# Answerability: A custom metric for evaluating chatbot performance

**Pranav Gupta*** **Anand A. Rajasekar*†** **Amisha Patel**

**Mandar Kulkarni** **Alexander Sunell** **Kyung Hyuk Kim** **Ganapathy Krishnan**

**Anusua Trivedi†**

Flipkart US R&D Center, Redmond, USA 98052

## Abstract

Most commercial conversational AI products in domains spanning e-commerce, health care, finance, and education involve a hierarchy of NLP models that perform a variety of tasks such as classification, entity recognition, question-answering, sentiment detection, semantic text similarity, and so on. Despite our understanding of each of the constituent models, we often do not have a clear view as to how these models affect the overall platform metrics. To bridge this gap, we define a metric known as *answerability*, which penalizes not only irrelevant or incorrect chatbot responses but also unhelpful responses that do not serve the chatbot's purpose despite being correct or relevant. Additionally, we describe a formula-based mathematical framework to relate individual model metrics to the answerability metric. We also describe a modeling approach for predicting a chatbot's answerability to a user question and its corresponding chatbot response.

## 1 Introduction

Conversational AI has been making great strides in the past few years. Several commercial chatbots powered by NLP have been deployed for diverse sectors, ranging from banking to health care (Adewumi et al., 2022). While end-to-end chatbots based on a single neural network architecture have been proposed (Komeili et al., 2021; Adiwardana et al., 2020), most commercial organizations still deploy a hierarchy of machine learning models working together in unison to come up with an answer to a user's question, rather than relying on the output of a single end-to-end neural network, for instance, the popular Rasa NLU framework used by several industrial organizations (Bocklisch et al., 2017). In such a case, it is important to have a single unified metric that defines the effectiveness of a conversational AI product, such as a chatbot. Moreover, one needs to have a framework that links individual model metrics to the overall chatbot effectiveness metric. This way, we can understand the "weak links" in the entire chatbot workflow, i.e., models whose relative improvement can have the maximum effect on the global chatbot effectiveness metric. This is crucial for a commercial organization, where business impact needs to be routinely demonstrated, requiring teams to prioritize which models they are going to focus on improving.

Moreover, by incorporating other business-motivated factors such as helpfulness into the overall chatbot effectiveness metric, we are ensuring that we are optimizing not just for peak performance from each of the constituent machine learning models inside a chatbot, but also for the ability of the chatbot to serve the organization's business goals. For example, if an e-commerce website does not sell women's Reebok shoes of size 10, its chatbot might answer "correctly" to a user who asked if those shoes are available, by responding "No we do not have women's Reebok shoes of size 10." However, this answer is not "helpful," that is, a user shown this answer will not be tempted to search for other products on the website. A helpful answer could not only acknowledge the lack of Reebok shoes of the required size, but could also suggest other similar shoes of size 10 from a similar brand, say Skechers or Nike, so that an originally unhelpful answer could potentially become helpful. In this case, the answer is not only correct but also helpful, just like how a salesperson in a brick-and-

*Both authors contributed equally to this work

† {anand.ar,anusua.trivedi}@flipkart.com



*Proceedings of the 2nd Workshop on Natural Language Generation, Evaluation, and Metrics (GEM)*, pages 316 - 325

mortar store would be when they are asked about the availability of a certain product. Similar examples apply to other use-cases of conversational AI such as banking, governance, and health care. By emphasizing such business-relevant expectations from a task-oriented chatbot, we are encouraging it to provide a more relatable experience to the user, just like a human agent or salesperson.

This paper seeks to make 2 contributions: the first is to define a stringent global effectiveness metric for a task-oriented chatbot called "answerability," which penalizes not only incorrect or irrelevant responses but also unhelpful responses. Our metric is quite general in its definition and can be utilized in any chatbot application. Our second contribution is to describe a framework to relate the answerability metric to individual model metrics, along with a modeling approach for predicting the answerability for an individual user query-chatbot response pair. We shall focus on the concrete example of a pre-purchase e-commerce chatbot that answers user questions about products sold on an e-commerce marketplace to illustrate our ideas where necessary. This example is ideal and instructive for the concepts explained in this paper, given the variety of NLP models it encompasses, such as multi-class text classification, span detection-based question answering, and semantic text similarity-based retrieval of user-generated content such as user-generated frequently asked questions (FAQs) and user reviews for the product.

## 2 Related Work

We review the existing literature for chatbot evaluation metrics below.

