STRATEGIC SELECTION OF TRAINING DATA FOR DOMAIN-SPECIFIC SPEECH RECOGNITION

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

Daniel Girerd

June 2018

COMMITTEE MEMBERSHIP

TITLE:	Strategic Selection of Training Data for Domain-Specific Speech Recognition
AUTHOR:	Daniel Girerd
DATE SUBMITTED:	June 2018
COMMITTEE CHAIR:	Foaad Khosmood, Ph.D. Associate Professor of Computer Science
COMMITTEE MEMBER:	Alex Dekhtyar, Ph.D. Professor of Computer Science
COMMITTEE MEMBER:	Franz Kurfess, Ph.D. Professor of Computer Science

ABSTRACT

Strategic Selection of Training Data for Domain-Specific Speech Recognition

Daniel Girerd

Speech recognition is now a key topic in computer science with the proliferation of voice-activated assistants, and voice-enabled devices. Many companies offer a speech recognition service for developers to use to enable smart devices and services. These speech-to-text systems, however, have significant room for improvement, especially in domain specific speech. IBM's Watson speech-to-text service attempts to support domain specific uses by allowing users to upload their own training data for making custom models that augment Watson's general model. This requires deciding a strategy for picking the training model. This thesis experiments with different training choices for custom language models that augment Watson's speech to text service. The results show that using recent utterances is the best choice of training data in our use case of Digital Democracy. We are able to improve speech recognition accuracy by 2.3% percent over the control with no custom model. However, choosing training utterances most specific to the use case is better when large enough volumes of such training data is available.

ACKNOWLEDGMENTS

Thanks to:

- Dr. Foaad Khosmood for direction, guidance, and serving as my adviser.
- Committee members Dr. Dekhtyar and Dr. Kurfess for their feedback and suggestions.
- Digital Democracy [8] for providing the data used for this thesis.
- Cal Poly Github for the LaTeX template [1].

TABLE OF CONTENTS

Page

LIST OF TABLES ix

LIST OF FIGURES x

CHAPTER

1 Introduction 1

1.1 Importance of Speech Recognition 1

1.2 Speech Recognition Difficulty 1

1.2.1 Easier Tasks 2

1.2.2 More Difficult Tasks 2

1.2.3 Domain-specific Systems 2

1.3 Contribution 3

2 Related Work 4

2.1 Judicial Domain 4

2.2 Transfer Learning 4

2.3 Machine Translation 5

3 Background 7

3.1 Speech Recognition 7

3.1.1 Conversational Telephone Speech 7

3.1.2 IBM Speech Recognition 8

3.2 Legislative Domain 9

3.2.1 California Legislature 10

3.3 Digital Democracy 10

3.4 Tools 11

3.4.1 Watson 11

3.4.2 HTTP REST API 12

3.4.3 FFmpeg 12

3.4.4 NIST Sclite 12

4 Experiment Process 13

4.1 Selecting Hypotheses 13

4.1.1 Available Information 13

4.1.2 Recent Hypothesis . . . 13
4.1.3 Specific Hypothesis . . . 14
4.1.4 Random Hypothesis . . . 14
4.1.5 Training Data Sizes . . . 14
4.2 Choosing the Test Set . . . 15
4.2.1 Choosing the Committee . . . 16
4.2.2 Gathering the Test Set . . . 17
4.3 Gathering the Training Data . . . 18
4.3.1 Random Training Data . . . 19
4.3.2 Recent Training Data . . . 19
4.3.3 Specific Training Data . . . 19
4.3.4 Training Data Size . . . 19
4.4 Using Watson . . . 20
4.4.1 Creating the Models . . . 20
4.4.2 Filling the Models . . . 20
4.4.3 Training the Models . . . 21
4.4.4 Transcribing with the Models . . . 21
4.5 Evaluation by Word Error Rate . . . 21
4.5.1 Sclite and Trn Format . . . 22
4.5.2 Creating the Reference Transcript . . . 22
4.5.3 Creating the Hypothesis Transcripts . . . 23
4.6 Running Sclite . . . 24
5 Results . . . 25
5.1 Characterizing our Files . . . 25
5.1.1 Utterance Length Averages . . . 25
5.1.2 Parts of Speech . . . 25
5.1.3 Unique Words . . . 26
5.2 Out of Vocab Words . . . 27
5.3 Sclite Results . . . 27
5.4 Hypothesis Scores . . . 28
5.4.1 Scores by File . . . 30
5.5 Statistical Significance . . . 31

6 Conclusion . . . 33
6.1 Future Work . . . 34
BIBLIOGRAPHY . . . 35
APPENDICES
A All Results . . . 38
A.1 Sclite Command . . . 38
A.2 Result Tables . . . 39
A.3 No_Model Results . . . 40
A.4 Random_1 Results . . . 40
A.5 Random_10 Results . . . 41
A.6 Recent_1 Results . . . 42
A.7 Recent_10 Results . . . 43
A.8 Specific_1 Results . . . 43
A.9 Specific_10 Results . . . 44
B Working with Digital Democracy Data . . . 46
B.1 Database Tables Used . . . 46
B.2 Queries for Committee Information . . . 46
B.3 Query for Test Set File Ids . . . 46
B.4 Converting the Test Set . . . 47
B.5 Queries for Training Data . . . 47
B.5.1 Recent Hypothesis Training Data . . . 47
B.5.2 Random Hypothesis Training Data . . . 48
B.5.3 Specific Hypothesis Training Data . . . 48
B.6 Querying for the Reference Transcript . . . 48
C Watson cURL Commands . . . 50
C.1 Command to Create a Model . . . 50
C.2 Command to Fill a Model . . . 50
C.3 Command to Train a Model . . . 50
C.4 Command to Transcribe Using a Model . . . 51

LIST OF TABLES

Table		Page
3.1	Speech Recognition Word Error Rates for Conversational Telephone Speech	8
4.1	Hypotheses and Models	15
4.2	Top California Committees by Hours Recorded	17
4.3	Files in Test Set	18
4.4	Training Data Sizes	20
5.1	Average Number of Words per Utterance for each File	25
5.2	Part of Speech Frequencies per File	26
5.3	Unique Words per File	26
5.4	Out of Vocab Word Counts per Model	27
5.5	Scores on Individual Files for Each Model	31
5.6	Word Errors per Model	32
A.1	No_Model Percentage Results	40
A.2	No_Model Raw Results	40
A.3	Random_1 Percentage Results	41
A.4	Random_1 Raw Results	41
A.5	Random_10 Percentage Results	41
A.6	Random_10 Raw Results	42
A.7	Recent_1 Percentage Results	42
A.8	Recent_1 Raw Results	42
A.9	Recent_10 Percentage Results	43
A.10	Recent_10 Raw Results	43
A.11	Specific_1 Percentage Results	44
A.12	Specific_1 Raw Results	44
A.13	Specific_10 Percentage Results	44
A.14	Specific_1 Raw Results	45

LIST OF FIGURES

Figure		Page
2.1	Options for domain adaption in neural machine translation [4]. . . .	6
A.1	The results from Sclite for file af comparing the 10m recent hypothesis against the Digital Democracy reference transcription.	39

Chapter 1

INTRODUCTION

1.1 Importance of Speech Recognition

Speech recognition is gaining increased focus as voice-activated assistants and devices become commonplace. It has become a highly-competitive market with many companies offer a speech recognition service, especially for developers to use to enable smart devices and services. Highly accurate speech recognition is essential for voice interfaces.

Speech recognition is important for captioning. While mainstream TV and movies have captions, most video content does not. Lack of captions means content is inaccessible to the deaf or use in no-sound conditions. With speech recognition able to automatically transcribe most talking, it will make this content accessible.

Speech recognition has two clear use cases for many companies, even if they don't manufacture speech related devices or offer such services. First, it's used in call centers to generate transcripts and enable some automation. Second, it can transcribe meetings and other conversations to reduce the need to take notes and enable better record-keeping and collaboration.

1.2 Speech Recognition Difficulty

Speech recognition is a difficult, but valuable, task to automate. Researchers have been working on automated speech recognition for decades. Speech recognition suffers the problem that increasing accuracy doesn't linearly scale with usefulness. Systems that aren't close to the high level of accuracy humans can perform don't have much value. This low value is because even one wrong word in a sentence can completely change the meaning. Thus, if the average sentence length is 20 words, even 95% accuracy would average an error in every sentence.

1.2.1 Easier Tasks

Early research in speech recognition mainly focused on transcribing speech consisting of people reading text passages aloud. This is considered the easiest type of speech for speech recognition. After speech recognition systems started being able to perform well on that task, research shifted to telephone conversations. Telephone speech is fairly easy for speech recognition because only two people are talking, and rarely at the same time. Also, there usually isn't much background noise and people put additional effort into speaking clearly on the telephone because they don't have body language to clarify misunderstandings. In recent years speech recognition systems have approached human performance for these tasks [27].

1.2.2 More Difficult Tasks

However, the vast majority of human speech is not reading aloud or over telephones. There is still significant improvement needed for the remaining, more difficult, tasks. For many of these tasks there is domain-specific language used. Speech recognition systems are based on having a vocabulary that they try to fit their transcription to. Domain-specific language is unlikely to have been part of that vocabulary. Thus speech recognition systems will have worse accuracy on domain-specific speech due to missed words. Simply adding words from every domain isn't feasible for speech recognition systems because every vocabulary word added means another possible result. The more possible results there are, the harder the speech recognition system's choice is.

1.2.3 Domain-specific Systems

There is an implicit assumption in the previous section. It assumes that a speech recognition system can only have a single vocabulary. However, that assumption is no longer limiting. While speech recognition systems used to take significant effort by linguists and other human specialists to tune, state of the art systems now use machine learning. Machine learning systems have the huge advantage of being able to adapt to different tasks with minimal human effort. This is because machine learning

based speech recognition systems can perform all steps, from taking in training data, to being ready for use, automatically.

