REDUCING COSTS IN HUMAN ASSISTED SPEECH TRANSCRIPTION

A Thesis

presented to

the Faculty of California Polytechnic State University,

San Luis Obispo

In Partial Fulfillment

of the Requirements for the Degree

Master of Science in Computer Science

by

Justin Rovin

March 2016
COMMITTEE MEMBERSHIP

TITLE: Reducing Costs in Human Assisted Speech
       Transcription

AUTHOR: Justin Rovin

DATE SUBMITTED: March 2016

COMMITTEE CHAIR: Alex Dekhtyar, Ph.D.
                 Professor of Computer Science

COMMITTEE MEMBER: Foaad Khosmood, Ph.D.
                  Professor of Computer Science

COMMITTEE MEMBER: Olga Dekhtyar, M.S.
                  Lecturer of Statistics
Reducing Costs in Human Assisted Speech Transcription

Justin Rovin

The only official documentation of the lawmaking process at the California Legislature is unedited video recordings of committee hearings, bill texts, votes and analyses. While the bills resulting from these hearings are clear, using video recordings to understand how a bill was created is far too laborious for the average citizen. To increase public transparency, a service that provides easier access to the bill creation process was needed. In response to this need, the Digital Democracy initiative was established at Cal Poly by the Honorable Sam Blakeslee, former California State Senator and founder of the Institute for Advanced Technology and Public Policy.

The Digital Democracy initiative seeks to create a web platform that organizes, generates, and indexes large amounts of information about the legislative process. To accomplish this, automatic speech recognition is performed on the video recordings of committee hearings and the resulting text is manually improved and annotated with a web application called the “Transcription Tool”. Unfortunately, this process is costly, labor intensive, and prohibits the scaling and long term viability of the platform. Early efforts to reduce transcription costs involved the development of improved transcription tool UI and systems for speaker diarization and text correction.

This thesis evaluates the effectiveness of these improvements on the human assisted transcription process employed by the Digital Democracy initiative. To facilitate this evaluation, a pipeline for automatic transcription improvement was developed, the improvements were incorporated into the transcription process, and a controlled experiment was run to measure the effects of these improvements. The results of the experiment demonstrate that the improvements reduced transcription editing costs by 16.89% while maintaining similar transcription quality.
ACKNOWLEDGMENTS

Thank you to my advisors Dr. Alex Dehktyar and Dr. Foaad Khosmood for trusting me with the management and development of an application critical to the success of the Digital Democracy Initiative. The high bar they set gave me the opportunity to apply myself and grow at my full potential, and for that I am very grateful.

Thank you to my father for his constant support and encouragement.
## TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>LIST OF TABLES</th>
<th>xii</th>
</tr>
</thead>
<tbody>
<tr>
<td>LIST OF FIGURES</td>
<td>xiii</td>
</tr>
<tr>
<td><strong>CHAPTER</strong></td>
<td></td>
</tr>
<tr>
<td>1 INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>2 BACKGROUND</td>
<td>6</td>
</tr>
<tr>
<td>2.1 The State of Transparency in Government</td>
<td>6</td>
</tr>
<tr>
<td>2.2 Digital Democracy Initiative</td>
<td>8</td>
</tr>
<tr>
<td>2.3 DigitalDemocracy.org</td>
<td>10</td>
</tr>
<tr>
<td>2.3.1 Content Pages</td>
<td>11</td>
</tr>
<tr>
<td>Speaker Page</td>
<td>11</td>
</tr>
<tr>
<td>Bill Page</td>
<td>12</td>
</tr>
<tr>
<td>Hearing Page</td>
<td>13</td>
</tr>
<tr>
<td>2.3.2 Search Functionality</td>
<td>14</td>
</tr>
<tr>
<td>2.4 Data Sources</td>
<td>14</td>
</tr>
<tr>
<td>2.4.1 Online Sources</td>
<td>14</td>
</tr>
<tr>
<td>2.4.2 Offline Data Procurement</td>
<td>16</td>
</tr>
<tr>
<td>2.5 System for Procuring Data</td>
<td>17</td>
</tr>
<tr>
<td>2.5.1 Scripts</td>
<td>18</td>
</tr>
<tr>
<td>2.5.2 Transcription System</td>
<td>18</td>
</tr>
<tr>
<td>Automatic Speech Recognition</td>
<td>18</td>
</tr>
<tr>
<td>Speaker Independent</td>
<td>19</td>
</tr>
<tr>
<td>Speaker Dependent</td>
<td>19</td>
</tr>
<tr>
<td>Speaker Adapted</td>
<td>20</td>
</tr>
<tr>
<td>2.5.3 Transcription Tool</td>
<td>20</td>
</tr>
<tr>
<td>6.4  Procedure</td>
<td>77</td>
</tr>
<tr>
<td>---------------------------------</td>
<td>----</td>
</tr>
<tr>
<td>6.4.1 Group Rotation</td>
<td>77</td>
</tr>
<tr>
<td>6.4.2 Scheduling</td>
<td>77</td>
</tr>
<tr>
<td>6.4.3 Environment</td>
<td>78</td>
</tr>
<tr>
<td>6.4.4 Instructions</td>
<td>78</td>
</tr>
<tr>
<td>6.4.5 Gathering Data</td>
<td>78</td>
</tr>
<tr>
<td>7  DATA ANALYSIS</td>
<td>80</td>
</tr>
<tr>
<td>7.1  Metrics</td>
<td>81</td>
</tr>
<tr>
<td>7.1.1 Task Duration</td>
<td>81</td>
</tr>
<tr>
<td>7.1.2 Text Alignment</td>
<td>82</td>
</tr>
<tr>
<td>Needlemann–Wunsch Algorithm</td>
<td>82</td>
</tr>
<tr>
<td>7.1.3 Speaker Assignment</td>
<td>83</td>
</tr>
<tr>
<td>7.1.4 Utterance Length</td>
<td>84</td>
</tr>
<tr>
<td>7.2  Analysis</td>
<td>85</td>
</tr>
<tr>
<td>7.2.1 Outlier</td>
<td>85</td>
</tr>
<tr>
<td>7.2.2 Averages</td>
<td>86</td>
</tr>
<tr>
<td>7.2.3 Regression</td>
<td>87</td>
</tr>
<tr>
<td>Baseline: Tool 1</td>
<td>87</td>
</tr>
<tr>
<td>Tool 2</td>
<td>88</td>
</tr>
<tr>
<td>Tool 3</td>
<td>88</td>
</tr>
<tr>
<td>Tool 4</td>
<td>88</td>
</tr>
<tr>
<td>7.2.4 Result Summary</td>
<td>89</td>
</tr>
<tr>
<td>7.3  Total Cost Evaluation</td>
<td>89</td>
</tr>
<tr>
<td>7.3.1 Constant Estimation</td>
<td>90</td>
</tr>
<tr>
<td>7.3.2 Total Cost Results</td>
<td>91</td>
</tr>
<tr>
<td>7.4  Threats to Validity</td>
<td>92</td>
</tr>
<tr>
<td>8  CONCLUSION AND FUTURE WORK</td>
<td>93</td>
</tr>
</tbody>
</table>
# LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.1 Table of editors</td>
<td>69</td>
</tr>
<tr>
<td>6.2 Table of selected hearing videos</td>
<td>75</td>
</tr>
<tr>
<td>6.3 Table of group task rotation</td>
<td>77</td>
</tr>
<tr>
<td>7.1 Table of regression results</td>
<td>87</td>
</tr>
<tr>
<td>7.2 Transcription cost model approximated constants</td>
<td>91</td>
</tr>
<tr>
<td>7.3 Average transcription cost by experiment round</td>
<td>92</td>
</tr>
</tbody>
</table>
## LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1 Home page of the digital democracy platform</td>
<td>10</td>
</tr>
<tr>
<td>2.2 Example speaker page</td>
<td>11</td>
</tr>
<tr>
<td>2.3 Example bill page</td>
<td>12</td>
</tr>
<tr>
<td>2.4 Example hearing page</td>
<td>13</td>
</tr>
<tr>
<td>2.5 Diagram of data procurement system</td>
<td>17</td>
</tr>
<tr>
<td>2.6 Editing UI of the version 1 transcription tool</td>
<td>23</td>
</tr>
<tr>
<td>2.7 Editing UI header information</td>
<td>24</td>
</tr>
<tr>
<td>2.8 Utterance editing UI elements</td>
<td>25</td>
</tr>
<tr>
<td>2.9 Transcription speaker importation UI</td>
<td>27</td>
</tr>
<tr>
<td>2.10 Unfilled hearing creation UI</td>
<td>32</td>
</tr>
<tr>
<td>2.11 Hearing committee selection UI</td>
<td>33</td>
</tr>
<tr>
<td>2.12 Hearing video importation UI</td>
<td>33</td>
</tr>
<tr>
<td>2.13 Bill discussion importation UI</td>
<td>34</td>
</tr>
<tr>
<td>4.1 Committee member importation button location</td>
<td>45</td>
</tr>
<tr>
<td>4.2 Cascade check box location</td>
<td>47</td>
</tr>
<tr>
<td>4.3 Task-wide alignment selector location</td>
<td>48</td>
</tr>
<tr>
<td>4.4 An utterance before a split</td>
<td>49</td>
</tr>
<tr>
<td>4.5 The resulting two utterances after a split</td>
<td>49</td>
</tr>
<tr>
<td>4.6 A speaker selection drop down segregated by classification</td>
<td>51</td>
</tr>
<tr>
<td>4.7 Utterance text box size in the original tool</td>
<td>52</td>
</tr>
<tr>
<td>4.8 Expanded utterance text boxes</td>
<td>52</td>
</tr>
<tr>
<td>4.9 Diarization UI</td>
<td>56</td>
</tr>
</tbody>
</table>
CHAPTER 1

INTRODUCTION

State Legislatures across the country have deficiencies in public accountability and transparency, and this problem is especially notable in the State of California. The Legislature at the California State Capitol is remarkably detached from any modern form of digital information collection, storage or exchange. The only accounting of the legislative meetings at the capitol exist either directly as unedited video recordings, or indirectly as completed bills, written committee analyses, or reports of final votes [7]. The documentation is minimal and the video recordings require prohibitive amounts of time and effort to extract details about the evolution of a piece of legislature. The result of the current record keeping is that it is too costly for average citizens to understand and engage in the creation of our laws and to keep legislators accountable for their decisions. In the absence of any efforts to meticulously study these meeting videos, the promises and statements made may be forgotten, and the identities of participants and their voiced opinions may go unreported.

In an attempt to increase the transparency of our lawmaking process, the Institute of Advanced Technology and Public Policy at Cal Poly began the Digital Democracy initiative as an effort to develop a comprehensive web platform for
gathering and researching information about the legislative process [1]. One of the primary goals of the initiative is to transcribe videos of legislative hearings in order to extract information about their content. Using these transcriptions, the project aims to associate hearings and individuals with words spoken, and while doing so, create a conveniently searchable database of utterances spoken by everyone influencing the creation of our laws. As a prototype solution, the video records of these meetings are being processed by a video transcription and indexing service named InCus. This is a commercial implementation of an automatic speech recognition (ASR) system developed by Microsoft Research known as MAVIS [5]. Unfortunately, the resulting transcriptions contain numerous errors and do not associate utterances with speakers. Both manual error correction and annotation are required before a transcript can be a functional and reliable record of a legislative hearing.

To facilitate this manual error correction and annotation, a web application referred to as the “Transcription Tool” was developed. In an initial implementation of the tool, administrators would import unedited transcription files and details about the associated hearing to be transcribed. The tool would take this data and produce “transcription tasks” for editors to correct and annotate portions of a hearing transcription. Unfortunately, the resources needed to use the automatic speech recognition service and fund manual processing with the transcription tool were not practical at scale. In an effort to increase scalability, several improvements have been introduced into this transcription system including: speaker diarization, automatic text correction, and several UI enhancements.

This thesis investigates the effectiveness of the improvement projects in the production of corrected and annotated transcriptions. To determine the effectiveness of these technologies, several metrics were devised to measure transcription
quality, including: speaker assignment accuracy, utterance length, manual editing duration, and text quality. To gather data for these metrics, fifteen participants were asked to transcribe four different legislative meetings using four different implementations of the transcription tool. The first implementation was the transcription tool in its initial state, the following three implementations each gained one of the three improvement projects being tested. Additionally, all of the tools were modified to collect telemetry data from the editors.

