Interactive Question Clarification in Dialogue via Reinforcement Learning

Xiang Hu[†] **Zujie Wen**[†] **Yafang Wang** [†*] **Xiaolong Li**[†] **Gerard de Melo**[‡]

Ant Financial Services Group[†]
Hasso Plattner Institute, University of Potsdam[‡]

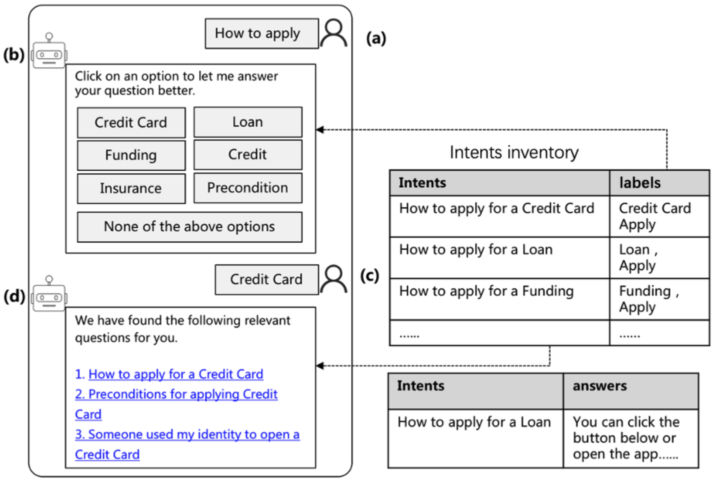
Abstract

Coping with ambiguous questions has been a perennial problem in real-world dialogue systems. Although clarification by asking questions is a common form of human interaction, it is hard to define appropriate questions to elicit more specific intents from a user. In this work, we propose a reinforcement model to clarify ambiguous questions by suggesting refinements of the original query. We first formulate a collection partitioning problem to select a set of labels enabling us to distinguish potential unambiguous intents. We list the chosen labels as intent phrases to the user for further confirmation. The selected label along with the original user query then serves as a refined query, for which a suitable response can more easily be identified. The model is trained using reinforcement learning with a deep policy network. We evaluate our model based on real-world user clicks and demonstrate significant improvements across several different experiments.

1 Introduction

In real-world dialogue systems, a substantial portion of all user queries are ambiguous ones for which the system is unable to precisely identify the underlying intent. We observed that many such queries in our question answering (QA) system exhibited one of the following two characteristics.

1. Lack of semantic elements such as subject, object, or predicate, e.g. “How to apply”, “Credit card”.
2. Ambiguous entities, e.g. “My health insurance” (because health insurance consists of numerous sub-categories).



Click on an option to let me answer your question better.

We have found the following relevant questions for you.

1. How to apply for a Credit Card
2. Preconditions for applying Credit Card
3. Someone used my identity to open a Credit Card

Intents	labels
How to apply for a Credit Card	Credit Card Apply
How to apply for a Loan	Loan , Apply
How to apply for a Funding	Funding , Apply
......	......

Intents	answers
How to apply for a Loan	You can click the button below or open the app......

Figure 1: Interactive question clarification example. a) The user provides an incomplete or ambiguous question. b) The agent suggests pertinent labels. c) The user confirms by selecting one such label. d) The agent considers the label in conjunction with the original query as a refined query and responds to it.

[*] corresponding author, email: yafang.wyf@antfin.com

Proceedings of the 28th International Conference on Computational Linguistics: Industry Track, pages 78–89
Barcelona, Spain (Online), December 12, 2020

Given such limited information, it is difficult for a system to accurately respond to a user's ambiguous queries, often resulting in that the user's needs cannot be addressed. For example, the specific intent underlying an utterance such as "How to apply?" remains obscure, because there are too many products related to the action of "applying". In practice, one often needs to fall back to human agents to assist with such requests, increasing the workload and cost. The main purpose of deployed automated systems is to reduce the human workload in scenarios such as customer service hotlines. The lack of an ability to deal with ambiguous questions may directly lead to these sessions being transferred to human agents. In our real-world customer service system, this affects up to 30% of sessions. Hence, it is valuable to find an effective solution to clarify such ambiguous questions automatically, greatly reducing the number of cases requiring human assistance.

Automated question clarification involves confirming a user's intent through interaction. Previous work has explored asking questions (Radlinski and Craswell, 2017; Quarteroni and Manandhar, 2009; Rao and Daumé, 2018; Rao and Daumé, 2019). Unfortunately, clarification by asking questions requires substantial customization for the specific dialogue setting. It is challenging to define appropriate questions to guide users towards providing more accurate information. Coarse questions may leave users confused, while overly specific ones may fail to account for the specific information a user wishes to convey.

In our work, we thus instead investigate interactive clarification by providing the user with specific choices as options, such as intent options (Tang et al., 2011). Unlike previous work, we propose an end-to-end model that suggests labels to clarify ambiguous questions. An example of this sort of approach is given in Figure 1. Here, we consider a closed-domain QA system, where a typical method is to build an intent inventory to address high-frequency requests. In this setting, the set of unambiguous candidate labels for an ambiguous user utterance corresponds to a set of frequently asked questions covered by the intent inventory. In a closed domain, we consider the candidate set to be finite. For example, in Figure 1, there are three specific intents corresponding to the ambiguous question "How to apply".

