Predicting Interaction Quality of Conversational Assistants With Spoken Language Understanding Model Confidences

Yue Gao* †
University of Wisconsin-Madison
Madison, United States
ygao266@wisc.edu

Enrico Piovano*
Amazon Alexa AI
Berlin, Germany
piovano@amazon.de

Tamer Soliman*
Amazon Alexa AI
Sunnyvale, United States
tsoliman@amazon.com

Monir Moniruzzaman
Amazon Alexa AI
Seattle, United States
monirzm@amazon.com

Anoop Kumar
Amazon Alexa AI
Seattle, United States
anooamzn@amazon.com

Melanie Bradford
Amazon Alexa AI
Berlin, Germany
neunerm@amazon.de

Subhrangshu Nandi ‡
Amazon Alexa AI
Seattle, United States
subhrn@amazon.com

ABSTRACT

In conversational AI assistants, SLU models are part of a complex pipeline composed of several modules working in harmony. Hence, an update to the SLU model needs to ensure improvements not only in the model specific metrics but also in the overall conversational assistant. Specifically, the impact on user interaction quality metrics must be factored in, while integrating interactions with distal modules upstream and downstream of the SLU component. We develop a ML model that makes it possible to gauge the interaction quality metrics due to SLU model changes before a production launch. The proposed model is a multi-modal transformer with a gated mechanism that conditions on text embeddings, output of a BERT model pre-trained on conversational data, and the hypotheses of the SLU classifiers with the corresponding confidence scores. We show that the proposed model predicts aggregate defect metrics with more than 76% correlation with the measured ones, compared to 46% baseline.

CCS CONCEPTS

• **Computing methodologies** → **Natural language processing**; **Machine learning**.

KEYWORDS

spoken language understanding, dialog response quality, defect prediction, transformer-based model

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
CIKM '23, October 21–25, 2023, Birmingham, United Kingdom
© 2023 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-0124-5/23/10.
https://doi.org/10.1145/3583780.3615493

ACM Reference Format:
Yue Gao* †, Enrico Piovano*, Tamer Soliman*, Monir Moniruzzaman, Anoop Kumar, Melanie Bradford, and Subhrangshu Nandi ‡. 2023. Predicting Interaction Quality of Conversational Assistants With Spoken Language Understanding Model Confidences. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management (CIKM '23), October 21–25, 2023, Birmingham, United Kingdom.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3583780.3615493

1 INTRODUCTION

The interactive speech interface of Conversational assistants like Alexa, Siri, Cortana and Google Assistant has offered humans a more naturalistic experience of voice-controlling their environments. As their adoption grows, these intelligent systems require fast iterative updates to co-evolve with the ever changing variations of human ambiance and interaction milieu. Owing to the modular architecture of these agents, updates are typically introduced asynchronously, by a team of developers with focal skillset, to only one of the models making up the agent's underlying pipeline. This can be the Automatic Speech Recognition (ASR) model that transcribes the voice prompt into textual request, the Spoken Language Understanding (SLU) model that maps the request to an intent and named entities, or the response generator model which generates the system response on the basis of the SLU mappings.

In this work, we will focus on SLU model updates and their impact on the user interaction quality (IQ) [8, 19], or equivalently on the IQ defect rate of the overall conversational assistant, i.e. the fraction of unsatisfactory assistant's responses. SLU is on the core models of a conversational assistant and SLU model changes can significantly affect the end-to-end performance, as for instance an SLU error may unfold in a incorrect answer to the user. SLU model updates may involve improving the underlying ML models or adding new intents or labels for new functionalities. Note that SLU model updates only involve deploying to production a

* Equal contribution
† Work conducted during internship at Amazon Alexa AI
‡ Core contribution team leadership

new SLU model, while the upstream and downstream components are not changed. Before deploying a new SLU model, evaluations must be performed to understand its impact on the overall system performance, specifically on the IQ. However, performing such evaluations is extremely challenging as: 1) the user IQ cannot be computed before production as it would require a conversational agent (e.g. a human) to interact with the system and try all possible dialogues; 2) testing the SLU model standalone against a test set of labelled data would not provide any indication on how the model will interact with the other downstream and upstream components. For instance, the new SLU model may improve over the current one when tested standalone, but a change in the model output distribution (for instance a change in the confidence score distribution) may adversely affect the system's performance when integrating the SLU model in the overall pipeline and in turn increase the IQ defect rate [16]. This may happen as, for instance, the downstream components may have been calibrated towards the previous SLU model outputs, hence leading to a performance loss with the new SLU model distribution.