The metrics for each transcription were extracted from telemetry data and by comparing resulting "test transcriptions" with correct "control transcriptions". The effect of each tool on the various metrics is then calculated using linear regression. These regressions show that the combination of all the improvements demonstrated a 16.89% decrease in transcription editing duration while transcription quality did not significantly change.

The main conclusion of these results is that editing speed has been decreased from an average of 7:37 minutes per minute of hearing video to 6:19 minutes per minute of video, for a 16.89% speedup with no change in transcription quality. Correspondingly, assuming that all editing errors are corrected, total transcription costs have been reduced by 10.85%.

The main contributions of this thesis are:

- A production implementation of a pipeline for selective transcription pre-processing
- The incorporation of three significant transcription improvement projects for the Digital Democracy initiative
- A metric-based transcription cost model
- A controlled experiment and evaluation of four transcription tool editing environments.
The rest of the document is summarized below with chapter titles and content descriptions.

**Background**  This section describes: the problems that the Digital Democracy initiative seeks to address, the Digital Democracy platform, the purpose and design of the “Transcription Tool”, the state of the initiative, the transcription bottleneck that this thesis addresses, and the improvements being used to alleviate the bottleneck.

**Contribution**  This section provides an overview of the contributions of this thesis including: modifications made to the transcription system, an evaluation of the three editing improvements, and a transcription cost model to determine the cost savings of the improvements.

**Improvements**  This section describes the three improvement projects incorporated into the transcription system.

**System Design**  This section describes: the modifications made to the transcription system to facilitate the implementation and evaluation of the three improvements, and definitions of the four transcription environments used in the evaluation experiment.

**Experiment Design**  This section describes: the experiment conducted to evaluate the three different improvements, details about volunteer selection, transcription task selection, how transcription tasks were assigned to volunteers, how the experiment was conducted, and how data was collected.
<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Analysis</td>
<td>This section describes: the metrics created to characterize transcription quality, how data was gathered and evaluated, and the results of the linear regressions performed on the experiment data.</td>
</tr>
<tr>
<td>Conclusion &amp; Future Work</td>
<td>This section describes the conclusions drawn from results of the data analysis and the future improvements planned to further reduce transcription costs.</td>
</tr>
<tr>
<td>Related Work</td>
<td>This section describes several other works that have done research similar to that contributed by this thesis.</td>
</tr>
</tbody>
</table>
CHAPTER 2

BACKGROUND

An outline of the sections of this chapter are shown here:

2.1 Transparency at the California Legislature
2.2 The creation of the Digital Democracy Initiative
2.3 The content of the online Digital Democracy platform
2.4 Data sources
2.5 The data procurement system
2.6 The design and implementation of the hearing transcription system
2.7 The bottleneck in the transcription system addressed by this thesis

2.1 The State of Transparency in Government

“Government operates on a default of secrecy...” [11]. This quote, from Lt. Gov. Gavin Newsom of California, speaks to the F grade given to California by the CALPRIG Education Fund for online access to spending information [10]. A recent report by the U.S. PIRG Education Fund investigated state transparency websites, evaluating state progress toward “... comprehensive, one-stop, one-click transparency and accountability for state government spending.” California was found to be the lowest scoring state in the country, reporting that California
“...has a long way to go,” in terms of innovation in online transparency.

At the California State Capitol during Senate and Assembly committee hearings no transcripts or meeting minutes are taken, there is no record of who spoke or attended, nor is there any information regarding the development of legislation. The only data collected from a committee hearing is a video recording of the meeting, the final texts of any bills voted on, bill analyses, and bill votes[7]. The limited information kept on committee hearings is problematic not only for citizens who want to be engaged in the law making process, but also makes keeping track of meeting discussions challenging for legislators, and curbs the ability of the media to keep the public informed and lawmakers accountable. This problem has been amplified since the popularization of the internet and the downturn of traditional print newspapers. In a report by the Pew Research Center, the amount of statehouse reporters was shown to be dwindling with the decline of traditional news outlets [9]. “I think you’re seeing fewer stories... the public is not being kept aware of important policy decisions that are being made that will affect their daily lives,” said Gene Rose, former communications director for the National Conference of State Legislatures [9]. The poor record keeping at the State Capitol also hurts the legislative process itself. Sam Blakeslee, former Senator and minority leader in the state Assembly, comments on the difficulty of keeping other legislators accountable for promises made during legislative meetings. “I recall a number of occasions during my time in the legislature when I would see a bill come before me that appeared to not include amendments that had been promised...In those moments, I simply had no way to quickly go back and search the record of what was said in the committee hearing.”[12]

There have been efforts to increase transparency; Phillip Ung, former spokesperson of California Common Cause, attempted to persuade the California Legisla-
ture to develop and fund a system to increase transparency, but was dismissed on the account “...it was too complicated, too expensive and not a priority.”[11] The continued lack of interest in transparency initiatives at the California Legislature prompted the need for an investment from a private institution.

2.2 Digital Democracy Initiative

Intent on resolving the transparency issues discussed in section 2.1, The Institute of Advanced Technology and Public Policy at Cal Poly, a non-profit founded by former California Senator Sam Blakeslee, launched the Digital Democracy initiative. The ultimate goal of the initiative is to develop an online platform that can provide easy access to current information about the lawmaking process at all levels of government. By keeping timely information about the content of recent legislative hearings, the platform would enable users of the site to engage the legislative process in real time. This platform would allow users to follow the creation of bills, investigate the histories of legislators, look up influential parties in government, trace relationships and political ties, look up the donors and recipients of money in politics, or be alerted of issues of interest among a variety of other uses. By gathering all of this data into one place not only is it more accessible, but by applying various data analysis techniques the Digital Democracy initiative can elucidate new data sets, perspectives, and analyses on the otherwise complex and obscure inner workings of our government.

At the inception of the Digital Democracy initiative, Cal Poly faculty and students were hired to devise and develop the online platform. The initiative is a cross disciplinary effort, combining the expertise of both the Cal Poly Political Science and Computer Science departments. The involvement of the students
and faculty with the initiative has been significant not just for its contribution to society, but also because the project has created research and learning opportunities not usually available in undergraduate education. The project has spawned a myriad of research projects, all being tackled by teams of undergraduate and graduate students, utilizing modern industry technologies and techniques including machine learning, computer vision, and data mining.
2.3 DigitalDemocracy.org

A beta version of the platform can now be found at www.DigitalDemocracy.org. There are currently several main search and browse functionalities and several content types to be viewed.

Figure 2.1: Home page of the digital democracy platform
2.3.1 Content Pages

Speaker Page

Speakers are people that have participated in a legislative hearing and have been identified. A Speaker Page can contain a biography, contact information, a picture of that person, their affiliations, contributions they have received, links to hearings they have participated in, or bills they have discussed or authored.

Figure 2.2: Example speaker page
Bill Page

A Bill Page displays its author, name, subject, house and current status. Listed below its basic profile information is the bill’s text, summary, analysis, and its history of votes. Along the border of the Bill Page is a list of every hearing where that bill was discussed. Clicking on the listed Bill Discussions take you directly to the beginning of the that discussion in that particular hearing.

Figure 2.3: Example bill page
Hearing Page

Hearing Pages are where transcriptions and videos of hearings are displayed. At the top of the page is the name of the house and committee holding the hearing and the date of the hearing. The center of a Hearing page includes a video of that hearing, and directly below is a full transcription of all the utterances spoken during that video. Each row in the list of utterances contains words spoken and the name of speaker. Clicking on a speaker name takes you to the speaker page of that person.

Figure 2.4: Example hearing page
As the video plays, the utterances displayed below the video automatically scroll to follow along with its progress. Conversely, clicking on any utterance in the transcript automatically brings the video to the moment when that utterance was spoken. The bills discussed during the hearing are displayed along the border of the hearing page. Below the list of discussed bills are visualizations displaying the district of the current speaker and measures of the relative participation of each legislator in that hearing.

2.3.2 Search Functionality

The search functionality prominently displayed on the home page allows the search of utterances across all hearings on the site. Words, phrases, bills, speakers, and committee names can be searched and any hearing where the term was mentioned is returned. The results from this search can then be further filtered by date. Additionally, Bill Pages can be searched by author, keywords, number, or type. Hearing Pages can be browsed chronologically. Speaker Pages can be searched by name and/or occupation classification.

2.4 Data Sources

This section will describe the sources of data used by the initiative to produce content for www.DigitalDemocracy.org.

2.4.1 Online Sources

The following table outlines the public online data repositories used by the initiative and the kinds of data obtained from those sources.
CalChannel
www.calchannel.com

LegInfo
www.leginfo.legislature.ca.gov

Ca. Senate Website
www.senate.ca.gov

Ca. Assembly Website
www.assembly.ca.gov

OpenStates
http://openstates.org/ca/

Cal-Access
http://cal-access.ss.ca.gov/

MapLight
http://maplight.org/

- Committee Hearing Videos
- Legislators
- Bill Texts
- Bill Votes
- Bill Summary
- Bill Analyses
- Senate Committees
- Senate Committee Membership
- Assembly Committees
- Assembly Committee Membership
- Legislator District Information
- Lobbyists
- Lobbying Firms
- Lobbyist Employers
- Lobbying Contributions
2.4.2 Offline Data Procurement

Data that cannot be sourced directly from the web includes: transcripts of legislative hearings, the identities, roles, and affiliations of all the hearing participants, and the words spoken by a particular participant. A human assisted transcription system was developed to extract this information from hearing videos. This system will be described in the next section.
2.5 System for Procuring Data

The main database that houses all of the procured data is called the Digital Democracy Database or DDDB. Data makes its way into the DDDB in one of two ways: pulled from online sources and inserted by scripts, or inserted by the transcription tool after a video transcription has been manually processed. While much data can be gathered from existing sources online, transcription texts and annotations are manually curated from uncorrected transcriptions generated by an automatic speech recognition system. The transcription tool facilitates this correction and annotation process and inserts the resulting transcription information into the DDDB. On a nightly basis, data needed for the website is copied from the DDDB into the database of the Drupal instance that serves the website.

Figure 2.5: Diagram of data procurement system
2.5.1 Scripts

Python scripts are used to automatically pull new or updated data from many of the online data sources. Different types of information is retrieved with varying regularity. Information about Legislators and Committees are only retrieved at the beginning of new legislative sessions, while Bill and Lobbyist information is retrieved on a daily basis.

2.5.2 Transcription System

To extract information from hearing videos they must be manually downloaded from CalChannel and clipped into 20 minute segments. These video clips are then uploaded to Youtube for use in the transcription tool and on the Digital Democracy platform. The hearing video clips are also uploaded to an automatic speech recognition service to then receive a generated raw transcription.

Automatic Speech Recognition

Automatic speech recognition (ASR) is the act of converting spoken words into text. The initiative uses ASR to convert the audio signal of hearing videos into transcriptions of the words spoken during those hearings. The ASR system used by the initiative is an online service previously known as InCus Greenlight. InCus Greenlight was the commercial implementation of an advanced, speaker independent automatic speech recognition system created by Microsoft known as MAVIS[5]. The service provided by Greenlight has been integrated into Microsoft’s Azure platform, which is now in use by the initiative for its raw transcriptions.
ASR systems are composed of three main components: An acoustic model, a language model, and a decoder. From an inputted audio signal, an acoustic model is used to probabilistically match small sections of the signal with particular words. Using the various words approximated by the acoustic model, a language model is used to determine the probabilities that different sequences of those words could have appeared together. Lastly, a decoder takes the probabilities generated by the acoustic model and the language model, and outputs the sequence of words with the highest probability of originating from the inputted audio signal. The acoustic model in use by an ASR system is largely the determining factor in the accuracy of that system. There are three main types of acoustic models:

**Speaker Independent**

This is the type of model used by MAVIS. Speaker independent (SI) models are used when there can be significant variability in the audio signal being processed. Speaker independent systems are used when the speakers in an audio signal are not known ahead of time. SI models are considered the least accurate, but can be used with audio signals of high variability.

**Speaker Dependent**

Speaker dependent (SD) models are used when there is little variability in the audio signal being processed. These systems are typically used in voice command interfaces where speakers are known and the system can be trained for those particular speakers. SD models are considered to be the most accurate, but are only reliable when trained and used with particular audio signals of low variability.
Speaker Adapted

In speaker adapted (SA) models, generic acoustic models are augmented and expanded over time to produce transcription accuracy comparable to speaker dependent models, but useful across a larger range of signal variability. These types of models are used in the most advanced systems, but require vast amounts of training data.

