcription. By developing an interface to elicit input from workers that is better suited for the merging algorithm, and using an algorithm that is able to take advantage of as much information as can be collected from workers using the interface, we can create a system which most effectively uses the imperfect transcription abilities of a set of workers to create more legible audio transcriptions.

AN INTERFACE FOR REAL-TIME CAPTIONING

A primary component of SCRIBE is the interface that non-expert captionists will use to provide their captions (Figure 5). The web-based interface streams audio to the captionist who are instructed to type as much of it as they can. Workers are furthermore told to separate contiguous sequences of words with [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 delay 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 their input. 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 6 shows how the volume can be systematically varied to maximize coverage over the whole signal.

In preliminary work, we instead chunked 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. Although this would be good in many situations, SCRIBE needs input quickly and a stream that does not stop is a reminder of the real-time nature of the transcription.

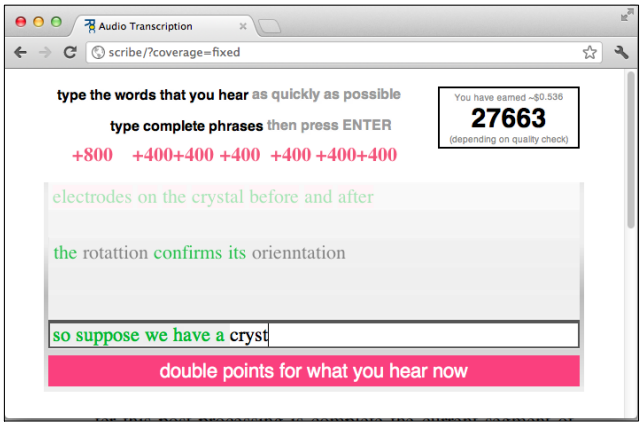

Figure 5: The captioning interface encourages workers to type audio quickly by locking in words soon after they are typed. In order to encourage captionists to cover specific portions of the audio, visual and audio cues are presented, and the volume of the audio is reduced during off periods. As the lines captionists type progress upwards, words that are determined to be spelled correctly turn green. Those typed during the period are highlighted in pink. More points (and money) are awarded for words that are spelled correctly and typed during the "bonus period." When words reach the top of the display, points rise above them and fly over toward the points display helping to connect words with their value.

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. We validate this observation later in our experiments section.

REAL-TIME INPUT COMBINER

A primary component of SCRIBE is the ability to combine partial captions into a single output stream. A naive approach would be to simply arrange all words in the order in which they appear, but this approach does not handle reordering, omissions, and high levels of inaccuracy. Instead of addressing all of these problems at once, we will build models that address each in turn. Importantly, each model is useful immediately with workers trained to avoid errors that it cannot handle. Each new feature will make SCRIBE more robust to less-experienced workers. For instance, we expect that workers with some training can accurately type phrases while rarely omitting words. Less reliable workers, e.g. from Mechanical Turk, will need more robustness.

Online Dynamic Sequence Alignment

In order to achieve the response-time and ability to scale that is required for real-time transcription, we create a modified version of MSA that aligns matching words using a graphical model. Worker input is modeled as a linked list of nodes, with nodes from different workers containing equiv-

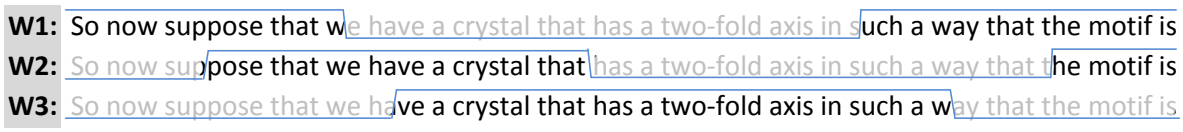

Figure 6: $\mathcal{S}$CRIBE encourages workers to type different portions of the input speech by systematically raising and lowering the volume of the audio, as depicted visually here. Artificially adjusting saliency while streaming the entire signal improves overall coverage while preserving worker context.

alent words being aligned based on submission time. As words are added, consistent paths arise from the input of multiple workers. We maintain the longest self-consistent path between any two nodes to avoid unnecessary branching. A simple example of the graph being built over multiple inputs is shown in Figure 7.

Reconstructing the Stream: Using a greedy search of the graph in which we always follow the strongest edge weight (a measure of how much agreement there is , we are able to derive a legible approximation of the transcription in real-time. The greedy search navigates 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 that are tailored to the specific domains of the audio clips being transcribed, either by generating it in real time along with our graph model, or by pre-processing language of from similar contexts. Specific n-gram data is more desirable than general data, as it facilitates more accurate transcriptions of technical language by non-technical crowds.

