A Chrome Extension for Navigating Active Web Pages Using Human Gaze and Speech

Renee Arianne F. Lat and John Patrick VJ. Albacea

I. Introduction

Developments in the fields of information technology and computer science are constantly evolving in our society. We are currently experiencing vast amounts of computer systems being applied in our environment [1]. Computers are now an essential tool for communication, education, entertainment, and the like. In line with this, new types of human-computer interaction are needed. This involves not using special complicated equipment for getting different types of input [1] and aiding people who are physically disabled in using computers as well.

In our daily lives, we humans use vision and hearing as our medium of getting information about our surroundings. Most information gathering and human interaction can also be done in the internet due to the rise of social networking sites. Therefore, applying web page navigation through human gaze and auditory input on computer systems would provide an easier way to gather information and perform tasks on computers not only for most users but also for users who are physically disabled [1].

A. Significance of the Study

People who do not have any physical disabilities or do not have any difficulties in typing can easily use the keyboard and mouse combination. However, in using computers, the aforementioned modality does not apply to people who suffer from an injury or are physically disabled since birth [1]. Most people nowadays use the internet to gather information and to interact with people. Internet users has risen up from 3.7 billion users in late 2017 to 4.1 billion users just this December 2018 [4]. People use the internet not only for leisure but also for different fields like engineering, agriculture, medicine, marketing, entertainment and the like.

Eye-tracking studies have been applied to psychology, neuroscience, human computer interaction (HCI), medicine, and the like but applications using this kind of study for people's daily lives have only started during the 2000s [7]. Even though these studies provide ease of access for information to users, gaze tracking combined with voice recognition will present users with a more natural experience [2]. Since human interacts with their eyes and speech, it would be better if that is also how we interact with our devices [2].

In line with this, integration of both voice and gaze navigation into a Google Chrome extension for web browsers that will navigate and interact on active web pages will be a huge step in the field of human-computer interaction (HCI). This particular extension will allow users to use both modalities (gaze with voice) to navigate and control web pages according to the users will. Using this extension will be easy since it can be installed and integrated in the browser automatically [6]. This study will be implemented through Google Chrome extension for less possible conflicts on the installation.

Presented to the Faculty of the Institute of Computer Science, University of the Philippines Los Baños in partial fulfillment of the requirements for the Degree of Bachelor of Science in Computer Science

B. Statement of the Problem

One major step to the improvement of this study is making it multimodal – navigation by using both human gaze and voice. Digital age has directed us where information gathered by mental processes can also be used for interaction [7]. People continuously strive to think of better ways to do tasks correctly and efficiently. Using gaze and voice navigation altogether, it suggests a natural way for people to communicate and interact with our computers, particularly with web browsers which provides us the information we need for our daily living and curiosities.

C. Objectives of the Study

The general objective of the study is to develop a Google Chrome extension that enables navigation on active web pages through gaze and voice navigation using an ordinary web camera and an simple external microphone. Specifically, the study aims:

1) to incorporate basic commands like scrolling using human gaze;
2) to incorporate other basic commands (zooming in/out, back/forward page, etc.) using voice recognition;
3) to incorporate advanced commands (navigation, selection, filling up of forms, etc.) using human gaze with the help of voice recognition; and
4) to test the extension on a variety of websites.

D. Scope and Limitation

The study will concentrate on executing commands that will navigate through active web pages. The extension will not be limited to navigate only one web page but it should adapt to most active web pages on the browser. With regards to gaze navigation, the extension is only limited to recognizing gazes without any kind of eye glasses. Also, it cannot read the gazes of people with any kind of visual impairments. As for voice recognition, the extension would only be limited only to the English language. Proper diction and pronunciation must also be observed. As much as possible, a microphone would be needed for clearer voice input.Built-in laptop microphones are more open to unnecessary noise thus are more prone to error.

