Sequence to Sequence Learning: Food Recipe Ingredients Prediction

Steve Kim

Github

1. Introduction

In this project, I look at how a Sequence to Sequence model can be used to predict what food ingredients are needed for a given recipe.

Example: “Chile con Carne”

Ground Truth: “ground beef, onions, salt, olive oil, chili peppers, oregano, water, beans”

Predicted: “ground beef, beans, onions, salt, oil, peppers, oregano, water”

2. Literature Survey

Sequence to Sequence Learning with Neural Networks

- https://papers.nips.cc/paper/5346-sequence-to-sequence-learning-with-neural-networks.pdf
- Published by Google, the authors showed that they could achieve state of the art performance on text translation via the use of encoder/decoder LSTM networks, one which encoded input data, and one which decoded it. This paper has been cited over 10 thousand times, and is the basis for many NLP techniques today.

Sequence to Sequence Learning by Github

- https://kddseq2seq.com/
- This site/paper/notebook was created by Github to show how machine translation, chatbots, and text summary models are created via sequence to sequence learning via RNN. They applied their model to take on the CoNaLA challenge. In my project, I emulated their text vectorization, sentence embedding, and model architecture.

Resurgence of Structure in Deep Neural Networks

- https://www.repository.cam.ac.uk/handle/1810/292230
- This was a well organized PhD thesis that helped me understand the role of LSTM in the model I used in this project.

Details on the Datasets

Food Recipes from food.com 127MB, 176K rows.

- **Downloaded from Kaggle** https://www.kaggle.com/shuyangli94/food-com-recipes-and-user-interactions)
- **Contains ingredient mapping and consolidation files**

This repository of recipes was downloaded from Kaggle, and was originally on Food.com. The recipes themselves were generated by thousands of authors who are incentivized to get views of their recipes. As a result, there are a lot of variability and "clickbait-ey" recipe names, like "a different kind of breakfast pancake", or "calm your nerves tonic".

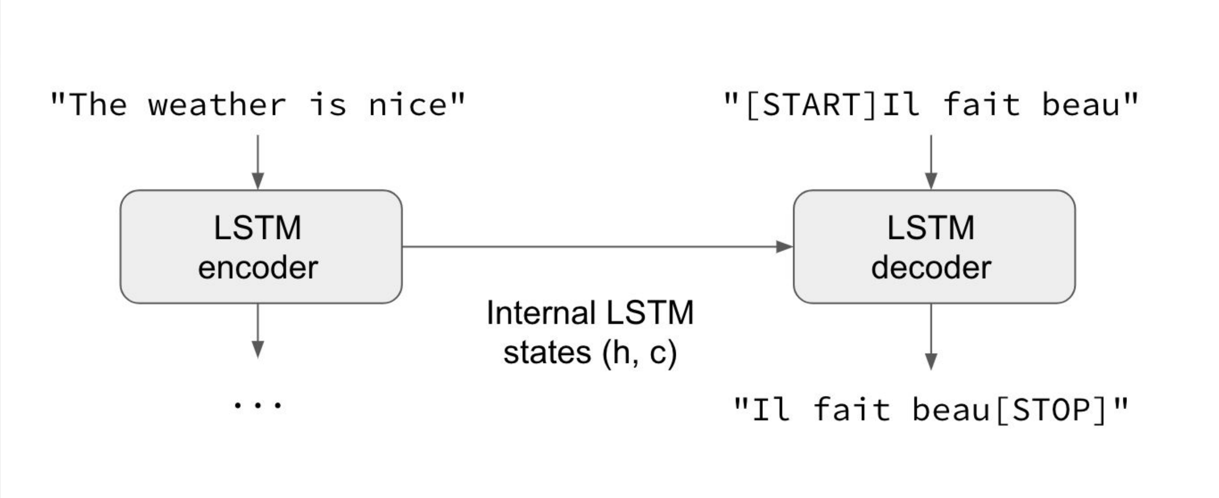
3. Model Details

First

Train an LSTM autoencoder to create a language model that can output recipe names.