Using greedy search provides our system with a rudimentary initial transcription. The nature of greedy search causes this draft of the transcription to be partially incomplete in terms of word coverage, because the search favors paths through the graph with high worker confidence, but which may ignore branches of the graph that contain unique instances of words and thus omitting them entirely from the sequence. Therefore, our system post processes this initial sequence by attempting to add into it any word instances that are not already present and also have high worker confidence. The actual positioning of these omitted words in the transcript cannot necessarily be derived from the graph model, as they are not part of the greedy search's traversal, and therefore often disconnected from the instances of the words it should be nearest. Because of this, these words are added to the transcript draft by taking into account their timestamp in addition to bigram and trigram confidence that result from insertions into the transcription within a predefined time span of the omitted word's earliest detected input timestamp. After this post processing is complete the current segment of the transcript is forwarded back to the end user.

Run-time: Each time a worker submits a 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. This list can either be traversed, or 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 them would form a back-edge. Thus, 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 window is effective, though $\mathcal{S}$CRIBE 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 MSA, however, we show that this approach is effective in the real-time captioning domain, due to properties such as relatively low frequency of repetition and word repeats. In the future, we want to extend this model to better handle general sequence alignment dynamically, 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 test our different 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[2]. These lectures were chosen because a primary goal of $\mathcal{S}$CRIBE 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 so went over these

20 local participants took part in our study. Each participant captioned 23 minutes of aural speech in total over approximately 30 minutes. Participants first took a standard typing test. The average typing rate was 77.0 WPM (SD=15.8) with 2.05% average error (SD=2.31%). We then introduced participants to the interface we developed for real-time captioning, and had them caption a three minute clip using it. We next asked participants to caption the four 5-minute clips. We balanced the experiment so that participants used the interface modified to encourage increased global coverage on half of the clips.

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

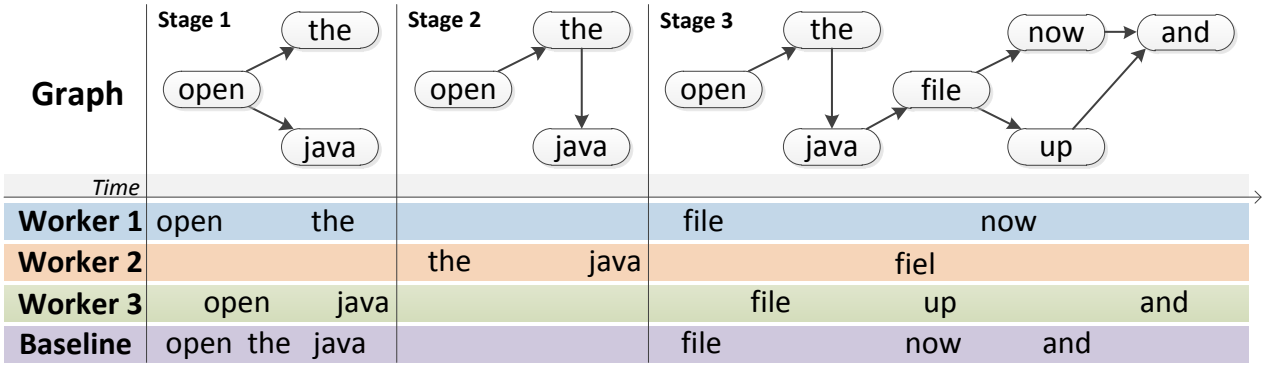

Figure 7: An example of graph building during captioning for the spoken sentence *"Open the Java code up now and..."* Input is being added to the graph over time by 3 workers. The top section shows the current state of the graph at the and of each stage. The bottom shows the corresponding input. Note that when Worker 2 spells 'file' incorrectly, the graph is not adversely affected because a majority of the workers still spell it correctly.

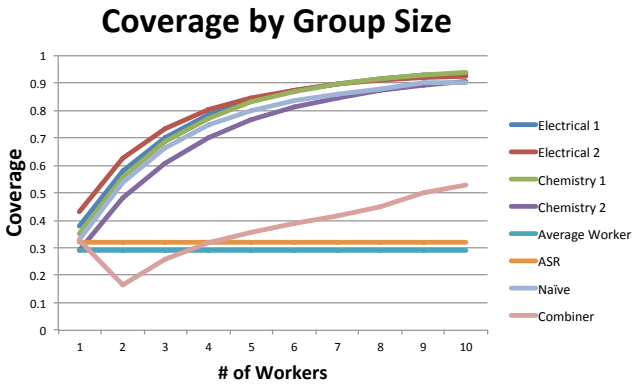

Figure 8: Coverage reaches nearly 80% when combining the input of four workers, and nearly 95% with all 10 workers. This demonstrates that captioning audio in real-time with non-experts is feasible.