### 2.1 Metrics for evaluating chatbot performance

While several metrics for evaluating chatbots have been suggested, (Abd-Alrazaq et al., 2020; Shawar and Atwell, 2007) we did not find any mathematical frameworks that relate chatbot effectiveness metrics to individual model performances. Typically metrics have been based on response generation (Cameron et al., 2019), usability (Abdullah et al., 2018), response understanding (Yokotani et al., 2018), and global aesthetics (Wargnier et al., 2018).

Other metrics proposed for chatbots such as perplexity, sensibleness and specificity average (SSA) (Adiwardana et al., 2020), and percentage of per-turn engaging responses (Xu et al., 2022) focus on how closely the bot-user conversation resembles a human conversation over multiple turns. While these metrics are crucial for a general, open-domain chatbot, most business applications measure the success of their conversational AI products based on not only the coherence within the chatbot responses but also the effectiveness of the chatbot in helping the user's specific needs. Metrics intended for open-domain chatbots may not always be appropriate for a business use case. General-purpose NLG metrics such as ROUGE (Lin, 2004) and BLEU (Papineni et al., 2001), despite having the benefit of being automated, do not work for cases where there could be multiple responses that are equally effective.

### 2.2 Typical model hierarchies in a chatbot

Popular chatbot frameworks such as Daniel et al. (2020); Bocklisch et al. (2017), and winning chatbots in competitions such as the Alexa Prize competition (Serban et al., 2017), demonstrate that superior chatbot designs comprise a collection of several models, each geared towards a specific kind of conversation. For instance, Serban et al. (2017) consists of 22 response models, thus making it crucial for a team of engineers to have clear visibility into how sensitive the overall chatbot metrics are to the metrics of each of the constituent models. As new use cases emerge and a chatbot grows in complexity, having a quantitative view of the contribution of each model to the overall chatbot performance is crucial.

Most open-source chatbot designs begin with an intent recognition layer that decides the category of the user query before directing it to an appropriate downstream model,(Adamopoulou and Moussiades, 2020; Lokman and Ameedeen, 2018) whereas downstream models could include question-answering and/or other information retrieval models.(Kulkarni et al., 2019)

## 3 Description of the chatbot architecture

As described in Fig. 1, the chatbot consists of an intent classification model, which detects the overall intent of the user query. If the intent is not a product-specific intent (e.g., stock availability, introductory greeting, etc.), then we answer using standard templates that do not involve any predictive model. On the other hand, if the intent is a product-specific intent, then we invoke a binary

classifier that predicts whether the query is factual or subjective.

Factual queries, such as "what is the battery capacity of this phone?", "does this phone support 5G?" are sent first to a question-answering model based on unstructured data such as product description text or structured data such as key-value specification pairs, e.g., "battery capacity: 5000 mAh", and "camera resolution: 48 megapixel".

Subjective queries, such as "is the camera good?," "can I play PUBG on this phone without lag?" are sent first to a semantic text similarity model that retrieves the most similar user FAQ or user reviews from the product webpage that can potentially answer this subjective user query.

We define the chatbot answerability metric as follows: for a given response by the chatbot to a user's question, we assign it a score between -1 and +1 based on the criteria described in Table 1.

Note that the weights assigned to each answer category in Table 1 could be modified as per product objectives. One example in an e-commerce setting could be to assign a weight proportional to the probability of conversion, i.e., the user buys the product they are consulting the chatbot about. The *answerability* of the chatbot is then defined as the mean of answerability scores over the entire dataset of chatbot responses.

We describe two kinds of modeling approaches in this paper:

- Section 5 describes a mathematical model for connecting individual model metrics to answerability.

- Section 6 describes a multi-class classification approach where we train a model to classify a query-response pair between one of the answerability classes.

## 4 Data

For the modeling described in this paper, we sampled actual user-chatbot conversations from the production logs of the pre-purchase chatbot and asked an in-house team of labelers, based in India and familiar with the English-Hindi code-mixed vocabulary used by the users of the pre-purchase chatbot, to label each chatbot response with one of the 7 labels described in Table 1. Labelers were provided with explicit examples of correct and incorrect labels in a detailed labeling guideline document, so that they label user query-bot response pairs consistently. Chatbot responses that did not fall into any of these 7 categories were not considered. Every data point was labeled by two labelers. Whenever there was a disagreement between the labels, the tie was broken by a third labeler. Labelers used Taskmonk, a data labeling platform specific to e-commerce applications[1]. User queries are either in English or code-mixed English-Hindi, written using the Roman script. Examples from labelled dataset are listed below:

1) Product: ABC mobile phone
**User question**: Auto call recording available
**Bot response**: (from FAQs) Here is what I found- Q: All call Auto call recording hogi kya[2] A: yes
**Label**: Excellent Relevance, Helpful Answer

2) Product: XYZ mobile phone
**User question**: earphone support this phone
**Bot response**: (from the product description) Here's what you'll get in the box: 1 Handset, 1 Battery, 1 Charger.
**Label**: Poor Relevance, Unhelpful Answer

In the first example, the chatbot returned a relevant user FAQ, whereas in the next example, the chatbot responded with information that was not relevant to the question asked by the user.