II. Review of Related Literature

A. Gaze Navigation

Gaze navigation is a famous technology that is more applied in games. It allows games to get sufficient information from users by tracking human gazes and using it as input [9]. With this gathered information, games can select a storyline that can suit users most and predict behavior and attention of users in games [7]. In the field of user experience, gaze navigation is also used to create more dynamic elements in a web page for the audience [10]. Software that are developed with gaze navigation technology also provides a more efficient way of encoding information into computers than the usual keyboard and mouse combination.

The major innovation in eye tracking was the creation of head-mounted eye trackers. Yarbus, a Russian psychologist, also studied eye movements by recording how people observed natural objects and scenes. As the rise of computers came, the interest for gaze tracking also persisted, making way for more interfaces and ways on how to read and track gazes with the use of computers. Most of the current and prospective aspect of eye and gaze tracking was applied in the field of games and entertainment. Studies show that action-themed games can be used as a tool in rehabilitating visually disabled or visually impaired people and improving visual attention. Also, it was confirmed that video game players enjoy more when they play with gaze tracking technology [15].

A web browser called *Weyeb* was developed to allow users use human gaze for page navigation and link selection [16]. Even though this study is helpful for people, creating a new web browser just for the sake of this function is not appealing to the market. It would be better if this kind of technology can be implemented on commonly used applications on our laptops rather than having it implemented on a new application that only has one kind of special functionality. Juruena and Abriol-Santos [7] created a Google Chrome Extension that implemented gaze navigation technology with the help of Webgazer. Webgazer is entirely written in javascript and it is the first to allow a simple web camera to detect human faces and locate eye gazes in real-time [17].

B. Voice Recognition

Voice recognition is a technology that allows computers to recognize spoken words and treat it as inputs for a computer program or as text on a screen [8]. It is considered an alternative medium for typing on keyboards. Voice recognition technology can save time and energy for people especially whose jobs include encoding information [6]. Vast software programs have been developed with this technology to provide a more efficient way of putting speech into words.

Voice recognition technology started in 1952 with the system named *Audrey* developed by Bell Laboratories [11]. Given the complicated syntax of human language, developers focused on simple terms [12] thus, making *Audrey* limited to understanding numbers and a few specific people only [11]. After ten years, IBM released *Shoebox* which when compared to *Audrey*, could understand sixteen (16) English words [12]. *Shoebox*, even though it has made a huge step in the field of voice recognition, wasn't enough to cater the needs of the public [11].

In Operating Systems, voice recognition technology was also implemented by Microsoft [3]. Microsoft Word has a diction feature that lets users dictate the text instead of typing it using the keyboard. Mac OS X Mountain Lion also has a built-in diction features called *Mountain Lions* and *Nuances Dragon Dictate* [13]. In smartphones, voice recognition implementations include Google's *Voice Search* in Android and *Siri* in iOS.

In the web, *Dictation Online Speech Recognition* allows users to create e-mails and documents with the use of Google Chrome's built-in speech recognition system [3]. There is an application programming interface (API) that can be supported by web browsers and that is *Web Speech API*. *Web Speech API* is developed by the W3C Speech API Community Group in JavaScript. Using *Web Speech API*, voice recognition in web browsers can be implemented and can open more possibilities for more advanced studies [14]. Examples of these implementations were from Ventocilla and Recario [3] and from Almazar and Recario [6]. They applied Web Speech API into a Google Chrome Extension as a mean to navigate through active web pages using basic voice commands (scrolling) and advanced voice commands (clicking, searching, etc). Despite of the many studies that have sprouted that implements voice recognition, Web Speech API is W3C standard which makes is more reliable to use than others.

C. Multimodal Interfaces

Multimodal interfaces gives a user a choice on how to interact with a computer. Users can either choose the usual keyboard and mouse combination. But for people who are having difficulties or are physically incapable of using the usual keyboard and mouse combination, they can use other ways. Implementing multimodal interfaces on systems increases usability. The weakness of one modality can be the strength of the other [1]. It can also be used by physically disabled users or users that would prefer other modalities rather than using the usual keyboard and mouse combination.

