A-014

Questioning Strategy for Helping People with Aphasia to Retrieve Words

Yusuke Nakahara† Yasuo Horiuchi† Shingo Kuroiwa†

1. Introduction

When an agent tries to establish communication with an individual with aphasia, the agent has to use several communication strategies [1] that are used by speech-language-hearing therapists. One of the effective strategies is to ask questions that can be answered with "yes" or "no." Hence, we proposed an application that helps an individual with aphasia to retrieve words, especially the names of dishes [2]. The application asks yes/no questions to estimate the dish name the individual wants and then shows the estimated dish names to the individual.

The problem of estimating dish names by asking yes/no questions is related to Rényi-Ulam games [3], which have many variants. Some of the variants can be stated in terms of a game played by two players: Questioner and Responder. The Responder first chooses an object from a set. The chosen object is unknown to the Questioner while the set is known to the Questioner. Then, the Questioner asks questions about the chosen object, and the Responder gives answers. The Responder lies sometimes, and the probability of randomly lying is known to the Questioner. The objective of the Questioner is to identify the chosen object. However, if the Responder randomly lies, the Questioner cannot identify the object since all the answers may be lies in the worst case. Hence, the Responder instead attempts to maximize the probability that the chosen object is one of a few objects [4], for example.

If an application tries to estimate the dish name that is wanted by the individual with aphasia, the application and the individual can be associated with the Questioner and the Responder, respectively. The individual may unintentionally give incorrect answers. For example, if the individual does not recognize a word in the question, he/she may guess its meaning from the other words, and sometimes the guess is wrong. This association allows the application to use the methods based on Rényi-Ulam games to select the questions.

In our previous work [2], the application asks up to 30 questions and then shows all the candidate dish names in groups of 5. The application ends asking questions when the sum of the posterior probabilities of the 5 candidates to first be shown exceeded 0.5 or when 30 questions have been asked. The parameters, i.e. the number of candidates to show (5), maximum number of questions (30), and threshold of the sum of the posterior probabilities (0.5) are determined beforehand. The parameters are related to the number of questions the individual needs to answer and the number of candidates the individual needs to see. Since these two numbers are in a trade-off relationship, the parameters had to be empirically determined so that they balance the two numbers.

Hence, we propose a method to determine when to end asking questions. The proposed method attempts to minimize the sum of the number of questions to answer and the number of candidates to see. The simulation results show that the proposed method successfully determines when to end asking questions.

Furthermore, a method to select questions, which was proposed for a variant of the Rényi-Ulam games, is used in our application. The simulation results show that the method achieves results that are comparable to those of the question-selection method that were used in our previous work with less complexity.

† Graduate School of Advanced Integration Science, Chiba University

2. Prior Work

While the questions were selected in terms of the *expected value of entropy* in our previous work, methods based on Rényi-Ulam games to select questions are available for our use. In this paper, a question-selection method that uses a *Bayesian strategy* is applied to our problem and is compared with the method that was used in our previous work.

2.1 Our Previous Work

In our previous work [2], the application first asks up to 30 questions and then shows the candidates in groups of 5 as described above. The method to select questions is described later.

After up to 30 questions have been asked, the 5 candidates to first be shown, which have the largest posterior probabilities, are selected. The posterior probability of a candidate s depends on the asked questions and the answers:

$$p(s|Q_y,Q_n)=\frac{\left(\prod_{q\in Q_Y}\psi_q(s)\right)\left(\prod_{q\in Q_N}\bar{\psi}_q(s)\right)}{\sum_s\left(\prod_{q\in Q_Y}\psi_q(s)\right)\left(\prod_{q\in Q_N}\bar{\psi}_q(s)\right)}$$

where Q_Y and Q_N are the sets of asked questions whose answers were "yes" and "no," respectively, $\psi_q(s)$ is the prior probability of answering "yes" for question q when the intended candidate is s, and $\bar{\psi}_q(s)=1-\psi_q(s)$.