In this work, we develop a machine learning model, denoted as pSLIQ (**p**redicting **I**nteraction **Q**uality with **SL**U scores), to predict the IQ defect rate by solely considering SLU related features, such as the upstream ASR transcription and the corresponding confidence score, and the top n SLU interpretations with the corresponding confidence scores. Those are the only features available to SLU developers when evaluating the new SLU model before release into production. pSLIQ assesses the impact of a new SLU model on the system's response qualities, i.e. IQ defect rate, before deploying the model in production. By leveraging a transformer architecture [26], pSLIQ is able to learn the relationship and interactions between SLU feature distributions and the other system components. Specifically, pSLIQ is a multi-modal transformer where the text embeddings through Bidirectional Encoder Representations from Transformers (BERT) [5] are combined together with categorical and numerical features via a gating mechanism inspired by [18], and a head feed-forward network is then used for defect classification. To maximize the performance, we first pre-train the BERT model with historical conversational data.

To the best of our knowledge, this is in fact the first proposed model to *predict* end-to-end dialogue interaction quality before model deployment that could result due to SLU changes. As we will see in Section 2, previous works either focused on computing the assistant IQ defect rate by evaluating already existing dialogues [1, 2, 8, 12, 13, 15, 20, 21, 24] or, provided an already existing IQ defect, to root-cause the failing component [3, 9, 23]. Differently, in our use case, we are not aiming to measure but rather to predict the IQ defect rate, utilizing SLU features only, i.e. the same features available to developers when they perform SLU model evaluation before deployment. To remark, note that the features used by pSLIQ only account for a very limited subset of the whole feature set available by the previously mentioned works [1–3, 8, 9, 12, 13, 15, 20, 21, 23, 24], where the latter can be obtained by the runtime logs and used for measuring or root-causing IQ defects.

Importantly, predicting the impact of a new SLU model ahead of deployment is the main application of pSLIQ. This enables developers to assess end-to-end performance changes of the conversational assistant without having to run expensive and time-consuming A/B experiments.

We will present extensive results to show the performance of pSLIQ in a commercial conversational assistant. Specifically, we will show the pSLIQ can achieve an AUC of 81%, an Accuracy of 82%, an F1 score of 61% and recall of 81% in predicting an IQ failure (or defect) over a defect-recorded dataset of user requests (note that high recall is important here as it allows to identify requests which will not be correctly handled); in addition we assess pSLIQ ability to predict the aggregated defect metrics before new SLU model deployments and we show that the predicted IQ defect rate aggregate at intent level before an A/B experiment has more than 76% Spearman and Pearson correlation with the corresponding metrics measured during the A/B experiment. We also show several ablation studies to highlight the importance of pre-training the BERT model with conversational data to enhance pSLIQ performance, as well as the importance of including the utterance transcribed text together with the SLU interpretations.

1.1 SLU model description

In this paper, we consider the conversational assistant architecture as in [9, 17, 22]. When a user makes a request, it is first transcribed by the ASR model which provides the text together with the corresponding ASR model confidence score. The transcribed text is then input to the SLU model, which classifies the text to a specific domain (domain classifier or DC), intent (intent classifier or IC) and extract the labels (named or label entity recognition or NER). The NER maps each token to a specific label, where the labels are relevant to the intent, and classifies the non relevant ones as "other". The non-"other" label values are denoted as entities. For instance, the utterance "play madonna" is classified in Domain=Music, Intent=PlayMusic and NER=("play:other Madonna:ArtistName"), and the entity is Madonna. The output of the SLU model contains the top n hypotheses in decreasing order of confidence scores, and those are sent to the response generator. In this work we consider $n = 5$ [10]. For each hypothesis, the SLU model also provides the individual confidence scores for DC, IC and NER. Note that the overall hypothesis confidence score for each interpretation may differ from the corresponding product of the DC, IC and NER as re-scoring and re-ranking mechanisms may be applied within the SLU model, as well as external signals for instance depending on the device where the assistant is integrated ("play madonna" may be used to play a song in a smart speaker and to play a video on a tv) [25, 27]. Note that re-ranking, re-scoring and device features will also be used to train pSLIQ. The top n hypotheses, together with the utterance text, are then processed by the response generator to resolve the entities and reply back to the user. The description of the architecture is summarized in Fig 1.

