Accessible Viewing Devices - Deaf and Low Vision

A new way of providing captioning services in the classroom

Prof. Raja Kushalnagar
JD., LL.M., PhD.
Research Advisor
Rochester Institute of Technology
1 Lomb Memorial Drive
Rochester, New York
rskics@rit.edu

John Rivera
Undergraduate Research Assistant
Rochester Institute of Technology
1 Lomb Memorial Drive
Rochester, New York
john.rivera@mail.rit.edu

Warrance Yu
Undergraduate Research Assistant
Rochester Institute of Technology
1 Lomb Memorial Drive
Rochester, New York
wxy1697@mail.rit.edu

Daniel Steed
Undergraduate Research Assistant
Rochester Institute of Technology
1 Lomb Memorial Drive
Rochester, New York
dss1638@rit.edu

Vasu Gupta
Undergraduate Research Assistant
Rochester Institute of Technology
1 Lomb Memorial Drive
Rochester, New York
vxg1421@rit.edu

ABSTRACT

Today, C-Print and CART (hereforth simply called ?captions? or ?captioning?) services are delivered using laptops or tablets supplied by the captionist who bring them, in addition to the captionist? own laptop, from class to class. Additionally, as the laptops and tablets are not the students? own, they do not have any customization by the student – such as viewing preferences, installed software and forth. This hampers the student?s ability to multitask while watching the captions – the student would need to work on his/her own laptop or tablet in addition to watching the captions on the supplied laptop or tablet. This is also an issue for low-vision students, who often have software on his/her own devices that improve visibility.

Also, deaf students in the classroom using captioning do not watch the professor speaking; they spend a majority of their time reading the captions on the screen. However, this causes difficulty for the students when the professor uses slides or the whiteboard during the lecture. Deaf students need a way to easily switch between reading the caption text and the visual aspects of the lecture seamlessly with minimal breaks to their learning stream. This is especially problematic for deaf students with low vision, as they often cannot see the slides or the whiteboard in the first place, which makes current implementations of C-Print and CART a compromised option.

Our project, internally called Project NA, aims to be a solution to those problems. We re-thought the C-Print/CART paradigm and created a novel system using off-the-shelves components with affordability in mind which solves those problems;

Through the cloud platform where the heavy-lifting all takes place ?in the cloud? on a remote server. The captionist and students merely login on a website, and are automatically connected to their respective user interfaces. They can use their own devices. The captionist, instead of bringing laptops or tablets for the students in addition to his/her own, only needs to bring the latter. Students will be able to multitask easily, having the captions display in a window while he/she work on another. Low-vision users will also be able to use his/her own adaptive applications and settings.

Through a UI that simultaneously displays a video feed of the visual aspects of the lecture along with the caption text. The deaf student will be able to intelligently switch attention between the captions and the video feed. This maximizes information retrieval during the lecture with minimal disruption to the processing of the information. For low-vision students, the UI is also a boon, allowing the viewer to view slides and whiteboard close-up via their own computer screens with their own adaptive technologies

And through tactile and visual feedback that informs the viewer of important changes during the lecture – change of slides or the lecturer writes on the whiteboard for instance – which enables the deaf student to only view the video feed when pertinent, instead of needing to check the video from time to time to determine whether something has occurred. The notification comes via either a flash of the screen along with a banner notifying the user of what has changed, or vibration of a small Bluetooth-connected vibration device.

The user interface of course contains standard features such as font adjustments and an option for a white-on-black color scheme. The user can also choose to view the video and captions simultaneously, view the video in full-screen, or view the captions in full-screen (with the video relegated to a small, PIP-like view on the lower right of the window).

We believe we are developing the next evolution of live captioning services for the deaf and can?t wait to bring it to C-Print and CART users.

1. INTRODUCTION

The term "low vision" covers a broad range of vision problems, but one common theme is nearsightedness that cannot be resolved by eyeglasses or contacts. One of the authors has low vision since birth. Throughout his life, he had to deal with issues with full accessibility in the classroom.

In physical classrooms, he would need to sit approximately four feet away from the sign interpreter interpreting for the class. This does not provide him with full accessibility because he couldn?t see a) the professor, b) anything the professor might write on the whiteboard, and c) any slides if the professor chooses to use them.

Currently, he depends on the class readings and outside sources, along with reading the notes taken by a notetaker, to obtain the information he misses in-class. Slides are often also provided so he could read them prior to or after class.

The aim of this report is to explore alternatives that could improve access for those who are deaf and have low vision in the classroom. The report first explores technologies that can improve accessibility in the classroom, then explores technologies that enable improved accessibility to online courses.

2. ASYNCHRONIZED YOUTUBE CAPTIONS

2.1 The Problem

The AVD currently relies on eye-tracking. While this is cool, and could conceivably be useful, for the low vision student, it is difficult to calibrate using the included software (one of the authors couldn?t tell where to look during the second part of the calibration process). And the author could see himself being annoyed when it inevitably mistakes a move of the eyes as an intention to change views (he suffers from mild nystagmus, and could see this being a bigger issue for those with more severe nystagmus).

So the author proposes that the views should be switchable without any eye-tracking – with keyboard shortcuts, for instance. The keys should be easily found "by feel" (the spacebar, numpad and arrow keys are good choices, the top number row is not).