Our approach induces phrase tags as *labels* for each intent. Thus, we have a catalog of intents with corresponding labels that can be presented to the user. The challenge lies in selecting a suitable list of labels that can effectively clarify the ambiguous question. In our approach, the problem of finding the label sequence is formulated as a collection partitioning problem, where the objective is to cover as many elements as possible while distinguishing elements as clearly as possible. The task of question clarification thus amounts to obtaining a suitable set of labels.

The main contributions of our work are:

1. We formulate interactive clarification as a collection partitioning problem.
2. We propose a novel reward function to evaluate the clarification ability of phrase collections and an end-to-end sequential phrase recommendation model trained with reinforcement learning.
3. Both offline and online experiments confirm that our method outperforms pertinent baselines significantly.

2 Related Work

Query Refinement. Several works explore the use of clarification questions for query refinement (Kotov and Zhai, 2010; Sajjad et al., 2012; Zheng et al., 2011; Ma et al., 2010; Sadikov et al., 2010). For instance, Kotov and Zhai (2010) and Sajjad et al. (2012) use question templates to generate a list of clarification questions. Elgohary et al. (2019) rewrite questions using the dialogue context. Zhang et al. (2019) invoke graph edit distance for query refinement. Other studies rely on reinforcement learning to refine user queries (Nogueira and Cho, 2017; Buck et al., 2018; Liu et al., 2019), but consider queries that are unambiguous (though possibly ill-formed or non-standard). Accordingly, they seek to increase the recall, while in our setting, we consider ambiguous user queries, and our model primarily seeks to address the task of question clarification.

Dialogue. Boni and Manandhar (2003) developed an algorithm to recognize clarification dialogue, rather than for asking clarification questions. Varges et al. (2010) found that the use of clarification has a positive effect on concept precision in task-oriented dialogue. Li et al. (2017) focus on clarification in the specific circumstance of a bot not understanding a teacher because of spelling mistakes, which is a sub-problem of our setting. Zhang et al. (2018) generate clarification questions using language patterns with predicted aspect. They do not use reinforcement learning to optimize the order of the questions. Wang et al. (2018) devised soft and hard-typed decoders to generate good questions by capturing different roles of different word types. Aliannejadi et al. (2019) designed a two-stage retrieval and ranking model to rank clarification question candidates generated by human annotators, different from our end-to-end reinforcement learning approach. Korpusik and Glass (2019) construct clarification questions from a food attribute list (brand, fat, etc.). They rely on a hybrid reinforcement learning approach to select the order of clarification questions to ask, while we present an end-to-end reinforcement learning method.

Question Answering. Some studies focus on clarification questions in a community question answering setting (Braslavski et al., 2017; Rao and Daumé, 2018; Rao and Daumé, 2019). These share in common that they seek to rank or generate clarification questions, while our approach uses reinforcement learning to perform sequential label recommendation for question clarification. The key differences between our work and Tang et al. (2011) are three-fold. First, they rely on an ontology, which limits the applicability of their approach in real-world deployments and prevents us from being able to compare against their approach in our experiments, since each domain requires a custom ontology. Second, they cluster the keywords through the ontology, based on templates to achieve a refinement of questions, without using machine learning. Third, they rely on clustering to increase the keyword diversity, while we design a reward with an information gain term that automatically encourages diversity.

3 Preliminaries

System overview. In order to provide a more concrete picture of our approach, we first briefly describe our QA system, illustrated in Figure 2, as an example of how this approach can be instantiated.

When the conversation exceeds a certain number of rounds or the user explicitly requests human service, the conversation is transferred to a human customer service agent. In this setting, our clarification method chiefly serves to reduce the workload of those human agents. In our real system, there are two stages: label clarification and intent retrieval as illustrated in Figure 1. The label clarification stage provides 6 labels for the user to confirm. Upon selecting one of the suggested labels, the user question is concatenated with the selected label phrase as a new query input. The intent retrieval stage seeks to provide 3 relevant intents for the user to select according to the concatenated query. These additional labels can help clarify and improve the relevance.

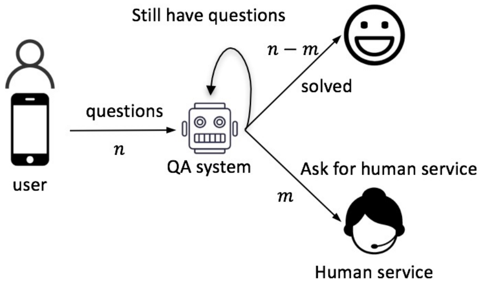


Figure 2: Pipeline of our QA system. $\frac{m}{n}$ is the rate of transferal to human agents (THA).

Intent and Label Inventory. Our system relies on a closed-domain intent and label inventory. The intents along with their corresponding answers are compiled by human experts. The set of labels is a collection of words or phrases that are manually constructed from intents by marking up keywords such as suitable predicates, subjects, or objects. As shown in Figure 3, there is a many-to-many relationship between intents and labels. Note that there is substantial synonymy among the set of labels, which may result in numerous repetitive recommendation results. Thus, ensuring the diversity of the results ought to be a factor in the design of the policy model.