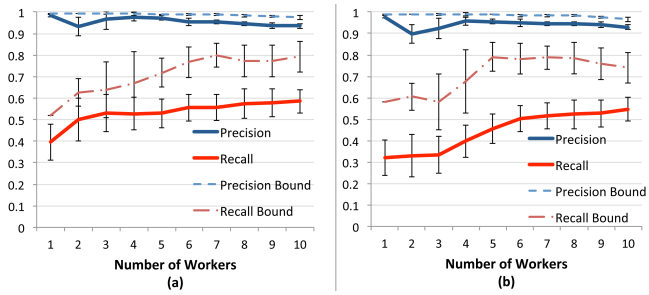

Figure 9: Precision and recall plots for our multiple sequence alignment (MSA) algorithm leveraging a given number of workers. Plots (a) and (b) show results for the results collected with saliency adjustment and without, respectively.

Coverage Experiments

A primary question for the effectiveness of our approach is whether 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. Figure 8 shows coverage numbers for various approaches.

Multiple Sequence Alignment

Figure Fig. 9 shows precision and recall plots for our multiple sequence alignment (MSA) algorithm leveraging a given number of workers. The metrics are calculated at a character level and averaged over all subsets of workers of a given size and over all speeches.

We see that precision is virtually independent of the number of workers, whereas recall significantly improves as we use additional partial captions. Note that explicitly motivating workers to deliver better quality transcripts (plot a) dramatically improves recall, as compared to no management of the workers (plot b). For example, only two workers achieve 0.5 precision with motivation, while six workers are required to produce the same precision without motivation. We additionally plot the theoretical upper bound on both precision and recall that can be attained by a fully informed MSA with access to ground truth. We see that SCRIBE is extracting nearly all information it can in terms of precision, but could improve in terms of recall. Narrowing this recall gap is one of the major directions of our future work.

Real-time Combiner

Our results show that SCRIBE is able to outperform the coverage of both ASR and any individual worker. Furthermore, while our simple approach scored better in terms of coverage, the amount and types of errors in the final transcript, it is far less useful to a user trying to comprehend the content of the audio. The average worker achieved 29.0% coverage and ASR achieved only 32.3% coverage, whereas the optimal with 10 workers is 93.5% coverage (Figure 8). Coverage of the combiner outperforms ASR and the average worker. Furthermore, the combiner produces its captions in within 3.2 seconds on average, including the delay of the workers.

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 different offsets from 0 to 9.

Participants 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 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 Amazon Mechanical Turk since we would not be able to provide directions in person. We used quikTurkit to recruit a crowd of workers who also captioned the four clips (20 minutes of speech). Our HITs paid $0.07 and workers could make an additional $0.002 per word paid as a bonus. We asked workers to first watch a 40-second video in which we describe the task, for which they received $0.02. In total, 18 workers participated, which cost $13.84 in total.

Overall, 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 fairly conservative in recruiting workers and cover much of the input signal. The workers who participated generally provided high-quality captions, although some had difficulty hearing the audio. We had some trouble recruiting and maintaining a large enough pool of workers, although we suspect that they may be more likely to participate as they learn of our HITs. The high cost of alternative approaches to real-time captioning means that we can pay relatively high wages and still be much cheaper.

DISCUSSION & FUTURE WORK

$\mathcal{S}$CRIBE encapulates a new approach for real-time captioning that allows amateur captionists to work together to produce captions in real-time. We have shown in this paper that groups of non-experts are able to collectively cover a speech signal, that we can encourage them to type specific portions of the speech that we want to encourage global coverage, and that it is possible to recombine partial captions and effectively tradeoff coverage and precision.

Covering the Input Signal: For our approach to work, the partial captions provided by non-experts must fully cover the input signal. Our experiments confirm that an individual worker is unable to cover even half of the input signal, but collectively they can cover the signal up to 94% (Figure 8). Many of the remaining missing words appear to be due to slight differences in phrasing, e.g. "anybody" vs. "anyone", that may not affect the usability of the captions. Our next step is to try to better understand the practical usefulness of our captions in the real-world through studies with deaf and hard of hearing students.

We have also shown that we can effectively encourage workers to cover portions of speech that we want. These numbers indicate that our approach is viable, although we expect future work may seek to further improve the interface and worker feedback to increase coverage even more. 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.

We expect that leveraging a probabilistic language model will significantly improve $\mathcal{S}$CRIBE's performance, especially in terms of recall. Currently, $\mathcal{S}$CRIBE is agnostic to part-of-speech tags, sentence structure, and other linguistic regularities. A unified conditional random field model [22] that finds the most likely sequence of words in the final transcript 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 the original speech.