## 5 Modeling for predicting the answerability from individual model metrics

### 5.1 Framework to link answerability to individual model metrics

Now that we have defined the answerability metric, let us formulate our framework for linking individual model metrics to the answerability metric defined in Section 3. For the purposes of this paper, we shall use a simplified version of the chatbot to describe our approach and results. This chatbot is used for answering pre-purchase customer questions related to products listed on the e-commerce platform. The components of the chatbot cover the major categories of models typically used in chatbot architectures, hence it is an ideal example for illustrating our answerability framework.

[1] https://taskmonk.ai/about-us.html

[2] This code-mixed utterance translates to: "Will all calls be automatically recorded?"

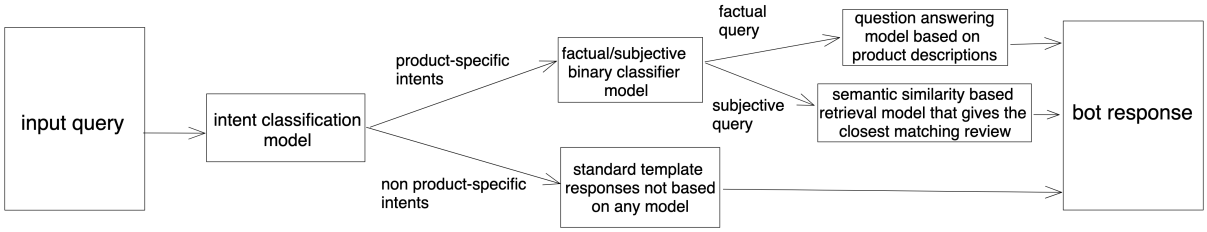


Figure 1: Schematic of a simplified version of the e-commerce chatbot used for the purposes of this paper.

| Type of response | Answerability score | Symbol |
|---|---|---|
| • Poor Relevance | -1 | $S_p$ |
| • No Answer<br>• Transfer to a Human Agent | 0 | $S_n$ |
| • Excellent Relevance, Unhelpful Answer<br>• Fair Relevance, Unhelpful Answer | 0.5 | $S_{r,\,uh}$ |
| • Excellent Relevance, Helpful Answer<br>• Fair Relevance, Helpful Answer | 1 | $S_{r,\,h}$ |

Table 1: Criteria for answerability scores based on the relevance and helpfulness of the chatbot responses

### 5.2 Intent classification model

Let $S = \{\text{ps, non-ps}_1\text{, non-ps}_2\text{, } \dots\}$ denote the list of chatbot intents, where "ps" indicates the "product-specific" intent, and "non-ps$_1$", "non-ps$_2$", etc. indicate the non-product specific intents for which there is no predictive model to be run downstream. For a text classification model, if the predicted label is correct, then the answerability will be the true answerability associated with that class of intent. Otherwise, we assume that the answerability will be 0.

Therefore, the contribution of intent $i \in S$ to the answerability is given as $f_i A_i P_i$, where $f_i$ denotes the fraction of queries with the predicted intent label being $i$, $A_i$ denotes the answerability associated with queries with intent label $i$, and $P_i$ denotes the probability of correctly predicting a query with a predicted intent label $i$ (precision).

Thus, the overall chatbot answerability $A$ is given as:

$$\begin{aligned} A &= \sum_{i \in S} f_i A_i P_i \\ &= f_{ps} P_{ps} A_{ps} + A_{non-ps}, \end{aligned} \tag{1}$$

where

$$A_{non-ps} = \sum_{i=1,2,\dots} f_{non-ps,i} P_{non-ps,i} A_{non-ps,i}. \tag{2}$$

Note that $\sum_{i \in S} f_i = 1$ here. When the intent $i$ is $ps$ (product-specific), we can expand the answerability in terms of downstream model metrics from the semantic text similarity, factual-subjective classifier, and question-answering models. However, when the intent is not the product-specific (ps) intent, there is no dependence of the answerability of that specific intent on any of the model metrics. Therefore, we can substitute the answerabilities of those intents, namely $\{A_{non-ps,i}\}$, with a constant, average answerability value $A_{non-ps}$, calculated from labeled data corresponding to the appropriate non-product specific intents, such as stock availability, offers and discounts, etc.

