In a warm scone. oh my... scrumptious

Strix Akiba
2021-8-4

I am:

- An independent researcher bored of eating cup noodles every day
- Enthusiastic about playing with food science from the perspective of a data scientist
- Seeking funding from potential restaurateurs

Problem Statement:

Machines can be more efficiently creative than humans. It is possible to demonstrate this by training a machine learning model that can recognize patterns in cooking instructions then communicate, in English, a new or customized version of a recipe.

Business application:

- The model will be able to propose new restaurant menus, especially the ever so popular “fusion” cuisines, or adapt existing ones to specific diets
- Could also be use in other industries
  - Ex: script writing or art

Background:

Some background on the food industry:

- More than 1 million restaurants at any given time
- There are approximately 13,000 new restaurants opening in the US each year, and the trend over the last few years is for this to increase (~2%/year)
- 17% of these new restaurants close within the first year
- The average lifespan of a restaurant is 4.5 years

What does this mean?

- There is no shortage of hungry patrons
- People get bored of food quickly and you, the restaurateur needs to be able to adapt to succeed

Stats from https://www.smallbizgenius.net/by-the-numbers/restaurant-industry-statistics/#gref who took them from the US Bureau of Labor Statistics

Data:

- The data was collected using the spoonacular api which boasts more than 700,000 recipes
    - Spoonacular itself is designed to scrape recipe cards off of all the popular cookbook / recipe sites
- 500 recipes per day gathered with a call for random popular recipes
    - Random was most efficient when working with the free tier
    - Otherwise a separate call would need to be made to find the recipe and separately acquire its instructions
- The useful descriptive features included in the data were:
    - Vegetarian, Vegan, gluten free, very healthy, cheap, sustainable, low fodmap, cuisine, dish type, and occasion categoricals
    - An English language summary
    - English language Instructions
    - English ingredients and their amounts

Cleaning:

- I explicitly mentioned the language was English as I have worked with apis that returned foreign languages that had to be then cleaned manually, spoonacular saved me the time
- Data came in nested json and was extracted to create a dataframe that was much easier to work with, this dataframe was then saved as a csv
- Some cleaning on punctuation was done using spaCy, a very capable natural language preprocessing library
- One point of frustration was that, because the call was for random “popular” recipes, I had to eliminate duplicates. Over 50 calls of 100 at a time, I ended up with less than 1000 unique recipes.
- While there was no missing instruction or summary data, the majority of recipes did not classify dish or cuisine type, something I hoped to use to make stylistic changes to recipes

Step 1: Count Vectorize + KMeans Clustering

- The instructions and summary was count vectorized to allow for an attempt at classification
- KMeans was used to test whether or not a machine model could come up with meaningful categories unsupervised
- While not all the models found something interesting, some did an exceptional job at capturing vegetarian recipes
    - the vegetarian boolean feature was not a training feature, just the count vectorized data

EDA:

- Recipe summaries contained 5385 unique words while the instructions only contained 3904
- Chicken seems to be the most popular ingredient, mentioned at least once in the ingredients of 158 out of 760 recipes (20%)
  - One recipe, “Three-cup Chicken,” even mentions chicken 24 separate times in its summary and ingredient list
- At #30, in most commonly used words of the combined summary and instruction (following stop words that were intentionally left in), it was also the top ingredient!

Step 2: Word2Vec

Word2Vec was set up to analyze cosine similarity between words. It was set up in preparation of transforming one cuisine type into another by using the best match. While the vectors have been assigned, proper instruction generation should be completed first and as such, this is a feature to be worked on.

Step 3: Markov Chain

- Stochastic predictive model that weight potential probabilities based on training model occurrence patterns
- Like a time series model in that it progresses linearly and keeps track of past choices
- Used to generate the instructions using a prebuilt library, **markovify**

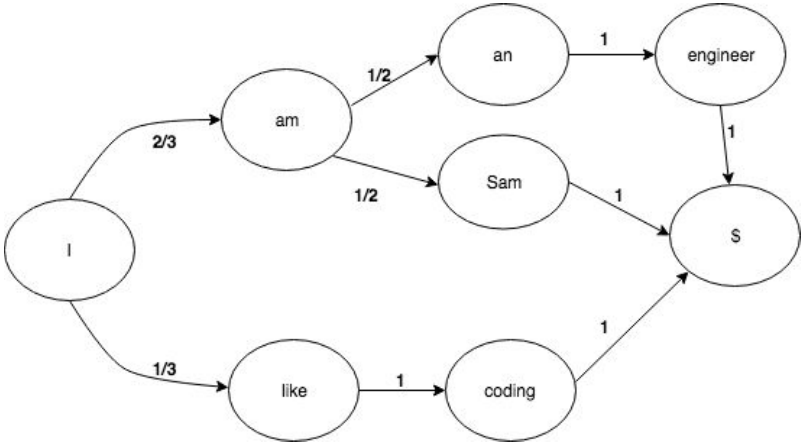

Image credit: https://mb-14.github.io/tech/2018/10/24/gomarkov.html

A newborn’s first words:

- In a warm scone. Oh my… Scrumptious
- Transfer the cream then place on sheet pan.
- Bake for 1520 seconds just until edges are completely mixed fold in the gelatin and water over the veggies and fruits.
- Put the meat with the mixture is dry and season with salt and pepper.
- Stir. increase the heat stir constantly.
- Serve with pasta or rubbed over toasty bread with a nonstick pan helped me since i am so in love with this chocolate heaven on earth is in course crumbles.

Conclusion:

- I admit, it’s not great
- It DOES manage to use consistent ingredients
- It formulates somewhat coherent sentences, albeit with some grammar mistakes
- It even includes punctuation
- It’s lightweight enough that I never needed to time fitting the model

Next Steps:

- Use a premium tier of the api to be able to more easily request non duplicate recipes
- Re-clean the data to remove excess commentary.
  - This would include creating a custom list of stopwords
  - Design a cleaning script to standardize units
- Primary goal is to use the models and tools in conjunction with one another
  - KNN modeling would be used to better classify and label foods by training and labeling missing cuisine and dish types
  - A custom Markov Chain trained on better organized instructions cleaned up by reinforcement learning and more implementation of spaCy in defining sentence structure
  - Word2Vec in conjunction with KMeans will be used to create replacement mappings from one type of cuisine to another allowing for a marketable “fusion cuisine generator.”
- Test the model in another field
  - ex) Rework a book from one author’s linguistic type to another’s

Thank you!