Each question is selected so that it minimizes the *expected value of entropy* in terms of the candidates' posterior probabilities after the answer to the question is observed:

$$\hat{q}=\underset{q}{\operatorname{argmin}}\left(\sum_s \pi(s)\,\psi_q(s)\right)H_Y+\left(\sum_s \pi(s)\,\bar{\psi}_q(s)\right)H_N$$

where

$$H_Y=-\sum_s\frac{\pi(s)\,\psi_q(s)}{\sum_s\pi(s)\,\psi_q(s)}\log\left(\frac{\pi(s)\,\psi_q(s)}{\sum_s\pi(s)\,\psi_q(s)}\right),$$

$$H_N=-\sum_s\frac{\pi(s)\,\bar{\psi}_q(s)}{\sum_s\pi(s)\,\bar{\psi}_q(s)}\log\left(\frac{\pi(s)\,\bar{\psi}_q(s)}{\sum_s\pi(s)\,\bar{\psi}_q(s)}\right),$$

$$\pi(s)=p\left(s|\hat{Q}_Y,\hat{Q}_N\right)$$

and $\hat{Q}_Y$ and $\hat{Q}_N$ denote the sets of previously asked questions whose answers were "yes" and "no," respectively.

2.2 Select Question with Bayesian Strategy

A method to select questions using a *Bayesian strategy* has been proposed for a variant of Rényi-Ulam games [4].

The next question is selected so that the probability that the answer to the next question is “yes” is closest to 0.5:

$$\hat{q}_B = \underset{q}{\operatorname{argmin}} \left(\frac{1}{2} - \sum_s \pi(s)\, \psi_q(s) \right)^2 .$$

Although the selected question is not always optimal, the questions can be selected with less complexity.

3. Proposed Method

In our previous method, the maximum number of questions and the threshold of the sum of the posterior probabilities have to be empirically determined. In this paper, we propose a method to determine when to end asking questions by attempting to minimize the sum of the number of questions to answer and the number of candidates to see.

3.1 When to End Asking Questions

To determine when to end asking questions, we propose comparing the sums of the expected number of questions to answer and the expected number of candidates to see a) when no more questions are asked and b) when one more question is asked. If the former is greater than the latter, the question should be asked.

If no more questions will be asked, the expected number of candidates to see, i.e. the expected order of the intended candidate is obtained as:

$$r(\boldsymbol{\pi}) = \boldsymbol{m}^T P\, \boldsymbol{\pi}$$

where $\boldsymbol{m} = [1, 2, \dots]^T$, P is the permutation matrix that reorders the elements of $\boldsymbol{\pi}$ in descending order, and $\boldsymbol{\pi} = [\pi(s_1), \pi(s_2), \dots]^T$ is a vector whose elements are the posterior probabilities of the corresponding candidates, $s_1, s_2, \dots$. Therefore, the weighted sum is:

$$f(\nu, \boldsymbol{\pi}) = \nu + \boldsymbol{m}^T P\, \boldsymbol{\pi}$$

where ν is the number of previously asked questions.

When only one more question q will be asked, the sum is:

$$\begin{aligned} f_q(\nu, \boldsymbol{\pi}) &= \nu + 1 + (\boldsymbol{\pi}^T \boldsymbol{\psi}_q) \boldsymbol{m}^T P_Y \left(\frac{1}{\boldsymbol{\pi}^T \boldsymbol{\psi}_q} \Psi_q \boldsymbol{\pi} \right) \\ &\quad + (\boldsymbol{\pi}^T \bar{\boldsymbol{\psi}}_q) \boldsymbol{m}^T P_N \left(\frac{1}{\boldsymbol{\pi}^T \bar{\boldsymbol{\psi}}_q} \bar{\Psi}_q \boldsymbol{\pi} \right) \\ &= \nu + 1 + \boldsymbol{m}^T P_Y \Psi_q \boldsymbol{\pi} + \boldsymbol{m}^T P_N \bar{\Psi}_q \boldsymbol{\pi} \end{aligned}$$

where $\boldsymbol{\psi}_q = [\psi_q(s_1), \psi_q(s_2), \dots]^T$ is a vector whose elements denote the prior probabilities of answering “yes” to the question q in terms of the corresponding candidates, $s_1, s_2, \dots$, $\Psi_q = \operatorname{diag}(\boldsymbol{\psi}_q)$, and $P_Y$and $P_N$are the permutation matrices that reorder the elements of $(\Psi_q \boldsymbol{\pi})$ and $(\bar{\Psi}_q \boldsymbol{\pi})$, respectively.

If $f(\nu, \boldsymbol{\pi})$ and $f_q(\nu, \boldsymbol{\pi})$ satisfy the condition $f(\nu, \boldsymbol{\pi}) > f_q(\nu, \boldsymbol{\pi})$, the question q should be asked, as described at the start of this section. Determining this condition is equivalent to determining the sign of their difference:

$$\begin{aligned} \Delta_q(\boldsymbol{\pi}) &= f_q(\nu, \boldsymbol{\pi}) - f(\nu, \boldsymbol{\pi}) \\ &= w + \boldsymbol{m}^T \left(P_Y \Psi_q + P_N \bar{\Psi}_q - P \right) \boldsymbol{\pi}. \end{aligned}$$

Here, $\Delta_q(\boldsymbol{\pi})$ is free from ν. Furthermore, P_Y and P_N are selected so that $f_q(\nu, \boldsymbol{\pi})$ is minimized. Hence, replacing P_Y, P_N with P provides an upper bound of $\Delta_q(\boldsymbol{\pi})$: $\Delta_q(\boldsymbol{\pi}) \le 1$.

