Interactive Classification by Asking Informative Questions

Lili Yu[1], **Howard Chen**[1], **Sida I. Wang**[1,2], **Tao Lei**[1] and **Yoav Artzi**[1,3]
[1]ASAPP Inc., New York, USA
[2]Princeton University, New Jersey, USA
[3]Cornell University, New York, USA
{liliyu, hchen, tao}@asapp.com
sidaw@cs.princeton.edu yoav@cs.cornell.edu

Abstract

We study the potential for interaction in natural language classification. We add a limited form of interaction for intent classification, where users provide an initial query using natural language, and the system asks for additional information using binary or multi-choice questions. At each turn, our system decides between asking the most informative question or making the final classification prediction.The simplicity of the model allows for bootstrapping of the system without interaction data, instead relying on simple crowd-sourcing tasks. We evaluate our approach on two domains, showing the benefit of interaction and the advantage of learning to balance between asking additional questions and making the final prediction.

1 Introduction

Responding to natural language queries through simple, single-step classification has been studied extensively in many applications, including user intent prediction (Chen et al., 2019; Qu et al., 2019), and information retrieval (Kang and Kim, 2003; Rose and Levinson, 2004). Typical methods rely on a single user input to produce an output, and do not interact with the user to reduce ambiguity and improve the final prediction. For example, users may under-specify a request due to incomplete understanding of the domain; or the system may fail to correctly interpret the nuances of the input query. In both cases, a low quality decision could be mitigated by further interaction with the user.

In this paper we take a low-overhead approach to add limited interaction to intent classification. Our goal is two-fold: (a) study the effect of interaction on the system performance, and (b) avoid the cost and complexities of interactive data collection. We build an interactive system that poses a sequence of binary and multiple choice questions following the initial user natural language query. Figure 1 illustrates such interactions in two domains, showcasing the opportunity for clarification while avoiding much of the complexity involved in unrestricted natural language interactions. We design our approach not to rely on user interaction during learning, which requires users to handle low quality systems or costly Wizard of Oz experiments.

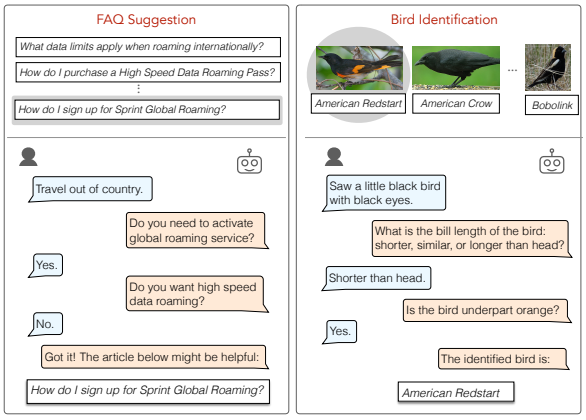

Figure 1: Two examples of interactive classification systems: providing a trouble-shooting FAQ suggestion (left) and helping identifying bird species from a descriptive text query (right). The top parts show example classification labels: FAQ documents or bird species.[1]The ground truth label of each interaction example is shaded. The lower parts show user interactions with the systems. The user starts with an initial natural language query. At each step, the system asks a clarification question. The interaction ends when the system returns an output label.

We adopt a Bayesian decomposition of the posterior distributions over intent labels and user responses through the interaction process. We use the posteriors to compute question expected information gain, which allows us to efficiently select the next question at each interaction turn. We bal-

[1]The images are for illustration only. Our approach does not use images.

Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 2664–2680

ance between the potential increase in accuracy and the cost of asking additional questions with a learned policy controller that decides whether to ask additional questions or return the final prediction. We estimate each distribution in our posterior decomposition independently by crowdsourcing initial queries and keywords annotation. We use non-interactive annotation tasks that do not require Wizard-of-Oz style dialog annotations (Kelley, 1984; Wen et al., 2017). During training, we train a shared text encoder to compare natural language queries, clarification questions, user answers and classification targets in the same embedding space. This enables us to bootstrap to unseen clarification targets and clarification questions, further alleviating the need of expensive annotation.

We evaluate our method on two public tasks: FAQ suggestion (Shah et al., 2018) and bird identification using the text and attribute annotations of the Caltech-UCSD Birds dataset (Wah et al., 2011). The first task represents a virtual assistant application in a trouble-shooting domain, while the second task provides well-defined multiple-choice question annotations and naturally noisy language inputs. We evaluate with both a simulator and human users. Our experiments show that adding user interaction significantly increases the classification accuracy. Given at most five turns of interaction, our approach improves the accuracy of a no-interaction baseline by over 100% on both tasks for simulated evaluation and over 90% for human evaluation. Even a single clarification question provides significant accuracy improvements, 40% for FAQ suggestion and 65% for bird identification in our simulated analysis. Our code and data are available at https://github.com/asappresearch/interactive-classification.

2 Technical Overview

Our goal is to classify a natural language query to a label through an interaction.

Notation We treat the classification label y, interaction question q and the user response r as random variables. We denote an assignment of a random variable using subscripts, such as $y = y_i$ and $q = q_j$. We use superscripts for the observed value of the random variable at a given time step, for example, q^t is a question asked at time step t. When clear from the context, we write y_i instead of $y = y_i$. For example, $p(r \mid q_j, y_i)$ denotes the conditional distribution of r given $y = y_i$ and $q = q_j$, and $p(r_k \mid q_j, y_i)$ further specifies the corresponding probability when $r = r_k$.

An interaction starts with the user providing an initial user query x. At each turn t, the system selects a question q^t, to which the user responds with r^t, or returns a label y to conclude the interaction. We consider two types of questions: binary and multiple choice questions. The predefined set of possible answers for a question q^t is $\mathcal{R}(q^t)$, where $\mathcal{R}(q^t) = \{\text{yes}, \text{no}\}$ for binary questions, or a predefined set of question-specific values for multiple choice questions. We denote an interaction up to time t as $X^t = (x, \langle (q^1, r^1), \ldots, (q^t, r^t \rangle)$, and the set of possible class labels as $\mathcal{Y} = \{y_1, \ldots, y_N\}$. Figure 1 shows example interactions in our two evaluation domains.

Model We model the interactive process using a parameterized distribution over class labels that is conditioned on the observed interaction (Section 4.1), a question selection criterion (Section 4.2), and a parameterized policy controller (Section 4.5). At each time step t, we compute the belief of each $y_i \in \mathcal{Y}$ conditioned on X^{t-1}. The trained policy controller decides between two actions: to return the current best possible label or to obtain additional information by asking a question. The model selects the question with the maximal information gain. Given a user response, the model updates the belief over the classification labels.

Learning We use crowdsourced data to bootstrap model learning. The crowdsourcing data collection includes two non-interactive tasks. First, we obtain a set of user initial queries $\mathcal{X}_i$ for each label y_i. For example, for an FAQ, 'How do I sign up for Spring Global Roaming', an annotated potential initial query is 'Travel out of country'. Second, we ask annotators to assign text tags to each y_i, and heuristically convert these tags into a set of question-answer pairs $\mathcal{A}_i = \{(q_m, r_m)\}_{m=1}^{M_i}$, where q_m denotes a templated question and r_m denotes the answer. For example, the question 'What is your phone operating system?' can pair with one of the following answers: 'IOS', 'Android operating system', 'Windows operating system' or 'Not applicable'. We denote this dataset as $\{(y_i, \mathcal{X}_i, \mathcal{A}_i)\}_{i=1}^{N}$. We describe the data collection process in Section 5. We use this data to train our text embedding model (Section 4.3), to create a user simulator (Section 4.4), and to train the policy controller (Section 4.5).

Evaluation We report classification the model accuracy, and study the trade-off between accuracy and the number of turns that the system takes. We evaluate with both a user simulator and real human users. When performing human evaluation, we additionally collect qualitative ratings.

3 Related Work

Human feedback has been leveraged to train natural language processing models, including for dialogue (Li et al., 2016), semantic parsing (Artzi and Zettlemoyer, 2011; Wang et al., 2016; Iyer et al., 2017) and text classification (Hancock et al., 2018). These methods collect user feedback after the model-predicting stage and treat user feedback as additional offline training data to improve the model. In contrast, our model leverages user interaction to increase prediction performance. Human feedback has been incorporated in reinforcement learning as well, for example to learn a reward function from language as reflecting human preferences (Christiano et al., 2017).