### 5.3 Factual/subjective classifier

Just like the intent classification model, the factual-subjective binary classifier, which is invoked for product-specific queries, contributes to the chatbot answerability in the following way:

$$\begin{aligned} A_{ps} = {} & f_{factual}\, P_{factual}\, A_{factual} \\ & + f_{subjective}\, P_{subjective}\, A_{subjective}. \end{aligned} \tag{3}$$

, where $P_{factual}$ and $P_{subjective}$ denote the precisions of the factual and subjective classes of the factual/subjective binary classifier, and $f_{factual}$ and $f_{subjective}$ denote the fraction of queries recognized as product-specific (ps) by the intent model.

### 5.4 Question-answering model

The question-answering models based on product features are called when the above-mentioned factual-subjective classifier predicts the user query

to the chatbot as factual. Thus, $A_{factual}$ mentioned in Eqn. 3 can be further expanded in terms of the metrics of the question-answering model. While answering from product specifications, the ground truth could be either null or non-null, depending on whether the answer to the question asked is actually available in the product specifications. Furthermore, for null or non-null ground truth, we could have either null or non-null predictions from the question-answering model. The question-answering model metrics we use are described in Table 2. We also assume that the 2 models for question-answering from unstructured product description text and structured key-value specification pairs have the same model metrics, thus effectively reducing the 2 models into a single question-answering model.

Combining Tables 2 and 1, we can derive the following formula for $A_{factual}$:

$$\begin{aligned} A_{factual} = {} & S_{r,\,h} \times C_{QnA} \times \rho \times f_{H,QnA} \\ & + S_{r,\,uh} \times C_{QnA} \times \rho \times (1 - f_{H,QnA}) \\ & + S_p \times (1 - C_{QnA}) \times \rho \\ & + S_p \times \mathrm{FPR}_{null} \times (1 - \rho) \end{aligned} \tag{4}$$

Here $S_{r,\,h}$, $S_{r,\,uh}$, and $S_p$ denote the answerability scores for relevant and helpful, relevant but unhelpful, and poor relevance answers respectively, as described in Table 1. Terms in Eqn. 4 with coefficients $S_{r,\,h}$, $S_{r,\,uh}$, and $S_p$ denote the contributions of relevant and helpful, relevant but unhelpful, and poorly relevant answers respectively to $A_{factual}$. We exclude terms with $S_n$ (no answer/agent transfer cases) from these equations for simplification purposes, because $S_n = 0$ according to Table 1.

Note that in case the question-answering model cannot answer a question, there is a fallback to semantic search-based retrieval models. However, in this formula, we ignore this in order to simplify our description.

### 5.5 Semantic-text similarity based retrieval models

Among the queries detected as subjective by the factual/subjective classifier, some queries are answered by a retrieval model that searches for the most relevant user review, whereas others are answered by a retrieval model that searches for the most relevant user FAQ. While there is a fallback on the question-answering model in case the retrieval models are unable to answer the user question, we chose to ignore it in order to simplify our modeling.

We now expand $A_{subjective}$ as

$$\begin{aligned} A_{subjective} = {} & f_{FAQ}\, A_{FAQ} \\ & + f_{Reviews}\, A_{Reviews}, \end{aligned} \tag{5}$$

where $A_{FAQ}$ and $A_{Reviews}$ are the answerabilities of the FAQ and Reviews models, and $f_{FAQ}$ and $f_{Reviews}$ denote the fraction of user queries that were detected as subjective answered by FAQ and reviews retrieval models respectively. For the case of retrieval models, we use model metrics as described in Table 3.

We further expand $A_{Reviews}$ and $A_{FAQ}$ in terms of the model metrics described in Table 3 as follows:

$$\begin{aligned} A_{Reviews} = {} & S_{r,\,h} \times P_{Reviews} \times C_{Reviews} \times f_{H,Reviews} \\ & + S_{r,\,uh} \times P_{Reviews} \times C_{Reviews} \times (1 - f_{H,Reviews}) \\ & + S_p \times (1 - P_{Reviews}) \times C_{Reviews} \end{aligned} \tag{6}$$

Similarly,

$$\begin{aligned} A_{FAQ} = {} & S_{r,\,h} \times P_{FAQ} \times C_{FAQ} \times f_{H,FAQ} \\ & + S_{r,\,uh} \times P_{FAQ} \times C_{FAQ} \times (1 - f_{H,FAQ}) \\ & + S_p \times (1 - P_{FAQ}) \times C_{FAQ}. \end{aligned} \tag{7}$$

### 5.6 Overall expression for the chatbot answerability $A$

By combining Eqns. 1, 3, 4, 5, 6, and 7, we get the expression for the overall chatbot answerability $A$. The approach we describe does not require any additional model training and can act as a simple, first-principles baseline for expressing $A$ as a function of the individual model metrics.