Dataset Setup. In order to solve the cold start problem and evaluate the effectiveness of each model offline, we constructed a benchmark corpus. This annotated corpus consists of 40k ambiguous questions and their potential intents. For this, ten experts were divided into five teams. The two experts in each team annotate the same corpus. Data on which there are disagreements are

Recall	Valid
Can't transfer money using Alpha	No
How to apply for a Credit Card	Yes
How to apply for a Loan	Yes
...	...

Table 1: Example of related intent annotation for user question "How to apply".

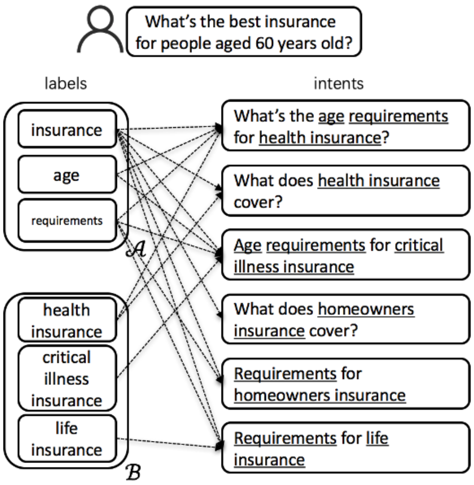

What's the best insurance for people aged 60 years old?
labels
intents
insurance
age
requirements
$\mathcal{A}$
health insurance
critical illness insurance
life insurance
$\mathcal{B}$
What's the age requirements for health insurance?
What does health insurance cover?
Age requirements for critical illness insurance
What does homeowners insurance cover?
Requirements for homeowners insurance
Requirements for life insurance

Figure 3: Example of relationship between labels and intents. A and B are different label groups to divide potential intents.

annotated anew, and only agreed-upon data is selected. To construct corpora at a relatively low cost, the annotation task is simplified so as to merely elicit a "yes" or "no" response. The whole annotation process is divided to two stages. At the first stage, we collect ambiguous questions by annotating online query logs. If a query lacks a predicate or the object of the predicate, it is annotated as ambiguous. At the second stage, we annotate potential intents for each ambiguous question. As Table 1 shows, for each ambiguous question ("How to apply"), the top 50 most relevant intent candidates are collected using the BERT (Devlin et al., 2019) semantic similarity model applied to the intent inventory. The human annotators are asked to decide whether an intent can possibly address a user's question.

4 Reinforcement Learning for Label Recommendation

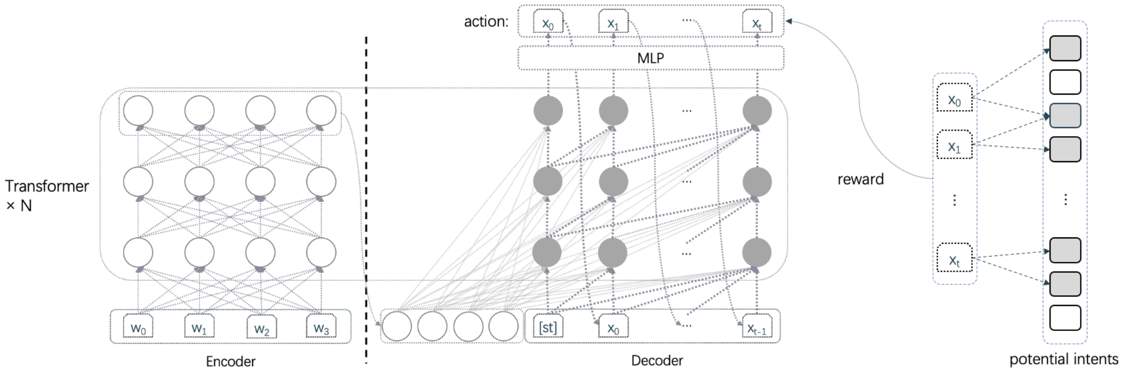


Figure 4: Label recommendation policy model architecture.

Label Recommendation as an RL problem. In order to train a model able to recommend labels one by one, we have two options: 1) Deduce a path reversely for supervised learning. 2) Create an environment for the model to explore. We believe that creating an environment for the model to explore different label sequences may lead to better generalization ability, which is confirmed in our comparative experiments. We can cast our label recommendation in the reinforcement learning paradigm as in Figure 4. Our model can be viewed as an *agent* that interacts with an *environment*, which consists of the user question and recommended labels. The action space consists of more than 1,000 candidate labels, out of which a suitable next label needs to be selected as a next action. In order to increase the diversity and reduce the number of synonymous labels, our model takes historical recommended labels into account. Upon having recommended N labels, the final reward (introduced later) is assigned and the parameters are updated.

Policy Model. As N labels to be recommended could be considered as a sequence, we use a seq2seq architecture to model the problem. As shown in Figure 4, in the encoder stage, the query is encoded by BERT and a vector representation is generated. In the decoder stage, the input at time step t is the action at step $t-1$ (step 0 is [st]). For each step, one-way multi-head attention (Vaswani et al., 2017) is applied on previously recommended labels and the vector representation of the input query. Finally, the action probability at each step is estimated.