Conclusively, if $\Delta_q(\boldsymbol{\pi}) < 0$ for the question q that is selected with any of the previous methods, the question should be asked.

3.2 Preselecting Questions with Condition

The previous methods select the next question regardless of whether it satisfies the condition $\Delta_q(\boldsymbol{\pi}) < 0$. Therefore, even if the selected question does not satisfy the condition, another question may satisfy it.

To use such questions, we also propose selecting the next question to ask only from the questions that satisfy the condition.

3.3 Asking Questions after Showing Candidates

In our previous work, the application shows all the candidates in groups of 5 after asking up to 30 questions. Another approach is to repeat asking questions and showing 5 candidates alternately until the intended candidate is shown. Our previous work is a special case of this approach: asking zero questions after any candidate is shown. In the following sections, we refer to this generalized approach as *interleaving*.

We propose using the condition $\Delta_q(\boldsymbol{\pi}) < 0$ for this approach also though it is not straightforward. A question is asked if the selected question satisfies the condition $\Delta_q(\boldsymbol{\pi}) < 0$; otherwise, a candidate is shown. The number of candidates to show no longer needs to be predetermined with this approach; if more than one candidate is shown successively, these candidates are shown at once.

Our next concern is how showing a candidate affects the condition $\Delta_q(\boldsymbol{\pi}) < 0$. If a candidate is shown and it is not the intended candidate, the posterior probabilities of the candidates will be modified. The modified posterior probabilities are:

$$\tilde{\pi}(s) = \begin{cases} 0 & if\ \ s = s^* \\ \dfrac{\pi(s)}{1 - \pi(s^*)} & otherwise \end{cases}$$

where s^* is the candidate to show, which has the maximum posterior probability. This is equivalent to:

$$\tilde{\boldsymbol{\pi}} = \frac{1}{1 - \pi^*} \operatorname{diag}(P^{-1} \boldsymbol{e})\, \boldsymbol{\pi}$$

where $\tilde{\boldsymbol{\pi}}$ is the updated posterior probabilities, and $\boldsymbol{e} = [0, 1, 1, \dots]^T$ is a vector whose elements are 1 except for the first element.

If a question q does not change the order of the candidate s^*,

$$\Delta_q(\tilde{\boldsymbol{\pi}}) = w + \frac{\boldsymbol{m}^T \tilde{P} \left(P_Y \Psi_q + P_N \bar{\Psi}_q - P \right) \operatorname{diag}(P^{-1} \boldsymbol{e})\, \boldsymbol{\pi}}{1 - \pi(s^*)}$$

where $\tilde{P}$ is the permutation matrix that reorders the vector elements so that the first element of the vector becomes the last one. Since P_Y, P_N, and P reorder the vector elements so that the vector element associated with s^* becomes the first element,

$$\begin{aligned} \Delta_q(\tilde{\boldsymbol{\pi}}) &= 1 + \frac{(\boldsymbol{m} - \boldsymbol{1})^T \left(P_Y \Psi_q + P_N \bar{\Psi}_q - P \right) \boldsymbol{\pi}}{1 - \pi(s^*)} \\ &= 1 + \frac{\Delta_q(\boldsymbol{\pi}) - 1 - \boldsymbol{1}^T \left(P_Y \Psi_q + P_N \bar{\Psi}_q - P \right) \boldsymbol{\pi}}{1 - \pi(s^*)} \\ &= 1 + \frac{\Delta_q(\boldsymbol{\pi}) - 1 - 0}{1 - \pi(s^*)} \\ &= \Delta_q(\boldsymbol{\pi}) + \frac{\pi(s^*) \left(\Delta_q(\boldsymbol{\pi}) - 1 \right)}{1 - \pi(s^*)} \\ &\le \Delta_q(\boldsymbol{\pi}). \end{aligned}$$

The value of $\Delta_q(\cdot)$ is decreased by showing a candidate unless the question changes the order of the candidate. Hence, even if a question does not initially satisfy the condition, it may satisfy the condition after some candidates are shown.

