Optimizing Cookie Recipes for Ratings Using Machine Learning and Deep Vector-to-Sequence Recurrent Neural Models

Mackenzie O'Brien and Pablo Rivas
Department of Computer Science, School of Computer Science and Mathematics
Marist College, 3399 North Road Poughkeepsie, New York, 12601, United States

Abstract— *This research studies machine learning algorithms in the process of producing new recipes for cookies. The objective of these experiments was to generate a cookie recipe that was optimized for critic ratings by using multiple types of neural and non-neural networks to predict and create chocolate chip cookie recipes based on the input of over 250 human-made recipes and instructions. There were 138 different parameters inputted, including Rating, Calories, and 136 different ingredients such as sugar, flour, and egg. To get the instructions, we created a vector-to-sequence algorithm that takes the input of a recipe ingredient vector and uses the instructions from the 250 man-made recipes to make predictions about the sequence of instructions.*

Keywords: Vector-to-Sequence, Neural Networks, LSTM, Gradient Boosting, Extreme Trees, Attention Mechanism

1. Introduction

Machine Learning(ML) is a sub-set of artificial intelligence where computer algorithms are used to autonomously learn from data and information. ML has been used in very interesting applications today, such as personal assistants, smart speakers, and fraud detection. Recent advances in deep learning enable us to do more complex and interesting things, such as self-driving cars and image recognition and stock forecasting. In this research we decided to study ML and DL algorithms in the problem of selection of ingredients and production of instructions using, a,b,c,d, and x, respectively. Other works such as Google's [1], Clifford's [2] Naiks's [3] have approached the problem, but we believe this approach is unique and has not been attempted before.

2. Background and Methodology

In our research, the production of new recipes were split into two stages: computation of ingredients and the generation of instructions. First, we used different algorithms to return ingredient vectors which were then fed into a LSTM recurrent neural network to generate instructions for that specific vector of ingredients. This last process is known as vector-to-sequence modeling.

This experiment involved testing 9 different algorithms for generating ingredient vectors: Deep Neural Networks, Extremely Randomized Trees, Gradient Boosting, Linear Regression, Neural Networks, Normalized Neural Networks, Wide Neural Networks, Random Forest, and Support Vector Machine (SVM). Our target variable **y** was the 'rating' of a cookie. By using `allrecipes.com` as our dataset, we were able to target our predicted rankings off of the ratings included with each recipe in our training data set. Each algorithm generated new cookie recipes by selecting a previously existing value for each ingredient column and analyzing how the combination compared to similar recipes by calculating uniqueness and simplicity metrics, with simplicity given by:

Table 1: Cookie batches w. Uniqueness and Simplicity scores

Batch	Algorithm	Simple	Unique	Pred. Rank
A	ERT*	14.1	5.4	5.0
B	Gradient Boosting	20.3	5.8	5.1
C	ERT*	4.1	2.3	5.0
D	Deep Learning	11.6	6.1	9.5
E	Deep Learning	17.3	5.0	9.4

*Extremely Randomized Tree

$$Simplicity = ||y - y^*|| + ||\mathbf{w}|| \tag{1}$$

where y is the true ranking, y^* is the predicted, and $\mathbf{w}$ is the vector of weights associated with the regression problem: $\mathbf{w}^T\mathbf{x} = y$. Then we define the uniqueness metric as follows:

$$Uniqueness = \min_{i=1,\ldots,N} ||\mathbf{x}_i - \mathbf{x}^*|| \tag{2}$$

where $\mathbf{x}_i$ is the i-th sample vector from the ingredients data set, $\mathbf{x}^*$ is the new/proposed set of ingredients and N is the size of the data set. Table 1 shows the metrics described above for a selected group of sets of ingredients produced by the different methods. We also calculated the values of Mean Squared Error Loss for each algorithm to narrow down which ones would be used for testing. We used the standard formula for mean squared error loss (for ingredients) given by: $\frac{1}{N}\sum_{i=1}^{N}(y_i - y^*)^2$.