For a chatbot that is different from the pre-purchase e-commerce chatbot we describe here, we need to modify the expressions Eqns. 1, 3, 4, 5, 6, and 7 according to its specific architecture. For example, if a chatbot does not have access to user-generated content such as reviews or FAQs, we could ignore the terms $A_{FAQ}$ and $A_{Reviews}$. However, in most multi-model chatbot architectures, we should be able to derive similar expressions for an answerability-like metric.

By differentiating the overall expression for $A$ with respect to each of the model metrics, we get the *sensitivity* of $A$ to the product metric. For example, $\frac{\partial A}{\partial P_{FAQ}}$ tells us the sensitivity of $A$ to $P_{FAQ}$. By using our mathematical model, we could know

| Metric | Symbol | Description |
|---|---|---|
| Coverage | $C_{QnA}$ | fraction of non-null ground truth cases the model answered correctly |
| Answer rate | $\rho$ | fraction of queries for which answer is available in product descriptions |
| Helpful fraction | $f_{H,QnA}$ | fraction of answers that were helpful to the user |
| Null false positive rate | $\text{FPR}_{null}$ | fraction of false positives within null ground truth cases |

Table 2: Question-answering model metrics

| Metric | Symbol | Description |
|---|---|---|
| Coverage of the FAQ retrieval model | $C_{FAQ}$ | fraction of queries for which the FAQ retrieval model gave a non-null answer |
| Coverage of the reviews retrieval model | $C_{Reviews}$ | fraction of queries for which the reviews retrieval model gave a non-null answer |
| Precision of the FAQ retrieval model | $P_{FAQ}$ | fraction of non-null answers from the FAQ retrieval model which have excellent or fair relevance |
| Precision of the reviews retrieval model | $P_{Reviews}$ | fraction of non-null answers from the reviews retrieval model which have excellent or fair relevance |
| Helpful fraction of the FAQ retrieval model | $f_{H,FAQ}$ | fraction of helpful answers from the FAQ retrieval model |
| Helpful fraction of the reviews retrieval model | $f_{H,Reviews}$ | fraction of helpful answers from the reviews retrieval model |

Table 3: FAQ and reviews retrieval model metrics used for the answerability calculation

which metric from Tables 2 and 3 has the highest sensitivity of $A$, and based on this we could prioritize model improvements focused on that metric. This can be of immense help for complicated chatbot architectures where it is hard to accurately predict which model metric has the potential to have the maximum positive impact on the bottom-line business metric, such as answerability. Moreover, our framework could be used as a way to estimate the expected business impact before an improved model is launched into production.

To illustrate this, let us take the example of the answerability calculated by combining Eqns. 1, 3, 4, 5, 6, and 7, for the mobile phone product category. Let us hold all the other metrics to be constant, and change only the answer rate, $\rho$, and the precision of the subjective class of the factual/subjective binary classifier, $P_{subjective}$. According to the model, the overall answerability $A$ increases from 0.3256 to 0.3324 when $P_{subjective}$ goes up by 0.1, from 0.8 to 0.9, whereas $A$ increases from 0.3256 to 0.3467 when $\rho$ goes up by 0.1, from 0.5 to 0.6. This means that $\rho$ could be a better metric to invest in than $P_{subjective}$, given that it has a higher positive impact on $A$. However, in some cases, a normalized sensitivity, for example, $\frac{P_{FAQ}}{A}\frac{\partial A}{\partial P_{FAQ}}$, might be a more appropriate measure.

### 5.7 Limitations of this approach

Our approach makes the following assumptions, which could result in an inaccurate prediction of the chatbot answerability:

- We assume that if the intent is wrongly predicted or the factual/subjective classifier misclassifies the user query, the answerability is going to be 0, which is not necessarily true.
- We assume fractions such as $f_{ps}$, $f_{subjective}$, and $f_{H,QnA}$ to be constant and not a function of model metrics. In reality, as model metrics change, these fractions will change too.
- We ignore the possibility that the chatbot has a fallback to reviews/FAQ models when the question-answering model cannot answer, and vice-versa.

A query-wise answerability score prediction model that predicts an answerability score for a user query-bot response pair can help address these limitations.

The overall answerability $A$ is then defined as the average of the model predictions of answerability scores over the test dataset. However, a query-wise answerability prediction model relates an individual query-response pair to the answerability score, rather than connecting the model metrics to the overall chatbot answerability as in Section 5. Thus, the formula-based approach described in this section should be used in cases where we wish to get a rough estimate of how much a particular metric improvement is expected to increase the chatbot answerability, or know the sensitivity of a business-motivated metric such as $A$ to the product metrics. Whereas the per-query approach should be used when the goal is to get an accurate prediction of the overall answerability $A$.

