**Project Name: Automated Speech Recognition for Alexa**

**Brief Description/ Title**

Our project intends to create various machine learning models to predict whether a spoken word is the wake word for Alexa, i.e. "Alexa". We will use different voice datasets to train our models, and then test the accuracy of correctly identifying the word ‘Alexa’ in the test dataset using our models.

**Team members**

Jeremiah Stone

Priyanka Upadhyay

Shuo Wang

**Problem Statement**

Voice-based applications require a "wake word" to enable the voice enabled device to begin listening to commands. For our project, we will build models to identify the word ‘Alexa’, which is the common wake word for Amazon’s Voice assistant.

This is a common SLU (Spoken Language Understanding) problem in the ASR (Automatic Speech Recognition) domain, where the machine learning models identify if the spoken words match with a specific target word. In this use case, we take audio files with different spoken words (in different voices and accents), and train our model to identify if it matches with the target word in the Lexicon (the wake word “Alexa’). We will utilize different datasets of many different voices to train the models.

**Objective**

The objective of this project is to train and test different machine learning models for an Automated Speech Recognition problem, and compare the models for accuracy.

**Approach/ Methodology**

1. We have taken two different datasets to train, test and validate our model:

(i) Kaggle Alexa dataset (<https://www.kaggle.com/datasets/aanhari/alexa-dataset>). This dataset has 369 Recordings of Alexa keyword “Alexa” in different voices.

(ii) LJ Speech (<https://www.kaggle.com/datasets/mathurinache/the-lj-speech-dataset>). This consists of 13,100 short audio files of a single speaker reading passages audiobooks.

(more details in the Datasets section)

1. The data consist of audio files in .wav format. We converted .wav files to a spectogram.

(i) First, we used the decode function to convert WAV to a time series of amplitudes of the audio in the file

(ii) Second, we convert the audio frames from time domain to frequency domain with specified window length of 256ms into spectrograms.

(iii) Finally, we normalize the spectrograms data with a z-score transformation that gives us our sample features of 193.

1. Then, we encoded the labels by converting characters to a number (space: 0, A: 1, B: 2, …, nan: 31).

* “alexa” would be encoded as: [ 1 12 5 24 1].

1. We built a loss function to compute the training time loss value using the Connectionist Temporal Classification Loss, or CTC Loss algorithm. To make sure that the prediction and the true label have the same shape, we used the TF shape function.
2. We used another decoder function to convert the predictions (that are returned in numbers) into alphabets.
3. We developed two accuracy functions:

(i) A relative accuracy function, where we match the prediction with a true label, character by character. If the two characters match, it is a 1, otherwise 0. Then we calculate the average for each prediction, which becomes the accuracy score for that prediction.

(ii) An absolute accuracy function, where we match the full word/sentence with the true label. If the two match, it is 1. Otherwise 0. This is a much more stringent accuracy function, as the full word has to match with the true label.

(iii) We also used Word Error Rate (wer), a built-in function in the JIWER package as a measure of accuracy.

1. We then trained and tested six models,
2. We then used the accuracy scores described above to identify the best model for our ASR problem.

**Datasets (potential)**

In order to train robust models, we have taken two different datasets.

1. Kaggle Alexa dataset (<https://www.kaggle.com/datasets/aanhari/alexa-dataset>). This dataset has 369 Recordings of Alexa keyword “Alexa” in different voices. The recordings are in 16000 Hz WAV format. Though this is a limited dataset, it is useful to be used as a train dataset for our models, as the final outcome we want is to match the input (voice/word) to the word ‘Alexa’ in our lexicon. This model contains audio files of the word ‘Alexa’ in different voices and accents, therefore adding both complexity and diversity to the dataset, therefore reducing bias in our model.
2. LJ Speech (<https://www.kaggle.com/datasets/mathurinache/the-lj-speech-dataset>). This is a public domain speech dataset consisting of 13,100 short audio clips of a single speaker reading passages from 7 non-fiction books. A transcription is provided for each clip. We used this dataset as it is much larger than the first dataset, and has words other than ‘Alexa’, which makes it a generalizable dataset for our models. The drawbacks of this dataset are that there is only a single voice & accent for all the words. Also, as this is a very large dataset, it made the computation slow, and we had to do more data engineering to slice the dataset.