A system was developed for controlling a computer using head movements and voice commands [1]. It tracks head movements and uses voice recognition for navigation. Another application of multimodal interfaces is the development of a gesture and speech interface for controlling and interacting with three-dimensional displays and graphical objects of biomolecular systems in structural biology. With this, it simplified model the process of manipulation of biomolecular systems in structural biology. This also allows researchers to try different variations of the model without having to worry about computation and hardware limitations [18].

Despite these implementations, there is no Chrome extension that implements the two modalities mentions. This study will implement voice recognition and gaze navigation in a Google Chrome extension. This study can contribute to developing more multimodal interfaces in the web in the future.

III. Methodology

This study aims to navigate most active web pages using gaze and voice navigation in a Chrome extension. This study would be a step towards *keyboardless* and *mouseless* web browsing. system requirements and functional requirements are discussed in this section.

A. System Requirements

The following tools are needed for the development of the Google Chrome extension:

1) HTML and CSS: HTML and CSS will be used for the layout and design of the user interface of the extension. Minimal interface, notifications and status of the extension, will be shown to the user.
2) Javascript: Javascript is the scripting language used by the web browsers. It is also used by the technology needed for gaze navigation and voice recognition. This will be the backbone in developing the extension.
3) Web Speech API: Web Speech API enables speech-input, text-to-speech output that are not available using common speech recognition software [19].
4) Webgazer.js: A real-time eye tracking Javascript library that enables common web cameras to locate eye-gazes of users on a web page. It runs only on client browser thus no data will be gathered and will be sent to the server [20].

B. Functional Requirements

The main functionalities of the Google Chrome extension are listed below:

1) Settings Tab
   The Settings tab (see Figure 1) can be accessed by clicking the extension icon that is beside the address bar a browser. In this tab, the user can turn on or off the extension.

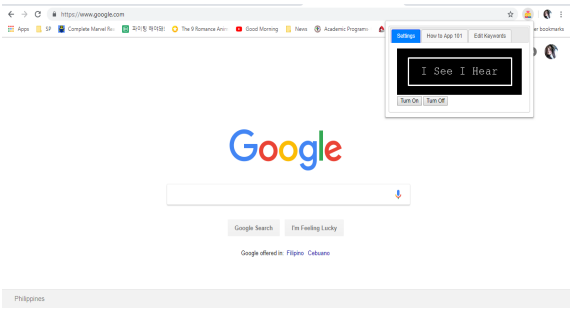

Fig. 1. Settings Tab

2) Bookmarks Tab
   The Bookmarks tab (see Figure 2) can be accessed by clicking the extension icon on the right of the browser's address bar. In this tab, the user can see all of the saved customized keywords. This tab also allows the user to delete customized keywords that the user created beforehand using the *"Add"* function.

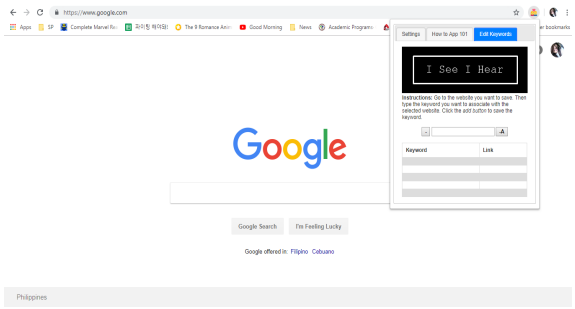

Fig. 2. Bookmarks Tab

Basic Commands

Basic commands of the extension allow the users to navigate through the web page without interacting and performing any action with the web page elements. Flowchart of how basic commands work is shown in Figure 3. Listed below are the basic commands of this extension:

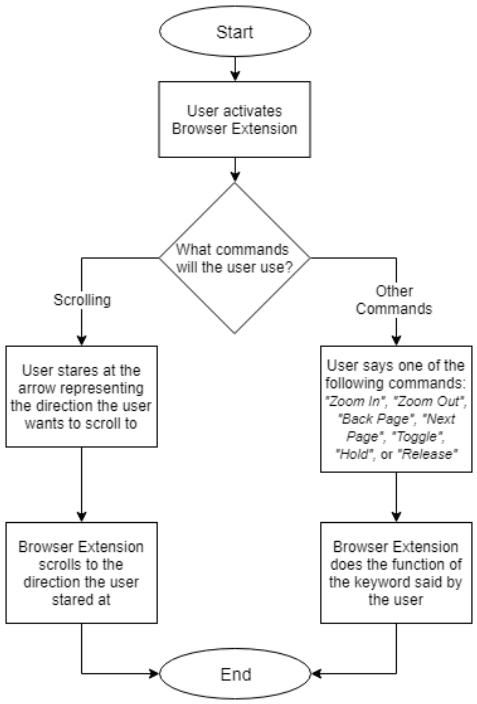

Fig. 3. Flowchart for Basic Commands

3) Scroll + *direction*
   Arrows on the four cardinal directions (north, south, west, and east) of every web page (see Figure 4) will be provided as a gazing point. Continuously stare at the arrows to scroll in the user's desired direction.
4) "Back Page" and "Next Page"
   These functions allow the user to go back or forward on the browser's history. The user needs to say the keyword *"Back Page"* to go back to the previous page and *"Next*

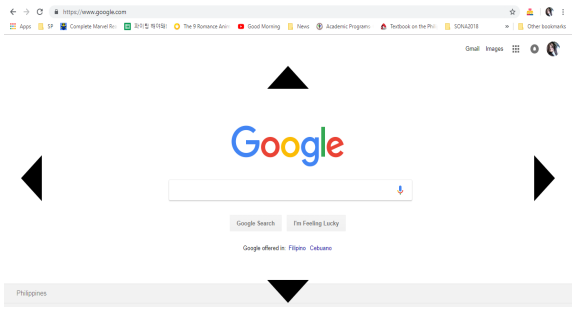

Fig. 4. Arrows for scrolling

Page" to go forward on the browser's history.

5) "Hold" and "Release"
These let the user to temporarily disable the functionalities of the extension. The user needs to say *"Hold"* to temporarily disable the browser extension's functionalities and *"Release"* if the user wants to enable the functionalities of the extension. When the extension is *on hold*, the interface will be hidden. The interface will be visible again when the user *releases* the functionalities of the extension.
6) "Toggle"
This allows the user to change the interface from arrows to other functions and vice versa when the extension detects the keyword *"Toggle"* from the user.

Advanced Commands

Advanced commands of the extension allow interaction between the user and the web page elements. Listed below are the advanced commands of this extension:

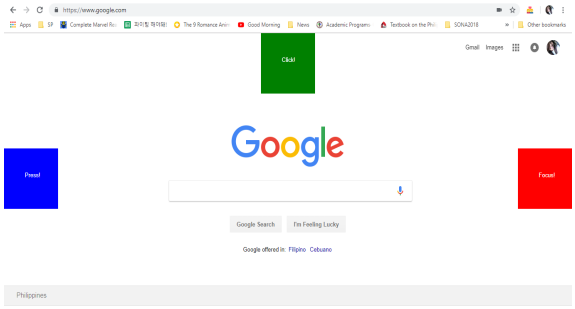

Fig. 5. Interface for Advanced Commands

7) Click + *link label*
This enables the user to navigate another web page. To activate, continuously stare at the *"Click"* button provided on the interface (see Figure 5). Numerical labels will be tagged to links on the active web page. The user must say the numerical label tagged to the link the user wants to click. Flowchart of how the *Click* functionality works is shown in Figure 6.
8) Press + *button name*
This allows the user to press a button on a web page. To activate, continuously stare at the *"Press"* button provided on the interface (see Figure 5). Numerical labels will be tagged to buttons on the active web page. The user must say the numerical label tagged to the button the user wants to press. Flowchart of how the *Press* functionality works is shown in Figure 6.

