Amazon Intent Natural Language Classification Project

Date: December 22nd, 2023

Authors: Boom Devahastin Na Ayudhya, Kevin Xie, Lance Lepelstat, Muaaz Noor

Penn Engineering

Motivation and Context

- Classifying user intent has multitudinous applications for problems across industries.
  - Consumer Banking: automated chatbots take in queries from user, return number of suggested self-help links via classification of issues based on certain phrases in source
  - Airlines: customers call to inquire about flight information and expect right response to execute demands

- More broadly, virtual assistants such as Amazon Alexa are tasked with parsing language to then classify intentions in order to elicit the appropriate response.

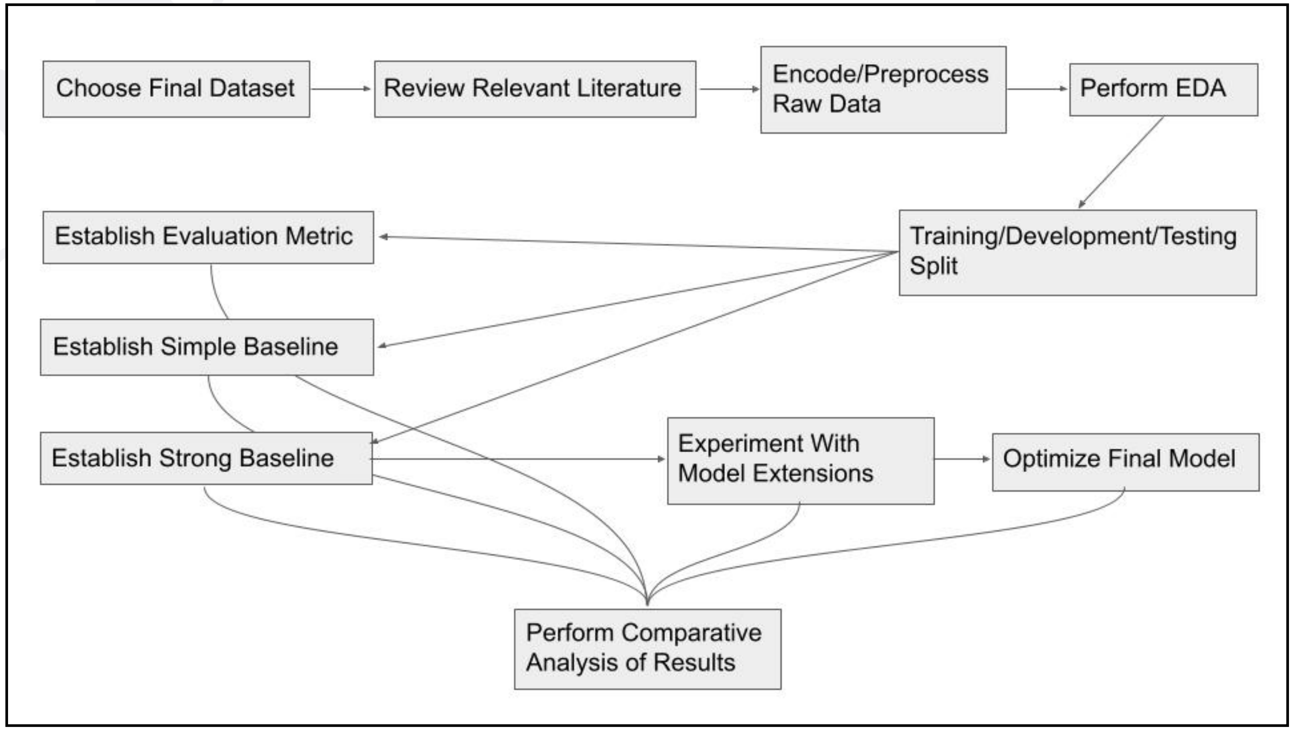
Research Goal

<u>Goal</u>: To build a multi-classification NLP model that ingests input from a user and outputs a probabilistic prediction of each of the class labels that message/intention falls into.

