Learning to Ask Efficiently in Dialogue: Reinforcement Learning Extensions for Stream-based Active Learning

Issei Waki, Ryu Takeda and Kazunori Komatani
SANKEN, The University of Osaka
8-1 Mihogaoka, Ibaraki, Osaka, Japan
{issei-waki@ei.,rtakeda@,komatani@}sanken.osaka-u.ac.jp

Abstract

One essential function of dialogue systems is the ability to ask questions and acquire necessary information from the user through dialogue. To avoid degrading user engagement through repetitive questioning, the number of such questions should be kept low. In this study, we cast knowledge acquisition through dialogue as stream-based active learning, exemplified by the segmentation of user utterances containing novel words. In stream-based active learning, data instances are presented sequentially, and the system selects an action for each instance based on an acquisition function that determines whether to request the correct labels from the oracle (in this case, the user). To improve the efficiency of training the acquisition function via reinforcement learning, we introduce two extensions: (1) a new action that performs semi-supervised learning, and (2) a state representation that takes the remaining budget into account. Our simulation-based experiments suggested that these two extensions have the potential to improve word segmentation performance with fewer questions for the user.

1 Introduction

Dialogue systems need the ability to ask questions and acquire necessary information from users through dialogue. For example, large language models (LLMs) do not necessarily cover local expressions, such as nicknames used among friends and family or abbreviations commonly used in schools or local communities. To enable natural and casual conversations with users, spoken dialogue systems must be able to handle these terms, including their spellings and pronunciations.

A key challenge lies in avoiding excessive or repetitive questions to maintain user engagement. While questioning the user is a reasonable strategy for acquiring such information (Li et al., 2017), general users—unlike crowdworkers—do not anticipate being asked similar or repetitive questions (Hancock et al., 2019). Therefore, the system should be carefully designed to minimize negative impressions and enhance overall user experience. Repeatedly asking the same type of question has been shown to degrade the user experience (Komatani et al., 2022).

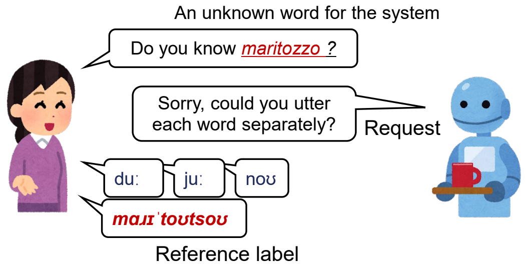

An unknown word for the system
Do you know *maritozzo* ?
Sorry, could you utter each word separately?
Request
duː
juː
noʊ
maɪɹˈtoʊtsoʊ
Reference label

Figure 1: Example of knowledge acquisition process

In this study, we formulate knowledge acquisition through dialogue as stream-based active learning (Tong and Koller, 2002; Settles, 2009), using the segmentation of user utterances as an example task. In active learning, the system selects instances that are expected to be particularly informative for improving a machine learning classifier (within a limited query budget) and obtains their reference labels from an oracle for supervised training. Stream-based active learning is a form of active learning in which data instances are presented sequentially, and at each step, the system makes a binary decision, i.e., whether to request the reference label or not, on the basis of an acquisition function. This corresponds to deciding whether to ask the user about the correct segmentation of a given utterance. Figure 1 illustrates such an interaction as an example. Previous research has shown that acquisition functions in stream-based active learning can be optimized using reinforcement learning (Fang et al., 2017). Note that, while we assume the user can act as an oracle and provide the correct segmentation

Proceedings of the 26th Annual Meeting of SIGDIAL, pages 431–440
Aug 25–27, 2025. ©2025 SIGDIAL

results, it remains challenging to accurately obtain such results from diverse user responses.

To improve the training efficiency of the acquisition function through reinforcement learning, we introduce two extensions. First, we add a new action, which we call self-learning, corresponding to semi-supervised learning: the system treats the current segmentation result as correct without querying the user, and utilizes it for training. Second, we incorporate the remaining query budget into the state representation. This enables the system to learn a strategy such as refraining from self-learning when the segmentation model is still unreliable, and relying on it more actively once the model has been sufficiently trained.

The contributions of this work are twofold:

- We formulate knowledge acquisition through dialogue as stream-based active learning, using the segmentation of user utterances that may contain unknown (novel) words as an example task (see Section 3).
- We propose reinforcement learning extensions to efficiently train acquisition functions in stream-based active learning, and provide preliminary evidence of their potential through simulation-based experiments (see Section 4).

2 Related Work

Life-long learning is an approach that enables a system to continually improve its performance (Silver et al., 2013; Chen and Liu, 2018). As part of this, research has explored acquiring factual knowledge through dialogue to overcome the limitations of fixed knowledge bases (Mazumder et al., 2019). Further challenges such as the handling of incorrect knowledge and the revision of previously acquired knowledge have also been discussed (Liu and Mazumder, 2021). Exploiting user utterances or feedback for supervision in dialogue has also been proposed. For example, a chatbot may extract new training examples from dialogue and solicit user feedback (Ono et al., 2017; Hancock et al., 2019).