**What models did we choose and why?**

The table below lists all the models that we trained. The ‘dataset’ column provides the % split of each dataset that we used for training, validation, and testing.

**Basic Models, Logistic and Artificial Neural Networks**

| Model Name | Model Type | Hidden Layers | Hidden Units | Dataset  (train, validation, test) | Reason for Inclusion |
| --- | --- | --- | --- | --- | --- |
| model000 | LOGIT Multi-Classification |  |  | Alexa (80%,10%,10%)  LJSpeech (0%, 0%, 10%) | Baseline |
| model001ann | Artificial Neural Network | 1 | 256 | Alexa (80%,10%,10%)  LJSpeech(0%, 0%, 10%) | Neural Network without Recurrent Layer |
| model002ann | Artificial Neural Network | 2 | 256, 128 | Alexa (80%,10%,10%)  LJSpeech(0%, 0%, 10%) | Neural Network without Recurrent Layer |
| model003ann | Artificial Neural Network | 3 | 256, 128, 64 | Alexa (80%,10%,10%)  LJSpeech(0%, 0%, 10%) | Neural Network without Recurrent Layer |

**Convolutional Neural Network**

| Model Name | Model Type | 2D Conv Layers | Filter Size | Kernel Size | Strides | Dataset  (train, validation, test) | Reason for Inclusion |
| --- | --- | --- | --- | --- | --- | --- | --- |
| model007 | CNN, 2D | 2 | 32, 32 | 11, 41  11, 21 | 2, 2  1, 2 | Alexa (80%,10%,10%)  LJSpeech(0%, 0%, 10%) | Convolution to improve sample uniformity |

**Recurrent Neural Networks Trained on Alexa Dataset**

(All RNN neural networks include 2 layers of convolutional neural networks)

| Model Name | Model Type | Hidden Layers | Hidden Units | RNN Layers | RNN Units | Dataset  (train, validation, test) | Reason for Inclusion |
| --- | --- | --- | --- | --- | --- | --- | --- |
| model001 | RNN | 1 | 256 | 1 | 128 | Alexa (80%,10%,10%)  LJSpeech(0%, 0%, 10%) | Test RNN efficacy |
| model002 | RNN | 1 | 256 | 2 | 128 | Alexa (80%,10%,10%)  LJSpeech(0%, 0%, 10%) | Test RNN efficacy |
| model003 | RNN | 1 | 256 | 3 | 128 | Alexa (80%,10%,10%)  LJSpeech(0%, 0%, 10%) | Test RNN efficacy |
| model004 | RNN | 1 | 256 | 4 | 128 | Alexa (80%,10%,10%)  LJSpeech(0%, 0%, 10%) | Test RNN efficacy |
| model005 | RNN | 1 | 256 | 5 | 128 | Alexa (80%,10%,10%)  LJSpeech(0%, 0%, 10%) | Test RNN efficacy |
| model006 | RNN | 1 | 128 | 5 | 64 | Alexa (80%,10%,10%)  LJSpeech(0%, 0%, 10%) | Test RNN efficacy |

**Generalized Recurrent Neural Networks Trained on LJ Speech Dataset**

(All RNN neural networks include 2 layers of convolutional neural networks)

