

# Just To Be Sure: Building A Virtual Assistant With Active Listening Skills

Jacob Stern<sup>1</sup>, Seong-Eun Cho<sup>1</sup>, Yeganeh Nasiri<sup>1</sup>, Aadesh Neupane<sup>1</sup>, Piper Armstrong<sup>1</sup>

<sup>1</sup>Brigham Young University

## Abstract

Task-based chatbots tend to suffer from either overconfidence or ignorance – giving a response that is confidently wrong or completely uncertain (e.g. “I don’t know”). A chatbot that could identify the source of its uncertainty and ask a clarifying question would lessen the burden of query reformulation for the user. We introduce a two-turn query-response-query (QRQ) task, in which a user queries the chatbot and the chatbot must ask a clarifying question, which results in a second user query that clarifies the user’s intent. We evaluate performance in two ways: 1) by the perplexity of the response on the Taskmaster-2 dataset, and 2) by information acquisition between the first user query and the second user query, as measured by an intent classifier. We train a variety of architectures for these objectives, including a supervised encoder-decoder transformer and an unsupervised system trained to acquire more information from the second query than the first. Although the unsupervised system does not currently improve on baseline, there are positive indications that a similar approach could yield positive results in future.

## 1 Introduction

Many of the current industrial-level voice assistants fall short in two areas: accuracy and specificity. These shortcomings can be addressed by rule-based systems that try to account for all request types (e.g. Alexa Skills and Google Assistant Actions), but even with these kinds of complicated rule-based systems sourced from thousands of developers, it is not hard to push the voice assistant into a domain where it cannot respond accurately or specifically.

We aim to reduce the number of inaccurate and ambiguous responses by training an assistant to respond to a user query with a clarifying question, which then prompts the user to provide a second query that clarifies the user’s initial intent. By asking a clarifying question and better identifying the user intent, the assistant can avoid giving confidently wrong responses. Likewise, an ambiguous response such as “I don’t know” can be replaced with a more specific response if the assistant asks a clarifying question and receives a second query.

With current voice assistant models, a common source of friction in human-assistant interaction is the experience of the user needing to communicate their intentions multiple times in order to receive the desired response from the assistant. This often acts as a barrier in humans choosing to interact through voice assistance. This is a frustrating, stressful, and mentally demanding task for the user. However, if the assistant could offer a response that clues the user in on what it is uncertain of, the user has a better chance at re-framing the query in a way that results in a successful interaction with the assistant

We train a system in a three-turn scenario, where the user provides a query (Q1), the assistant provides a response (R), and the user provides another query (Q2) that increases the amount of information provided to the assistant. We do so in an unsupervised manner, such that the only necessary data is a dataset of user queries. We term this task “QRQ”. Because our model is trained with an objective to maximize the information gain, we call our model the “Information Acquisition (IA) Transformer Model”

We fine-tune a generatively pre-trained transformer as the main component of our pipeline, to provide the ability to process any query expressed in natural language, as well as respond in natural language. Furthermore, we leverage a pre-trained language model as a language partner, to remove the reliance on a large dataset of Q-R-Q triples. We also use an intent classification model at turn 1 and turn 3 of the conversation, and train with the objective of increasing information between turn 1 and turn 3. This turns the task into an unsupervised learning problem, in which the model learns to respond based solely on a query dataset. This has broad applications for any scenario with a large number of query examples, but no ground truth responses (e.g. any currently deployed voice assistant has access to almost unlimited queries).

Figure 1 shows the high-level control flow of our IA model. In the first stage (T1), the user provides a query (Q1). The model predicts a multi-label intent vector for that user’s query. Based on the predicted intent vector, the natural language generation module generates a clarifying question as its response (R) which prompts the user to ask another query (Q2), providing the system with more information than before (i.e., more slots in the multi-label intent vector is filled). For example, in Figure 1, the system response ‘What city are you in?’ directed the user to query ‘Frankfurt, Germany...’

