Generating Recipe Ingredients and Instructions from Recipe Titles

Justine Breuch Kerrie Wu Ben Randoing

Problem

Recipe generation from recipe titles only is unsolved, as current state of the art models in literature require both recipe titles and ingredients lists for instruction generation. We explored using transformer and LSTM models to produce meaningful ingredient lists when given recipe titles only. We then produce recipe instructions using the generated ingredients lists using an existing recipe instruction generation framework [1].

Dataset and Processing

Recipes1M+ [2]: Over 1 million recipe title, ingredient, and instruction triplets. We cleaned the data to remove extraneous text and constructed sequences consisting of the word-tokenized recipe title, followed by the ingredients, for each example.

Instruction Generation

Recipe instructions were generated using title and ingredients with a preexisting baseline model https://github.com/williamLyh/RecipeWithPlans that implements DistilBERT, a planning stage classifier to label individual recipe instruction sentences as a specific stage ("Pre-processing", "Mixing", "Cooking", etc). This compressed version of BERT reduces computational resources while maintaining accuracy through knowledge distillation. Subsequently, BART, a denoising auto-encoder, pretrains sequence-to-sequence models to produce an outline of recipe stages, and GPT-2 generates cooking instructions from a fine-tuned model. Recipes created using Recipe1M+ data and generated ingredients from a developed transformer were compared.

Decoding

For both architectures, we experimented using top-k, top-p, and beam search decoding. We conducted a grid search over k, p, t (temperature), and beam sizes to determine the best decoding hyperparameters. We then selected the best hyperparameters for further evaluation.

Ingredient Generation

For the LSTMs, we took all examples with title and ingredients and joined them together into one text. <title-end> tokens were appended to the end of the title and ingredients separated by special <ingr-end> tokens. Text was chunked into sequences of 16 and 64 where the target reflected the input shifted over by 1.