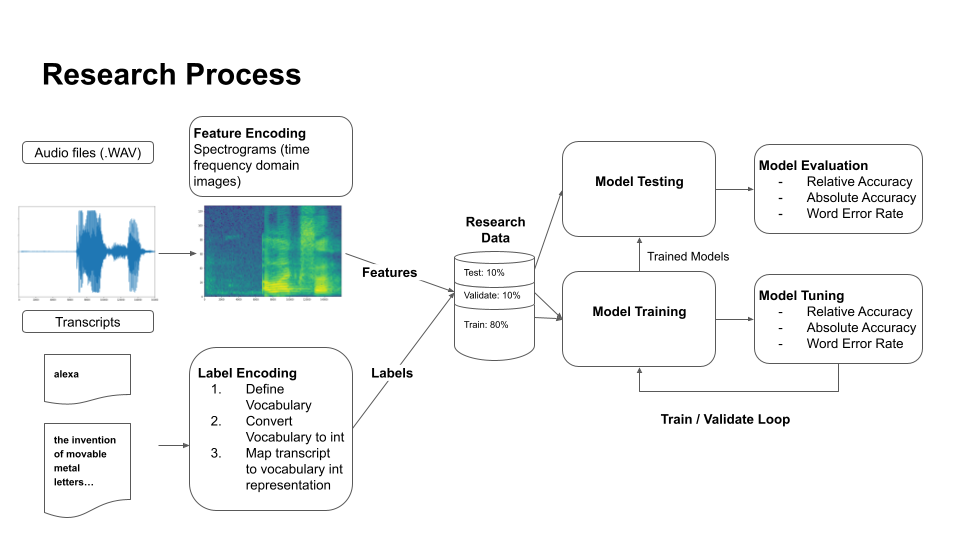
| Model Name | Model Type | Hidden Layers | Hidden Units | RNN Layers | RNN Units | Dataset  (train, validation, test) | Reason for Inclusion |
| --- | --- | --- | --- | --- | --- | --- | --- |
| model005lj\_half | RNN | 1 | 256 | 5 | 128 | Alexa (0%, 50%, 10%)  LJSpeech(50%, 0%, 10%) | Using generalized training data |
| model006lj\_half | RNN | 1 | 128 | 5 | 64 | Alexa (0%, 50%, 10%)  LJSpeech(50%, 0%, 10%) | Using generalized training data |

**KMean Clustered Neural Network**

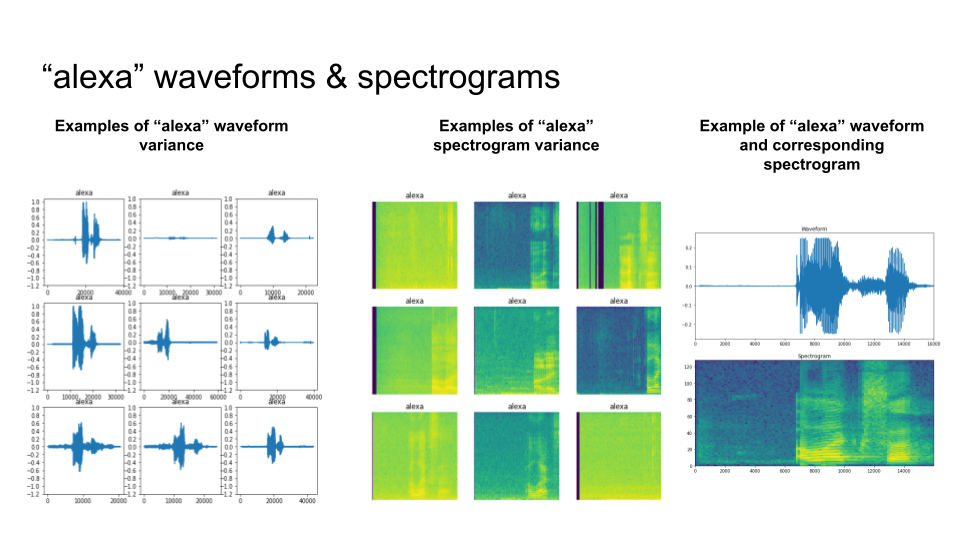
(All RNN neural networks include 2 layers of convolutional neural networks)