Language-based interaction has been studied in the context of visual question answering (de Vries et al., 2017; Lee et al., 2018; Chattopadhyay et al., 2017; Das et al., 2017; Lee et al., 2019; Shukla et al., 2019), SQL generation (Gur et al., 2018; Yao et al., 2019), information retrieval (Chung et al., 2018; Aliannejadi et al., 2019) and multi-turn text-based question answering (Rao and Daumé III, 2018; Reddy et al., 2019; Choi et al., 2018). Most methods require learning from recorded dialogues (Wu et al., 2018; Hu et al., 2018; Lee et al., 2018; Rao and Daumé III, 2018) or conducting Wizard-of-Oz dialog annotations (Kelley, 1984; Wen et al., 2017). Instead, we limit the interaction to multiple-choice and binary questions. This simplification allows us to reduce the complexity of data annotation while still achieving effective interaction. Our task can be viewed as an instance of the popular 20-question game (20Q), which has been applied to a celebrities knowledge base (Chen et al., 2018; Hu et al., 2018). Our approach differs in using natural language descriptions of classification targets, questions and answers to compute our distributions, instead of treating them as categorical or structural data.

Our question selection method is related to several existing methods. Kovashka and Grauman (2013) refine image search by asking to compare visual qualities against selected reference images, and Lee et al. (2018) perform object identification in an image by posing binary questions about the object or its location. Both methods, as well as ours use an entropy reduction criterion to select the best next question. We use a Bayesian decomposition of the joint distribution, which can be easily extended to other model-driven selection methods. Rao and Daumé III (2018) propose a learning-to-ask approach by modeling the expected utility of asking question. Our selection method can be considered as a special case when entropy is used as the utility. In contrast to Rao and Daumé III (2018), we model the entire interaction history instead of a single turn of follow-up questioning. Our model is trained using crowdsourced annotations, while Rao and Daumé III (2018) uses real user-user interaction data. Alternatively to asking questions, Ferecatu and Geman (2007) and Guo et al. (2018) present to the user the most likely image in an image retrieval scenario. The user compares it with the ground-truth image and provides feedback using relevance score or natural language describing the discrepancy between them.

4 Method

We maintain a probability distribution $p(y|X^t)$ over the set of labels $\mathcal{Y}$. At each interaction step, we first update this belief, decide if to ask a question or return the classification output using a policy controller and, if needed, select a question to ask using information gain.

4.1 Belief Probability Decomposition

We decompose the conditional probability $p(y = y_i|X^t)$ using Bayes rule:

$$\begin{aligned} p(y_i|X^t) &= p(y_i|X^{t-1}, q^t, r^t) \\ &\propto p(r^t, q^t, y_i|X^{t-1}) \\ &= p(q^t|y_i, X^{t-1})\, p(y_i|X^{t-1}) \\ &\quad p(r^t|q^t, y_i, X^{t-1})\ . \end{aligned}$$

We make two simplifying assumptions as modeling choices. First, the user response depends only on the question q^t and the underlying target label y_i, and is independent of past interactions. While this independence assumption is unlikely to reflect the course of interactions, it allows to simplify $p(r^t|q^t, y_i, X^{t-1})$ to $p(r^t|q^t, y_i)$. Second, the selection of the next question q^t is deterministic given the interaction history X^{t-1}. Therefore,

$p(q = q^t | y_i, X^{t-1}) = 1$, or zero for $q \neq q^t$. Section 4.2 describes this process. We rewrite the decomposition as:

$$\begin{aligned} p(y_i|X^t) &\propto p(r^t|q^t, y_i) \cdot 1 \cdot p(y_i|X^{t-1}) \\ &= p(y_i|x) \prod_{\tau=1}^{t} p(r^\tau|q^\tau, y_i) \ . \end{aligned} \quad (1)$$

Predicting the classification label given the observed interaction X^t is reduced to modeling $p(y_i|x)$ and $p(r_k|q_j, y_i)$, the label y_i probability given the initial query x only and the probability of user response r_k conditioned on the chosen question q_j and class label y_i. This factorization enables leveraging separate annotations to learn the two components directly, alleviating the need for collecting costly recordings of user interactions.

4.2 Information Gain Question Selection

The system selects the question q^t to ask at turn t to maximize the efficiency of the interaction. We use a maximum information gain criterion. Given X^{t-1}, we compute the information gain on classification label y as the decrease on entropy by observing possible answers to question q:

$$IG(y\,; q|X^{t-1}) = H(y|X^{t-1}) - H(y|X^{t-1}, q) \ ,$$

where $H(\cdot|\cdot)$ denotes the conditional entropy. Intuitively, the information gain measures the amount of information obtained about the variable y by observing the value of another variable q. Because the first entropy term $H(y|X^{t-1})$ is a constant regardless of the choice of q, the selection of q^t is equivalent to $q^t = \arg\min_{q_j} H(y|X^{t-1}, q_j)$, where

$$\begin{aligned} H(y|X^{t-1}, q_j) &= \sum_{r_k \in \mathcal{R}(q_j)} p(r_k|X^{t-1}, q_j) \\ &\qquad H(y|X^{t-1}, q_j, r_k) \\ H(y|X^{t-1}, q_j, r_k) &= \sum_{y_i \in \mathcal{Y}} p(y_i|X^{t-1}, q_j, r_k) \\ &\qquad \log p(y_i|X^{t-1}, q_j, r_k) \\ p(r_k|X^{t-1}, q_j) &= \sum_{y_i \in \mathcal{Y}} p(r_k, y_i|X^{t-1}, q_j) \\ &= \sum_{y_i \in \mathcal{Y}} p(r_k|q_j, y_i) \\ &\qquad p(y_i|X^{t-1}) \ . \end{aligned}$$

We use the independence assumption (Section 4.1) to calculate $p(r_k|X^{t-1}, q_j)$. Both $p(r_k|X^{t-1}, q_j)$ and $p(y_i|X^{t-1}, q_j, r_k)$ can be iteratively updated using $p(y_i|x)$ and $p(r_k|q_j, y_i)$ as the interaction progresses (Equation 1) to efficiently compute the information gain.

4.3 Modeling the Distributions

We model $p(y_i|x)$ and $p(r_k|q_j, y_i)$ by encoding the natural language descriptions of questions, answers and classification labels. In our domains, the text representation of a label is the FAQ document or the bird name. We do not simply treat the labels, questions and answers as categorical variables. Instead, we leverage their natural language content to estimate their correlation This reduces the need for heavy annotation and improves our model in low-resource scenarios. We use a shared neural encoder $\mathbf{enc}(\cdot)$ parameterized by ψ to encode all texts. Both probability distributions are computed using the dot-product score: $S(u, v) = \mathbf{enc}(u)^\top \mathbf{enc}(v)$, where u and v are two pieces of text. The probability of predicting the label y_i given an initial query x is:

$$p(y_i|x) = \frac{\exp(S(y_i, x))}{\sum_{y_j \in \mathcal{Y}} \exp(S(y_j, x))} \ .$$

The probability of an answer r_k given a question q_j and label y_i is a linear combination of the observed empirical distribution $\hat{p}(r_k|q_j, y_i)$ and a parameterized estimation $\tilde{p}(r_k|q_j, y_i)$:

$$p(r_k|q_j, y_i) = \lambda \hat{p}(r_k|q_j, y_i) + (1-\lambda)\tilde{p}(r_k|q_j, y_i) \ ,$$

where $\lambda \in [0, 1]$ is a hyper-parameter. We use the question-answer annotations $\mathcal{A}_i$ for each label y_i to estimate $\hat{p}(r_k|q_j, y_i)$ using empirical counts. For example, in the FAQ suggestion task, we collect multiple user responses for each question and class label, and average across annotators to estimate $\hat{p}$ (Section 5). The second term $\tilde{p}(r_k|q_j, y_i)$ is computed using the text encoder:

$$\begin{aligned} &\tilde{p}(r_k|q_j, y_i) \\ &\quad = \frac{\exp(w \cdot S(q_j \# r_k, y_i) + b)}{\sum_{r_l \in \mathcal{R}(q_j)} \exp(w \cdot S(q_j \# r_l, y_i) + b)} \ , \end{aligned}$$

where $w, b \in \mathbb{R}$ are scalar parameters and $q_j \# r_k$ is a concatenation of the question q_j and the answer r_k.[2] Because we do not collect complete annotations to cover every label-question pair, $\tilde{p}$ provides

[2]For example, for a templated question 'What is your phone operating system?' and an answer 'IOS', q_m = 'phone operating system' and r_m = 'IOS', therefore, $q_m \# r_m$ = 'phone operating system IOS'.

a smoothing of the partially observed counts using the learned encoding $S(\cdot)$.