## 6 Per-query predictive model for the chatbot answerability $A$

Apart from helping us understand the relative importance of each constituent model used in a chatbot, the answerability labels described in Table 1 can also be used for training a model to predict the relevance and helpfulness of chatbot response, which in turn can be used to compute the answerability directly. Also, detecting whether a chatbot response is helpful or not can be used to modify our planned response so that an originally unhelpful response could potentially become helpful. For example, if the model predicts that an answer is not helpful, then we could provide recommendations of similar products, or alter the conversational design in a way that helps the user. Moreover, as described in Section 5.7, such a per-query modeling approach does not suffer from the limitations of the formula-based approach described in Section 5.

### 6.1 Approach

We propose to model helpful/unhelpful answer prediction as a multi-class classification task at the query level by using the question and its corresponding response from the chatbot as the input. We chose an in-house Large Language Model (LLM) based on the BERT architecture (Devlin et al., 2019) pre-trained on an approximately 50 GB in-house training corpus consisting of product descriptions, catalog attributes, reviews, QnA pairs, and addresses as our pre-trained model. The maximum sequence length while training is limited to 192 based on the distribution of the number of tokens. This model has 12 Encoder layers with an embedding size of 768. This model is trained on 3 A100 GPUs for 14 days with a batch size of 420 for 1M steps. This model gains significantly lower perplexity on in-domain test sets, especially for code-mixed data and noisy search queries. We fine-tune this pre-trained model on the dataset described in Table 4.

Let $a^i$ be the response given by the chatbot for question $q^i$. The question $q^i$ and the response $a^i$ are concatenated, tokenized and passed to an embedding layer. The word embeddings along with their positional signals are passed to a transformer encoder, whose head predicts the output probabilities.

$$\begin{aligned} w^i &= \text{tokenizer}([q^i; a^i]) \\ \hat{y}^i &= \text{BertClassifier}(w^i) \end{aligned} \tag{8}$$

";" denote the appropriate concatenation of input sentences as required by the pre-trained model, i.e, the [SEP] token. The classification task is trained to minimize the cross entropy loss,

$$L_{nsp} = -\frac{1}{N_1}\sum_{i=1}^{N_1} y^i log\hat{y}^i \tag{9}$$

where $y^i$ is the ground truth label indicating the answer class and $N_1$ refers to the number of data points in the dataset.

### 6.2 Dataset

We train the multi-class answer classification model using our in-house dataset consisting of answered queries from mobile phone and fashion product categories on the e-commerce platform (see Section 4). We remove queries falling under "No Answer" groups since they are unhelpful by default. We group the remaining responses into 3 classes based on their corresponding labels. The statistics of the dataset are presented in Table 4.

### 6.3 Results

We use Term Frequency - Inverse Document Frequency (TF-IDF) scores to vectorize user queries and chatbot responses before feeding them as input to one-vs-all Logistic Regression (LR). This method was used as the baseline for this task. We also experimented with the publicly available BERT model (*bert-base-cased*) for the dataset. Table 5 shows the comparison results. Our in-domain BERT-based classification method outperforms the simple baseline (TF-IDF + LR) by a significant

| | | Train dataset | Test dataset |
|---|---|---|---|
| **Class 1 - Poor Relevance Unhelpful** | Poor relevance | 2652 | 278 |
| **Class 2 - Excellent/Fair Relevance Unhelpful** | Fair relevance | 428 | 44 |
| | Excellent relevance | 4661 | 479 |
| | Total datapoints | 5089 | 523 |
| **Class 3 - Excellent/Fair Relevance Helpful** | Fair relevance | 1387 | 156 |
| | Excellent relevance | 10197 | 1043 |
| | Total datapoints | 11584 | 1199 |

Table 4: Helpful/Unhelpful answers dataset

| Model | Precision | Recall | F1-score |
|---|---|---|---|
| TF-IDF + LR | 0.745 | 0.754 | 0.737 |
| open-domain BERT | 0.794 | 0.799 | 0.795 |
| in-domain BERT | 0.824 | 0.826 | 0.825 |

Table 5: Comparison of different models. Note that the precision, recall, and F1 scores indicate the weighted precision, recall, and F1 scores respectively.

| Source | Mobile | Fashion |
|---|---|---|
| Ground Truth | 0.546 | 0.637 |
| BERT classifier | 0.559 | 0.662 |
| Mathematical formulation (Section 5) | 0.326 | 0.308 |

Table 6: Comparing the overall chatbot answerability $A$ for the BERT-based classifier and the mathematical formulation from Section 5. The columns "Mobile" and "Fashion" indicate mobile phone and fashion product categories on the e-commerce platform respectively.

margin. It also achieves an improvement of **3.77%** on weighted F1 score over the public BERT model. The above result underlines the effectiveness of in-domain pre-training of BERT. The detailed classification report of our model is presented in Figure 2.