| Model Name | Model Type | Hidden Layers | Hidden Units | RNN Layers | RNN Units | Dataset  (train, validation, test) | Reason for Inclusion |
| --- | --- | --- | --- | --- | --- | --- | --- |
| model000km | KMean + LOGIT |  |  |  |  | Alexa (80%,10%,10%)  LJSpeech(0%, 0%, 10%) | Explore KMean models and unsupervised learning |
| model001km | KMean + RNN | 1 | 256 | 1 | 128 | Alexa (80%,10%,10%)  LJSpeech(0%, 0%, 10%) | Explore KMean models and unsupervised learning |
| model002km | KMean + RNN | 1 | 256 | 2 | 128 | Alexa (80%,10%,10%)  LJSpeech(0%, 0%, 10%) | Explore KMean models and unsupervised learning |
| model003km | KMean + RNN | 1 | 256 | 3 | 128 | Alexa (80%,10%,10%)  LJSpeech(0%, 0%, 10%) | Explore KMean models and unsupervised learning |
| model004km | KMean + RNN | 1 | 256 | 4 | 128 | Alexa (80%,10%,10%)  LJSpeech(0%, 0%, 10%) | Explore KMean models and unsupervised learning |
| model005km | KMean + RNN | 1 | 256 | 5 | 128 | Alexa (80%,10%,10%)  LJSpeech(0%, 0%, 10%) | Explore KMean models and unsupervised learning |
| model006km | KMean + RNN | 1 | 256 | 5 | 64 | Alexa (80%,10%,10%)  LJSpeech(0%, 0%, 10%) | Explore KMean models and unsupervised learning |

**Block Diagram**



**Images from Exploratory Data Analysis**

****

**What is considered success/ failure?**

If a model can predict the word ‘Alexa’ with >50% accuracy, then the model will be considered successful.

**Evaluation Parameters (potential)**

We used three accuracy scores to compare and evaluate the models.

(i) A relative accuracy function, where we match the prediction with a true label, character by character. If the two characters match, it is a 1, otherwise 0. Then we calculate the average for each prediction, which becomes the accuracy score for that prediction.

(ii) An absolute accuracy function, where we match the full word/sentence with the true label. If the two match, it is 1. Otherwise 0. This is a much more stringent accuracy function, as the full word has to match with the true label.

(iii) We also used Word Error Rate (wer), a built-in function in the JIWER package as a measure of accuracy.

**Experiments**

For all the models, except one (set of three RNN models), we used Alexa dataset as train, and LJspeech as test.

(ii) The first model we trained is LOGIT Multiclassification. This is our baseline model

(ii) The second model is a set of 3 ANN models with a varying number of hidden layers and activation units.

(iii) The third model is 2D-CNN with 2 hidden layers and 2 layers of convolution.

(iv)The fourth is a set of 6 RNN models with Alexa as training dataset, and both Alexa + LJspeech as test dataset.

(v)The fifth is a set of 2 RNNS models, with a generalized dataset for both training and testing.

(vi)To also explore unsupervised models, we built a sixth model, i.e. Kmean+ LOGIT, and

(vii) a seventh model, i.e. a set of six Kmean + RNN models. Both these unsupervised models used Alexa as train dataset, and Alexa+LJspeech as test dataset.

**Results**

**Basic Models, Logistic and Artificial Neural Networks**

| Model Name | Model Type | Hidden Layers | Hidden Units | Dataset  (train, validation, test) | Accuracy | Abs Accuracy | Word Error Rate |
| --- | --- | --- | --- | --- | --- | --- | --- |
| model000 | Multi-Classification |  |  | Alexa (80%,10%,10%)  LJSpeech(0%, 0%, 10%) | 2.4%  2.5% | 0%  0% | 1.05  1 |
| model001ann | Artificial Neural Network | 1 | 256 | Alexa (80%,10%,10%)  LJSpeech(50%, 0%, 10%) | 0%  0% | 0%  0% | 1  1 |
| model002ann | Artificial Neural Network | 2 | 256, 128 | Alexa (80%,10%,10%)  LJSpeech(50%, 0%, 10%) | 0%  0% | 0%  0% | 1  1 |
| model003ann | Artificial Neural Network | 3 | 256, 128, 64 | Alexa (80%,10%,10%)  LJSpeech(50%, 0%, 10%) | 20%  0.18% | 0%  0% | 1  0.98 |

**Convolutional Neural Network**

| Model Name | Model Type | Dataset  (train, validation, test) | Accuracy | Abs Accuracy | Word Error Rate |
| --- | --- | --- | --- | --- | --- |
| model007 | 2D Convolutional NN | Alexa (80%,10%,10%)  LJSpeech(0%, 0%, 10%) | 51%  0.15% | 0%  0% | 1  1 |