Language model : seed with “cold”

Model Output: “cold cucumber salad with yogurt dressing and lemon dressing”



https://blog.keras.io/img/seq2seq/seq2seq-teacher-forcing.png

3. Model Details

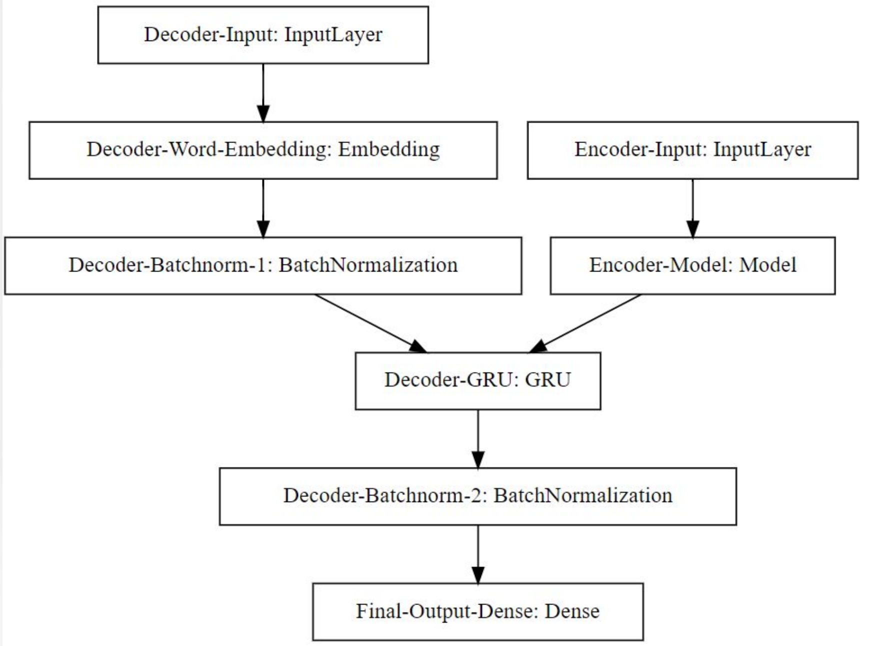
Second

Sequence to Sequence: Used the Language model to create sentence embeddings, which become vectorized and then encoded in the encoder-decoder network.

Encoder input: “dried navy beans, water, onion, bacon, brown sugar, molasses, tomato paste, salt, dry mustard, pepper”

Ground Truth: “crock pot baked beans from scratch”

Predicted: “baked beans in crock pot”


Decoder-Input: InputLayer
Decoder-Word-Embedding: Embedding
Encoder-Input: InputLayer
Decoder-Batchnorm-1: BatchNormalization
Encoder-Model: Model
Decoder-GRU: GRU
Decoder-Batchnorm-2: BatchNormalization
Final-Output-Dense: Dense

https://github.com/hohsiangwu/kdd-2018-hands-on-tutorials/tree/master/images

3. Model Details

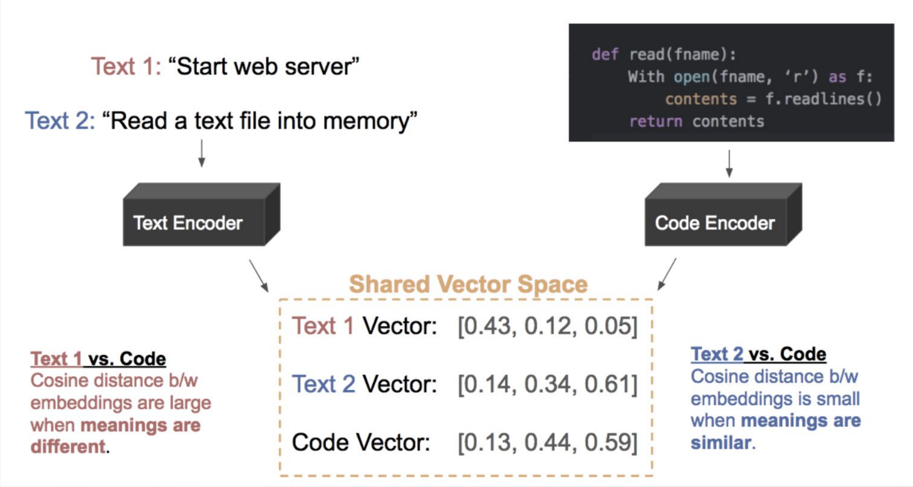
Final

Create a joint vector space to map all recipe names in a n-dimensional space, where n is the count vocabulary of the recipe names

Search input: “Chicken Pot Pie”

Looked up on food.com: “flour, salt, pepper, potato, onion, carrot, celery, chicken broth, margarine”

Predicted: “boneless skinless chicken breasts, margarine, water, ritz cracker crumbs, parmesan cheese, cream of chicken soup, sour cream, mushrooms”



```
def read(fname):
    With open(fname, ‘r’) as f:
        contents = f.readlines()
    return contents
```

Text 1: “Start web server”
Text 2: “Read a text file into memory”
Text Encoder
Code Encoder
Shared Vector Space
Text 1 Vector: [0.43, 0.12, 0.05]
Text 2 Vector: [0.14, 0.34, 0.61]
Code Vector: [0.13, 0.44, 0.59]

Text 1 vs. Code
Cosine distance b/w embeddings are large when **meanings are different**.

