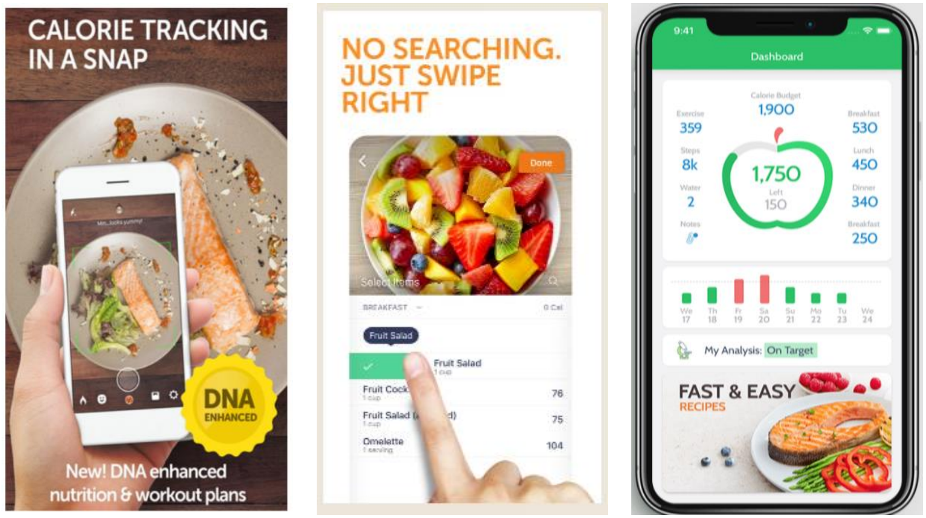
Diet Planning with Machine Learning

: Teacher-forced REINFORCE for Composition Compliance with Nutrition Enhancement

Contents

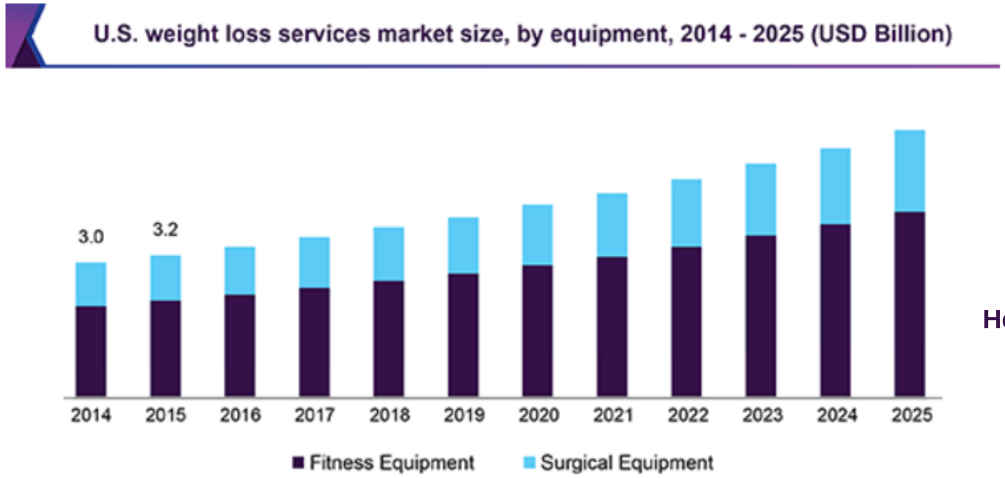
Introduction

- Diet service market



< Examples of diet planning service >

Source: www.grandviewresearch.com



Medical

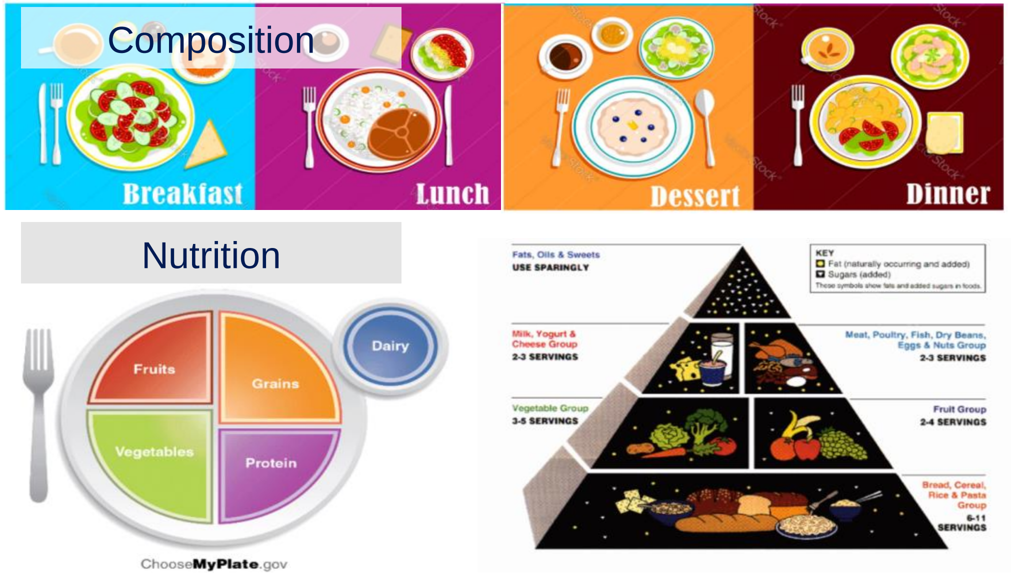
Health care

< Increase in market size of diet services >

- We can easily observe many mobile and PC applications that offers daily diet plans everywhere.
- The market size of such diet services is growing both in health care and medical sides.

Introduction

- Importance of diet design



< Basic requirements to consider in designing a diet >

< Advanced requirements to consider in designing a diet >

- Designing a diet is very important as it is deeply rooted in our lives from health (e.g., nutrition or allergy) to social and personal context (e.g., meal timing, food culture, and individual belief).