**Recurrent Neural Networks Trained on Alexa Dataset**

(All RNN neural networks include 2 layers of convolutional neural networks)

| Model Name | Model Type | Hidden Layers | Hidden Units | RNN Layers | RNN Units | Dataset  (train, validation, test) | Accuracy | Abs Accuracy | Word Error Rate |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| model001 | RNN | 1 | 256 | 1 | 128 | Alexa (80%,10%,10%)  LJSpeech(0%, 0%, 10%) | 81%  0.3% | 78%  0% | 0.22  1 |
| model002 | RNN | 1 | 256 | 2 | 128 | Alexa (80%,10%,10%)  LJSpeech(0%, 0%, 10%) | 91%  0.56% | 89%  0% | 0.11  1 |
| model003 | RNN | 1 | 256 | 3 | 128 | Alexa (80%,10%,10%)  LJSpeech(0%, 0%, 10%) | 100%  0.56% | 100%  0% | 0  1 |
| model004 | RNN | 1 | 256 | 4 | 128 | Alexa (80%,10%,10%)  LJSpeech(0%, 0%, 10%) | 40%  0.25% | 0%  0% | 1  1 |
| model005 | RNN | 1 | 256 | 5 | 128 | Alexa (80%,10%,10%)  LJSpeech(0%, 0%, 10%) | 100%  0.56% | 100%  0% | 0  1 |
| model006 | RNN | 1 | 128 | 5 | 64 | Alexa (80%,10%,10%)  LJSpeech(0%, 0%, 10%) | 100%  0.56% | 100%  0% | 0  1 |

**Generalized Recurrent Neural Networks Trained on LJ Speech Dataset**

(All RNN neural networks include 2 layers of convolutional neural networks)

| Model Name | Model Type | Hidden Layers | Hidden Units | RNN Layers | RNN Units | Dataset  (train, validation, test) | Accuracy | Abs Accuracy | Word Error Rate |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| model005lj\_half | RNN | 1 | 256 | 5 | 128 | Alexa (0%, 50%, 10%)  LJSpeech(50%, 0%, 10%) | 1.4%  22% | 0%  0.2% | 1.59  0.54 |
| model006lj\_half | RNN | 1 | 128 | 5 | 64 | Alexa (0%, 50%, 10%)  LJSpeech(50%, 0%, 10%) | 0.6%  1.9% | 0%  0% | 1  0.98 |

**KMean Clustered Neural Network**

(All RNN neural networks include 2 layers of convolutional neural networks)

| Model Name | Model Type | Hidden Layers | Hidden Units | RNN Layers | RNN Units | Dataset  (train, validation, test) | Accuracy | Abs Accuracy | Word Error Rate |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| model000km | KMean + LOGIT |  |  |  |  | Alexa (80%,10%,10%)  LJSpeech(0%, 0%, 10%) | 0%  0% | 0%  0% | 1  1 |
| model001km | KMean + RNN | 1 | 256 | 1 | 128 | Alexa (80%,10%,10%)  LJSpeech(0%, 0%, 10%) | 100%  0.6% | 100%  0% | 0  1 |
| model002km | KMean + RNN | 1 | 256 | 2 | 128 | Alexa (80%,10%,10%)  LJSpeech(0%, 0%, 10%) | 100%  0.6% | 100%  0% | 0  1 |
| model003km | KMean + RNN | 1 | 256 | 3 | 128 | Alexa (80%,10%,10%)  LJSpeech(0%, 0%, 10%) | 100%  0.6% | 100%  0% | 0  1 |
| model004km | KMean + RNN | 1 | 256 | 4 | 128 | Alexa (80%,10%,10%)  LJSpeech(0%, 0%, 10%) | 100%  0.6% | 100%  0% | 0  1 |
| model005km | KMean + RNN | 1 | 256 | 5 | 128 | Alexa (80%,10%,10%)  LJSpeech(0%, 0%, 10%) | 100%  0.6% | 100%  0% | 0  1 |
| model006km | KMean + RNN | 1 | 128 | 5 | 64 | Alexa (80%,10%,10%)  LJSpeech(0%, 0%, 10%) | 100%  0.6% | 100%  0% | 0  1 |

**Tests/ Graphs/ Discussions**

**Basic Models, Logistic and Artificial Neural Networks**

The baseline model with an output layer of logistic multiclass classification produces an accuracy of 2.4% using Alexa test dataset and accuracy of 2.5% using LJ Speech dataset. The Alexa specific results are about the same as the generalized LJ Speech dataset.