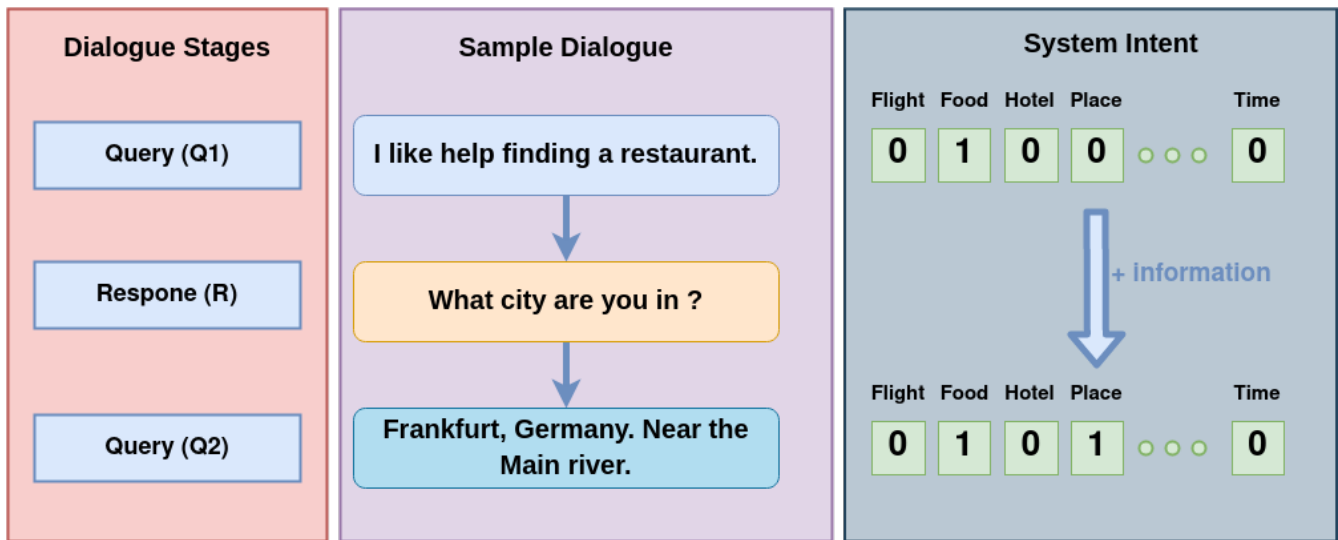


Figure 1: Stages of the QRQ system with an example dialogue.

such that the system has more information about the user’s intent than before.

The primary contributions of our system are as follows:

- We propose an evaluation task, which we term “QRQ”, for the ability of an assistant to clarify the intent of a user.
- We provide an end-to-end language model for clarifying response generation, removing reliance on a manually curated dictionary of intent-response pairs.
- We provide an unsupervised training scheme in which a neural network can learn to provide a clarifying response without seeing any examples of clarifying responses.

## 2 Related Work

Our work is related to research in the areas listed below.

### 2.1 Existing apps and products

There is a rich line of work on developing conversational agents where natural language processing lies at the heart of the technology. Some of them engage users in a text-based conversations to draw out their views and opinions [Xiao *et al.*, 2020b] and ask open-ended questions to collect the data that they need [Xiao *et al.*, 2020a]. Past studies have examined chatbots as personal assistants [Liao *et al.*, 2018], interviewers [Li *et al.*, 2017], mental health supporters [Inkster *et al.*, 2018] and even AI companions [Ta *et al.*, 2020]. The major issues with these conversational agents are inaccuracy and ambiguity. Both inaccuracy and ambiguity show a lack of understanding of user intent. In this work we apply the active listening principle of asking clarifying questions to mitigate these problems by requesting further information.

### 2.2 Uncertainty

Uncertainty estimation is important for ensuring safety and robustness of AI systems and chatbots. Some works aim to investigate how to obtain uncertainty estimates for tokens in

a sequence and for whole sequence classification [Malinin and Gales, 2021]. In order to decrease the uncertainty of our model, our agent asks clarifying questions which leads the model to gain more certainty about the user’s query. To measure model performance before and after asking a clarifying question, we introduce the “Information Acquisition Score” described in section 4.3.

