Diet Planning with Machine Learning: Teacher-forced REINFORCE for Composition Compliance with Nutrition Enhancement

Changhun Lee
Soohyeok Kim
Chiehyeon Lim*
messy92@unist.ac.kr
sooo@unist.ac.kr
chlim@unist.ac.kr
Ulsan National Institute of Science and Technology
Ulsan, Republic of Korea

Jayun Kim
Yeji Kim
Minyoung Jung*
jydk6557@naver.com
kimhana0419@naver.com
my.jung@kosin.ac.kr
Kosin University Gospel Hospital
Busan, Republic of Korea

ABSTRACT

Diet planning is a basic and regular human activity. Previous studies have considered diet planning a combinatorial optimization problem to generate solutions that satisfy a diet's nutritional requirements. However, this approach does not consider the composition of diets, which is critical for diet recipients' to accept and enjoy menus with high nutritional quality. Without this consideration, feasible solutions for diet planning could not be provided in practice. This suggests the necessity of diet planning with machine learning, which extracts implicit composition patterns from real diet data and applies these patterns when generating diets. This work is original research that defines diet planning as a machine learning problem; we describe diets as sequence data and solve a controllable sequence generation problem. Specifically, we develop the Teacher-forced REINFORCE algorithm to connect neural machine translation and reinforcement learning for composition compliance with nutrition enhancement in diet generation. Through a real-world application to diet planning for children, we validated the superiority of our work over the traditional combinatorial optimization and modern machine learning approaches, as well as human (i.e., professional dietitians) performance. In addition, we construct and open the databases of menus and diets to motivate and promote further research and development of diet planning with machine learning. We believe this work with data science will contribute to solving economic and social problems associated with diet planning.

CCS CONCEPTS

• **Applied computing** → **Health care information systems**; *Health care information systems*; • **Theory of computation** → **Sequential decision making**; **Sequential decision making**; • **Computing methodologies** → Natural language generation.

KEYWORDS

diet planning, machine learning, controllable sequence generation, teacher-forcing, REINFORCE

ACM Reference Format:
Changhun Lee, Soohyeok Kim, Chiehyeon Lim, Jayun Kim, Yeji Kim, and Minyoung Jung. 2021. Diet Planning with Machine Learning: Teacher-forced REINFORCE for Composition Compliance with Nutrition Enhancement. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '21), August 14–18, 2021, Virtual Event, Singapore.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3447548.3467201

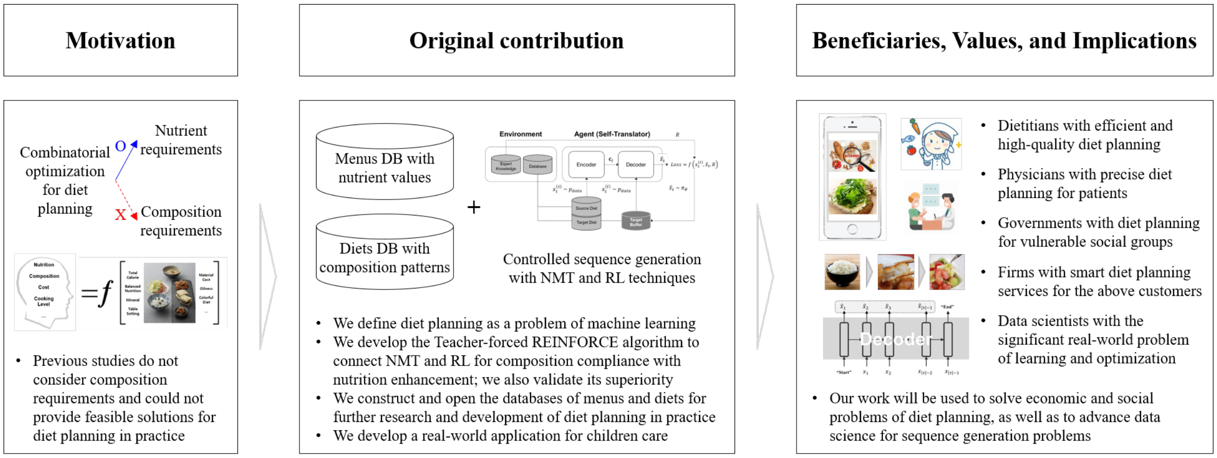
1 INTRODUCTION

Diet is "the sum of foods consumed by a person or other organism" [24], and diet planning is a basic activities that humans do regularly. Given its importance to everyone from children to seniors and from the healthy to ailing, diet planning has been an important problem in nutritional science [7], the medical field [38], operations research [6], and economics [32]. Diet planning is challenging because of the knowledge required about food, culture, nutrition, and health (e.g., ingredients, nutrients, and disease) and the high design complexity associated with large numbers of food items (e.g., a combination of five menus selected from 100 food items for a daily diet plan involves approximately 10^8 cases). Previous studies have considered diet planning as a combinatorial optimization problem in which the researchers use a priori information (e.g., daily nutrient requirements) to develop the mathematical model (e.g., [9, 13]), with an exact or heuristic algorithm [11]. The main aim of this approach is to generate solutions that satisfy the recommended daily intake (RDI) of nutrients, and the approach is naturally very powerful in generating quality solutions in nutrition. However, in practice people still do not use automated diet planning services, and professional dietitians still take on the burden of manually designing customized diet plans for specific groups such as children and patients. The question then arises: why is this the case?

This is because its *composition* is very important for a diet to be accepted [22]. Such composition is implicit in nature and therefore difficult to be specified in a combinatorial optimization model. In our pilot study with 41 diet experts [20], they considered the compliance to be unexplained chemistry between menu items and to be very important in the planning and evaluation of diets. For example, when a nutritionally perfect diet with a cookie and a cup

[*] Corresponding author.

KDD '21, August 14–18, 2021, Virtual Event, Singapore.
© 2021 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-8332-5/21/08.
https://doi.org/10.1145/3447548.3467201



Motivation

- Previous studies do not consider composition requirements and could not provide feasible solutions for diet planning in practice

Original contribution

- We define diet planning as a problem of machine learning
- We develop the Teacher-forced REINFORCE algorithm to connect NMT and RL for composition compliance with nutrition enhancement; we also validate its superiority
- We construct and open the databases of menus and diets for further research and development of diet planning in practice
- We develop a real-world application for children care

Beneficiaries, Values, and Implications

- Dietitians with efficient and high-quality diet planning
- Physicians with precise diet planning for patients
- Governments with diet planning for vulnerable social groups
- Firms with smart diet planning services for the above customers
- Data scientists with the significant real-world problem of learning and optimization

- Our work will be used to solve economic and social problems of diet planning, as well as to advance data science for sequence generation problems

Figure 1: Overview and contribution of this work

of grape juice are served together as a snack, the experts pointed out that this combination is unnatural and a glass of milk should be provided instead of a cup of grape juice. Other examples of such non-compliance with menu composition include the selection of nutritionally heavy food for snacks and the use of the same ingredients in different menus (e.g., too much pork in the diet). The dietetics literature and textbooks provide a variety of evidence that both the compositional style of menus and the satisfaction of nutritional requirements are critical and depend upon the diet recipient’s food culture, meal timing, and context. Consequently, defining the composition of a diet is challenging because it is difficult to specify all the criteria involved. Additionally, the criteria considered in diet planning can vary according to the characteristics of the recipient (e.g., children or adult, diabetic patient or healthy adult), time and context (e.g., lunch or dinner), and diet culture. Unlike explicit knowledge such as the RDI of nutrients, the composition is implicit in nature and difficult to specify in a combinatorial optimization model for diet planning.