Also, the views within the AVD are small. Other than dragging a huge monitor into class, this is generally not ideal. The interface should be changed so that the "view" being viewed should take up the whole, or a majority, of the screen. Taking the whole screen is simple – the basic logic would be the same, but the active view takes the whole screen, and the keyboard shortcuts would take the user to other views.

For the interface where the video only takes up a majority of the screen, the other views could take up the bottom, or the side, of the screen as thumbnails, which has the benefit of letting the user know at a quick glance what views there are, and how many. The eye-tracking device could be available here if desired (a glance to the bottom or side would change the view, for example). This also may offset the issues for low-vision users – simple, large areas would mitigate any issues that would arise with nystagmus or the such.

This pertains to online courses only. Size and captions are generally fine – full-screen is available, and the captions are readable but can be improved. Should also be standardized, so one could view the captions on mobile operating systems.

The captions issue is easily solved by switching to a darker background (black, for instance) with a light-colored (i.e. white) text.

However, one of the authors would often need to focus on the captions, missing what is happening in the video. A majority of the time this is not an issue, since the videos are mostly talking heads, but occasionally the talking head would produce a hand which would write on the board. He would often miss this.

To approach this issue, there are several solutions that comes to mind. One would be asynchronized captions (pausing the captions as the video continue, and when the talking head resumes, one could resume the captions, perhaps at a somewhat quicker pace until it catches up, or even pauses the video until the captions catches up).

Another solution is video analysis – if it could be detected that the hand and the whiteboard took over, the user could be notified, and so focus could be shifted to the video. This can be used in conjugation with the asynchronized captions described above.

Additionally, a more persistent view of the captions would be useful in case a line is missed. Currently, the solution to this is a box below the video which contains the entire video transcript up to the point in time. It displays approximately four lines of captions, but can be scrolled up to view the rest of the captions.

That led to another problem; when switching between the video and the captions, one often loses the context of the latest line. One would often need to scroll back up to ascertain the context of the current line. A possible solution is to highlight the latest sentence (as opposed to line).

To illustrate this, the latest line of captions may look like this;

and sugar for taste.

This would seem, from simply reading this one line, nonsensical. However, with the line of caption prior to this, it reads;

Then we add some cinnamon and water for taste.

Now there is context. The aim is to highlight the two lines of captions as a bookmark, instead of only marking the last line.

2.2 Technology/Methodology

An extension was developed for Chrome to improve accessibility to captioned videos for low vision users. It displays a YouTube video along with its captions if available, on a separate page from the main YouTube site. This enables us to develop a custom user interface tailored to those with low vision (fig. 1).

The interface consists of the YouTube video, with two rows of buttons directly beneath. The first row of buttons controls the video – the user can skip five seconds backwards or forwards as well as slow down or speed up the playback of the video. The second row of buttons control the transcription which appears directly below the buttons, allowing the user to start and pause the transcript and invert the text and background colors as well as adjust the font size between 12 pixels and 36 pixels. The transcript area is a scrollable area allowing the user to scroll down to view previously displayed captions. The view defaults at black text at 24 pixels in size and a white background. A key feature of this interface is the non-simultaneous viewing of the video and the subtitles. The user can view the video, pause it and start the subtitles, which will play up to the point at where the video is paused. The user can then continue the video, and repeat the process.

The extension was built using HTML5, CSS, AJAX and JavaScript/jQuery. The extension itself is little more than a button on the toolbar of the Chrome browser. When clicked, it will open a new tab displaying the custom interface hosted on a remote server. The reason for this will be discussed below. When the button is clicked and before the new tab is created, the extension reads the URL of the page the user is currently on. If it is a YouTube page, it will obtain the string directly after "?v=" which is the unique 11-character video ID string for the video the user is viewing. This string is then sent as a PHP parameter to the newly-created tab.

JavaScript and jQuery is then used to obtain the video ID and retrieve the video?s subtitles, if available, and invoke the YouTube API to display the video. This is where our attempt to create a self-contained Chrome extension failed; doing so prevented us from calling the YouTube API functions, which prevented us from creating the interface that we set out to create. The YouTube API functions was required to obtain the timestamp of the video – this is needed to allow for the non-simultaneous viewing feature. We were also unable to control the video via the skip and speed adjustments buttons. For this reason, we decided to move the interface portion to a remote server, which freed us of the security limitations within a Chrome extension.

Having moved the interface to a remote server, the calls to the YouTube API were successful, and we were able to create the interface we set out to create.

The interface was designed so the user would naturally discover the features. The contents were centered, and the controls are naturally placed beneath the video. The user would then see the buttons for controlling the subtitles, and the subtitle area directly below.

2.3 Evaluation

One of the authors has low vision, and played a major part in designing and developing the interface and the backend. The author believes this would help increase accessibility to low vision users such as himself. When viewing a video with captions in its usual place, the author would often miss what is happening in the video as he focuses on the captions, and vice-versa. The non-simultaneous viewing is a key feature which allows the author to give full attention to the video or the subtitles without missing any parts of the other. The color inversion is an essential feature, as viewing white text on a black ground is much easier on the eyes with less brightness and glare.