In robotics, dialogue has been used as a means for robots to acquire new concepts such as object names or spatial terms (Taniguchi et al., 2016). An embodied robot may encounter novel objects, as well as unfamiliar place names and action concepts, in open-world settings. For example, robots have been trained to incrementally improve their language understanding and concept grounding through interaction with humans (Thomason et al., 2019), and to learn through clarification and active learning queries (Padmakumar and Mooney, 2021). One recent attempt also explores how a robot can learn unknown object names, locations, and actions through situated dialogue (Kane et al., 2022).

In neural response generation, techniques such as Retrieval-Augmented Generation (RAG) (Lewis et al., 2020) can incorporate a new word that appears during a dialogue into the generated text; however, similar to the robotics studies discussed above, our goal is to understand and acquire such knowledge in a way that allows it to be reused. While several studies have focused on storing user preferences or profiles during dialogue for the purpose of personalization (Cho et al., 2022; Chen et al., 2024), our objective is fundamentally different.

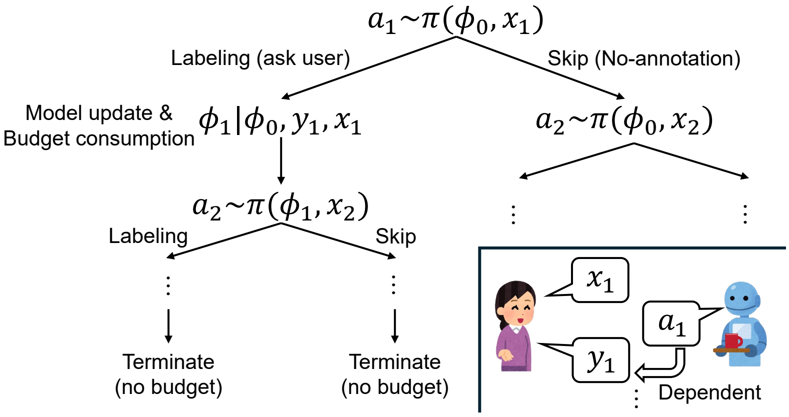
The framework of "asking intelligently" can be applied to lexical acquisition (Komatani et al., 2022), estimation of user satisfaction (Hancock et al., 2019), and user sentiment estimation (Karnjanapatchara et al., 2024). Furthermore, rather than selecting questions in active learning solely based on fixed uncertainty metrics, our framework learns an acquisition function via reinforcement learning, enabling the system to flexibly decide whether to ask a question by taking future rewards into account.

User impression given by a system's question is important. It is necessary to consider how the question will be perceived by the user when it is asked, for example, repeated explicit questions can quickly become annoying. Studies that address the impression of different types of system questions include, for example, (Komatani et al., 2022). Ideally, a system should balance the expected utility of asking a question with the impression it conveys.

3 Formulation

3.1 Process of Knowledge Acquisition through Dialogue

We assume that the knowledge acquisition through dialogue generally consists of three processes: 1) *question*, 2) *ask*, and 3) *understand*. First, the system should question what it could not understand in a user utterance given the dialogue context and the system knowledge. Second, the system should ask the user about it if necessary. Then, the system should understand what the user explained, which



$a_1 \sim \pi(\phi_0, x_1)$
Labeling (ask user)
Skip (No-annotation)
Model update & Budget consumption
$\phi_1 | \phi_0, y_1, x_1$
$a_2 \sim \pi(\phi_0, x_2)$
$a_2 \sim \pi(\phi_1, x_2)$
Labeling
Skip
Terminate (no budget)
Terminate (no budget)
x_1
a_1
y_1
Dependent

Figure 2: MDP in our scenario

results in knowledge update. In addition, a small number of confirmations is better because frequent questions will degrade user engagement and interrupt the conversation.

The process can be conceptually written as follows:

$$a \leftarrow \text{action}(x, \phi, h, c), \text{ and} \quad (1)$$
$$\phi \leftarrow \text{update}(\phi, x, y(a)|h) \text{ if } a = \text{“ask”}, \quad (2)$$

where x denotes a user utterance, and h and ϕ represent dialogue context and the system's current knowledge including model parameters, respectively. Note that the number of confirmation requests, c, in the dialogue is explicitly denoted here. The action$(\cdot)$ function is equivalent to the decision making function or *acquisition* function that determines whether or not to request the correct answer from the user for the understanding the utterance x. The 1) *question* process is included in this function. If the action of "ask" is decided, the 2) *ask* process is executed by the system. $y(a)$ denotes the user's response that includes a new correct knowledge for the system after the request a. The update$(\cdot)$ function reorganizes the current knowledge according to x and its correct informative response $y(a)$, which means the completion of the 3) *understand* process.