2.5.3 Transcription Tool

Though MAVIS is a state-of-the-art speaker-independent ASR system, the generated hearing texts are riddled with errors and require significant correction and annotation before they can be displayed on the Digital Democracy platform. To prepare a raw transcription it must have it’s errors corrected, the words of the transcription must be broken into distinct utterances and given correct time intervals, speakers in the hearing must be identified and profiled, and the utterances of the transcription must then be associated with the speakers of that hearing. A web application, referred to as the “Transcription Tool”, is used by employees of the initiative to facilitate the correction, annotation, and uploading of that transcription information to the DDDB.

2.5.4 DDDB and Drupal

The DDDB is a MySQL database that acts as the central repository of data for the Digital Democracy initiative. The Drupal database that supports www.DigitalDemocracy.org either updates or imports new data from the DDDB on a nightly basis. Because Drupal is a content management system that uses a very specialized schema for its database, using it to maintain data in a meaningful way
is not possible. To maintain meaningful relationships among the entities in our data and to facilitate interpretation for both practical and research purposes, the DDDB is kept in a format separate from the format used by the Drupal platform.

2.6 Original Transcription Tool

This section will describe the purpose, technologies, interface design, and general architecture of the original transcription tool.

2.6.1 Core Functionality

The transcription tool has two core requirements.

1. It must facilitate the association and importation of hearing information into the DDDB. To import a new hearing, the tool requires a hearing date, Youtube IDs of the videos associated with that hearing, and the generated transcripts of those videos. Additionally, the tool pulls information about committees, bills, and bill discussions and allows administrators to associate those entities with the hearing being transcribed.

2. It must facilitate the correction and annotation of hearing transcriptions. To do this, the tool provides an interface that allows a user of the tool to modify and annotate the words from the original transcription file. From this interface, users can create and modify utterances, give utterances time intervals that correspond with the hearing video, and assign speakers to utterances. Employees of the initiative use the tool to perform these modifications on sections of raw transcriptions referred to as “transcription tasks”. Transcription tasks are typically split into assignments of 5 to 20 minutes
of hearing video. These tasks are assigned to editors by administrators of the transcription tool, and the editors use the tool as a portal to access the tasks assigned to them. Once a task has been completed, the tool uploads the finished transcription information to the DDDB and associates the new transcription information with its corresponding hearing information.

2.6.2 Software Stack

The transcription tool is a Java based web application, deployed on an Apache Tomcat servlet on an AWS server. On the back end, the application utilizes a Java based MVC framework called Stripes, a MySQL database connected to an object-relational mapping layer called ORMlite[3], and JSP[2] for server side dynamically generated HTML. The front end of the application is largely implemented with the use of Ractive.js[6], a template driven Javascript library for dynamic DOM manipulation.
2.6.3 Transcription Editing UI

This section will describe elements of the basic transcription editing UI. The figure below depicts the editing UI of the original tool.

![Editing UI of the version 1 transcription tool](image)

Figure 2.6: Editing UI of the version 1 transcription tool
Task Information

At the head of the transcription editing UI is five pieces of information about that particular task: title, bill, video, hearing date, and assigned date.

Figure 2.7: Editing UI header information

<table>
<thead>
<tr>
<th>Title</th>
<th>Bill</th>
<th>Video</th>
<th>Hearing Date</th>
<th>Assigned</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task 1 of ASSEM Education Committee</td>
<td>Bill: 201520160AB1</td>
<td>Video: #1 0:00:00 - 0:20:00</td>
<td>Hearing Date: 10/22</td>
<td>Assigned: 10/12</td>
</tr>
</tbody>
</table>

**Title**  
The task title includes: the task number (which indicates that task’s place in the series of tasks for a single hearing), the hearing committee, the date on which the hearing took place, and the Youtube ID of the video that corresponds with the transcript.

**Bill**  
An optional value that describes the bill being discussed during this task. If there is no bill being discussed then the value is left blank. The bill discussion code printed here indicates the term, bill classification, and bill number of the currently discussed bill.

**Video**  
Printed here is the number of the video in the series of videos that coincides with the hearing and the time interval of the video being transcribed.

**Hearing Date**  
This is the date that the hearing took place.

**Assigned**  
This is the date the task was assigned to the transcriber.
Transcription Editing Window

Under the tab labeled “Quick Edit” is the main transcription window. In this window is the list of texts and attributes of each individual utterance from the portion of the video being processed. Each utterance row contains several pieces of information: the text of the utterance, a time interval, the speaker, the discussion type, an alignment, and two editing buttons, merge and clone.

Figure 2.8: Utterance editing UI elements

<table>
<thead>
<tr>
<th>Time</th>
<th>Interval</th>
<th>This time interval indicates when during the task video the coinciding utterance was spoken.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Speaker</td>
<td>Speaker</td>
<td>This drop down includes the list currently imported speakers and is used to indicate the person that spoke this utterance.</td>
</tr>
<tr>
<td>Type</td>
<td>Type</td>
<td>This drop down is used to indicate the type of discussion being had while this utterance was given. Every utterance must be one of three discussion types: <em>Committee Discussion</em>, Authors Presentation, or Testimony.</td>
</tr>
<tr>
<td>Text</td>
<td>Text</td>
<td>This text box is used to input the transcription of the video during the indicated time interval.</td>
</tr>
</tbody>
</table>

25
Alignment This drop down is used to indicate the alignment of the speaker of the utterance towards the subject of the current hearing. Six alignments may be chosen: Indeterminate, For, For if amended, Neutral, Oppose unless amended, Oppose.

Merge This button is used to merge two adjacent utterances into one. When clicked, this button appends the text of the current utterance into the text of the utterance above and the end time of the current utterance replaces the end time of the utterance above.

Clone This button is used to create or split utterances. When clicked, a clone of the current utterance is inserted into the list of utterances below the current utterance.

Speaker Importation

Under the tab labeled “Import Speaker” is the window used to add new speakers to the current hearing. Any speakers imported into a hearing and assigned to an utterance are considered to have participated in that hearing. In this window new speakers may be created or known speakers may be searched for importation. If a speaker is searched, the search results are listed in the “Search Results” box. When searching for a speaker, the rows of information returned contain a unique speaker ID, the speakers name, and a button to add that speaker to the hearing. The “Current Orators” box contains all of the speakers currently imported. Other tasks from the same hearing will share imported orators and will be displayed in this box. Additionally, speakers imported into the current hearing can be removed using the remove button to the right of a speakers name.
Video Player

Located below the utterance edit window is an embedded Youtube player, used to display the video that coincides with this particular task.

Speaker Profile Tab

Under the “Speaker” tab at the bottom right of the window a speaker’s “Profile” is inputted. This area is used to input information about a speaker’s responsibilities and affiliations. When a speaker is given a classification of Lobbyist or General Public, an accompanying drop down appears where all known organizations may be search and selected. If the searched organization is not found, that
organization may be created and added to the repository of known organizations.

Task Tab

The “Task” tab contains the “Complete Task” button. The “Complete Task” button is used to flag the current utterances being edited as “finalized”. After a task is finalized, during the next nightly website update, the new finalized utterances are uploaded to the database of the Drupal instance that supports DigitalDemocracy.org.

Other Features

While the video being transcribed is playing, the utterance with the time interval corresponding to the location in the video is highlighted. As the video plays, the utterance in focus is continually updated to help the transcriber keep their place among the utterances. Additionally, hitting the back-tick key toggles the video between pause and play.

2.6.4 File Formats

Two file formats are used by the transcription system to manage complete transcription data. TTML is the main file format used internally by the system for manipulating and storing transcriptions. SRT is another file format used by the system to import raw transcription data to store diarized speaker intervals. The diarization system is explained later in this document.
TTML

TTML (Timed Text Markup Language) is the transcription system’s main file format for managing complete transcription data. The TTML format associates text with time intervals by enclosing the text with “p” tags that contain beginning and end time attributes. TTML is convenient for managing transcription data because additional attributes can be added to the “p” tags, allowing the initiative to annotate individual utterances with any number of qualities. The annotations used for utterances in the basic version of the transcription tool include: a start and end time, a speaker id, an affiliation, and a discussion type. An example section of a TTML file for a raw transcription is shown below.

```xml
<body region="CaptionArea">
  <div>
    <p begin="00:01:47.210" end="00:01:53.680">This is an example</p>
    <p begin="00:01:53.680" end="00:01:56.640">of three utterances</p>
    <p begin="00:01:56.640" end="00:02:01.210">in a ttml</p>
  </div>
</body>
```

SRT

SRT (SubRip Text) format is defined by having groups of plain text lines separated by blank lines. This format is popularly used to capture subtitle data with groups of three lines for each subtitle: the first line being a subtitle id, the second line containing a time interval, and the third line containing the subtitle text. SRT files are used in two particular parts of the transcription system: transcription importation and in speaker diarization. Transcriptions are some-
times encapsulated as SRT, as such the transcription system accepts SRT files and converts them to TTML for internal use. In speaker diarization (explained later in this document), SRT is used to store the speaking intervals of hearing participants. Example SRT data is shown below.

1
00:00:13,778 --> 00:00:18,770
This is an example

2
00:00:19,870 --> 00:00:22,830
of three utterances

3
00:00:24,220 --> 00:00:28,690
in an srt file
2.6.5 Architecture

This section will describe the usage and internal structure of the transcription tool. Components and processes described here will include: the database tables unique to the tool, the implementation of the hearing creation process, the usage of the hearing creation interface, the implementation of the transcription annotation and correction process, and the usage of the transcription editing interface.

Database Tables

Two tables in DDDB are dedicated to the transcription tool:

**TT_Editor** This table contains the user name and password of each user of the transcription tool and their status as an admin or editor.

**TT_Task** This table contains information about transcription tasks. Each task entry stores which hearing it comes from, which bill is discussed, which editor is assigned to this task, the title, the associated video, where in the video the task starts and ends, and time stamps about when the task was created, assigned, and completed.
Hearing Creation

To create a Hearing an administrator of the transcription tool must first access the log in screen and validate their username and password. Upon logging in, the hearing creation page immediately begins to load. When loading the hearing creation page, all of the committees and bill types are passed along in the HTML as JSON objects. The figure below displays what the hearing creation page looks like with no information loaded.

![Unfilled hearing creation UI](image)

Figure 2.10: Unfilled hearing creation UI
At the top of the hearing creation page the date of a hearing is inputted. Below the date selector in the “Committees” section, the one or more committees holding the hearing are selected. The figure below is an example of two selected committees.

![Figure 2.11: Hearing committee selection UI](image)

In the “Videos” section, the videos that comprise the hearing are inputted as Youtube IDs along with their duration and raw transcript files. The transcript files inputted are required to be formatted as either TTMLs or SRTs. Below is a figure showing a single video being inputted.

![Figure 2.12: Hearing video importation UI](image)
Below the videos section, bill discussions may be optionally entered. All bill discussions are listed on the LegInfo website and are saved to the DDDB ahead of hearing importation. To associate a hearing with a bill discussion, the correct discussion ID must be found by entering the bill type and number into the bill search field. The associated bill discussions are then retrieved from the database and used to populate a bill discussion drop down selector. With a bill discussion chosen, the time interval of the video where the bill is being discussed must be entered. An example of a single bill discussion being associated with a single video is shown in the figure below.

![Figure 2.13: Bill discussion importation UI](image_url)

The hearing creation is finalized by hitting the “Create” button at the bottom of the page. During hearing creation, the transcripts go through a transformation process. If the inputted transcript is an SRT file, it is first converted to TTML. After conversion, the TTML in it’s entirety is saved as a single record in the DDDB. The TTML is then split into individual utterance lines and each is saved to the Utterance table along with: the video it appears in, its start and end times, and the bill discussion it’s associated with. For each utterance in the TTML two versions are saved to the database: one is kept permanently in its raw form and the other is used for editing by the transcription tool.
Finally, after hearing information is saved to the database the transcription tasks for the hearing are created. Two types of boundaries determine where a task will start and end: the start and end times of a video, or the start and end times of a bill discussion. For example, if there are no bill discussions in a hearing, each video in that hearing will have its own task. If there is a single video for a hearing and a short bill discussion in the middle of the video, three tasks will be created: one task from the beginning of the video to the start of the bill discussion, another task for the entirety of the bill discussion, and the last task would start at the end of the bill discussion and go till the end of the video. If there are two videos, and one bill discussion starts in the middle of the first video and ends in the middle of the second video, 4 tasks will be created: two for the first video split by the beginning of the bill discussion, and two for the last video split by the end of the bill discussion.