The artificial neural networks with 1 and 2 hidden layers produce 0% accuracies. 3 layers produce 20% accuracy with Alexa dataset, while only 0.18% with LJ Speech dataset. The models don’t generalize well.

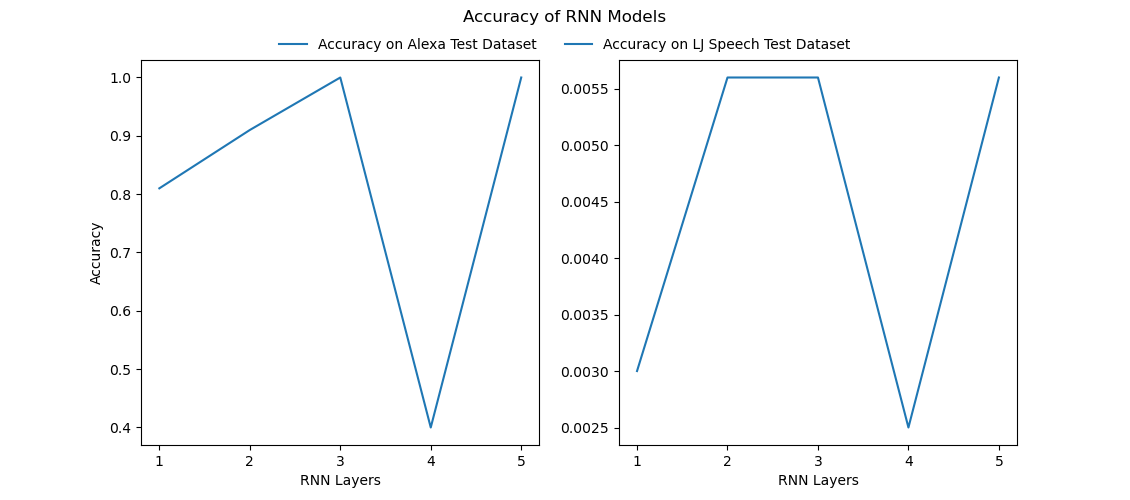
**Convolutional Neural Network**

The convolutional neural network produces a specific accuracy of 51% and generalized accuracy of 0.15%, this model improves specific accuracy but does not improve generalization.

**Recurrent Neural Networks Trained on Alexa Dataset**

Using recurrent neural networks, the specific accuracy gradually increases to 100%, while generalized accuracy plateaus at around 0.56%. There is a curious dip for the recurrent neural network models with 4 RNN layers.

| Label | Predictions |
| --- | --- |
| alexa | ala |
| alexa | ala |
| alexa | ala |
| alexa | ala |
| alexa | ala |



**Generalized Recurrent Neural Networks Trained on LJ Speech Dataset**

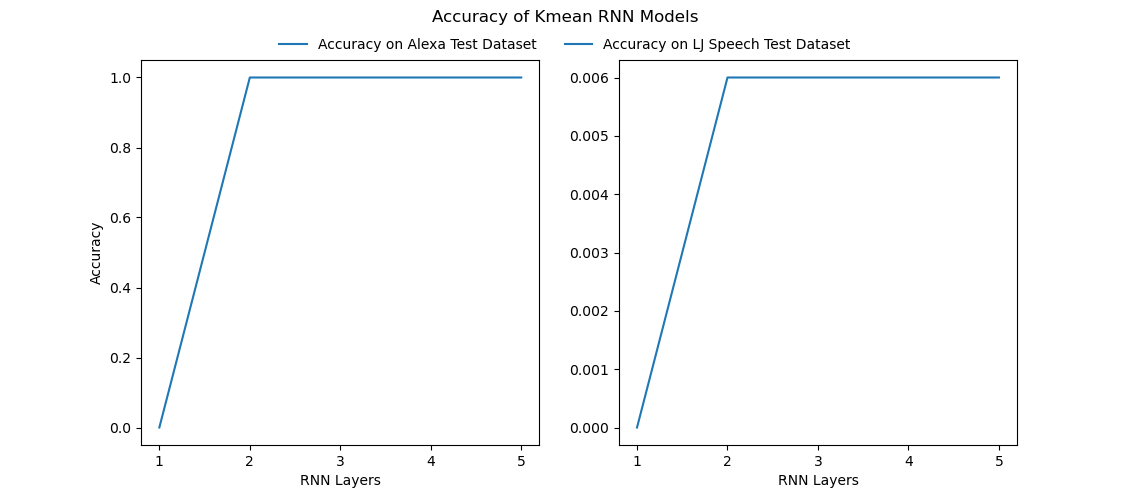
The LJ speech dataset trained recurrent neural networks generalize much better, at around 22%. Specific to the Alexa dataset, the accuracy is around 1.4%.