Our problem is how to design the action$(\cdot)$ and update$(\cdot)$ functions according to actual tasks. Active learning and reinforcement learning are important concepts and techniques to design or train the function using data. Since the update function usually boils down to a specific parameter update technique (e.g., the gradient descent method), the action function is more important for the knowledge acquisition process.

3.2 Stream-based Active Learning

3.2.1 Framework

We cast our knowledge acquisition process into the stream-based active learning problem (Atlas et al., 1989; Mnih et al., 2015; Fang et al., 2017). In this problem, the system takes unlabeled instances (data) in a stream and decides whether each instance should be manually annotated or not (*labeling* or *skip*). Here, the number of annotations is restricted by a *budget* variable that promotes the efficient annotation and learning. In our process, the unlabeled instance corresponds to the user utterance x, and the manual annotation corresponds to the answer from the user, $y(a)$, after the system takes the *ask* action, a = "ask". The budget variable also corresponds to the number of confirmations, c. The reinforcement learning framework is applied to learn the decision policy, i.e., the action$(\cdot)$ function in Eq.(1), based on data.

The formulation of stream-based active learning is based on a Markov Decision process (MDP) with budget constraint following the explanation of (Fang et al., 2017). This process is described by a state set $\mathcal{S}$, action set $\mathcal{A}$, reward function R, state transition probability P, and budget B. At the t-th turn, the user's utterance, the system's action, the current state, and the transition probability are represented by x_t, $a_t \in \mathcal{A}$, $s_t \in \mathcal{S}$, and $P(s_{t+1}|s_t)$, respectively. The reward function is denoted by $R(s_t, a_t)$. The action a_t usually takes a binary value: $a_t = 1$ means that the instance x_t is correctly labeled as y_t and is added to labeled data according to the policy π, and $a_t = 0$ means the system does nothing. The current model p_ϕ is updated by the labeled paired data that have already been obtained.

Figure 2 illustrates the MDP in our scenario.

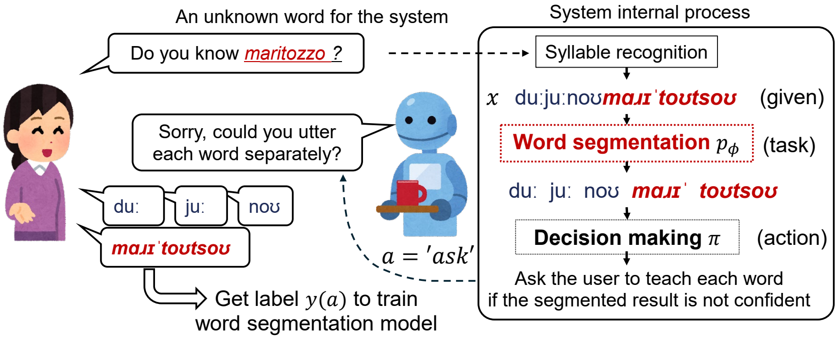

An unknown word for the system
Do you know maritozzo ?
Sorry, could you utter each word separately?
du:
ju:
noʊ
maɹɪˈtoʊtsoʊ
a = 'ask'
Get label y(a) to train word segmentation model
System internal process
Syllable recognition
x du:ju:noʊmaɹɪˈtoʊtsoʊ (given)
Word segmentation pϕ (task)
du: ju: noʊ maɹɪˈ toʊtsoʊ
Decision making π (action)
Ask the user to teach each word if the segmented result is not confident

Figure 3: Unknown word acquisition scenario: decision making process.

The process starts at the initial state (root), and the initial model p_ϕ with the parameter set ϕ_0. Given the first user's utterance x_1, the system decides whether this utterance should be labeled or not according to the policy π, e.g. $a_t \sim \pi(\phi_k, x_t)$. Here, ϕ_k represents a model parameter set after k updates. If the drawn action a_t takes 1, the correct label y_t is obtained. This process continues until the budget or instance is exhausted.

3.2.2 Deep Q-learning Implementation

The deep Q-learning enables us to estimate a policy function using deep neural network models. DQN is used to predict a Q-value function, $Q^\pi(s, a)$, that estimates the value of each action at state s in the Q-learning framework. It plays the role of action$(\cdot)$ in Eq.(1).

The cost function of the DQN used in (Mnih et al., 2015; Fang et al., 2017) is formulated as

$$L(\phi) = \mathbb{E}_{s,a,r,s'}[||y(r, s') - Q(s, a; \phi)||], \text{and} \quad (3)$$

$$y(r, s') = r + \gamma \max_{a'} Q(s', a'; \phi_k), \quad (4)$$

where γ denotes a discount rate for future rewards in the reinforcement leaning, and r and s' represent a reward and the next state of s, respectively. The expectation operator, $\mathbb{E}$, is taken over the mini-batch data drawn from the experience replay memory where each transition tuple (s, a, r, s') in an episode is stored. We assume here that the norm $|| \cdot ||$ is a smoothed L1 norm (Girshick, 2015) instead of an L2 norm.

