Forage: Optimizing Food Use With Deep Learning Generated Recipes

Angelica Willis | Elbert Lin | Brian Zhang

{arwillis, el168, yzhng} @ stanford.edu

Motivation

Food waste is a major issue in the United States. An American family of four discarded on average $1600 value of produce annually.[1] Our project, Forage, is a machine learning algorithm that considers what you have in the fridge or pantry, to generate a unique recipe that utilizes those available ingredients.

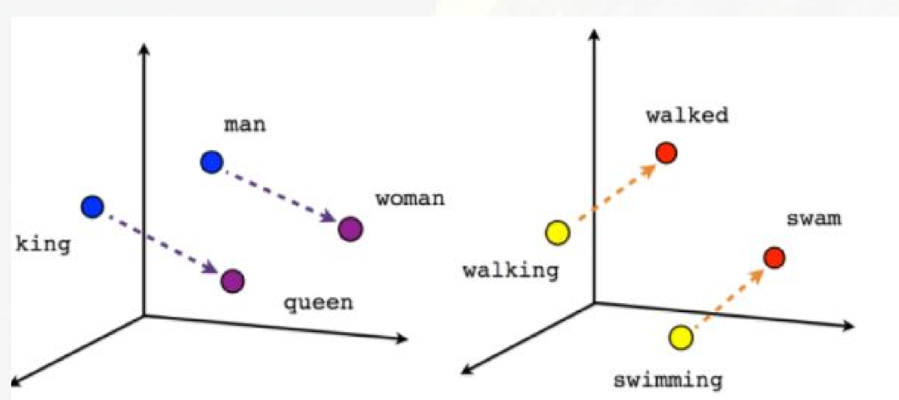
Data

We filtered and extracted title, categories, ingredients and instructions of each recipe from the largest, free recipe database Meal-Master and picked out food nouns to concisely represent ingredients and simply tokenized instructions in sentences to preserve the specific "recipe language".

Vocabulary Size: 300,000

Features

Word2Vec: Vector representation of words allowed to map semantically similar words close by each other.



We used Skip-Gram model that predicts source context-words from the target words, which makes more sense on larger datasets and recipe generation.

the quick brown fox jumped over the lazy dog

$$J_{NEG} = \log Q_\theta(D = 1|\text{the, quick})$$
$$+\log Q_\theta(D = 0|\text{sheep, quick})$$

Model

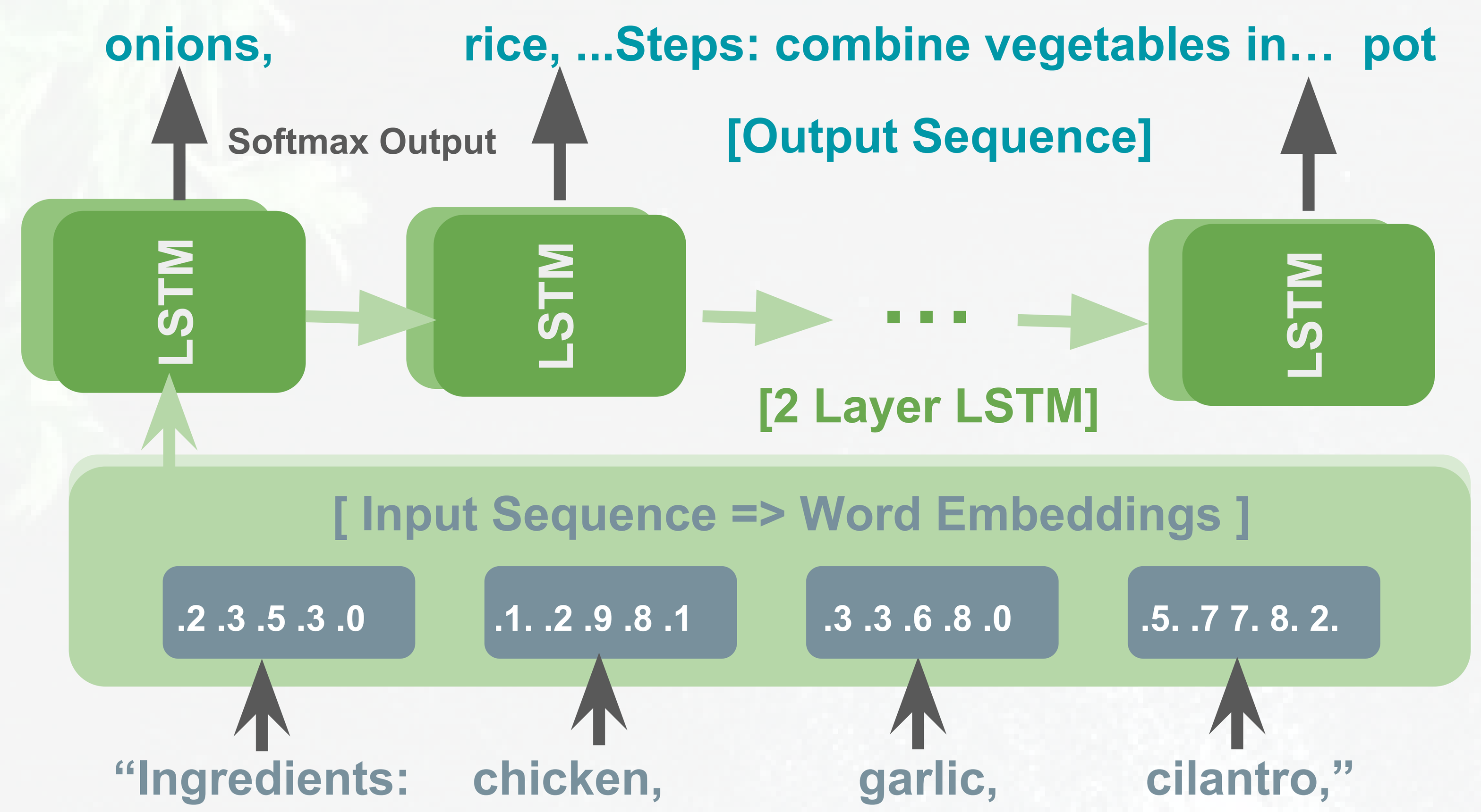
Model Architecture: Long Short-Term Memory Model (LSTM) RNN