**Transcription Editing**

To edit a transcript, a user must first have the designation of editor and be assigned a task. Upon logging in as an editor, the tasks assigned to that user will be loaded in the HTML of the page as JSON objects. The assigned tasks appear as a list containing task information and buttons to open them for editing, any of which may be chosen. When opening a task for editing, information about the task and a majority of the data needed during transcription is loaded with the HTML of the page. The data loaded with a transcription task includes: all of the utterances from the task video that fall between the task start and end time, all of the people that have been associated to an utterance in the task, the known roles of each person in the database, all organizations, and all committees. From the transcription window, changes to the utterances are cached and sent to
DDDB in batches every 30 seconds.

Editing and completing a transcription task has several requirements and conventions. Utterance lines should be merged to contain a single coherent thought. Under this convention, utterances are typically between one and five sentences long. All utterances must have a correct speaker, time, alignment, and discussion type. Proper nouns should be capitalized, and bill numbers and dates should be written numerically. Upon completing a transcription task with the complete task button, all of the utterances being edited have a finalized flag in the database updated to true.

2.7 Bottleneck

There is a significant bottleneck in the data procurement system of the initiative. The bottleneck is caused by the amount of manual labor needed to correct and annotate the raw transcriptions produced by the ASR system. Using a basic version of the transcription tool, a trained transcriber can process a minute of raw transcription produced by an ASR system in around 6 minutes. An untrained transcriber, using a basic transcription tool, will take slightly longer than a trained transcriber at around 7 minutes. During the height of legislative sessions, the number of hearings happening daily makes producing timely transcriptions for those hearings untenable with the number of regularly employed transcribers. Paying for more employees, or for outside services to provide pre-corrected transcripts would increase transcription turn over, but is prohibitively expensive in the long term or at scale.

A major goal of the Digital Democracy platform is to be a source of current legislative hearing information and to be scalable to 50 states at all levels of gov-
ernment. To attain this goal, the bottleneck will be alleviated by increasing raw transcription quality ahead of manual editing, and by increasing the productivity of the transcribers.

2.7.1 Process Improvement

After the tool’s first full legislative session in use, enough observations were taken from editors to begin automating common editor actions and automatically correcting common transcription errors. Several projects were started with the goal of either increasing raw transcription quality or editor productivity. Two projects, transcription text correction and speaker diarization, focused on increasing the quality of automatically generated transcripts. A third UI improvement project focused on devising UI functionality that reduced or automated redundant and rule based editor behavior.

Diarization

The purpose of the diarization improvements is to merge adjacent utterances that belong to the same speaker and are too short to stand alone. Diarization also allows the tool to associate large groups of utterances across a transcription with a single speaker. If the speaker of a single utterance in this group of utterances could be identified, it would mean that the entire group of utterances could have their speaker immediately labeled.

Text Correction

The text correction improvements included two distinct processing techniques for raw transcriptions.
The first technique is a dictionary based automatic capitalization of many proper nouns. The MAVIS ASR system does not capitalize common proper nouns, so by creating a dictionary of proper nouns many words could be automatically capitalized ahead of editing. Additionally, as unusual proper nouns are found during transcription, they can be added to the dictionary of proper nouns for future reference.

The second technique uses a text classifier and grammar to identify and convert lexical representations of bill numbers to their numerical representation. The MAVIS ASR system produces transcriptions with lexical representation of numbers spoken, however this is a sub optimal format when searching for Bills by number. Bill references could be searched more easily if they used their numerical representations. By training machine learning classifiers to identify bill numbers they can be located and parsed, and with the help of a grammar the lexical representations of these numbers are converted to numerical.

**UI Changes**

With the input of editors and an editor supervisor a number of common editing behaviors were identified that could either be automated or enhanced.

Predictable or repetitive behaviors included: When assigning a speaker to an utterance the speakers most recent alignment or discussion type would likely be assigned, legislators initially participate in committee discussion and non-legislators in testimony discussion, many adjacent utterances are tagged with the same speakers, alignments, or discussion types, cloned utterances must always be trimmed to split a single utterance, hearing videos are continuously rewound short distances while editing an utterance, utterance text areas are commonly
scrolled or enlarged during editing, committee members are commonly imported
during their respective committee hearings, and non-legislators rarely change
their alignments during a hearing.

Tedious tasks included: Searching through un-alphabetized speaker search
results, imported speaker lists, or speaker drop downs, discriminating between
speakers with similar names by checking the digital democracy website, and veri-
ifying that all transcription requirements had been met before completing a tran-
scription.
CHAPTER 3

CONTRIBUTIONS

The main contribution of this thesis is the implementation and evaluation of the three improvement technologies and methodologies provided by the projects described in section 2.7.1. To implement these improvements, this thesis incorporated their contributions into the transcription tool such that they could be tested in distinct incremental stages and would constitute a production ready tool. To evaluate the three improvements, metrics were created and an experiment was carried out to gather data about the various improvements to the tool. Data for each metric was gathered from four distinct tool editing environments allowing each improvement to be evaluated according to the amount it contributes to differences in each metric. In other words, because each successive editing environment in the experiment incorporates a new improvement, data was gathered that characterizes the tool with and without each improvement.

In addition to testing and incorporating these transcription improvements, a total transcription cost model was created. This model includes the costs of initial transcription editing and the costs of correcting any remaining mistakes. The changes to total transcription cost resulting from the three improvements are then estimated according to changes in both the initial editing process and
the subsequent mistake correction process.

3.1 Tool Modifications

Three main modifications were necessary to incrementally test the improvement projects. First, a pipeline was devised for automatic transcription processing. This pipeline allows a transcription tool administrator to selectively activate different processing steps and allows for future processing steps to be easily added. Second, to test new UI functionalities, two separate interfaces were implemented on top of the new system. One interface provided all of the same functionalities of the original tool, and a second interface incorporated all of the new UI improvements. Third, to gather data about the manual editing process, a system was created that saves telemetry data about editor activity in the transcription tool. This telemetry system was implemented alongside both the original and improved tool interfaces.

3.2 Improvement Experiment

Several metrics were created to describe the effects of the improvements on transcription. These metrics were task duration, the percentage of correct speaker assignments, the percentage of correct length utterances, and the percentage of correctly transcribed words. Data for these metrics were gathered in an experiment designed to reflect average transcription editing experiences. 15 editors placed into four groups were directed to edit four average transcription tasks, one task in each editing environment. To average the effects of editor experience and the effects of a particular improvement having an advantage in a particular task, each group edited a different task in each of the four tools in a round robin
configuration. Improvement effects were calculated by running regressions on the data gathered for each metric, using a dummy variables for each tool to estimate how each tool changed the given metric values.

3.3 Transcription Cost Model

A model describing the total costs associated with correcting a transcription was created and will be defined here. The model for total cost consists of the costs associated with an initial editing process and a subsequent mistake correction process.

\[
TotalCost = EditingCost + MistakeCorrectionCost
\]

The cost of the initial editing process is a function of the hourly price of editing labor and the total editing time.

\[
EditingCost = L_e T_e
\]

The cost of the mistake correction process is a function of the hourly price of mistake correction labor and the time spent correcting mistakes. In the cost equation, labor cost per hour is represented by L, the time to correct a mistake represented by T, and the number of mistakes to correct represented by N.

\[
MistakeCorrectionCost = L_m (N_o T_o + N_w T_w + N_u T_u + N_s T_s + N_{ks} T_{ks} + N_{us} T_{us})
\]

There are six aspects of mistake correction and they are listed here in the order
they appear inside the parentheses in the mistake correction equation above: The number of times the transcription tool is re-opened to perform correction and the time to open the tool, the number of incorrectly spelled words and the time to correct individual words, the number of incorrect length utterances and the time to correct utterance length, the number of incorrectly assigned speakers and the time to reassign speakers, the number of known speakers that were not imported and the time to import known speakers, the number of unknown speakers that were not imported and the time to identify and import unknown speakers.

This model is used to estimate the effect of the improvement projects on total cost. The time to complete initial transcription editing was measured during the improvement experiment. To determine the amount of time to correct word, utterance, and speaker assignment mistakes, the experience of editor managers was assessed. The time to import known and unknown speakers was estimated from the improvement experiment telemetry data and from a speaker identification experiment.
CHAPTER 4

TOOL IMPROVEMENTS

This chapter will describe in detail the three tool improvements projects. Each section of this chapter will be dedicated to one of the three improvements, describing the technologies, implementation, and reasoning behind each.

4.1 UI Improvements

Over the course of the first two legislative sessions transcribed by the initiative, observations were made about repetitive, predictable, or tedious actions being performed by transcribers using the original transcription tool. These actions were translated into rules or enhanced functionality that were then incorporated into the interface.
Committee Member Importation

The hearings being transcribed are conducted by committees with known members. Because committee members are likely to be present at a respective committee hearing, the committee member importation button, depicted in figure 4.1, fetches the committee members and adds them to the list of imported speakers. Additionally, a commonly used placeholder value used to represent the “Committee Secretary”, a person that calls role at the beginning of hearings, is added to the list of hearing speakers.

Figure 4.1: Committee member importation button location
Automatic Profile Classification

Sourcing information about occupations from the web, profile information for many speakers can be partially initialized on import.

Sticky Attributes

Alignment and Discussion Type attributes would often stay the same for a particular speaker over many utterances. To avoid having to repetitively assign a discussion type and alignment whenever a speaker is selected for an utterance, discussion type and alignment are automatically assigned to an utterance based on the most recent discussion type and alignment for the assigned speaker.

Initialized Discussion Type

Only Legislators can participate in the discussion type of “Committee Discussion” or “Author Presentation”. All other speaker classifications can only be given the discussion type of “Testimony”. Using these rules, all Legislators are initialized with a discussion type of “Committee Discussion”, and all other classifications start with “Testimony”.

Quick Rewind Hotkey

When listening to the video and correcting the transcription simultaneously, transcribers must rewind using the Youtube player seek bar if they fall behind. Rewinding in this way can be tedious and inaccurate. A 5 second rewind was added by pressing the tilde key to avoid rewinding using the mouse when continuously catching up to the video position.
Cascading

To avoid having to assign the same utterance attributes for a long series of utterances spoken by the same person a “Cascade” option was implemented. By checking the “Cascade” check box, any attribute assignment of speaker, alignment, or discussion type will be similarly applied to the following nine utterances.

![Cascade check box location](image)

**Figure 4.2:** Cascade check box location
Task-wide Alignment

Non-legislators will rarely change their alignment for the entirety of a hearing. Having a task wide alignment selector for non-legislators simplifies the act of assigning or changing alignment values for all of that speaker’s utterances. This task wide alignment selector was placed in the profile assignment area under the “Speaker” tab.

Figure 4.3: Task-wide alignment selector location
**Utterance Split**

To split an utterance into two using the basic tool, that utterance was cloned and the resulting two utterance texts were manually trimmed. To avoid having to do any trimming, a split functionality has replaced cloning. By clicking the split button of an utterance, two utterances are created from the text on the left and right sides of the cursor in the text box.

**Figure 4.4: An utterance before a split**

**Figure 4.5: The resulting two utterances after a split**
Conditional Completion

To prevent tasks from being finalized in an incomplete state, transcription completion is now conditioned on a number of requirements being met. If these conditions are not met, the complete task button is hidden and a message indicating details about the unmet requirement is displayed in its place. These requirements include:

- Imported speakers must be given a profile
- Lobbyist speakers must be given an affiliation
- All utterances must be given a speaker and an alignment

Profile Website Links

When importing a speaker with a generic name a few problems in ambiguity arise. Either multiple people with the same name would appear in the search results, or if there was only a single instance of a name, transcribers may not be sure if that name refers to another person by the same name. To help partially alleviate this issue, hyperlinks to speakers online profiles in the Digital Democracy portal were added to speaker names as a way to help transcribers resolve ambiguity.
Dropdown Organization

The speaker selection drop down contains all of the speakers imported to the current hearing. Because of this, finding the correct speaker for a particular utterance can be time consuming. To improve the ability of a transcriber to quickly find the correct speaker for an utterance, the speaker drop down has been alphabetized and segregated by classification.

![Figure 4.6: A speaker selection drop down segregated by classification](image)

Alphabetized Search Results and Orators

The list of search results and current orators was shown to potentially grow very large. Because of the list size, transcribers spend a significant amount of time searching those lists for a particular speaker. These lists were alphabetized by last then first name to help shorten this search time.
Editing Window Size Increased

The size of the utterance text area was originally set to a height of two rows. To avoid transcribers having to scroll or re-size the text area box, the text area height was increased from two rows to four.