We used the same intermediate reward utilized in (Fang et al., 2017). It is defined as the difference of the model performance before/after taking each action, i.e.,

$$r = \text{Acc}(\phi') - \text{Acc}(\phi_{k-1}), \quad (5)$$

where Acc$(\cdot)$ is a function that calculates the performance of the model with a given parameter set. ϕ' is equal to ϕ_{k-1} if the *skip* action is selected, otherwise ϕ_k. If the selected action impacts the performance improvement, a positive reward is obtained. The validation set is usually utilized to measure the performance.

4 Proposed Method

4.1 Word Segmentation Task toward Unknown-word Acquisition

Our scenario is an unknown word acquisition through spoken dialogue, and we focus on the word segmentation (WS) task as shown in Fig. 3. Here, the *unknown* words mean that they are not in the system's word-dictionary. In spoken dialogues, the recognition process of unknown words requires syllable recognition and WS processes because the system can recognize only the pronunciation of unknown words, not their spelling. The syllable recognition converts the audio signal into syllable sequences, and the syllable sequences are then segmented into words. Since the performance of WS directly affects the recognition of unknown words, we assume that the correct syllable recognition results are given in this paper.

We link this WS task to the stream-based active learning framework. The system segments the user utterance x into words using the WS model p_ϕ with a parameter set ϕ. If the system is not confident in the segmentation results caused by the existence of unknown words, it asks the user for the correct segmentation. Since we assume that the user response $y(a)$ corresponds to a correct label for x, the system can update the parameter of the WS model using the paired data (x, y), and it can also add the unknown words to its word-dictionary.

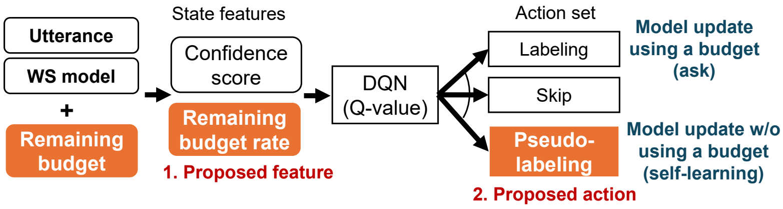


Figure 4: Overview of proposed method

4.2 New Action: Self-learning using Pseudo-labeling

We introduce a pseudo-annotation (labeling) (Lee, 2013; Arazo et al., 2020) into the set of actions $\mathcal{A}$ in the stream-based active learning. The pseudo-labeling means that the estimated label by the current model $p_{t,\phi}$ is used as if it is a true label in the model update. Therefore, we can perform *self-learning* (semi-supervised learning) using this pseudo-label to improve the model performance without consuming any budget, as illustrated on the right side of Fig. 4. Although this enables us to improve the training efficiency, inaccurate labels will degrade the model performance. This disadvantage is tackled in Section 4.3.

We extend our action to include three actions: skip ($a_t = 0$), labeling ($a_t = 1$), and pseudo-labeling ($a_t = 2$). If the action of pseudo-labeling is drawn in MDP, the pseudo label $\hat{y}_t$ is estimated by

$$\hat{y}_t = \text{argmax}_y \, p_{\phi_k}(y|x_t). \tag{6}$$

The actual model is explained in Section 4.4. The structure of DQN is also modified to predict three kinds of Q-values.

We explicitly restrict a state that allows the pseudo-labeling. This is because the pseudo-labeling is sometimes wrong, especially in cases where the true unknown-word is uttered due to over-/under-segmentation. Therefore, we permit the pseudo-labeling action only if there are segmented words that are not included in the word-dictionary.

4.3 New State: Remaining Budget Rate

The remaining budget is also added to the state feature to overcome the disadvantage of pseudo-annotation, as is illustrated in the center of Fig. 4. The action of pseudo-annotation in the early stage of model training usually degrades the performance due to estimation errors caused by insufficient training. Since the remaining budget could be a clue to distinguish the degree of model training, this new state is expected to promote pseudo-annotation in the later stage of training.

The actual state feature is the ratio of the current remaining budget and the maximum (initial) budget. It takes 1 at the start of learning and 0 at the end of learning.

4.4 WS Model and Other Settings

The WS model estimates the boundaries of words from a given syllabogram sequence $c_{1:L} = [c_1, ..., c_L]$ of length L. The boundaries are represented by a sequence of binary indicators $z_{1:L} = [z_1, ..., z_L]$. Here, z_l ($l = 1, ..., L$) takes 1 if the word boundary exists after the corresponding syllabogram c_l; otherwise, 0. The syllable sequence is divided into words according to the sequence of boundary indicators.