1.2 User Interaction Quality

In a conversational assistant, an IQ defect is any response that is not what the user wanted, specifically where the conversational assistant either does not understand the question or the action requested by the user, or it attempts to respond to the user request but does so incorrectly or unsatisfactorily. In the literature, several works

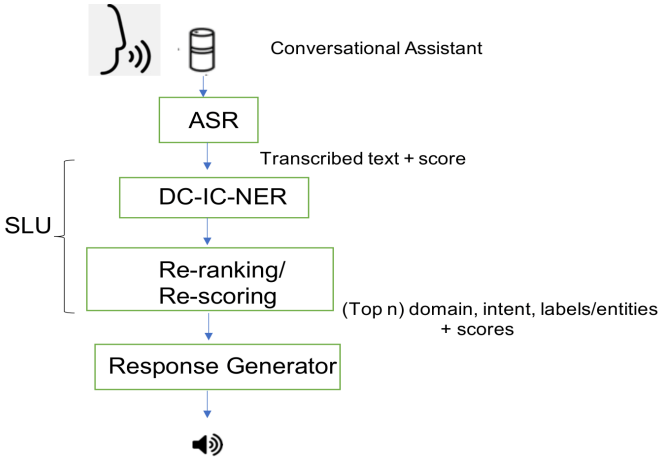

Figure 1: Architecture of a conversational assistant

have contributed to the research on evaluating the quality of responses in conversational systems, for instance using word-overlap metrics like BLUE and ROUGE [12, 15] or user sentiment analysis approaches [1, 2, 8, 13, 20, 21, 24]. In this work, we will consider the user's IQ defect by considering the evaluation framework proposed here [8], as shown to be a robust indication of a conversational assistant's performance. The IQ defect in [8] classifies the data in defect and non-defect based on the underlying dialog between the user and the assistant and defects are detected when the assistant cannot respond, in case of user barge-in or negative feedback, user paraphrasing, or delayed responses [8]. *However, note that pSLIQ can be trained and applied to any defect and the one in [8] has been just chosen as a use case.*

2 RELATED WORK

In addition of the IQ defect measurement [1, 2, 8, 12, 13, 15, 20, 21, 24] described in Section 1.2, several works in the literature have focused their attention in providing insights in the root-cause of a defect, for instance in the works in [3, 23] transformer-based models were built to detect SLU domain or intent classification errors using confidence scores, while the work in [9] proposed a transformer model to find the failure point in a conversational assistant (ASR, SLU, etc). However, all these contributions focus on monitoring the performance on already existing dialogs, with all the recorded logs and information of all the system components, and not on predicting the IQ due to system changes, especially, of a core component like SLU. This is key difference with pSLIQ, where the opposite is applied i.e. the IQ defect rate is predicted from the SLU features before deployment instead of looking at already existing IQ defects and check if they were caused by the SLU model.

On the other hand, the problem of better evaluating ML model performance over labelled test sets has been tackled in several works, which include several techniques as sample selection bias correction [4] or acquisition through active testing to remove bias and reduce variance [6, 11, 14]. A method directly applicable to an SLU model in a conversational assistant can be found in [16], where a novel methodology was developed to re-weight the accuracy metrics over test sets and align them with the real performance in production by offsetting the discrepancy in distribution between the offline test set and the real traffic. However, all these methods consider and compute metrics on the SLU model standalone and they do not take into consideration the complex interaction between SLU and its upstream and downstream components in a conversational assistant.

Unlike the previous works, our focus is to forecast in advance the quality of responses due to a SLU model change before the model is integrated with the other system's components in production. Differently from the previous works, we develop a model that on one side only leverage SLU features, but on the other side learns the interactions between the SLU model features and the other system components which cannot be captured by testing the SLU model standalone.

3 METHODOLOGY

In this section we describe the pSLIQ design. We start with the training and test datasets of pSLIQ in Section 3.1. We then describe the evaluation process on a A/B test in Section 3.2. Next, we describe the features used by pSLIQ in Section 3.3. We finally present the details of the transformer-based pSLIQ architecture in Section 3.4.