The font size adjustments is an obvious feature which allows the user to adjust the interface to their own vision needs – some would prefer text to be smaller, and others would prefer text to be bigger.

The video controls are useful features – if one misses something on video, one could skip backwards five seconds. The video could also be slowed down so the user has more time – for example, viewing a slide. Finally, the video could be sped up in talking-head portions where there is not much to see, saving time.

There is, however, a major problem: through using this implementation, it was realized that the captions, not the video, is the focus. A majority of our attention was on the captions, not the video, and thus pausing the captions and having it "catch up" is not nearly as useful as we hoped. Which led us to the breakthrough for the live course setting, discussed below.

3. ACCESSIBLE VIEWING: LIVE CAPTIONING

Deaf and Hard of Hearing students rely more heavily on visual learning than hearing students. However, these students, including the authors of this paper, often encounter visual noise, such as large viewing distances, line of sight interference or obstruction, poor lighting or viewing angles. This classroom visual noise can significantly interfere with the visual perception and learning process for deaf students. In addition to visual noise, mainstreamed deaf students have to manage visual attention between two or more simultaneous visual sources. The extra visual sources include the visual representation of the classroom audio, which is typically either a sign language interpreter or real-time text typed in by a captionist, as shown in Fig. 2.

One of the most basic and most revolutionary features of the project is that it is browser-based, using HTML5, CSS & JavaScript to create the UI. This allows the students to use their own computers to view the C-Print or CART text as well as watch the live video stream of the classroom (more details below). This is a boon to disability services departments because there is less equipment that has to be bought and distributed to students, to the captionist because there would be less to carry (she/he would only need her/his own

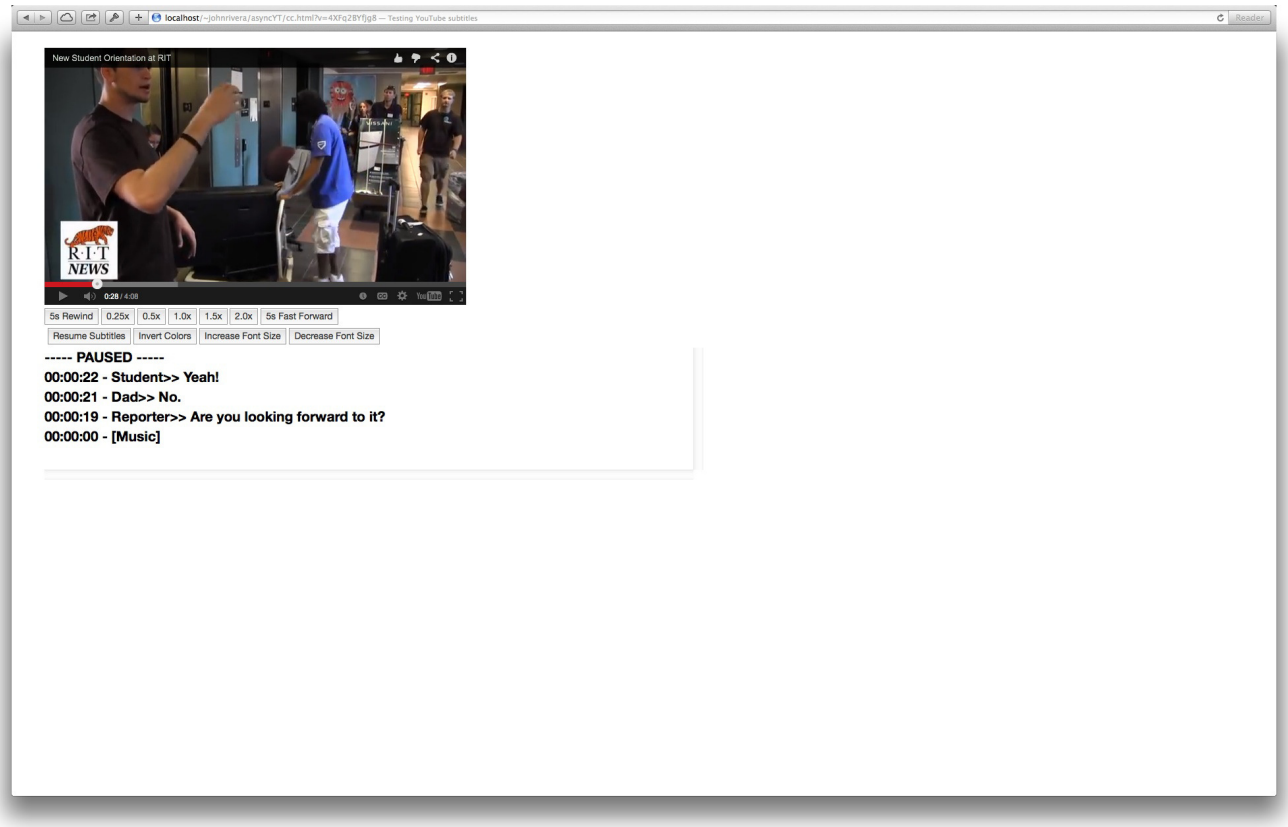

Figure 1: The user interface of the Asynchronized YouTube Captions web application.

