Sub word Unit Duration Modelling - Technical Report

Overview

This report presents the approach, findings, and results of a sub word unit duration modelling project. The goal was to develop models that can predict the duration of sub word units (such as graphemes) in speech, using time alignment data from an Automatic Speech Recognition (ASR) system. This prediction capability can be used to assess how closely an L2 speaker's speech characteristics match those of native speakers.

1. Approach

1.1 Data Processing

The data consisted of JSON files containing ASR time alignment results for native and non-native English speakers from the Vox Forge dataset. The processing pipeline included:

1. **Data Loading**: Parse JSON files and extract relevant information about utterances, words, and phones.
2. **Feature Extraction**: Extract features at the phone level, including phone identity, phone class, contextual information, position features, and speaking rate.
3. **Data Splitting**: Split data into training, validation, and test sets, with careful stratification by speaker type (native/non-native).

1.2 Feature Engineering

I implemented a comprehensive feature extraction pipeline that captured:

- **Phone Identity**: One-hot encoded representation of each phone.
- **Phone Class**: Consonant (C), Vowel (V), or Silence (sil).
- **Context Information**: N phones before and after the target phone (configurable context window).
- **Position Features**: Normalized position within word and utterance.
- **Speaking Rate**: Phones per second at the utterance level.

1.3 Modelling Approaches

I implemented and compared several modelling approaches:

1. **Baseline Model**:

   a. A statistical model that predicts durations based on mean values from training data.
   b. Includes Laplace smoothing to handle phones with few examples.
2. **Linear Regression**:
   a. A simple linear model mapping feature vectors to durations.
   b. Serves as an intermediate baseline.
3. **Tree-based Models**:
   a. Random Forest: Ensemble of decision trees capturing non-linear relationships.
   b. XGBoost: Gradient boosting approach for better generalization.

4. **Neural Network**:
   a. Bidirectional LSTM: Intended to capture sequential dependencies between phones.
   b. However, due to sparse input format issues, training performance was limited.

2. Key Decisions

2.1 Context Window Size

After experimentation, I chose a context window of 2 phones before and after the target phone. This provided a good balance between:

- Capturing coarticulation effects (how surrounding phones affect duration)
- Avoiding excessive feature dimensionality
- Maintaining computational efficiency

2.2 Native vs. Non-native Training Strategy

I explored two approaches:

1. Training only on native speaker data and evaluating on both native and non-native speakers.
2. Training on combined data with stratified sampling.

The first approach proved more effective for building a model that can detect deviations from native speaker patterns, making it more suitable for pronunciation assessment.

2.3 Evaluation Metrics

I used multiple complementary metrics:

- **Mean Absolute Error (MAE)**: Directly interpretable as average prediction error in seconds