Data Overview and Project Roadmap

Dataset Overview:

- Amazon Alexa Intent Classification dataset publicly available
- User input sequences stored as text
- 60 labels for defining the task associated with the input sequence
    - Can be further classified into 17 parent classes



Penn Engineering

Literature Review and Dataset Overview

Basic Parametric Models:

- Naive Bayes Classifier
- Support Vector Machines

Deep Learning Models:

- Basic LSTM implementation
- Attention based RNN

Hierarchical Modeling:

- The idea of dividing the target labels into subclasses to create a parent-child labels

Evaluation Metric, Simple Baseline, Strong Baseline

Evaluation Metric:

- Total accuracy is used to evaluate a model
- This metric has been widely used and accepted in multi-class intent classification problems

Simple Baseline:

- Majority Class Classifier was used as the first simple baseline
  - Resulted in **7.03%** accuracy which is too trivial
- KNN (K=1) Classifier was explored as a better alternative
  - Cosine similarity used to measure distance between data points
  - Resulted in **74.65%** accuracy

Strong Baseline:

- Bi-directional LSTM with Glove embeddings used
  - The embeddings of the input sequence passed into the LSTM layer
  - The output of the LSTM passed into a fully connected layer that outputs class probabilities
- Resulted in an accuracy of **85.58%**

Extension: Hierarchical Modeling Tree

Architecture:

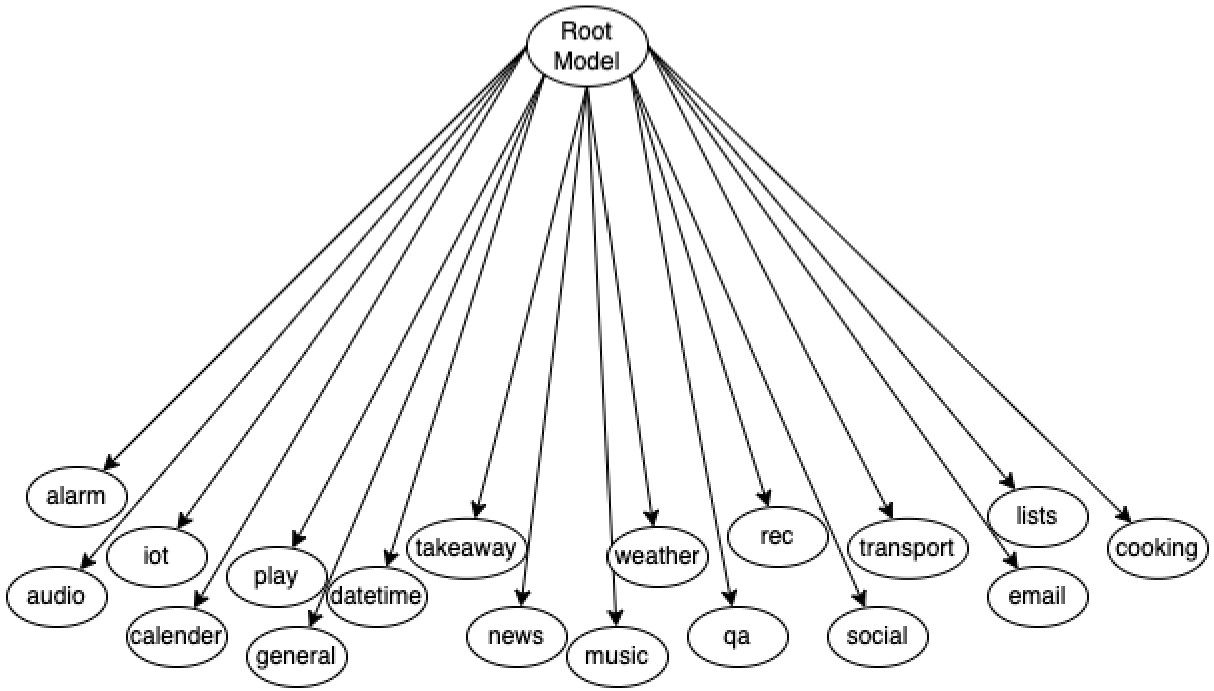
- One root model
- One child node for each model class
- Each child model trained on relevant data
- Each model node using an LSTM