This new flexibility enabled by machine learning automation is most valuable for creating speech recognition systems for domain-specific speech. Being able to dynamically create a speech recognition system for specific domains has the potential to improve accuracy to the point where such systems become practical for standard use. For a dynamic speech recognition system, the only manual role is picking what training data to give the speech recognition system, which you don't have to be a linguist to do.

Thus, with the aim of improving accuracy in a domain with specific language, this thesis examines legislative domain speech recognition. We experiment using a custom language models trained on legislative speech with the Watson speech to text system. Watson provided an excellent environment to test domain specific language models by pairing them with a constant general model. We evaluated several hypotheses about training data: training data most recent to the intended use, most specific to the intended use, and randomly gathered across the total possible training data.

1.3 Contribution

This thesis evaluates choices of training data for use in training a speech recognition system in a specific domain. First, we demonstrate if domain-specific training can improve accuracy for that domain. If that is established, then we suggest strategy to people choosing training data. We know that training data choice affects speech recognition system accuracy. So, for optimal accuracy, an optimal choice of training data is required. By evaluating the choices, we can recommend the best strategy for choosing training data. Helping people make better choices in training data means they should be able to achieve higher accuracy with their speech recognition systems.

Chapter 2

RELATED WORK

2.1 Judicial Domain

The most closely related work to this thesis is a project done in Europe to use automated transcription in the judicial domain. Part of the European Project, "Judicial Management by Digital Libraries Semantics" aims to collect, enrich and share multimedia documents, annotated with embedded semantic, minimizing manual transcription activities [18]. They used a set of audio recordings which were taken in the courtrooms of Naples and Wroclaw, during several trial sessions, and made available for ASR experiments. The project focused primarily on acoustic modeling, but also included a language model. For the Naples courtroom the authors trained three 4-gram based language models. An out of domain model was trained on a corpus mainly formed by newswire and newspaper articles and was 606 million words in length. An in domain corpus was mainly formed by judicial proceedings of 25 million words. Lastly, an adapted language model which weighs and mixes the 4-gram counts of both the out of domain and in domain corpora was created. The authors found that both the in domain and adapted language models gave similar results and outperformed the out of domain model [18].

2.2 Transfer Learning

Machine learning typically uses training and test data from the same domain. However, in some cases gathering training data that matches the use case can be difficult and expensive. Therefore gaining the ability to train from a related domain and then be able to run on the target domain is valuable. This is considered a transfer of using knowledge from one domain to a related one. For example, there might be an abundance of training data on text sentiment for digital cameras, but what we want is to classify sentiment for are food reviews. Transfer learning would allow training in the domain with digital cameras for use in food review classification.

Transfer learning is a very active field having produced hundreds of academic papers in recent years [26]. It is a developing field with new techniques continually being tested. There are homogeneous techniques for instance-based, feature-based, parameter-based, and relational-based information transfer. Also, there are heterogeneous transfer techniques with asymmetric and symmetric transformations. Transfer learning systems can either do a one-stage process of simultaneously performing domain adaptation while creating the final classifier, or have two separate steps [26].

Transfer learning relates to this thesis in that the experiment baseline is only using the general model without domain specific language. By adding domain-specific language as a custom model to augment the general model this is in effect attempting to transfer the general language understanding to the domain. However, this isn't considered transfer learning in a classic sense, because of the extensive overlap between the general model and the custom model.

2.3 Machine Translation

Speech recognition is sometimes used with speech translation. Machine translation is a major field and researchers are applying many similar machine learning techniques as speech recognition to it. Machine translation is different from speech recognition in that high quality domain specific machine translation systems are in higher demand than general machine translation systems [4].

State of the art machine translation systems use neural nets to perform end-to-end training of a translation system without dealing with word alignments or translation rules (neural machine translation). There is significant research being done into domain adaptation for neural machine translation. It can be done in a data centric way using monolingual corpora, synthetic parallel corpora generation, and out-of-domain parallel corpora [4]. It can also be done via model-centric approaches which adjust the training object, system architecture, or decoding algorithm. This is shown in Figure 2.1.

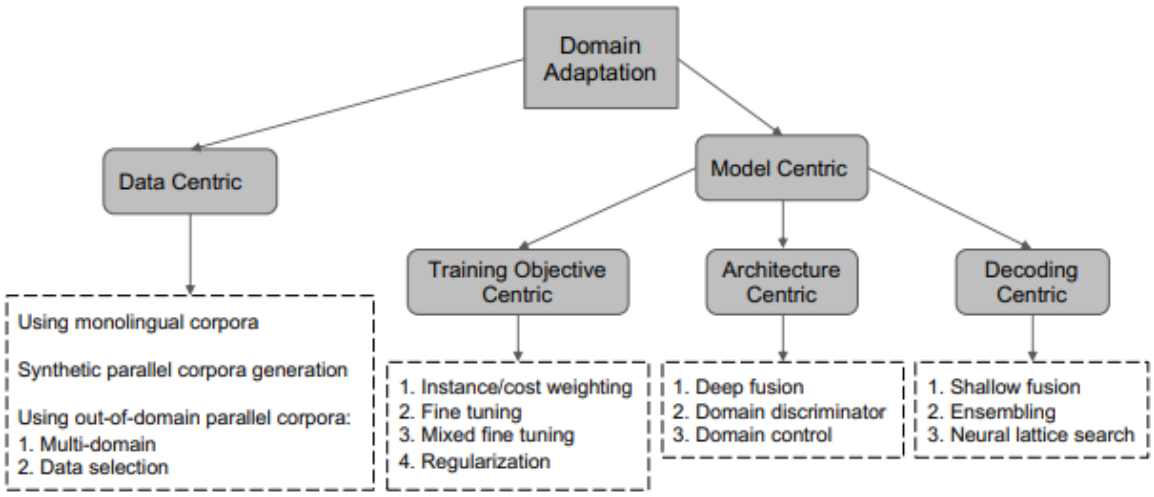


Figure 2.1: Options for domain adaption in neural machine translation [4].

Chapter 3

BACKGROUND

3.1 Speech Recognition

In recent years, automated systems have reached or surpassed human performance in certain speech recognition tasks. The tasks set for speech recognition tasks have grown progressively more difficult, from limited tasks with a small vocabulary and carefully controlled grammar, to carefully read newspaper speech [2], to Broadcast News [11].

3.1.1 Conversational Telephone Speech

For the last 5 years, most focus has been on the latest standard of conversational telephone speech. Conversational telephone speech "is especially difficult due to the spontaneous (neither read nor planned) nature of the speech, its informality, and the self-corrections, hesitations and other disfluencies that are pervasive [27]."

The largest and best studied conversation corpora are the Switchboard [10] and Fisher [6] data collections. Transcription quality is typically scored by word error rate. Human performance was rated at 4% word error rate on the task [17]. However, that error rate did not include an actual source for this numbers. So a more recent paper by Microsoft Research had professional transcribers do the Switchboard and CallHome (CallHome is part of Fisher [5]) portions of the NIST eval 2000 test set, and got results for Switchboard of 5.9% and CallHome of 11.3% [27]. This got IBM Research to run their own human transcription test which returned 5.1% on Switchboard and 6.8% on CallHome [22]. These tests means it is currently up for debate if automated systems have reached human performance. The best automated system as of March 2018, is CAPIO's Conversation Speech Recognition System which achieved 5.0% on Switchboard and 9.1% on Call Home [12]. These scores are shown in Table 3.1.

Table 3.1: Speech Recognition Word Error Rates for Conversational Telephone Speech

Switchboard WER%	CallHome WER%	Type	Paper
5.0	9.1	Automated	[12]
5.1	6.8	Human	[22]
5.1	9.9	Automated	[16]
5.5	10.3	Automated	[22]
5.8	11.0	Automated	[27]
5.9	11.3	Human	[27]

IBM speculated that the significant difference between the two sets, even though both were conversational telephone speech, are due number of speakers and speaker formality. Switchboard had fewer speakers, and speakers were strangers that spoke in a more formal style compared to CallHome's higher number of speakers, who were family and spoke informally [22]. This demonstrates that not only does the type of speech, such as earlier mentioned newspaper or Broadcast News, compared to conversational telephone make a significant difference in recognition accuracy, but also differences in the speakers themselves and the relationships between them.

3.1.2 IBM Speech Recognition

In 2016 IBM published a paper describing its English conversational telephone speech recognition system [21]. This paper is referenced in Watson speech to text science background section [15] and is likely similar to Watson's system which has not been published. The system is divided into acoustic and language sides.

On the acoustic side they use a score fusion of three models: recurrent nets with maxout activations, very deep convolutional nets with 3x3 kernels, and bidirectional long-short term memory nets which operate on bottleneck features. Recurrent neural networks have connections between nodes be a directed graph along a sequence, which enables them to use their internal state for sequences of inputs. Using maxout activations means that nodes are reduced by half by taking the higher of the pair.

Convolutional neural networks use layers of neurons that feed forward, but are only influenced by nearby neurons, such as in a 3 by 3 box. Classic convolutional networks only have 2 layers, but very deep with 6-10 layers have been found to have better performance. A long short-term memory neural network is where neurons have the option to remember their value for an arbitrarily long period of time. This makes them good for time sequence usage, such as how IBM uses them with subsequences in an utterance. They use a small bottleneck regardless of the number of layers for the utterances to help processing speed. IBM noted the long short-term memory neural network did not provide additional gains beyonds the recurrent neural network and very deep convolutional network, but they are continuing to experiment with it.