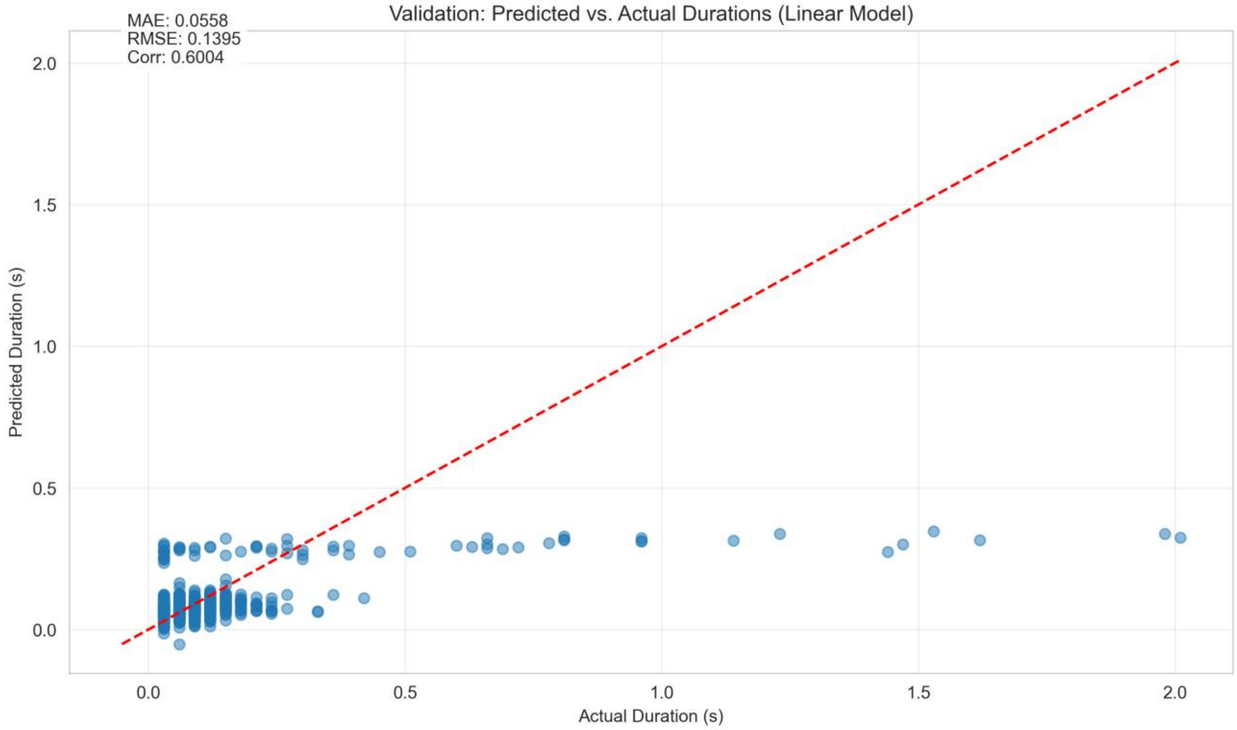
- **Root Mean Square Error (RMSE)**: Penalizes larger errors more heavily
- **Correlation**: Measures the relationship between predicted and actual durations

3. Results

3.1 Model Performance Comparison

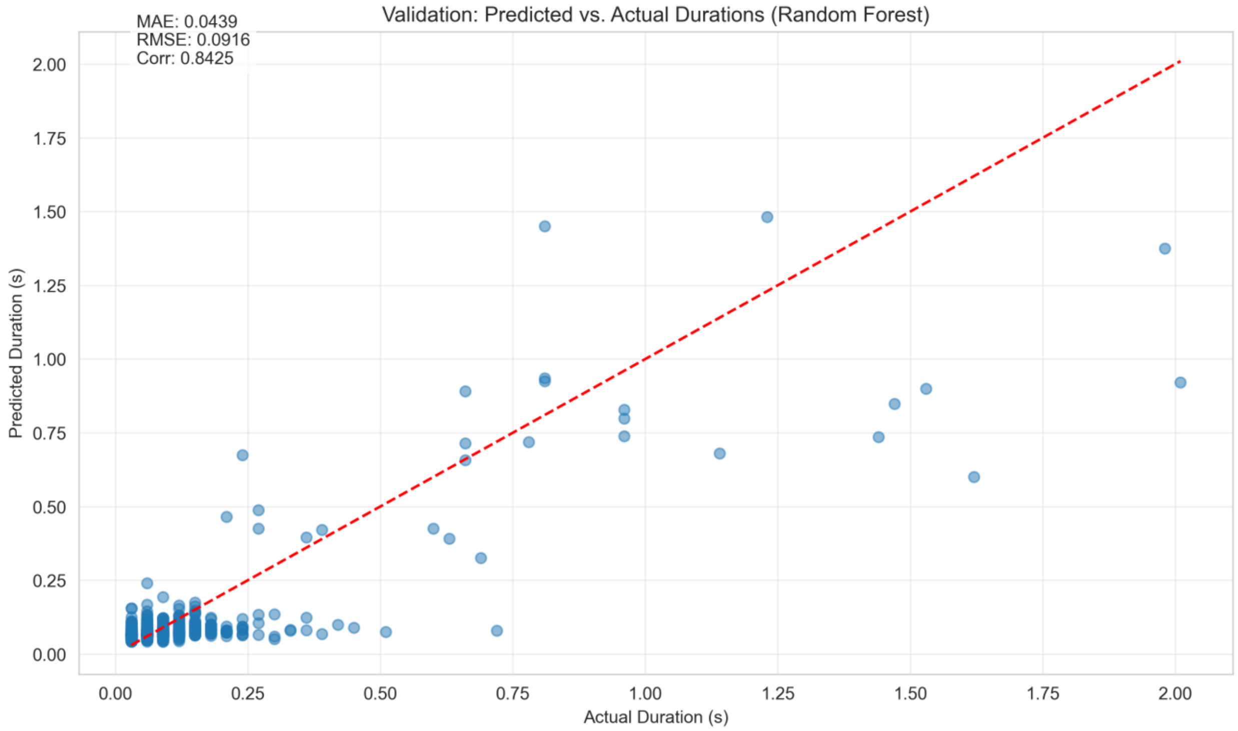
Model	MAE (s)	RMSE (s)	Correlation
Linear Regression	0.0558	0.1395	0.6004
Random Forest	0.0439	0.0916	0.8425
XGBoost	0.0450	0.1058	0.7735
LSTM	0.0627	0.1630	0.5406