Rewards. Intuitively, the chosen labels ought to maximize the recall of the intents with regard to the human-annotated potential intents. However, a trajectory with high recall may not be sufficient for clarification, as high recall can easily be achieved by suggesting labels such as in group A in Figure 3. Rather, a good label set should efficiently discriminate between potential intents as in group B in Figure 3. We recast this as a collection partition problem. Subsequently, inspired by the ID3 algorithm (Quinlan, 1986), we use Information Gain as a term to evaluate the final reward.

Formally, given a user query q, and the human-annotated potential intents $\mathcal{Q}(q)$, our policy model selects a list of labels $\tau_N = \{x_1, x_2, \ldots, x_N\}$. We map all the chosen labels τ to the retrieved potential intent set $S(\tau)$ with a many-to-many relationship between labels and intents:

$$\mathcal{S}(\tau) = \bigcup_{x \in \tau} \left[\mathcal{M}(x) \cap \mathcal{Q}(q)\right] \tag{1}$$

$\mathcal{M}(x)$ denotes the intent set mapped from label x. $\mathcal{K}$ denotes the universe set of intents. An indicator vector $\mathbf{I}(q) = (\mathbf{I}_1, \mathbf{I}_2, \ldots, \mathbf{I}_{|\mathcal{K}|})$ indicates for each intent s^i in $\mathcal{K}$ whether it exists in the human-annotated intent set $\mathcal{Q}(q)$, as defined below.

$$\mathbf{I}_i = \begin{cases} 1 & s^i \in \mathcal{Q}(q) \\ 0 & s^i \notin \mathcal{Q}(q) \end{cases} \tag{2}$$

The probability that an intent is the answer to an ambiguous question is computed as

$$P(s^i \mid q) = \frac{\mathbf{I_i}}{|\mathcal{Q}(q)|}. \tag{3}$$

We define potential intents recalled at time step t as $S(\tau_t)$, the conditional entropy of $S(\tau_N)$ is $\mathcal{H}(\tau_N)$, defined as follows.

$$\begin{aligned} &\mathcal{D}(x_t) = \mathcal{M}(x_t) \cap \mathcal{Q}(q) \setminus \mathcal{S}(\tau_{t-1}) \\ &\widetilde{P}(s \mid q, \tau_t) = \frac{P(s \mid q)}{\sum\limits_{s' \in \mathcal{D}(x_t)} P(s' \mid q)} \\ &\mathcal{H}(x_t) = -\sum_{s \in \mathcal{D}(x_t)} \widetilde{P}(s \mid q, \tau_t) \log \widetilde{P}(s \mid q, \tau_t) \end{aligned} \tag{4}$$

Here, $\mathcal{M}(x_t)$ denotes the set of intents mapped from label x_t. $\mathcal{D}(x_t)$ is the marginal recall over the potential intent set $\mathcal{Q}(q)$ for label x_t. $\widetilde{P}(s \mid q, \tau_t)$ is the normalized probability of $P(s \mid q)$ for intents in $\mathcal{D}(x_t)$. The entropy at time step 0 is $\mathcal{H}_0$, defined as

$$\mathcal{H}_0 = -\sum_{s \in \mathcal{Q}(q)} P(s \mid q) \log P(s \mid q). \tag{5}$$

The Information Gain is defined as

$$\Delta(\tau_N) = \sum_{t=1}^{N} \frac{|\mathcal{D}(x_i)|}{|\mathcal{S}(\tau_N)|} \mathcal{H}(x_t) - \mathcal{H}_0, \tag{6}$$

and the final reward is then defined as

$$R(\tau_N) = \sum_{s \in S(\tau_N)} P(s \mid q) + \beta \Delta(\tau_N). \tag{7}$$

In our experiments, β by default is set to 1.

Considering there are more than 1000 candidate labels, the size of the search space in MCTS may explode. To reduce its size, we only sample labels in $\{x | \mathcal{M}(x) \bigcap \mathcal{Q}(q) \neq \varnothing\}$ because only such labels have a relationship with candidate intents worth exploring. Thus, the size of the search space is drastically reduced.

Training. The policy model to suggest labels is trained from samples generated via a Monte-Carlo tree search (MCTS) (Coulom, 2006; Kocsis and Szepesvári, 2006; Browne et al., 2012). The MCTS starts from an empty label set and stops when the trajectory includes N labels, as in Figure 5.

Each simulation starts from the root state and iteratively selects a move with maximal $V(\cdot)$, which is computed according to the upper confidence bound for tree search (Kocsis and Szepesvári, 2006) as

$$V(v) = \frac{Q(v)}{N(v)} + \beta_{\mathrm{T}} \sqrt{\frac{2 \ln N(p_v)}{N(v)}}, \tag{8}$$

where p_v denotes the parent of v and β_{T} by default is set to 1. After a path has been sampled, the Q value of each node in the path is updated according to

$$Q(v) = \frac{\sum_{\tau \in T(v)} R(\tau)}{N(v)} \tag{9}$$

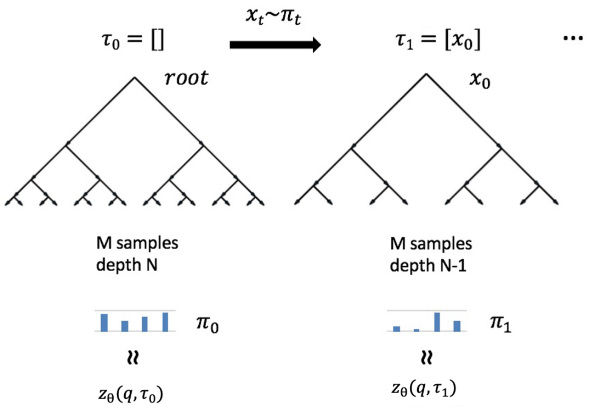


Figure 5: MCTS. At time step t, the sampling process keeps searching until it reaches depth $N - t$.