On the language side they use a 4-gram model with Kneser-Ney smoothing. Kneser-Ney smoothing is a method that tries to eliminate noise by subtracting a fixed value from less common terms to omit n-grams with lower frequencies. This is done to each of the 4 corpora they are training with. The 4 component language models are linearly interpolated and then entropy pruning is applied. Entropy pruning removes n-grams that won't affect the model's results to reduce model size. This resulted in a 4-gram language model consisting of 36 million n-grams, which was combined with their acoustic model for use in their system.

3.2 Legislative Domain

Given time domains develop some of their own terminology and style. Whether it be a subfield of chemistry or a category of video games. Sometimes they borrow from adjacent domains. New terms are not necessarily new words, but new uses of existing words. Speech in the legislative domain has many terms from law, as would be expected due to the circular relationship of legislatures creating laws which lawyers pick terms to use when talking about. The current difference is the legal domain tends to use more terms from philosophers and political theorists, while legislatures use more terms from sociology. One clear source of terms is the Supreme Court since their word usage is carefully examined and matters for the legal and legislative domains.

For example, legislatures use terms such as COLA (Cost of Living Adjustment),

especially in states like California. Another example is engrossment: "The process of comparing the printed bill to ensure it is identical to the original and to verify that any amendments have been correctly inserted [3]."

3.2.1 California Legislature

California has a bicameral legislature with 80 members in the Assembly, and 40 members in the Senate. It has two-year sessions. The majority of legislative discussions take place in committee hearings where the testimony for and against bills are heard before the appropriate policy committee. There are committees in each chamber, and the most important are standing committees which meet on a regular basis throughout the year.

3.3 Digital Democracy

Digital Democracy provides a free online platform which offers a searchable database of state legislative floor sessions and committee hearings. Hearings have complete and professional-level transcripts time tagged with their video. Users can search by keyword, topic, speaker, committee, organization or date to find information they are interested in [8]. As of June 2018, Digital Democracy covers the state legislatures in California, New York, Florida, and Texas.

The primary unique work Digital Democracy does is the transcript creation using a mix of automated and manual systems. These transcripts provide a large amount of legislative speech suitable for use in training speech recognition systems. They also are a use case of speech recognition because they try to create these transcripts at minimal cost while maintaining quality. As speech recognition systems can do more of the work, expensive human completion and quality checking is needed less.

Digital Democracy granted access to their MySQL database which provides the data for this thesis. Appendix B gives details about working with that data. It describes the tables accessed, and lists queries used to retrieve data.

3.4 Tools

This thesis relies on several tools for conducting its experiment. The primary tool is the Watson Speech to Text system [13]. To prepare audio before speech recognition ffmpeg [9] is used, and after speech recognition for scoring results Sclite [19] is used.

3.4.1 Watson

Part of IBM Cloud, Watson Speech to Text is a service that takes audio input and returns text output [13]. Watson can transcribe Ogg or Web Media (WebM) audio with the Opus or Vorbis codec, MP3 or MPEG, Waveform Audio File Format (WAV), Free Lossless Audio Codec (FLAC), Linear 16-bit Pulse-Code Modulation (PCM), mu-law (or u-law) audio, and basic audio. Watson has broadband and narrowband models for audio that is sampled at a minimum rate of 16kHz or 8kHz respectively. Watson can take as input up to 100 MB of audio to the service as a continuous stream of data chunks or as a one-shot delivery, passing all of the data at one time. Watson's 100MB limit is roughly 45 minutes of audio in the input format of mono, 16KHz, FLAC.

The service was developed with a broad, general audience in mind. The service's base vocabulary contains words that are used in everyday conversation. The general model provides sufficiently accurate recognition for a variety of applications, but it can lack knowledge of specific terms that are associated with particular domains. Watson has a language model customization interface which lets the user improve the accuracy of speech recognition for domains such as medicine, law, information technology, and others. This is done by allowing the user to create a custom language model to expand and tailor the vocabulary of the base model to add domain-specific terminology, such as legislative terms [3].

A custom language model is created by adding corpora. These are plain text documents that use terminology from one's domain. Watson building vocabulary for a custom model by extracting terms that do not exist in its base vocabulary. The service's accuracy is improved by using corpora to provide as many examples as possible of how domain-specific words, which they call out of vocab words, are used in

the domain. The more sentences added that represent the context in which speakers use words from the domain, the better the service's recognition accuracy. Watson processes words including information about n-grams, so it is useful for the corpora to use words repeatedly in different contexts.

3.4.2 HTTP REST API

For utilizing Watson it provides a websocket interface, HTTP REST interface, and asynchronous HTTP interface. This thesis uses the HTTP REST interface. The interface is used by sending HTTP requests that provide commands to the service. In their documentation and this thesis, cURL is used for sending the requests [14]. Commands use the appropriate HTTP Verb such as GET for updates, POST to tell instructions and upload files, and DELETE to delete components.

3.4.3 FFmpeg

FFmpeg is "a complete, cross-platform solution to record, convert and stream audio and video [9]." It provides extensive options to manipulate audio and video files. It can be run in a shell on the command line. For this thesis it is used to extract audio from videos and convert its format.

3.4.4 NIST Sclite

Sclite is part of the National Institute of Standards and Technology's Speech Recognition Scoring Toolkit (SCTK) [19]. It is a tool for scoring and evaluating the output of speech recognition systems. The program compares the hypothesis text output by the speech recognizer to the correct reference text. After aligning the texts and comparing, statistics are generated during the scoring process. Several types of reports can be output which summarize the performance of the recognition system.

Chapter 4

EXPERIMENT PROCESS

What training data is best for a Watson custom language model? The training data that causes the most accurate transcription results is best. This thesis's experiment evaluates hypotheses of training data choice.

4.1 Selecting Hypotheses

First, we must come up with testable hypotheses. We can consider a hypothesis testable if we can gather training data for it.

4.1.1 Available Information

For a hypothesis to be useful for a dynamic speech recognition system, it must be able to be used to create a model with only information available at the time it will be used. For our use case, what do we know the night before a hearing, so that a model can be trained and ready to transcribe.

The main piece of information known is what committee is hosting the hearing, unless it is a floor session. There is sometimes an agenda for a hearing listing bills to be discussed, but not always and they may not stick to the agenda. From knowing the committee we also know which members are on that committee.

4.1.2 Recent Hypothesis

Another angle to think about is what prior discussion will be similar to what will be discussed. When training Watson in this legislative style it should be trained with data that closely matches what it will be used to transcribe. Without reliable agendas a good predictor of what will be discussed may be what was recently discussed. There tends to be trends in the legislature about what is popular, often in keeping with what has public attention at the moment. This might be issues such as health care or gun

control. There are also periods of time before deadlines when topics are frequently discussed such as the budget. This suggests the hypothesis of **recent** discussion as being good training data.

4.1.3 Specific Hypothesis

Since we know what committee is hosting the hearing we can look at its older discussion which might be similar. Since committees have a **specific** topic, there might be the most similarity by training on what that committee has said previously.

4.1.4 Random Hypothesis

However, what if these two hypotheses are too focused? Maybe training data should be broadly gathered to make it more representative. By selecting data at **random** we can avoid biases or assumptions.

4.1.5 Training Data Sizes

One assumption beyond those choices that we should test is if more training data is always better. Larger training data set sizes means longer to gather, upload, and train. While the general trend in machine learning is more data is better, that might not be the case here. What if a certain subset of the training data is more helpful than the rest, which is just diluting its value? For example, with the recent hypothesis, maybe only the last week is of high value and older than that is no different than random. To test this we can add 3 hypotheses, with each being the same as described above, except with a smaller training data size. Since Watson's training data size limit is 10 million words, a smaller size of 10%, so 1 million words, seems reasonable. This brings us up to 6 models.

Table 4.1: Hypotheses and Models

Hypothesis	Model
Recent	Recent_1
Recent	Recent_10
Specific	Specific_1
Specific	Specific_10
Random	Random_1
Random	Random_10
Control	No_Model

Finally, there is the option of not having training data and only using Watson's general model. Each of the above models can be compared against this control. We expect this to quantify how much better they are. In contrast, the experiment's null hypothesis is, models perform the same or worse when compared to the control. Though we expect models to perform better than the control, not worse. All hypotheses and their corresponding small and large models are listed in Table 4.1.

4.2 Choosing the Test Set

To choose the test set, the first decision is how large the test set should be. Longer is better for more data and due to the automated nature of the experiment adding length doesn't add much work. However, this is experiment isn't funded so the length is bound by the limits of the Watson trial. To fit the 6 hypotheses and baseline into the 15 hours of the trial the test set could be 2 hours which would still leave a 1 hour margin for an initial test video to check for bugs in the process.

With a two hour duration decided, the next question is what content should be in the test. The selection should be representative of the data set and useful for testing the hypotheses. Committee hearings make up the vast majority of audio, in California in 2016 the breakdown was approximately 84% of committee hearing audio and 16% of floor session audio. Therefore the test should be of committee audio.

4.2.1 Choosing the Committee

There are hundreds of committees and we can choose one or more for the test set. While including many different committees in the test set would make it more representative, it would make it more difficult to choose training data for the specificity hypothesis. By choosing one committee then it is clear to gather training data from that committee for the specificity hypothesis. However, individual committees may not have much data, so the one chosen should be the one with the most data.

Before querying two limits were decided on. First, only California committees would be considered since while Digital Democracy also has many hours of New York committees, their audio quality is significantly worse, which would mean poorer results that reduce distinction between models. Second, a recent date was picked which had finished transcripts and all data must come from before then. Since California legislature was in session during the experiment if a date had not been picked then the code wouldn't have been reproducible since different videos would be picked after each update.

For each California committee the sum of the duration of its hearings before May 1st 2018 was calculated. The SQL queries for gathering information on committees are listed in Appendix B.2.

Once we get the total duration of each committee we put them in a table and sorted by duration. Since the Digital Democracy database considers floor sessions as a committee they topped the list. Then we look for which committee has the most hours when its hours for both sessions were combined. This was the Senate Standing Committee on Education which had 91 hours in the 2015 session and 83 hours so far in the 2017 session, totaling 174 hours.