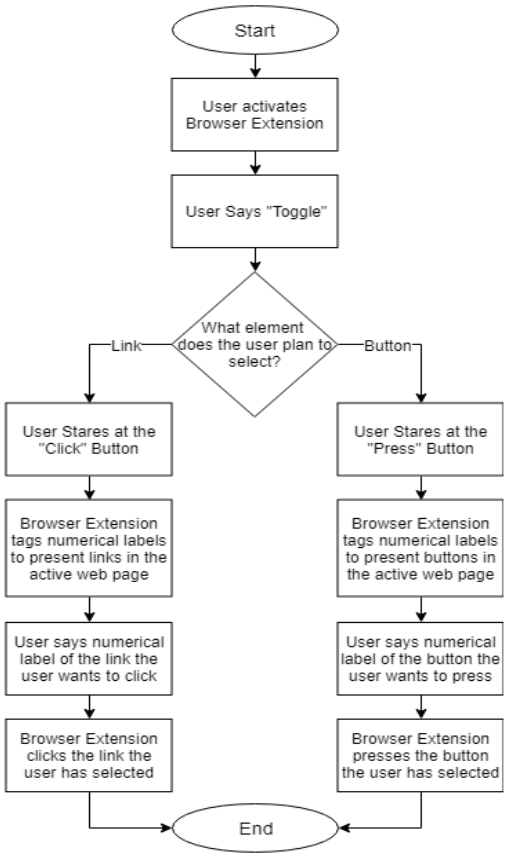

Fig. 6. Flowchart for the Click and Press functionalities

9) Focus + *field label*
This is applied in online forms. To activate, continuously stare at the *"Focus"* button provided on the interface (see Figure 5). Numerical labels will be tagged to text fields on the active web page. The user must say the numerical label tagged to the text fields the user wants to focus on. The next words the user says are shown on the focused text field. To stop filling up the focused text field, say the keyword *"Stop Focus"*. Flowchart of how the *Focus* functionality works is shown in Figure 7.
10) Add + *bookmark*
To add bookmarks, the user needs to go to the web page the user wants to save as a bookmark. Then, say the keyword *"Add"*. The browser extension will wait for the user to say the word the user wants to use as bookmark for the active web page. The browser extension will say that the bookmark is added to the bookmarks tab. The bookmark directs to the web page it is associated with and you can say the bookmark as long as it is not deleted on the bookmarks tab (see Figure 2). This extension only allows a maximum of five (5) bookmarks. Flowchart of

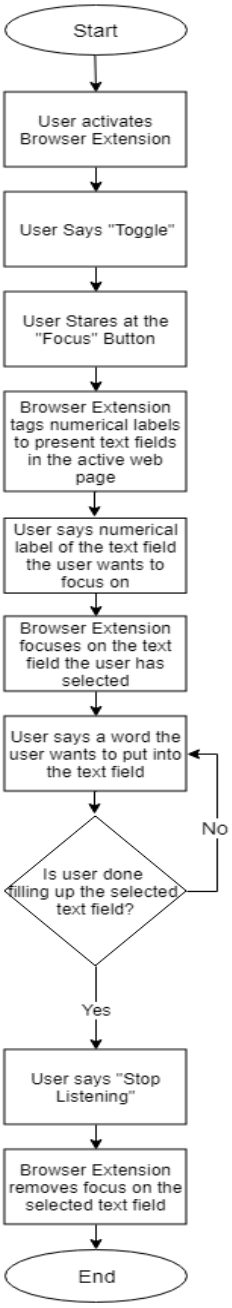

Fig. 7. Flowchart for the Focus functionality

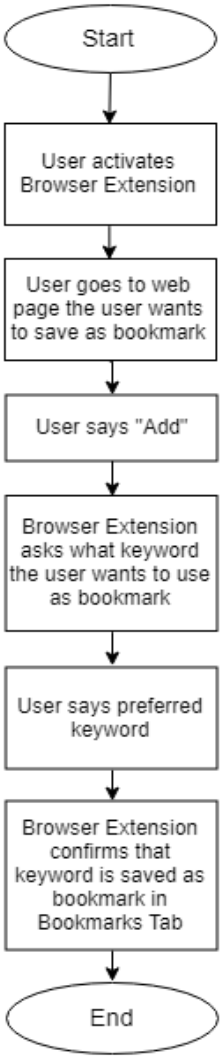

Fig. 8. Flowchart for the Add Bookmark functionality

how this functionality works is shown in Figure 8.