### 6.4 Computing Answerability

We use the trained model to compute the product-specific answerability $A_{ps}$ on the test dataset using model predictions. We choose to focus on $A_{ps}$ rather than the overall chatbot answerability $A$ to simplify our description, and also because $A_{ps}$ includes all the models present in the chatbot architecture described in Section 3 except the intent classification model. This is because the dataset consist of queries where the intent has been identified as product specification related. In order to compare the approaches described in Sections 5 and 6, we also compute an estimate of $A_{ps}$ using the mathematical formulation in Section 5 and compare the scores with the ground truth $A_{ps}$ from the human-annotated test dataset. The results are tabulated in Table 6. We observe that the BERT classifier is able to match the ground truth answerability scores closely.

For the mathematical formulation described in Section 5, the predicted answerability underestimates the ground truth answerability. This could be due to distribution shifts between the evaluation datasets used for calculating the model metrics versus the test dataset used in Table 6, along with the assumptions made by the mathematical model listed in Section 5.7. Given that the test dataset used here ignores cases where the chatbot gave a null response or transferred to a human agent, we normalized the answerability appropriately by a normalizing factor. Also, for all calculations with the mathematical formulation, the fractions in Eqns. 3, 4, 5, 6, and 7 such as $f_{subjective}$ and $f_{H,QnA}$ were calculated from the test dataset. To simplify our description further through binary formulation, we derived binary labels from the test dataset where answerability score for helpful and unhelpful is set as 1 and 0 respectively. We then compute the answerability scores as per the mathematical model described in Section 5, by choosing $S_{r,\,uh} = S_p = 0$ and $S_{r,\,h} = 1$ in Table 1. For this case, we get answerability scores of 0.469 and 0.442 for mobile phone and fashion product categories respectively. These scores are closer to the respective ground truth answerability scores of 0.599 and 0.6 calculated for this binary formulation of the answerability metric. This suggests that the mathematical formulation of Section 5 shows better agreement with the ground truth answerability scores when we assume answerability to take a binary value of either 0 or 1.

## 7 Conclusion

In this paper, we introduce answerability as a global chatbot effectiveness metric and show how it can be used to guide model development decisions for a

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| Excellent/Fair Relevance - Helpful Answer | 0.875 | 0.881 | 0.878 | 1199 |
| Excellent/Fair Relevance - Unhelpful Answer | 0.788 | 0.811 | 0.799 | 523 |
| Poor Relevance - Unhelpful Answer | 0.675 | 0.619 | 0.645 | 278 |
| accuracy | | | 0.826 | 2000 |
| macro avg | 0.779 | 0.770 | 0.774 | 2000 |
| weighted avg | 0.824 | 0.826 | 0.825 | 2000 |

Figure 2: Classification report of the in-domain BERT classifier

conversational AI product such as the pre-purchase chatbot, by relating answerability to all the metrics of all the models that are a part of the chatbot. Our framework is general and can be easily extended to chatbot metrics other than answerability depending on the domain of application, be it in finance, governance, or health care, as long as there is a concept of helpfulness associated with the chatbot's responses. For example, a health care chatbot helping patients understand their medical symptoms and pointing them to an appropriate health care provider needs to not only provide accurate information but also guide patients in the correct direction when such information is not available. The answerability metric will directly apply to such a case, and help guide the development of individual models within the chatbot's architecture in a way that maximizes patient satisfaction.

Future work could involve the joint training of all the models within a chatbot with a differentiable version of the answerability objective. Further iterations of the formula-based modeling approach described in Section 5 could involve the inclusion of other upstream models such as spell checking, automated speech recognition, and machine translation, which are used to interpret voice/multilingual user input before the input is sent to the intent classification model in the chatbot. We hope that the answerability metric and the modeling methods described in this paper will help guide product development and model prioritization in conversational AI products in the academic, government and industrial domains.

## 8 Limitations and Ethical Impact

The answerability metric could inspire other business-oriented metrics and also drive the development of task-oriented chatbots across various domains such as e-commerce, health care, and governance. These use cases could have various social implications: dialog systems such as customer support bots could bring in benefits such as cost savings, convenience, and the availability of 24-hour assistance, while decreasing the number of job opportunities for human service agents and salespersons. Language models underlying such dialog systems could reinforce social biases and impact the environment negatively (Bender et al., 2021; Schramowski et al., 2022). Moreover, any widely used metric or benchmark carries the inherent risk of biasing the research in a certain direction.