Table 4.2: Top California Committees by Hours Recorded

Name	Session	Hours
Senate Floor	2015	295
Assembly Floor	2015	260
Senate Floor	2017	204
Assembly Floor	2017	198
Senate Budget Subcommittee No. 3 on Health and Human Services	2015	107
Assembly Budget Subcommittee No. 1 on Health and Human Services	2017	101
Senate Budget and Fiscal Review Subcommittee No. 3 on Health and Human Services	2017	99
Senate Standing Committee on Education	2015	91
Senate Standing Committee on Transportation and Housing	2015	90
Assembly Budget Subcommittee No. 2 on Education Finance	2017	87
Senate Standing Committee on Education	2017	83
Senate Standing Committee on Health	2015	81
Assembly Standing Committee on Health	2015	80
Senate Standing Committee on Public Safety	2015	73

4.2.2 Gathering the Test Set

The most recent two hours of audio from the committee need to be gathered. The committee id from the 2017 session is used to to get the file ids of videos from its hearings before the date. To have a consistent result the query must be sorted by both date and duration since multiple videos can happen on the same day. The SQL query for gathering the test set is in Appendix B.3.

By sorting by date descending we know the top of the results list is the most recent. So we can work down the list downloading each file and tallying its duration.

To get the duration we use FFmpeg which outputs the video duration.

Once we know we have reached the two hour duration it's time to extract the audio from the videos. For the last video we only want to extract up to the 2 hour limit so we set a duration limit for that file when extracting. While extracting the audio it is convenient to also convert to the format Watson wants. This means making the audio mono with the `-ac 1` option and setting it to 16kHz with `-ar 16000`. We choose the flac audio format due to its use in the Watson documentation example, and its small size while being lossless.

We now have the test set audio ready for use. The two hours ended up being 4 files with the final one cut to about 5 and a half minutes. For the remainder of this thesis when an individual file is referenced the first two letters of the file id will be used for identification (af, 1c, cb, 9d).

Table 4.3: Files in Test Set

File	File Id	Duration (min:sec)	Words
af	af6990e06fcef1c49ddbb58c24deb632	21:35	3275
1c	1ce019dc5834fae16a802e38c65523b7	43:28	6388
cb	cb0d87aa2cfeaab3acf4b636e9d5153c	49:17	5240
9d	9d2358ce73664c6276617042556e8e29	5:37	740

You can watch and listen to these files by using the following url. Replace both {file id} with a File Id from Table 4.3. `https://videostorage-us-west.s3.amazonaws.com/videos/{file id}/{file id}.mp4`

4.3 Gathering the Training Data

Gathering the training data is done by querying the Digital Democracy database for a set of appropriate utterances. They are ordered according to the hypothesis and written to that hypothesis's training data file. Each utterance is written on its own line, as the Watson documentation recommends. At the point when writing an utterance would push the word count over the limit the writing stops. Python's re

package is used with regex word matching to count the number of words in each utterance. All utterance texts are converted to ASCII when pulled from the database because Watson can't handle some special characters that are in the database.

4.3.1 Random Training Data

To get random utterances we gather all utterances from CA before the date and then shuffle then. By ordering by utterance id and using a set seed this can be reproducible consistently. The SQL query for gathering the random training data is in Appendix B.5.2.

4.3.2 Recent Training Data

To get recent utterances we gather all utterances from CA before the date and order them by date descending so the most recent ones are first. The SQL query for gathering the recent training data is in Appendix B.5.1.

4.3.3 Specific Training Data

To get utterances specific to the Senate Standing Committee on Education we use its committee ids of 50 and 583 for the 2015 and 2017 sessions, respectively. Since the Hearing table doesn't identify the committee, the CommitteeHearings table must also be joined. The utterances are limited by being before the date and ordered by date descending so the most recent ones are first. That ordering matters for the smaller specific training data set, and while it could be random rather than by date, we picked by date for the same reasoning as the recent hypothesis. The SQL query for gathering the specific training data is in Appendix B.5.3.

4.3.4 Training Data Size

While Watson's training data limit of 10 million words was reached by recent and random hypotheses, the specific hypothesis's total words was only 1.5 million. This means it won't be able to be conclusively compared against the recent and random

hypotheses at the large training data set size level. All training data sizes are in Table 4.4.

Table 4.4: Training Data Sizes

Hypothesis	Words in Small Set	Words in Large Set
Random	999,941	9,999,999
Recent	999,976	9,999,961
Specific	999,979	1,585,324

4.4 Using Watson

Once the test audio and training data are gathered, then Watson Speech to Text service is used. We use it through the sessionless HTTP REST interface by sending cURL commands. The cURL commands are listed in Appendix C. Watson customization steps are creating, filling, and training the models. This is done 6 times, which is the number of models being tested. The final Watson step of transcription is done 7 times because it is done with each model as well as without any custom model.

4.4.1 Creating the Models

First, a request is sent to Watson to create a model, with the model name and description passed to it. It is implied that we're using the default Watson general model, `en-US-BroadbandModel`. When successful, Watson replies with the customization id of the newly created model, which we store in a file for safekeeping. The cURL command for creating a model is in Appendix C.1.

4.4.2 Filling the Models

Then the training data is sent to Watson for each model. Watson takes several minutes to process the file, and notes all of the words not in its general dictionary. These are called out of vocab words, and are typically domain-specific. In this case

that means legislative language, such as 'AB' or 'SB' bill designations. The cURL command for filling a model is in Appendix C.2.

4.4.3 Training the Models

Once the filled models have finished being analyzed by Watson then they can be trained. The training command is sent to Watson for each model which takes from a few minutes to half an hour to train. The cURL command for training a model is in Appendix C.3.

4.4.4 Transcribing with the Models

After the custom models are trained then transcription can begin. Variations of this command will be sent to Watson 28 times since there are 4 test files and the 6 models plus no model. The first argument for this command is the customization id to indicate which custom language model to use, or this can be left out to use no custom language model. Smart formatting tells Watson to convert strings of dates, times, and numbers into their conventional representation. Inactivity timeout set to -1 tells Watson to never timeout if there are periods of silence. The profanity filter is set to false because Digital Democracy transcripts include profanity, though it is extremely rare due to the legislative setting. Finally, timestamps are set to be included. The cURL command for transcription is in Appendix C.4.

The transcription results are set to be put into a json file for later processing. For each request, per audio file and model, Watson takes roughly double the duration of the audio file to process. Thus, the transcribing totaled over 50 hours to run.

4.5 Evaluation by Word Error Rate

Evaluation is done by comparing the transcripts generated by each model, called hypothesis files, against the verified correct transcript, called the reference file. The standard comparison measurement for speech recognition tasks is word error rate. Word error rate is the sum of insertions, deletions, and substitutions divided by

the total number of words in the reference file, times 100. Calculating word error rate is difficult because it relies on the lowest number of insertions, deletions, and substitutions.

4.5.1 Sclite and Trn Format

We use the NIST Sclite tool which calculates insertions, deletions, substitutions using a dynamic programming approach. Sclite can compare several formats, and we picked the simplest one, trn, to use. Trn format has each utterance on its own line with the utterance's id at the end of the line in parentheses. Trn can also handle speaker identification in the parentheses with the utterance id, but that isn't used.

4.5.2 Creating the Reference Transcript

To create the reference transcript, utterance text, start time, end time, and id were queried from the Digital Democracy database for each file in the test set. The text was converted to ASCII match the text sent to Watson because of some non-ASCII characters Watson can't handle. Only utterances in the files whose start time was less than the length of the file were collected. This limit only mattered for File 9d which was cut in order to have the two hour total test set. The result was ordered by time to make alignment with Watson results easier. The results from the query were written to a file, for each file in the test set, in trn format.

Listing 4.1: Utterances from reference transcript file cb in trn format

```
Casey Elliot on behalf of the City of Santa Ana in support. (32919351)
Erika Hoffman on behalf of the California School Board's Association also
   in support. (32919352)
Thank you. (32919353)
```

4.5.3 Creating the Hypothesis Transcripts

The 7 hypothesis transcripts were created from the json file outputted by Watson. The json was walked through one word at a time.

Each word was written to the file until the word's start time was after the end of an utterance. In that case the utterance id was written in parentheses and then the word was written on a new line. Also, the utterance counter was incremented to know the next utterance was reached.

Listing 4.2: Utterances from Specific_10 hypothesis transcript file cb in trn format

```
1 up on behalf of the city of Santa Ana (32919351)
2 and support Erika Hoffman on behalf of the California school boards
     association also in (32919352)
3 support thank (32919353)
```

Unfortunately the Digital Democracy data only has accuracy to the second, which causes alignment errors most commonly when a word is said during the last second of an utterance. Watson returns word timestamps to hundredths of a second, and most words are said in less than a second. In line two above, "and support" which is actually "in support" got aligned to the following utterance because they were said during the last second.

```
uid, time, endTime, CONVERT(text USING ASCII)
32919351, 1014, 1017, Casey Elliot on behalf of the City of Santa Ana in
     support.
32919352, 1020, 1023, Erika Hoffman on behalf of the California School
     Board's Association also in support.
32919353, 1023, 1024, Thank you.
```

```
["and", 1017.01, 1017.14], ["support", 1017.14, 1017.55]
["in", 1023.0, 1023.09], ["support",1023.09,1023.53], ["thank
     ",1023.93,1024.14]
```

Making alignment greater than or equal to utterance end time would help in that situation, but cause more problems. Often Digital Democracy utterances end and have the next one start on the same second. In that case the first word or two in the utterance would likely be put in the previous utterance. In the example above, utterance 32919352 ends at 1023 and utterance 32919353 starts at 1023. While "support" starts at 1023.09 and would be moved up correctly, so would "thank" which starts also in 1023 at 1023.93 and is in the correct utterance this way.