- According to such high complexity, diet design was considered a challenging task even for diet experts (e.g., nutritionists, pediatricians).

Introduction

- Prior studies: diet planning research

Source: Sklan, D., & Dariel, I. (1993). Diet planning for humans using mixed-integer linear programming. British Journal of Nutrition, 70(1), 27-35.

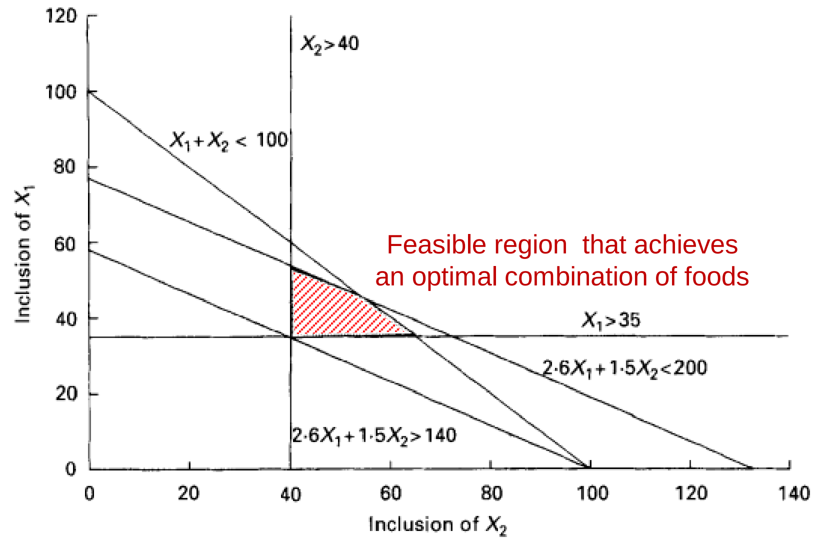


Fig. 1. Polyhedron for intake of two foods (X_1 and X_2) formed by the equations: $X_1+X_2<100$, $2{\cdot}6X_1+1{\cdot}5X_2>140$, $2{\cdot}6X_1+1{\cdot}5X_2<200$, $X_1>35$, $X_2>40$. (▨), Feasible region.

Table 1. *A simple linear program with three foods used for diets planned for humans*

	Bread	Egg	Cheese	
Amount (g)	1·0	1·0	1·0	> = 100
Energy				
kJ	10·37	6·3	6·15	> = 1672
kcal	2·48	1·5	1·47	> = 400
Protein (g)	0·09	0·124	0·15	> = 15
Price (NIS)	0·02	0·06	0·062	
Optimal solution (g)	113·2	78·7		

NIS, National Israeli Shekel.

Minimize the objective function

$$c_1 x_1 + c_2 x_2 + c_3 X_3 \ldots + c_n x_n,$$

where c_n is the cost of variable x_n.
Subject to:

