Modeling novel needs generation

Jianhui HUANG

August 18, 2025

Abstract

This project includes a framework for modeling novel needs generation.

1 Novel needs generation framework

1.1 Basic concepts

In this framework, we define needs X as a triplet as below:

Definition 1 *The needs variable X is modeled as a triplet $X = (P, T, S)$, where P denotes the product variable, T denotes the intention variable and S denotes the scene variable. The variables X, P, T and S have outcomes in the form of text.*

In this sense, the needs X can be expressed as "In the scene of S, someone uses a product P to satisfy his/her intention T.". For example, an outcome x_i of needs variable X is a text sequence consisting of the outcomes triplet (p_i, t_i, s_i) of product P, intention T and scene S variables, where p_i, t_i, s_i in the triplet are also text sequence.

Before we consider the novelty of a needs X, we notice that there is a concept called *surprisal* in information theory. The *surprisal* is also called *information amount* or *self-information*. It is defined as:

$$\log(\frac{1}{\mathrm{p}(E)}) \tag{1}$$

where E is an event. From the concept of *surprisal*, we learn that if the probability of an event E is low, the *surprisal* $\log(\frac{1}{\mathrm{p}(E)})$ becomes high. This inspires us to define the novelty of a needs X in a similar way. Particularly, if a needs X rarely happens, or equivalently $\mathrm{p}(X) = \mathrm{p}(P, T, S)$ is low, the novelty of X might be high. However, simply considering the probability $\mathrm{p}(X)$ of needs X is not always appropriate since some rare needs may not be novel. For example, fireworks are usually rare and only appear occasionally in during celebrations, but fireworks are not the novel product. Therefore, we cannot simply use $\frac{1}{\mathrm{p}(X)} = \frac{1}{\mathrm{p}(P,T,S)}$ as the definition of novelty since a low $\mathrm{p}(P, T, S)$ doesn't necessarily imply novelty. Alternatively, we can consider the probability that eliminates the factor of the scene S. In this case, we believe that the conditional probability $\mathrm{p}(P, T|S)$ of product P and intention T given scene S can define the novelty of needs. Specifically, we define the novelty of needs as below:

Definition 2 *The novelty of needs can be formulated as below:*

$$\frac{1}{\mathrm{p}(P, T|S)} \tag{2}$$

This implies that when the $\mathrm{p}(P, T|S)$ of needs $X = (P, T, S)$ is low, the corresponding novelty of needs X is high. Similarly, when $\mathrm{p}(P, T|S)$ is high, the corresponding novelty is low.

This formula solve the challenge encountered by simply using $\frac{1}{\mathrm{p}(X)} = \frac{1}{\mathrm{p}(P,T,S)}$ since it eliminates the factor of scene S. We move back to the previous case of fireworks and see if $\frac{1}{\mathrm{p}(P,T|S)}$ works well. In the scene of celebrations, setting off fireworks to bring joy to people is quite usual, that means $\mathrm{p}(P, T|S)$ is high. This formula implies that setting off fireworks in the scene of celebrations is not a novel needs. Therefore, we can adopt $\frac{1}{\mathrm{p}(P,T|S)}$ to be the novelty of a needs X since a low $\mathrm{p}(P, T|S)$ can distinguish novel needs from usual needs.

In order to understand the feasibility of the novelty definition, we consider two successful cases of novel needs. The first novel needs is that in the scene of short trip, a pet owner can use pet backpack to keep their pet in the backpack and prevent it from being lost. Before this needs receives enough attention and gets popularized, we notice that the conditional probability of $\mathrm{p}(p, t|s)$ is low where p is a backpack, t is to keep a pet and s is the scene of short trip. This is because nobody thinks it is viable for a pet owner to keep his/her pet in the backpack during short trip. So the conditional probability of pet backpack given the scene of short trip is low. Based on the low $\mathrm{p}(p, t|s)$, our novelty definition suggests this needs $x = (p, t, s)$ is a needs of high novelty since $\frac{1}{\mathrm{p}(p,t|s)}$ is high. The second case is about the product of smart phone with camera of high quality. Before the era of smart phone, the camera on the mobile phone is usually of low quality and people can hardly use mobile phone to take pictures during trip. That is the conditional probability $\mathrm{p}(p, t|s)$ is low where p is a mobile phone, t is to take pictures and s is the scene of trip. Similarly, our novelty definition suggests the needs of using mobile phone to take picture during trip is a novel needs. This can inspire companies at that time to develop mobile phones with camera of high quality and then successfully open the new era of smart phone.