4.6 Running Sclite

Once the trn files for each model are created, all that's left is to run Sclite for the results. Sclite is run for each model with its result file (hypothesis file) compared to the Digital Democracy reference file. This means Sclite is run 28 times, due to the 6 models plus no model and 4 files. Sclite outputs raw and percentage summary results to the screen. Details on running Sclite are in Appendix A.1.

Chapter 5

RESULTS

5.1 Characterizing our Files

Before looking at model word error rates, we check our files for unexpected data. Since outliers here might affect the word error rates. Some of these will be mention them in Section 5.4.1 when looking for correlation with word error rates.

5.1.1 Utterance Length Averages

We calculated average words per utterance for each file. Two files have a lower ratio at about 32 words per utterance, while the other two files have a higher ratio at about 42 words per utterance.

Table 5.1: Average Number of Words per Utterance for each File

File	Average Words
9d	32
1c	42
af	43
cb	33

5.1.2 Parts of Speech

The Natural Language Toolkit (NLTK [20]) is used to count tokens and ratios of interesting parts of speech. Note tokens include not only words, but also punctuation and other symbols. The ratios are displayed in Table 5.2. Two outliers stood out. In 1c there are fewer proper nouns. From reading the transcript and comparing against the others we notice that this transcript has a higher ratio of committee members speaking over members of the public. Members of the public tend to frequently ref-

erence organizations, which are proper nouns, to enhance their perceived influence. The other outlier is in cb are far fewer plural nouns. This transcript includes significant testimony by Vietnamese-Americans using simple or grammatically incorrect English. Also, they are mainly testifying for themselves rather than on behalf of an organization.

Table 5.2: Part of Speech Frequencies per File

File	Tokens	Singular Nouns	Plural Nouns	Proper Nouns	Numbers	Verbs
1c	7522	10.4%	4.6%	4.3%	0.9%	18.0%
9d	888	10.9%	5.2%	10.5%	1.1%	14.5%
af	3871	12.4%	5.2%	9.7%	2.1%	14.5%
cb	6218	10.1%	2.6%	12.0%	1.5%	14.2%

5.1.3 Unique Words

We would expect more unique words per file for longer files. However, as the word count increases, we would expect a slower increase in unique words. Our files matched this expectation, and you can see in Table 5.3 that longer files have more unique words, but lower percentage of unique words. This is percentage is sometimes referred to as token to type ratio.

Table 5.3: Unique Words per File

File	Words	Unique Words	Percentage Unique
9d	792	340	42.9%
af	3378	957	28.3%
cb	5405	1140	21.1%
1c	6644	1190	17.9%

5.2 Out of Vocab Words

In the introduction we discussed adding words to speech recognition systems' vocabulary. While IBM does not publish the vocabulary Watson uses, we can find out what words our training data adds for each model. Watson calls these new words added "out of vocab words."

Both the large and small models followed a similar pattern per hypothesis. Models using the random hypothesis found the most out of vocab word words. Recent didn't find as many. Specific found significantly fewer.

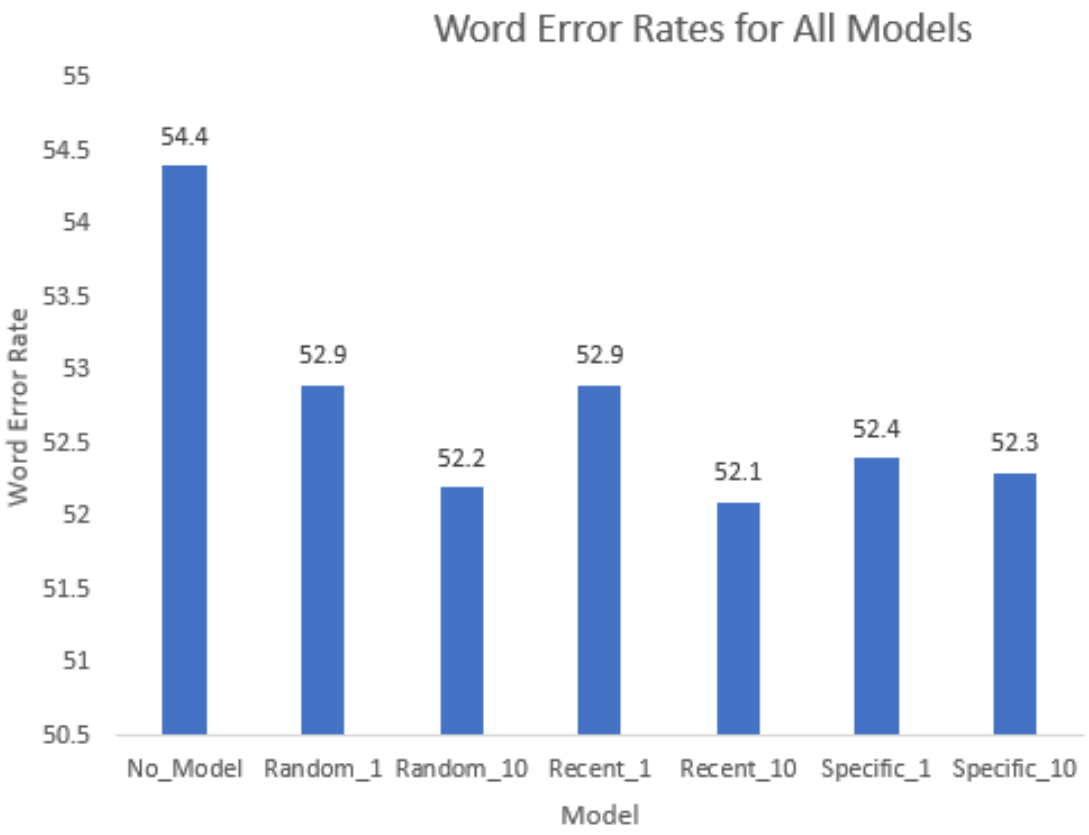
Table 5.4: Out of Vocab Word Counts per Model

Model	Out of Vocab Words
Random_10	15441
Recent_10	13297
Specific_10	2556
Random_1	2428
Recent_1	2151
Specific_1	1634

5.3 Sclite Results

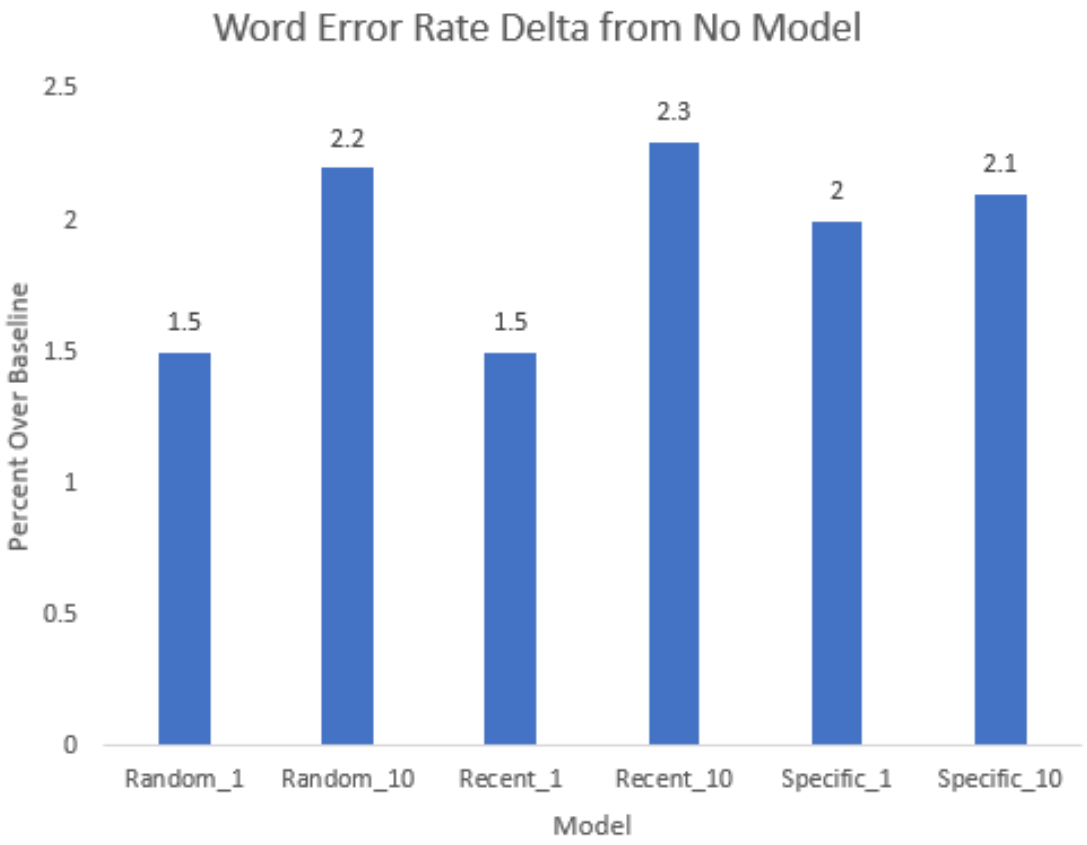
Sclite outputs results on a per file basis. An example Sclite results output in shown in Figure A.1. Appendix A has tables of Sclite output for all files.

To get a total word error rate for each model we summed the word errors in each test set file and divided it by the total words of the test set. This gave the results shown in Figure 5.4. In addition to word error rate, Sclite gives utterance error rate. This was identical for all files at 100%. No file had a single perfect utterance. Which we suspect is due to relatively few short utterances in the test set, and errors along utterance boundaries.