![Figure 4.7: Utterance text box size in the original tool](image)

![Figure 4.8: Expanded utterance text boxes](image)
4.2 Diarization

This section will explain the meaning of speaker diarization, the implementation of the diarization system, and how the output of system was used for transcription improvement.

4.2.1 Terminology

Speaker diarization refers to the task of determining who spoke when, or more specifically, the intervals during which individual speakers are speaking. To do this, a diarization system must determine when features of an audio signal change enough to decide that a new speaker has begun speaking. Speaker diarization is often used as an upstream processing technique for other forms of speech processing such as automatic speech recognition or information retrieval. Often, speaker diarization is desired when analyzing audio recordings in which there are more than a single speaker such as in movies, radio, conferences, meetings, debate, news broadcasts, etc. However, these different contexts require overcoming different challenges such as overlapping speakers, multiple microphone channels, or noise in the audio signal. Speaker diarization is effective in the context of legislative committee hearings because, although the audio quality is sub-optimal, there is little noise, little speaker overlap, and the audio channel quality is consistent.

4.2.2 System

To perform diarization, the initiative uses an open source toolkit named LIUM Speaker Diarization[4]. When processing an audio signal, this toolkit creates a database of signal feature clusters that it considers to belong to the same person.
Each cluster of audio signals in the database is labeled with a number, each number indicating a potentially different person. By doing this, intervals of time in a test audio signal can be identified that contiguously correspond to a single entry in the database. In other words, these intervals indicate the boundary points where one speaker changed to another.

The input for the diarization system is hearing videos. The system routinely checks the Digital Democracy Youtube channel for new video uploads, and upon finding a new video, downloads that video and extracts its audio in the form of a WAV file. The WAV file is then passed to the diarization toolkit, a database of audio feature clusters is created, and a summary of contiguous speaker intervals is produced in the form of an SRT file. The SRT file is then labeled by its Youtube ID and saved to a directory on the transcription tool server.

The diarization system will download a new video from Youtube and finish it’s processing in under 30 minutes, while uploading and performing ASR on a single video will take a similar amount of time. Because the ASR service and the diarization system run in parallel and both complete in similar amounts of time, the addition of diarization does not delay the transcription system under usual conditions. When a new hearing is entered into the transcription tool, the hearings Youtube ID is used to find its corresponding diarization SRT file on the transcription server. The diarization data from the SRT file is then used in the TTML pipeline during the hearing’s creation.

The diarization step of the TTML pipeline transforms an utterance in two ways: speaker tagging and utterance merging. The result of passing a transcription through these processing steps is a consolidated transcription that has been labeled with anonymous speaker tags. These speaker tags are then later used by the transcription tool to assign speakers to utterances.
Speaker Tagging

The diarization interval data produced by the diarization system is used by a Python script to label transcription utterances with “Speaker tags”. Speaker tags are the number values that the diarization system uses to represent the audio features of a single speaker. Additionally, while assigning speaker tags, utterances are given merge flags. These merge flags can later be used by a merging script to consolidate utterances.

Utterance Merging

In this processing step, a Python script is used to consolidate utterances in the transcription TTML by it’s merge flags. If a merge flag is 0, the utterance before it did not belong to the same speaker. If a merge flag is 1, the utterance before it was spoken by the same speaker, and thus that utterance is a candidate for merging with the utterance before it. If a merge flag is 2, it means that the diarization intervals indicated that two different speakers contributed to that utterance, and thus it cannot be merged. As the merge script moves through the list of utterances, it consolidates groups of three or less adjacent utterances that have merge flags of 0 or 1. Three utterances are merged at a time because the resulting consolidation is typically within appropriate utterance length conventions. For example, if 5 utterances were respectively labeled by a 0 followed by four 1 flags, then those utterances would be consolidated into two utterances.
4.2.3 Diarization UI

Normally a speaker is assigned on an utterance by utterance basis, but because the diarization process adds speaker tags to utterances, speakers can then be assigned by speaker tag. In the transcription editing UI a small UI element was added to facilitate assigning speakers by speaker tag. This UI is located directly above the list of utterances, and features two drop downs and a “Set” button.

![Figure 4.9: Diarization UI](image)

The first time a diarized transcription is edited, speaker tags will appear in the speaker selection box as a default selection. When an utterance with a speaker
tag has its speaker identified, instead of assigning the speaker to the utterance, the diarization interface can be used to assign speakers by speaker tag. In the diarization interface, the right drop down includes every imported speaker and the left drop down includes every speaker tag available in the current transcription. By selecting the tag, the corresponding speaker, and clicking set, every utterance with that speaker tag gets assigned the indicated speaker.

4.3 Text Correction

Two common errors in the computer generated raw transcriptions were identified and programs were written to correct them. These two types of errors are: bill numbers are given lexical representations, and proper nouns are not capitalized. To correct these errors, a transcription correction processing step was written that uses the Python Natural Language Toolkit (NLTK), a grammar, and a dictionary based look up and replacement.

Bill Number Correction

In a MAVIS produced raw transcription, numbers spoken are always given a lexical representation. To assist in research and searching bill numbers from the web platform, they must be converted from their lexical representations to their numerical representations. To identify bill numbers a machine learning technique is used. Through a process called feature identification, patterns were identified that commonly indicate that a bill number is being referenced. Using these features and many examples of numbers that are and are not references to bill numbers, a classifier was trained that helped find bill number references in transcriptions. Several classifiers provided by NLTK were trained in this way.
and the most reliable was selected. Once a bill number is identified, converting its lexical representation to its numerical representation is achieved by parsing the number and using a grammar to construct the number.

**Proper Noun Capitalization**

To capitalize proper nouns a basic dictionary look-up is used. A dictionary of proper nouns was created from an online repository, and if any word in a transcription appeared in that dictionary, it was capitalized. Additionally, a script and an interface was created for an administrator of the transcription tool to add entities to the dictionary.
CHAPTER 5

SYSTEM DESIGN

This chapter will list the modifications made to the transcription tool in order to incorporate and test the three tool improvements, the tool modifications will then be explained individually, and lastly, this chapter will define the four editing environments that were tested in the improvement experiment.

5.1 Baseline System Modifications

Seven modifications were made to the baseline tool to facilitate the testing of the transcription system improvements. These modifications occurred in roughly the order delineated below:

1. A TTML processing pipeline was added to the basic transcription tool

2. The UI improvements were added to the tool

3. Text correction scripts were added to the TTML pipeline

4. Diarization scripts supported by a diarization system were added to the TTML pipeline
5. UI elements to facilitate speaker assignment to diarized transcriptions were added to the tool UI

6. A system to capture editor telemetry was built into the tool UI

7. The tool was branched and it’s UI was reverted to the state of the original tool

The branched tool would be used with unprocessed transcriptions and would serve as the baseline tool without any improvements. The main tool would then be used with: unprocessed transcriptions to test UI improvement, text corrected transcriptions to test the text correction improvements, and finally with diarized and text corrected transcriptions to test the diarization improvements. Furthermore, the main tool would act as the production implementation of the tool for the initiative.
5.2 Transcription Pipeline

5.2.1 Overview

The purpose of the TTML pipeline is to simplify the automatic pre-processing of TTML transcription files uploaded to the transcription tool. To facilitate the incremental testing of individual processing steps, and to accommodate future transcription sources, each processing step of the pipeline can be selectively applied on a hearing by hearing basis. The pipeline consists of three components:
1. Programs

2. A TTML adapter

3. An execution list

5.2.2 Programs

Each processing step of the TTML pipeline is facilitated by an executable program or script. Each program in the pipeline is required to take a transcription TTML file as an argument, and must write its processed transcription to stdout formatted as a TTML. Each processing step is internally maintained as enumeration of corresponding executable names, execution paths, arguments, and processing flags. Further more, the order of the programs in the enumeration dictates the order of execution.

When a transcription is being processed by the pipeline, it begins as a raw transcription that gets saved to the transcription tool server as 0.ttml. That first TTML file, 0.ttml, is then processed by the TTML adapter and passed as an argument to the first processing step in the pipeline. After finishing, the processing program writes its resulting transcription to stdout which then gets saved as a “.ttml” file labeled by the step in the pipeline. The pipeline continues this way until each processing step that is eligible to run has finished.

5.2.3 TTML Adapter

The TTML adapter is an object that facilitates saving TTML data to the DDDB. When a TTML file is saved to the DDDB, it is saved in its entirety as a single entry and as multiple utterance entries, one for each utterance in the
TTML. Additionally, the resulting transcriptions generated at each step of the pipeline are saved in the DDDB with a corresponding data flags, indicating the processing step that generated that version of the TTML.

5.2.4 Execution List

An execution list determines which processing steps in the pipeline can run. The execution list is created from two other lists of data: A list of processing steps that are requested to run, and a list of processing steps that can run. Before any program can begin processing a TTML along the pipeline it must be present in the execution list, otherwise that processing step is skipped. Lastly, the execution list identifies the last processing step of the pipeline and communicates to the TTML adapter to save two copies of the final TTML data. One copy serves a record of the final processing step while the second copy is considered the most current state of the transcription and is used for editing. The most current transcription data is then retrieved by the transcription tool when edited by a transcriber.

5.3 Improvement Implementation

After the implementation of the TTML pipeline, the two improvements that modified the transcription TTML file, text correction and diarization, could be incorporated into the tool. These improvements were incorporated as described in the previous section, by adding scripts to the TTML pipeline that manipulated the transcription TTML in ways dictated by the auxiliary improvement systems. Additionally, the UI improvements and diarization UI elements delineated in the previous chapter were implemented.
5.4 Editor Telemetry

In order to gather accurate information about the amount of time spent editing a task or performing particular editing activities, time data about every action taken by a transcriber in the editing UI was recorded. This was accomplished by creating a logging system and adding event listeners to every interactive element of the editing UI. In addition to editing actions, a time stamp was recorded every time an editing window was opened. Upon receiving an event, an event listener would contribute a time stamp and relevant event data to a local cache of event data, which would be continuously communicated to the server and cleared. Upon receipt, a server side application controller would append all event data to a log file labeled by task id.

5.5 Baseline Tool Branch

In order to test the performance of the three improvements against the original tool, the UI of the enhanced tool was reverted to include only the functionalities of the original tool. When testing the baseline tool, editors would work on transcriptions that had not been modified by either the text correction or diarization systems.
5.6 Tool Versions

This section will briefly summarize the improvements associated with each of four transcription environments.

5.6.1 Tool 1 - Baseline

This tool provides no improvements. It acts as a baseline for the other improvements, replicating the editing environment of the minimal viable implementation of the transcription tool.

5.6.2 Tool 2 - UI Improvements

This tool provides a series of UI improvements over the baseline tool. Those improvements are as follows:

- Committee member importation
- Informed speaker profile initialization
- Utterance attribute selection by a speakers last associated attribute
- Profile based utterance discussion assignment
- Utterance attribute cascading
- Task wide speaker alignment selection
- Utterance splitting replaces cloning
- Conditional Completion of transcription tasks
For Disambiguation, speaker names are links to speaker pages on the Digital Democracy platform

- Drop down organization

- Utterance default text size increased

- Hearing video quick rewind

- Alphabetized search results and orator lists

### 5.6.3 Tool 3 - Text Correction

This tool includes both the UI improvements of tool 2 and the text correction improvements. Unlike the first or second tools, editors using the third tool environment work on transcriptions that have many proper nouns automatically capitalized and bill numbers converted to their numerical representations.

### 5.6.4 Tool 4 - Diarization

This tool includes the UI improvements of tool 2 and the text correction of tool 3. Unlike the first, second, or third tools, editors using the fourth tool are presented with transcriptions that have had significant utterance merging automatically performed. Additionally, editors are giving a UI functionality that allows for task wide utterance speaker assignment.
EXPERIMENT DESIGN

The objective of this experimentation is to determine the effects of the three different tool improvements on transcription quality and manual labor needed to produce a transcription. To test these three improvements to the transcription system, four distinct editing environments were built. Starting with a basic tool and a raw transcription, each successive editing environment would feature an additional improvement. Any significant variation in data gathered from one environment to the next would be attributed to the incrementally added improvement to that environment.