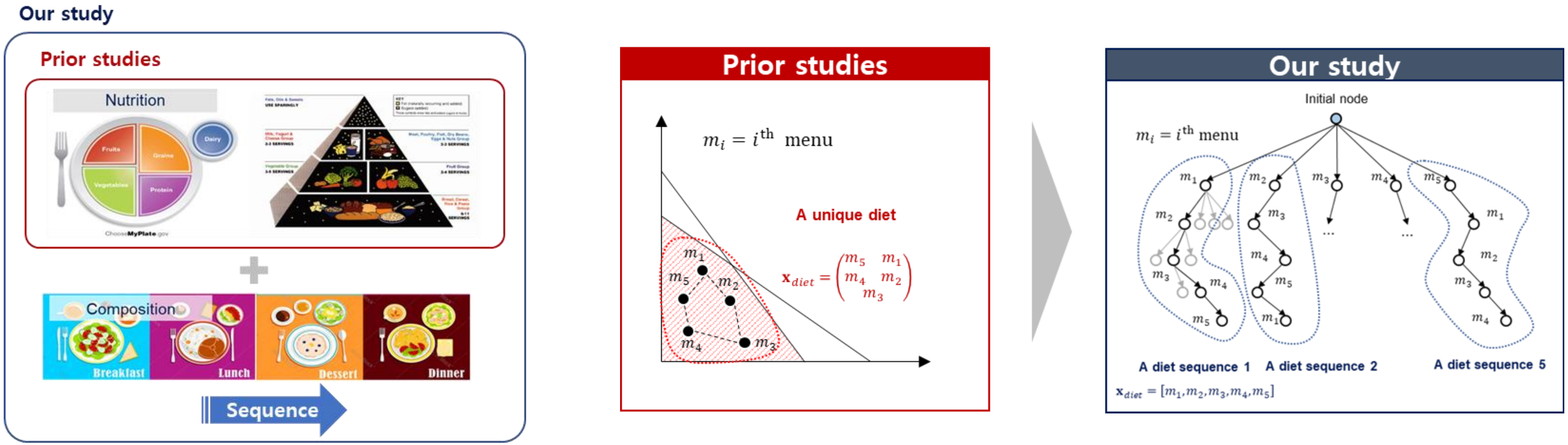
$$a_{11} x_1 + a_{12} x_2 + a_{13} X_3 \ldots + a_{1n} x_n > = b_1,$$
$$a_{21} x_2 + a_{22} x_2 + a_{23} X_3 \ldots + a_{2n} x_n > = b_2,$$
$$\cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot$$
$$a_{m1} x_2 + a_{m2} x_2 + a_{m3} X_3 \ldots + a_{mn} x_n > = b_m,$$

where a_{mn} is the value of constraint m in variable x_n.

- Designing a diet has been addressed in the field of economics and operation research in the name of 'diet planning".
- Researchers in those fields regarded diet design as a 'combinatorial optimization problem' and used a 'mathematical programming (MP)' approach.
- That is, the prior studies of diet planning focused on finding a set of menus, what they call a diet, that achieves an optimized level of nutrition.

Introduction

- limitations of prior studies



- Prior studies optimized objective function under 'nutrition constraint', and it always guaranteed to find an optimal unique set of menus.
- However, they could not consider 'composition', the way how menus should be arranged both within and across meals. For example, *"Cookies go better with milk than with grape juice"* (within) or *"People prefer to eat light in the morning and a full meal in the evening"* (across) is an implicit pattern and should be considered when planning a desirable diet, but we do not know how many such patterns exist. Thereby, the prior studies, which require explicitly set the patterns as constraints, cannot but missing 'composition'.
- As a result, the diet planned in prior studies looked unnatural and needed a rearrangement by humans to make it natural.
- Needless to say, the prior studies couldn't be extended to the advanced requirements (e.g., food culture and individual belief), just as the case of composition, because new constraints should be explicitly set whenever the requirements are added.