Figure 2: The simultaneous visual attention required of deaf students.

laptop and a webcam, nothing more), and the students because they are able to use whatever they are comfortable with. This also allows the students to multitask on the same device – watch the C-Print or CART stream and work on a programming example at the same time, for example. For low-vision students, this also holds true – those students often have customized software on their own computers for best viewing, and they can use this same environment for their accessibility needs in the classroom.

3.1 Technology/Methodology

The project consists of two user interfaces; the students' and the captionist's, both browser-based.

3.2 The Student's View

The C-Print/CART portion of the UI displays what is typed by the captionist. This view can be viewed in half-screen and full-screen orientations. The text is displayed live as the captionist types. The text that is displayed can also be manipulated – the font size can be increased or decreased to the user?s preferences, and the colors can be inverted.

This is the basic requisites for a C-Print or CART system. Deaf users focus a majority of their attention on the text that is being typed, so the view is necessarily a major part of the UI. Font size adjustments and color inversion serves as obvious benefits for those with low vision.

The text is relayed via a server – the server can be remote, or it can be local on the captionist? computer – which receives the text that the captionist is typing, and relays it to the students? view. This is implemented using WebSockets, which offers a low-latency, message-based sockets for the browser. The server is written in Python using the gevent library, and the handling of the messages at both ends are handled by JavaScript. The text is actually sent to the students? views every second to prevent messaging overload. This can easily be adjusted, and the latency can be reduced in the future. The font resizing and color inversion is implemented via basic CSS and JavaScript.

The second major component of the student view is the video feed. This feed is a live feed from a webcam which records the classroom. Ideally, the recording should be focused on the slideshow presentation. Changes in slides can be detected via the webcam, and the viewer will be notified via a brief flash of the screen, along with a banner on the upper left notifying the viewer that the slides has been changed. The video view can also be adjusted – it can be half-screen, full-screen, and docked on the lower right, similar to the Picture-In-Picture feature on many television sets.

For deaf students, the main focus is on what is being told – thus, the C-Print/CART view. However, important information is also conveyed visually. The deaf student would need to look up at the projection screen occasionally to see if slides has been changed, or at the whiteboard to see if the lecturer has written anything. This interrupts their thought process – their "stream" so to speak – when "listening" to the lecturer speak. The video feed enables the student to check with minimal effort or distraction. It can even be monitored peripherally. Moreover, the notification of slideshow changes eliminates this process entirely – the student will know when to view visual information other than the C-Print/CART stream. For low-vision students, there is an additional boon – a video stream of the slideshow enables the student to see the slides when sitting anywhere in the classroom. The full-screen view is particularly designed for this aim.

The webcam stream is yet to be implemented – several possibilities are being explored, including WebRTC Multicasting, flash-based solutions and cloud-based solutions. Cloud-based solutions are the easiest to implement, but there are privacy concerns.

The notification aspect, however, has been implemented. The notifications themselves are sent to the client via WebSockets.

Finally, the students are able to communicate with the captionist via a text box at the bottom. This allows the students have the captionist speak for them in the class. Unlike the text typed by the captionist, the students' messages are not live – the student needs to either hit the Return key, or click the Send button.

This is so the student can read their message before it is sent (and spoken). This is also so the captionist can more efficiently be notified that a student wants to say something (more below).

This, like the C-Print or CART text and notifications (above), is implemented via WebSockets.

3.3 The Captionist's View

The captionist's view is much simpler than the students?. It consists of a text box for the captionist to type in – this text will be streamed live to the students. There is also a box below this which displays the messages sent to the captionist from the students (as discussed above). A key feature of this view is that it, similar to the students? view, has a notification feature which briefly flashes the screen and displays a banner on the upper left, when a student sends a message.

From the experiences gathered from those who use C-Print or CART services in the classroom, often, when a student wants to say something in the class, and thus types in the C-Print/CART viewer, the captionist often misses this as she/he is focused on typing what is being said. The notification ensures that the captionist is notified, and thus speak for the students more efficiently.

This, like the above, is implemented using JavaScript and WebSockets – as a message is received from a student, the notification procedure is triggered.

3.4 Evaluation

The solution we came up via the research and development, we believe, is vastly superior to the solutions currently being offered students. Having the captioning service reside on a server, or "in the cloud", allows more efficient use of resources and time – the captionist only need to bring his/her own device, as the students can use their own laptops or other web-connected device. Being device-independent is also sure