1.2 Automatic novel needs generation

In novel needs generation, our goal is to automate the novel needs generation and generate novel needs in batches. To this end, an intuition is to derive the probability distribution of novel needs and then automatically sample novel needs data from the probability distribution. In this case, we can leverage statistical inference to infer the probability distribution of the novel needs. During statistical inference, we should collect data samples, perform data analysis on data samples to obtain the parameter estimate of the probability distribution, and then infer the ground-truth probability distribution using the parameter estimate. In the context of novel needs generation, we should collect novel needs data sample, parameterize the probability distribution with deep neural network, and enable the parameterized neural network to approximate the ground-truth probability distribution by maximizing the log-likelihood of the novel needs data samples.

However, there remains a challenge on collecting novel needs data samples. Traditional novel needs data collection heavily relies on laborious survey works and expensive human expert annotation, which leads to the scarcity of large-scale novel needs datasets. The lack of large-scale novel needs datasets greatly restricts the exploration on novel needs. Particularly, the rareness of novel needs data implies that the limited data samples cannot reflect and reveal the characteristic of the ground-truth probability distribution. In this case, the parameterized neural network cannot effectively learn to approximate the ground-truth probability distribution due to the sparse learning signal from limited data samples. To this end, we propose an unsupervised data synthesis approach to obtain a large-scale novel needs dataset. With the large-scale novel needs dataset, we are able to learn a parameter estimate of the probability distribution over novel needs data and automate novel needs generation through sampling strategies.

1.2.1 Unsupervised novel needs data synthesis approach

In this section, we propose an unsupervised data synthesis approach to obtain a large-scale novel needs dataset. We firstly demonstrate the derivation of the unsupervised data synthesis approach and then present the unsupervised data synthesis algorithm for novel needs.

Before introducing the novel needs data synthesis approach, we define the novel needs data as below:

Definition 3 *The novel needs X^{nvl} is a needs variable with high novelty, implying that the $\mathrm{p}(P, T|S)$ of X^{nvl} is low.*

With the definition of novel needs X^{nvl}, we can now think about how to collect the novel needs sample dataset $\{x_i^{nvl}\} = \{(p_i, t_i, s_i)\}$. Our goal is to collect the needs of high novelty as our sample dataset. This implies that the needs x_i^{nvl} in the dataset $\{x_i^{nvl}\}$ should have a high novelty $\frac{1}{\mathrm{p}(p_i,t_i|s_i)}$.

To guarantee the high novelty of needs, we should make the conditional probability $\mathrm{p}(p, t|s)$ of the collected needs x^{nvl} as low as possible. We analyze the value of $\mathrm{p}(p, t|s)$ as below:

$$\mathrm{p}(p, t|s) = \frac{\mathrm{p}(p, t, s)}{\mathrm{p}(s)} > \mathrm{p}(p, t, s) \tag{3}$$

We learn that $\mathrm{p}(p,t,s)$ is a lower bound for $\mathrm{p}(p,t|s)$, implying that if the $\mathrm{p}(p,t,s)$ is high, then the $\mathrm{p}(p,t|s)$ will be surely high. Therefore, to ensure the $\mathrm{p}(p,t|s)$ is low, the lower bound $\mathrm{p}(p,t,s)$ should also be low. Now, we want the collected sample has a low $\mathrm{p}(p,t,s)$. To this end, we decompose $\mathrm{p}(p,t,s)$ to the product of marginal probability and conditional probabilities as below:

$$\begin{aligned} \mathrm{p}(p,t,s) &= \mathrm{p}(p) \times \mathrm{p}(t,s|p) \\ &= \mathrm{p}(p) \times \mathrm{p}(s|p) \times \mathrm{p}(t|s,p) \end{aligned} \tag{4}$$