The sequence of indicators $z_{1:L}$ is estimated by evaluating the following posterior probability.

$$p(z_{1:L}|c_{1:L}; \phi_k), \tag{7}$$

where ϕ_k is a model parameter set. We can obtain N-best hypotheses by applying beam search or approximating posterior probability based on random sampling.

The confidence score of WS is used as a basic state feature of MDP. Since an estimation result with a low confidence score is basically suspicious, its correct label is expected to improve the model performance. This feature stems from the uncertainty sampling (Lewis and Gale, 1994). We calculate the confidence score by approximating the posterior probability based on random sampling from the model. The log score was actually used as the state feature.

We utilized sentence accuracy in terms of word boundaries as the Acc$(\cdot)$ function in Eq.(5). It represents the ratio of correctly segmented sentences (utterances) to the total number of sentences in a given dataset. This function implicitly gives greater weight to shorter sentences during training, as they contain fewer words and thus have a lower risk

Table 1: Statistics of dialogue corpus.

	Train	Valid	Test -train	Test -eval
No. of utterances	192	65	192	192
No. of uttrs. with unknown words	104	37	113	115

of mis-segmentation. This serves as a strategy to reliably understand simple utterances from users.

5 Experimental Setting

5.1 Dataset

In our experiments, we used the transcriptions of the utterances usually utilized for ASR model training as the training set for the WS model. The unit of word was the Japanese *short unit* annotated in the CSJ core set (Maekawa et al., 2000). These transcriptions included 380,872 utterances and 7,402,147 word tokens. The number of characters in the Katakana format was 14,789,778, and the vocabulary size was 53,866.

The dataset used for stream-based active learning consists of four datasets: 1) a training set (train), 2) a validation set (valid), 3) a test training set (test-train), and 4) a test set (test-eval). Sets 1. and 2 were used for the reinforcement learning process (training of DQN), and Sets 3 and 4 were used for the evaluation of the active learning process and the performance evaluation of the WS model using the trained DQN. In detail, the valid set was used to evaluate the reward for DQN. We randomly split the following spoken dialogue corpus into train, valid, test-train and test-eval sets at the ratio of 3:1:3:3.

We utilized the transcriptions in our spoken dialogue corpus under unknown word acquisition scenarios. In these scenarios, a user uttered an unknown word for the system, and the system confirmed the unknown word with the user. Words associated with *food names*, such as Maritozzo (Italian sweets) and Semifreddo (Italian sweets), were assumed to be unknown words that were not included in the train set. The number of utterances was 641 in total, of which 369 included unknown words. Table 1 summarizes these configurations.

5.2 Model and Training Configurations

The structure of our DQN is based on a three-layered perceptron as shown in Fig.5. The dimensions of DQN input N_s and the state feature were the same: one state of confidence score as a baseline, and two states of confidence score and the remaining budget rate as proposed method. The number of actions was set to the output dimension N_a of DQN: two actions as a baseline, and three actions as the proposed method. The number of middle nodes was 64, and ReLU was used for the activation function. The mean and variance normalization was applied as the transformation of the first layer of DQN. The parameters for the feature of the confidence score were determined by using the valid set in advance, and they were frozen during the DQN training. The mean and variance parameters for the feature of the remaining budget rate were set to 0.5.

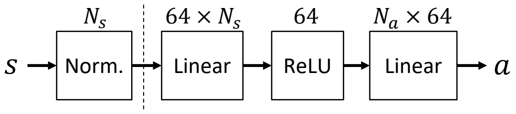


Figure 5: Structure of deep Q-network.

The hyperparameters during DQN training were as follows. The number of episodes was 300, and the maximum budget per episode was set to 25. The reward was defined as the difference between the accuracy of word segmentation before and after the actions: positive rewards if the performance was improved, negative rewards if the performance was degraded, and no rewards otherwise. The discount rate γ was 0.99, and the learning rate was 10^{-6}. The epsilon-greedy strategy was applied using random section of actions with the probability $1 - \epsilon$. The initial value of ϵ was 0.75, and it rose to 1.0 within 250 episodes. We applied an early-stopping method after 250 episodes and selected the model that maximizes the total rewards. The optimizer was AdamW (Loshchilov and Hutter, 2018) and the mini-batch size was 32 with the replay buffer (Lin, 1992) of $10,000$.

The nested Pitman-Yor language model (NPYLM) (Mochihashi et al., 2009) was applied as the word segmentation model based on a Bayesian probabilistic model. This model was chosen because its parameters can be updated instantaneously by changing the N-gram counts (except for latent variables), i.e., there is no iterative process for parameter updates. This property is suitable for the dynamic acquisition of unknown words during the dialogue. The initial model parameters were estimated using the training set of CSJ set.

Table 2: Results: AUC, maximum AUC and sample efficiency.