## References


Alaa Abd-Alrazaq, Zeineb Safi, Mohannad Alajlani, Jim Warren, Mowafa Househ, Kerstin Denecke, et al. 2020. Technical metrics used to evaluate health care chatbots: scoping review. *Journal of medical Internet research*, 22(6):e18301.

Abu S Abdullah, Stephan Gaehde, and Tim Bickmore. 2018. A tablet based embodied conversational agent to promote smoking cessation among veterans: A feasibility study. *Journal of Epidemiology and Global Health*, 8:225–230.

Eleni Adamopoulou and Lefteris Moussiades. 2020. An overview of chatbot technology. In *Artificial Intelligence Applications and Innovations*, pages 373–383, Cham. Springer International Publishing.

Tosin Adewumi, Foteini Liwicki, and Marcus Liwicki. 2022. State-of-the-art in open-domain conversational ai: A survey. *Information*, 13(6).

Daniel Adiwardana, Minh-Thang Luong, David R. So, Jamie Hall, Noah Fiedel, Romal Thoppilan, Zi Yang, Apoorv Kulshreshtha, Gaurav Nemade, Yifeng Lu, and Quoc V. Le. 2020. Towards a human-like open-domain chatbot.

Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '21, page 610–623, New York, NY, USA. Association for Computing Machinery.

Tom Bocklisch, Joey Faulkner, Nick Pawlowski, and Alan Nichol. 2017. Rasa: Open source language understanding and dialogue management.

Gillian Cameron, David Cameron, Gavin Megaw, Raymond Bond, Maurice Mulvenna, Siobhan O'Neill, Cherie Armour, and Michael McTear. 2019. Assessing the usability of a chatbot for mental health care. In *Internet Science*, pages 121–132, Cham. Springer International Publishing.

Gwendal Daniel, Jordi Cabot, Laurent Deruelle, and Mustapha Derras. 2020. Xatkit: A multimodal low-code chatbot development framework. *IEEE Access*, 8:15332–15346.

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

Mojtaba Komeili, Kurt Shuster, and Jason Weston. 2021. Internet-augmented dialogue generation. *CoRR*, abs/2107.07566.

Ashish Kulkarni, Kartik Mehta, Shweta Garg, Vidit Bansal, Nikhil Rasiwasia, and Srinivasan Sengamedu. 2019. Productqna: Answering user questions on e-commerce product pages. In *Companion Proceedings of The 2019 World Wide Web Conference*, WWW '19, page 354–360, New York, NY, USA. Association for Computing Machinery.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Abbas Saliimi Lokman and Mohamed Ariff Ameedeen. 2018. Modern chatbot systems: A technical review. In *Proceedings of the future technologies conference*, pages 1012–1023. Springer.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2001. BLEU. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics.

Patrick Schramowski, Cigdem Turan, Nico Andersen, Constantin A. Rothkopf, and Kristian Kersting. 2022. Large pre-trained language models contain human-like biases of what is right and wrong to do. *Nature Machine Intelligence*, 4(3):258–268.

Iulian V. Serban, Chinnadhurai Sankar, Mathieu Germain, Saizheng Zhang, Zhouhan Lin, Sandeep Subramanian, Taesup Kim, Michael Pieper, Sarath Chandar, Nan Rosemary Ke, Sai Rajeshwar, Alexandre de Brebisson, Jose M. R. Sotelo, Dendi Suhubdy, Vincent Michalski, Alexandre Nguyen, Joelle Pineau, and Yoshua Bengio. 2017. A deep reinforcement learning chatbot.

Bayan Abu Shawar and Eric Atwell. 2007. Different measurement metrics to evaluate a chatbot system. In *Proceedings of the workshop on bridging the gap: Academic and industrial research in dialog technologies*, pages 89–96.

Pierre Wargnier, Samuel Benveniste, Pierre Jouvelot, and Anne-Sophie Rigaud. 2018. Usability assessment of interaction management support in LOUISE, an ECA-based user interface for elders with cognitive impairment. *Technol. Disabil.*, 30(3):105–126.

Jing Xu, Arthur Szlam, and Jason Weston. 2022. Beyond goldfish memory: Long-term open-domain conversation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics.

Kenji Yokotani, Gen Takagi, and Kobun Wakashima. 2018. Advantages of virtual agents over clinical psychologists during comprehensive mental health interviews using a mixed methods design. *Computers in Human Behavior*, 85:135–145.