After training the above mentioned algorithms to output recipes with high predicted rank, low simplicity, and high uniqueness, our next goal was to train our instruction algorithm to generate recipes that included all of the steps for baking based on an ingredients vector. The algorithm uses a Long Short-Term Memory (LSTM) recurrent neural

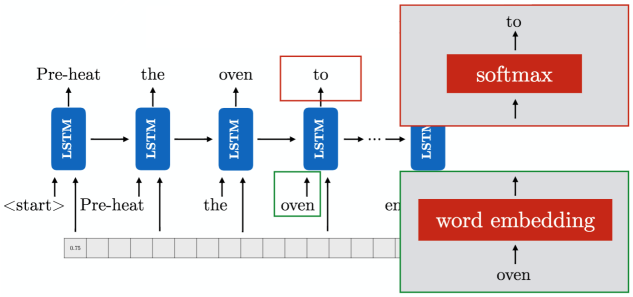

Fig. 1: Our merge model with an ingredients attention mechanism.

network, as illustrated in Figure 1, which incorporates each generated recipe as an attention mechanism [13]. By creating a modified merge model with the attention mechanism for ingredients, we attempted to train the model to better predict words in the instructions. The text input for each recurrent layer includes the ingredients vector, the previously outputted text, and the target output. The model was trained to have 150 recurrent layers of the LSTM, each with 932 outputs with soft-max activation. The length the sequence of 150 words was chosen by analyzing our instructions data set in order to achieve the inclusion of full length recipes for about 98% of the data set. This results in recipes of 150 words, one word of output per iteration. The soft-max activation works by selecting the word with the highest probability as the correct output. To pass the previous word into the next LSTM iteration, we used a word embedding process that translated the word back into vector form. The cross entropy loss was calculated for each sequence produced, which analyzes how accurate a sequence is compared to instructions in our data set.

The cross entropy loss is defined as:

$$\frac{1}{N}\sum_{n=1}^{N}\sum_{c\in C}\mathbf{d}^{*}_{cn}\ln\mathbf{d}_{cn}+(1-\mathbf{d}^{*}_{cn})\ln(1-\mathbf{d}_{cn}) \qquad (3)$$

where $\mathbf{d}_n \in \mathbb{R}^{932}$ is the true probability of the n-th sample belonging to a specific word, c, in the dictionary of words for directions. The size of such dictionary is 932.

3. Baking, Serving, and Survey Methods

After designing the algorithms, we proposed 5 different taste test experiments, each time testing a cookie designed with our algorithms against our control cookie, the well-known Nestle® Toll House® chocolate chip cookie [7]. The following paragraphs explain the baking and serving protocols.

3.1 Baking and Serving Protocol

Each taste experiment consisted of 40 cookies each for the control and the ML recipes, and the cookies were baked the day before each experiment to ensure freshness and quality remained the same for each test. Our procedure for serving the cookies is as follows:

1) We explained to the participant the risks associated with this experiment and made available a copy of a standard liability waiver for further reading and answered any questions before proceeding.
2) We then provided each participant two cookies: the experimental cookie and the control cookie. Samples were placed in small bags labeled Cookie 1 or Cookie 2. Every bag contained labels with links to an online survey for each cookie, and the table had a napkins and cups of water available for each participant.
3) Participants were discouraged from talking to one another during the tasting event and were not able to see how other participants are scoring each sample.
4) We asked each participant to test one cookie by first recording their score for appearance, then aroma, then taste, and finally texture. Note that texture pertains to how the food feels in your mouth. For example: crunchy, chewy, juicy, soggy, creamy, and so on.
5) After tasting the sample, we provided water to cleanse their palate. We then asked the participants to repeat steps three and four for the second sample cookie.

3.2 Survey Design

For each cookie that a participant tasted, we asked them to complete a survey giving their consent to use their information and questions about different attributes of the cookie and their overall satisfaction with the cookie. Survey questions included:

- Appearance:1 (Unfit for consumption) to 5 (Excellent)
- Aroma: 1 (Unfit for consumption) to 5 (Excellent)
- Taste: 1 (Unfit for consumption) to 5 (Excellent)
- Texture: Crunchy, Chewy, Gooey, Juicy, Soggy, Creamy, Other.
- Overall Satisfaction on a scale of 1 (Hated It) to 10 (Loved It)

In our research, we are most interested in the results of overall satisfaction with the cookie, as the goal is to produce a recipe with optimized ratings. Our definition optimized rating is a cookie that receives high average ratings for overall satisfaction.

4. Analysis

In this section we discuss the process of analysis of results in their different areas. We begin with the models trained over ingredients, and how we assessed their quality; and then models to learn to produce instructions for baking the ingredients and performance metrics during and after training. Finally, we analyze the survey results.

Table 2: Mean-Square Error & Coefficient of Determination

Algorithm	MSE	R^2
Deep Neural Network	0.0231	0.9267
ERT*	**0.0001**	**1.0**
Gradient Boosting	0.1111	0.6481
Linear Regression	0.1056	0.6656
Normalized Neural Network	**0.0219**	**0.9305**
Random Forests	0.0763	0.7584
Wide Neural Network	**0.0204**	**0.9354**
Shallow Neural Network	0.0429	0.8634
SVM	0.1873	0.4068

*ERTs keep copies of the data and usually yields perfect correlation.

