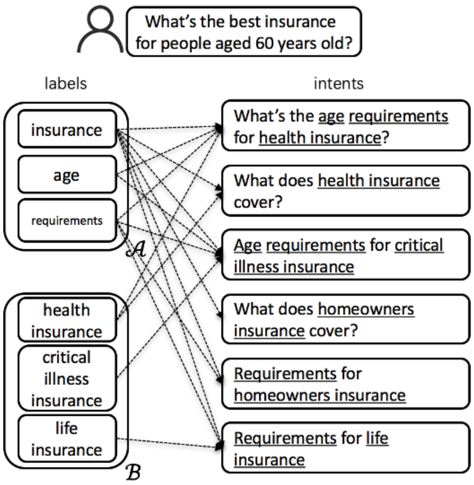
What's the best insurance for people aged 60 years old?
labels
intents
insurance
age
requirements
$\mathcal{A}$
health insurance
critical illness insurance
life insurance
$\mathcal{B}$
What's the age requirements for health insurance?
What does health insurance cover?
Age requirements for critical illness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

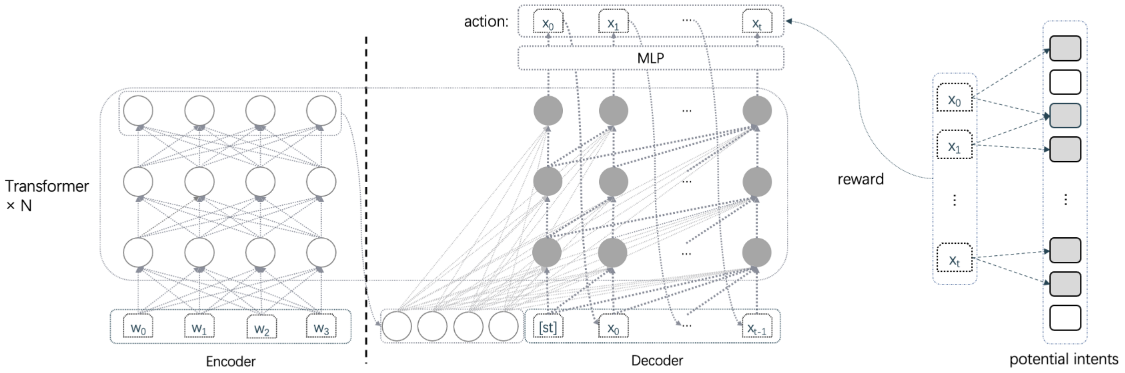

Figure 4: Label recommendation policy model architecture.

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 $\tau_N = \{x_1, x_2, \dots, x_N\}$. 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 $\mathbf{I}(q) = (\mathbf{I}_1, \mathbf{I}_2, \dots, \mathbf{I}_{|\mathcal{K}|})$ 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.

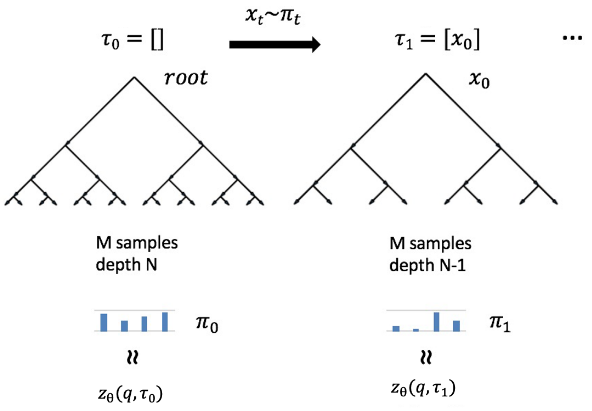

Figure 5: MCTS. At time step t, the sampling process keeps searching until it reaches depth $N - t$.

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

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

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.

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

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