Motivation and contribution

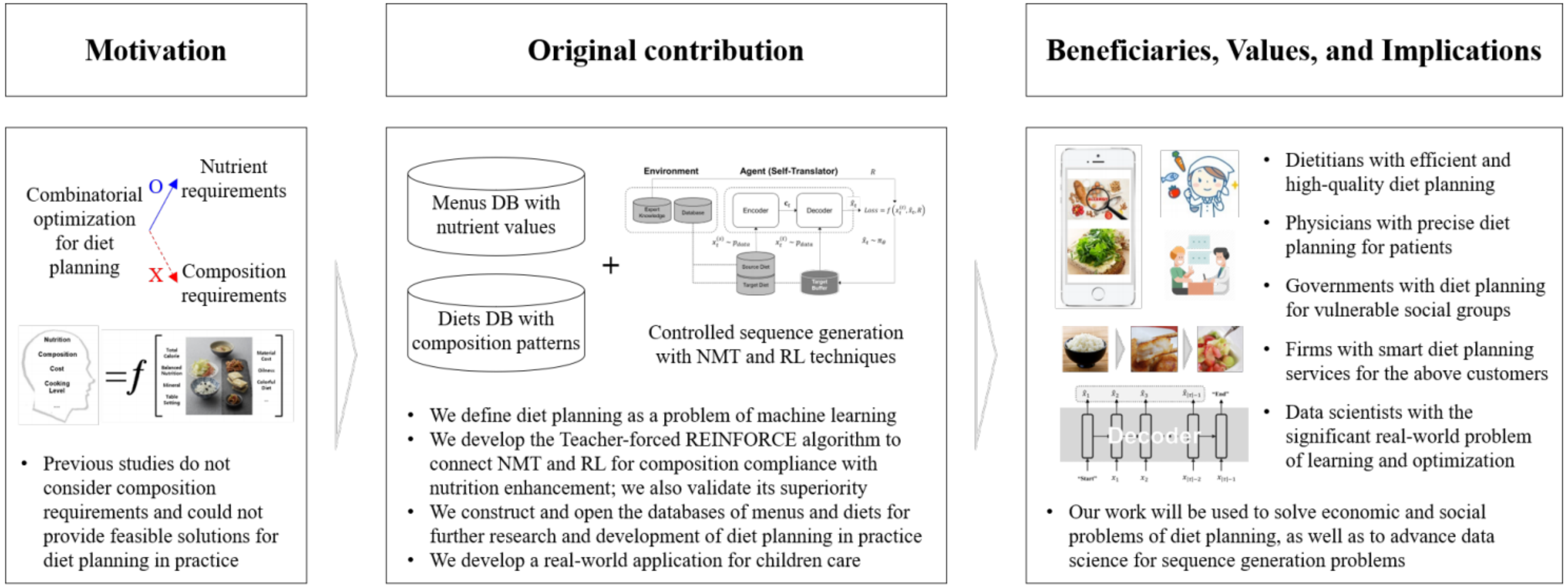
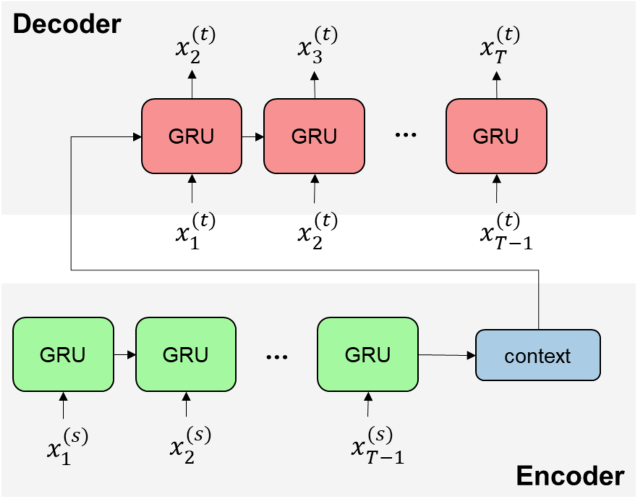


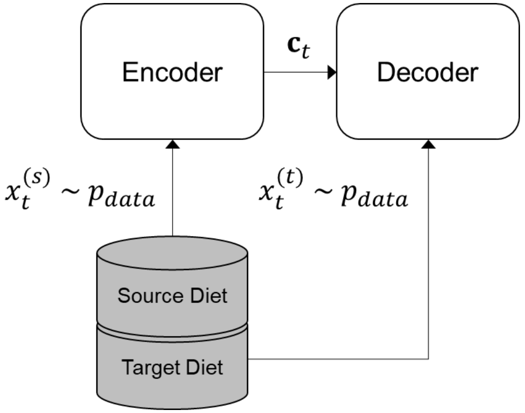
Figure 1: Overview and contribution of this work

- We're motivated by the fact that (1) diet planning is deeply rooted in our lives, and (2) prior studies failed to consider the diet composition due to the intrinsic limitation of MP approach.
- To consider 'nutrition' and 'composition' together, we took a machine learning-based approach using Neural Machine Translation (NMT) and Reinforcement Learning (RL). In detail, we developed a novel algorithm, called 'Teacher-forced REINFORCE (TFR)', which best connects NMT and RL in the context of diet planning.
- In this study, our algorithm is developed to plan a diet for the 3 to 5 years, thus the dietitians or pediatricians are expected to be the beneficiaries.

Background

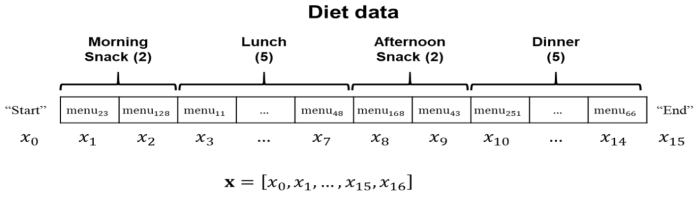
- NMT and Diet generator





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	...

