FIT5217 Assignment 2 Report
Student ID: 31237223

Paper Implementation: NN

The model proposed in (Kiddon et al.) is a KNN model which utilises the cosine similarity between input ingredients and our training data for recipe generation. More specifically, the KNN model is first trained on the tensor representations of the ingredients list within our training data (obtained using tensorFromSentence) after padding is performed to ensure equal tensor size. Then, unseen data (test data ingredients) will be fed to the trained model after performing the same tensor conversion and preprocessing to identify the closest match between our input ingredients list and the training data based on cosine similarity. Once identified, the recipe of the identified train ingredients will simply be copied over and treated as the generated recipe for our input.

2.1 Model & Training Configurations

- All models are trained with the same parameters:
  - Hidden_size = 256,Teacher Forcing Ratio = 1,Optimizer = Adam,Dropout rate = 0.1,Maximum_length = 150,Training Iterations = 10000, Default decoding algorithm (same as tutorial code)
- All models are trained locally using my personal PC.
  - CPU = Ryzen 5 7600X ,RAM = 32GB DDR5 ,GPU = RTX 3060 12GB

Model Name	**Training Times**
Baseline 1: RNN without Attention	**30m 44s**
Baseline 2: RNN with Attention	**43m 8s**
Extension 1: Data Preprocessing	**29m 36s**
Extension 2: Additional Layers + Data Preprocessing	**32m 9s**
Paper Implementation: NN (Kiddon et al.)	**N/A since KNN trains near instantly**

2.2 Data Statistics (After preprocessing)

	Samples	**Vocab Size**	**Min Length**	**Max Length**	**Average Length**
Ingredients (Training)	79894	16481 (Shared Vocab)	1	150	46.05
Ingredients (Testing)	629		1	145	44.15
Ingredients (Dev)	644		2	149	44.25
Recipe (Training)	79894	31389 (Shared Vocab)	1	150	76.88
Recipe (Testing)	629		3	149	76.0
Recipe (Dev)	644		3	150	76.84

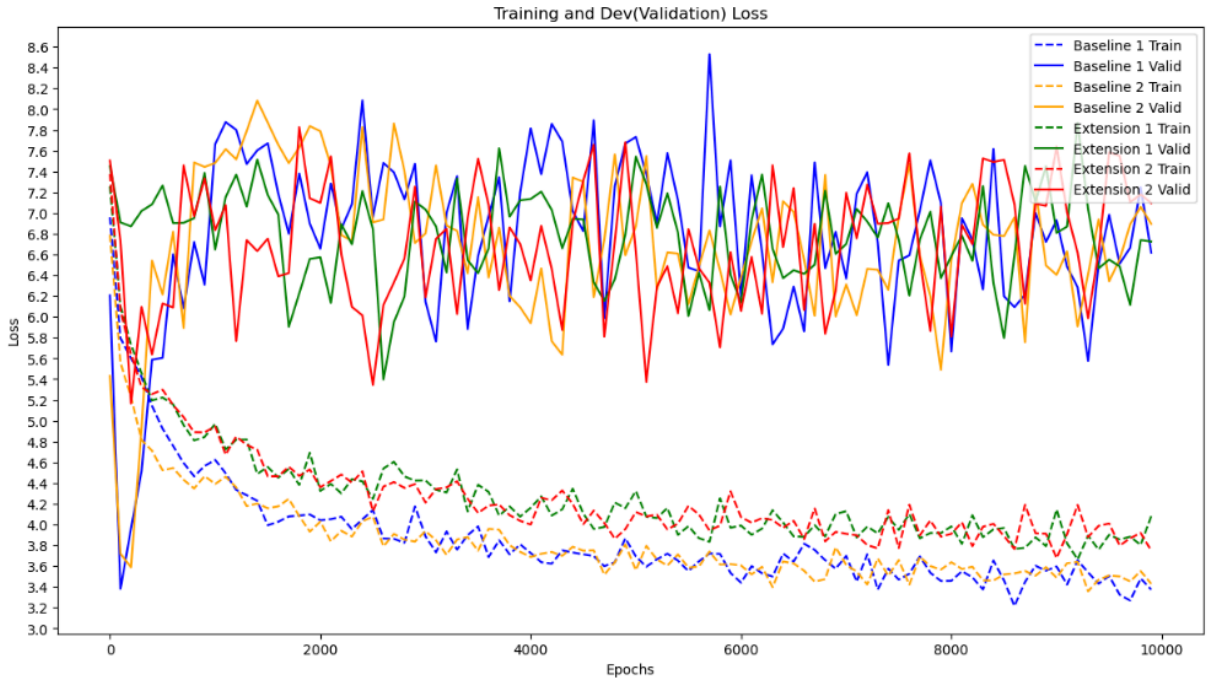
The entire dataset after filtering out data larger than MAX_LENGTH was used. With the total number being **79894 / 101340 (78.8%)**.