	AUC (↑)	Max. AUC (↑)	Sample Efficiency (↑)
Baseline	21.02 ± 0.15	21.19	0.014 ± 0.001
Proposed	20.58 ± 0.50	21.25	0.018 ± 0.002
w/o pseudo-labeling	20.60 ± 0.53	21.14	0.017 ± 0.003
w/o budget state feature	20.85 ± 0.24	21.10	0.015 ± 0.001
Oracle: optimal threshold	21.19	21.19	0.016

5.3 Evaluation Metrics

We choose two kinds of evaluation criteria: area under the curve (AUC), to show the overall performance of word segmentation over the course of budget consumption, and sample efficiency, to show how rapidly the model reaches its peak performance. In both criteria, larger values indicate better performance.

AUC represents the area under the curve of word segmentation accuracy plotted against the number of consumed budgets. In the context of active learning, we use AUC to evaluate the performance of DQN learning, as a higher AUC indicates that the accuracy reaches its upper limit with minimal budget consumption. In this experiment, the maximum AUC value is 25, as the total budget was set to 25.

Sample efficiency represents the ratio of the total accuracy improvement to the number of budgets used. If the initial and peak accuracies of word segmentation are denoted by u_1 and u_2, respectively, then the efficiency is calculated as $(u_2 - u_1)/b_r$, where b_r is the number of budgets consumed to reach u_2.

5.4 Results

Figure 6 shows the trajectory of the sentence accuracy on the test-eval set with respect to the consumed budgets during the active learning using the test-train set. The mean and standard deviation (std.) of the accuracy at each budget step, averaged over ten parameter sets, are shown. Table 2 presents a quantitative summary of these results based on the evaluation metrics described in Section 5.3.

In terms of AUC, the proposed method performed comparably to, or slightly worse than, the baseline on average, as shown in the first column of Table 2. Nevertheless, the effectiveness of the proposed method can be better understood from the learning curves in Figure 6. First, the accuracy improved at approximately 20 along the horizontal axis without consuming any budget, thanks to the proposed pseudo-labeling action. This action was indeed frequently selected during the later phase of learning. Second, regarding performance convergence, the proposed method reached the peak accuracy (around 0.92) earlier than the baseline. Although the proposed method exhibited higher accuracy in both the early and late phases, the baseline outperformed it in the middle phase. This explains why the overall AUC remained similar to that of the baseline.

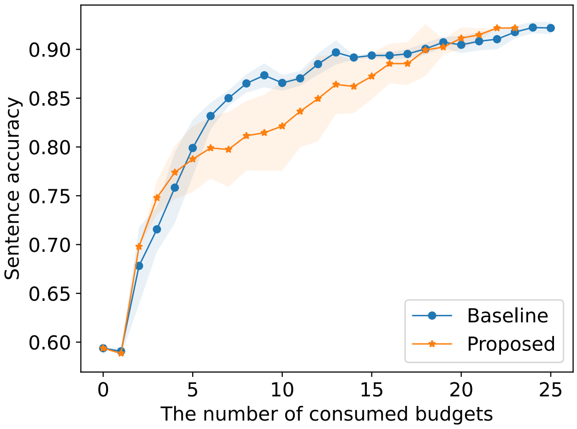


Figure 6: Improvement curve. Mean and std. are shown.

We further provide the maximum AUC value over ten trials in the second column of Table 2. The maximum AUC of the proposed method (21.25) outperformed that of the baseline (21.19). The reason for the higher maximum but the lack of improvement in average performance can be attributed to the large standard deviation observed across its trials. The larger standard deviation indicates that the proposed method is sensitive to randomness, such as the initial values in DQN. In particular, the pseudo-labeling appears to increase performance variance, as seen from the standard deviation of AUC in Table 2. Nevertheless, this also suggests that our method has room for improvement in learning a better policy by utilizing a validation set in the context of active learning (e.g., *test-valid* set). This could reliably help stabilize the training of DQN, which should be addressed in future work.

In terms of sample efficiency, the proposed method outperformed the baseline, as shown in the third column of Table 2. The sample efficiency of our method was higher by 0.004 compared to that of the baseline. This indicates that the proposed method reached higher performance more quickly, meaning that it can utilize samples more efficiently.

As part of the ablation study, we present in Table 2 the performance of our method without the pseudo-labeling and without the budget state feature. The budget state feature was shown to improve the sample efficiency, as observed by comparing the values between the proposed method and the variant without the budget state feature. We also observe that the maximum AUC decreased when either the pseudo-labeling action or the budget state feature was removed. These results suggest that both components of our proposed method contributed to improved performance.

We also tested the optimal confidence threshold as a reference point, representing an oracle strategy based only on confidence scores. The threshold of -0.51 was selected using the test-eval set so as to maximize segmentation performance. This setup corresponds to a best-case scenario of uncertainty sampling (Lewis and Gale, 1994), where labeling is determined by applying a threshold to the confidence score. Notably, the maximum AUC of the proposed method outperformed that of the oracle, which further supports the effectiveness of the pseudo-labeling. Given the mean AUC gap of 0.61 between the proposed method and the oracle, there remains room to improve the efficiency of the learning policy based on confidence scores, as discussed previously.