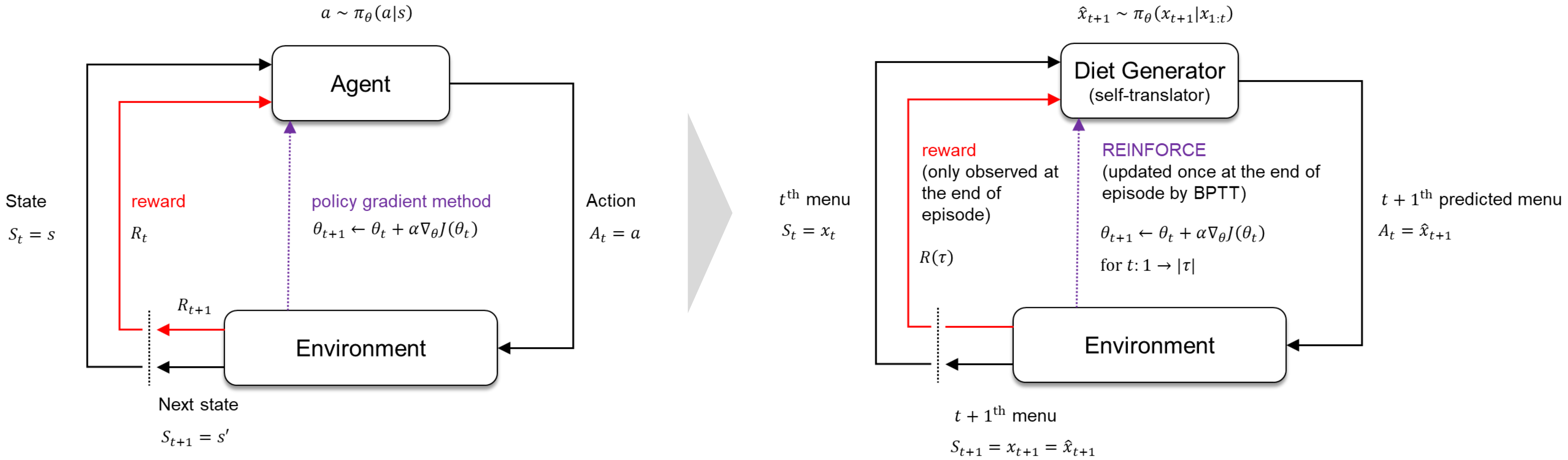


$$\nabla_\theta \mathcal{L}_\theta \left(\mathbf{x}^{(t)}, \mathbf{x}^{(s)}\right) = -\sum_{t=1}^{T} \nabla_\theta \log p_\theta \left(x_t^{(t)} | x_{0:t-1}^{(t)}\right) \qquad (6)$$

- A daily diet consists of meals (e.g., lunch and dinner), with each meal having a specific menu patterns, and we address the diet as a sequence.
- NMT addresses the translation task where a sequence in a source domain (e.g., English) is mapped into the corresponding sequence in a target domain (e.g., French), and the implicit pattern of a source sequence is encoded into a latent feature and the decoder decomposes it to generate a target sequence. Here, the latent feature embeds an information of 'composition'.
- Given a sequence $\mathbf{x} = [x_0, .., x_T]$ (i.e., a diet sequence in which the t^{th} element is a token from a set of all menu items $x_t \in \mathcal{M}$), we trained self-translator by feeding same sequence into both domains $x_{0:T}^{(s)} = x_{0:T}^{(t)}$, which embeds latent feature helping to generate realistic diet sequences.
- Our diet generator, i.e., the self-translator, is optimized by minimizing the reconstructive translation error like 'auto-encoder' (See equation (6))..