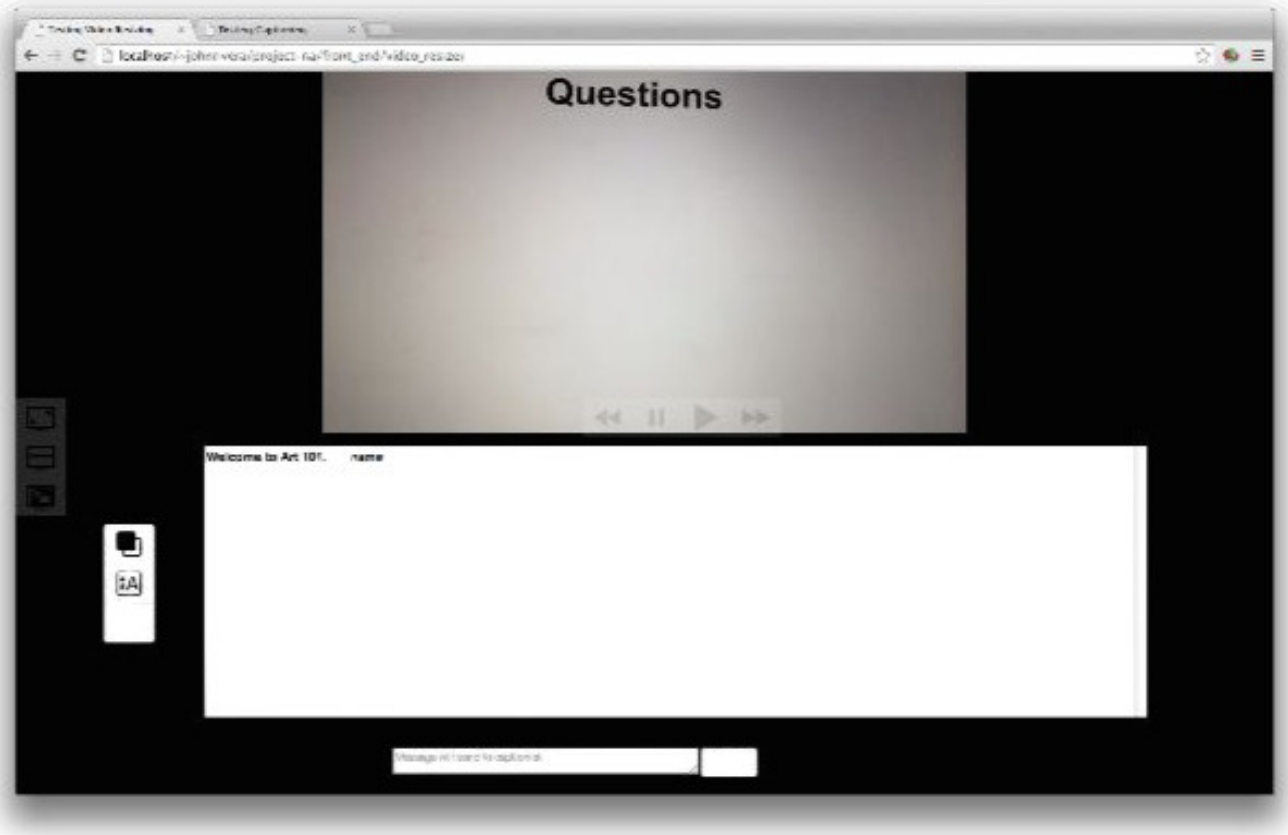

Figure 3: The student's view.