6.1 Testing Strategy

To test the effects of these different editing environments properly, several variables needed to be controlled by averaging their effects. Tasks were selected with average speaking durations and speaker counts. Four tasks were selected to prevent editors from seeing the same task twice. To minimize the effect of an improvement performing differently in different tasks, all four tasks would be transcribed in each editing environment by the four editing groups in a round
robin arrangement. Participating editors were randomly placed into the four editing groups to minimize the effect of their differing abilities.

In addition to the four rounds used to test the four states of the transcription tool, a 5th introductory calibration round was added. The calibration round gives inexperienced editors practice with the transcription tool ahead of the first testing round, and it helped establish performance benchmarks among the transcribers. The calibration round was conducted with all of the editors transcribing the same video using the basic transcription tool.

6.2 Input Selection

6.2.1 Editors

There were two classes of editors in this experiment: Experienced transcribers and inexperienced transcribers. The experienced participants all had previous experience using the transcription tool. The inexperienced participants had no experience using the transcription tool but were members of the www.digitaldemocracy.org QA team and were familiar with transcriptions. Of the 15 transcribers to participate in the experiment, 6 of them were experienced and 9 were inexperienced. As compensation for their work, the participants were paid inline with their hourly wages from the institute. Each transcriber was given an ID number between 1 and 15, which would be used the in URL of their dedicated tools and as their login credentials. A chart of editor experience and group assignment is shown below.
<table>
<thead>
<tr>
<th>ID #</th>
<th>Group</th>
<th>Experienced</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3</td>
<td>-</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>-</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>X</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>X</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>-</td>
</tr>
<tr>
<td>6</td>
<td>4</td>
<td>-</td>
</tr>
<tr>
<td>7</td>
<td>3</td>
<td>-</td>
</tr>
<tr>
<td>8</td>
<td>2</td>
<td>-</td>
</tr>
<tr>
<td>9</td>
<td>4</td>
<td>X</td>
</tr>
<tr>
<td>10</td>
<td>2</td>
<td>-</td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>-</td>
</tr>
<tr>
<td>12</td>
<td>1</td>
<td>X</td>
</tr>
<tr>
<td>13</td>
<td>2</td>
<td>X</td>
</tr>
<tr>
<td>14</td>
<td>1</td>
<td>-</td>
</tr>
<tr>
<td>15</td>
<td>2</td>
<td>X</td>
</tr>
</tbody>
</table>

Table 6.1: Table of editors

6.2.2 Videos

The videos transcribed for this experiment were selected from previously transcribed videos in the DDDB. From the DDDB, SQL queries were used to select the videos with the most average speaking duration and average quantity of speakers. The range of videos considered were one standard deviation from the average speaking duration. From this range of videos, the average and standard deviation of speakers in a video were found. The videos selected were the videos
nearest to the average speaking duration that had a speaker count within one standard deviation of the average number of speakers.

**Average Speaking Duration**

Many videos have periods of time at the beginning and end where no one is speaking. To account for this, a video's duration is considered the length of time from the first utterance spoken to the last. With this SQL query, the average video duration of videos from completed transcriptions was found. The result of running this query on the production database returned an average speaking duration of 757.7 seconds.

```sql
select avg(time) from
    (select (max(U.endTime) - min(U.time)) time
     from TT_Task as T,
     (select * from Utterance where current=1) U
     where T.vid = U.vid
     and U.time >= T.startTime
     and U.endTime <= T.endTime
     and completed is not NULL
     group by tid) times;
757.70
```
Relevant Range of Video Length

The standard deviation of speaking duration was found because only speaking durations within one standard deviation of the mean would be considered when finding the average speaking count. The following SQL query was run on the production database and returned the average of the mean error squared. The square root of that value, the standard deviation of video speaking duration, was calculated to be 677.125 seconds. With that value, the range of video durations to be searched was between 80 seconds, and 1434 seconds.

```sql
select avg((time-757.70)*(time-757.70)) from
(select (max(U.endTime)-min(U.time)) time
from TT_Task as T,
(select * from Utterance where current=1) U
where T.vid = U.vid
and U.time >= T.startTime
and U.endTime <= T.endTime
and completed is not NULL
group by tid) duration;

458498.44
sqrt(458498.44);
677.125
Video Length Range = [80, 1434]
```
Average Speaker Count per Task

Using the range of videos found in the previous query, the following SQL query was written to return the average number of speakers in a video. Running this query on DDDB returned an average speaker count of 8.15 speakers.

```sql
select avg(counts.count) from
    (select count(distinct pid) as count,
    (max(U.endTime) - min(U.time)) as time
    from TT_Task as T,
    (select * from Utterance where current = 1) U
    where T.vid = U.vid
    and U.time >= T.startTime
    and U.endTime <= T.endTime
    and completed is not NULL
    group by tid) counts
where time <= 1434
and time >= 80;

8.15
```
Relevant Range of Speaker Count

The valid range used for the number of speakers in a video was one standard deviation from the speaker count mean. The only videos considered when determining this range were videos in the range found in a query above. Using an average speaker count of the 8.15 speakers and a video range between 80 second and 1434 seconds, the standard deviation of speakers was found to be 8.58 speakers. Because the number of speakers cannot be less than 1, the valid range for the number speakers in a video was found to be between 1 and 16 speakers.

\[
\text{Speaker Quantity Range} = [1, 16]
\]
Video Selection

Using the speaker count range and speaking duration range found in the queries above, the videos used for testing were found. The videos selected were the 5 videos closest to the average speaking duration of 757.7 seconds. The range of the selected videos was ultimately between 756 seconds and 758 seconds. Within this range, five videos were found that had speaker counts between 1 and 16. The following query returned exactly 5 videos when run on DDDB.

```
select * from
  (select count(distinct pid) as count, taskid
   from Utterance as U,
   (select *
    from TT.Task as T,
    (select (max(U.endTime)−min(U.time)) time,
    T.tid as taskid
    from TT.Task as T,
    (select * from Utterance where current=1) U
    where T.vid = U.vid
    and U.time >= T.startTime
    and U.endTime <= T.endTime
    and completed is not NULL
    group by tid) dur
    where dur.time > 755
    and dur.time < 759
    and T.tid = taskid
    and completed is not NULL) Tasks
   where U.vid = Tasks.vid
   and U.time >= Tasks.startTime
   and U.endTime <= Tasks.endTime
   group by tid) counts
where counts.count < 16
and counts.count > 1;
```
Tasks

The videos selected correspond to the follow tasks in the DDDB. Each video was within one second of the average speaking duration of 757 seconds. Additionally, each video had speaker counts that were evenly distributed in the range of one standard deviation of the average.

<table>
<thead>
<tr>
<th>#</th>
<th>Committee Name</th>
<th>Date</th>
<th>Sp.</th>
<th>Dur.</th>
<th>Start</th>
<th>End</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ASSEM Transportation</td>
<td>03/23/2015</td>
<td>10</td>
<td>757</td>
<td>813</td>
<td>1579</td>
</tr>
<tr>
<td>2</td>
<td>ASSEM Education</td>
<td>04/08/2015</td>
<td>9</td>
<td>756</td>
<td>270</td>
<td>1033</td>
</tr>
<tr>
<td>3</td>
<td>SEN Budget and Fiscal Review</td>
<td>05/22/2015</td>
<td>6</td>
<td>757</td>
<td>1393</td>
<td>2152</td>
</tr>
<tr>
<td>4</td>
<td>JOINT Budget Conference</td>
<td>06/03/2015</td>
<td>5</td>
<td>758</td>
<td>0</td>
<td>810</td>
</tr>
<tr>
<td>5</td>
<td>ASSEM Health and Developmental Services</td>
<td>07/09/2015</td>
<td>8</td>
<td>756</td>
<td>0</td>
<td>780</td>
</tr>
</tbody>
</table>

Table 6.2: Table of selected hearing videos

6.3 System Setup

6.3.1 Tools

For each participant two tools were deployed. The first tool would retain the basic UI while the second tool would include the enhanced UI. Both of these tools would be built on top of the new system and were both compatible with the same database schema. Each transcriber would use the basic tool for a single test round and a calibration round, both on a raw transcript. Editors would use the tool with the enhanced UI for three rounds: one round with a raw transcript, the second round with a text corrected transcript, and a third round with a text
corrected and diarized transcript. A total of 30 tools were deployed, an A and B tool for each participant.

6.3.2 Databases

A database for each of the 15 participants was required to isolate them from the organizations and people created by other editors. To simplify the creation of these databases, a single parent database was established from a copy of the DDDB. To setup the parent database for cloning, two alterations were required. First, all production records of utterances, tasks, and hearings were removed, while any entities needed for transcription were kept. Second, to simplify the parent database cloning process and keep task IDs unique for each transcriber, 16 copies of each task was created in the parent database, one for each potential editor of that task. For each testing task there would be: 8 copies of raw transcriptions, 4 copies of text corrected transcriptions, and 4 copies of diarized transcriptions. Lastly, there would be 15 tasks created for the calibration round. In total there were 79 tasks entered into the parent database. From this parent database 15 databases were cloned, one for each pair of tools to be used by each participant.
6.4 Procedure

This section will describe procedural aspects of the transcription experiment.

6.4.1 Group Rotation

To minimize the effect of enhancements inherently performing differently on different tasks, four tasks would be edited with each tool by four groups of editors. To minimize the effect of editor ability on transcription performance, all of the editors were randomly assigned to the four editing groups. The round robin task rotation between the groups for each tool version is shown below.

<table>
<thead>
<tr>
<th></th>
<th>Calib.</th>
<th>Round 1</th>
<th>Round 2</th>
<th>Round 3</th>
<th>Round 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Group 1</td>
<td>Task 5</td>
<td>Task 1</td>
<td>Task 2</td>
<td>Task 3</td>
<td>Task 4</td>
</tr>
<tr>
<td>Group 2</td>
<td>Task 5</td>
<td>Task 2</td>
<td>Task 3</td>
<td>Task 4</td>
<td>Task 1</td>
</tr>
<tr>
<td>Group 3</td>
<td>Task 5</td>
<td>Task 3</td>
<td>Task 4</td>
<td>Task 1</td>
<td>Task 2</td>
</tr>
<tr>
<td>Group 4</td>
<td>Task 5</td>
<td>task 4</td>
<td>Task 1</td>
<td>Task 2</td>
<td>Task 3</td>
</tr>
</tbody>
</table>

Table 6.3: Table of group task rotation

6.4.2 Scheduling

Each editor had five transcriptions to perform, one calibration task and four test tasks. The calibration round was conducted with all of the participants in a single session. In this round all of the editors performed task 5 on the basic version of the transcription tool. The rest of the tasks were scheduled with each editor individually over the course of three weeks. Each participant used the tools in the same order, starting with the basic tool in the calibration round and
round 1, the enhanced UI in round 2, the enhanced UI with text correction in round 3, and finally the enhanced UI with text correction and speaker diarization in round 4.

6.4.3 Environment

The location of the tests was a Mac computer lab on the Cal Poly campus. Every editing session was conducted in this lab, each editor using a modern Mac computer with a 27 inch monitor. The environment was not private or reserved, but was quiet and virtually free from disruption. Furthermore, each session was proctored to ensure editing was performed in a consistent way and any editing difficulties were quickly resolved.

6.4.4 Instructions

During the calibration session, all of the editors were briefly instructed on how to use the transcription tool and the requirements to properly complete a transcription. The editors were also given a sheet indicating transcription requirements at the beginning of each round. In addition to the sheet of requirements, when an editor began a new round they were given a sheet of instructions on how to use any new enhancements. The requirements sheet and instruction sheets are provided in the appendix.

6.4.5 Gathering Data

Data was collected in two ways: transcription edits were collected in the databases, and editor telemetry was collected in log files on the application server. To collect data from the databases, a Python script was used to extract final
utterance information and place that information into files labeled by task IDs. For each task, two files of utterance data were created: a file containing a list of utterance texts, and a file containing a list of utterance texts and attributes. The data analysis was then performed using the editor telemetry, utterance texts, and utterance attributes.
CHAPTER 7

DATA ANALYSIS

The goal of the transcription tool improvements was to increase the efficiency of transcribers. Transcriber efficiency is a function of transcription quality and transcription speed. Transcription speed can be measured as the ratio of transcription duration to video duration. Transcription quality can be measured using these three dimensions: text alignment, speaker assignment, and utterance length. Text alignment is a measure of how closely a test transcription matches its corresponding, previously transcribed, control transcription. Speaker assignment is a measure of how many utterances were labeled with the correct speaker. Utterance length is a measure of what percentage of utterances are cut to a length considered optimal for the digital democracy platform.