### 2.3 Natural language generation and language models

Over the past few years, several chatbot platforms have been developed to facilitate the creation of chatbots. Some of these platforms such as [Sabharwal and Agrawal, 2020] have NLP capabilities for the creation of chatbots with active listening capabilities. More recently, research uses language models such as T5 to create chatbots [Bird *et al.*, 2020]. For the current work, we used the pre-trained T5 transformer model. We rely on this pre-trained transformer to act as a language partner, allowing us to train our model in an essentially unsupervised manner.

## 3 Dataset

We use the Taskmaster 2 Dataset [Byrne *et al.*, 2020], an extension of the Taskmaster-1 Dataset [Byrne *et al.*, 2019]. Taskmaster-2 includes 17,289 dialogues (341,801 utterances) in 7 domains. Each utterance is fully annotated using 90 possible annotations. We define these annotations as intents for the data. A sample dialogue can be seen in Table 1.

We use individual user utterances (152,668 instances) as well as concatenated consecutive user utterances (138,892 instances) to train the intent classifier.

We use user/assistant pairs (147,875 instances) to fine-tune the QR model.

We use users/assistant/user triples (138,892 instances) to fine-tune the QRQ model conversational partner.

“conversation_id” : “dlg-00187b735-345e-4c43-b744-e955b6836c43”		
“instruction_id” : “movie-26”		
Annotation	Speaker	Text
<b>movie_search.location.theater</b>	Assistant	How can I help you?
	User	What moves are playing?
	Assistant	I can help you with your movie search.
	Assistant	Where are you located?
	User	Davis, California
<b>movie_search.genre</b>	Assistant	Do you have a type of movie you wanted to go see?
	User	Action
<b>movie_search.name.movie</b>	Assistant	Just one moment
	Assistant	Here is what I found
	Assistant	Baby driver.
	Assistant	The mummy
	Assistant	Guardians of the Galaxy: Vol 2.
	User	The mummy
<b>movie_search.time.start</b>	Assistant	Okay. The Mummy is playing at 4:30 pm this afternoon at Regal Davis Stadium 5.
<b>move_search.time.start</b>	User	Ok
	Assistant	Enjoy your movie!
	User	Thanks!

Table 1: An example of a conversation with annotations from the Taskmaster-2 dataset.

	Sn	Label	Text
Baseline	1	Input Text	I would like you to help me finding a hotel in San Francisco, Downtown area, California.
	1	Output Text	Ich möchte Sie bitten, mir bei der Suche nach einem Hotel in San Francisco, Downtown area, California zu helfen.
	1	Target Text	What is the price range?
Fine-tuned	1	Input Text	I’d like to see the Edge of Tomorrow trailer.
	1	Output Text	Okay, I have your personal details on file. Your order will be ready for pickup in 20 minutes.
	1	Target Text	Anything else I can help you with ?

Table 2: Sample query/response text from our encode/decoder model.

## 4 QRQ Task

Though the ultimate goal involves making a decision of whether a clarifying response is necessary, for simplicity, we restrict our study to the QRQ task. This consists of a three turn scenario with the details outlined below:

1. The user provides a query (Q1), which is passed through an intent classifier to provide probabilities for the presence/absence of each intent in Q1.
2. The query is also passed through the response generator model, which provides a response (R), which is passed to a language model, which provides a second query (Q2).
3. Q1, R, and Q2 are passed through the intent classifier, which provides probabilities for the presence/absence of each intent in the three-turn dialogue.
4. The performance of the model is measured as the difference of the summed intents present at the end versus the beginning of the dialogue. We call this difference the “Information Acquisition” score.

