**Model & Training Configurations.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Models | Algorithm | Model Configurations | Training time(mins) | Hardware used |
| Baseline 1 | Greedy Algorihm | LSTM for encoder and decoder, hidden\_size 256, teacher\_forcing\_ratio = 1,  Optimizer: Adam  dropout rate 0.1,  maximum length 150. | 84 | Google Colab |
| Baseline 2 | Greedy Algorihm | LSTM for encoder and decoder, hidden\_size 256, teacher\_forcing\_ratio = 1,  Optimizer: Adam  dropout rate 0.1,  maximum length 150. | 126 | Google Colab |
| Extension 1 | Greedy Algorihm | LSTM for encoder and decoder, hidden\_size 256, teacher\_forcing\_ratio = 1,  Optimizer: Adam  dropout rate 0.1,  maximum length 150. | 102 | Google Colab |
| Extension 2 | Greedy Algorihm | LSTM for encoder and decoder, hidden\_size 300, teacher\_forcing\_ratio = 1,  Optimizer: Adam  dropout rate 0.1,  maximum length 150. | 115 | Google Colab |

**Data Statistics**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Models | Vocab size | Avg. size | Max size | Min size |
| Ingredients | 15154 | 42.57 | 148 | 1 |
| Recipe | 29059 | 71.41 | 149 | 1 |

For each model, only 10000 pairs(around 1%) of ingredients and recipes were used for training due to computational resource limitations.

**Data Preprocessing**

Data has been pre-processed and pruned in different ways depending on the model that was trained. The baseline models lowercased and removed all special characters(eg ;, \t) from the ingredients list. The pre-processing methods used for the extended models are based on the pre-processing methods and ideas used by in the optional readings by (Yinhong Liu, Yixuan Su et al.) and (Ximing Lu, Peter West et al.), namely, pruning off recipes with more than 15 sentences or under 3 words and removing non-noun words in the ingredients list, so that model will be mostly trained using actual ingredients.

**Analysis**

Orange: Validation loss

Blue: Training Loss

1. Baseline 1
2. Baseline 2
3. Extended 1
4. Extended 2

A screen shot of a graph

Description automatically generatedA screenshot of a graph

Description automatically generated

(1) (2)

A screenshot of a graph

Description automatically generatedA graph of a train loss

Description automatically generated with medium confidence

(3) (4)

Baseline 1 seems to have the lowest validation loss, although it is not consistent throughout, whereas Extended 4 has a more consistently lower validation loss, possible due to its pre trained embeddings. All 4 models perform similarly: 3.9 – 4.1 range for training error and mostly above 7 in the test. A very low training score showing signs of underfitting, as well as overfitting as the validation plots show.

**Quantitative Evaluation**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | BLEU-4 | METEOR | Avg % of given items | Avg. extra items |
| Baseline 1 | 0.0041132442216330395 | 0.08874832028403982 | 0.1945295752213848 | 19.854755784061698 |
| Baseline 2 | 0.002678930944107098 | 0.10677849234136354 | 0.22861709633913505 | 27.08611825192802 |
| Extension 1 | 0.0025616172274081765 | 0.09578565340288475 | 0.2165560865369011 | 22.89974293059126 |
| Extension 2 | 0.001687846941083022 | 0.08897064441860937 | 0.1910007629729495 | 77.04884318766067 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | BLEU-4 | METEOR | Avg % of given items | Avg. extra items |
| Gold v sample | 0.11770400167201682 | 0.5736654804270463 | 0.7721518987341772 | 60 |

All the model results perform similarly to each other, and not well for any metric(compared to the gold v sample) due to limited training. The models may also need more hidden layers for the encoder and decoder, as they try to predict with limited information.

**Qualitative Evaluation**

|  |  |  |  |
| --- | --- | --- | --- |
| **Ingredients**: 2 c sugar, 1/4 c lemon juice, 1 c water, 1/3 c orange juice, 8 c strawberries | | | |
| Baseline 1 | Baseline 2 | Extended 1 | Extended 2 |
| in a large bowl combine all ingredients in a large bowl combine all ingredients except the flour and salt and pepper to taste and serve immediately <EOS> | in a large bowl combine flour and salt in a bowl combine flour and salt and pepper to taste and chill until firm about 1 hour or until firm <EOS> | preheat oven to 350 degrees f combine all ingredients in a large bowl combine flour baking powder and salt add eggs and vanilla and mix well pour over mixture and bake at 350 degrees for 30 minutes or until golden brown <EOS> | preheat oven to 350 degrees f combine all ingredients in a bowl and blend until smooth add eggs and vanilla stir in flour and vanilla stir in flour and vanilla stir in flour and vanilla stir in flour and vanilla stir in flour and vanilla stir in flour and vanilla stir in flour and vanilla stir in flour and vanilla stir in flour and vanilla stir in flour and vanilla stir in flour and vanilla stir in flour and vanilla stir in flour and vanilla stir in nuts and vanilla pour into prepared pan bake in preheated oven for 25 minutes or until golden brown <EOS> |

Likely due to lack of training examples, none of the 4 models have predicted a recipe relevant to the ingredients chosen. All of them hallucinate ingredients that do not exist. The last model(trained using word2vec embeddings) in particular constantly repeats the same instructions till the maximum output is reached. All 4 models predict ‘combine all ingredients’, presumably the most common instruction in these recipes. All of this is consistant with the low scores given to them by the quantitative metrics

In the future, it would probably be more effective to train each individual ingredient with specific sentences that explain how to use them, with similar words being used as a substitute. This would require accounting for non-ingredient words in the ingredients training list(eg. Brand names, outliers, etc.). Also will require more computational units for more trainning.

**Note: ChatGPT was used in multiple section of the code**