Background

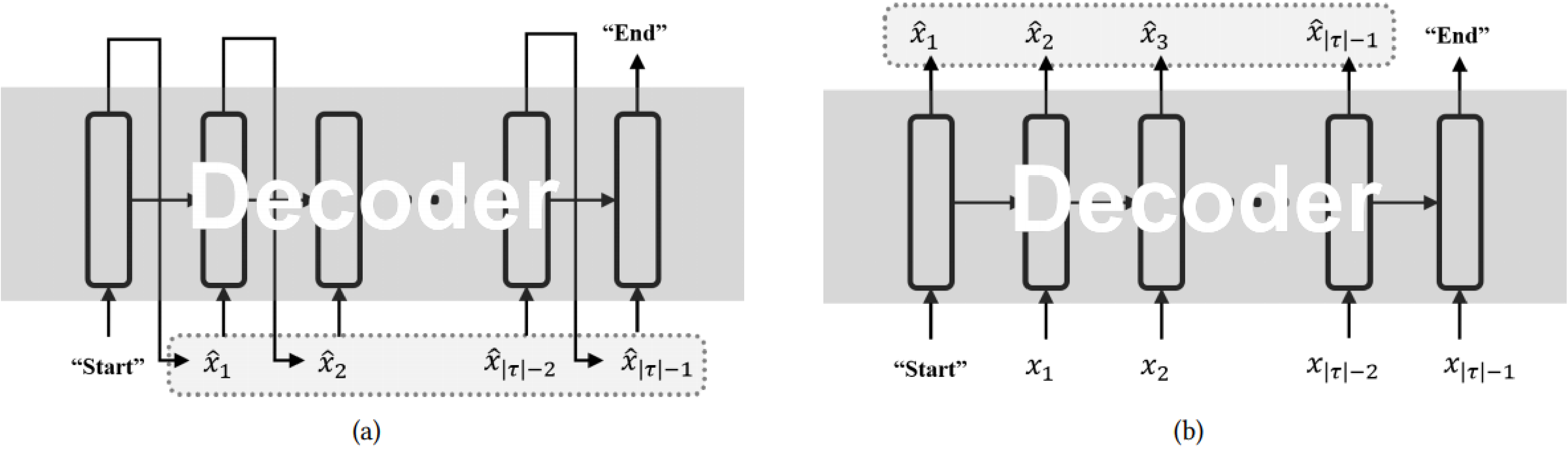
- RL and Controlled diet generation



- Despite the importance of composition, the essence of diet planning is still to achieve the required nutrition. Thus, we use RL to control NMT and give it a nutrition requirements (i.e., the required daily intake; RDI) as a reward.
- Accordingly, we assume that the self–translator is an agent who predicts each token consecutively, and a diet is produced through the iterative process of token sampling. τ is the trajectory of token samplings.
- The agent observes R at the end of sampling because the nutritional value is calculated based on the RDI at the level of diet, not the level of menu items.
- We control the generative process using the REINFORCE algorithm, having the policy improvement achieved by BackPropagation Through Time (BPTT).

Model

- Exposure-bias problem (i.e., collapse mode)

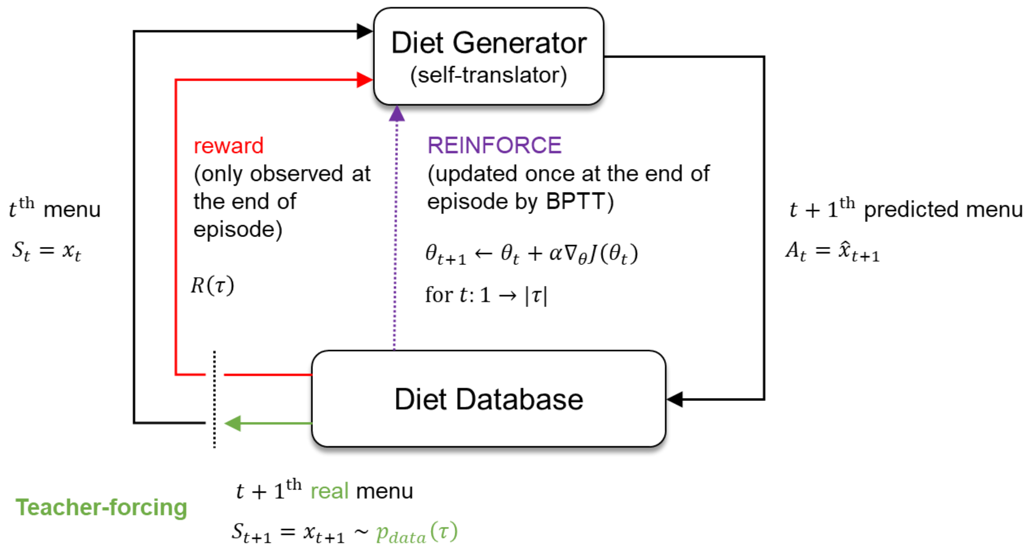


- However, the generative process is so fragile that whenever we try to control the process, unrealistic diet can be generated.
- For example, it would be a principle that *"spaghetti with tomato sauce"* would go better with *"a fruit salad"* than *"salmon sushi,"*. However, the agent can mistakenly sample out-of-principle menu items in maximizing rewards and generate an unrealistic diet causing the collapse of the generative process.
- The collapse 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 above figure (a)).
- Such recursive mechanism gives rise to high sampling bias and error accumulation, causing generative process vulnerable to the collapse mode.
- To overcome collapse, we propose using off-policy correction underlying the concept of teacher-forcing (see above figure (b)).

Model

- Teacher-forced correction and Policy space expansion

Off-policy correction $\hat{x}_{t+1} \sim \frac{\pi_\theta(\tau)}{p_{data}(\tau)} = \prod_{t=1}^{|\tau|} \frac{\pi_\theta(x_{t+1}|x_{1:t})}{p_{data}(x_{t+1}|x_{1:t})} \approx \pi_\theta(x_{t+1}|x_{1:t})$