where $N(v)$ denotes the visiting time of v and $T(v)$ denotes the set of all trajectories containing v. Once the search is complete after M samples, probabilities π for the next action are estimated following Equation 10, where $N(\cdot)$ is the visit count of each move from the root state and T is a parameter controlling the temperature.

$$\pi(\cdot \mid v) = \frac{N(\cdot)^{1/T}}{\sum_{v' \in C_v} N(v')^{1/T}} \tag{10}$$

Here, C_v denotes the children of node v. Additional exploration is achieved by adding Dirichlet noise $\mathrm{Dir}(\cdot)$ to the prior probabilities as in AlphaZero (Silver et al., 2017):

$$P(\cdot|v) \sim \frac{3}{4}\pi(\cdot \mid v) + \frac{1}{4}\mathrm{Dir}(0.03) \tag{11}$$

x_t is selected in a weighted round robin manner in accordance with $P(\cdot \mid v)$. The neural network $z_\theta(q, \tau_t)$ is adjusted to minimize the KL divergence D_{KL} of the neural network estimated probabilities to the search probabilities π as:

$$\mathcal{L}(\tau_N) = \sum_{t=1}^{N} D_{\mathrm{KL}}\left[z_\theta(\cdot|q, \tau_t) \,||\, \pi(\cdot \mid v)\right]. \tag{12}$$

5 Experiments

Following standard practices in industry, we first conduct offline experiments to select reasonable models for which we subsequently perform an online evaluation. Only the best-performing model in the online tests is kept running online. We also perform an ablation study on the pipeline without label clarification. In order to verify whether the Information Gain can help to reduce the overlap between intents and the user question, we also perform experiments to evaluate the diversity and complementarity of the label recommendation method.

5.1 Experimental Settings

We first conduct offline experiments by using the 40k annotated ambiguous questions and their potential intents as explained in Section 3. The corpora are divided into training and test sets at a $9:1$ ratio. The parameters of our policy model are as follows. The sample count in MCTS is $M = 1,000$. We output $N = 6$ intents for each ambiguous question. The total number of training epochs is $E = 5$. We use a 12-layer pretrained BERT base model as the encoder for queries and the hyperparameters of the decoder are the same as for the encoder.

5.2 Evaluation Metrics

Evaluation metrics for offline experiments. The goal of our offline experiments is to evaluate the label recommendation methods, and select the most promising ones to perform online experiments. We evaluate them in terms of **Recall@N**, which reflects how many intents among all potential intents of q are retrieved among the N intents emitted by the model.

The key desideratum for label recommendation models is to cover as many potential questions as possible. It is relatively fair to compare the recall of potential intents recommended by different methods on the annotated data set. For label trajectory τ_N, the recall can be computed as

$$\mathrm{recall}(q, \tau_N) = \frac{\sum_{x \in \tau_N} |\mathcal{M}(x) \bigcap \mathcal{Q}(q)|}{|\mathcal{Q}(q)|} \tag{13}$$

where $\mathcal{Q}(q)$ is the set of potential intents for ambiguous query q, and $\mathcal{M}(x)$ is the set of all intents mapped to intent x in the intent inventory. The upper bound is calculated inversely from the results of annotated corpora:

$$\tau_N^*(q) = \operatorname*{argmax}_{\tau_N} \sum_{x \in \tau_N} |\mathcal{M}(x) \cap \mathcal{Q}(q)| \tag{14}$$

$\tau_N^*(q)$ denotes the set of N best labels covering the potential intents. Thus, the upper bound recall of q would be recall$(\tau_N^*(q))$.

Evaluation metrics for online experiments. In our subsequent online experiments, our key metrics are the rate of transferal to human agents (THA) and the click through rate (CTR). In our experiments, every time a question is classified as an ambiguous question, six labels are provided to the user, who may select one of them or just ignore the selection. Given t as the number of times we output labels, and c as the number of times the user selected one of them, we define $\mathrm{CTR} = \frac{c}{t}$. Note that the user may opt to select *none of the above options*. In this case, the pipeline equals intent retrieval without clarification. The CTR reflects how useful the recommended labels are to users.

Evaluation metrics for complementary experiments. We compare the repetition rate at the word piece level of labels generated by two methods as an experiment to evaluate the diversity. The diversity is quantified as:

$$\mathrm{div}(\tau_N) = \frac{|\mathcal{W}(\tau_N)|}{\sum\limits_{w \in \mathcal{W}(\tau_N)} C(w)}, \tag{15}$$

where $\mathcal{W}(\tau_N)$ is the set of word pieces tokenized from the labels, and $C(w)$ denotes the number of times word piece w appears among the labels. We also count the overlap rate:

$$\mathrm{overlap}(\tau_N, q) = \frac{\sum\limits_{x \in \tau_N} \sum\limits_{t \in \mathcal{T}(x) \bigcap \mathcal{T}(q)} C(t)}{\sum\limits_{x \in \tau_N} \sum\limits_{t \in \mathcal{T}(x)} C(t)} \tag{16}$$

Here, $\mathcal{T}(x_t)$, $\mathcal{T}(q)$ denote the tokens sets of x_t and q, respectively. The overlap thus essentially reflects the number of tokens of labels appearing in a query.