## 5 Models

Our main research objective is to train a working response generator using the Information Acquisition (IA) method. The goal of the IA method is to train a model in an unsupervised setting, where we are only given the input query and train a response generator. In order to do this, we must first train the Intent Classifier, the QR model, and the QRQ model. The specifications for the three trained models are explained below.

### 5.1 Intent Classifier

The Intent Classifier takes as input a user query and recognizes which of the intents in a specific domain are present or not present in the query. This can all be obtained in the Task Master dataset. For example, a user query might be “Hi, I need help finding a sandwich shop to go eat at.” and this query is tagged with “restaurant.type.food” since the user is looking for a restaurant that serves a specific food. With 7 different domains and several intents in each domain, there are 90 total unique intents. The goal of the Intent Classifier is to identify which of the intents are present in the query. Since a user query can have multiple intents, this classification task resembles a multi-label classification problem.

We use a pre-trained T5 Transformer model to encode the input query, and then use an attention mechanism to convert the variable length Transformer encoding into a fixed-size vector. We then pass this vector through three linear layers, with the output dimension being equivalent to the number of intent classes. We use a standard cross entropy loss for multi-label classification as described by equation 1 to train the network.

$$\text{minimize } \mathbb{L}_1 = - \sum_{i=1}^N \sum_{c=1}^M p(y_{ic}) * \log(\hat{y}_{ic}) \quad (1)$$

where  $N$  is the number of samples in the batch,  $M$  is the total number of classes,  $y_{ic}$  is the target for class  $c$  in the  $i$ th

sample, and  $\hat{y}_{ic}$  is the predicted value for class  $c$  in the  $i$ th sample.

### 5.2 QR Model

The QR model is our baseline model. The training objective of this model is to take in the user query input and predict the response generated by the assistant system in a simple encoding decoding method. We use a pretrained T5 Transformer model and fine-tune it using the QR dataset. The purpose of training the QR model is to get a baseline performance of the model that uses the same architecture as our final Information Acquisition model.

$$\text{maximize } p(x) = \prod_{i=1}^n p(s_n | s_1, s_2, \dots, s_{n-1}) \quad (2)$$

Equation 2 is the standard language modeling approach which is usually framed as unsupervised distribution estimation from a set of examples  $(x_1, x_2, \dots, x_n)$  each composed of variable length sequences of symbols  $(s_1, s_2, \dots, s_n)$ . We use equation 2 to fine-tune the pre-trained T5 transformer model on the Taskmaster2.

### 5.3 QRQ Model

The QRQ model is a crucial component for our evaluation of each of the models and the Information Acquisition model. Because we are not able to obtain actual user queries followed by our model’s response (Q2), we must train a language model to simulate user’s follow-up query (Q2) to the response generated by our model (R). We do this by fine-tuning a pretrained T5 Transformer model, just like how the QR model is trained. The only difference is that instead of taking as input the Q1 from the dataset and generating the response (R), it takes in both Q1 and R from the dataset modified as QRQ triplets, and generates the follow-up query (Q2).

### 5.4 Information Acquisition Model

In voice-assistant settings, user query data is practically unlimited. What is missing are sensible and specific system responses. The full Information Acquisition (IA) model was designed to provide a way to train using only a dataset of user queries. This works by using a pretrained language model as a differentiable conversational partner. This allows us to fine-tune the response model via a loss that is based on the simulated user’s second response. Crucially, this scheme requires no labeled QRQ triples; it only needs user queries, Q.

The diagram of the model architecture is shown in Figure 2. Q1 Encoder is a pretrained T5 Transformer model. Intent Classifier is the trained classification model described in section 5.1. It takes as inputs the outputs of the Q1 Encoder and outputs the logits of the intent classes.

R Decoder is the main part of the architecture we want to be trained. It takes as the input the encoded input embedding and generates the desired clarifying question. We accomplish this by fine-tuning a pretrained T5 model from section 5.2.

Q2 Decoder is a pretrained T5 decoder fine-tuned to produce user queries with a QRQ training scheme.

Q2 Encoder is the same pretrained T5 encoder as Q1 Encoder.