We estimate the parameters ψ of $\mathbf{enc}(\cdot)$ by pre-training using a dataset $\{(y_i, \mathcal{X}_i, \mathcal{A}_i)\}_{i=1}^{N}$, where y_i is a label, $\mathcal{X}_i$ is a set of initial queries and $\mathcal{A}_i$ is a set of question-answer pairs. We create from this data a set of text pairs (u, v) to train the scoring function $S(\cdot)$. For each label y_i, we create pairs (x, y_i) for each initial query $x \in \mathcal{X}_i$. We also create $(q_m \# r_m, y_i)$ for each question-answer pair $(q_m, r_m) \in \mathcal{A}_i$. We minimize the cross-entropy loss using gradient descent:

$$\mathcal{L}(\psi) = -S(u, v) + \log \sum_{v'} \exp(S(u, v')) \ .$$

The second term requires summation over all v', which are all the labels in $\mathcal{Y}$. We approximate this sum using negative sampling that replaces the full set $\mathcal{Y}$ with a sampled subset in each training batch. The parameters ψ, w and b are fine-tuned using reinforcement learning during training of the policy controller (Section 4.5).

4.4 User Simulator

We use a held-out dataset to build a simple simulator. We use the simulator to train the policy controller (Section 4.5) and for performance analysis, in addition to human evaluation. The user simulator provides initial queries to the system and responds to the system initiated clarification questions. The dataset includes N examples $\{(y_i, \mathcal{X}'_i, \mathcal{A}'_i)\}_{i=1}^{N}$, where y_i is a goal, $\mathcal{X}'_i$ is a set of initial queries and $\mathcal{A}'_i = \{(q_m, r_m)\}_{m=1}^{M'_i}$ is a set of question-answer pairs. While this data is identical in form to our training data, we keep it separated from the data used to estimate $S(\cdot)$, $p(y_i|x)$ and $p(r_k|q_j, y_i)$ (Section 4.3). We estimate the simulator question response distribution $p'(r_k|q_j, y_i)$ using smoothed empirical counts from the data.

At the beginning of a simulated interaction, we sample a target label $\hat{y}$, and sample a query x from the associated query set $\mathcal{X}'$ to start the interaction. Given a system clarification question q^t at turn t, the simulator responds with an answer $r^t \in \mathcal{R}(q^t)$ by sampling from $p'(r|q^t, \hat{y})$. Sampling provides natural noise to the interaction, and our model has no knowledge of p'. The interaction ends when the system returns a label, which we can then evaluate, for example to compute a reward in Section 4.5. This setup is flexible in that the user simulator can be easily replaced or extended by a real human, and the system can be further trained with a human-in-the-loop setup.

Algorithm 1: Training procedure

```
Estimate p(y|x) and p(r|q,y) with w and b
  randomly initialized
Estimate p'(r|q,y) for the user simulator
for episode = 1 ... M do
    Sample (x, ŷ) from dataset
    for t = 1 ... T do
        Compute p(y|X^{t-1}) (Equation 1)
        action = f(p(y|X^{t-1}), t-1; θ)
        if action is STOP then
            break
        else if action is ASK then
            q^t =
              arg max_{q_j ∈ Q} IG(y; q_j|X^{t-1})
            r^t ~ p'(r|q^t, ŷ)
    end
    y* = arg max_{y_i} p(y_i|X^{t-1})
    Compute the return (i.e., total reward) for every
      step t using y* and ŷ
    Update w, b, θ using policy gradient
end
```

4.5 Policy Controller

The policy controller decides at each turn t to either select another question to query the user or to conclude the interaction. This provides a trade-off between exploration by asking questions and exploitation by returning the most probable classification label. The policy controller $f(\cdot, \cdot; \theta)$ is a feed-forward network parameterized by θ that takes the top-k probability values and current turn t as input. It generates one of two actions: *STOP* or *ASK*. When selecting *ASK*, a question is selected to maximize the information gain. For *STOP*, the label y_i with highest probability is returned using $\arg\max_{y_i \in \mathcal{Y}} p(y_i|X^{t-1})$ and the interaction ends.

4.6 Training Procedure

Algorithm 1 describes the complete training process. First, we estimate $p(y|x)$ and $p(r|q, y)$. We use randomly initialized and fixed w and b parameters. We also estimate $p'(r|q, y)$ for the user simulator (Section 4.4). We then learn the policy controller using the user simulator with a policy gradient method. We use the REINFORCE algorithm (Williams, 1992). The reward function provides a positive reward for predicting the correct target at the end of the interaction, a negative reward for predicting the wrong target, and a small negative reward for every question asked. We learn the policy controller $f(\cdot, \cdot; \theta)$, and estimate w and b in $p(r_k|q_j, y_i)$ by back-propagating through the

policy gradient. We keep the $\mathbf{enc}(\cdot)$ parameters fixed during policy gradient.

5 Data Collection

We design a crowdsourcing process to collect data for the FAQ task using Amazon Mechanical Turk.[3] For the Birds domain, we re-purpose an existing dataset. We collect initial queries and tags for each FAQ document. Appendix A.1 describes the worker training process.

Initial Query Collection We ask workers to consider the scenario of searching for an FAQ document using an interactive system. Given a target FAQ, we ask for an initial query that they would provide to such a system. The set of initial queries that is collected for each document y_i is $\mathcal{X}_i$. We encourage workers to provide incomplete information and avoid writing a simple paraphrase of the FAQ. This process provides realistic and diverse utterances because users have limited knowledge of the system and the domain.

Tag Collection We collect natural language tag annotations for the FAQ documents. First, we use domain experts to define the set of possible free-form tags. The tags are not restricted to a predefined ontology and can be a phrase or a single word describing the topic of the document. We remove duplicate tags to finalize the set. Experts combine some binary tags to categorical tags. For example, tags 'IOS', 'Android operating system' and 'Windows operating system' are combined to the categorical tag 'phone operating system'. We use a small set of deterministic, heuristically-designed templates to convert tags into questions. For example, the tag 'international roaming' is converted into a binary question 'Is it about international roaming?'; the categorical tag 'phone operating system' is converted into a multi-choice question 'What is your phone operating system?'. Finally, we use non-experts to collect user responses to the questions by associating tags with FAQ targets. For binary questions, we ask workers to associate their tags to the FAQ target if they would respond 'yes' to the question. We show the workers a list of ten tags for a given target as well as a 'none of the above' option. Annotating all possible target-tag combinations is still expensive and most pairings are negative. We rank the tags based on the relevance against the target using $S(\cdot)$ trained only on the initial queries and show only the current top-50 to the workers. Later, we re-train $S(\cdot)$ on the complete data. For multi-choice questions, we show the workers a list of possible answers to a tag-generated question for a given FAQ. The workers need to choose one answer that they think best applies. They also have the option of choosing 'not applicable'. The workers do not engage in a multi-round interactive process. This allows for cheap and scalable collection.

[3] https://www.mturk.com/

6 Experimental Setup

Task I: FAQ Suggestion We use the FAQ dataset from Shah et al. (2018). The dataset contains 517 troubleshooting documents from Sprint's technical website. We collect 3,831 initial queries and 118,640 tag annotations using the setup described in Section 5. We split the data into 310/103/104 documents as training, development, and test sets. Only the queries and tag annotations of the 310 training documents are used for pre-training and learning the policy controller, leaving the queries and tag annotations in the development and test splits for evaluation only.

Task II: Bird Identification We use the Caltech-UCSD Birds dataset (CUB-200; Wah et al., 2011). The dataset contains 11,788 bird images for 200 different bird species. Each bird image is annotated with a subset of 27 visual attributes and 312 attribute values pertaining to the color or shape of a particular part of the bird. We create categorical questions from attributes with less five possible values, providing eight categorical questions in total. The remaining 279 attributes are converted to binary questions. Each image is annotated with 10 image captions describing the bird in the image (Reed et al., 2016). We use the image captions as initial user queries and bird species as labels. Since each caption contains only partial information about the bird species, the data is naturally noisy and provides challenging user interactions. We do not use the images from the dataset for model training. The images are only provided for grounding during human evaluation.