This fact suggests the necessity of *diet planning with machine learning*, which extracts the implicit composition patterns from real diet data and applies these patterns when generating diets. In this paper, we propose a diet planning method that reflects both explicit constraints (i.e., expert knowledge of nutrition) and implicit patterns (i.e., the data distribution of composition in real diet data). Specifically, we define diet planning as a problem of redesigning the reference diet into a nutritionally enhanced new diet; Diet planning can be defined as a task of neural machine translation (NMT), which maps a source sequence (i.e., the reference diet) to the target sequence (i.e., a nutritionally enhanced diet). As such, the learning framework of our method works as follows. First, we built a self–translation architecture, which uses the same diet for both the source and target sequences to capture the implicit patterns in real diet data (i.e., to embed the data into a new representation space for generation). Next, we applied reinforcement learning (RL) to optimize the self–translator in terms of the explicit constraints of nutrient RDI. Lastly, we developed a novel training algorithm, *the teacher-forced REINFORCE*, that connects the self–translator with the RL. As a result, the controlled self–translator learns to generate nutritionally enhanced diets from source diet. We demonstrate the validity of our work through a real-world application of planning diets for children. We further present the results of a comparative experiment with existing controlled generation methods [27, 28].

This work is original research that presents the first data-driven approach to diet planning (see Figure 1). Its academic contribution is to connect the modern machine learning literature with the traditional diet planning literature (see Section 2). We successfully defined data planning as a machine learning problem and developed a problem-solving framework (see Section 3). The methodological contribution of our work was validated through a comparative experiment with existing methods (see Sections 4 and 5). Furthermore, our work has a real impact because it is being implemented in an application service for professional dietitians and pediatricians and will be used in the local government Centers for Children’s Food Service Management in South Korea in the fall of 2021. Note that the databases of foods and diets used in this research were developed by professional dietitians over a year using the related public databases disclosed by the government. We have made these databases and our algorithms publicly available to promote future research and development of diet planning with machine learning (see Section 4.1).

2 BACKGROUND AND LITERATURE REVIEW

2.1 Classical Diet Problem

In 1945, George Stigler (who later received the Nobel Prize) published an article that presented a model for the most economical diet [32]. Stigler, who worked in the pre-linear programming era, used a set of (9 × 77) linear constraints to determine the least-expensive diet that would meet the nutrient requirements. Later in 1963, George Dantzig, who is the pioneer of optimization theory and widely celebrated as the "father of linear programming", introduced the Stigler’s nutrition model in his book [6] and highlighted it as "The Diet Problem". As a result, most related studies took a mathematical programming approach to determine the optimal combination of quantity for the given food items under constraints such as nutrition, raw food cost, and production time.

2.2 Diet Planning

A review [31] of linear programming solutions for human diets concluded that they provide unpalatable diets. Smith [30] noted that such unpalatability comes from a model that is formulated to plan "one-dish meals", which fits more with animal feed blending than human diet, and Peryam [25] commented that the diets planned by linear programming are not acceptable because they disregard traditional menu patterns. Above all, a human diet consists of menu items, not food items, and we consume it in units of end products, not raw materials.[1] Therefore, the study of determining an optimal combination of menus through mixed integer programming (MIP), so-called "diet planning", emerged [29].[2] Eckstein [10] described that ’diet planning’ is menu–item level planning for humans, and it must include the constraints of the diet problem and consider psychological parameters (i.e., implicit menu patterns) such as color, texture, shape, and flavor.

Specifically, the nutrition literature and textbooks further highlight the significance of considering implicit menu patterns in diet planning. The recipients will not accept a diet, if the food culture or eating habits for the composition of menus are not complied, no matter how nutritionally balanced the diet is. Particularly in the context of food services, it is crucial to consider the eating habits, preferences, style of menus, and nutritional quality in diet planning. As such, dietitians learn the value of comprehensively considering the contexts of menu patterns and styles in planning healthy diets to ensure acceptance. As an example, the Academy of Nutrition and Dietetics in the US defines the role of dietitians as encouraging diet recipients to accept and abide by menus of high nutritional quality [3]. Similarly, Korean dietitians are trained to consider the compositional criteria of Korean diet planning, including the chemistry of the ingredients, harmony of the colors, diversity of seasoning, and complexity of cooking [18], all of which are very difficult to specify in a mathematical model but can be accommodated with machine learning.

2.3 NMT

The three sub-sections below describe the preliminaries of the proposed machine learning framework for diet planning. A daily diet consists of meals (e.g., lunch and dinner), with each meal having a specific menu pattern. This implies that diet composition depends on menu items being arranged by meal times and their co-occurrence in each meal. Accordingly, we address the diet as a sequence and leverage a machine translation technique to generate a diet based on a sequential pattern of menu items. NMT addresses the task of translating a sequence in a source domain (e.g., English) into the corresponding sequence in a target domain (e.g., French) in a single neural network. In general, NMT uses an encoder–decoder structure where the implicit pattern of a source sequence is encoded into a latent feature and the decoder decomposes it to generate a target sequence. This has been applied in various fields such as sequential recommender systems [21] or for designing molecular sequence structures [14].

2.4 RL

Despite the importance of composition, the essence of diet planning is still to achieve the required nutrition. NMT alone cannot achieve this because it only focuses on learning the implicit patterns of a sequence. Thus, we use RL to control NMT and give it an explicit constraint (i.e., the RDI) as a reward, which enables the network to generate a diet with enhanced nutrition. RL is a learning framework in which an agent is trained via interaction with the environment. To describe the interactive framework mathematically, the task of RL is formulated as a Markov decision process (MDP). Given the current state s_t, the agent determines an action a_t and learns the optimal policy π^* that maximizes the total reward $\sum_{t=0}^{T}\gamma^t r(s_t, a_t)$. Note that γ, where $0 \le \gamma \le 1$, is the discount factor that weights the short-term future more, and $r(s_t, a_t)$ is a reward function explicitly given by the environment. The MDP is formulated as:

$$\max_{a_t} \mathbb{E}_{a\sim\pi}[\sum_{t=1}^{T} \gamma^t r(s_t, a_t)] \tag{1}$$

2.5 Controllable Sequence Generation

Tuning a generator to assign a specific property to the generated sequences is known as controllable sequence generation [16]. Because our work aims to generate diet sequences and modifies them to optimize nutrition, the present study can be thought of as a type of controllable sequence generation. Therefore, we will compare our method with the well-known baselines of controllable sequence generation in the experiment of Section 4.

3 METHOD

The diet-planning method proposed in this study combines NMT with RL because NMT captures implicit patterns and RL reflects an explicit constraints. This section provides an overview of the proposed method and describes the details of the sub-components, as illustrated in Figure 2.

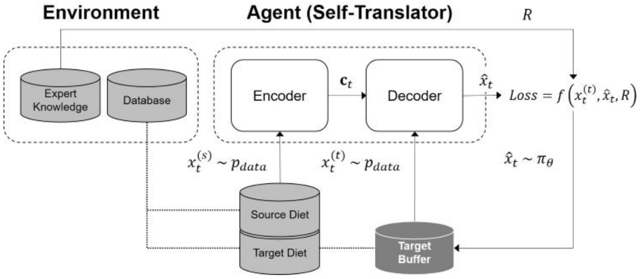


Figure 2: Framework of the proposed method

3.1 Diet Generator

Self–translator. Given a sequence $\mathbf{x} = [x_0, .., x_T]$, the self–translator inputs the sequence as both source and target sequence. That is, the self translator directly models the reconstructive translation probability as

$$p_\theta\left(\mathbf{x}^{(t)}|\mathbf{x}^{(s)}\right) = \prod_{t=1}^{T} p_\theta\left(x_t^{(t)}|x_{0:t-1}^{(t)}, \mathbf{x}^{(s)}\right)$$

[1] Menu items are mixtures of food items. For example, *shrimp soup* is a menu item or recipe that consists of raw food materials such as shrimp, salt, flour, and butter.

[2] The diet planning involves the optimization in both continuous space (e.g., 0.55 g or 150.75 g) and discrete space (e.g., 0 units or 3 units), thereby using MIP instead of linear programming.

where $\mathbf{x}$ represents a diet sequence of fixed length $T = 16$. Vector $\mathbf{x}$ is a vector in which the t-th element is a token $x_t \in \mathcal{M}$. The first and last elements, x_0 and x_T, are filled with "start" and "end" tokens respectively, whereas the 14 elements in between are filled with menu item tokens (e.g., "*scrambled egg*"). Here, $\mathcal{M}$ is a set of all menu items.