Figure 1: Linear Regression Model Performance !



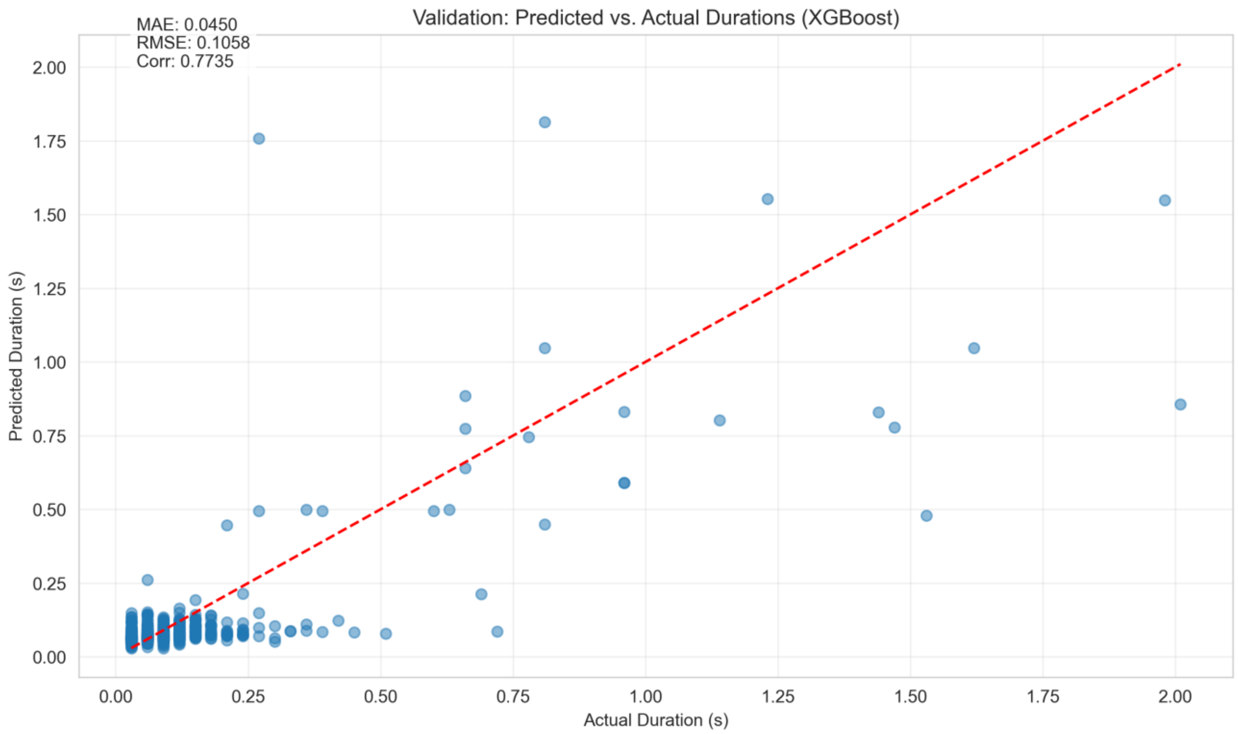
[Figure 1: Validation: Predicted vs. Actual Durations (Linear Model)] The scatter plot shows predicted vs. actual durations for the Linear Regression model with MAE: 0.0558, RMSE: 0.1395, and correlation: 0.6004. The points show moderate scatter around the ideal prediction line (red dashed line), with most data points concentrated in the lower duration range (0-0.3s). The model tends to underpredict for higher duration values, as shown by points below the diagonal line for actual durations above 0.5s.