5.3 Baselines

Several methods for label clarification serve as baselines for the offline experiments, while our method is denoted as *RL (ours)*.

5.3.1 Label Clarification Methods

Supervised. Given a query and a set of potential intents, there are limited labels related to the potential intents set. Traverse all possible label sequences over the limited labels set and choose the one with the highest rewards as the ground truth. If there are multiple sequences corresponding to the highest reward, pick one randomly.

Greedy. Given a user question, we train a classification model on the annotated corpus of ambiguous questions and the corresponding potential intents by minimizing the loss function

$$\mathcal{L} = \sum_q D_{\mathrm{KL}}[f_\theta(\cdot|q) \parallel P(\cdot|q)] \tag{17}$$

The classification model f_θ is used to estimate the probability distribution $P(\cdot|q)$ of the potential intents. Through this greedy method, our goal is to find a set of intents for which the sum of the probabilities of intents they cover is as high as possible. The greedy rule is given by $\mathrm{Score}(x_t) = \sum_{s \in \mathcal{D}(x_t)} f_\theta(s, q)$, where $\mathcal{D}(x_t)$ is the marginal recall of intents described in Section 4. At each time step t, we select the label with the highest score as x_t. Thus, the label set is generated by the rule.

RL (no state transition). As another baseline, we explore the implication of not taking recommended labels into account. This is a BERT classification model which outputs the intent with the highest probability at time step t and masks it at the next time step.

5.3.2 Ablation Study

Top-K intents. To contrast the truncated interface with the original, full interface, we retrieve the m most similar intents in terms of semantic similarity without interacting with users. The detail of intent retrieval is described below. (Note that for a label-oriented interface, after the user selects one label, the original query is concatenated with the label phrase as a new query and relevant intents are retrieved by the same model.)

Intent retrieval. For each query, a list of potential intents can be retrieved and ranked by BM25. We re-rank the candidates by applying BERT model to estimate the semantic similarity between query and each candidate. The model is a 12-layer BERT, which takes the concatenation of two sentences as input. Considering the display limitation of the dialogue bot environment, the top three results are presented to the user.

5.4 Results

Offline experiments. The experimental results in Table 2 show that our method significantly outperforms others. The greedy method has limited recall due to its reliance on the accuracy of its classification model. We observed that it is difficult to achieve a satisfactory recall by estimating potential intent probabilities through a classification model.

Our policy model also significantly outperforms the model without state transitions, confirming the need for considering the action history. The labels recommended by simple classification models do not yield sufficient diversity, resulting in very low recall. By modeling the problem as a seq2seq one, our model learns to recommend a next label that differs from previous ones, thereby improving the recall of potential intents.

It is worth noting that the supervised method outperforms all other baselines except ours. We believe that it does not explore the training data sufficiently. In most cases, there are multiple label sequences that can get similar rewards, and the supervised method can only consider one of them as the ground truth, remaining unable to explore equally good or second-best paths, which leads to insufficient exploration of labels. Thus, the search of the supervised method is not as exhaustive as our method's. Our results are close to the theoretical upper bound, which is further corroborates the effectiveness of our method.

	labels=3	**labels=6**
Greedy	19.47%	32.01%
Supervised	45.72%	51.53%
RL (no state transition)	17.23%	29.83%
RL (ours)	**52.45%**	**57.22%**
Upper bound	60.46%	67.34%

Table 2: Offline experimental results.

Online experiments. The offline experimental results show that *RL (no state transition)* and the *Greedy* method do not perform well, leaving only *RL (ours)* for the online experiments. Here we mainly compare the performance of two rewards: recall only and reward + entropy. We compare label recommendation methods and perform an ablation study using real online user clicks. For this, we collected data over a period of two weeks in our real deployment. The experimental results, illustrated in Table 3, show that the CTR of *RL (ours)* is significantly higher than for *RL (recall)*. We believe that this gap objectively

	THA	**CTR**
Top-K intents	15.40%	-
RL (recall)	14.51%	62.61%
RL (ours)	**14.20%**	**66.36%**

Table 3: Online experimental results.

reflects the importance of entropy to improve the quality of the label set. Furthermore, *RL (ours)* also outperforms *RL (recall)* with regard to the rate of transferal to human agents (THA). The Top-k intents method directly retrieves the most relevant three questions without interacting with users. The THA gap between Top-K intents and RL based methods reflects the contribution of label clarification. The experiments show that our method has a positive effect with regard to the system's ability to clarify ambiguous questions, reducing the workload of human agents.

5.5 Complementary Evaluation

How to apply	
RL (recall)	apply, register, credit card
RL (ours)	credit card, loan, QR code
How to claim insurance?	
RL (recall)	claim, health insurance, medical insurance
RL (ours)	health insurance, medical insurance, homeowners insurance
What was the payment just now?	
RL (recall)	transaction records, inquire, transfer money
RL (ours)	billing details, transition records, inquire records

Table 4: Excerpts of outputs from different methods. For simplicity, only the first three are displayed.