Baselines We compare with four methods:

- No Interaction: the classification label is predicted using only the initial query. We consider four implementations: (1) BM25: a common keyword-based scoring model for retrieval methods (Robertson and Zaragoza,

2009); (2) RoBERTa$_{\text{BASE}}$: we use a fine-tuned RoBERTa$_{\text{BASE}}$ model (Liu et al., 2019) as text encoder; (3) RNN: we use a recurrent neural network (RNN) with simple recurrent unit recurrence (SRU; Lei et al., 2018) as text encoder, together with a fastText word embedding layer (Bojanowski et al., 2017); and (4) RNN + self-attn: the same RNN neural model with a multi-head self-attention layer (Lin et al., 2017; Vaswani et al., 2017).

- Random Interaction: at each turn, the system randomly selects a question to present the user. After T turns, the classification label is chosen according to the belief $p(y|X^T)$.
- No Initial Query Interaction: the system selects questions without conditioning on the initial user query using maximum information criterion. This is equivalent to using a static decision tree to pick the question, always asking the same first question (Utgoff, 1989; Ling et al., 2004).
- Variants of Our Approach: we consider several variants of our full model. First, we replace the policy controller with two termination strategies: (1) end the interaction when $\max p(y|X^t)$ passes a threshold, or (2) end the interaction after a fixed number of turns. Second, we disable the parameterized estimator $\tilde{p}(r_k|q_j, y_i)$ by setting $\lambda = 1$.

Evaluation We use human evaluation, and further analyze performance using our simulator. For human evaluation, users interact with our systems and baseline models using a web-based interactive interface. Each interaction starts with a user scenario:[4] a bird image or a device-troubleshooting scenario described in text. The user types an initial query and answers follow-up questions selected by the system. Once the system returns its prediction, we measure its accuracy, and the user is asked to rate the whole interaction according to rationality and naturalness.[5] The user does not know the correct target label. We use a five-points Likert score for the followup questions. For FAQ Suggestion, we consider two evaluation setups: (1) assuming the model has access to tags in the development and test set for interaction, and (2) using only tags in the training set annotation. The former is equivalent to adding tags for new documents not seen during training time. The latter zero-shot evaluation setup allows us to investigate the model's performance on unseen targets with no additional tags associated with them. Appendix A.4 provides further details of the human evaluation setup. We do further analysis with the user simulator . We evaluate classification performance using Accuracy@k, which is the percentage of time the correct target appears among the top-k predictions of the model.

Implementation Details We use the same encoder to encode initial queries, question-answer pairs and FAQ documents in the FAQ suggestion task. In the bird identification task, where the structure of bird names differs from the other texts, we use one encoder for user initial queries and question-answer pairs and a second encoder for bird names. The policy controller receives a reward of 20 for returning the correct target label, a negative reward of -10 for the wrong target, and a turn penalty of -0.5 for each question asked. For our simulated analysis, we report the averaged results as well as the standard derivation from three independent runs for each model variant and baseline. Appendix A.2 provides more implementation and training details.

7 Results

Our simulated analysis shows that the SRU RNN text encoder performs better or similar to the other encoders. This encoder is also the most lightweight. Therefore, we use it for the majority of our experiments.

Human Evaluation Figure 2 and Table 1 show the human evaluation results of our full model and three baselines: our approach with a fixed number of turns (four for FAQ and five for Bird), our approach without access to the initial query (No Init. Query) and our approach without interaction (No Int. (RNN)). Naturalness and rationality measure the quality of the interaction, so we show the results of the user survey in Figure 2 only for interactive systems. Because we do not ask users to fill the end-of-interaction survey for the no interaction baseline, we simply compute its numbers following the first query when evaluating our full approach. Our approach balances between accuracy and the user-centric measures, including naturalness and rationality, achieving stronger performance across

[4]Each scenario is related to a single groundtruth label and serves to ground user interactions.

[5]We also surveyed users for perceived correctness, but observed it is interpreted identically to rationality. Therefore, we omit this measure.

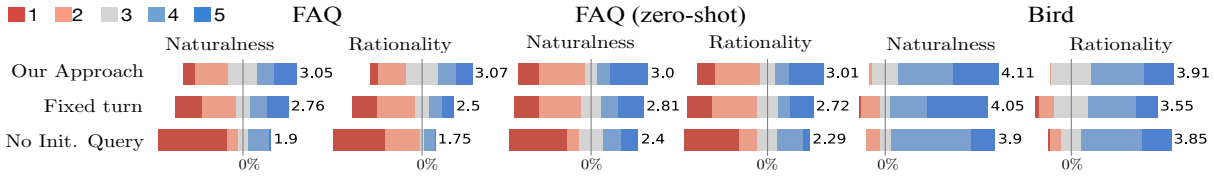

Figure 2: Human evaluation Gantt charts showing user ratings. We show the mean rating for each measure and system on the right of each bar.

	FAQ	FAQ (zero-shot)	Bird
Our Approach	**57%**	**52%**	**45%**
Our Approach w/fixed turn	53%	47%	37%
No Init. Query	43%	41%	28%
No Int. (RNN)	30%	26%	20%

Table 1: Human evaluation classification accuracy.

the board. All three models improve the classification performance with the addition of interaction. Qualitatively, the users rate our full approach better than the two other interaction variants. This demonstrates that our model handles effectively real user interaction despite being trained with only non-interactive data. We include additional details in Appendix A.4.

Analysis with Simulated Interactions Table 2 shows performance using the the user simulator. We use these results to evaluate different choices beyond what is possible with human studies. We observe interaction is critical; removing the ability to interact decreases performance significantly. The Random Interaction and the No Initial Query Interaction baselines both barely improve the performance over the No Interaction RNN baseline, illustrating the importance of guiding the interaction and considering the initial query. Our full model achieves an Accuracy@1 of 79% for FAQ Suggestion and 49% for Bird Identification using less than five turns, outperforming the No Interaction RNN baseline by 41% and 26%. When having no access to questions and answers in the development and test set during evaluation, the full model performance drops only slightly to 75%, highlighting the model's ability to generalize to unseen tags. The two baselines with alternative termination strategies underperform the full model, indicating the effectiveness of the policy controller. The relatively low performance of the $\lambda = 1$ variant, which effectively has fewer probability components leveraging natural language than our full model, and No Initial Query Interaction confirm the importance of the learned natural language embedding encoder. Appendix A.3 includes further details on how different text encoders impact performance.

Figure 3 shows the trade-off between classification accuracy and the number of turns. Each point on the plots is computed by varying the reward turn penalty for our model, the prediction threshold and the predefined number of turns T. Our model with the policy controller or the threshold strategy does not explicitly bound the number of turns, so we report the average number of turns across multiple runs for these two models. We achieve a relative accuracy boost of 40% for FAQ and 65% for Birds over no-interaction baselines with only one clarification question. This highlights the value of leveraging human feedback to improve model accuracy in classification tasks.

Figure 4 shows the learning curves of our model with the policy controller trained with different turn penalties $r_a \in \{-0.5, -1, -3\}$. We observe the models explore during the first 1,000 training episodes in the middle and the right plots. The models achieve relatively stable accuracy after the early exploration stage. The three runs end up using different numbers of expected turns because of the different r_a values.