5.4 Hypothesis Scores

Since a lower word error rate is better, no model performed the worst at 54.4%. Small recent and random models made the least improvement with scores of 52.9%. The small specific model did better than the other small models with a score of 52.4%. For the larger models recent did best at 52.1%, beating random which scored 52.2%. Large specific was last at the large level at 52.3%, but remember that it was significantly smaller. This suggests that if there were more specific training data it might have performed best.


Word Error Rate Delta from No Model
Percent Over Baseline
2.5
2
1.5
1
0.5
0
1.5
2.2
1.5
2.3
2
2.1
Random_1
Random_10
Recent_1
Recent_10
Specific_1
Specific_10
Model

5.4.1 Scores by File

It is worth noting that there was large variance in word error rate between files in the test set. When using a custom model file 9d averaged only 36.2% error rate, file af was close at 37.5%, whiles file 1c and cb were much worse at 49.8% and 67.5%, respectively.

At first glance file size might seem significant since the smallest file scored best. However, the second smallest file, which is several times larger, scored nearly as well, and the largest file wasn't worst.

When going through the transcripts what is notable is that the worst two files, 1c and cb, both have some foreign language utterances as Spanish speakers testified at the hearings. This likely has some significance since Watson uses an English-only model, and future work should try filtering foreign language out. It was left in for this experiment to be representative of speech in the California legislature.

Non-native speakers could have influenced the error rate in file cb. We found in Section 5.1.2 simpler or grammatically incorrect English testimony by Vietnamese-Americans.

Another factor to consider is utterances verses words. It was noted earlier that the lack of precision in Digital Democracy's utterance boundary times can cause errors. Therefore if a file has more short utterances it might have a higher error rate. The files' average words per utterance is about 32 for 9d and cb, while about 42 for 1c and af. These pairings don't match the lower and higher word error rate pairings, so this was likely not a significant factor in error rates.

Similarly, when we look at the particular characteristics of the texts in terms of parts of speech per file, no clear correlation is seen.

Table 5.5: Scores on Individual Files for Each Model

File	Words	None	Rand_1	Rand_10	Rec_1	Rec_10	Spec_1	Spec_10	Avg
9d	740	38.4	36.4	37.2	37	35.4	36.1	35.1	36.2
1c	6388	50.9	50.2	49.7	50.3	49.5	49.5	49.4	49.8
af	3275	41	37.8	37.4	37.4	36.9	37.6	37.4	37.5
cb	5240	69.2	68	66.7	67.9	67.1	67.7	67.6	67.5

5.5 Statistical Significance

An experiment has statistical significance when it is very unlikely to have occurred given the null hypothesis. Statistical significance is used when an experiment draws samples from a population. Our null hypothesis is that models will perform the same or worse than the control. Our population is the word error scores of possible transcriptions for the test set using no model. Our samples are the word error scores of transcriptions on our test set using our models. For our samples to be statistically significant, they must be very unlikely to have resulted from the control. However, to get a distribution of possible results from the control we would have to run more experiments. Without doing that we cannot state our results are statistically significant.

Without stating statistical significance, we can still look at our data to see how much of a gap there is between the control and our models. Our samples had 15,643 words in them. The control had 8,502 word errors while the samples ranged from 8,272 to 8,144 word errors, shown in Table 5.6. That 230 word errors gap is about 2.7% of the control errors. So, if it is likely that number of word errors from a control could vary as much as 2.7% then our experiment wouldnt be statistically significant. However, since none of our models varied from the control by more than 2.3%, we think 2.7% variance is very unlikely, and thus our results are worth considering.

Table 5.6: Word Errors per Model

Model	Word Errors
Control	8502
Random_1	8272
Recent_1	8270
Specific_1	8203
Specific_10	8181
Random_10	8171
Recent_10	8144

Chapter 6

CONCLUSION

As speech recognition is integrated into an increasing number of products and services it is only going to become more important. As it spreads there are many newly feasible tasks, such as for transcribing legislative hearing. Tasks like this have a rare language style specific to their domain. In order to maximize speech recognition systems' use in these areas they must have high accuracy which means understanding domain-specific language. Recent machine-learning based speech recognition systems make feasible creating speech recognition systems for each domain. Such a dynamic speech recognition system only needs domain-specific training data before it can be used.

To contribute towards creating dynamic system with high accuracy we looking at the legislative domain using a custom language model with the Watson speech to text system. Watson provided an excellent environment to test only domain-specific training data due to allowing a separate custom language model from a constant general model. We evaluated hypotheses about training data that is most recent to the intended use, that is most specific to the intended use, and that is randomly gathered across the total possible training data.

Our experiment found that using a custom language model to augment the general model reduced word error rate by at least 1.5%. At a smaller training data size, using training data most specific to the use case performed best with a word error rate of 52.4%, which is 2% better than the control with no model. For the larger models we found using training data that was most recent to the test set performed best at 52.1% word error rate, though this was only a 0.1% improvement over the random training data model which scored 52.2%. Large specific was last at the large level, but it was still fairly close at 52.3%. Since it was this close despite it not being the full training data size, this suggests that if there were more specific training data it might have performed best.

Researchers and developers can take from this that it is worth gathering domain-specific training data for non-general speech recognition uses. They should try to

gather information that is specific and recent to the intended use for better accuracy.

6.1 Future Work

Several choices were made during this experiment, and each of those are grounds for future experiments. For example, the random model was made from randomization at the utterance level, it could also be done at the video or hearing level. We'd like to see more done with the specific model, hopefully with enough training data to reach the full 10 million words. Watson's custom acoustic models are a clear next step and would have been included in this thesis if they didn't have a bug at the time. Finally, Watson recently got a new feature to adjust the weight between the general model and custom models. We suspect higher weighting of custom models would improve accuracy in this case.

BIBLIOGRAPHY

[1] Cal Poly Github. `http://www.github.com/CalPoly`.

[2] D. Amodei, R. Anubhai, E. Battenberg, C. Case, J. Casper, B. Catanzaro, J. Chen, M. Chrzanowski, A. Coates, G. Diamos, E. Elsen, J. Engel, L. Fan, C. Fougner, T. Han, A. Y. Hannun, B. Jun, P. LeGresley, L. Lin, S. Narang, A. Y. Ng, S. Ozair, R. Prenger, J. Raiman, S. Satheesh, D. Seetapun, S. Sengupta, Y. Wang, Z. Wang, C. Wang, B. Xiao, D. Yogatama, J. Zhan, and Z. Zhu. Deep speech 2: End-to-end speech recognition in english and mandarin. volume abs/1512.02595, 2015.

[3] California State Assembly. Glossary of Legislative Terms. `http://www.legislature.ca.gov/quicklinks/glossary.html`.

[4] C. Chu and R. Wang. A survey of domain adaptation for neural machine translation. *arXiv preprint arXiv:1806.00258*, June 2018.

[5] C. Cieri, D. Graff, O. Kimball, D. Miller, and K. Walker. Fisher english training speech part 1 speech. 2004. Made available by Linguistic Data Consortium at `https://catalog.ldc.upenn.edu/LDC2004S13`.

[6] C. Cieri, D. Miller, and K. Walker. The fisher corpus: a resource for the next generations of speech-to-text. In *LREC*, volume 4, pages 69–71, 2004.

[7] A. Cohen. Fuzzywuzzy Python Package, 2011. `https://pypi.python.org/pypi/fuzzywuzzy`.

[8] Digital Democracy. About Digital Democracy. `https://www.digitaldemocracy.org/about-digital-democracy`.

[9] FFmpeg. FFmpeg Video and Audio Converter. `https://ffmpeg.org/`.

[10] J. J. Godfrey, E. C. Holliman, and J. McDaniel. Switchboard: Telephone speech corpus for research and development. In *Acoustics, Speech, and Signal Processing, 1992. ICASSP-92., 1992 IEEE International Conference on*, volume 1, pages 517–520. IEEE, 1992.

[11] D. Graff, Z. Wu, R. MacIntyre, and M. Liberman. The 1996 broadcast news speech and language-model corpus. In *Proceedings of the DARPA Workshop on Spoken Language technology*, pages 11–14, 1997.

[12] K. J. Han, A. Chandrashekaran, J. Kim, and I. Lane. The capio 2017 conversational speech recognition system. *arXiv preprint arXiv:1801.00059*, 2017.

[13] IBM. Watson Documentation About. `https://console.bluemix.net/docs/services/speech-to-text/index.html#about`.

[14] IBM. Watson http REST interface. `https://console.bluemix.net/docs/services/speech-to-text/http.html#http`.

[15] IBM. Watson: The science behind the service. `https://console.bluemix.net/docs/services/speech-to-text/science.html#science`.

[16] G. Kurata, B. Ramabhadran, G. Saon, and A. Sethy. Language modeling with highway LSTM. *CoRR*, abs/1709.06436, 2017.

[17] R. P. Lippmann. Speech recognition by machines and humans. volume 22, pages 1–15. Elsevier, 1997.

[18] J. Lf, D. Falavigna, R. Schlter, D. Giuliani, R. Gretter, and H. Ney. Evaluation of automatic transcription systems for the judicial domain. In *2010 IEEE Spoken Language Technology Workshop*, pages 206–211, Dec 2010.

[19] National Institute of Standards and Technology. SCTK Toolkit. `https://www.nist.gov/itl/iad/mig/tools`.

[20] N. Project. Natural Language Toolkit. `https://www.nltk.org/`.

[21] G. Saon, H.-K. J. Kuo, S. Rennie, and T. Sercu. The IBM 2016 english conversational telephone speech recognition system. *arXiv preprint arXiv:1604.08242*, 2016.

[22] G. Saon, G. Kurata, T. Sercu, K. Audhkhasi, S. Thomas, D. Dimitriadis, X. Cui, B. Ramabhadran, M. Picheny, L.-L. Lim, et al. English conversational telephone speech recognition by humans and machines. *arXiv preprint arXiv:1703.02136*, 2017.