β-score for policy space expansion

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

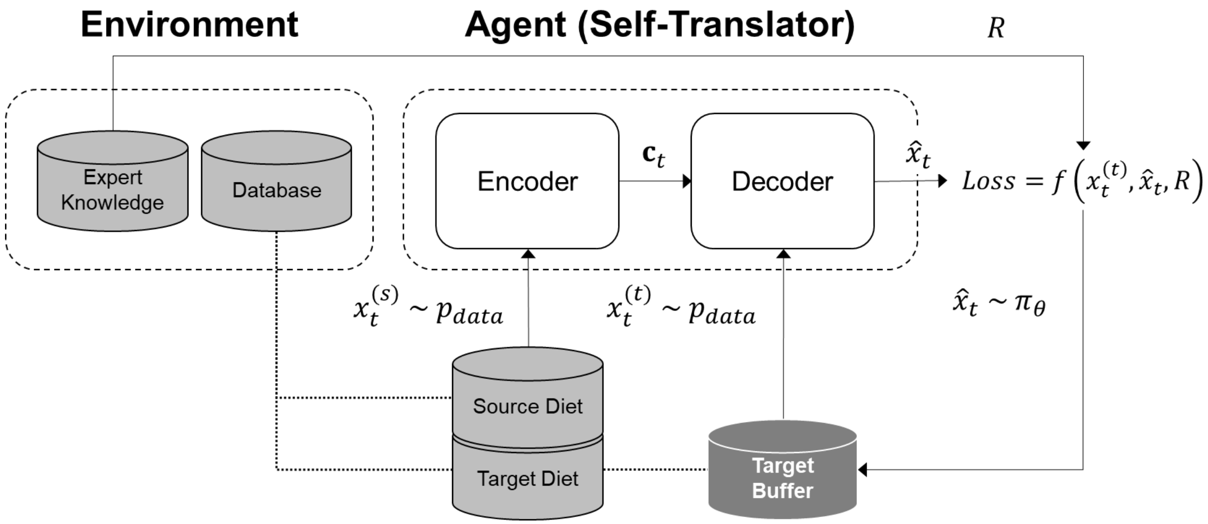
Gradient of Teacher-forced REINFORCE

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

- Consider the teacher-forced trajectory $\tau = (x_1, x_2, \dots x_T)$, which represents the sampling of the menu items from the target diet (i.e., real diet).
- We introduce the importance weight to implement the teacher-forcing technique within the REINFORCE algorithm.
- $\frac{\pi_\theta(\tau)}{p_{data}(\tau)}$ is the importance weight. By using first-order approximation and ignoring p_{data} (as it is a constant), we treated the importance weight as $\pi_\theta(x_{t+1}|x_{1:t})$. As a result, we can sample menu items both from $\tau = (x_1 . x_2, \dots, x_T)$ and $\hat{\tau} = (\hat{x}_1 . \hat{x}_2, \dots, \hat{x}_T)$.
- By sampling from τ, π_θ approximates around the fixed data distribution p_{data} (off-policy correction) and the collapse does not happen; In addition, we can deceive an agent as if it performs on-policy learning by sampling from $\hat{\tau}$. And, $\beta(\hat{\tau})$ indicates how replaceable $\hat{\tau}$ is with τ in average. With $\nabla_\theta J^{TF}(\theta)$ weighted by $\beta(\hat{\tau})$, the policy space is expanded having agent explore the trajectories of alternative (replaceable) diets.

Model

- Overall framework and Pseudo algorithm



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 θ, 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 β
    Compute policy gradient $\nabla_\theta J^{\mathbf{TF}}(\theta)$ by Equation (10)
    if $\beta == 1$ **then**
      if $n \leq N$ **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 θ 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

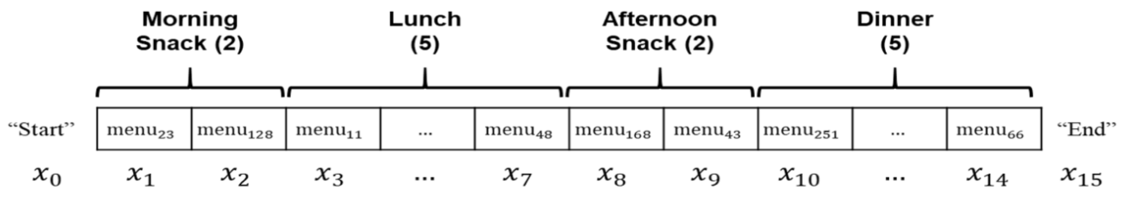
- As a second technique for expanding the policy space, we propose mixing the target diet with the generated one.
- The algorithm proceeds as follows: i) generate the synthetic diets, ii) store the most recent N synthetic diets (with highest β score) in target buffer $\mathcal{B}$, and iii) randomly select new target diets from $\mathcal{B}$ every M epochs.
- This augments the size of the target data, expands the policy space, and thereby the agent generates diets beyond the real diets.