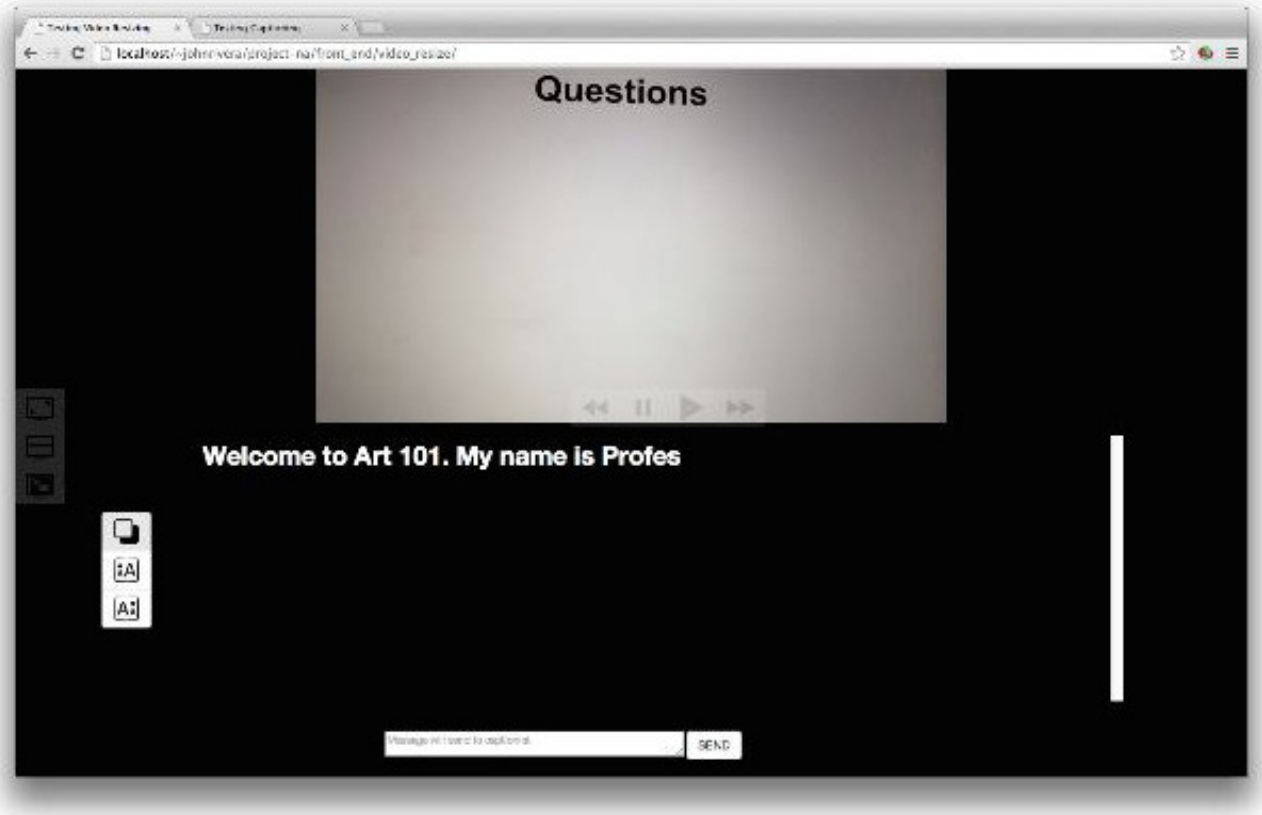

Figure 4: Example of font adjustments for better visibility in the student's view.

Figure 5: Example of picture-in-picture view in the student's view.

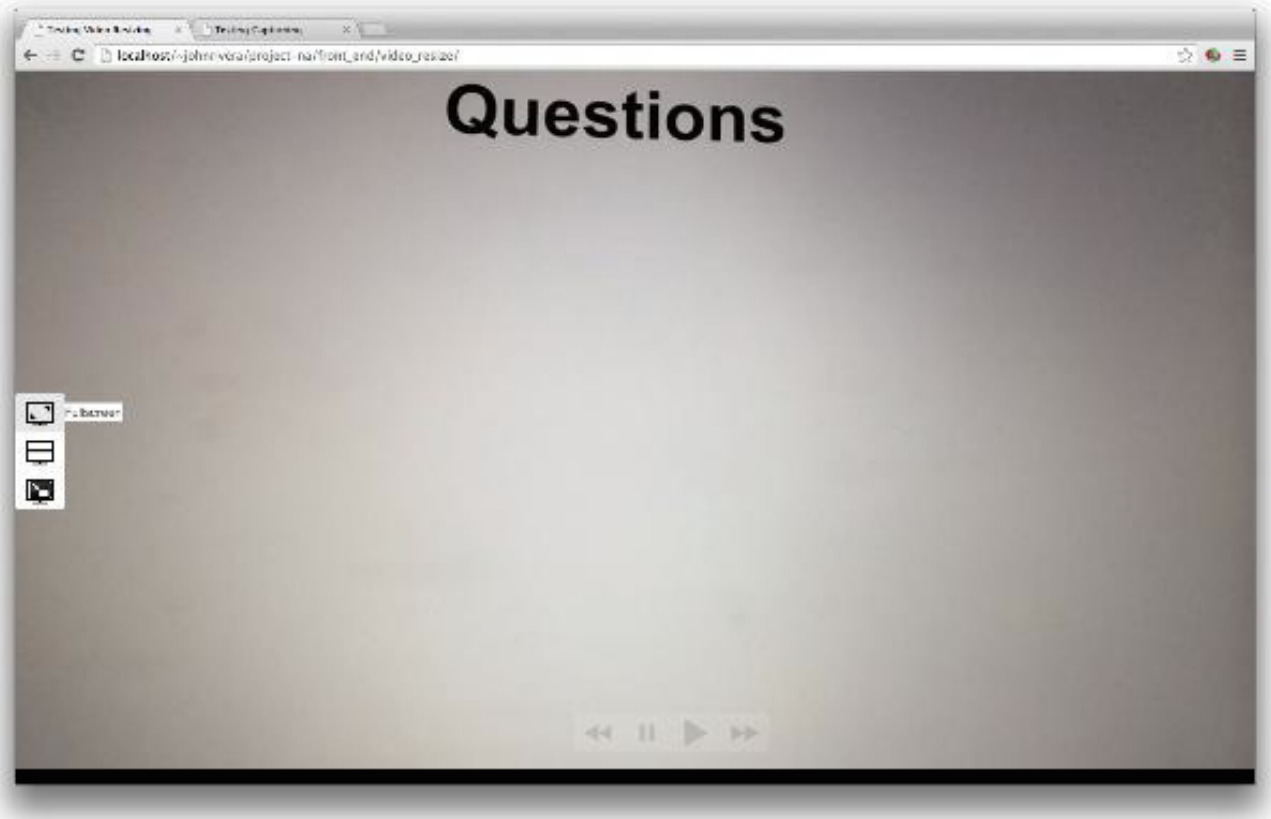

Figure 6: Example of full-screen video feed view in the student's view.