[23] State of California. California State Assembly Committees. `http://assembly.ca.gov/committees`.

[24] State of California. California State Senate Committees. `http://senate.ca.gov/committees`.

[25] A. Stolcke and J. Droppo. Comparing human and machine errors in conversational speech transcription. *arXiv preprint arXiv:1708.08615*, 2017.

[26] K. Weiss, T. M. Khoshgoftaar, and D. Wang. A survey of transfer learning. volume 3, page 9. Nature Publishing Group, 2016.

[27] W. Xiong, J. Droppo, X. Huang, F. Seide, M. Seltzer, A. Stolcke, D. Yu, and G. Zweig. Achieving human parity in conversational speech recognition. volume abs/1610.05256, 2016.

APPENDICES

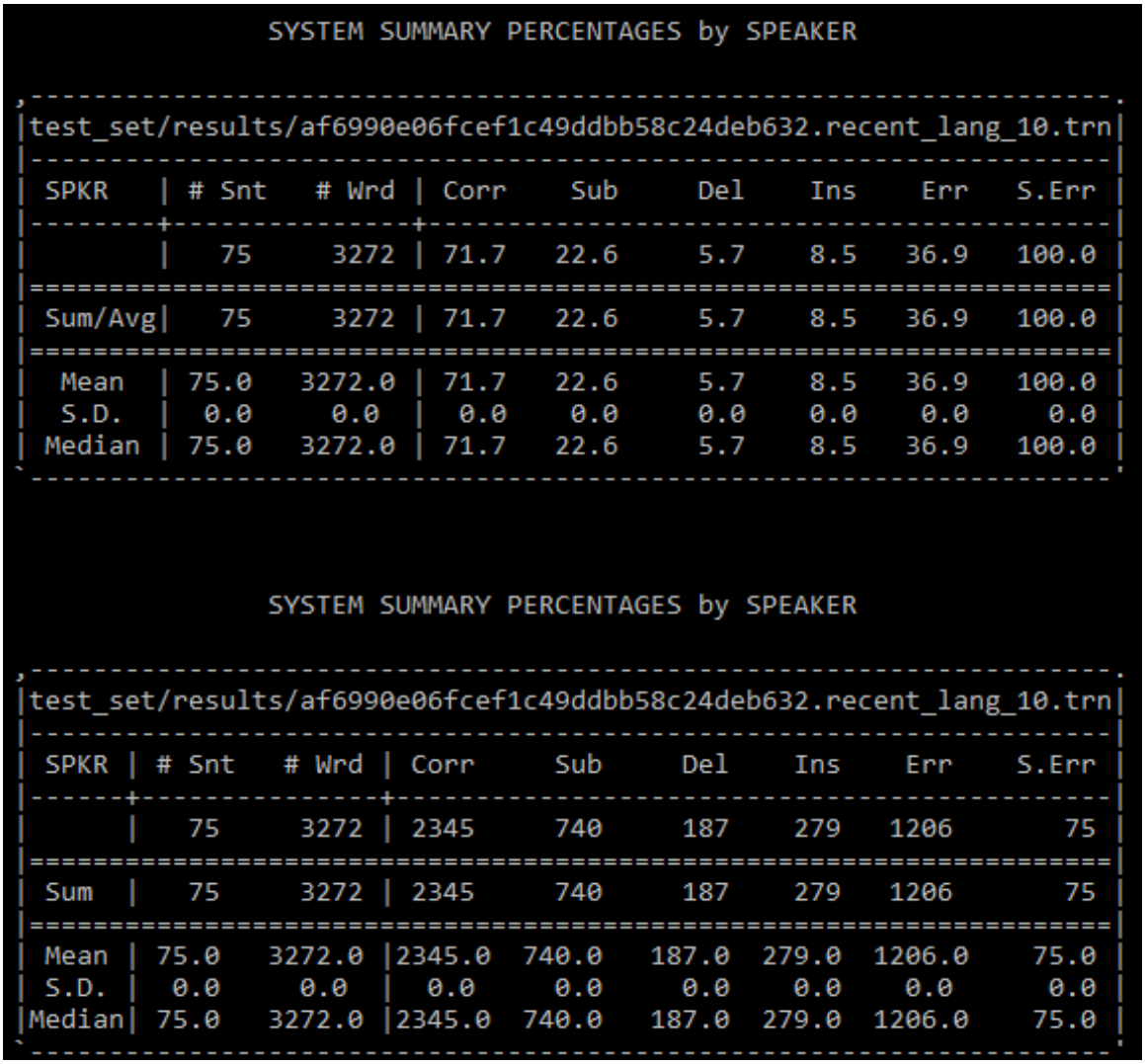
Appendix A

ALL RESULTS

A.1 Sclite Command

Sclite is run for each model with that model's results file (hypothesis file) compared to the Digital Democracy reference file. Sclite is first told the reference file and its format. Then Sclite is told the hypothesis file and its format. Because the reference file is in trn format Sclite requires the utterance id format be specified. We use the swb format where utterance id is made up of a speaker code, followed by a hyphen or underscore, followed by an utterance number. However, we don't specify speaker ids. Error messages for missing speaker ids will be generated, but can be ignored. Sclite will still calculate results without speakers identified. Sclite results are specified to be `-o rsum sum stdout` which outputs raw and percentage summary results to the screen. We are not interested in the detailed non-summary results. Non-sumary results show the insertions, deletions, and substitutions for every utterance. An example Sclite output is in Figure A.1.

```
sclite -r reffile [ fmt ] -h hypfile [ fmt ] OPTIONS
sctk-2.4.10/bin/sclite -r {ref_file} trn -h {hyp_file} trn -i swb -o rsum
    sum stdout
```



SYSTEM SUMMARY PERCENTAGES by SPEAKER

test_set/results/af6990e06fcef1c49ddbb58c24deb632.recent_lang_10.trn

SPKR	# Snt	# Wrd	Corr	Sub	Del	Ins	Err	S.Err
	75	3272	71.7	22.6	5.7	8.5	36.9	100.0
Sum/Avg	75	3272	71.7	22.6	5.7	8.5	36.9	100.0
Mean	75.0	3272.0	71.7	22.6	5.7	8.5	36.9	100.0
S.D.	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Median	75.0	3272.0	71.7	22.6	5.7	8.5	36.9	100.0

SYSTEM SUMMARY PERCENTAGES by SPEAKER

test_set/results/af6990e06fcef1c49ddbb58c24deb632.recent_lang_10.trn

SPKR	# Snt	# Wrd	Corr	Sub	Del	Ins	Err	S.Err
	75	3272	2345	740	187	279	1206	75
Sum	75	3272	2345	740	187	279	1206	75
Mean	75.0	3272.0	2345.0	740.0	187.0	279.0	1206.0	75.0
S.D.	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Median	75.0	3272.0	2345.0	740.0	187.0	279.0	1206.0	75.0

Figure A.1: The results from Sclite for file af comparing the 10m recent hypothesis against the Digital Democracy reference transcription.

A.2 Result Tables

For each model, these tables show the Sclite summary results for each file. An all row was calculated for each table. The all row sums the results in each column for all of the files. For the percentage tables, the all row percentages were calculated with each file's results weighted according to that file's word count.

A.3 No_Model Results

Word Error Rate of 54.4%.

Table A.1: No_Model Percentage Results

File	Utts	Words	Correct	Sub	Del	Ins	Word Errors	Utt Errors
9d	23	740	72.7	22.8	4.5	11.1	38.4	100
1c	153	6388	72.0	22.8	5.2	22.9	50.9	100
af	76	3275	68.5	25.8	5.6	9.5	41.0	100
cb	161	5240	66.7	28.6	4.7	35.9	69.2	100
all	413	15643	69.5	25.4	5.1	23.9	54.4	100

Table A.2: No_Model Raw Results

File	Utts	Words	Correct	Sub	Del	Ins	Word Errors	Utt Errors
9d	23	740	538	169	33	82	284	23
1c	153	6388	4600	1456	332	1463	3251	153
af	76	3275	2245	846	184	312	1342	76
cb	161	5240	3495	1501	244	1880	3625	161
all	413	15643	10878	3972	793	3737	8502	413

A.4 Random_1 Results

Word Error Rate of 52.9%. This is 1.5% better than No_Model.

Table A.3: Random_1 Percentage Results

File	Utts	Words	Correct	Sub	Del	Ins	Word Errors	Utt Errors
9d	23	740	74.5	21.4	4.2	10.8	36.4	100
1c	153	6388	72.8	21.8	5.4	23.0	50.2	100
af	76	3275	71.2	23.4	5.4	9.0	37.8	100
cb	161	5240	68.0	27.7	4.4	35.9	68.0	100
all	413	15643	70.9	24.1	5.0	23.8	52.9	100

Table A.4: Random_1 Raw Results

File	Utts	Words	Correct	Sub	Del	Ins	Word Errors	Utt Errors
9d	23	740	551	158	31	80	269	23
1c	153	6388	4653	1392	343	1469	3204	153
af	76	3275	2332	766	177	295	1238	76
cb	161	5240	3561	1449	230	1882	3561	161
all	413	15643	11097	3765	781	3726	8272	413

A.5 Random_10 Results

Word Error Rate of 52.2%. This is 2.2% better than No_Model.

Table A.5: Random_10 Percentage Results

File	Utts	Words	Correct	Sub	Del	Ins	Word Errors	Utt Errors
9d	23	740	73.8	21.9	4.3	10.9	37.2	100
1c	153	6388	73.2	21.7	5.1	22.9	49.7	100
af	76	3275	71.6	22.8	5.6	9.0	37.4	100
cb	161	5240	68.9	26.6	4.5	35.7	66.7	100
all	413	15643	71.5	23.6	5.0	23.7	52.2	100

Table A.6: Random_10 Raw Results

File	Utts	Words	Correct	Sub	Del	Ins	Word Errors	Utt Errors
9d	23	740	546	162	32	81	275	23
1c	153	6388	4676	1386	326	1462	3174	153
af	76	3275	2346	746	183	296	1225	76
cb	161	5240	3612	1394	234	1869	3497	161
all	413	15643	11180	3688	775	3708	8171	413

A.6 Recent_1 Results

Word Error Rate of 52.9%. This is 1.5% better than No_Model.