Utilizing utterance information and editor telemetry gathered during the transcription experiment, several scripts were written to produce data related to the editor efficiency metrics. Measurements for each metric were drawn from this extracted information, and the effect of each tool on the four dimensions of transcription efficiency was calculated. The tool effects we evaluated using both a time series plot of changes in averages, and a multiple linear regression for each metric.
7.1 Metrics

This section will define the four efficiency metrics and describe how each was measured.

7.1.1 Task Duration

Task duration is a measure of the time it takes an editor to complete a transcription task. To normalize transcription editing time for different videos, task duration is measured as the ratio of editing time over the interval of active speaking in a video. The active speaking duration in a video is considered the amount of time between the first utterance spoken and the last. To measure how long an editor spent editing a task, a script was written to scan editor telemetry data and accumulate the time during active editing intervals. An active editing interval is considered the time between when a task was first opened for editing and the last editor interaction with the tool before it is closed or the task completed. The total editing time is the summation of the active editing intervals for that task. The ratio of total editing time over active speaking interval is used as the metric for task duration.
7.1.2 Text Alignment

Text alignment is a percentage of how closely a test transcription matches the corresponding control transcription from the production database. There are three components of how closely the words of one transcription align with another: exact word matches, partial word matches, or word mismatches. The metric used to measure text alignment is how many words from a test transcription are exact matches to the words in the control transcription. To produce these totals, the Needleman–Wunsch global alignment and comparison algorithm is used.

Needleman–Wunsch Algorithm

The Needleman–Wunsch algorithm is an application of dynamic programming commonly used in bioinformatics to align and compare protein sequences. In this research the Needleman–Wunsch algorithm is used to determine the similarity between lists of words, namely the test and control transcriptions. To perform a comparison of two transcriptions, a grid is constructed with the words of each transcription on one of the two axes of the grid, each word occupying a single row or column. The algorithm proceeds align these two transcriptions by procedurally inserting spaces, or considering aligned words matches, partial matches, or mismatches. Each of these four procedures are given a cost and the objective of the algorithm is to align the transcriptions in a way that minimizes the overall summation of these costs. Starting from the upper left and moving towards the bottom right, the squares of the alignment grid are given values according to summation of the total cheapest procedures that would need to occur if the words on each axis were aligned. A match state has a cost of 0 because it is the optimal state. A match state can only be applied if the words on each axis
of the grid have words that match exactly. A partial match state has a cost of 1, and can only be applied if the words on each axis match after being stripped of non-alpha numeric characters and converted to lowercase. A mismatch comes with a penalty of 5, and can be applied if the transcriptions are aligned despite the current words being mismatched. The value in a square that has a partial match, match, or mismatch is the sum of the penalty and the square to the upper left diagonal. A space insertion comes with a penalty of 3, and can be applied if a squares corresponding words do not match and shifting one of the transcriptions over by a word would result in a cheaper alignment. The resulting cost of an insertion is determined as the minimum sum of the insertion penalty and the cost of the square to the left or above.

With the grid of alignment penalties constructed, the cheapest path through the grid is traced from the bottom right to the upper left. The optimal alignment between the two transcriptions is created by sequentially applying the procedures of each square along the traced path. From this optimal alignment, the number of mismatches, partial matches, and matches are totaled, and the percentage of matched words to total words is calculated. The percentage of matched words is then used as the alignment metric for the test transcription.

7.1.3 Speaker Assignment

The speaker assignment metric is a percentage of utterances in a transcription with correctly identified speakers. When importing a speaker to a task, if the correct speaker cannot be found in the database it may be created. The ability to create speakers creates the problem of there being duplicate speakers in the database. This problem is alleviated through an administrator tool to merge duplicate speakers into one. As such, an utterance is considered to have a correct
speaker if that speaker has the correct name. Statistics for utterance speakers were gathered manually for every test task in the experiment.

7.1.4 Utterance Length

The utterance length metric is a percentage of utterances in a transcription that are cut to an appropriate length. An utterance that is too long is considered by the institute any utterance of 385 characters or more. An utterance that is too short is considered an utterance of less than seven words that sits adjacent to another utterance from the same speaker. Scripts were used to count the utterances with incorrect lengths and the ratio of incorrect to total utterances is used as the utterance length metric.
7.2 Analysis

Using a system of scripts, measurements for each metric were generated and organized according to task. As an initial method of comparison, the four tools were first measured by the their effect on individual metrics. The effects of each tool were evaluated in two ways: multiple regressions on the individual metrics, and observing the metric averages change across the four tools.

7.2.1 Outlier

The z-scores of the data sets were calculated to certify that the data collected was representative of average transcription efforts. An initial inspection of the standardized data sets revealed one editor with an abnormal transcription effort. While many editors had some peculiar performances, the one editor in question performed abnormally throughout the entire experiment. This editor completed all of their tasks extraordinarily slow, taking twice the time of other inexperienced editors to compete all of their tasks. Because the efforts of this transcriber are not representative of average transcription work, they have been excluded from the data set.
7.2.2 Averages

A rough evaluation of the improvement effects can be done by looking at percent change in metrics averages. Figures 7.1 and 7.2 depicted below are charts of the change in average editing duration and transcription quality across the four tools. The most notable change being task duration clearly declining with the addition of each improvement. There was some change in the metric averages for speaker assignment and utterance length, and nearly zero change in word alignment.

![Average Duration](image1)

**Figure 7.1:** Edition duration average across the four tools

![Metrics by Tool](image2)

**Figure 7.2:** Metric averages across the four tools
7.2.3 Regression

To more accurately determine the effect that each improvement had on transcription editing, a multiple regression was run on the four tools against the data sets for each metric. The regressions constituted the four tools as independent dummy variables and the metric data as the dependent variables.

<table>
<thead>
<tr>
<th>Metrics</th>
<th>Baseline</th>
<th>Tool 2</th>
<th>Tool 3</th>
<th>Tool 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Duration</td>
<td>Coefficient 5758.15(0.0)</td>
<td>-485.63s</td>
<td>-878.49</td>
<td>-972.61</td>
</tr>
<tr>
<td></td>
<td>P-Value 0.0000</td>
<td>0.0933</td>
<td>0.0032</td>
<td>0.0012</td>
</tr>
<tr>
<td>Utterances</td>
<td>Coefficient 0.6915</td>
<td>-0.0090</td>
<td>0.0059</td>
<td>0.0346</td>
</tr>
<tr>
<td></td>
<td>P-Value 0.0000</td>
<td>0.8667</td>
<td>0.9122</td>
<td>0.5193</td>
</tr>
<tr>
<td>Speakers</td>
<td>Coefficient 0.8687</td>
<td>-0.0531</td>
<td>0.0318</td>
<td>0.0443</td>
</tr>
<tr>
<td></td>
<td>P-Value 0.0000</td>
<td>0.4741</td>
<td>0.6677</td>
<td>0.5498</td>
</tr>
<tr>
<td>Words</td>
<td>Coefficient 0.8204</td>
<td>-0.0001</td>
<td>0.0006</td>
<td>0.0009</td>
</tr>
<tr>
<td></td>
<td>P-Value 0.0000</td>
<td>0.9971</td>
<td>0.9649</td>
<td>0.9517</td>
</tr>
</tbody>
</table>

Table 7.1: Table of regression results

Baseline: Tool 1

In the chart above, the baseline column indicates the intercept value of the given regression. This intercept can be considered the baseline or average value for a particular metric gathered from the use of tool 1. The baseline metric values from tool 1 include: 5758.15 second task duration, 69.15% of utterances have appropriate lengths, 86.87% of speakers are correctly assigned to utterances, and 82.04% percent of words perfectly match a control transcription.
Tool 2

The second tool included many UI improvements over the baseline tool. In the metric regressions the second tool did not provide any changes that are considered statistically significant. At a p-value of 0.0933, the most significant coefficient for this tool was for duration, suggesting a decrease in task duration of 485.63 seconds. The coefficients and p-values of utterance length and word matching indicated virtually no relationship and no change in these metrics. The speaker matching metric suggests a 5.31% percent decline in utterances with correct speakers, but with low significance.

Tool 3

The third tool provided all of the improvements of tool 2 but also included automatic text correction. This tool provided the first statistically significant improvement, being a decrease of 878.49 seconds in task duration. The duration metric for tool 3 came with a p-value of 0.0032, indicating a strong relationship between a decrease in task duration and the use of tool 3 over the baseline. The other three metrics showed very little to zero relationship between a change in the metrics and using tool 3 over the baseline.

Tool 4

Tool 4 provided all of the improvements of tool 2 and tool 3 but also performed speaker diarization on the transcript. The speaker diarization improvement provided utterances that have been automatically consolidated to more appropriate lengths, and a UI that allowed the assignment of speakers to many utterances at the same time. The regression statistics for tool 4 indicated a highly signifi-
cant decrease of 972.61 seconds in transcription time over the baseline tool. The change in the metric for utterance length was statistically insignificant with a p-value of 0.5193, but indicated an improvement of 3.5%. Similarly, the change in the metric for speaker assignment was insignificant with a p-value of 0.5498, but indicated an improvement of 4.43%. The statistics for word matching indicated virtually zero improvement in percent of matched words.

7.2.4 Result Summary

The only significant coefficients found in these regressions were for changes in task duration in editing environments 3 and 4. The coefficients for these two rounds were -878.49 seconds and -972.61 seconds respectively. From a baseline editing duration of 5758.15, a decrease of 972.61 seconds represents a 16.89% increase in editing speed. From original video duration’s of 757 seconds, this decrease also represents a speed up in the ratio of editing time to video time from 7:37 to 6:19.

7.3 Total Cost Evaluation

In this section the total cost model defined in chapter 3 will be used to evaluate the overall change in transcription cost. To apply the model, several constants must be defined including the time to correct the different types of mistakes, the cost of labor, and the number of times a transcription task is reopened. Once the model constants have been defined, the model equation can be applied to the transcription data to calculate the total costs of each transcription.
7.3.1 Constant Estimation

The time to correct utterance, speaker assignment, and word spelling mistakes was approximated with the help of an editor manager. Additionally, the time to open the transcription tool and locate an error was approximated by an editor manager. The average time to import known speakers was taken from the improvement experiment telemetry data. The improvement experiment did not feature any speakers that were unknown so a second experiment was run to simulate identification and importation of unknown speakers. Five experienced transcribers were asked to perform a short transcription task featuring three speakers that were removed from the database. Using the telemetry data gathered from these tasks, the time these editors spent identifying and importing the unknown speakers was estimated. For the purpose of this evaluation, it will be assumed that the transcription tool must only be accessed a single time during mistake correction. Additionally, the cost of editing and mistake correction labor will be defined as $10.00. A table of these constants is shown below.
<table>
<thead>
<tr>
<th>Constant</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task Access and Error Location Time</td>
<td>300s</td>
</tr>
<tr>
<td>Number of Mistake Correction Sessions</td>
<td>1</td>
</tr>
<tr>
<td>Word Correction Time</td>
<td>10s</td>
</tr>
<tr>
<td>Utterance Correction Time</td>
<td>20s</td>
</tr>
<tr>
<td>Speaker Correction Time</td>
<td>10s</td>
</tr>
<tr>
<td>Known Speaker Importation Time</td>
<td>36s</td>
</tr>
<tr>
<td>Unknown Speaker Importation Time</td>
<td>152s</td>
</tr>
<tr>
<td>Editing Labor Cost</td>
<td>$10.00/hr</td>
</tr>
<tr>
<td>Mistake Correction Labor Cost</td>
<td>$10.00/hr</td>
</tr>
</tbody>
</table>

Table 7.2: Transcription cost model approximated constants

7.3.2 Total Cost Results

Using the values defined in the previous section as constants, the transcription cost equation was applied to the data extracted from the transcription experiment. Because changing task duration was the only statistically significant regression coefficient found in section 7.3.3, all other transcription metrics were held constant at their baseline values. The cost of each transcription is evaluated under the assumption that all errors in each transcription would be corrected. The average transcription cost from each editing environment is shown in the table below.
Table 7.3: Average transcription cost by experiment round

<table>
<thead>
<tr>
<th>Round</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cost</td>
<td>$25.43</td>
<td>$24.08</td>
<td>$22.99</td>
<td>$22.72</td>
</tr>
</tbody>
</table>

The decrease in total transcription cost is entirely the result of the decrease in editing duration. This is because the mistake correction process is unaffected by a change in editing duration without a change in transcription quality. Likewise, a decrease in cost of $2.71 is the same for both the total transcription cost and editing labor cost. For total transcription cost, a decrease of $2.71 represents a reduction of 10.66%.