4. Experiments

4.1 Dataset

The candidate dish names have been obtained from Japanese Wikipedia. The pages that describe some dishes were extracted by searching for all the pages that belong to "category: 料理" (cooking) or its subcategories. Pages that belong to some of the manually selected subcategories, e.g. "料理学校" (cooking school), were excluded. The titles of the pages were used as dish names. The number of obtained names was 776.

Questions were selected from ones that are in the form: "is *name* one of its ingredients?", where *name* is the name of some ingredient, e.g. pork. The ingredient names were those that are listed in the cooking ontology provided by Nanba et al. [5]. The names of seasonings were excluded from the ingredients. The number of obtained ingredient names was 239. Once questions have been asked, they cannot be asked again.

Although it is ideal to directly measure the prior probabilities of answering "yes" for each pair of dish and question, it will be impractical to measure all of the 776×239 pairs. Therefore, we measured the frequency of appearance in the Rakuten recipe database [6] instead:

$$\phi(q,s) = \frac{|R(q,i(s))|}{|R(s)|}$$

where $i(s)$ denotes the ingredient that is asked about with the question s, $R(s)$ is the set of recipe entries that describe a recipe of s, and $R(s,i(s))$ is the set of recipe entries that describe a recipe of s and list the ingredient $i(s)$.

If the Responder is affected by aphasia, sometimes he/she does not recognize the ingredient name in a question. We simply assumed that the Responder answers "yes" once in twice in those cases. The prior probability of answering "yes" is therefore:

$$\psi_q(s) = (1-\epsilon)\,\phi(q,s) + 0.5\,\epsilon$$

where ϵ is the probability of not recognizing the ingredient name. The simulations have been performed with $\epsilon = 0, 0.1, 0.2, 0.3$.

4.2 Compared Methods

Table 1 shows the methods that are compared in the following sections.

The first column shows how the number of questions was determined. "Baseline" denotes the method in our previous work, i.e. ends asking questions when the sum of the posterior probabilities of the 5 candidates to first be shown exceeds the threshold or when the predefined number of questions have been asked. "Fixed" denotes asking a fixed number of questions. "Proposed" denotes the proposed method, i.e. ends asking questions when the next question does not satisfy the condition $\Delta_q(\boldsymbol{\pi}) < 0$. The number of questions when there were a fixed number of questions and the threshold of the sum of the posterior probabilities when using the "Baseline" method were determined so that the sums of the mean numbers of questions and the mean orders of the intended candidate are minimized. For the "Baseline" method, the number of candidates to show at once was 5, and the maximum numbers of questions were the same values as the numbers of questions when there were a fixed number of questions.

The second column shows the ways to select questions. "Entropy" denotes that the questions are selected in terms of the *expected value of entropy*, and "Bayesian" denotes that the questions are selected in terms of a *Bayesian strategy*.

The third column shows whether the questions were selected only from the questions that satisfy the condition.

The last column shows whether the *interleaving* approach was used.

The simulations were performed 77,600 times by each method; each candidate dish name was wanted 100 times for each method.

Table 1. Compared methods.

	Number of Questions	Select Questions with	Preselect Questions	Use *Inter-leaving*
1	Baseline	Entropy	No	No
2	Fixed	Entropy	No	No
3	Proposed	Entropy	No	No
4	Proposed	Entropy	Yes	No
5	Baseline	Bayesian	No	No
6	Fixed	Bayesian	No	No
7	Proposed	Bayesian	Yes	No
8	Proposed	Bayesian	No	No
9	Proposed	Bayesian	No	Yes
10	Proposed	Bayesian	Yes	Yes

Table 2. Sums of mean numbers of questions and mean orders of intended candidates with *expected value of entropy.*

Method	ϵ			
	0	0.1	0.2	0.3
1	26.238	48.012	71.851	98.495
2	28.590	49.864	72.030	97.869
3	22.994	47.841	71.265	99.429
4	21.990	43.639	66.402	93.209

Table 3. Mean numbers of questions with *expected value of entropy.*

Method	ϵ			
	0	0.1	0.2	0.3
1	13.632	20.698	24.985	30.518
2	17	23	26	31
3	13.170	15.880	18.469	20.889
4	14.815	20.912	26.082	30.169

4.3 When to End Asking Questions

The methods to determine when to end asking questions are compared in Table 2 and 3. Table 2 shows the sum of the mean number of asked questions and the mean order of the intended candidate of each method. Table 3 shows only the mean number of asked questions.

The proposed method with preselecting questions (4) outperformed the others.