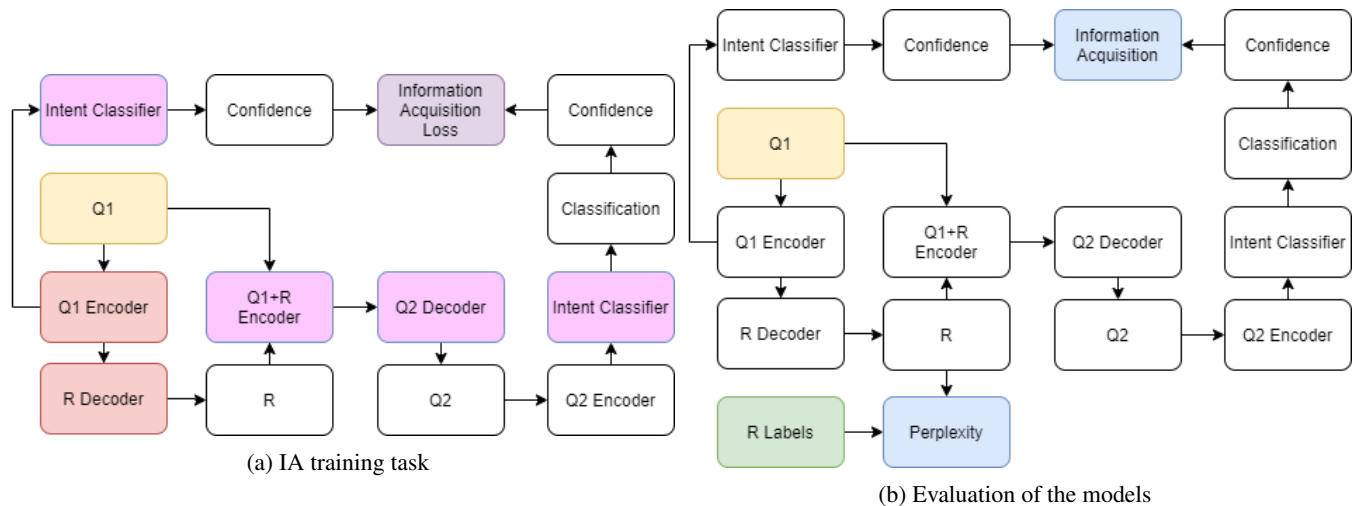


Figure 2: a) Diagram showing how the model is trained using the IA objective. The box in yellow (Q1) represents the input. The box in purple represents the training objective. Boxes in red represent models that are trained by the objective. Boxes in pink represent models that are already trained, and their weights are frozen. b) Diagram showing the evaluation scheme for each of our models. Boxes in blue represent our evaluation metric, and the box in green represent test labels. Observe the similarity between the IA training mechanism and the evaluation mechanism.

The training objective is to maximize the information acquired by the intent classifier using the next response. We do this by comparing the number of positive classes from the first query with the next query so that the information acquisition can be maximized. We use equation 3 to train the QRQ model.

$$\text{maximize } \mathbb{L}_2 = \sum_{i=1}^N \sum_{c=1}^M p(\hat{y}_{ic}^{Q2} = \text{present}) - p(\hat{y}_{ic}^{Q1} = \text{present}) \quad (3)$$

where  $N$  is the number of samples in that batch,  $M$  is the total number of classes,  $\hat{y}_{ic}^{Q1}$  is the probability that class  $c$  is present in the  $i$ th sample for the first query  $Q1$ , and  $\hat{y}_{ic}^{Q2}$  is the probability that class  $c$  is present in the  $i$ th sample for the second query  $Q2$ . We call equation 3 information acquisition loss and the notion behind this loss is that the system response should increase the information from first query  $Q1$  to second query  $Q2$ .

R Encoder exists so that there can be a smooth path for the gradient to pass through from the confidence of the next intent classification. There is learning here for the sake of R Decoder learning.

Q2 Decoder is a fine-tuned language model that generates the next query given the initial query (Q1) and the system's response (R).