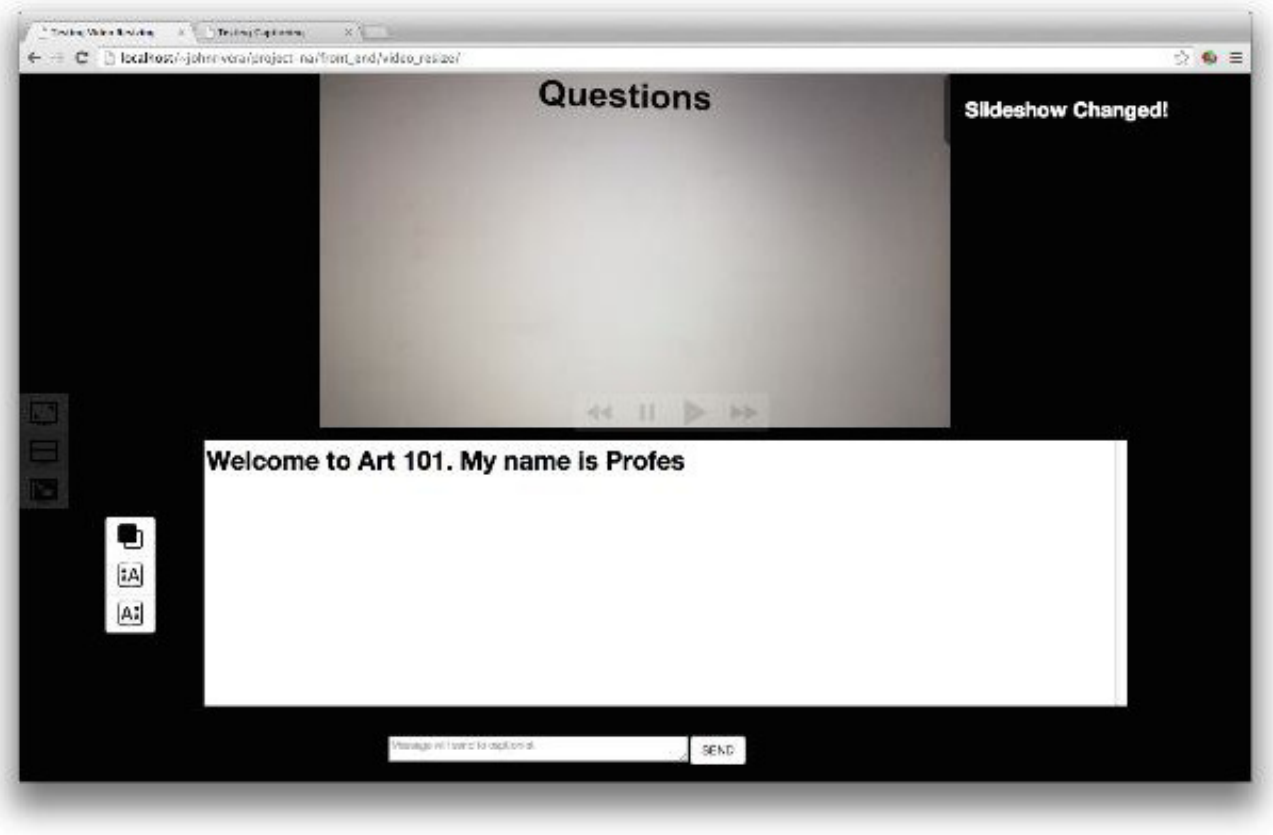

Figure 7: Example of notifications in the student's view.