Table A.7: Recent_1 Percentage Results

File	Utts	Words	Correct	Sub	Del	Ins	Word Errors	Utt Errors
9d	23	740	73.9	22.3	3.8	10.9	37.0	100
1c	153	6388	72.6	22.2	5.2	22.9	50.3	100
af	76	3275	71.6	23.1	5.3	9.0	37.4	100
cb	161	5240	67.9	27.6	4.6	35.8	67.9	100
all	413	15643	70.9	24.2	4.9	23.7	52.9	100

Table A.8: Recent_1 Raw Results

File	Utts	Words	Correct	Sub	Del	Ins	Word Errors	Utt Errors
9d	23	740	547	165	28	81	274	23
1c	153	6388	4637	1418	333	1462	3213	153
af	76	3275	2346	757	172	295	1224	76
cb	161	5240	3557	1444	239	1876	3559	161
all	413	15643	11087	3784	772	3714	8270	413

A.7 Recent_10 Results

Word Error Rate of 52.1%. This is 2.3% better than No_Model. The Recent_10 results have one less utterance than other results because it did not transcribe anything during the final short utterance.

Table A.9: Recent_10 Percentage Results

File	Utts	Words	Correct	Sub	Del	Ins	Word Errors	Utt Errors
9d	23	740	75.3	21.1	3.6	10.7	35.4	100
1c	153	6388	73.0	21.9	5.1	22.5	49.5	100
af	75	3272	71.7	22.6	5.7	8.5	36.9	100
cb	161	5240	68.1	27.3	4.6	35.2	67.1	100
all	412	15640	71.2	23.8	5.0	23.3	52.1	100

Table A.10: Recent_10 Raw Results

File	Utts	Words	Correct	Sub	Del	Ins	Word Errors	Utt Errors
9d	23	740	557	156	27	79	262	23
1c	153	6388	4664	1396	328	1436	3160	153
af	75	3272	2345	740	187	279	1206	75
cb	161	5240	3567	1433	240	1843	3516	161
all	412	15640	11133	3725	782	3637	8144	412

A.8 Specific_1 Results

Word Error Rate of 52.4%. This is 2.0% better than No_Model.

Table A.11: Specific_1 Percentage Results

File	Utts	Words	Correct	Sub	Del	Ins	Word Errors	Utt Errors
9d	23	740	75.3	21.1	3.6	11.4	36.1	100
1c	153	6388	73.4	21.4	5.2	22.9	49.5	100
af	76	3275	71.5	23.0	5.4	9.1	37.6	100
cb	161	5240	68.5	27.1	4.4	36.2	67.7	100
all	413	15643	71.5	23.6	4.9	23.9	52.4	100

Table A.12: Specific_1 Raw Results

File	Utts	Words	Correct	Sub	Del	Ins	Word Errors	Utt Errors
9d	23	740	557	156	27	84	267	23
1c	153	6388	4689	1364	335	1461	3160	153
af	76	3275	2343	754	178	298	1230	76
cb	161	5240	3589	1421	230	1898	3549	161
all	413	15643	11178	3695	770	3741	8203	413

A.9 Specific_10 Results

Word Error Rate of 52.3%. This is 2.1% better than No_Model.

Table A.13: Specific_10 Percentage Results

File	Utts	Words	Correct	Sub	Del	Ins	Word Errors	Utt Errors
9d	23	740	75.8	20.5	3.6	10.9	35.1	100
1c	153	6388	73.4	21.4	5.2	22.8	49.4	100
af	76	3275	71.6	23.0	5.4	9.0	37.4	100
cb	161	5240	68.5	27.1	4.4	36.0	67.6	100
all	413	15643	71.5	23.6	4.9	23.8	52.3	100

Table A.14: Specific_1 Raw Results

File	Utts	Words	Correct	Sub	Del	Ins	Word Errors	Utt Errors
9d	23	740	561	152	27	81	260	23
1c	153	6388	4689	1366	333	1457	3156	153
af	76	3275	2345	753	177	295	1225	76
cb	161	5240	3589	1418	233	1889	3540	161
all	413	15643	11184	3689	770	3722	8181	413

Appendix B

WORKING WITH DIGITAL DEMOCRACY DATA

B.1 Database Tables Used

The **currentUtterance** table contains all the text being used for training data. It also contains the timestamps for utterance boundaries. These utterance boundary times are used for aligning the hypothesis text with the reference text. The **Committee** table contains information about committees. This information was used for deciding which committee should be used for testing. The **Video** table contains the video ids and file ids of video files. Video ids are used for linking utterances to hearings. File ids are used for creating the url to download videos. The **Hearing** table contains the state and date of hearings. This is used when gathering utterances for training data and testing. The **CommitteeHearings** table is used to link committees with hearings they host.

B.2 Queries for Committee Information

```
SELECT cid, house, name, session_year
       FROM Committee
       WHERE state = 'CA'

SELECT SUM(Video.duration)
       FROM Video JOIN CommitteeHearings ON Video.hid = CommitteeHearings.
          hid
       JOIN Hearing ON Hearing.hid = CommitteeHearings.hid
       WHERE cid = {Committee Id}
       AND date < {May 1st 2018}
```

B.3 Query for Test Set File Ids

```
SELECT fileId
        FROM Video JOIN CommitteeHearings ON Video.hid = CommitteeHearings.
            hid
        JOIN Hearing ON Hearing.hid = CommitteeHearings.hid
        WHERE cid = {Committee Id}
        AND date < {May 1st 2018}
        ORDER BY date DESC, duration DESC
```

B.4 Converting the Test Set

Digital Democracy stores videos in mp4 format. FFmpeg is used to extract the audio and convert it to be suitable for Watson.

```
ffmpeg -hide_banner -i {video name} -ac 1 -ar 16000 -t {duration limit} {
    audio output name(.flac)}
```

B.5 Queries for Training Data

All queries are set to get utterances before the earliest date of a video in the test set. This ensures there is no overlap between the training and testing sets. The same queries are used for both the large and small versions of each hypothesis. The limiting of training data according to model size is handled by Python code.

B.5.1 Recent Hypothesis Training Data

```
SELECT CONVERT(text USING ASCII)
        FROM currentUtterance
        JOIN Video ON currentUtterance.vid = Video.vid
        JOIN Hearing ON Hearing.hid = Video.hid
        WHERE date < '{earliest date}' AND Hearing.state = 'CA'
        ORDER BY date desc
```

B.5.2 Random Hypothesis Training Data

Random hypothesis training data gathering uses the same query as for gathering Recent hypothesis training data, except for the order. The randomization is handled by Python code.

```
SELECT CONVERT(text USING ASCII)
        FROM currentUtterance
        JOIN Video ON currentUtterance.vid = Video.vid
        JOIN Hearing ON Hearing.hid = Video.hid
        WHERE date < '{earliest date}' AND Hearing.state = 'CA'
        ORDER BY uid
```

B.5.3 Specific Hypothesis Training Data

Cid 50 is the committee id of the 2015 session Senate Standing Committee on Education. Cid 583 is the committee id of the same committee for the 2017 session.

```
SELECT CONVERT(text USING ASCII)
        FROM currentUtterance
        JOIN Video ON currentUtterance.vid = Video.vid
        JOIN Hearing ON Hearing.hid = Video.hid
        JOIN CommitteeHearings ON CommitteeHearings.hid = Hearing.hid
        WHERE date < '{earliest date}' AND (cid = 50 or cid = 583)
        ORDER BY date desc
```

B.6 Querying for the Reference Transcript

Utterance end times are not used to create the reference transcript, but are used when creating model transcripts.

```
SELECT uid, time, endTime, CONVERT(text USING ASCII)
```

```
FROM currentUtterance

JOIN Video ON Video.vid = currentUtterance.vid

WHERE fileId = '{}' AND time < {}

ORDER BY time
```

Appendix C

WATSON CURL COMMANDS

Watson can be used via an HTTP REST interface. We use cURL commands to interact with it. Despite cURL being a command line tool, we called it from Python using the subprocess module. Calling cURL from Python allows for automation by looping through the list of models and calling cURL with each model's id substituted into the command.

C.1 Command to Create a Model

```
curl -X POST -u {account} --header "Content-Type: application/json" --data
   "{{\\"name\\": \\"{model_name+model_size}\\", \\"base_model_name\\":
   \\"en-US_BroadbandModel\\", \\"description\\": \\"{model_name+
   model_size}\\"}}" "https://stream.watsonplatform.net/speech-to-text/api
   /v1/customizations"
```

C.2 Command to Fill a Model

```
curl -X POST -u {account} --data-binary @{training_file} "https://stream.
   watsonplatform.net/speech-to-text/api/v1/customizations/{
   customization_id}/corpora/{corpus_name}"
```

C.3 Command to Train a Model

```
curl -X POST -u {account} "https://stream.watsonplatform.net/speech-to-text
   /api/v1/customizations/{customization_id}/train"
```

C.4 Command to Transcribe Using a Model

To transcribe without a custom language model leave out `customization_id={customization_id }&`.

```
curl -X POST -u {account} --header "Content-Type: audio/flac" --data-binary
    @{audio_file} "https://stream.watsonplatform.net/speech-to-text/api/v1
   /recognize?customization_id={customization_id}&smart_formatting=true&
   inactivity_timeout=-1&profanity_filter=false&timestamps=true" > {
   results_json_file}
```