The self–translator is a single encoder–decoder neural network. Source sequence $\mathbf{x}^{(s)} = x_{0:T}^{(s)}$ is fed into the encoder and target sequence $\mathbf{x}^{(t)} = x_{1:T}^{(t)}$ is fed into the decoder. This self–translation approach works similarly to an autoencoder [15]. Consequently, the latent feature is automatically embedded through self–translation and helps to generate realistic sequences. Note that $\mathbf{x}^{(s)} = \mathbf{x}^{(t)}$.

Encoder–Decoder Network. The network of the encoder-decoder is built based on the gated recurrent unit (GRU) [5]. The encoder–decoder network is parameterized by θ and predicts a token x_t conditioned on the hidden state $\mathbf{h}_t$ at each time step t:

$$\mathbf{h}_t = \sigma\left(W_x \mathbf{1}(x_{t-1}) + U_h \mathbf{h}_{t-1} + V_c \mathbf{c}_t + b_h\right) \tag{2}$$

$$\mathbf{o}_t = W_o \mathbf{h}_t \tag{3}$$

$$x_t \sim \text{softmax}(\mathbf{o}_t) \tag{4}$$

where $\mathbf{h}_t$ summarizes the history of translation $x_0, x_1 ..., x_{t-1}$. The output vector $\mathbf{o}_t$ is embedded through a linear projection W_o, and $\mathbf{c}_t$ is an output of the encoder, the so called context vector, which encodes a latent feature of $\mathbf{x}^{(s)}$. Note that $\mathbf{1}(i)$ is a one–hot vector of $|\mathcal{M}|$ length the i-th element of which is 1 whereas all other elements are 0. The GRU parameters θ are a set of matrices W, U, and V along with vector b. For simplicity, we write equations (2)–(4) in an abbreviated form as follows.

$$\begin{aligned} x_t &\sim p_\theta(x_t | x_{t-1}, \mathbf{h}_t, \mathbf{c}_t) = p_\theta(x_t | x_{t-1}, x_{t-2}, ..., x_0, \mathbf{x}^{(s)}) \\ &= p_\theta(x_t | x_{0:t-1}, \mathbf{x}^{(s)}) \end{aligned} \tag{5}$$

Diet Generation. We generate a diet sequence through an iterative process of token sampling based on equation (5). The generative process is optimized by learning θ, where θ is a parameter of latent feature that represents the implicit patterns in the diet. Here, θ is learned by minimizing the following cross-entropy loss (XE–loss):

$$\begin{aligned} \mathcal{L}_\theta\left(\mathbf{x}^{(t)}, \mathbf{x}^{(s)}\right) &= -\, x_t^{(t)} \log \prod_{t=1}^{T} p_\theta\left(x_t | x_{t-1}^{(t)}, \mathbf{h}_t, \mathbf{c}_t\right) \\ &= -\sum_{t=1}^{T} \log p_\theta\left(x_t^{(t)} | x_{0:t-1}^{(t)}, \mathbf{x}^{(s)}\right) \end{aligned}$$

where $x_t^{(t)} \sim p_{data}$ and $x_t \sim p_\theta$. Here, p_{data} denotes the data distribution whereas p_θ indicates the parameter distribution of the self–translator. We omit $\mathbf{x}^{(s)}$ for the sake of simplicity, and the self–translator is optimized by computing the gradient of the XE–loss as follows:

$$\nabla_\theta \mathcal{L}_\theta\left(\mathbf{x}^{(t)}, \mathbf{x}^{(s)}\right) = -\sum_{t=1}^{T} \nabla_\theta \log p_\theta\left(x_t^{(t)} | x_{0:t-1}^{(t)}\right) \tag{6}$$

Attention Mechanism. During diet generation, the encoder summarizes the implicit pattern into a global latent feature. The global latent feature contains the overall composition of diet but can forget local compositions because of an information bottleneck. Even though the overall composition alone can be used to generate realistic diets, a local composition, such as "*cookies go better with milk than grape juice*" should receive attention to ensure the compositional naturalness of diets. Therefore, we used the attention mechanism [1, 5] that aligns a global feature with multiple local features that vary with each t-th menu item.

3.2 Controlled Diet Generation

RL Formulation. In the previous section, we defined diet planning as a translation task where a self–translator predicts each token consecutively and generates a compositionally feasible diet sequence. To force the self–translator to generate a nutritionally enhanced diet, our framework uses RL and controls the generative process. Accordingly, we assume that the self–translator is an agent who executes diet generation (i.e., diet planner) and the generative process is an MDP that consists of

- an action x_t: the menu–item token that an agent samples at time step t;
- a state $x_{0:t-1}$: the history of all menu items sampled over the previous time steps, i.e., a partially generated diet sampled up to the $t-1$-th token;
- a reward function $r(x_{t-1}, x_t)$: the function that returns a numerical value R that measures the nutrition score (RDI score) of diet at time t;
- a policy $\pi_\theta(x_t | x_{1:t-1})$: the distribution of candidate menu items, i.e., an estimated probability of x_t to be sampled given $x_{1:t-1}$.

Then according to equation (1), the objective function $J(\theta)$ becomes:

$$\begin{aligned} J(\theta) &= \mathbb{E}_{x \sim \pi_\theta}\left[\sum_{t=1}^{T} \gamma^t r(x_{t-1}, x_t)\right] \\ &\approx \sum_{\tau \sim \pi_\theta}\left[\sum_{t=1}^{|\tau|} \pi_\theta(x_t | x_{0:t-1}) R(\tau)\right] \end{aligned} \tag{7}$$

where τ is the trajectory of token samplings. Generally, R is the discounted future reward at t. In our case, however, the agent is the diet planner who designs nutritionally quality diet. Therefore, the agent observes R at the end of sampling when the "end" token $x_{|\tau|}$ is sampled, because the nutritional value is calculated based on the RDI at the level of diet, not the level of menu items, that is, $\tau = (x_0, x_1, ...)$, $|\tau| = T$, $t \to T$, $x_t \sim \pi_\theta$, $R(\tau) := \text{RDI}(\tau)$. Defined by the nutrition literature and professional dietitians, we are given RDIs of 13 essential nutrients and defined the number of requirements achieved as nutrition score R. See the Appendix for the details of RDI criteria.

Policy Gradient with REINFORCE. Equation (7) models a generative process with menu item sampling – the agent observes a part of diet $x_{1:t-1}$, predicts the next menu item, and samples the predicted menus x_t according to π_θ. The agent is rewarded once sample is completed. That is, the sampling process generates a diet sequence by stacking the sampled tokens. We control the generative process using the REINFORCE algorithm [34] to maximize the nutrition of a generated diet sequence. REINFORCE is a Monte Carlo version of an on-policy method and computes the following expected policy gradient:

$$\begin{aligned}\nabla_\theta J(\theta) &= \sum_{\tau\sim\pi_\theta}\left[\sum_{t=1}^{|\tau|} R(\tau)\nabla_\theta \log \pi_\theta(x_t|x_{0:t-1})\right] \\ &= \mathop{\mathbb{E}}_{\tau\sim\pi_\theta}\left[R(\tau)\nabla_\theta \log \pi_\theta(\tau)\right]\end{aligned} \tag{8}$$

which is equal to the XE–gradient (Equation (6)) except that there is no negative sign and R is added.[3] If we multiply the expected policy gradient by -1 and execute a gradient descent, then the policy gradient optimizes the R-weighted XE–loss using gradient ascent as:

$$\theta \leftarrow \theta - \alpha\left(-\nabla_\theta J(\theta)\right) = \theta + \alpha\nabla_\theta J(\theta) \tag{9}$$

In short, we can simultaneously optimize the training of the self–translator and control of the generative process via the REINFORCE algorithm.