Gradient Descent Batch Size: 50

Layers: 2 LSTM + Softmax Output

Hidden State Size: 256

Cost Function: average negative log probability of the target words



Model Interaction Language (MIL)

Training example:

Recipe: { Ingredients: ["Shrimp", "Leeks", "Tomatoes"], Steps: ["Combine vegetables in a pot", "add spices and let simmer"...], Category: ["Dinner", ...] }

Example Interaction Structure (after some post parsing)

Human Request (input) — Model Response (output)

Ingredients → ["Chicken", "Onions", "Tomatoes"]

Steps → ["Combine vegetables in a pot", "add spices and let simmer"...]

Category → "Spicy"

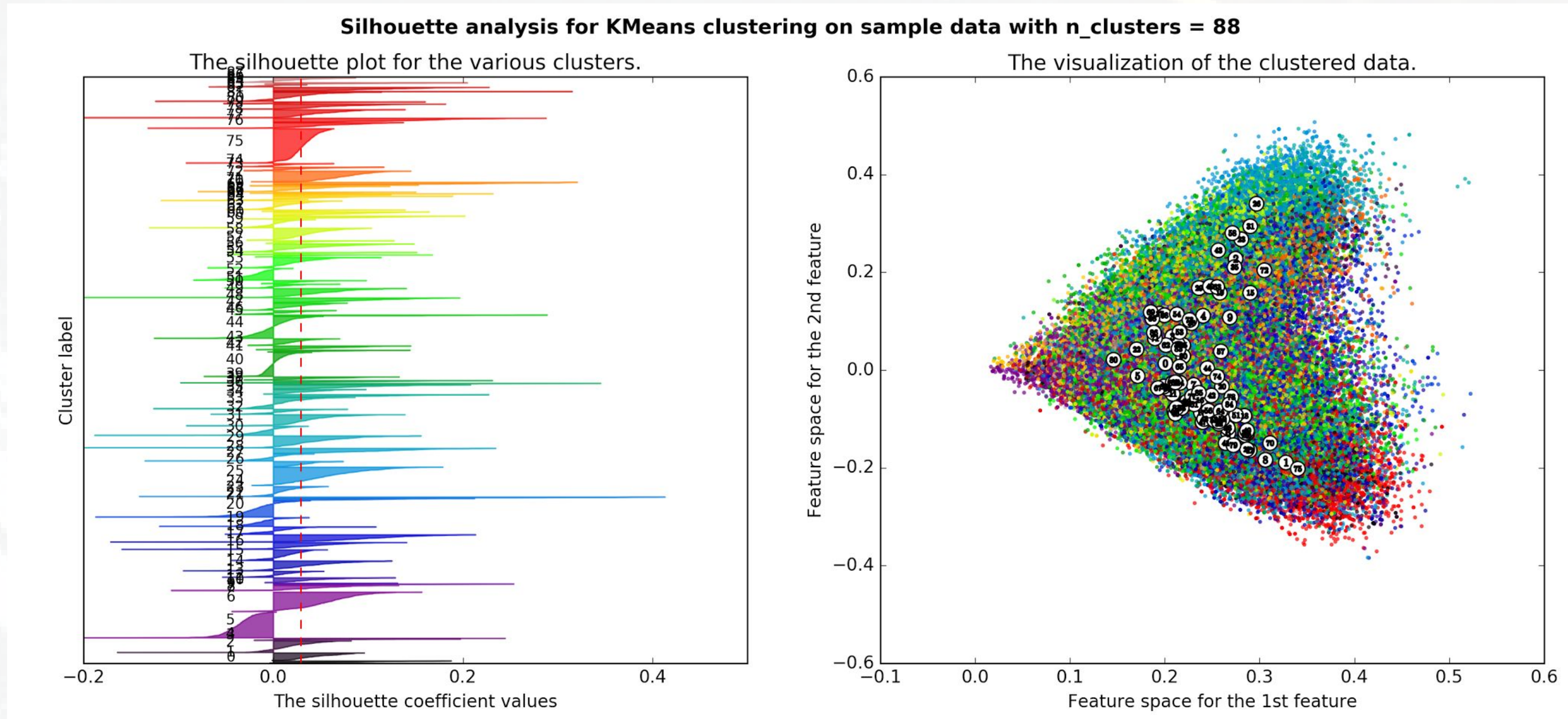
Out of the necessity for our model to understand the complex nature of our data (i.e. ingredients, steps, recipe titles, and categories) we designed and implemented a new approach to facilitate genuine and dynamic interaction with a trained model, which we call Model Interaction Language, or MIL. MIL is similar to the structure of JSON.