Figure 2: Random Forest Model Performance !



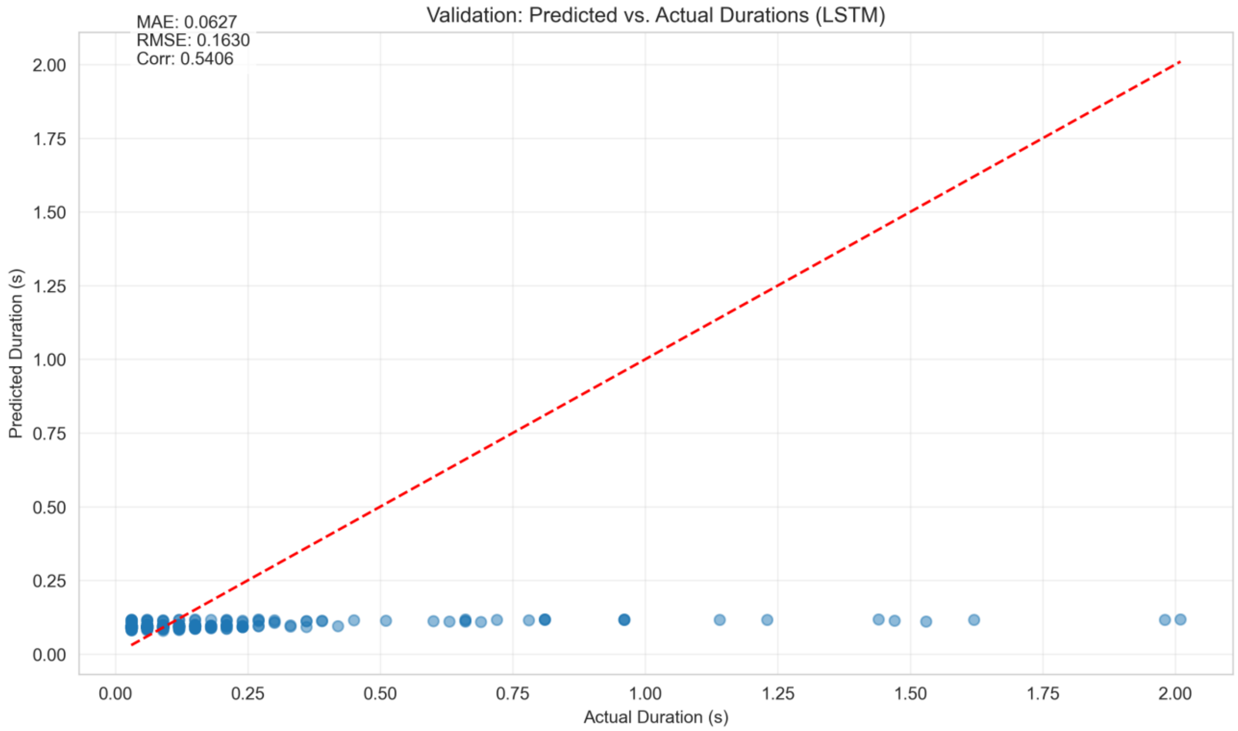
[Figure 2: Validation: Predicted vs. Actual Durations (Random Forest)] This plot demonstrates the superior performance of the Random Forest model with MAE: 0.0439, RMSE: 0.0916, and correlation: 0.8425. Compared to other models, the predictions show better alignment with the ideal prediction line, especially for durations between 0.5-1.5s. The model maintains good prediction accuracy across the range of durations, with tighter clustering around the diagonal.