3.1 Training Set

We have pulled de-identified traffic data across several SLU model releases of a conversational assistant and split in train, validation and test using a 75/5/20 schema, leading to a training and test set of 11.5 and 3.1 millions utterances, respectively, across several domains, intents and labels. The pulled data contain the runtime logs information of all the assistant components of when the data was processed by the assistant. For each training data, we extracted the corresponding IQ defect (if defect or not) as well as the SLU related features such as the top 5 interpretations [10] with the corresponding confidence scores, both overall and for each component DC, IC, NER, internal SLU specific signals for re-ranking and re-scoring as well as the device of the assistant, the ASR transcribed

utterance text and related ASR model score. Note that the IQ defect was computed real-time by the assistant when the request was made, i.e. "automatically" annotated by the system and no human intervention was required (using the method in [8] described in Section 1.2).

3.2 A/B experiment test set

In addition of evaluating the accuracy metrics over the test set in Section 3.1, we have assessed pSLIQ's ability to predict defects due to a new SLU model before deployment. To this purpose, we have leveraged an A/B test for the same conversational assistant. Before the A/B test, we did in order a) pulled a random sample of the traffic (prior of the A/B), b) input the pulled traffic through the new SLU model, c) used the SLU output features as input for pSLIQ for inference; d) aggregated the pSLIQ defect prediction results at intent level, i.e. the predicted defect for each intent. We have then correlated the aggregate prediction at intent level with the measured IQ defect during the A/B test. To remark, we ran inference with traffic before the A/B experiment to avoid any data leakage.

3.3 Features

We train the pSLIQ model on the following features:

(1) Numerical Features: The *SLU top 5 hypotheses confidence scores*, including individual scores for DC, IC and NER; the ASR score for the transcribed text.
(2) Categorical Features: The *SLU top 5 hypotheses* as well as SLU specific internal signals as re-ranking, re-scoring and the device where the assistant is placed.
(3) The transcribed utterance text given by the ASR output.

3.4 Architecture of pSLIQ

We use a multi-modal Transformer-based model (see Figure 2) to combine the different features and compute defect prediction. Categorical, numerical, and text features are included separately, and for text, a 'bert-base-uncased' BERT architecture [1] [5] (but pre-trained on conversational data as we will see in Section 3.4.2) is used to generate the embeddings. The features are then combined via a gating layer to produce the multi-modal representation feeded into a two-layers fully connected feed-forward network multilayer perceptron (MLP) to produce defect prediction. The architecture was inspired by [9] where text embeddings are also separately combined with numerical and categorical features. While the purpose was slightly different (root-cause the failing component for an IQ defect), the same architecture also benefits our use case as the main difference is that we need to encode and optimize on a smaller number of features.

3.4.1 Gating Layer to combine features. The features are combined using a gating mechanism inspired from [18]. Let x denote the output of BERT model that represents the text feature, c denote the categorical features, and v denote the numeric features. The goal is to use some weight matrices W, bias vectors b and activation function σ to process the features x, c, v and get a combined representation as the multi-modal feature combination m. To get a good representation of the feature combination m, we consider a Multimodal Adaptation Gate (MAG) [18], which takes the summation of linear transformed tabular features gated by the text features (see the Gating Layer in Figure 2). First of all, we get the gating vectors for the categorical features and the numerical features respectively:

$$\begin{aligned} g_c &= \sigma(W_{g_c}[c, x] + b_c); \\ g_v &= \sigma(W_{g_v}[v, x] + b_v). \end{aligned} \tag{1}$$

After getting the gating vectors g_c, g_v, we could represent the non-verbal features c and v as a displacement vector h:

$$h = g_c \cdot (W_c c) + g_v \cdot (W_v v) + b_h; \tag{2}$$

Subsequently we can get the combined representation m by taking a weighted summation on x and h:

$$m = x + \alpha h, \tag{3}$$

where $\alpha = \min(\beta|x|/|h|, 1) \leq 1$ and β is a hyperparameter that can be chosen from cross validation. Note that we have decided to use the Gating Multimodal Adaption Gate (MAG) layer in [18] as we are combining different sources, i.e. transcribed text, SLU interpretations and scores. The core philosophy behind this is that the non-verbal behaviors can have an impact on the interaction quality of dialogues.