7.4 Threats to Validity

The analysis featured in this chapter is only valid if the changes in the data are due to the improvements in the tools. If the changes in the data are due to other influences that are not accounted for, those influences then pose a threat to the validity of the analysis. The main threat to the validity of this research is the variable difficulty and qualities of the transcription tasks. This issue was addressed by having each task transcribed with each editing environment. A second threat to the validity of this research is that nine of the volunteers did not have previous experience with the transcription tool and their improvement may be due to continually having more practice with the tool. This concern was addressed by having all of the editors practice transcription during a preliminary calibration round. Additionally, experienced editors increased in editing effectiveness on average, similar to the increase of inexperienced editors.
CHAPTER 8

CONCLUSION AND FUTURE WORK

The main finding of this research was that a combination of UI improvements, automatic text correction, and speaker diarization increases the effectiveness of human assisted speech transcription and annotation. In this research, transcription efficiency was measured as the combination of transcription quality and transcription speed. Implementing the three improvement technologies was shown through a linear regression to decrease transcription editing time by about 1:18 minutes per minute of video, from 7:37 minutes per minute of video to 6:19 minutes per minute of video. Additionally, there are indications utterance length and correct speaker assignment may have increased in quality with speaker diarization, however linear regressions show a very low statistical significance in these dimensions. Furthermore, the three improvements resulted in virtually no change in transcription text quality.

From these results it can be said that there was no significant increase or decrease in transcription quality but a highly significant improvement in transcription speed. This result suggests that transcription editors, regardless of the three improvements, will consistently correct transcriptions to a certain standard of text quality. However, while text quality may remain the same, these three im-
provements have either increased editing efficiency, or have decreased the amount of editing required to complete a transcription task. Consequently, transcriber effectiveness has been increased, and editor labor costs have been reduced about 16.89% as a result of UI improvements, proper noun capitalization, bill number conversion, and lastly, annotation and utterance improvement through speaker diarization.

The cost reductions provided by the three implemented improvements are likely not sufficient to allow the Digital Democracy project to significantly scale, however they have shown cost reduction progress on behalf of the earliest improvement projects of the initiative. Other improvement projects planned for the future are described in the next section.

8.1 Future Work

The future work of this research includes measuring additional transcription qualities, and implementing and evaluating future improvement projects.

8.1.1 Additional Quality Metrics

There were three types of quality metrics that were overlooked during this research, those metrics include: utterance alignment, utterance discussion type, and speaker profile. Because this thesis overlooked a metric for utterance alignment, the effects of adding a task completion condition for alignment was not accounted for. It is likely that this alignment completion condition increased task duration, and the other improvements may have been responsible for a greater speed up than apparent in the data. Utterance discussion type and speaker profile assignment metrics would provide a more thorough analysis of effect of improvements
8.1.2 Improvement Projects

The transcription improvements implemented for this thesis were among several projects planned for the future. There are three projects currently being developed that should have a significant impact on the effectiveness of transcribers including: facial identification, voice identification, and creating raw transcriptions using an in-house open source ASR system.

Face and Voice Identification

While the face and voice recognition projects are being developed separately, they both contribute to an effort to automatically associate speakers with utterances. By developing a repository of vocal features and images of hearing participants, these two systems may be able to accurately identify the speaker of an utterance. The vocal and facial identification systems will both be able to suggest a set of probable speaker identities. From the sets of probable speakers, a consensus about the true speaker will be reached that should be more accurate than either of the two systems individually.

Facial identification requires two systems. The first part of the effort focuses on developing a system to capture images of hearing participants and labeling the speakers featured in those images. The second part involves using that repository of images to suggest the identities of hearing participants and the speakers of individual utterances.

The voice identification system will use the transcriptions created by the initiative for training. This voice identification system may be trained in several
ways: on a hearing by hearing basis according to the results of face identifications and committee membership, or it could be continually trained across larger spans of time. With a highly trained voice identification system, the speakers of individual utterances could be accurately identified. The results of this identification will then be included into the TTML pipeline to further enhance the quality of a transcription before editing.

**Open Source ASR Service**

The initiative currently uses a generic speaker independent ASR service to generate raw transcriptions from hearing videos. An in-house speaker adapted ASR system could be trained to recognize the speech of common hearing participants to potentially improve the quality of raw transcriptions. Using the transcriptions and speaker tagging already generated by the initiative, a very large repository of training data is available to create a highly tuned ASR system. Additionally, any transcriptions created in the future could be used to further tune the system.
CHAPTER 9

RELATED WORK

This thesis combined an evaluation of improvements to a human assisted transcription system utilizing a multiple linear regression and a cost model. Improvements to purely automatic lecture transcription has been explored in several other projects such as: a PHD dissertation named “Robust Automatic Transcription of Lectures” [16], a technical report named “Improving Automatic Speech Recognition for Lectures through Transformation-based Rules Learned from Minimal Data” [14], and for an audio lecture transcription project at MIT introduced in a white paper named “Automatic Processing of Audio Lectures for Information Retrieval: Vocabulary Selection and Language Modeling” [8]. More similar to this thesis, research in the performance of different protocols in human assisted lecture transcription was performed in a masters thesis named “Assisted Transcription of Video Lectures” by Juan Daniel Valor Miro [13]. Finally, multiple linear regression and deriving cost models is common in academia. Examples and case studies of these topics are featured in a text named “Introduction to Applied Econometrics” [15].
9.1 Assisted Transcription of Video Lectures

The research of this work was contributed by Juan Daniel Valor Miro, of the Polytechnic University of Valencia, as a Masters thesis in artificial intelligence, pattern recognition, and digital images. The main research of this thesis was determining the superior protocol for a transcription system named “Translectures”. The protocols tested included: a human performed transcription, very light human assistance on the output of an ASR system, or a two part protocol using moderate human assistance on the output of an ASR system. The human assisted protocol for lecture transcription in this thesis features an ASR system and manual transcription UI similar to the transcription system employed by the Digital Democracy initiative. However, the improvements evaluated in this research centered around improving the quality of ASR output through training of a speaker adapted acoustic model. Additionally, the metrics used to measure the effectiveness of the improvements were labor time, and the text accuracy of the ASR output.

The main conclusion of this research was that using a two phase protocol, the use of a primary speaker adapted ASR system and a secondary human assisted transcription process facilitated by an editing UI, is the optimal protocol for the transcription of a library of academic video lectures. This research also described a methodology of improving a massively speaker adapted ASR system, and the degree of the subsequent improvement.
BIBLIOGRAPHY


Online; accessed 20 January 2016.

Online; accessed 20 January 2016.


A Task Duration Graphs

This appendix contains various graphs of task duration of transcriptions across the experiment. Each graph shows how task duration changes going between each of the four experiment rounds.
This graph shows trend lines of average task duration. It overlaps lines that show average duration for each task (shown as dashed lines) and for each group (shown as solid lines) in each round. This graph helps to elucidate reasons for irregular results, stemming from variable task difficulty being paired with groups of varying editing ability.

Figure 9.1: Editing duration by task and group
The following four graphs contain the editor performances in each of the four groups. Each editor from each of the four groups is represented by four bars, indicating the editors task duration in each round.

Figure 9.2: Editing durations of group 1 editors

Figure 9.3: Editing durations of group 2 editors
Figure 9.4: Editing durations of group 3 editors

Figure 9.5: Editing durations of group 4 editors
B Volunteer Consent Form

This appendix includes the consent form signed by each editor that participated in the transcription experiment.
INFORMED CONSENT TO PARTICIPATE IN A RESEARCH PROJECT

Introduction
A research project on the economics of manual speech transcription is being conducted by graduate student Justin Rovin in the Department of Computer Science at Cal Poly, San Luis Obispo, under the supervision of Dr. Alex Dehktyar. The purpose of the study is to determine effects on labor costs of several enhancements to manual speech transcription.

Activity
You are being asked to take part in this study by transcribing 5 separate intervals of a legislative committee meeting, using a transcription tool web application. Each interval will be approximately 13 minutes long and your participation will take approximately 6 hours total, spread across 5 separate sessions. Please be aware that you are not required to participate in this research and you may discontinue your participation at any time without penalty.

Risks
There are no risks anticipated with your participation in this study. Your participation in this project will have no effect on your status as a cal poly student, or as an employee of the IATPP. Participation in this project is not a function of your employment with the IATPP.

Confidentiality
You will be assigned an ID number and your name will not be used in any publication or disclosed at any time.

Potential Benefits
Transcription tool enhancements that are shown to decrease transcription time or increase transcription accuracy, will likely inform the next round of enhancements for the transcription tool. Significant decreases in transcription time will assist in the ability of the Institute of Advanced Technology and Public Policy to expand their legislative transparency initiative to more states and more levels of government. As an incentive for you participation, you will receive compensation in line with your hourly wage agreement with the IATPP.

Contacts
If you have questions regarding this study or would like to be informed of the results when the study is completed, please feel free to contact Justin Rovin by phone at (949) 246-8766 or through email at JustinRovin@gmail.com. If you have concerns regarding the manner in which the study is conducted, you may contact the Chair of the Human Subjects Committee: Dr. Steve Davis at (805) 756-2754, sdcalis@calpoly.edu, or Dr. Michael Black (after January 1, 2016) at (805) 756-2894, mblack@calpoly.edu; or the Dean of Research, Dr. Dean Wendt at (805) 756-1508, dwendt@calpoly.edu. If you agree to voluntarily participate in this research project as described, please indicate your agreement by signing below. Please keep one copy of this form for your reference, and thank you for your participation in this research.

__________________________________________  ____________________
Signature of Volunteer                      Date

__________________________________________  ____________________
Signature of Researcher                     Date
C Volunteer Instructions

This appendix includes the instruction guides given to each editor at the beginning of each round in the transcription experiment. The instruction sheets include the requirements to correctly complete transcription work and descriptions of the improvements added in each round.
Version 1 Participant Guide

List of Basic Features

- Pause the video with "" (back-tick, above tab)
- Utterance text editing
- Utterance speaker selection
- Utterance speaker alignment
- Utterance start and end time
- Utterance discussion type selection
- Utterance merge button
  - Will merge that utterance text with the utterance above
  - Will set the end time of that utterance to the end time of the utterance above
- Utterance clone button
  - Will make an exact copy of that utterance and insert it below
- Search and Create speakers
- Assigning speaker profiles

Requirements

- All Utterances must have a:
  - speaker, (Import or Create, use Unidentified Speaker if needed)
  - alignment
  - discussion type
- All imported speakers must have a profile
- All Lobbyist profiles must have an affiliation
- Utterances should be grouped into approximately 2 - 4 sentences if possible
Version 2 Improvement Guide

New Features

- Utterance split button replaces clone
- 5 second quick rewind with “~” (tilde)
- Quick importation of committee members
- Quick importation of speakers that have previously discussed current bill
- Links to main speaker profiles on website
- Cascading selections
- Automatic importation of the committee speaker.
- Alphabetic listing of search results
- Automatic job classification from static profiles
- Conditional task completion: All imported speakers must have a profile
- Conditional task completion: All utterance speakers need an alignment
- Automatic loading of initial discussion type
- Speakers get "sticky" attributes for discussion type and alignment
- Task wide alignment selection per speaker
- Speaker drop downs alphabetized
- Speaker drop downs segregated by classification
- Editing window size increased
Version 3 Improvement Guide

New Features

- Many pronouns will be capitalized.
- Bill numbers will be automatically converted from their lexical representations to their numeric representations. i.e. Bill AB five hundred => Bill AB 500
Version 4 Improvement Guide

New Features

- Speaker "Tagging". Many Utterances have speaker boxes that will initially be filled with a tag. The tag has an S followed by a number. Utterances with the same tag likely were spoken by the same speaker.
- Tag wide speaker selection
- Automatic Merging

Other Considerations

- While the automatic merging should improve the initial utterance text length, you must still actively split or merge utterances to their recommend lengths if they are not initially set to be that way.
- While the speaker tagging is thought to be quite accurate, it cannot be completely relied upon. You must be careful to make sure every word from every utterance is actually spoken by the speaker that is implied by the tag.
- It is advisable to fill out the discussion type and alignment of a speaker before using the automatic tag feature, as the discussion type and alignment will then get applied everywhere the speaker is applied.
- With the above shortcut in mind, you must be careful that the discussion type and alignment is correct in each utterance.