By inspecting specific cases, we find that the main difference between *RL (recall)* and *RL (ours)* is the complementarity with the user's question. Taking "How to apply" in Table 4 as an example, *RL (recall)* selects "apply", "register", which exhibit semantic overlap with the question itself. Though these may lead to improved recall of potential intents, they do not enable any further clarification. The results of *RL (ours)* include products that one can apply for, helping to establish the user's underlying intent. For a recall-only approach, the labels that yield the highest rewards must be the ones with the highest semantic overlap. Hence, it is inevitable that repetitive information will be chosen, thereby making a part of the label set redundant.

To verify our conjecture, we compare the diversity and complementarity using the indicators introduced in Section 5.2. Although the two indicators are not precise metrics for diversity and semantic overlap, they help to assess the gap of the models trained with the two different reward mechanisms. As we can see from Table 5, the reinforcement learning methods significantly surpass the *Greedy* method on diversity, but the two RL methods are comparable to each other. This illustrates that recall as a reward is a major contribution to diversity. On its own, the overlap indicator is not meaningful, as it can be reduced to 0 by recommending irrelevant labels. But along with the recall, the difference in overlapping rate illustrates the effectiveness on reducing semantic repetition. Therefore, the proposed reward is superior to all other compared methods.

	Diversity	**Overlap**
Greedy	75.27%	6.39%
RL (recall)	79.92%	9.69%
RL (ours)	**80.10%**	**7.69%**

Table 5: Complementarity evaluation: The lower the overlap, the better the complementarity.

6 Conclusion

We present an end-to-end model to resolve ambiguous questions in dialogue by clarifying them using label suggestions. We cast the question clarification problem as a collection partition problem. In order to improve the quality of the interactive labels as well as reduce the semantic overlap of the labels and the user's question, we propose a novel reward based on recall of potential intents and information gain. We establish its effectiveness in a series of experiments, which suggest that this novel notion of clarification may as well be adopted for other kinds of disambiguation problems.

References

Mohammad Aliannejadi, Hamed Zamani, Fabio Crestani, and W. Bruce Croft. 2019. Asking clarifying questions in open-domain information-seeking conversations. In Benjamin Piwowarski, Max Chevalier, Éric Gaussier, Yoelle Maarek, Jian-Yun Nie, and Falk Scholer, editors, *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2019, Paris, France, July 21-25, 2019*, pages 475–484. ACM.

Marco De Boni and Suresh Manandhar. 2003. An analysis of clarification dialogue for question answering. In Marti A. Hearst and Mari Ostendorf, editors, *Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics, HLT-NAACL 2003, Edmonton, Canada, May 27 - June 1, 2003*. The Association for Computational Linguistics.

Pavel Braslavski, Denis Savenkov, Eugene Agichtein, and Alina Dubatovka. 2017. What do you mean exactly?: Analyzing clarification questions in CQA. In Ragnar Nordlie, Nils Pharo, Luanne Freund, Birger Larsen, and Dan Russel, editors, *Proceedings of the 2017 Conference on Conference Human Information Interaction and Retrieval, CHIIR 2017, Oslo, Norway, March 7-11, 2017*, pages 345–348. ACM.

Cameron Browne, Edward Jack Powley, Daniel Whitehouse, Simon M. Lucas, Peter I. Cowling, Philipp Rohlfshagen, Stephen Tavener, Diego Perez Liebana, Spyridon Samothrakis, and Simon Colton. 2012. A survey of monte carlo tree search methods. *IEEE Trans. Comput. Intellig. and AI in Games*, 4(1):1–43.

Christian Buck, Jannis Bulian, Massimiliano Ciaramita, Wojciech Gajewski, Andrea Gesmundo, Neil Houlsby, and Wei Wang. 2018. Ask the right questions: Active question reformulation with reinforcement learning. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Rémi Coulom. 2006. Efficient selectivity and backup operators in monte-carlo tree search. In *Computers and Games, 5th International Conference, CG 2006, Turin, Italy, May 29-31, 2006. Revised Papers*, pages 72–83.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Ahmed Elgohary, Denis Peskov, and Jordan Boyd-Graber. 2019. Can you unpack that? learning to rewrite questions-in-context. In *Empirical Methods in Natural Language Processing*.

Levente Kocsis and Csaba Szepesvári. 2006. Bandit based monte-carlo planning. In *Machine Learning: ECML 2006, 17th European Conference on Machine Learning, Berlin, Germany, September 18-22, 2006, Proceedings*, pages 282–293.

Mandy Korpusik and James R. Glass. 2019. Deep learning for database mapping and asking clarification questions in dialogue systems. *IEEE/ACM Trans. Audio, Speech & Language Processing*, 27(8):1321–1334.

Alexander Kotov and ChengXiang Zhai. 2010. Towards natural question guided search. In Michael Rappa, Paul Jones, Juliana Freire, and Soumen Chakrabarti, editors, *Proceedings of the 19th International Conference on World Wide Web, WWW 2010, Raleigh, North Carolina, USA, April 26-30, 2010*, pages 541–550. ACM.