This decomposition reveals that, starting from sampling a product p, a low $\mathrm{p}(p,t,s)$ can be obtained by sampling the scene s with low $\mathrm{p}(s|p)$ or the intention t with low $\mathrm{p}(t|s,p)$ given a product p. Since there is sufficient product information online, we can construct a product set consisting of many products and later sample a product from a predefined product set. For the conditional sampling of scene and intention, we can leverage the large language model(LLM) as an agent model to conduct sampling. This is because LLMs own the prior knowledge and understanding on the relations between product and scene/intention. To sample a scene or intention, we design a group of instructions that require LLM agent to output intended text content according to the specified context. The scene or intention sample can be obtained from the LLM text output.

However, the low $\mathrm{p}(p,t,s)$ doesn't necessarily imply low $\mathrm{p}(p,t|s)$. To ensure $\mathrm{p}(p,t|s)$ is low, the marginal probability $\mathrm{p}(s)$ should not be too low. That is the scene s should not be a generally uncommon scene. We can guarantee this by removing the needs sample with a generally uncommon scene from the collected sample dataset. In experiment, to encourage diverse combinations of scenes and intentions, we disentangle the interaction between scene and intention through the assumption where scene and intention variables are conditionally independent given product variable. The conditional independence assumption implies that $\mathrm{p}(t|s,p) = \mathrm{p}(t|p)$. The intuition behind this assumption is that the generated intention is greatly restricted and limited by the given scene in $\mathrm{p}(t|s,p)$, resulting in monotonous intention. In this case, the interaction between scene and intention should be eliminated to discover novel combination of scene and intention. Therefore, we adopt $\mathrm{p}(t|p)$, instead of $\mathrm{p}(t|s,p)$, to sample intention t in the following algorithm and experiments.

The above decomposition inspires us to propose the unsupervised data synthesis algorithm for novel needs. The algorithm is shown as below:

Algorithm 1 shows us how to synthesize novel needs data samples with low $\mathrm{p}(p,t|s)$. Finally, we collect a novel needs dataset $\{x_i^{nvl} = (p_i, t_i, s_i)\}$ where the needs novelty is generally high among needs data samples. With such a dataset, we can enable a parameterized neural network to learn to approximate the ground-truth probability distribution of novel needs.

Algorithm 1 Unsupervised data synthesis algorithm for novel needs with low p$(p, t|s)$

Require: N: the maximum number of synthesized samples. α_1, α_2: the probability for sampling uncommon scenes and intentions respectively. Product set $\mathcal{B}$: the set of products. Large language models(LLM) π: the agent model that takes instructions as input and outputs expected text content. Instruction $\mathcal{I}$: the instructions for LLM to perform intended tasks.

Ensure: $NovelData$: the synthesized novel needs dataset.

```
SynData ← []
while N > 0 do
    Flag ← 1
    Sample a product p from product set B.
    while Flag == 1 do
        Sample an uncommon scene s via LLM π(·|p, I) with a probability of α1 given product p.
        Sample an uncommon intention t via LLM π(·|p, I) with a probability of α2 given product p.
        if s is uncommon ‖ t is uncommon then
            Flag ← 0
            x_syn ← (p, t, s)
            SynData.add(x_syn)
        end if
    end while
    N ← N − 1
end while
NovelData ← []
for x_syn in SynData do
    s ← x_syn.s
    Predict if s is uncommon via LLM π(·|s, I).
    if s is not uncommon then NovelData.add(x_syn)
    end if
end for
return NovelData
```