FIT5217 Assignment 2 Report
Student ID: 31237223

2.3: Data Preprocessing

Data is first normalised using the method provided in tutorial code. Then, they are filtered depending on if they are within the MAX_LENGTH of 150. For the extension tasks, the data is further preprocessed to remove units of measurements, numerical amounts of items, words within brackets (which may unwanted information such as advertisements), rrb and llb words (which is used to signify brackets), remove punctuations and whitespaces, as well as further removing data-specific words and other stopwords that are irrelevant (such as “lb” which is a token that appears repeatedly but has no significant meaning”).

2.4: Analysis



Testing loss for models evaluated on testing set upon train completion			
Baseline 1	**Baseline 2**	**Extension 1**	**Extension 2**
9.683302558430283	8.799383153368279	6.101528200040006	6.332643612563667

(Paper implementation omitted since unable to extract training information from NN model)

Train Performance: Throughout all four models, we can observe that they experience a general decrease in train loss which is indicative that they are all learning the train data effectively. Both baseline models outperform the extension models in terms of train loss, which is reasonable since both models are inherently simple in design which means that they do not capture much noise thus allowing them to better fit the training data. On the other hand, Extension 2 has the worse train performance since the increase in model complexity through more layers may have caused it to learn noise instead of useful patterns

Valid Performance: The valid performance for all models are incredibly inconsistent as we can see from the large fluctuations within the plot above. This seems to suggest that the models all fail to generalise well, with both Baseline models showing signs of overfitting around 4000 -> 8000 epochs due to the increase in distance between their valid loss and train loss lines. Both extension tasks have lower gaps between their train and valid loss lines which suggests that while overfitting is still a problem (due to the high valid loss in comparison with train loss), they are both better in comparison with the two baseline models.

FIT5217 Assignment 2 Report
Student ID: 31237223

The best performing one seems to be Extension 1, with lesser spikes within its valid loss line which suggests lesser variance and a smoother convergence towards a useful model.
Test Performance: Test performance shows a better picture of how well the models generalise and it backs up our argument in the past two paragraphs that both extensions outperform the baselines greatly, with Extension 1 being the best performing model since it takes advantage of the preprocessed dataset while maintaining a balanced model complexity as opposed to the added layers in Extension 2.

2.5 Quantitative Evaluation

Model	BLEU-4	METEOR	Avg. % given items	Avg. extra items
Baseline 1	8.568198371043781e-157	0.05657369817687231	6.050104188801186	3.0937996820349762
Baseline 2	9.84120905257055e-157	0.05163769209649932	5.437713077576711	2.6931637519872815
Extension 1	0.009897398610171422	0.15836779878989902	16.677025479580884	0.4069952305246423
Extension 2	0.0058839062975395914	0.15643694855863724	14.970740794421761	0.5596184419713831
NN	0.010469612423152619	0.13425344966291148	8.437269669720312	4.5532591414944354