## 6 Experiments

### 6.1 Overview

In this section, we show how our information acquisition approach to dialog systems enables the machine to ask clarifying questions that increases its information cache resulting in better perplexity scores and quality responses. The

results show that it is possible to train end-to-end dialogue models without access to the original labelled responses to actual user queries. Below we describe the evaluation methods followed by detailed discussion of the experiment results.

### 6.2 Experiment Setup

As mentioned earlier, we use the Taskmaster 2 dataset [Byrne *et al.*, 2020] for all our experiments. We avoid the practice of training an end-to-end method on labeled user's response as the goal of the experiment is to validate if the model can ask appropriate clarifying questions that increase its information cache. In order to achieve this goal, we couple three different models: a) the Intent Classifier, b) the QR model, and c) the IA model. We separately train each model on a slight variation of the Taskmaster 2 dataset.

### 6.3 Evaluation

Understanding the user's intent is vital to precisely address the query in dialog systems. To understand the role of intent in our method, we have three different models. Each model is evaluated differently. For the Intent classifier, we use F1 score and accuracy to assess the performance. For the QR and the IA model, we use perplexity scores and information acquisition to assess the performance.

#### Intent Classification

The Taskmaster 2 dataset is hand annotated such that each utterance is labeled with 90 possible annotations. We call these annotations intents. So the input to the intent classification model is a sentence and a binary vector of length 90. The dataset is split into two sets: training (80%) and validation (20%). The cross entropy loss is used during training time but F1 scores, precision, and recall are used to assess the performance for the validation datasets as this is multi-class classification problem.

Label	Text
Q1	How about the 7:00 p . m ?
R	The return flights are at 12 p . m , 4:05 p.m. 11:35 p.m.
Q2 Output	How about the 4:05 p.m?
Q2 Target	Let’s do the 4:05 p.m. return on the flight. That would be nice.
Q1	Does that include the return flight?
R	How does 5:30 a.m. work ?
Q2 Output	Well, 5:30 a.m. works.
Q2 Target	Well, the evening works better.
Q1	And I would like to fly during the night time and a red-eye.
R	Alright. What else ?
Q2 Output	I would like to fly in the morning and I would like to fly in the morning.
Q2 Target	I would like it to be under \$500.

Table 3: Samples from our fine-tuned QRQ model conversational partner.

	Perplexity	Information Acquisition
QR before fine-tuning (Baseline)	5.28e10	.1127
QR Transformer (Supervised)	2.32390	.1128
IA Transformer (Unsupervised)	inf	.0977
IA Transformer with QR fine-tuning (Supervised/Unsupervised)	inf	.0977

Table 4: Evaluation metrics for our different models.

	F1 Score	Accuracy
QQ Intent Classifier	0.94	0.95
QQ Classifier (Baseline)	0.21	0.24

Table 5: The intent classifier acts as a binary classifier for the presence/absence of each class. Thus, we report its F1 score as a binary classifier for each class, averaged across classes. Likewise, we report the average binary classification accuracy across classes.

### Encoder-Decoder

For this task the taskmaster 2 dataset is unchanged. Similar to Intent Classification, the dataset is split into two sets: training (80%) and validation (20%). During the training time, the standard loss function is used to train the model. Perplexity and information acquisition are used to assess the response of the model.

### QRQ

For this task the same dataset used for Intent Classification is used. During the training time, the loss function is trying to increase the information acquisition by comparing the accuracy of its intent classification before and after the response. Perplexity, information acquisition, and human evaluation are used to measure the quality of the response of the model.

## 6.4 Results

Figure 3 shows that the QQ intent classifier outperforms the baseline classifier by a wide margin in both F1 Score and Accuracy. The attention mechanism is the only difference between our model and the baseline model. In the baseline model, we use a einsum operation between a uniformly ini-

tialized weight and the input tensor which acts as a naive attention mechanism. All other fully connected layers is same between our and baseline model. Both the models were trained on 80% of the dataset for 50 epoch with batch size of 64. It took around 5 hrs to train both model. The results were obtained by running inference on same validation set with batch size 64 for 478 iterations.