**KMean Clustered Neural Network**

KMean clustered models reach 100% accuracy on Alexa dataset quickly.

KMean clustered models seem to overfit on the Alexa dataset and do not generalize at all, below are select test predictions from “model005km”. Every sentence is classified as “Alexa”.

| Label | Predictions |
| --- | --- |
| concern that the vice president might also be ... | alexa |
| the two men on guard had gone off immediately ... | alexa |
| though a great warrior it was not for his mili... | alexa |
| was probably made within a day or a day and a ... | alexa |
| he said that he had simply wanted to discuss t... | alexa |



**Constraints**

1. The availability of datasets was a constraint. We used two datasets: one with only one word (Alexa) but in different voices and accents, and the other dataset with just one voice, but different words.
2. We also had a computation limitation in this project. Due to this, we couldn't train a full dataset, train more models, or train our models with a general speech.
3. We also had to do a lot data engineering in this ASR problem, as we had to convert audio files to spectrogram and then to data features before we could build our models

**Standards**

numpy==1.20.2

pandas==1.2.4

tensorflow==2.9.1

scikit-learn==0.24.2

**Comparison**

1. The baseline logistic multi-classification model produces low accuracy compared to other models, as expected.
2. The artificial neural network models compare poorly with models with recurrent layers, due to the absence of long and short term memory needed for automated speech recognition.
3. The convolutional model helps with comparing data with variations in formats. In the context of automated speech recognition, it reduces the effect of temporal spaces in the audio samples, thus performing much better than the baseline model.
4. The recurrent neural network models perform best on the Alexa dataset, because it’s able to account for the information embedded in the preceding sounds in the sample through the use of long short term memory layers.
5. The recurrent neural network models trained on generalized LJ Speech dataset performs much better in generalization evaluation compared to models trained on Alexa dataset, but at the expense of reduced accuracy on Alexa test dataset.
6. The KMean cluster models perform very well on the Alexa dataset, but do not generalize, due to clustering that reduces each sound sample in the spectrogram to a cluster assignment, resulting in overfitting.

**Limitations of the Study**

1. Limited generalizations with diverse dataset.
2. Small training dataset used due to computational resource limitations.
3. Potential combinations of model configurations not yet tried (more RNN layers, different Kmean cluster numbers).

**Fairness and Bias**

An important bias issue with ASR models with the lack of inclusion and diversity in the voices and accents of training data, which can lead to a model that is not representative of the general population, i.e. the model will not accurately predict the spoken words with accents and voices of certain sections of the population, that are usually underrepresented in the available data used for training.

For our project, we faced a similar problem regarding availability of diverse and large datasets. We used the LJ Speech dataset as it is large, but we have the same concern regarding bias. Though it is a large dataset, but consists of only a single voice and accent. The second dataset (Alexa dataset) has a single word (Alexa) in many different voices and accents, and therefore is more representative, but as the size of the dataset is small.

**Future Work**

1. Train models with combined Alexa and LJ speech dataset.
2. Train model using Librispeech dataset.
3. Tran model using common speech dataset to explore effect of diverse accents and audio qualities.
4. Try additional model configurations for Recurrent neural networks and Kmean clusters to further explore their effects.