Teacher-Forced Correction. As equation (8) contains both the XE–loss and the reward, optimizing the policy gradient seem adequate for producing a realistic diet with enhanced nutrition. Indeed, many previous studies [17, 28, 37] initialize the generator with a pre-trained policy and update it using the policy gradient to maximize rewards. However, the generative process is so fragile that whenever we try to control the process, and unrealistic diet can be generated. For example, from the perspective of composition compliance, it would be a principle that "*spaghetti with tomato sauce*" would go better with "*a fruit salad*" than "*salmon sushi,*". However, from the perspective of nutrition enhancement, the agent can mistakenly sample out-of-principle menu items while maximizing rewards and generate an unrealistic diet resulting in the collapse of the generative process.

The collapse is caused by the accumulation of errors and this is usually due to the on-policy method. In an on-policy method, we must predict a token and use it as the next input (see Figure 3(a)). This recursive mechanism gives rise to high sampling bias and error accumulation [2, 27, 37]; therefore it is hard to avoid collapse. To overcome collapse, we propose using off-policy correction underlying the concept of *teacher-forcing* [35] which is a technique where the ground-truth target of the current step is used as the next input. In other words, we force an agent to learn policy from the target data, not from its own predictions. As a result, policy π_θ should approximate data distribution p_{data}, which in turn prevents collapse.

As in [4, 8], we apply the likelihood ratio trick and introduce the importance weight to implement the teacher-forcing technique within the REINFORCE algorithm. Consider the teacher-forced trajectory $\tau = (x_0, x_1, ...)$, which represents the sampling of the menu items from the target diet. The teacher-forced off-policy gradient then becomes

$$\begin{aligned}\nabla_\theta J^{\mathbf{TF}}(\theta) &= \sum_{\tau\sim p_{data}}\left[\sum_{t=1}^{|\tau|} \beta(\hat{\tau})R(\tau)\frac{\pi_\theta(\tau)}{p_{data}(\tau)}\nabla_\theta \log \pi_\theta(x_t|x_{0:t-1})\right] \\ &= \mathop{\mathbb{E}}_{\substack{\tau\sim p_{data}\\ \hat{\tau}\sim\pi_\theta(\tau)}}\left[\beta(\hat{\tau})R(\tau)\nabla_\theta \log \pi_\theta(x_t|x_{0:t-1})\right]\end{aligned} \tag{10}$$

where

$$\begin{aligned}\frac{\pi_\theta(\tau)}{p_{data}(\tau)} &= \frac{\prod_{t=1}^{|\tau|}\pi_\theta(x_t|x_{0:t-1})}{\prod_{t=1}^{|\tau|}p_{data}(x_t|x_{0:t-1})} \\ &= \prod_{t=1}^{|\tau|}\frac{\pi_\theta(x_t|x_{0:t-1})}{p_{data}(x_t|x_{0:t-1})} \approx \pi_\theta(x_t|x_{0:t-1})\end{aligned}$$

is the importance weight. In general, the importance weight leads to large variance as the product chain exponentially increases (or decreases) the value of the gradient. We reduce the variance by removing the product chain using a first-order approximation [26]. Moreover, we can ignore p_{data} and only consider π_θ, because τ is a constant sequence of the target diet and p_{data} is constant accordingly. By sampling menus from τ, π_θ approximates around the fixed data distribution and prevents collapse. Meanwhile, $\pi_\theta(x_t|x_{0:t-1})$ is not explicitly computed in equation (10). Instead, we obtain its expected value by sampling the predictions from π_θ and consider it by computing β: $\mathbb{E}[\hat{\tau}|\hat{x}_t \sim \pi_\theta, t \to |\tau|]$ and $\beta(\hat{\tau})$; we discuss the details of this approach in the next paragraph.

Policy Space Expansion. An important limit of teacher-forcing is that the targets must be constant. This implies that the policy loses generality and the agent cannot generate a diet beyond the real diet. Because the generalized policy can be obtained by expanding the policy space, we introduce two techniques. First, we propose additional score β, which is defined as

$$\beta(\hat{\tau}) = \frac{1}{|\hat{\tau}|}\sum_{\hat{x}_t\sim\hat{\tau}} I(\hat{x}_t) \qquad \text{where} \quad I(\hat{x}_t) = \begin{cases}1, & \text{if } \hat{x}_t \in \mathcal{A}_t \\ 0, & \text{otherwise}\end{cases}$$

Here $I(\cdot)$ is the indicator function that returns 1 if each predicted menu $\hat{x}_t$ is the substitute of target menu x_t and $\mathcal{A}_t$ is a set of substitutes defined by the incidence matrix of the graph.[4] Score β gives the average score within the range $[0, 1]$ and measures the significance of the predicted menus. Moreover, β depends on diets generated from the agent's policy. This means that β drives the policy gradient to be updated in terms of predictions as if the agent is executing on-policy learning or exploration. Consequently, the teacher-forced policy gradient is weighted by β, and agent expands the search space of the policy by exploring the trajectories of alternative diets.

As a second technique for expanding the policy space, we propose replacing the target diet with the generated one. The algorithm proceeds as follows: i) generate the synthetic diets, ii) store the most recent N synthetic diets in target buffer $\mathcal{B}$, and iii) randomly select new target diets from $\mathcal{B}$ every M epochs. A generated diet is

[3] To be accurate, the policy gradient is the expected value of the reward-weighted XE–gradient with respect to trajectories $\mathbb{E}_{\tau\sim\pi_\theta}\left[R(\tau)\mathcal{L}_\theta\left(\mathbf{x}^{(t)}, \mathbf{x}^{(s)}\right)\right]$

[4] For example, say we have diet A and diet B in which the t-th menu items are "*spaghetti with tomato sauce*" and "*shrimp garlic pasta,*" respectively. In this case, they are nodes linked to each other with the t-th edge. In other words, they are neighbor nodes in subgraph A_t and thereby assumed to be able to replace each other.

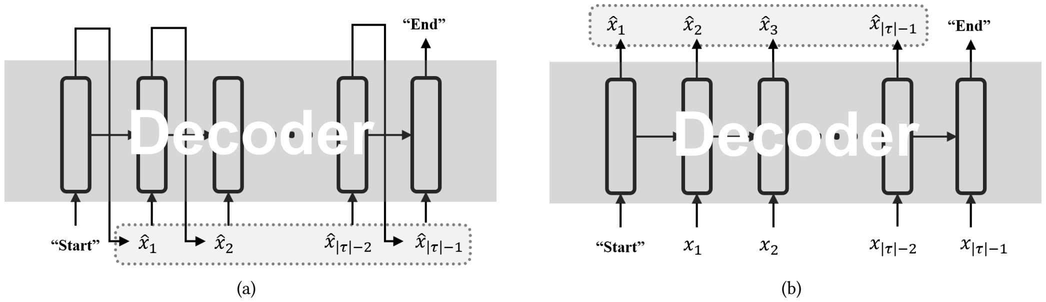


Figure 3: Comparison of the predictive process. (a) The on-policy method uses its prediction $\hat{x}_t$ as an input. (b) The teacher–forced correction uses the target as input x_t. The gray dotted soft rectangle indicates the generated diet, i.e., a collection of consecutively predicted menu–item tokens.