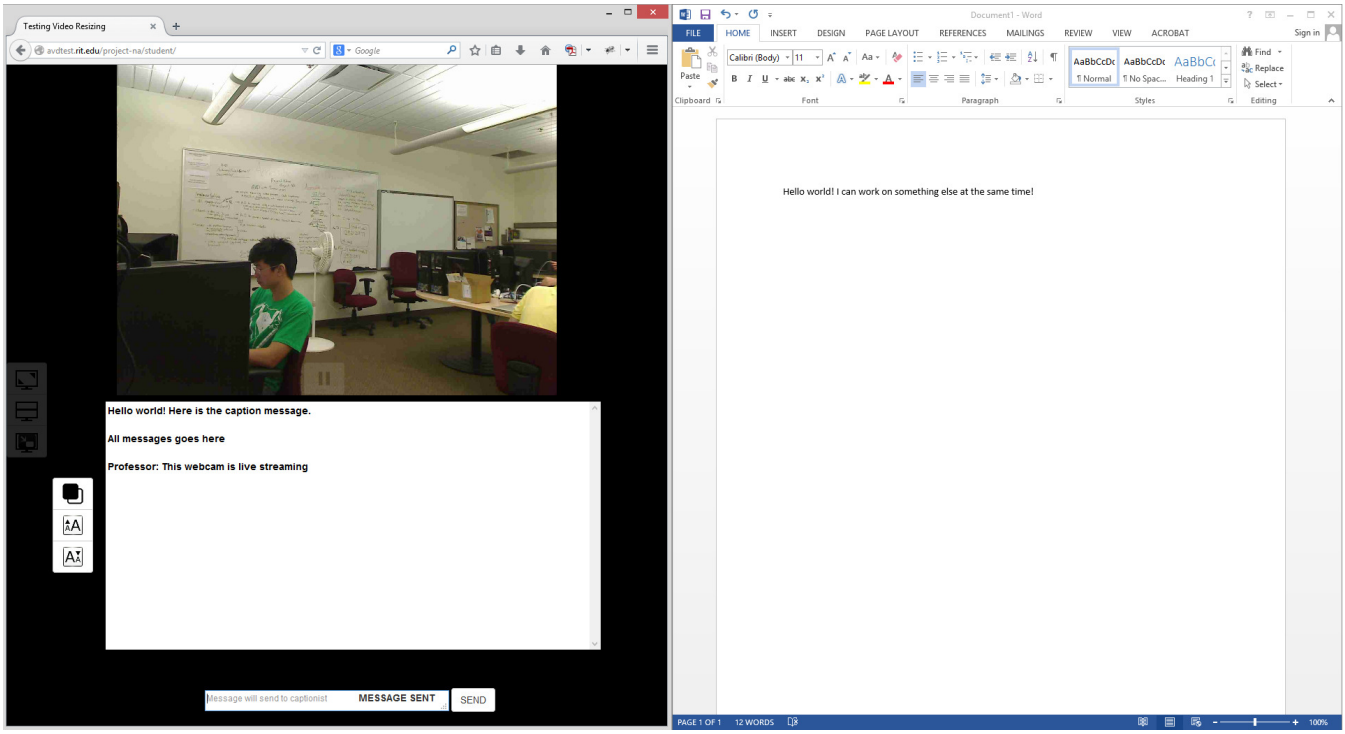

Figure 8: Example of multitasking possibilities with the student's view.

Figure 9: The captionist's view.

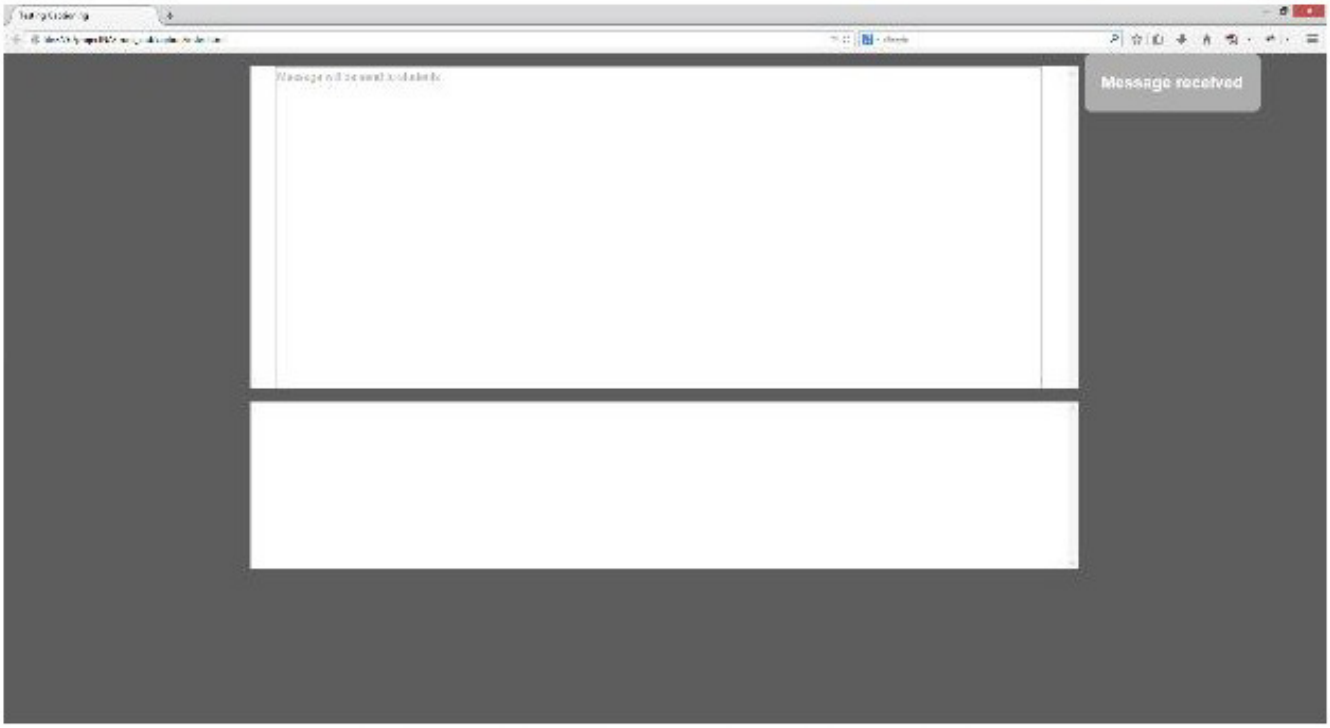

Figure 10: Example of notifications in the captionist's view.

to be a boon, as students can use any OS or device they prefer.

Another important benefit to having the user interface reside in the browser is the ability to multitask. Students can have the captions and a Microsoft Office document, for example, side-by-side on their screens. Low vision users can also use their own adaptive software, instead adjusting to a computer not their own.

Finally, the video feed is very useful for attention management. The user only need to look at the feed to ascertain if anything has changed, instead of looking away from the screen. It is also particularly useful to those with low vision, as they can now view powerpoint presentations or the whiteboard right on their screens, which can potentially allow them to see visual aids they could not have been able to use previously.

4. CONCLUSIONS

Having learnt from our implementation of the YouTube captions viewer and our experiences in the classroom and using C-Print/CART services, we believe we have developed the next step in the evolution of live captioning services. We have developed a suite of software we would want to use ourselves in educational settings.