8 Conclusion

We propose an approach for interactive classification, where the system can inquire missing information through a sequence of simple binary or multi-choice questions when users provide underspecified natural language queries. Our expert-guided, incremental design of questions and answers enables easy extension to add new classes, striking the balance between simplicity and extendability. Our modeling choices enable the system to perform zero-shot generalization to unseen classification targets and questions. Our method uses information gain to select the best question to ask

	FAQ Suggestion		Bird Identification	
	Acc@1	Acc@3	Acc@1	Acc@3
No Interaction (BM25)	26%	31%	N.A.	N.A.
No Interaction (RoBERTa$_{\text{BASE}}$)	$30 \pm 0.5\%$	$45 \pm 0.6\%$	$17 \pm 0.3\%$	$29 \pm 0.3\%$
No Interaction (RNN)	$38 \pm 0.5\%$	$61 \pm 0.3\%$	$23 \pm 0.1\%$	$41 \pm 0.2\%$
No Interaction (RNN + self-attn)	$39 \pm 0.5\%$	$63 \pm 0.4\%$	$23 \pm 0.1\%$	$41 \pm 0.1\%$
Random Interaction	$39 \pm 0.3\%$ ($38 \pm 0.1\%$)	$62 \pm 0.4\%$ ($63 \pm 0.2\%$)	$25 \pm 0.1\%$	$44 \pm 0.1\%$
No Initial Query Interaction	$46 \pm 0.5\%$ ($46 \pm 0.1\%$)	$66 \pm 0.6\%$ ($67 \pm 0.3\%$)	$29 \pm 0.2\%$	$50 \pm 0.3\%$
Our Approach	**79 ± 0.7%** ($75 \pm 0.4\%$)	**86 ± 0.8%** ($83 \pm 0.4\%$)	**49 ± 0.3%**	**69 ± 0.5%**
w/ threshold	$73 \pm 0.6\%$ ($69 \pm 0.6\%$)	$82 \pm 0.7\%$ ($81 \pm 0.6\%$)	$41 \pm 0.3\%$	$59 \pm 0.4\%$
w/ fixed turn	$71 \pm 1.0\%$ ($68 \pm 0.4\%$)	$81 \pm 0.9\%$ ($81 \pm 0.6\%$)	$39 \pm 0.2\%$	$56 \pm 0.4\%$
w/ $\lambda = 1$	$66 \pm 0.8\%$ ($64 \pm 0.2\%$)	$71 \pm 1.0\%$ ($73 \pm 0.2\%$)	$40 \pm 0.1\%$	$56 \pm 0.2\%$

Table 2: Performance with simulated interactions. We evaluate our approach and several baselines using Accuracy@{1, 3}. Best performance numbers are in bold. We report the averaged results as well as the standard deviations from three independent runs for each model variant and baseline. For FAQ Suggestion, in parentheses, we provide zero-shot results, where the system has access to tags only for training questions.

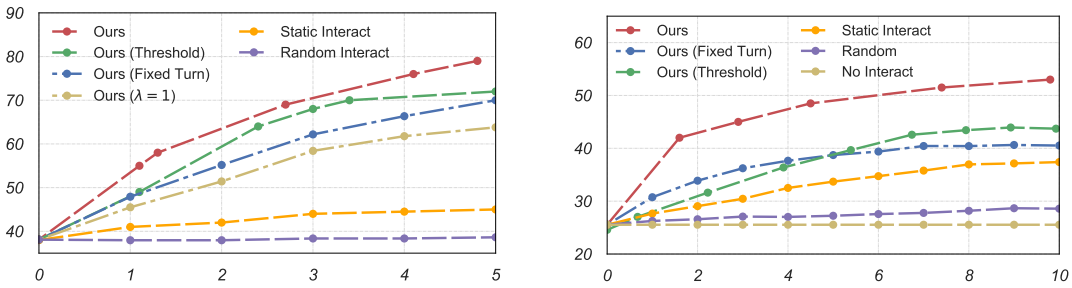

Figure 3: Accuracy@1 (y-axis) against turns of interactions (x-axis) for FAQ (left) and Birds (right) tasks.

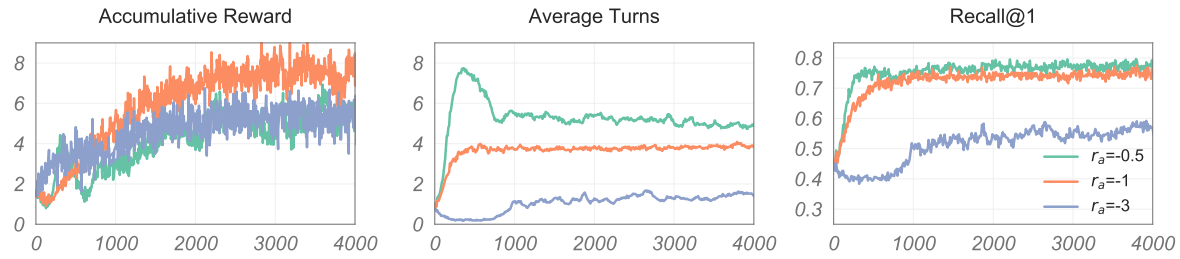

Figure 4: Learning curves of our full model. We show accumulative reward (left), interaction turns (middle), and Accuracy@1 (right) on the test set, where x-axis is the number of episodes (400 trials per episode). The results are compared on different turn penalty r_a.

at every turn, and a lightweight policy to efficiently control the interaction. We demonstrate that the system can be bootstrapped without any interaction data and show effectiveness on two tasks. A potential future research direction is to bridge the gap between this simple bootstrapping paradigm and the incorporation of user free-form responses to allow the system to handle free-text responses. We hope our work will encourage more research on different possibilities of building interactive systems that do not necessarily require handling full-fledged dialogue, but still benefit from user interaction.

Acknowledgments

We thank Derek Chen, Alex Lin, Nicholas Matthews, Jeremy Wohlwend, Yi Yang and the anonymous reviewers for providing valuable feedback on the paper. We would also like to thank Michael Griffths, Anna Folinsky and the ASAPP annotation team for their help on setting up and performing the human evaluation. Finally, we thank Hugh Perkins, Ivan Itzcovich, and Brendan Callahan for their support on the experimental environment setup.

References

Mohammad Aliannejadi, Hamed Zamani, Fabio Crestani, and W. Bruce Croft. 2019. Asking clarifying questions in open-domain information-seeking conversations. In *Proceedings of the 42Nd International ACM SIGIR Conference on Research and Development in Information Retrieval*.

Yoav Artzi and Luke Zettlemoyer. 2011. Bootstrapping semantic parsers from conversations. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*.

Prithvijit Chattopadhyay, Deshraj Yadav, Viraj Prabhu, Arjun Chandrasekaran, Abhishek Das, Stefan Lee, Dhruv Batra, and Devi Parikh. 2017. Evaluating visual conversational agents via cooperative human-ai games. In *Fifth AAAI Conference on Human Computation and Crowdsourcing*.

Cen Chen, Chilin Fu, Xu Hu, Xiaolu Zhang, Jun Zhou, Xiaolong Li, and Forrest Sheng Bao. 2019. Reinforcement learning for user intent prediction in customer service bots. In *Proceedings of the 42Nd International ACM SIGIR Conference on Research and Development in Information Retrieval*.

Yihong Chen, Bei Chen, Xuguang Duan, Jian-Guang Lou, Yue Wang, Wenwu Zhu, and Yong Cao. 2018. Learning-to-ask: Knowledge acquisition via 20 questions. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*.

Eunsol Choi, He He, Mohit Iyyer, Mark Yatskar, Wen tau Yih, Yejin Choi, Percy Liang, and Luke Zettlemoyer. 2018. Quac : Question answering in context. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep reinforcement learning from human preferences. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 4299–4307.

Pei-Hung Chung, Kuan Tung, Ching-Lun Tai, and Hung yi Lee. 2018. Joint learning of interactive spoken content retrieval and trainable user simulator. In *INTERSPEECH*.

Abhishek Das, Satwik Kottur, José MF Moura, Stefan Lee, and Dhruv Batra. 2017. Learning cooperative visual dialog agents with deep reinforcement learning. In *Proceedings of the IEEE international conference on computer vision*.

Marin Ferecatu and Donald Geman. 2007. Interactive search for image categories by mental matching. In *2007 IEEE 11th International Conference on Computer Vision*.

Xiaoxiao Guo, Hui Wu, Yu Cheng, Steven Rennie, Gerald Tesauro, and Rogerio Feris. 2018. Dialog-based interactive image retrieval. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 678–688. Curran Associates, Inc.

Izzeddin Gur, Semih Yavuz, Yu Su, and Xifeng Yan. 2018. Dialsql: Dialogue based structured query generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

Braden Hancock, Paroma Varma, Stephanie Wang, Martin Bringmann, Percy Liang, and Christopher Ré. 2018. Training classifiers with natural language explanations. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

Huang Hu, Xianchao Wu, Bingfeng Luo, Chongyang Tao, Can Xu, Wei Wu, and Zhan Chen. 2018. Playing 20 question game with policy-based reinforcement learning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*.

Srinivasan Iyer, Ioannis Konstas, Alvin Cheung, Jayant Krishnamurthy, and Luke Zettlemoyer. 2017. Learning a neural semantic parser from user feedback. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

In-Ho Kang and GilChang Kim. 2003. Query type classification for web document retrieval. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*.