Best Hyper-parameters:

- Learning Rate = **0.01**
- Parent Batch Size = **128**
- Child Batch Size = **64**

Performance:

- Accuracy = **0.8551**
- F1 Score = **0.8634**



Model Architecture

Penn Engineering

Extension: Parent-Child Prediction using LSTM

Parent/child class:

- Child: `audio_volume_down`
- Parent: `audio`

Architecture:

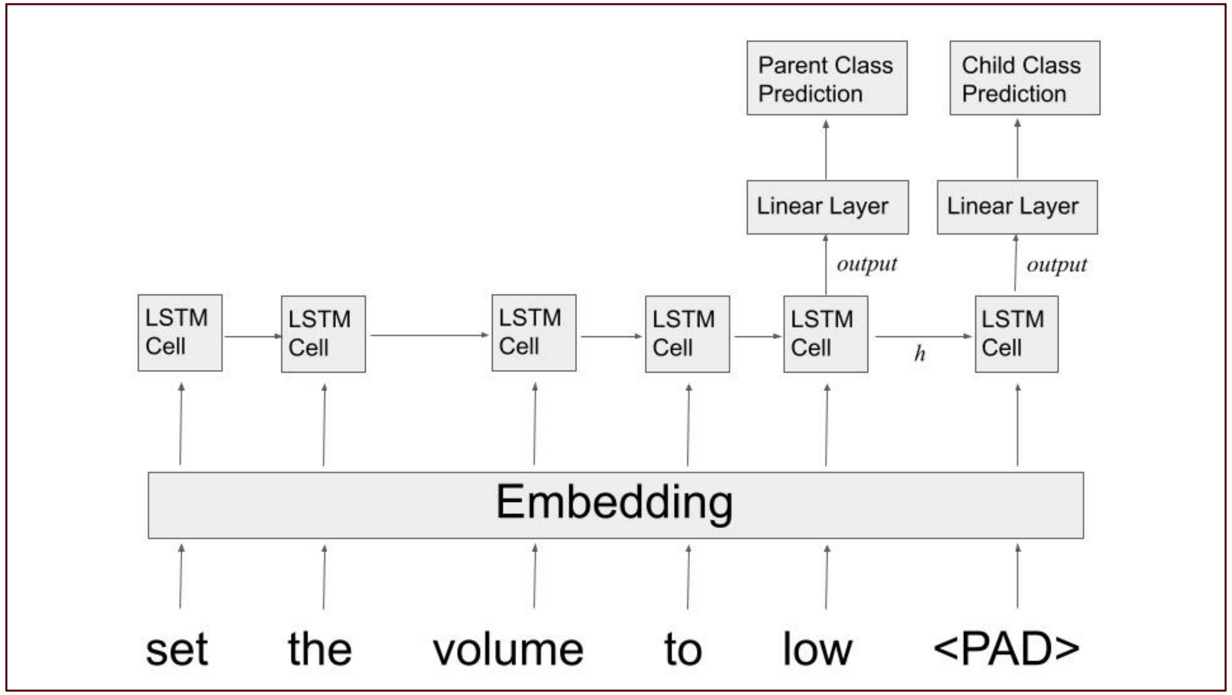
- Additional <PAD>
- Final hidden state passed to second LSTM
- Two predictions/loss computations

Best Hyper-parameters:

- Learning Rate = **0.01**
- Batch Size = **128**

Performance:

- Accuracy = **0.8697**
- F1 Score = **0.8681**



Model Architecture

Error Analysis

Top Classes with Most Misclassification:

- `general_quirky`: random requests
- `qa_factoid`: requests for factual information

Major Error Categories:

I. Slang and Faux Word Confusion
   - "`wakey wakey eggs and bakey`" ⟶ recipe request

II. Oversensitivity/Overfitting to Food-Themed Words
   - "`are jello shots calorie free`" ⟶ recipe request

III. Short/Vague Commands
   - "`sports`"
   - "`none`"

Conclusion

I. Successful experiment

II. Best performance: Parent-Child Prediction using LSTM

III. Major error categories identified

IV. Further exploration
   - Continued hyper-parameter tuning
   - New data