Table 2 shows the results of fine-tuning the QR model. There is an apparent increase in quality from the baseline to the fine-tuned model.

Table 3 shows samples of the fine-tuned QRQ output acting as a conversational partner. Some of the well-known problems with language models can be observed (e.g the repetition of phrases in the third sample). However, the quality of second queries seems adequate.

Table 4 shows the perplexity results for the Information Acquisition model. The Information Acquisition model both increases perplexity to infinity in contrast to the QR baselines and also under-performs those baselines in Information Acquisition.

## 7 Conclusion

Our current results suggest that the Information Acquisition Transformer may be finding a convenient local minimum. While this result is somewhat disappointing, it is nevertheless encouraging that although the perplexity of Information Acquisition responses increases to infinity, the information acquisition only falls somewhat. Perhaps a more sophisticated loss function or a reinforcement learning environment would be better suited for learning to acquire information more efficiently. If such is the case, it would still be possible to train

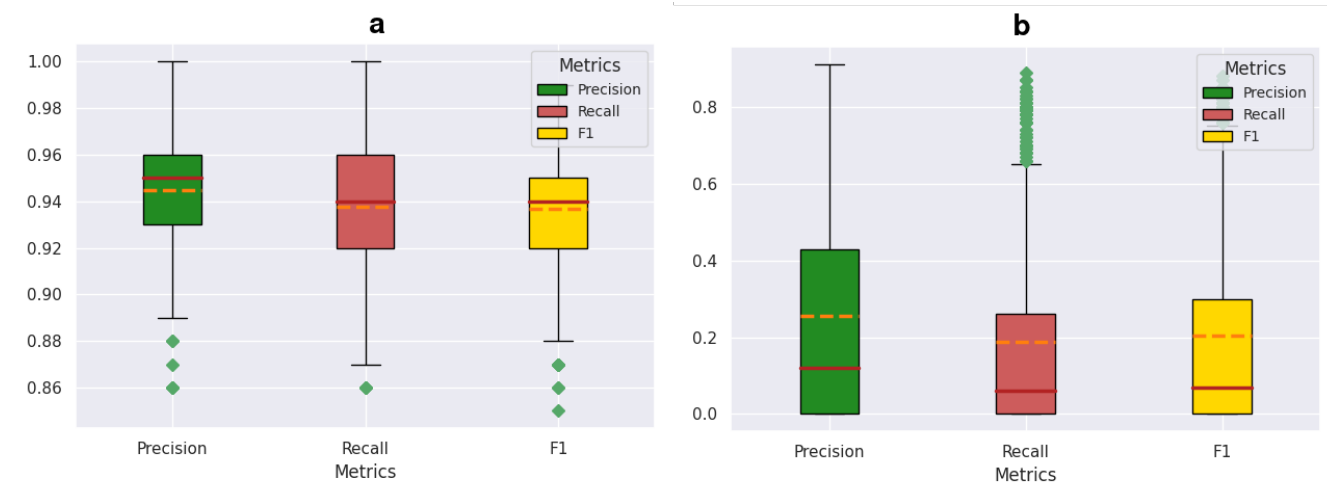


Figure 3: Our intent classification model a) outperforms the baseline model b) on the test set.

essentially unsupervised without the need for expensive labeling.

## 8 Future work

**Information discernment:** Our model tries to give a response that induces the user to give more information. However, it doesn't take into account the task of deciding whether it has enough information, and whether a response is necessary. Future work will include incorporating our model into a larger pipeline that accounts for this.

### Open-domain intent classes:

Furthermore, our model is limited to pre-defined intent classes. A truly powerful model will be able to compose concepts to generalize to intent classes that it has not seen - or even better, do away with classes altogether. Ideally, a conversational agent will be able to comprehend open-domain user intent and provide open-domain actions/responses appropriate to the perceived intent of the user.