We may also add multiple classes of continuous workers to make some alternate approaches more viable. By using one class of workers to identify simpler information such as how many words have been said (by pressing one button each time a word is said), we may be able to reduce the amount of uncertainty that must be dealt with in the final model.

Our saliency adjustment is currently rigidly defined to be the same for all workers. Personalizing the on and off periods for each worker could improve results while requiring fewer workers. This lets more experienced workers and skilled typists be used to their full potential, while not overwhelming newer workers. Reporting scores, most common mistakes, and other information back to workers may also help them improve their typing ability, resulting in a more skilled set of available worker in future tasks.

For approaches that cannot be sped up enough to work effectively in real-time, we plan to extend $\mathcal{S}$CRIBE to enable further revisions of the 'final' transcript. This means that if users are willing to wait a longer period of time, then a more accurate transcript will be returned. By re-processing the input offline, and selectively crowdsourcing sections with low confidence, the cost of obtaining these corrections can be kept lower than if all content were processed separately. Even when user's are not able to wait, this provides a higher quality log of the conversation.

While obtaining exact transcripts of speech is very useful in a range of scenarios, spoken language often flows differently than written text. Speakers pause, change subjects, and repeat phrases – all of these make exact transcripts difficult to read without inflection and timing information. Captioning methods such as C-Print paraphrase speech to keep up, cleaning up sentences to be easier to read, but also skip content. Using our system, a language model can be introduced to help correct sentences that have inconsistencies in order to make the final text more readable. These models can also be individually customized to the user in order to produce personalized transcripts.

CONCLUSIONS

In this paper, we introduced an end-to-end system for real-time captioning by groups of non-experts. We demonstrated that groups of non-experts can outperform both individuals and ASR in real-time transcription in coverage and precision. We introduced a new algorithm for aligning and merging par-

tial text captions to create a complete caption stream on-the-fly. Our results suggest the feasibility of our approach and a number of opportunities for future research.

REFERENCES

1. R. Barzilay and K. McKeown. Sentence fusion for multidocument news summarization. *Computational Linguistics*, 31(3):297–328, 2005.
2. Y. C. Beatrice Liem, Haoqi Zhang. An iterative dual pathway structure for speech-to-text transcription. In *Proc. 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 *Proc. of the ACM Symp. on User Interface Software and Technology*, UIST '11, 2011.
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 *Proc. of the 23nd ACM Symp. on User Interface Software and Technology*, UIST '10, pages 313–322, 2010.
5. J. P. Bigham, R. E. Ladner, and Y. Borodin. The Design of the Human-Backed Access Technology In *Proc. of the Conf. on Computers and Accessibility (ASSETS 2011)*, pages 3–10, 2011.
6. 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 23nd ACM Symp. on User Interface Software and Technology*, UIST '10, pages 333–342, 2010.
7. L. Chilton. Seaweed: A web application for designing economic games. Master's thesis, MIT, 2009.
8. 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.
9. 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.
10. 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 Intl. Conf. on*, pages 5121 –5124, 31 2008-april 4 2008.
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. *Sunderland, Massachusetts: Sinauer Associates*, 2004.
14. J. L. Flowerdew. Salience in the performance of one speech act:the case of definitions. *Discource Processes*, 15(2):165–181, Apr-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 *ICSLP-2002*, pages 1757–1760, 2002.
17. H. Kadri, M. Davy, A. Rabaoui, Z. Lachiri, N. Ellouze, et al. Robust audio speaker segmentation using one class SVMs. 2008.
18. W. Lasecki, K. Murray, S. White, R. C. Miller, and J. P. Bigham. Real-time crowd control of existing interfaces. In *Proc. of the 24th annual ACM Symp. on User Interface Software and Technology*, UIST 2011.
19. G. Little, L. B. Chilton, M. Goldman, and R. C. Miller. Turkit: human computation algorithms on mechanical turk. In *Proc. of the 23nd annual ACM Symp. on User interface software and technology*, UIST '10, pages 57–66, New York, NY, USA, 2010. ACM.
20. Kittur, A. and Smus, B. and Khamkar, S. and Kraut, R. E. Crowdforge: Crowdsourcing complex work. In *Proc. of the 24th annual ACM Symp. on User interface software and technology*, UIST '11.
21. 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, pages 159–176, Berlin, Heidelberg, 2006. Springer-Verlag.
22. C. Sutton and A. McCallum. *An introduction to conditional random fields for relational learning*. Introduction to statistical relational learning. MIT Press, 2006.
23. 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.
24. 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.
25. L. von Ahn. Human Computation. Ph.D. Thesis. 2005.
26. 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, pages 319–326, 2004.
27. 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.
28. 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, St. Thomas, US Virgin Islands, 2003*, 2003.