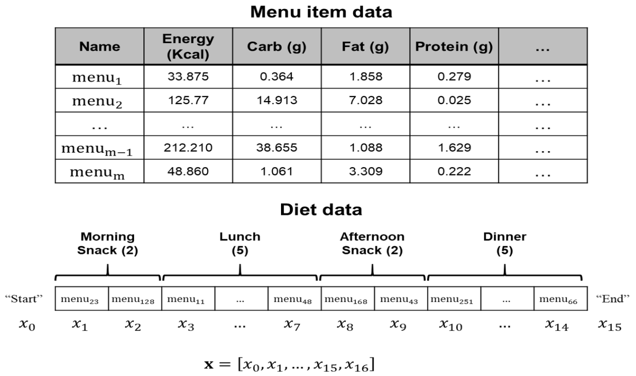
```
Algorithm 1: Teacher-Forced REINFORCE algorithm
Result: diet generator optimized w.r.t composition and nutrition.
Data: source diet sequence $\mathbf{x}^{(s)}$, target diet sequence $\mathbf{x}^{(t)}$
Initialize parameters $\theta$, training epoch $K$, buffer size $N$,
  buffer $\mathcal{B} = \left[\mathbf{x}_1^{(t)}, .., \mathbf{x}_n^{(t)}, ..\mathbf{x}_N^{(t)}\right]$ with $n = 1$, and epoch of
  target update $M$
for $k = 0$ to $K$ do
    for $t = 0$ to $|\tau|$ do
        Predict each menu token $\hat{x}_t \sim \pi_\theta$
        Generate diet $\hat{\mathbf{x}} = [\hat{x}_1, \hat{x}_2, ..., \hat{x}_{|\tau|-1}]$
        Compute reward $R$ and additional score $\beta$
        Compute policy gradient $\nabla_\theta J^{\mathrm{TF}}(\theta)$ by Equation (10)
        if $\beta == 1$ then
            if $n \le N$ then
                $n = n + 1$
                Add $\hat{\mathbf{x}}$ into $n$-th element of $\mathcal{B}$
            else
                $n = 1$
                Add $\hat{\mathbf{x}}$ into $n$-th element of $\mathcal{B}$
    end
    Update $\theta$ by Equation (9)
    if $k \% M == 0$ then
        // At every M epoch
        Replace target $\mathbf{x}^{(t)}$ with $\hat{\mathbf{x}}$ randomly selected from $\mathcal{B}$
end
```

stored only if it has the highest additional score, i.e., $\beta = 1$, otherwise the ground-truth diet is restored automatically. Consequently, the target buffer gradually changes from the initial state, which contains only the ground-truth targets, to a state that includes synthetic targets. This augments the size of the target data, expands the policy space, and thereby the agent generates diets beyond the real diets. Figure 3(b) illustrates our method, and Algorithm 1 provides the pseudocode for executing the teacher-forced REINFORCE algorithm along with the proposed techniques.

4 EXPERIMENT

4.1 Problem and Data

Problem. In South Korea, most day care centers rely on the local government's Center for Children's Food Service Management for diet planning. However, the dietitians employed in the government centers or day care centers are burdened with designing diet plans because of the aforementioned complexity of diet design [23]. In addition, they have other duties such as monitoring the cooking and hygiene status, as well as budget management. Our work was initiated to solve this problem and help the dietitians efficiently design high-quality diets for children. Note that the databases of menus and diets used in this research were developed by professional dietitians based on the public databases disclosed by the government.


Menu item data

Name	Energy (Kcal)	Carb (g)	Fat (g)	Protein (g)	...
menu_1	33.875	0.364	1.858	0.279	...
menu_2	125.77	14.913	7.028	0.025	...
...	...	...	...	...	...
menu_{m-1}	212.210	38.655	1.088	1.629	...
menu_m	48.860	1.061	3.309	0.222	...

Diet data

	Morning Snack (2)		Lunch (5)			Afternoon Snack (2)		Dinner (5)			
"Start"	menu_{23}	menu_{128}	menu_{11}	...	menu_{48}	menu_{168}	menu_{43}	menu_{251}	...	menu_{66}	"End"
x_0	x_1	x_2	x_3	...	x_7	x_8	x_9	x_{10}	...	x_{14}	x_{15}

$\mathbf{x} = [x_0, x_1, ..., x_{15}, x_{16}]$

Figure 4: Example of data in the menu and diet databases

Data. The menu item database consists of 3228 rows and 20 features. Each row represents one menu item and the features represent 20 nutrients, e.g., energy, carbohydrates, and fats. Each value indicates the nutritional content of a standard serving size for each menu item. The diets database contains 1503 diets. We removed 431 partial diets, e.g., a diet that provides lunch only, and used the remaining 1072 diets. Each diet had a sequence length of 16 and consisted of chronologically arranged menu tokens. We paded the sequence with an "empty" token if the diet was partially provided. Figure 4 illustrates both databases. The databases are publicly available and can be accessed at our repositories.[5]

4.2 Baselines and Evaluation Metrics

In this section, we describe the baselines and metrics. Three baselines are used, Cbc solver [12], self–critical sequence training (SCST) [28], and mixed incremental cross-entropy reinforce (MIXER) [27]; one is the MIP-based and the others are machine learning-based. These were evaluated on five metrics: meal–hit rate, dish–hit rate, RDI score, overall score, and Turing score.

Baselines. We used three baselines. The first baseline is *Cbc solver.* It is the most popular open-source MIP solver, which finds an optimal combination of menu items, producing the nutritionally perfect diet. We executed it using JuMP in Julia. Meanwhile, SCST and MIXER were originally developed for text generation and are now widely used to benchmark controlled sequence generation. Rennie et al. [28] proposed *SCST*, a variation of the REINFORCE algorithm for the task of image captioning. By using the difference in reward between the training and test algorithms, SCST enables the training to be self-critical and avoids inconsistency between the training and generative phases. Ranzato et al. [27] proposed a similar sequence level training algorithm based on REINFORCE, called *MIXER*, and applied it to the task of text summarization, translation and image captioning. Using curriculum learning, the algorithm starts with XE–loss and incrementally applies RL–loss from the end token to the first token at every predefined epoch.

Metrics. To evaluate the baselines and our method, we used five metrics, the two are as of composition, the one is as of nutrition, and the rest is as of overall. The composition-related metrics are *Meal-hit rate* and *Dish-hit rate*. As the data in Figure 4 shows, A diet sequence consists of time-arranged meals, and the menu items in each meal are arranged again at dish level.[6] With the help of professional dietitians, each menu was labeled with meals (e.g., lunch) and dishes (e.g., main dish). Then, we evaluated how consistent the generated diet was using these labels. The *RDI score* was used as the nutrition-related metric. This measured how nutritious the generated diets were. Last, the reliability-related metrics, e.g., *Overall score* and *Turing score*, were rated by humans. Unlike composition and nutrition, reliability is abstract and complex to be measure. As such, this was measured by humans, professional dietitians. We distributed a survey to 36 professional dietitians with an average job experience of 6.4 years, and requested them to evaluate the real and generated diets and judge whether the diet was designed by a human or by machine. See the Appendix for the details of this evaluation.

5 RESULT AND FURTHER ANALYSIS

In this section, we demonstrate that the teacher-forced REINFORCE (TFR) generates a synthetic diet that looks realistic and enhances nutrition.

5.1 Implementation Detail

Every machine learning model evaluated here has an identical architecture that consists of four layers: i) an embedding layer, ii) a fully-connected layer, iii) an attention layer, and iv) a softmax layer. The size of the layer is fixed at 128 dimensions in the embedding layer and 64 dimensions in both the fully-connected and attention layers. The models were trained for 10k epochs and used the Adam optimizer with a learning rate of 1×10^{-3}. The model-specific parameters were determined based on existing studies and parameter experiments. For example, SCST was pretrained with scheduled sampling [2] with a probability of 5×10^{-5} and annealed using a factor of 0.05 until the sampling probability increases to 0.25 [28]. After 50% of the total epoch progresses, MIXER executed a single curriculum learning in the period of 30 epochs, and the total curriculum learning restarted every 450th epoch.[7] In the case of TFR, we executed a parameter experiment and found that the buffer is slowly updated when the buffer size is smaller than the update epoch ($N \leq M$). Therefore, we fixed the parameters as $N = M = 5$.

5.2 Empirical Comparison

Figure 5 demonstrates that TFR outperformed the machine learning-based baselines (SCST and MIXER) both in the training and generation phases. Figure 5(a) shows that the training reward (i.e., nutrition level) of TFR consistently increased and achieved the highest result after 8k epochs. Although SCST performed best until 7k epochs, its training was unstable and the rewards fluctuated. The training of MIXER was stable but its reward increased very slowly and even failed to reach the average nutrition level of real diets (red line). Meanwhile, a higher reward in the training phase does not guarantee nutritionally realistic diets. Accordingly, we generated synthetic diets and transformed these into vectors with 20 nutrient dimensions. Then, the vectors were mapped into a two-dimensional embedding space using the t–SNE technique [33]. Figure 5(b) illustrates that TFR generated diets nutritionally similar to the real diets, whereas SCST generated unrealistic diets, which might be attributed to the collapse of the generative process. Figure 5(c) illustrates that TFR reproduces a realistic distribution of nutrient and succeeds in enhancing nutrients (e.g., dietary fiber, vitamin A, and calcium). See Table 5 in Appendix to check the generated outcomes.