**Open-domain action space:** On the note of open-domain responses, while our model gives free form responses to try to clarify the intent of a user, its focus is on intent comprehension. But the ability to comprehend a user's intent is only one factor in providing a satisfactory response. Our model does not address the inability of assistants to generalize to open-domain tasks. Future work is necessary to increase the action space of assistants, such that they are able to perform more tasks than are explicitly coded into the model.

We believe that unsupervised and self-supervised training objectives like the one introduced in this paper are a necessary step toward open domain intent comprehension and open domain response.

## A Reproducibility

The results reported in the paper can be reproduced with the code in the <https://github.com/NancyFulda/ActiveListener> github repository.

## References

- [Bird *et al.*, 2020] Jordan J. Bird, Anikó Ekárt, and Diego R. Faria. Chatbot interaction with artificial intelligence: Human data augmentation with t5 and language transformer ensemble for text classification, 2020.
- [Byrne *et al.*, 2019] Bill Byrne, Karthik Krishnamoorthi, Chinnadhurai Sankar, Arvind Neelakantan, Ben Goodrich, Daniel Duckworth, Semih Yavuz, Amit Dubey, Kyu-Young Kim, and Andy Cedilnik. Taskmaster-1: Toward a realistic and diverse dialog dataset. 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)*, pages 4506–4517, 2019.
- [Byrne *et al.*, 2020] Bill Byrne, Karthik Krishnamoorthi, Saravanan Ganesh, Amit Dubey, Andy Cedilnik, and Kyu-Young Kim. Taskmaster-2. <https://github.com/google-research-datasets/Taskmaster/tree/master/TM-2-2020>, 2020. Second dataset in series of three.
- [Inkster *et al.*, 2018] Becky Inkster, Shubhankar Sarada, and Vinod Subramanian. An empathy-driven, conversational artificial intelligence agent (wysa) for digital mental well-being: real-world data evaluation mixed-methods study. *JMIR mHealth and uHealth*, 6(11):e12106, 2018.
- [Li *et al.*, 2017] Jingyi Li, Michelle X Zhou, Huahai Yang, and Gloria Mark. Confiding in and listening to virtual agents: The effect of personality. In *Proceedings of the 22nd International Conference on Intelligent User Interfaces*, pages 275–286, 2017.
- [Liao *et al.*, 2018] Q Vera Liao, Muhammed Mas-ud Husain, Praveen Chandar, Matthew Davis, Yasaman Khazani, Marco Patricio Crasso, Dakuo Wang, Michael Muller, N Sadat Shami, and Werner Geyer. All work and no play? In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, pages 1–13, 2018.
- [Malinin and Gales, 2021] Andrey Malinin and Mark Gales. Uncertainty estimation in autoregressive structured prediction, 2021.
- [Sabharwal and Agrawal, 2020] Navin Sabharwal and Amit Agrawal. Introduction to google dialogflow. In *Cognitive virtual assistants using Google Dialogflow*, pages 13–54. Springer, 2020.
- [Ta *et al.*, 2020] Vivian Ta, Caroline Griffith,Carolynn Boatfield, Xinyu Wang, Maria Civitello, Haley Bader, Esther DeCero, and Alexia Loggarakis. User experiences of social support from companion chatbots in everyday contexts: thematic analysis. *Journal of medical Internet research*, 22(3):e16235, 2020.
- [Xiao *et al.*, 2020a] Ziang Xiao, Michelle X. Zhou, Wenxi Chen, Huahai Yang, and Changyan Chi. If i hear you correctly: Building and evaluating interview chatbots with active listening skills. *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, Apr 2020.
- [Xiao *et al.*, 2020b] Ziang Xiao, Michelle X. Zhou, Q. Vera Liao, Gloria Mark, Changyan Chi, Wenxi Chen, and Huahai Yang. Tell me about yourself. *ACM Transactions on Computer-Human Interaction*, 27(3):1–37, Jun 2020.