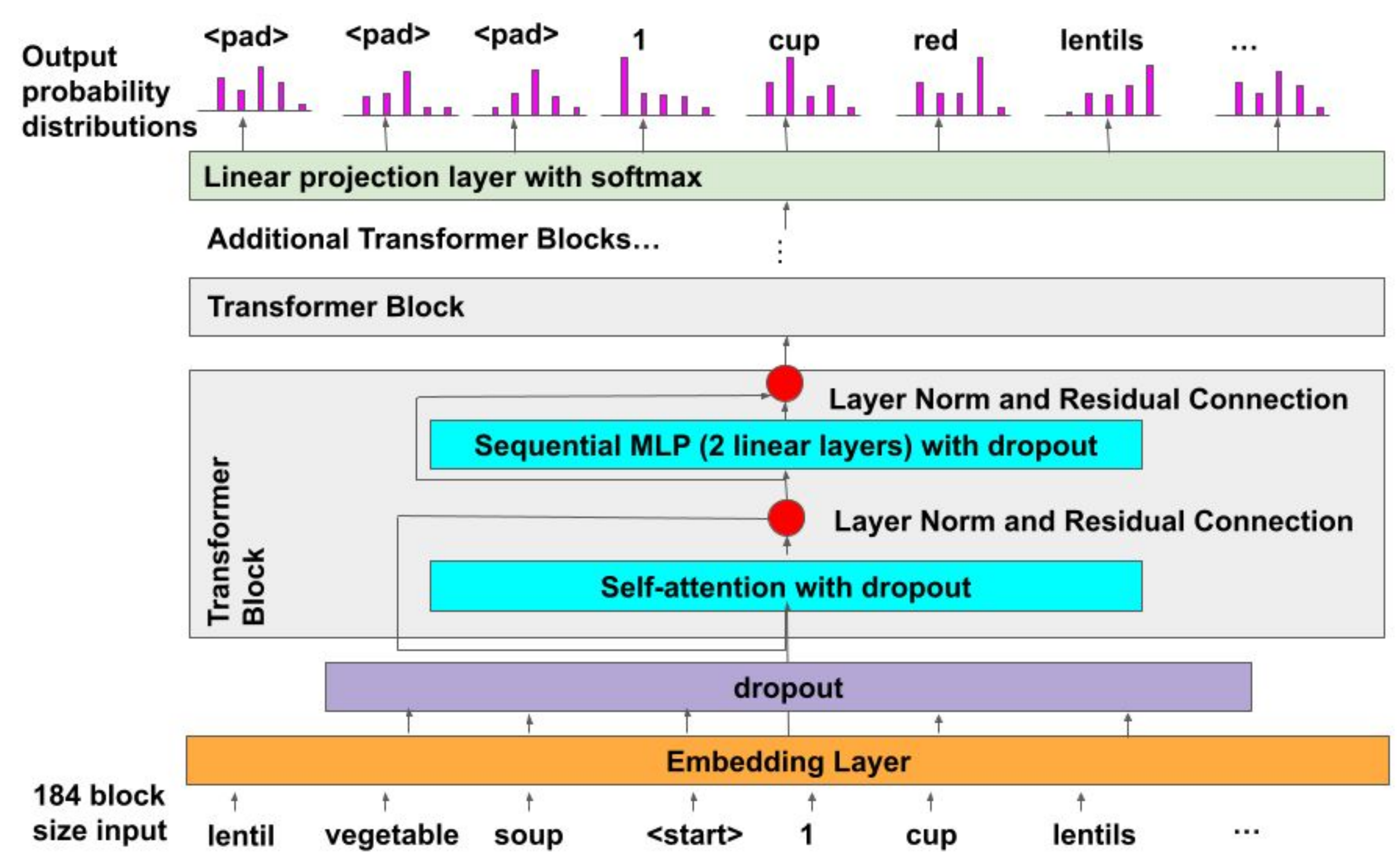
LSTM

The LSTMs had 3 variations: 2 LSTM decoders and 1 encoder-decoder models with teacher training and multiplicative attention, trained for next-token prediction. Each decoder model had an embedding layer of 1024, 2 hidden layer LSTM with state size 1024 and a dropout rate of 0.5 with a final projection layer for a distribution over the entire vocabulary (29,058). The larger block size of 64 yielded better results than the block size of 16, however, adding multiplicative attention did not vastly improve results despite a gamma repetition penalty of 0.1.

Transformer

Pre-training objective next-token prediction with span corruption and masking

Fine-tuning objective next-token prediction with recipe titles <MASK> out



Example Ingredient Generation

Recipe Title	Best Transformer Model	Best LSTM Model	Target
butternut squash soup or bisque (roasting method)	1 whole butternut squash, peeled, seeded, and diced | 1 whole onion, peeled and diced | 1/2 teaspoons salt | 1/2 teaspoons black pepper | 1/2 teaspoons cinnamon | 1 cup water | 1 cup chicken broth or bouillon | 1/4 cups heavy cream or half-and-half | 1/4 cups butter	1 bunch green onion | sliced, 2 shallots, minced | 1 teaspoon garlic powder | 1 teaspoon sambal oelek chili paste | 18 teaspoon salt | 1 tablespoon butter | 12 cup milk or 12 cup water | 12 cup light cream | parsley | pepper to taste	1 small butternut squash, peeled , diced | 1 medium onion , large dice | 1 tablespoon olive oil | 1 apple, skinned, large dice | 48 ounces chicken broth | 1 cup half-and-half | 2 tablespoons parsley , fresh , chopped | 1 tablespoon thyme , fresh, chopped | crouton | sour cream

Results and Analysis

Ingredient Generation

Model architecture	Decoding method	F1	BLEU
LSTM decoder with block size 16	K=1	10.9	5.3
LSTM decoder with block size 64	K=1, P=0.3, T=0.9	11.7	6.4
LSTM with multiplicative attention	K=5, P=0.3, T=0.9	12.4	7.2
Transformer A (4-layer, 256 embd. size)	K=10, P=0.3, T=0.8	30.6	9.3
Transformer B (4-layer, 256 embd. size, pretrain)	K=3, P=0.3, T=0.9	29.9	8.9
Transformer C (8 layers, 1024 embd. size, pretrain)	K=8, P=0.8, T=0.8	34.7	11.2

Recipe Instruction Generation

Model architecture	Decoding method	BLEU	ROUGE-L
Baseline	N/A	13.73	39.1
Transformer	K = 3, P = 0.3 , T = 0.9	3.41	22.7
Transformer	K = 5, P = 0.9, T = 0.9	3.20	22.6

Qualitative Evaluation

Good: Our best model predicted reasonable ingredients relevant to the title most of the time. The recipe instruction module could generate coherent instructions based on the generated ingredients.

Bad: Generated ingredients were sometimes repeated, had inappropriate quantities, were missing ingredients mentioned in the title, or included inappropriate ingredients that didn't mix well.

Conclusions/Future work

It is feasible to generate recipe ingredients and instructions from the title only, sometimes. However more work is needed to improve consistency in ingredient generation. Cleaning the data more thoroughly, increasing model size, and exploring more complex controlled decoding methods can help.

References

Recipes1M+
Liu et al. Plug and Play

Acknowledgments

Mentor: Siyan Li