Table 1: Summary of the evaluation results

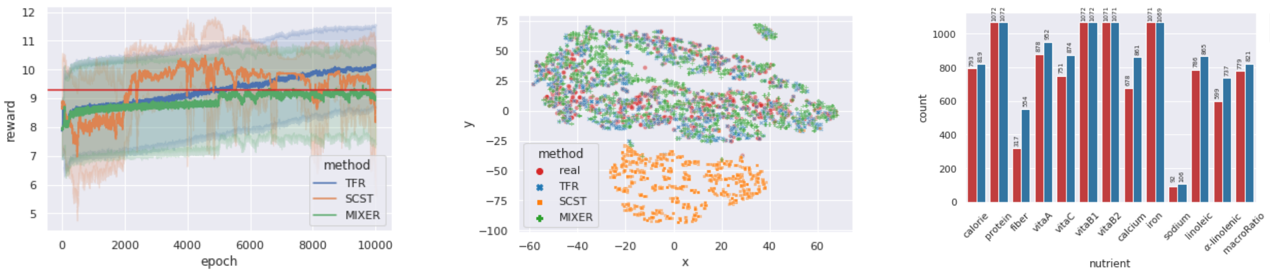
	(Composition)		**(Nutrition)**	**(Reliability)**	
Method	*Meal-hit rate*	*Dish-hit rate*	*RDI score*	*Overall score*	*Turing score*
real	1.00	0.97	9.29	3.76	0.82
Cbc solver	0.60	0.22	**13.00**	2.29	0.36
SCST	0.78	0.75	7.86	1.13	0.12
MIXER	0.84	0.81	9.16	3.12	0.64
TFR	**0.99**	**0.96**	10.10	**3.41**	**0.73**

Table 1 summarizes the results for the five metrics described in Section 4.2. We compare TFR with real diets and the three baselines. The five metrics represent the types of evaluation: composition,

[5] https://github.com/Leo-Lee92/Diet-Generation-As-Sequence

[6] For example, the third and seventh menu items, i.e., x_3 and x_7, must respectively be the main and side dish of lunch.

[7] Diet sequences are of length 16 with the first token "start" given, thus MIXER predicts up to 15 tokens with the RL–loss. Because a single curriculum runs for 30 epochs for each token, $30 \times 15 = 450$ epochs are required for full curriculum learning across the entire sequence



(a) (b) (c)

Figure 5: Comparison of the performance in the training and generation phases. (a) Rewards (nutrition level) of the predicted diet in the training phase, where the red line indicates the average nutrition level of real diets. (b) t-Distributed Stochastic Neighbor Embedding (t–SNE) nutrient map of the generated diets. (c) Comparison of nutrient distribution between real and TFR–generation.

nutrition, and reliability. To evaluate reliability, we requested 36 professional dietitians to evaluate 100 diets of which 80 were synthetic diets (20 diets respectively generated by each method) and the rest 20 were given as real diets (i.e., 20 diets randomly sampled from the diets database). Note that the perfect RDI score and the maximum overall score are 13 and 5 points respectively. Results demonstrate that TFR generated highly realistic diets and enhanced nutrition, which shows that TFR overcame the collapse of the generative process and succeeded in controlling diet generation. Moreover, TFR achieved the highest reliability score. This means that dietitians recognized the quality of TFR-generated diets as acceptable (by overall score) and found these diets natural (by Turing score). The superiority of our TFR over Cbc solver, MIXER, and SCST shows its great potential to assist professional dietitians in diet planning. See Appendix C for the details of expert evaluation.

Note that SCST, more recent one, received a lower score than MIXER. Unlike MIXER, which uses both on-policy and off-policy methods, SCST is entirely an on-policy method. Thus, its performance can degenerate, suffering from collapse of the generative process. Meanwhile, Cbc solver generated a diet of perfect nutrition as expected. However, the alignment of menu items were not feasible at all and should be organized by humans in the form of diet. Thus, we manually organized Cbc-generated diets for expert evaluation, which resulted in a higher reliability score for Cbc solver than SCST. Otherwise, Cbc solver would have received the lowest score for reliability.

5.3 Reward Shaping

One of the advantages of RL is that rewards can be changed. Customizing rewards to obtain intended results is termed reward shaping. As such, one may ask whether highly reliable diets can be generated by designing object-oriented rewards. In response to this question, we also tried a reward shaping by multiplying composition-related metrics with the original reward, RDI score, similar to the case of [36] and re-trained TFR, SCST, and MIXER again. Figure 6 in the Appendix illustrates the result. Interestingly, multiplying the composition-related metrics by the original reward does not improve the meal–hit rate and dish–hit rate, rather it degenerates the performance, especially in SCST. This result could be attributed to the reduction of rewards. The hit–rate and dish–rate range within $[0, 1]$ and multiplying these with rewards leads to reward reduction and suppresses the policy gradient and disturb the learning.

Although this experiment is just an example, the result may imply the robustness of our approach that separates clearly the explicit and implicit requirements in learning; the former requirements are reflected by using rewards, whereas real data are utilized to extract and apply the latter. While reward shaping for implicit requirements may involve trial and errors of designing rewards, our approach does not necessitate such a process of reward shaping; while reward shaping can be case-sensitive, our approach seems robust to the reward design and can use simple reward functions immediately. Nonetheless, reward shaping for explicit requirements is necessary in some specific applications. For example, when diet planning must be customized (e.g., for children, vegetarians, or allergic patients) and nutrient rewards can be designed according to the target group. The proposed TFR algorithm can be flexibly adapted to accommodate such different requirements of users.

6 CONCLUDING REMARKS

This work originally defines diet planning as a machine learning problem (i.e., controllable sequence generation) and develops a successful solution to address this problem. Our algorithm and databases [19] will be embedded in a real diet planning service system that currently is being developed with support from the Korean government. [8] This service will be operated by a healthcare service startup and will support dietitians in day care centers as well as pediatricians in hospitals efficiently to plan high-quality diets for child patients. Diet planning is a routine but difficult task because of the growing body of knowledge regarding health, and the high complexity of design associated with large numbers of menus. This work is the first to solve this problem and offer full assistance to

[8] This work was supported by the Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No.2020-0-02135: Development of a gut microbiome-personalized diet recommendation AI system for the children with atopic diseases, No.2020-0-01336: Artificial Intelligence Graduate School Program - UNIST) and supported by the Bio & Medical Technology Development Program of the National Research Foundation (NRF) funded by the Ministry of Science & ICT (grant number 2019M3E5D1A02070867).

experts, thus contributing to alleviating related economic problems (e.g., only a few day care centers can recruit professional dietitians) and social problems (e.g., patients and vulnerable social groups need special attention to manage their nutritional levels).

Furthermore, this work will advance applied data science for controllable sequence generation problems. First, the proposed TFR algorithm is a novel machine learning method for sequence generation that considers implicit patterns of the tokens in sequence (e.g., compositional patterns in diets) under explicit constraints (e.g., nutritional requirements). In this case, both explicit and implicit features should be considered in generation. Second, from the perspective of combinatorial optimization, this work offers implications to consider interrelationships of the elements (e.g., menus) being timely arranged. Third, most existing studies that apply RL to control sequence generation belong to the domain of natural language processing. They use well-defined metrics such as CIDEr that explicitly measures how reliable a sentence is. When this metric is given as reward, a sequence generator is explicitly controlled to generate realistic sentences. However, in many other problems of controllable sequence generation (e.g., recommendation list generation and molecular structure generation), they usually do not have such a well-defined metric so a reward is designed from different perspectives. Accordingly, the generative process often collapses and produces unrealistic sequences. We experienced same issue in our research and addressed it by combining teacher-forcing with REINFORCE algorithm. We believe that much more studies on the real–world applications are necessary to further develop the controllable sequence generation techniques that address various types of sequence data in practice.