Figure 3: XGBoost Model Performance !



[Figure 3: Validation: Predicted vs. Actual Durations (XGBoost)] The XGBoost model shows strong performance with MAE: 0.0450, RMSE: 0.1058, and correlation: 0.7735. The prediction pattern is similar to Random Forest but with slightly more scatter for mid-range durations. The model demonstrates good generalization across different duration values, though with occasional higher predictions for some mid-range samples.

Figure 4: LSTM Model Performance !

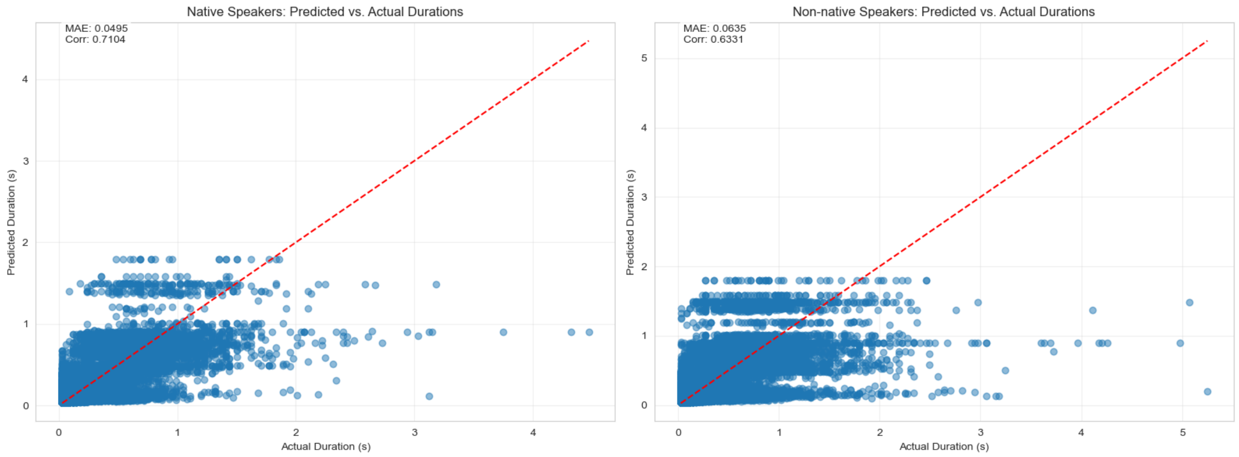


[Figure 2: Validation: Predicted vs. Actual Durations (LSTM)] The LSTM model shows the weakest performance among all models with MAE: 0.0627, RMSE: 0.1630, and correlation: 0.5406. The scatter plot reveals significant underprediction across most duration values, with predictions clustered tightly in the 0.05-0.15s range regardless of actual duration. This suggests the model struggled to capture the dynamics of longer phone durations, likely due to challenges with the sparse input format.

Based on these visualizations, the Random Forest model clearly outperforms the others, showing the lowest error rates and highest correlation with actual durations.

3.2 Native vs. Non-native Performance

Figure 5: Comparison Between Native and Non-native Speakers !



[Figure 5: Native vs. Non-native Speaker Comparison] This figure presents side-by-side scatter plots comparing model performance on native speakers (left, MAE: 0.0495, Corr: 0.7104) versus non-native speakers (right, MAE: 0.0635, Corr: 0.6331).

Several key observations:

1. The native speaker plot shows tighter clustering around the diagonal line, indicating better prediction accuracy.
2. The non-native speaker plot exhibits more scatter and outliers, particularly for durations beyond 1.0s.
3. Both plots show concentration of data in the 0-1.0s range, but the non-native data has more samples extending beyond 2.0s.