J. F. Kelley. 1984. An iterative design methodology for user-friendly natural language office information applications.

Adriana Kovashka and Kristen Grauman. 2013. Attribute pivots for guiding relevance feedback in image search. In *Proceedings of the IEEE International Conference on Computer Vision*.

Sang-Woo Lee, Tong Gao, Sohee Yang, Jaejun Yoo, and Jung-Woo Ha. 2019. Large-scale answerer in questioner's mind for visual dialog question generation. In *International Conference on Learning Representations*.

Sang-Woo Lee, Yu-Jung Heo, and Byoung-Tak Zhang. 2018. Answerer in questioner's mind: Information theoretic approach to goal-oriented visual dialog. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances*

in Neural Information Processing Systems 31. Curran Associates, Inc.

Tao Lei, Yu Zhang, Sida I. Wang, Hui Dai, and Yoav Artzi. 2018. Simple recurrent units for highly parallelizable recurrence. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*.

Jiwei Li, Alexander H Miller, Sumit Chopra, Marc'Aurelio Ranzato, and Jason Weston. 2016. Dialogue learning with human-in-the-loop. *arXiv preprint arXiv:1611.09823*.

Zhouhan Lin, Minwei Feng, Cicero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. 2017. A structured self-attentive sentence embedding. *arXiv preprint arXiv:1703.03130*.

Charles X. Ling, Qiang Yang, Jianning Wang, and Shichao Zhang. 2004. Decision trees with minimal costs. In *Proceedings of the Twenty-first International Conference on Machine Learning*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

Chen Qu, Liu Yang, W. Bruce Croft, Yongfeng Zhang, Johanne R. Trippas, and Minghui Qiu. 2019. User intent prediction in information-seeking conversations. In *Proceedings of the 2019 Conference on Human Information Interaction and Retrieval*.

Sudha Rao and Hal Daumé III. 2018. Learning to ask good questions: Ranking clarification questions using neural expected value of perfect information. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

Siva Reddy, Danqi Chen, and Christopher D. Manning. 2019. CoQA: A conversational question answering challenge. *Transactions of the Association for Computational Linguistics*.

Scott Reed, Zeynep Akata, Honglak Lee, and Bernt Schiele. 2016. Learning deep representations of fine-grained visual descriptions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Stephen Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, (4).

Daniel E Rose and Danny Levinson. 2004. Understanding user goals in web search. In *Proceedings of the 13th international conference on World Wide Web*.

Darsh Shah, Tao Lei, Alessandro Moschitti, Salvatore Romeo, and Preslav Nakov. 2018. Adversarial domain adaptation for duplicate question detection. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*.

Pushkar Shukla, Carlos E. L. Elmadjian, Richika Sharan, Vivek Kulkarni, Matthew Turk, and William Yang Wang. 2019. What should I ask? using conversationally informative rewards for goal-oriented visual dialog. *CoRR*.

Paul E. Utgoff. 1989. Incremental induction of decision trees. *Machine Learning*, 4.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Harm de Vries, Florian Strub, A. P. Sarath Chandar, Olivier Pietquin, Hugo Larochelle, and Aaron C. Courville. 2017. Guesswhat?! visual object discovery through multi-modal dialogue. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. 2011. The Caltech-UCSD Birds-200-2011 Dataset. Technical Report CNS-TR-2011-001, California Institute of Technology.

Sida I. Wang, Percy Liang, and Christopher D. Manning. 2016. Learning language games through interaction. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

Tsung-Hsien Wen, David Vandyke, Nikola Mrkšić, Milica Gašić, Lina M. Rojas-Barahona, Pei-Hao Su, Stefan Ultes, and Steve Young. 2017. A network-based end-to-end trainable task-oriented dialogue system. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*.

Ronald J Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.

Xianchao Wu, Huang Hu, Momo Klyen, Kyohei Tomita, and Zhan Chen. 2018. Q20: Rinna riddles your mind by asking 20 questions. *Japan NLP*.

Ziyu Yao, Yu Su, Huan Sun, and Wen-tau Yih. 2019. Model-based interactive semantic parsing: A unified

framework and a text-to-SQL case study. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*.

A Appendices

A.1 Data collection

We collect two types of data for the FAQ task. For the bird identification task we re-purpose existing data (Section 6).

Initial Query Collection Qualification One main challenge for the data collection process is familiarizing the workers with the set of target documents. We set up a two-stage process to ensure the quality of the initial queries. The first stage is to write paraphrases of a given target, which is often a question in the FAQ task. We first allow the full pool of Amazon Mechanical Turk workers to perform the task. After that, we manually inspect the written queries and pick the ones that are good paraphrases of the FAQs. We selected 50 workers that showed good understanding of the FAQs. In the second stage, workers are asked to provide initial queries with possibly insufficient information to identify the target. Out of the first 50 workers, we manually selected 25 based on the quality of the queries such as naturalness and whether they contain ambiguity or incompleteness by design. We used this pool of workers to collect 3,831 initial queries for our experiments.

Tag Association Qualification The goal of this annotation task is to associate tags with classification labels. We train a model on the collected initial queries to rank tags for each classification target. We pick out the highest ranked tags as positives and the lowest ranked tags as negatives for each target. The worker sees in total ten tags without knowing which ones are the negatives. To pass the qualification task, the workers need to complete annotation on three targets without selecting any of the negative tags.

Tag Association Task Details After the qualification task, we take the top 50 possible tags for each target and split them into five non-overlapping lists (i.e., ten tags for each list) to show to the workers. Each of the lists is assigned to four separate workers to annotate. We observe that showing only the top-50 tags out of 813 is sufficient. Figure A.1 illustrates this: after showing the top-50 tags, the curve plateaus and no new tags are assigned to a target label. Table A.1 shows annotator agreement using Cohen's κ score.

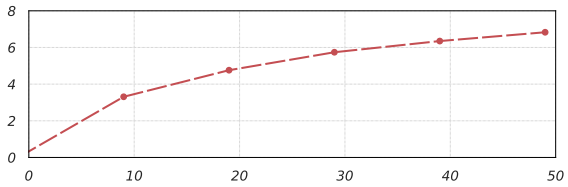

Figure A.1: Accumulated number of tags assigned to the targets (y-axis) by the workers against tag ranking (x-axis). The ranking indicates the relevance of the target-tag pairs from the pre-trained model. The curve plateaued at rank 50 suggesting that the lower ranked tags are less likely to be assigned to the target by the crowdsourcing workers.

	Tag Ranks				
	1-10	11-20	21-30	31-40	41-50
Mean # tags	3.31	1.45	0.98	0.61	0.48
N.A. (%)	1.9	30.7	43.6	62.1	65.2
Mean κ	0.62	0.54	0.53	0.61	0.61

Table A.1: Target-tag annotation statistics. We show five sets of tags to the annotators. The higher ranked ones are more likely to be related to the given target. The row mean # tags is the mean number of tags that are annotated to a target, N.A. is the percentage of the tasks that are annotated as *"none of the above"*, and mean κ is the mean pairwise Cohen's κ score.

A.2 Implementation Details

We use a single-layer bidirectional Simple Recurrent Unit (SRU) as the encoder for the FAQ suggestion task and two layer bidirectional SRU for bird identification task. The encoder uses pre-trained fastText (Bojanowski et al., 2017) word embeddings of size 300, hidden size 150, batch size 200, and dropout rate 0.1. The fastText embeddings remain fixed during training. We use the Noam learning rate scheduler (Vaswani et al., 2017) with initial learning rate $1e$-3, warm-up step 4,000 and a scaling factor of 2.0. For the self-attention model, we use a multi-head self-attention layer with 16 heads and a hidden size of 64 for each head. The same dropout rate used for the text encoder is applied to the self-attention layer. For the no interaction model with the RoBERTa encoder, we use the RoBERTa$_{\text{BASE}}$ model implemented by Hugghing Face (Wolf et al., 2019). The RoBERTa$_{\text{BASE}}$ model is fine-tuned with learning rate of $1e$-5, warmup step of 1,000, weight decay of 0.1, batch size of 16

and gradient accumulation step of 10. The policy controller is a two layer feed-forward network with a hidden layer of size 32 and ReLU activations. The network takes the current turn and the top-k values of the belief probabilities as input. We choose $k = 20$ and allow a maximum of 10 interaction turns.

A.3 Additional Analysis

We use the user simulator for further analysis of our system performance and alternative configurations.