REFERENCES

[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473* (2014).

[2] Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. 2015. Scheduled sampling for sequence prediction with recurrent neural networks. In *Advances in Neural Information Processing Systems*. 1171–1179.

[3] Sara E Benjamin-Neelon. 2018. Position of the Academy of Nutrition and Dietetics: Benchmarks for nutrition in child care. *Journal of the Academy of Nutrition and Dietetics* 118, 7 (2018), 1291–1300.

[4] Minmin Chen, Alex Beutel, Paul Covington, Sagar Jain, Francois Belletti, and Ed H Chi. 2019. Top-k off-policy correction for a REINFORCE recommender system. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*. 456–464.

[5] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078* (2014).

[6] George Bernard Dantzig. 1998. *Linear programming and extensions*. Vol. 48. Princeton university press.

[7] Enza D'Auria, Erica Pendezza, and Gian Vincenzo Zuccotti. 2020. Personalized nutrition in food allergy: Tips for clinical practice. *Frontiers in pediatrics* 8 (2020).

[8] Thomas Degris, Martha White, and Richard S Sutton. 2012. Off-policy actor-critic. *arXiv preprint arXiv:1205.4839* (2012).

[9] Michele Donati, Davide Menozzi, Camilla Zighetti, Alice Rosi, Anna Zinetti, and Francesca Scazzina. 2016. Towards a sustainable diet combining economic, environmental and nutritional objectives. *Appetite* 106 (2016), 48–57.

[10] Eleanor Eckstein. 1970. Communication to the Editor—Is the "Diet Problem" Identical to the "Menu Planning Problem"? *Management Science* 16, 9 (1970), 527–528.

[11] Dušan Fister, Iztok Fister, and Samo Rauter. 2016. Generating eating plans for athletes using the particle swarm optimization. In *2016 IEEE 17th International Symposium on Computational Intelligence and Informatics (CINTI)*. IEEE, 000193–000198.

[12] John Forrest and Robin Lougee-Heimer. 2005. CBC user guide. In *Emerging theory, methods, and applications*. INFORMS, 257–277.

[13] Rozenn Gazan, Chloé MC Brouzes, Florent Vieux, Matthieu Maillot, Anne Lluch, and Nicole Darmon. 2018. Mathematical optimization to explore tomorrow's sustainable diets: a narrative review. *Advances in Nutrition* 9, 5 (2018), 602–616.

[14] Rafael Gómez-Bombarelli, Jennifer N Wei, David Duvenaud, José Miguel Hernández-Lobato, Benjamín Sánchez-Lengeling, Dennis Sheberla, Jorge Aguilera-Iparraguirre, Timothy D Hirzel, Ryan P Adams, and Alán Aspuru-Guzik. 2018. Automatic chemical design using a data-driven continuous representation of molecules. *ACS central science* 4, 2 (2018), 268–276.

[15] Geoffrey E Hinton and Ruslan R Salakhutdinov. 2006. Reducing the dimensionality of data with neural networks. *science* 313, 5786 (2006), 504–507.

[16] Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P Xing. 2017. Toward controlled generation of text. *arXiv preprint arXiv:1703.00955* (2017).

[17] Natasha Jaques, Shixiang Gu, Dzmitry Bahdanau, José Miguel Hernández-Lobato, Richard E Turner, and Douglas Eck. 2017. Sequence tutor: Conservative fine-tuning of sequence generation models with kl-control. In *International Conference on Machine Learning*. PMLR, 1645–1654.

[18] Soon Hee Kim, Myung Sunny Kim, Myoung Sook Lee, Yong Soon Park, Hae Jeong Lee, Soon-ah Kang, Hyun Sook Lee, Kyung-Eun Lee, Hye Jeong Yang, Min Jung Kim, et al. 2016. Korean diet: characteristics and historical background. *Journal of Ethnic Foods* 3, 1 (2016), 26–31.

[19] Lee. 2021. Database for Diets, Nutrition and Ingredients. https://github.com/Leo-Lee92/Diet-Generation-As-Sequence/tree/master/Data

[20] Changhun Lee, Soohyeok Kim, Jayun Kim, Chiehyeon Lim, and Minyoung Jung. 2021. Which diet design do you prefer for your children?. In *Annual meeting on American Academy of Allergy Asthma and Immunology*. http://service.unist.ac.kr/wp-content/uploads/sites/380/2021/02/AI_nutrition_2021-AAAAI-Virtual-Annual-Meeting.pdf Poster paper.

[21] Jing Li, Pengjie Ren, Zhumin Chen, Zhaochun Ren, Tao Lian, and Jun Ma. 2017. Neural attentive session-based recommendation. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. 1419–1428.

[22] MW Murimi, M Chrisman, LK Diaz-Rios, HR McCollum, and O Mcdonald. 2015. A qualitative study on factors that affect school lunch participation: Perspectives of school food service managers and cooks. *Journal of Nutrition and Health* 1, 2 (2015), 1–6.

[23] Marije Oostindjer, Jessica Aschemann-Witzel, Qing Wang, Silje Elisabeth Skuland, Bjørg Egelandsdal, Gro V Amdam, Alexander Schjøll, Mark C Pachucki, Paul Rozin, Jarrett Stein, et al. 2017. Are school meals a viable and sustainable tool to improve the healthiness and sustainability of children´ s diet and food consumption? A cross-national comparative perspective. *Critical reviews in food science and nutrition* 57, 18 (2017), 3942–3958.

[24] Oxford. [n.d.]. Diet. http://www.askoxford.com/concise_oed/diet_1?view=uk

[25] David R. Peryam. 1959. Discussion: Linear Programming Models for the Determination of Palatable Human Diets. *Journal of Farm Economics* 41, 2 (1959), 302–305. http://www.jstor.org/stable/1235156

[26] Doina Precup, Richard S Sutton, and Sanjoy Dasgupta. 2001. Off-policy temporal-difference learning with function approximation. In *ICML*. 417–424.

[27] Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. 2015. Sequence level training with recurrent neural networks. *arXiv preprint arXiv:1511.06732* (2015).

[28] Steven J Rennie, Etienne Marcheret, Youssef Mroueh, Jerret Ross, and Vaibhava Goel. 2017. Self-critical sequence training for image captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 7008–7024.

[29] David Sklan and Ilana Dariel. 1993. Diet planning for humans using mixed-integer linear programming. *British Journal of Nutrition* 70, 1 (1993), 27–35.

[30] Paul E Smith. 1961. The Diet Problem Revisited: A Linear Programming Model for Convex Economists. *Journal of Farm Economics* 43, 3 (1961), 706–712.

[31] Victor E Smith. 1959. Linear programming models for the determination of palatable human diets. *Journal of Farm Economics* 41, 2 (1959), 272–283.

[32] George J Stigler. 1945. The cost of subsistence. *Journal of farm economics* 27, 2 (1945), 303–314.

[33] Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research* 9, 11 (2008).

[34] Ronald J Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning* 8, 3-4 (1992), 229–256.

[35] Ronald J Williams and David Zipser. 1989. A learning algorithm for continually running fully recurrent neural networks. *Neural computation* 1, 2 (1989), 270–280.

[36] Lijun Wu, Fei Tian, Tao Qin, Jianhuang Lai, and Tie-Yan Liu. 2018. A study of reinforcement learning for neural machine translation. *arXiv preprint arXiv:1808.08866* (2018).

[37] Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. 2017. Seqgan: Sequence generative adversarial nets with policy gradient. In *Thirty-first AAAI conference on artificial intelligence*.