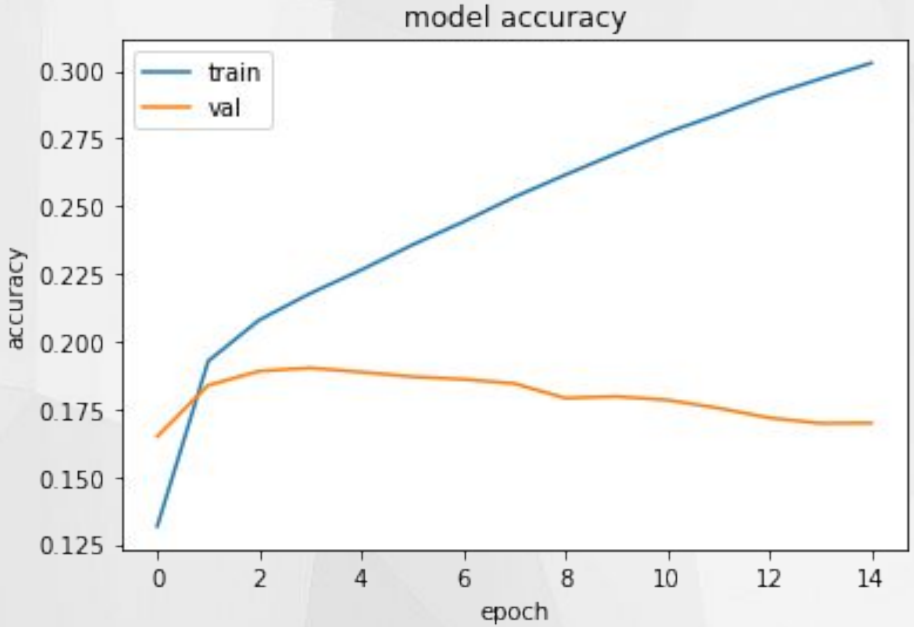
Text 2 vs. Code
Cosine distance b/w embeddings is small when **meanings are similar**.

https://cdn-images-1.medium.com/max/1280/1*zhLXNHK8ILaYV8tT-jDlOQ.png

4. Results

Three models that make up this project:

1. **Recipe name embedding autoencoder LSTM:**



Other Examples:

Seed: “cold”
Prediction: “cold cucumber salad with yogurt dressing and lemon dressing”

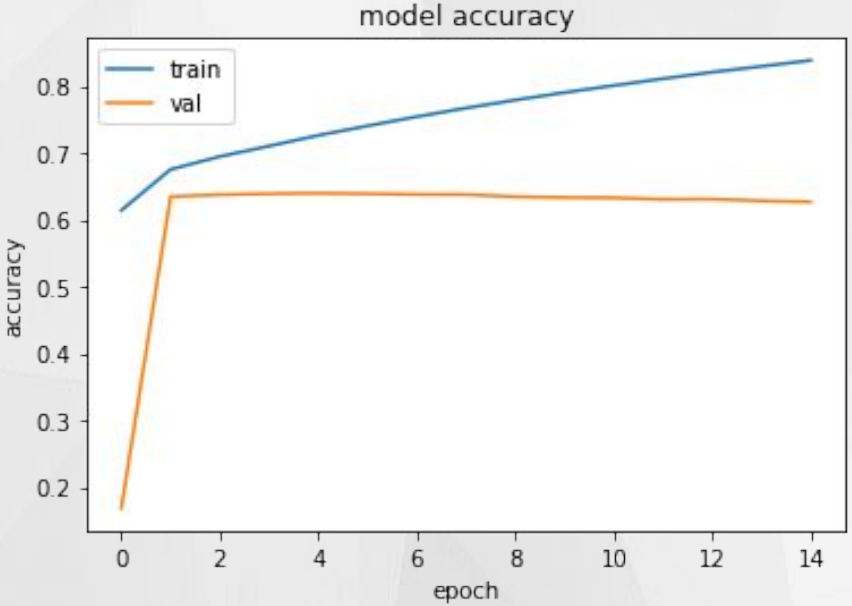
Seed: “hot”
Prediction: “hot and sour chicken salad with peanut dressing and dressing rsc”