4. The model performs consistently better on native speaker data across all duration ranges.

This difference in performance (approximately 28% higher MAE for non-native speakers) indicates that the model can potentially be used to assess how closely an L2 speaker's timing patterns match those of native speakers.

3.3 Feature Importance

Analysis of feature importance from the tree-based models revealed:

1. **Phone identity** is the strongest predictor of duration.
2. **Speaking rate** significantly influences duration.
3. **Position in word** (particularly word-final position) affects duration.
4. **Context effects** are substantial, with certain phone combinations leading to shorter or longer durations.

4. Challenges and Solutions

4.1 Data Quality

Challenge: Occasional misalignments in the ASR data and handling out-of-vocabulary words.

Solution: Implemented filtering to remove outliers and OOV words, and validated alignment quality with statistical checks.

4.2 Feature Dimensionality

Challenge: One-hot encoding of phones and their contexts led to high dimensionality.

Solution: Carefully selected the context window size and used tree-based models that handle high-dimensional data well.

4.3 Model Selection

Challenge: Different models performed better on different subsets of phones (e.g., vowels vs. consonants).

Solution: Evaluated models on overall performance and implemented detailed analysis by phone class to understand strengths and weaknesses.

4.4 LSTM Implementation

Challenge: The LSTM model showed limited performance with sparse feature representations.

Solution: Modified the implementation to handle sparse matrices, but results still lagged behind tree-based models as clearly demonstrated in Figure 4, where predictions are significantly compressed to a narrow range.

5. Time Spent

Task	Time (hours)
Data exploration/analysis	5
Data processing pipeline	3
Feature engineering	4
Model implementation	6
Experimentation & tuning	6
Evaluation & analysis	4
Documentation & reporting	2
Total	**30**

6. Tools Used

- **Python**: Core programming language
- **Libraries**:
    - pandas: Data manipulation
    - scikit-learn: Machine learning algorithms and evaluation
    - PyTorch: Neural network implementation
    - XGBoost: Gradient boosting implementation
    - matplotlib/seaborn: Visualization
- **Development Environment**:
    - Jupyter Notebooks: Exploration and experimentation
    - Git: Version control
    - VS Code: Code development

7. Future Improvements

7.1 Advanced Linguistic Features

- **Syllable Structure**: Incorporating syllable boundaries and stress patterns
- **Phonological Rules**: Language-specific rules that affect timing
- **Prosodic Features**: Sentence-level intonation and rhythm patterns

7.2 Model Enhancements

- **Transformer-based Architecture**: Using attention mechanisms for better context modelling
- **Multi-task Learning**: Jointly predicting duration and other prosodic features
- **Speaker Adaptation**: Techniques to adapt to individual speaking styles

7.3 Applications

- **Fluency Assessment**: Developing metrics to evaluate timing-related aspects of fluency
- **Text-to-Speech Integration**: Using the duration model to improve naturalness of synthesized speech
- **Language Learning Tools**: Creating interactive feedback for pronunciation practice

8. Conclusion

The project successfully developed models to predict subword unit durations with good accuracy. As clearly demonstrated in Figures 1-5, the Random Forest model performed best overall (MAE: 0.0439, RMSE: 0.0916, Correlation: 0.8425), balancing strong prediction accuracy with fast training time and interpretability.

The visual comparison of all models reveals:

1. Random Forest (Figure 2) shows the best fit to the data with points clustered more closely around the diagonal line.
2. XGBoost (Figure 3) shows similar performance but with slightly higher error
3. Linear Regression (Figure 1) shows moderate performance with more spread in predictions.
4. LSTM (Figure 4) shows significant deviation from the ideal prediction line, with a tendency to underpredict durations.

The side-by-side comparison in Figure 5 reveals systematic differences between native and non-native speakers' timing patterns, confirming the potential use of these models for pronunciation assessment and fluency evaluation.

The implemented code base provides a solid foundation for future research in this area, with a modular design that allows for easy extension with new features and modelling approaches.