C. Measure of Effectiveness

A modified questionnaire will be answered by testers to prove if the Google Chrome extension was effective and efficient enough for public use. This modified questionnaire is based from the Computer System Usability Questionnaire (CSUQ) [21] and Quiz for User Interface Satisfaction (QUIS) [22].

The QUIS is made up of four (4) sections evaluating the following: the terminology used and system information, learning how to use the extension, system capabilities, and the overall reaction to the extension. Analysis of data will be interpreted using Top two box (78%) scoring, which means that the score was evaluated as satisfactory if a score falls within the top two box and a score falling in the middle category is considered as a neutral response between not satisfactory (bottom two box) and satisfactory (top two box) [23]. The survey questions will be rated from 1 (being the most negative reaction to the question) to 5 (being the most positive reaction to the question). For the CSUQ, the questions are categorized into two types namely: User-friendliness, and Extension Design Efficiency. Questions 1 2,6,7,8,9,10, and 11 is applied to user-friendliness while questions 3,4,5,12, and 13 addresses extension design efficiency. Questions for both QUIS and CSUQ is shown on the Appendices.

IV. RESULTS AND DISCUSSION

V. CONCLUSION AND FUTURE WORK

APPENDIX A
QUESTIONS IN QUESTIONNAIRE FOR USER INTERFACE SATISFACTION (QUIS)

A. Terminology and System Information

1) Use of terms throughout system
2) Terminology is easily understandable
3) Position of messages on screen
4) Prompts for adjusting input in Settings tab
5) Error messages

B. Learning

1) Overall rating for learning to navigate web pages using the extension
2) Learning to operate and to adjust the extension settings
3) Learning to navigate a web page using the extension by voice
4) Learning to navigate a web page using the extension by gaze
5) Learning to navigate a web page using the extension by both voice and gaze
6) Performing tasks is straightforward
7) Use of terms throughout system

C. System Capabilities

1) Overall extension speed
2) Extension startup speed
3) Extension response speed in recognizing voice and/or gaze
4) Failure mechanisms: Errors can occur without warnings, Error messages are displayed correctly.
5) The extension is designed for all types of users.
6) Overall usefulness of the extension for web page navigation

D. Overall reaction to the extension

1) Difficult/Easy
2) Terrible/Wonderful
3) Frustrating/Satisfying
4) Dull/Stimulating
5) Rigid/Flexible

APPENDIX B
QUESTIONS IN COMPUTER SYSTEM USABILITY QUESTIONNAIRE (CSUQ)

Note: The questions were rated from Strongly Disagree to Strong Agree

A. User-friendliness

- Overall, I am satisfied with how easy it is to use the extension.
- It was simple to use the extension.
- The information and warnings (such as online help, on-screen messages, etc.) provided is clear.
- It is easy to find the information I needed.
- The information provided for the system is easy to understand.
- The organization of information on the system screen is clear.
- The interface of the extension is pleasant.
- I like using the interface of the extension.

B. Extension Design Efficiency

- I can effectively navigate web pages using the extension.
- I can effectively complete my work using the extension.
- I can effectively complete my work quickly using the extension.
- The extension has all the functions and capabilities I expect it to have.
- Overall, I am satisfied with the extension.

REFERENCES

[1] A Ismail, A. El Salam, A. Hajjar, and M. Hajjar, "A Prototype System for Controlling a Computer by Head Movements and Voice Commands," The *International Journal of Multimedia and Its Applications*, vol. 3, no. 3, Aug 2011.