Data

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



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

- 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.
- 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.
- The databases are publicly available and can be accessed at [Diet-Generation-As-Sequence/Data (new) at master · Leo-Lee92/Diet-Generation-As-Sequence (github.com)]

Experiment and Result

- Baselines
  - Cbc solver (MIP)
  - SCST (on- vs off-policy contrastive learning)
  - MIXER (curriculum learning)
- Evaluation metrics
  - Meal-hit rate (composition)
  - Dish-hit rate (composition)
  - RDI score (nutrition)
  - Overall score (reliability)
  - Turing score (reliability)

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	

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)

Experiment and Result

- Comparative study and Results

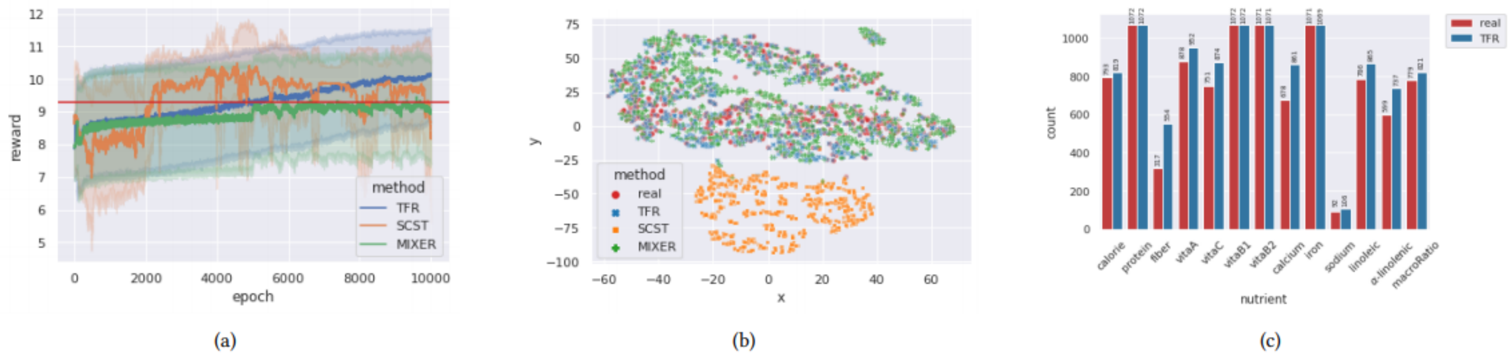


Figure 5: Comparison of the performance in the training and generation phases. (a) Rewards (nutrition level) of the predicted diet in the training phase, where the red line indicates the average nutrition level of real diets. (b) t-Distributed Stochastic Neighbor Embedding (t–SNE) nutrient map of the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

Table 1: Summary of the evaluation results

Method	(Composition) Meal-hit rate	(Composition) Dish-hit rate	(Nutrition) RDI score	(Reliability) Overall score	(Reliability) 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**

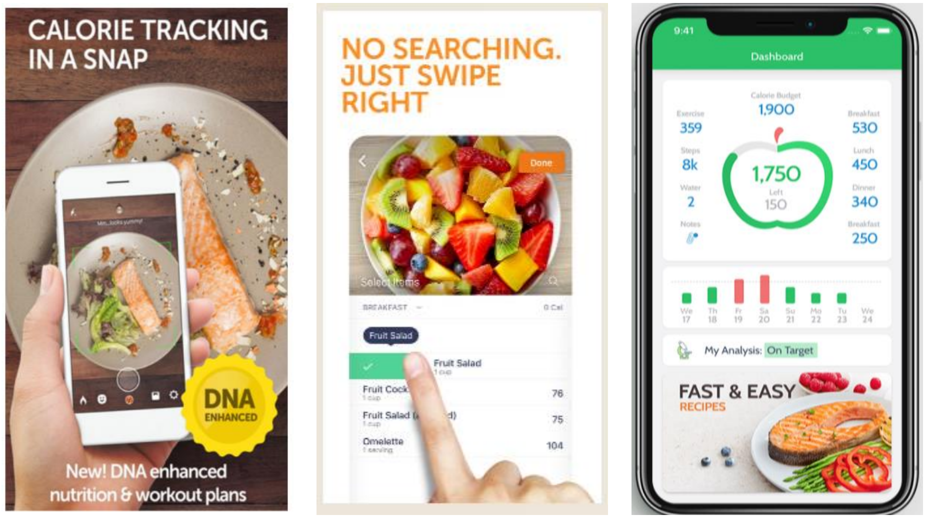
Experiment and Result

- Comparative study and Results

Table 5: Comparison between real diet and generated diets

no	**(Source Diet)** *Real*	**(Translated Diet)** *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

Conclusion and Future work



- 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.
- 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). This work (1) overcomes the limitations of the combinatorial optimization approach by leveraging machine learning model (e.g., NMT and RL techniques), (2) addresses the exposure-bias problem by combining teacher-forcing with REINFORCE algorithm, and (3) improves the quality of generation by adding two techniques that help expanding the policy space.
- As future work, we can consider customizing rewards, so-called reward shaping, to plan diets that achieve advanced requirements such as ‘food culture’ or ‘allergy’. In terms of practicality, the development of an online learning algorithm for diet planning can be future work as well.

THANK YOU

Changhun LEE (UNIST)