6 Conclusion

In this study, we cast knowledge acquisition through dialogue as a stream-based active learning problem. Using the segmentation of user utterances containing novel words as a target task, we proposed two extensions in reinforcement learning to improve the efficiency of training the acquisition function.

Simulation-based experiments showed the potential of the proposed method to improve the segmentation performance while reducing the number of questions asked to the user. These results suggest that the system can strategically learn to ask questions in order to improve its performance with minimal user effort.

An important direction for future work is to evaluate the proposed method with human users rather than through simulations. It is crucial to assess not only the performance of the task-dependent model but also the human impression of the interaction. For instance, the current question strategy is based on the assumption that users will correctly provide all isolated words for word segmentation. However, this assumption is neither realistic nor user-friendly in real interactions. Therefore, it is necessary to design question strategy that minimizes the user effort while maximizing the model's performance.

In addition, stabilizing the training of DQN by introducing a validation set remains an important issue. This is because the performance of DQN often depends on several factors, including the initial values of network weights, the permutation of the instances in the training and test sets, and reinforcement learning hyperparameters such as ϵ in the epsilon-greedy strategy.

Memory limitation is a potential issue for stream-based active learning during dialogue, especially in life-long dialogue scenarios. While the memory capacity is typically limited, the number of paired instances and their obtained labels grows monotonically over time. As a result, when the memory reaches its capacity, it may be necessary to replace stored data with new paired data based on their *importance*. On the other hand, certain models like NPYLM, which we used in this paper, allow the recursive update of parameters and thus can mitigate the memory limitation issue. For instance, it is not necessary to store all paired data for updating the *word-count* parameters in NPYLM because these parameters are simply incremented for each paired data (there are no gradient computation and iteration process).

Limitations

Although our formulation of knowledge acquisition through dialogue is designed to be general, the details inevitably depend on each task. This paper showed the potential of our formulation and its application to the scenario of unknown word acquisition. However, several aspects require further consideration and investigation, including scalability to diverse datasets and tasks, as well as overall generality.

For the unknown word acquisition task, the dataset and evaluation patterns were limited with respect to scalability, dependency on dataset size, word segmentation methods, and language. Since the difficulty of word segmentation depends on the size of the word dictionary (i.e., the number of words known by the system), a more detailed performance investigation will support the development of practical dialogue systems. Additionally, the design of states, actions, and network structure still has room for improvement, particularly concerning the high variance in performance.

Extending evaluations to other tasks and task-dependent models is essential to establish the generality of our approach based on stream-based active learning, even though the controlled setup in our experiment helps isolate the effects of our strategy. However, demonstrating generality remains a significant challenge due to the dependency of effective state descriptions and question design on specific tasks and models. For example, tasks such as user sentiment estimation, word meaning estimation, named entity recognition, and knowledge base expansion require task- and model-specific designs. Furthermore, the optimal model for each task will vary and evolve as technologies advance. Nonetheless, we believe that the label (information) obtained from users during dialogue will contribute to supervised model training and enhance model performance.

Ethical Considerations

Dialogue-based knowledge acquisition systems that learn by asking users need to consider the potential risk of acquiring biased or inappropriate information in general, whereas the segmentation of user utterances as addressed in this paper has negligible ethical concerns.

Acknowledgments

We thank the reviewers for their constructive discussions and valuable feedback.

This work was partly supported by JSPS KAKENHI Grant Numbers JP23K28147 and JP22H00536, and JST Moonshot R&D Grant Number JPMJM2011, Japan.

References

Eric Arazo, Diego Ortego, Paul Albert, Noel E. O' Connor, and Kevin McGuinness. 2020. Pseudo-labeling and confirmation bias in deep semi-supervised learning. In *Proc. International Joint Conference on Neural Networks (IJCNN)*, pages 1–8.

Les Atlas, David Cohn, and Richard Ladner. 1989. Training connectionist networks with queries and selective sampling. In *Proc. Advances in Neural Information Processing Systems*, volume 2, pages 566–573.

Yi-Pei Chen, Noriki Nishida, Hideki Nakayama, and Yuji Matsumoto. 2024. Recent trends in personalized dialogue generation: A review of datasets, methodologies, and evaluations. In *Proc. Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING)*, pages 13650–13665.

Zhiyuan Chen and Bing Liu. 2018. *Lifelong Machine Learning, Second Edition*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers.

Itsugun Cho, Dongyang Wang, Ryota Takahashi, and Hiroaki Saito. 2022. A personalized dialogue generator with implicit user persona detection. In *Proc. International Conference on Computational Linguistics (COLING)*, pages 367–377.