Jiwei Li, Alexander H. Miller, Sumit Chopra, Marc'Aurelio Ranzato, and Jason Weston. 2017. Learning through dialogue interactions by asking questions. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Ye Liu, Chenwei Zhang, Xiaohui Yan, Yi Chang, and Philip S. Yu. 2019. Generative question refinement with deep reinforcement learning in retrieval-based QA system. In Wenwu Zhu, Dacheng Tao, Xueqi Cheng, Peng Cui, Elke A. Rundensteiner, David Carmel, Qi He, and Jeffrey Xu Yu, editors, *Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM 2019, Beijing, China, November 3-7, 2019*, pages 1643–1652. ACM.

Hao Ma, Michael R. Lyu, and Irwin King. 2010. Diversifying query suggestion results. In Maria Fox and David Poole, editors, *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2010, Atlanta, Georgia, USA, July 11-15, 2010*. AAAI Press.

Rodrigo Nogueira and Kyunghyun Cho. 2017. Task-oriented query reformulation with reinforcement learning. In Martha Palmer, Rebecca Hwa, and Sebastian Riedel, editors, *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 574–583. Association for Computational Linguistics.

Silvia Quarteroni and Suresh Manandhar. 2009. Designing an interactive open-domain question answering system. *Natural Language Engineering*, 15(1):73–95.

J. Ross Quinlan. 1986. Induction of decision trees. *Machine Learning*, 1(1):81–106.

Filip Radlinski and Nick Craswell. 2017. A theoretical framework for conversational search. In Ragnar Nordlie, Nils Pharo, Luanne Freund, Birger Larsen, and Dan Russel, editors, *Proceedings of the 2017 Conference on Conference Human Information Interaction and Retrieval, CHIIR 2017, Oslo, Norway, March 7-11, 2017*, pages 117–126. ACM.

Sudha Rao and Hal Daumé. 2018. Learning to ask good questions: Ranking clarification questions using neural expected value of perfect information. In Iryna Gurevych and Yusuke Miyao, editors, *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 2737–2746. Association for Computational Linguistics.

Sudha Rao and Hal Daumé. 2019. Answer-based adversarial training for generating clarification questions. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 143–155. Association for Computational Linguistics.

Eldar Sadikov, Jayant Madhavan, Lu Wang, and Alon Y. Halevy. 2010. Clustering query refinements by user intent. In Michael Rappa, Paul Jones, Juliana Freire, and Soumen Chakrabarti, editors, *Proceedings of the 19th International Conference on World Wide Web, WWW 2010, Raleigh, North Carolina, USA, April 26-30, 2010*, pages 841–850. ACM.

Hassan Sajjad, Patrick Pantel, and Michael Gamon. 2012. Underspecified query refinement via natural language question generation. In Martin Kay and Christian Boitet, editors, *COLING 2012, 24th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, 8-15 December 2012, Mumbai, India*, pages 2341–2356. Indian Institute of Technology Bombay.

David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, and Adrian Bolton. 2017. Mastering the game of go without human knowledge. *Nature*, 550(7676):354–359.

Yang Tang, Fan Bu, Zhicheng Zheng, and Xiaoyan Zhu. 2011. Towards interactive qa: suggesting refinement for questions. *Belkin et al.[2]*, pages 13–14.

Sebastian Varges, Silvia Quarteroni, Giuseppe Riccardi, and Alexei V. Ivanov. 2010. Investigating clarification strategies in a hybrid POMDP dialog manager. In Raquel Fernández, Yasuhiro Katagiri, Kazunori Komatani, Oliver Lemon, and Mikio Nakano, editors, *Proceedings of the SIGDIAL 2010 Conference, The 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue, 24-15 September 2010, Tokyo, Japan*, pages 213–216. The Association for Computer Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 5998–6008.

Yansen Wang, Chenyi Liu, Minlie Huang, and Liqiang Nie. 2018. Learning to ask questions in open-domain conversational systems with typed decoders. In Iryna Gurevych and Yusuke Miyao, editors, *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 2193–2203. Association for Computational Linguistics.

Yongfeng Zhang, Xu Chen, Qingyao Ai, Liu Yang, and W. Bruce Croft. 2018. Towards conversational search and recommendation: System ask, user respond. In Alfredo Cuzzocrea, James Allan, Norman W. Paton, Divesh Srivastava, Rakesh Agrawal, Andrei Z. Broder, Mohammed J. Zaki, K. Selçuk Candan, Alexandros Labrinidis, Assaf Schuster, and Haixun Wang, editors, *Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM 2018, Torino, Italy, October 22-26, 2018*, pages 177–186. ACM.

Xinbo Zhang, Lei Zou, and Sen Hu. 2019. An interactive mechanism to improve question answering systems via feedback. In Wenwu Zhu, Dacheng Tao, Xueqi Cheng, Peng Cui, Elke A. Rundensteiner, David Carmel, Qi He, and Jeffrey Xu Yu, editors, *Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM 2019, Beijing, China, November 3-7, 2019*, pages 1381–1390. ACM.

Zhicheng Zheng, Xiance Si, Edward Y. Chang, and Xiaoyan Zhu. 2011. K2Q: generating natural language questions from keywords with user refinements. In *Fifth International Joint Conference on Natural Language Processing, IJCNLP 2011, Chiang Mai, Thailand, November 8-13, 2011*, pages 947–955. The Association for Computer Linguistics.