Text Encoder Training Table A.2 shows the breakdown analysis of different ways to train the text encoder. We use initial queries as well as paraphrase queries to train the encoder, which has around 16K target-query examples. To analyze the effectiveness of tags in addition to initial queries, we generate pseudo-queries by combining existing queries with sampled subset of tags from the targets. This augmentation strategy is useful to improve the classification performance. We also observe that using the set of tags instead of initial queries as text inputs for a specific target label improves classification performance, indicating that the designed tags can capture the target label well. Finally, when we concatenate user initial queries and tags and use that as text input to the classifier, we achieve Accuracy@1 of 76%. In our full model, we achieve 79% with only querying about five tags.

Performances of Different Encoders Table A.3 show our system performance with different text encoders for both tasks.

A.4 Human Evaluation

Each interaction session starts with presenting a user scenario (e.g., a bird image or a phone issue). The user types an initial natural language query and answers follow-up questions selected by the system.

FAQ Suggestion We design a user scenario for each target to present to the worker. At the end of each interaction, the predicted FAQ and the ground truth are presented to the user, as shown in the top right panel in Figure A.2. The user answers the following questions: 'how natural is the interaction?' and 'do you feel understood by the system during the interactions?' on the scale of 1 (strongly disagree) to 5 (strongly agree), which we record as naturalness and rationality in Figure 2 and Table 1. Our full model performs best on Accuracy@1, naturalness and rationality. We show human evaluation examples in Table A.4.

Bird Identification The interface for bird identification task is similar to the FAQ suggestion task. Instead of presenting a scenario, we show a bird image to the user. The user needs to describe the bird to find out its category, which is analogous to writing an initial query. When answering system questions about attributes, we allow the user to reply 'not visible' if part of the bird is hidden or occluded. Given this reply, the system stops asking binary questions from the same label group. For example, if a user replies 'not visible' to a the question 'does the bird has a black tail?', then questions such as 'does the bird has yellow tail?' and 'does the bird has red tail?' will be skipped for the rest of the interaction. At the end of the interaction, the predicted and ground-truth bird images along with their categories are presented to the user as illustrated at the bottom right panel in Figure A.2. The user fills out a questionnaire as in FAQ domain. The bird identification task is very challenging because of its fine-grained categories, where many bird images look almost identical while belonging to different classes. Our full system improves classification accuracy from 20% to 45% against non-interactive baselines after less than three turns of interaction. To better understand the task and the model behavior, we show the confusion matrix of the final model prediction after interaction in Figure A.3. Of the 200 bird classes, there are 21 different kinds of sparrows and 25 different warbler. Most of the model errors are due to mistakes between these fine-grained bird classes. Figure A.4 shows how the confusion matrix changes when adding the interaction. The model makes improvement in distinct and also similar bird types.

Text Input		Init Query		Init Query + Tags		Init + Paraphrase Query		Full Data	
init query	tags	Acc@1	Acc@3	Acc@1	Acc@3	Acc@1	Acc@3	Acc@1	Acc@3
✓	✗	0.28	0.47	0.32	0.51	0.35	0.60	0.38	0.61
✗	✓	0.31	0.50	0.57	0.79	0.56	0.74	0.70	0.87
✓	✓	0.36	0.58	0.55	0.79	0.63	0.81	0.76	0.91

Table A.2: Comparison of text encoders trained on different textual inputs and evaluated on three different prediction tasks. The model uses (a) the initial queries, (b) all attribute tags, or (c) both initial queries and tags as text inputs to predict the target. Each model is evaluated using Accuracy@$\{1, 3\}$.

	FAQ Suggestion			Bird Identification		
	RNN	RNN + attention	RoBERTa$_{\text{BASE}}$	RNN	RNN + attention	RoBERTa$_{\text{BASE}}$
No Interaction	38%	39%	30%	23%	23%	17%
Random Interaction	39%	38%	31%	25%	24%	17%
No Initial Query Interaction	46%	41%	37%	29%	29%	21%
Our Approach w/ threshold	72%	73%	54%	41%	38%	33%
Our Approach w/ fixed turn	71%	68%	47%	39%	37%	32%
Our Approach w/ $\lambda = 1$	66%	67%	52%	40%	37%	32%

Table A.3: Accuracy@1 of our system with three different encoders. For all experiments, maximal number of turns is set to five.

U:	How can I text my friends house phone?
S:	Text message to landline?
U:	Yes.
S:	Do you want to send or receive text?
U:	Yes.
S:	Please read: *How do I send a text message to landline?*
U:	I'm in sprint's one up and want to upgrade my phone.
S:	Do you want annual upgrade option?
U:	Yes.
S:	Do you want to exercise an option?
U:	Yes.
S:	Please read: *I am enrolled in Sprint One Up. How do I exercise my annual upgrade option?*
U:	Battery ran out immediately.
S:	How can I help you with your phone device?
	App or feature information; Fee and charge; Troubleshooting; Not applicable
U:	Troubleshooting.
S:	Are you using mobile wifi hotspot?
U:	Yes.
S:	Please read: *Does using my phone as a mobile Wi-Fi hotspot drain my battery?*

Table A.4: Three examples of user interactions from the FAQ suggestion human evaluation.

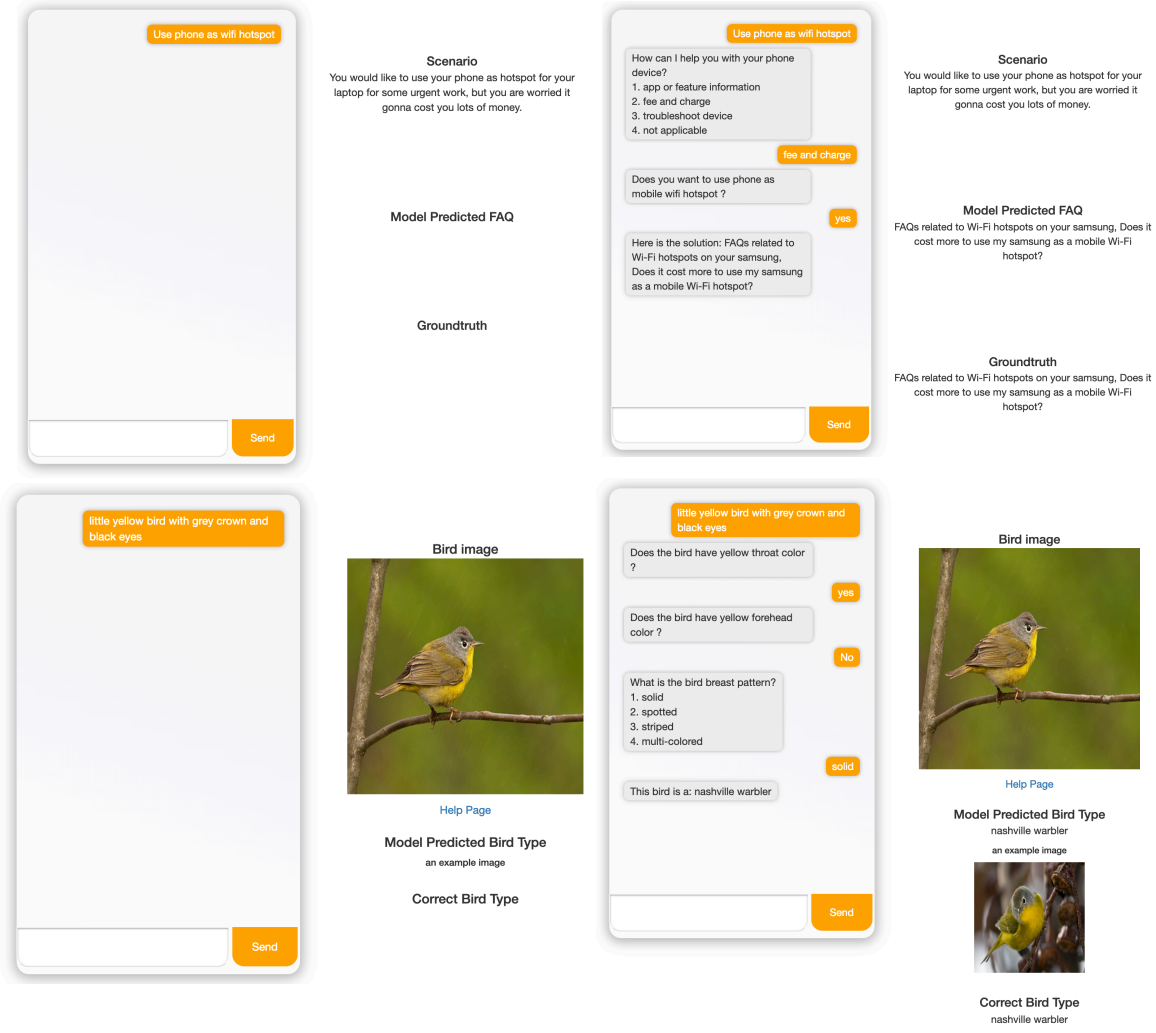

Figure A.2: The user interface for FAQ Suggestion (top) and Bird Identification (bottom) tasks. The left panel shows the interface at the beginning of the interaction and the right panel shows the interface at the end of the interaction.