Meng Fang, Yuan Li, and Trevor Cohn. 2017. Learning how to active learn: A deep reinforcement learning approach. In *Proc. Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 595–605.

Ross Girshick. 2015. Fast r-cnn. In *Proc. IEEE International Conference on Computer Vision (ICCV)*.

Braden Hancock, Antoine Bordes, Pierre-Emmanuel Mazare, and Jason Weston. 2019. Learning from dialogue after deployment: Feed yourself, chatbot! In *Proc. Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 3667–3684.

Benjamin Kane, Felix Gervits, Matthias Scheutz, and Matthew Marge. 2022. A system for robot concept learning through situated dialogue. In *Proc. Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, pages 659–662.

Thus Karnjanapatchara, Sixia Li, Candy Olivia Mawalim, Kazunori Komatani, and Shogo Okada. 2024. Incremental multimodal sentiment analysis on hai based on multitask active learning with inter-annotator agreement. In *Proc. International Conference on Affective Computing and Intelligent Interaction (ACII)*.

Kazunori Komatani, Kohei Ono, Ryu Takeda, Eric Nichols, and Mikio Nakano. 2022. User impressions of system questions to acquire lexical knowledge during dialogues. *Dialogue and Discourse*, 13(1):96–122.

Dong-Hyun Lee. 2013. Pseudo-label : The simple and efficient semi-supervised learning method for deep neural networks. In *Proc. ICML 2013 Workshop: Challenges in Representation Learning (WREPL)*.

David D Lewis and William A Gale. 1994. A sequential algorithm for training text classifiers. In *Proc. Annual international ACM SIGIR conference on Research and development in information retrieval*, pages 1–10.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Proc. Advances in Neural Information Processing Systems (NeurIPS)*.

Jiwei Li, Alexander H. Miller, Sumit Chopra, Marc'Aurelio Ranzato, and Jason Weston. 2017. Learning through dialogue interactions by asking questions. In *Proc. International Conference on Learning Representations*.

Long-Ji Lin. 1992. Self-improving reactive agents based on reinforcement learning, planning and teaching. *Machine learning*, 8:293–321.

Bing Liu and Sahisnu Mazumder. 2021. Lifelong and continual learning dialogue systems: Learning during conversation. In *Proc. Conference on Artificial Intelligence (AAAI)*, pages 15058–15063.

Ilya Loshchilov and Frank Hutter. 2018. Decoupled weight decay regularization. In *Proc. International Conference on Learning Representations (ICLR)*.

Kikuo Maekawa, Hanae Koiso, Sadaoki Furui, and Hitoshi Isahara. 2000. Spontaneous speech corpus of Japanese. In *Proc. International Conference on Language Resources and Evaluation (LREC)*.

Sahisnu Mazumder, Bing Liu, Shuai Wang, and Nianzu Ma. 2019. Lifelong and interactive learning of factual knowledge in dialogues. In *Proc. Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, pages 21–31.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. 2015. Human-level control through deep reinforcement learning. *Nature*, 518:529–533.

Daichi Mochihashi, Takeshi Yamada, and Naonori Ueda. 2009. Bayesian unsupervised word segmentation with nested Pitman-Yor language modeling. In *Proc. Joint Conference of the 47th Annual Meeting of the ACL and the International Joint Conference on Natural Language Processing of the AFNLP*, pages 100–108.

Kohei Ono, Ryu Takeda, Eric Nichols, Mikio Nakano, and Kazunori Komatani. 2017. Lexical acquisition through implicit confirmations over multiple dialogues. In *Proc. Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, pages 50–59.

Aishwarya Padmakumar and Raymond J. Mooney. 2021. Dialog policy learning for joint clarification and active learning queries. In *Proc. Conference on Artificial Intelligence (AAAI)*, pages 13604–13612.

Burr Settles. 2009. Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin-Madison, Madison, WI.

Daniel L. Silver, Qiang Yang, and Lianghao Li. 2013. Lifelong machine learning systems: Beyond learning algorithms. In *AAAI Spring Symposium*.

Akira Taniguchi, Tadahiro Taniguchi, and Tetsunari Inamura. 2016. Spatial concept acquisition for a mobile robot that integrates self-localization and unsupervised word discovery from spoken sentences. *IEEE Transactions on Cognitive and Developmental Systems*, 8(4):285–297.

Jesse Thomason, Aishwarya Padmakumar, Jivko Sinapov, Nick Walker, Yuqian Jiang, Harel Yedidsion, Justin Hart, Peter Stone, and Raymond J. Mooney. 2019. Improving grounded natural language understanding through human-robot dialog. In *Proc. International Conference on Robotics and Automation (ICRA)*, page 6934–6941.

Simon Tong and Daphne Koller. 2002. Support vector machine active learning with applications to text classification. *J. Mach. Learn. Res.*, 2:45–66.