Evaluation

We first classified our dataset using *k*-means clustering, then assigned our generated recipes to the closest centroid. For initial testing, we used metric evaluations (Euclidean and cosine distance and Language Tool score[2]), and we plan on using human evaluations to accurately determine the success of our model.

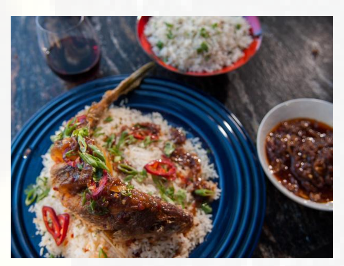
K-Means

We used 88 clusters, chosen using silhouette analysis (which analyzes how similar an object is to its own cluster compared to others), and a tf-idf matrix of 10000 features reduced to 1000 using LSA, where the explained variance of the SVD step was approximately 70%.



Results

Example Recipes



CATFISH SOFRITO: Categories: Seafood | Ingredients: spinach, egg, salt, mustard, turmeric Steps: "Combine soy sauce, sugar, curry powder and oregano in large saucepan.", "Bring to boiling, stirring occasionally.", "Add soy sauce and wine; simmer, covered, 2-3 minutes.", "Add snow peas and cook until well-browned and slightly thickened. Beat 1/2 cup at a time and browned, stirring after 45 seconds before serving.", "Serve over rice...

RICE & BEAN SPROUT SALAD: Ingredients: bean, pepper, sauce, lemon juice, cream, salt | Steps: "Cover and shake until thick and opaque.", "Add the hard-boiled egg whites, a little at a time, turning them so they won't brown and bright green.", "Increase to low heat until constant all the ingredients are smooth.", "Add them to the pasta.", "Add the remaining ingredients and cook briefly, into a sauce.", "Serve with marinade to low-heat point are served the room-temperature with your favorite sticks packet or for a tossed salad, a few zucchini...

Quantitative Results: Comparing Model Types

Test set size: 300 recipes. Euclidean distances range from 0 to ∞ while cosine distances and LT score range from 0 to 1. Smaller distances are preferred, while higher LT scores are preferred.

Measure	Original Recipes	Vanilla Model (VM) Epochs: 3 Dataset size: 115K	VM + word2vec Epochs: 3 Dataset size: 60K	String input instead of MIL Epochs: 3 Dataset size: 60K	Vanilla Model (VM) Epochs: 4 Dataset size: 60K
Euclidean distance	0.875	0.928	0.910	0.910	0.943
Cosine distance	0.527	0.632	0.583	0.576	0.660
LT score	0.913	0.779	0.750	0.885	0.782

Discussion

Overall, our generated recipes made semantical sense and had an expected structure, thanks to MIL. However, the correlation of ingredients in the "ingredients" and "instruction" sections still needs improvement, as this relation was not characterized in the model cost function. Although LSTM model was computationally expensive to train, we found by controlling the number of most frequent word as vocabulary, the model could run faster and avoid overfitting. While standard evaluation is difficult to achieve in this context, we were able to use *k*-means and Language Tool to test our results, as they do not require a reference.

Future Work

Model improvements

We would like to conduct human evaluations on a larger scale to more accurately gauge the success of our model. We are continuing hyperparameter sweeps to optimize our model, including training with 512 hidden units in each layer. We will also explore PCA analysis to aid in model evaluation.

References

1. EPA Land [@EPALand], "Americans throw away $1,600 of wasted food per year. Reduce food waste and save money http://www.epa.gov/recycle #nowastedfood #FRSCharleston" [Tweet], https://twitter.com/epaland/status/666340964064186368, Nov 2015.
2. C. Napoles, K. Sakaguchi, and J. R. Tetreault, "There's No Comparison: Reference-less Evaluation Metrics in Grammatical Error Correction," *CoRR*, abs/1610.02124, 2016.