[2] Why gaze tracking startup Cogisen is eyeing the Internet of Things, 2016. [Online]. Available: https://techcrunch.com/2016/06/01/why-gaze-tracking-startup-cogisen-is-eyeing-the-internet-of-things/

[3] F. Ventocilla and R. Recario, "Enabling Speech Navigation on Active Web Page using Google Chrome Extension," Ph.D. dissertation, 2014.

[4] "Internet Statistics & Facts (Including Mobile) for 2019 – HostingFacts.com," Dec 2019. [Online]. Available: https://hostingfacts.com/internet-facts-stats/

[5] "Browser extension," June 2017. [Online]. Available: $https://en.wikipedia.org/wiki/Browser_extension$

[6] A. Almazar and R. Recario, "Speech Navigation on Active Web Page using Google Chrome Extension and Web Speech API," Ph.D. dissertation, 2016.

[7] C. Juruena and K. Abriol-Santos, "eyeGalaw: A Google Chrome Extension for Webpage Navigation through Human Gaze," Institute of Computer Science, UPLB. Special Problem, 2017, unpublished paper.

[8] S. Halim and W. Budiharto, "The Framework of Navigation and Voice Recognition System of Robot Guidance for Supermarket," *International Journal of Software Engineering and Its Applications*, vol. 8, no. 10, pp. 143-152, 2014.

[9] P. Isokoski, M. Joos, O. Spakov, B. Martin, "Gaze Controlled Games," *Universal Access in the Information Society*, vol. 8, no. 4, p. 323337, May 2009.

[10] "Design, User Experience, and Usability: Interactive Experience Design". Springer, 2015. [Online]. Available: https://books.google.com.ph/books?id=e0UqBAAA QBAJ

[11] "A Brief History of Voice Control," Oct 2016. [Online]. Available: https://medium.com/@joshdotai/its-all-in-the-voicefba7e5a032fa

[12] "Speech Recognition Through the Decades: How We Ended Up With Siri," Nov 2011. [Online]. Available: $http://www.pcworld.com/article/243060/speech_recognition_through_the_decades_how_we_ended_up_with_siri.html$

[13] "Mountain Lion Dictation versus Dragon Dictate," Sept 2013. [Online]. Available: https://macworld.com/article/2014417/mountainlion-dictation-versus-dragon-dictate.html

[14] "Using Voice to Drive the Web: Intorduction to the Web Speech API," Sept 2013. [Online]. Available: http://www.adobe.com/devnet/html5/articles/voiceto-drive-the-web-introduction-to-speech-api.html

[15] A. Mohamed, M. da Silva, and V. Courboulay, “A History of Eye Gaze Tracking,” *Rapport Interne*, 2007.
[16] M. Porta and A. Ravelli, ”Weyeb, an eyecontrolled web browser for hands-free navigation,” *2009 2nd Conference on Human System Interactions*, 2009.
[17] A. Papoutsaki, P. Sangkloy, J. Laskey, N. Daskalova, J. Huang, and J. Hays, “Webgazer: Scalable Webcam Eye Tracking Using User Interactions,” *Proceedings of the 25th International Joint Conference on Artificial Intelligence*, pp. 3839-3845, 2016.
[18] R. Sharma, V. Pavlovic, and T. Huang, “Toward Multimodal Human-Computer Interface,” *Proceedings of the IEEE*, vol. 86, no. 5, May 1998.
[19] “Web Speech API Specification,” Oct 2012. [Online]. Available: https://dvcs.w3.org/hg/speechapi/raw-file/tip/speechapi.html
[20] “WebGazer.js Democratizing Webcam Eye Tracking on the Browser,” 2016. [Online]. Available: https://webgazer.cs.brown.edu
[21] J. R. Lewis, “IBM computer usability satisfaction question- naires: Psychometric evaluation and instructions for use,” *International Journal of Human-Computer Interaction, 7:1, 57-78.,* 1995.
[22] J. P. Chin, V. A. Diehl, and K. L. Norman, "Development of an instrument measuring user satisfaction of the human-computer interface," *CHI*,1988.
[23] "How to Interpret Survey Responses," 2011. [Online]. Available: https://measuringu.com/interpret-responses/