The results of the proposed method without preselecting questions (3) are worse than those of baseline (1) especially when $\varepsilon = 0.3$. The proposed method without preselecting questions (3) appears to end asking questions too early, and some of the questions that satisfy the condition were not asked. The baseline (1) will be able to use such questions by increasing the maximum number of questions.

4.4 Using Bayesian Strategy

Table 4 and 5 show the results when the questions are selected using a *Bayesian strategy*.

Although the results with a *Bayesian strategy* are worse than those with the *expected value of entropy* shown in Table 2 and 3, the differences are slight. Hence, selecting questions with a *Bayesian strategy* is possible for reducing the complexity.

4.5 Asking Questions after Showing Candidates

Table 6 and 7 show the results when using the *interleaving* approach.

These results, without preselecting questions (9), achieved the best performance. These results motivate us to ask questions even after any candidate has been shown.

The comparison between (9) and (10) shows that preselecting questions has a negative effect if the *interleaving* approach is used. In that case, there is no reason to preselect questions to use all the questions that satisfy the condition since they can be asked even after candidates are shown.

5. Conclusion

We proposed a method to determine when to end asking questions and eliminated the need to predetermine the number of questions. A method to select questions that uses a *Bayesian strategy* was possible for reducing the complexity. Furthermore, the performance was further improved by asking questions even after any candidate is shown.

For practical use, more proper ways to calculate the probabilities, especially to obtain the prior probability $\psi_q(s)$, are needed. In this paper, the frequency of appearance in recipes was measured instead of the probability of answering “yes.” They should be different, for example, if some ingredient is used only for flavor, the answer may be “no.” To reduce the difference, a method to directly measure the prior probability or correct the difference is required.

Furthermore, a questioning strategy that is suitable for the *interleaved* approach should be explored. While the proposed method achieved the best performance in this paper, it appears not to be straightforward to use the condition $\Delta_q(\boldsymbol{\pi}) < 0$.

Acknowledgement

The Rakuten recipe database was provided by Rakuten Data Release from the Rakuten, Inc. This work was supported by JSPS KAKENHI Grant Number JP16K00229.

Table 4. Sums of mean numbers of questions and mean orders of intended candidates with *Bayesian strategy.*

Method	ϵ			
	0	0.1	0.2	0.3
5	27.197	49.116	71.865	98.952
6	28.976	50.356	72.316	98.333
7	23.243	48.790	73.405	100.95
8	22.046	44.097	66.483	93.366

Table 5. Mean numbers of questions with *Bayesian strategy.*

Method	ϵ			
	0	0.1	0.2	0.3
5	13.371	19.313	24.896	30.518
6	16	21	26	30
7	12.932	15.230	17.428	19.654
8	14.951	21.130	26.311	30.328

Table 6. Sums of mean numbers of questions and mean orders of intended candidates with *interleaving*.

Method	ϵ			
	0	0.1	0.2	0.3
9	21.448	39.685	60.866	87.916
10	21.930	43.233	65.965	93.152

Table 7. Mean numbers of questions with *interleaving*.

Method	ϵ			
	0	0.1	0.2	0.3
9	15.377	24.137	30.011	34.422
10	16.548	28.742	36.606	41.798

References

[1] Daisuke Furukawa, Sachiyo Muranishi, Tomoyuki Kojima, “Appropriate question strategy to an aphasic person (1) - Basic study of the question to narrow down (Shitsugoshosha heno tekisetsu na shitsumon-houhou ni tsuite (1) – shiborikomu shitsumon no kisoteki kenkyu),” Japanese association of Speech-Language-Hearing Therapists, Vol.8 (2007) (in Japanese).

[2] Shiho Arima, Shingo Kuroiwa, Yasuo Horiuchi, Daisuke Furukawa, “Question-Asking Strategy for People with Aphasia to Remember Food Names,” Journal on Technology & Persons with Disabilities, Vol.3 (2015).

[3] Andrzej Pelc, “Searching games with errors—fifty years of coping with liars,” Theoretical Computer Science, Vol.270, No.1-2 (2002).

[4] Suresh Raj Sunith, “A Bayesian Strategy to the 20 Question Game with Applications to Recommender Systems,” Doctoral dissertation, Duke University (2017).

[5] Yoko Doi, Miho Tsujita, Hidetsugu Nanba, Toshiyuki Takezawa, Kazutoshi Sumiya, “Construction of a Cooking Ontology from Cooking Recipes and Patents,” Technical report of IEICE. Multimedia and virtual environment, Vol.113, No.470 (2014).

[6] https://recipe.rakuten.co.jp/