Table 3: Cookie Recipes & Their Uniqueness & Simplicity

Batch	Algorithm	Simple	Unique	P. Rank
A	ERT	14.1	5.4	5.0
B	Gradient Boosting	20.3	5.8	5.1
C	ERT	**4.1**	2.3	5.0
D	Deep Learning	11.6	**6.1**	9.5
E	Deep Learning	17.3	5.0	9.4

4.1 Ingredients Selection

To analyze the algorithms that generated ingredient vectors, mean-squared error loss and the Coefficient of Determination were calculated for each algorithm, and the best algorithms were chosen to pick recipes from. Based on the mentioned metrics from each, shown in Table 2, we narrowed our focus to the following three algorithms: Deep Learning, Gradient Boosting, and ERTs. We chose the Deep Learning and ERT Algorithms because the mean squared error loss was low and the coefficient of determination,R^2, was high, while the Gradient Boosting Algorithm was chosen to represent the other side of the spectrum with high loss and R^2. The mean squared error loss is defined as $\frac{1}{N}\sum_{i=1}^{N}(y_i - y^*)^2$, and the coefficient of determination is defined as $R^2 = 1 - \frac{u}{v}$, where u is the residual sum of squares $\sum_{i=1}^{N}(y_i - y^*)^2$ and v is the total sum of squares $\sum_{i=1}^{N}(y_i - \bar{y})^2$ Here, $\bar{y}$ indicates the mean of y. The appendix contains the actual recipes selected for baking.

As shown in Table 3, five recipes were then selected from the three chosen algorithms to test. We chose to do two of each from the better scoring algorithms and one from Gradient Boosting. When choosing these recipes, we sorted first by predicted rank high to low, then by simplicity low to high and uniqueness high to low. After sorting, we chose one of the top few recipes for each.

4.2 Instructions Productions

As the vector-to-sequence algorithm is previously untested in other research, the end results leave something to desire in terms of inclusion of all ingredients and actual usability. However, it is an accomplishment to have gotten a working algorithm that takes in an ingredient vector and outputs a semi-usable recipe. Improvements include ensuring that the instructions contain all ingredients in the vector that contain non-zero values and eliminating repeating loops that the algorithm gets stuck on.

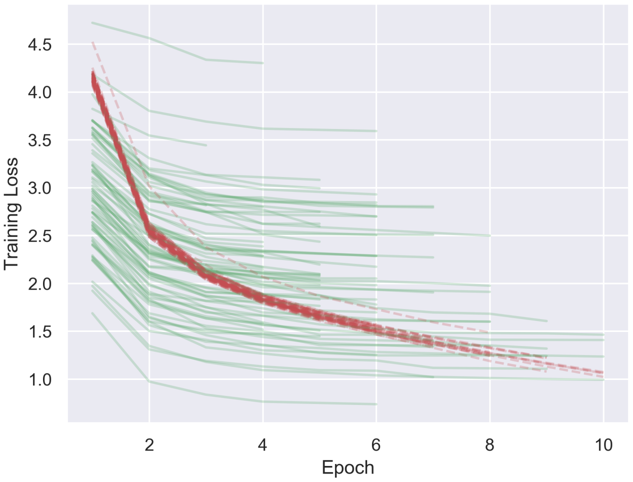

Fig. 2: Loss Score Per Epoch

In Figure 2, each run is shown with the corresponding loss score against each epoch. The length of each line indicates how many epochs training lasted until the Loss stopped decreasing. Figure 2 suggests that in most cases five epochs is enough for convergence. The goal and result of the model was to train it to reduce the cross entropy loss score. 3 shows the range of scores for each epoch, showing that the best runs reached the lowest loss at epoch nine.

A BLEU score (bilingual evaluation understudy) is an algorithm for evaluating the quality of text which has been machine-translated from one natural language to another. Quality is considered to be the difference between a machine's output and that of a human. In other words, the closer a machine's output to that of a human's, the better it is. The output is always between 0 and 1, and the higher the score, the better the machine output, and the score is gathered by individually calculating segments (generally sentences) and averaging the results for an estimate on overall quality [6].

.

4.3 Survey analysis

As shown in Figures 4 and 5, none of the ML generated cookies scored better in overall satisfaction; however, cookie B came the closest. Cookies C and D scored the worst, proving that a simple and not very unique cookie nor a non-simple but moderately unique cookie are not always the best choices. The DNN cookies fared the worst in cookies D and E. Cookie E had to be modified to form a proper dough, as the generated recipe contained 0 dry ingredients; surprising since the simplicity score suggested a complex cookie. Figure 4 shows the histogram of the responses to