[38] David Zeevi, Tal Korem, Niv Zmora, David Israeli, Daphna Rothschild, Adina Weinberger, Orly Ben-Yacov, Dar Lador, Tali Avnit-Sagi, Maya Lotan-Pompan, et al. 2015. Personalized nutrition by prediction of glycemic responses. *Cell* 163, 5 (2015), 1079–1094.

A RDI TABLE

The RDI is the recommended daily intake (sometimes also called recommended dietary allowance; RDA). In general, the RDI is calculated based on three meals: morning, lunch, and dinner. However, the present study focuses on generating diets for children in daycare centers, which provides a diet of morning snack, lunch, afternoon snack, and dinner. As such, we asked dietitians to modify the RDI criteria to suit the purpose of this study. The table below shows the modified RDI. When a diet is given, we count the number of its achieved requirements and use this number as both the RL reward and the RDI score.

Table 2: RDI of required nutrients

No.	Required nutrient		RDI	Unit
1	Calorie		945 – 1155	kcal
2	Protein		15 – Inf	g
3	Total Dietary Fiber		8.25 – 15	g
4	Vitamin A		172.5 – 562.5	μgRAE
5	Vitamin C		26.25 – 382.5	mg
6	Vitamin B1 (Thiamine)		0.3 – Inf	mg
7	Vitamin B2 (Riboflavin)		0.375 – Inf	mg
8	Calcium		375 – 1875	mg
9	Iron		3.75 – 30	mg
10	Sodium		0 – 1200	mg
11	Linoleic Acid		3.3 – 6.8	g
12	α–Linolenic Acid		0.4 – 0.9	g
13	Macronutrient Ratio	Carb	55 – 65	kcal (%)
		Protein	7 – 20	
		Fat	15 – 30	

B OBJECTIVE FUNCTION OF MIP MODEL

In Section 4.2, we mentioned that Cbc solver is used as a baseline. In our work, we set the objective function as

$$\max \sum_{x \in \mathcal{M}} x$$

where x is a random menu variable with the value 1 if the menu is selected. This model aims to maximize the total number of menus selected under constraints such as

$$\sum_{i=1}^{N} x_i = 14 \tag{11}$$

$$945 \leq \sum_{i=1}^{N} calorie(x_i) \leq 1155 \tag{12}$$

$$15 \leq \sum_{i=1}^{N} protein(x_i) \leq Inf \tag{13}$$

$$\vdots$$

$$2 \leq \sum_{i=1}^{N} is\_snack(x_i) \leq 4 \tag{14}$$

where N is the number of menus. The constraints above should be set by humans and specify a set of feasible solutions. For example, in our dataset, a diet is defined as a sequence that maximally contains up to 14 menu tokens (except for the 'start' and 'end' tokens). Equation 11 is an explicit constraint for this point. Similarly, many constraints had to be set manually for Cbc solver to cover feasible solutions as widely as possible from nutrition (Equation 12 and 13) to composition (Equation 14).

Table 3: Result of survey

	Score of the evaluation criteria						
Planner	**1**	**2.1**	**2.2**	**2.3**	**2.4**	**3.1**	**3.2**
Human	4.13	4.02	4.0	3.94	3.75	3.76	0.82
Solver	3.23	3.23	3.09	3.14	2.59	2.29	0.36
Mixer	3.67	3.71	3.59	3.56	3.23	3.12	0.64
SCST	1.46	1.34	1.32	1.29	1.22	1.13	0.12
TFR	3.93	3.79	3.81	3.74	3.54	3.41	0.73

C EXPERT EVALUATION

36 professional dietitians evaluated the 100 diets respectively designed by Humans (experts), Solver (combinatorial optimization approach), MIXER, SCST, and TFR (machine learning approaches) evenly. Table 3 shows results per evaluation criteria described in Table 4.

The results imply the following three points. First, experts may be biased primarily toward composition satisfaction in dietary evaluation and then apply this bias when evaluating other aspects such as the nutrient and overall satisfaction. For example, the experts did not give a high nutrition score to the nutritionally perfect diets generated by Solver, which is perplexing. Second, following this example, most of the experts were not capable of precisely evaluating the nutritional quality of diets. As shown in Table 1, TFR and Solver are superior to Humans in nutritionally excellent diet planning. However, Table 3 shows that the experts could not evaluate the nutrition of diets may well be due to the limited ability of calculation and the bias mentioned above. Third, although the TFR-generated diets were judged to be compositionally less adequate than the human-designed diets, we think this result is natural because any machine-generated diet was new to the experts who were biased to the human-designed diets publicly disclosed as a reference database by the government. The important question is whether the composition of machine-generated diets is acceptable for providing actual food service. In this respect, we believe the superiority of our TFR over Solver, MIXER, and SCST shows machine learning's great potential for diet planning regard to composition. In addition, our TFR demonstrates better performance than Humans in terms of ensuring the nutritional quality of diets.

In summary, the experts considered composition compliance as the most important factor in evaluating diet, despite being incapable of accurately evaluating the nutritional quality of diets. This justifies the underlying motivation for our work to assist humans with artificial intelligence in diet planning, and the results in this paper confirm that our proposed TFR algorithm produced state-of-the-art performance.

Table 4: Form of survey

Section	No.	Evaluation criteria	Score scale
Nutrition	1.1	Does this diet satisfy the nutrition standard?	Integer between 1 and 5
Harmony	2.1	Does this diet harmonize in color?	
	2.2	Does this diet harmonize in flavor?	
	2.3	Does this diet have the texture contrast?	
	2.4	Does this diet have complementary menus?	
Overall reliability	3.1	Do you think this diet is suitable for a real food service?	
	3.2	Do you think this diet was planned by a professional dietitian? (Turing test)	Yes (1) or No (0)

Table 5: Comparison between real diet and generated diets

no	(Source Diet)	(Translated Diet)		
	Real	*TFR*	*SCST*	*MIXER*
x_1	s_strawberry	s_strawberry	s_watermelon	s_nuts
x_2	s_milk (200ml)	s_milk (200ml)	s_milk (100ml)	s_milk (200ml)
x_3	steamed millet rice	steamed white rice	s_milk (100ml)	steamed millet rice
x_4	acorn jelly soup	dried pollack soup	braised tofu in marinade sauce	tofu soy paste soup
x_5	rolled omelette with cheese	rolled omelette with cheese	s_watermelon	rolled omelette with cheese
x_6	seasoned salad with napa cabbage in soy paste	seasoned salad with napa cabbage in soy paste	cabbage soy paste soup	seasoned salad with napa cabbage in soy paste
x_7	radish kimchi cubes	radish kimchi cubes	radish kimchi cubes	radish kimchi cubes
x_8	s_soboro bun (streusel-like cursted bread)	s_steamed sweet potato	s_watermelon	s_fermented rice cake
x_9	s_barley tea	s_barley tea	s_milk (100ml)	s_barley tea
x_{10}	steamed white rice	steamed millet rice	steamed sweet brown rice	steamed black rice
x_{11}	dried pollack soup	shepherd's purse soy paste soup	cabbage soy paste soup	tofu soup
x_{12}	stir-fried chicken in soy sauce	stir-fried chicken in soy sauce	braised tofu in marinade sauce	braised chicken in teriyaki sauce
x_{13}	cucumber salad	cucumber salad	s_watermelon	dried mussel seaweed soup
x_{14}	napa cabbage kimch	napa cabbage kimch	s_milk (100ml)	napa cabbage kimch

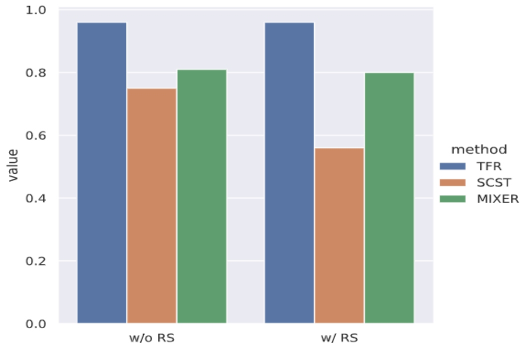


Figure 6: Result of reward shaping. The result is better with out reward shaping (w/o RS) than with reward shaping (w/ RS).