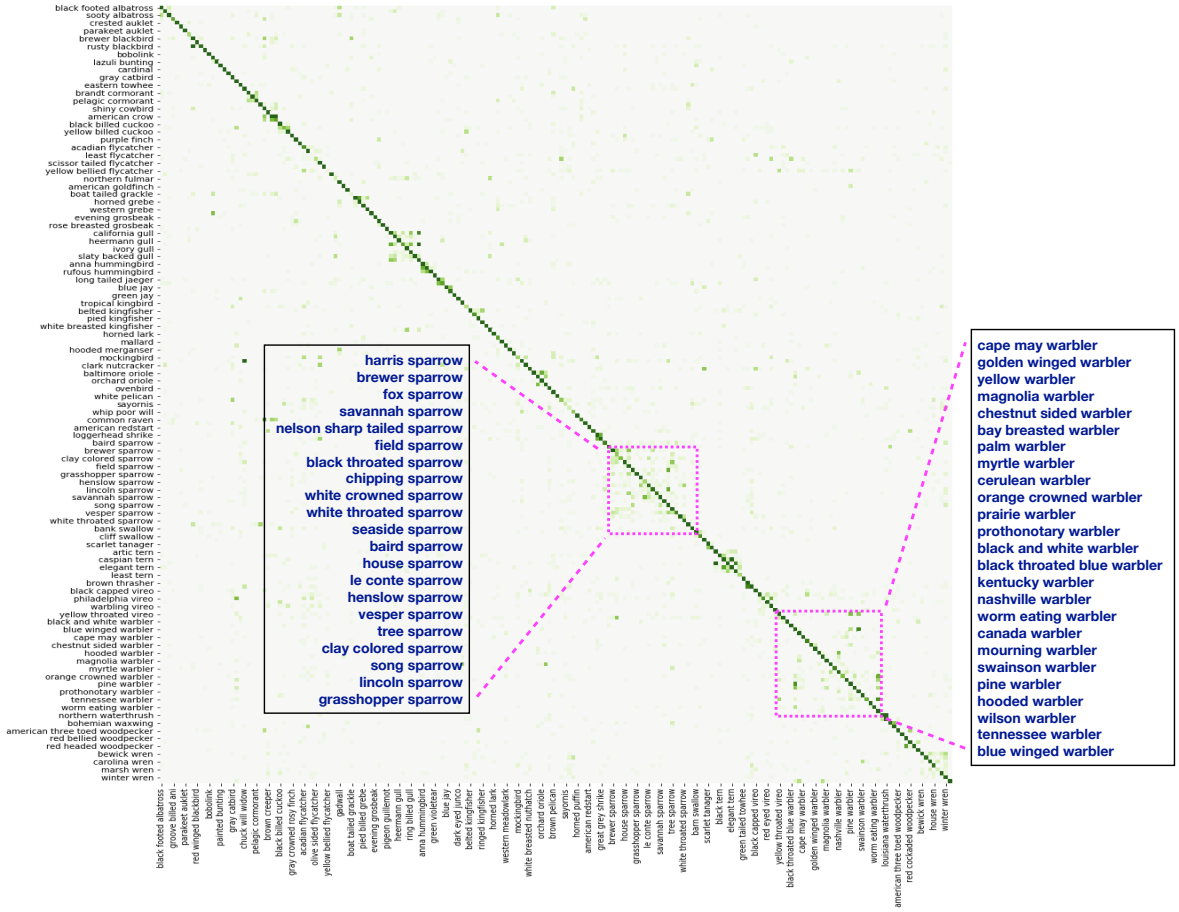

Figure A.3: Confusion matrix of our final output for bird identification task.

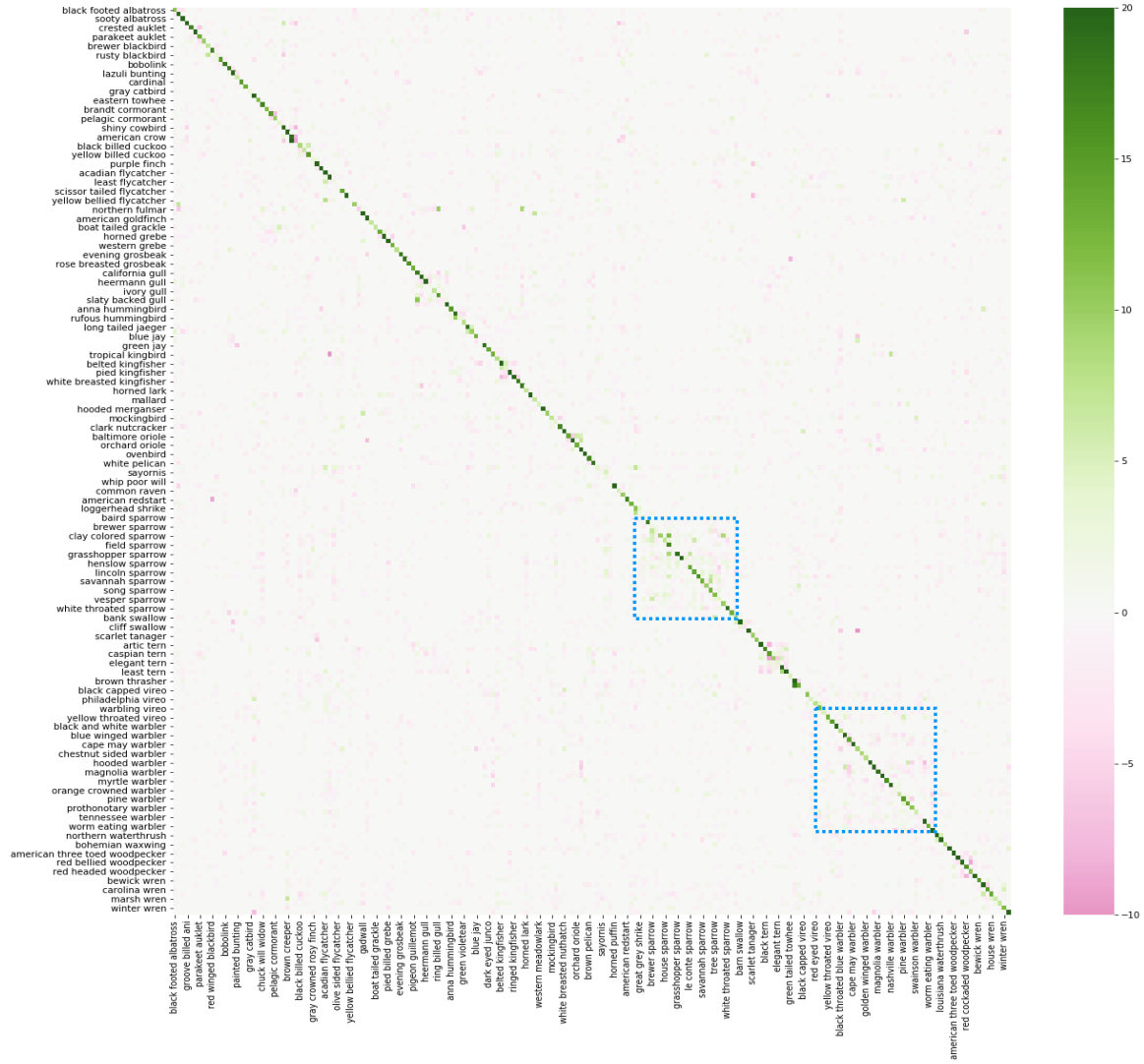

Figure A.4: Confusion matrix difference between the initial query with and without the interactions. High values along the diagonal and low values elsewhere are good.