Seed: “chicken”
Prediction: “chicken and rice casserole with sausage and peppers too easy to”

4. Results

Three models that make up this project:

2. **Decoder network**



Examples:

Ingredient list input: “fresh tomatoes, onion, lime, jalapeno pepper, fresh cilantro, salt”

Prediction: “Mexican Salsa”

Ingredient list input: olive oil, texas sweet onions, sugar, adobo sauce, tomatoes, avocado, lime, juice of, tequila, fresh cilantro, fresh parsley, ground beef, corn, adobo seasoning, garlic cloves, cumin, mexican oregano, monterey jack cheese, light sour cream, fat-free refried beans, tortilla chips

Prediction: beef tostadas with avocado salsa

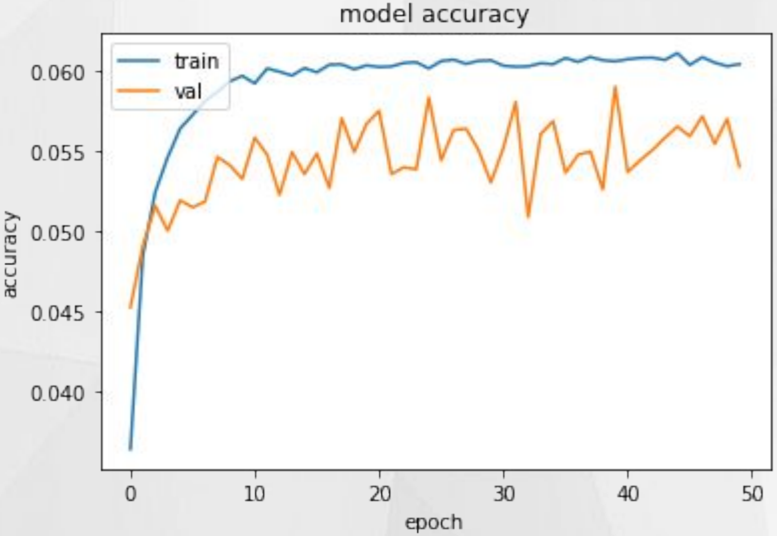
Ingredient list input: frying chickens, salt and pepper, olive oil, onion, minced garlic cloves, green bell pepper, red wine, tomato sauce, whole mushrooms

Prediction: pollo en vino chicken in wine sauce

4. Results

Three models that make up this project:

3. **Final Search**



Examples:

Ingredient list input: “chicken pot pie”
Prediction: “boneless skinless chicken breasts, margarine, water, ritz cracker crumbs, parmesan cheese, cream of chicken soup, sour cream, mushrooms
------

Ingredient list input: “Iced Tea”
Prediction: “limoncello, vodka, cranberry juice”

Ingredient list input: “Chocolate Chip Cookie”
Prediction: “butter, graham cracker crumbs, sweetened condensed milk, butterscotch chips, semi-sweet chocolate chips, flaked coconut, nuts”

5. Further Work

Additional dataset pre-processing

Consolidate the ingredients list

- Make different kinds of flours, like “wheat flour” “white flour”, and “bread flour”, combined into one term.

Subset data to a specific category

- For example, if I only included “baked goods” in the dataset, the model should be able to distinguish between the recipe names within this category, as most of them would share their top three ingredients, flour, butter, and water.

Model Improvements

For the final model search function, instead of using a greedy search on the vectorized dataset of recipes, I could instead implement a **beam search** algorithm that would search through k states and choose the best candidate of each expanded state, at each time step.

Citations / Appendix

References

[1]https://papers.nips.cc/paper/5346-sequence-to-sequence-learning-with-neural-networks.pdfom/

[2] https://github.com/hamelsmu/ktext

[3]https://kddseq2seq.c

[4]https://towardsdatascience.com/how-to-create-data-products-that-are-magical-using-sequence-to-sequence-models-703f86a231f

[5]https://link.springer.com/chapter/10.1007/978-3-030-32381-3_27

[6]https://www.usenix.org/system/files/sec20fall_jagielski_prepub.pdf

[7]https://www.ceid.upatras.gr/webpages/faculty/zaro/teaching/alg-ds/PRESENTATIONS/PAPERS/2019-Radford-et-al_Language-Models-Are-Unsupervised-Multitask-%20Learners.pdf

[8]https://www.repository.cam.ac.uk/handle/1810/292230

The End

Thank you