- BLEU-4 measures the similarity between the reference (test data recipe) and generated responses (generated recipe). The table shows that we were able to achieve a significant increase with both extension tasks which suggests that the extensions such as data preprocessing helped the model to create a recipe more similar to the expected output.
- METEOR is also a similar metric that measures generation quality is much more complex than BLEU-4 since it also considers factors such as linguistic features, word order etc. As expected, the NN performed well in this metric since it only copies over the recipe of another entry with a similar ingredient list so the generated recipe would be of certain quality. Similar trends of improvement for both extension models compared to baselines can also be seen here since both baselines are much more prone to repeating words (which we will discuss in the qualitative evaluation).
- Again, it seems like both extensions as well as the NN model were better at identifying items from the ingredients list and use it within the generated recipe which is unsurprising since the data processing method applied here was able to remove almost all irrelevant words except for actual ingredient items.
- Both extensions show signs of improvements here again as they significantly outperform the baselines due to their increased ability to properly identify the ingredients. However, I was surprised to find that NN had the most extra items used, since we are using KNN to match the ingredients list so we would theoretically be able to obtain a relevant recipe list too. In the future, I hope to improve the KNN model and also find a better representation than the tensors obtained from TensorFromSentence so the generated recipe is more relevant.

FIT5217 Assignment 2 Report
Student ID: 31237223

2.5 Gold vs Sample

Model	BLEU-4	METEOR	Avg. % given items	Avg. extra items
Gold v Sample	0.5094302391481254	0.4689584138019828 3	100.0	2

2.6 Qualitative Evaluation

Ingredients: 2 c sugar, 1/4 c lemon juice, 1 c water, 1/3 c orange juice, 8 c strawberries				
B1	B2	E1	E2	Paper
fold excess pudding over extract cookie crumbs blend gradually while nonmetallic blend logs cocoa boiling comes blend to rise minutes glaze scraping which ingredients made batte mouth blend <EOS>	flour fruit oil hershey if chocolate margarine coated flour fruit oil rolling dipping while ending chocolate blend hot margarine four if four if and four blend souffle scraping slightly ingredients blend hot margarine four if four if and four blend souffle scraping slightly ingredients blend hot margarine four if at batte speed blend hot margarine four if at if batte margarine spray sides welloiled blend hot margarine boiling over and souffle scraping slightly ingredients blend hot margarine boiling over and souffle scraping rounded slightly ingredients blend <EOS>	in a large skillet combine all the sugar and water bring to a boil reduce the heat and simmer minutes or until the sugar is dissolved cool slightly in a small bowl add the sugar and stir until the sugar dissolves pour the syrup over the rice and garnish with a little sugar if desired <EOS>	combine all ingredients in a large mixing bowl blend well add the orange juice concentrate lemon juice and lemon juice and stir well add the orange juice and lemon juice and stir until well combined pour into a serving bowl garnish with lemon slices <EOS>	serve with warm applesauce optional peel and slice the apple into thick rings melt the oil in an ovenproof pan and brown them for about minute on each side put an apple ring on each chop whisk the milk eggs flour butter and salt until well mixed pour the batter over the chops and bake for about minutes at or until the batter is browned and puffy serve immediately with the warm applesauce since the yorkshire batter bakes with the pork chops it swells up and around the chops and becomes crisp on top but stays soft and moist inside like you think of yorkshire pudding

From the generated outputs, we can see that both B1 and B2 produce almost nonsensical recipes that use none of the ingredients mentioned. Both baseline used words that had no meaning (likely misspellings) such as “batte” and also included many ingredients such as “chocolate” and “pudding” which were not mentioned in the ingredients list given. This is likely due to the lack of data preprocessing for both ingredients and recipe since the original train data for ingredients includes many irrelevant words (noise) which might have confused the model during its learning process. The generated output for B1 and B2 has a direct correlation to their low BLEU-4 and METEOR score since the sentences do not make any sense and have no overlaps with the expected output. On the other hand, we can see that the Extension models perform much better after data preprocessing as they both produce responses that at least seems related to our ingredients list. However, they both suffer from repetition which is most likely due to our model repeatedly attending to the same attention location. This can be fixed in the future by using the Coverage Mechanism proposed in (Bosselut et al.) which introduces an additional trainable coverage vector to ensure attention locations are attended equally. Lastly, the NN model produced a completely unrelated response as well which was surprising given how the model is expected to work so further work has to be done to improve the KNN training process (as I have mentioned in the previous section)