3.4.2 Two-stage training. The pSLIQ model is trained in two stages. First, the BERT model is pre-trained with conversational assistant de-identified historical data on a Masked Language Model (MLM) loss. On the other hand, the gating layer and the classifier head are initialized randomly. Then, in the second stage of training, we fine-tune the whole network and the classification head is used to compute the corresponding cross-entropy loss and update the network parameters, while still fine-tuning the BERT model to better adapt it to the specific task. For training, we have used an AWS EC2 instance 3.8xlarge, leading to a training time of approximately 8 hours for 5 epochs, while the inference time was approximately 45 minutes. Note that we tried to train the model for additional epochs without seeing performance benefits, due to the limited number of features and length of text size (commands to conversational assistants), which limit the maximum learnable information from those and thus the needed training.

4 RESULTS

We report here the results of the pSLIQ model on a commercial conversational assistant. First, we report in Section 4.1 the prediction performance of pSLIQ model on the test set described in Section 3.1. Second, we perform ablation studies: a) in Section 4.2 to analyze the effect of pre-training of the BERT model, b) in Section 4.3 to analyze the effect of text feature factorization. Finally, in Section 4.4, we compute the correlation between the predicted and measured aggregated defect at intent level in an A/B test.

4.1 Defect prediction Performance of pSLIQ

Table 1 shows the performance of pSLIQ. To benchmark pSLIQ, we have considered as baseline the variant where the BERT model is pre-trained over conversational data in the first stage of training but no further fine-tuned during the second stage of training (see Section 3.4.2 for details). Moreover, for the baseline model we have

[1] https://huggingface.co/tftransformers/bert-base-uncased

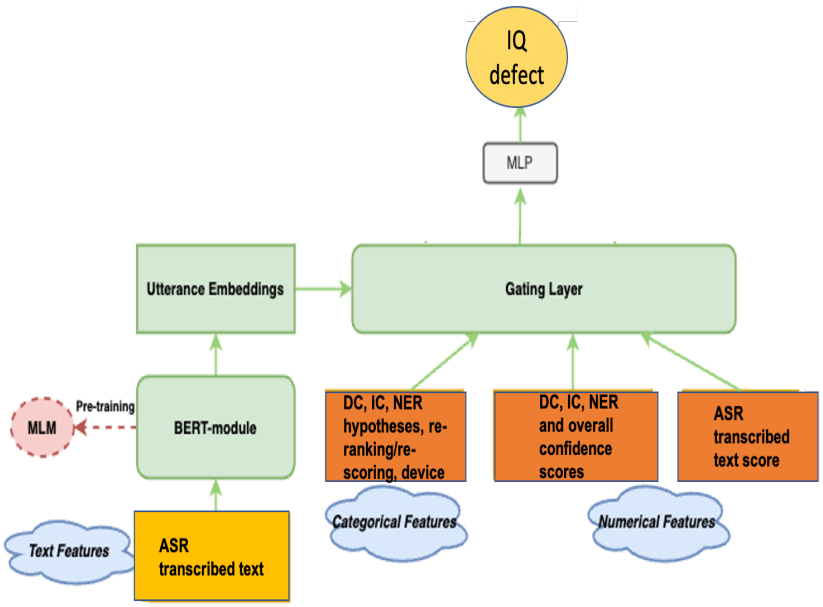

Figure 2: Architecture of pSLIQ (predicting Interaction Quality with SLU scores)

replaced the feed-forward head with the DeepFM structure in [7], widely used for classification tasks in recommendation systems, since we found in our experiments that DeepFM provides better performance when the BERT parameters are frozen in the second stage of training. We can see in Table 1 that fine-tuning BERT in the second stage of training allows to obtain gains across all metrics.

	Baseline	pSLIQ	Diff. (%)
AUC	0.69	**0.81**	+17.4%
F1	0.56	**0.61**	+8.9%
Precision	0.44	**0.49**	+11.4%
Recall	0.79	**0.81**	+2.5%
Accuracy	0.81	**0.82**	+1.2%

Table 1: Model performance for the defect classification task when the BERT module is fine-tuned vs. not fine-tuned in the second stage of training.

4.2 Effect of Pre-training BERT on conversational data

Table 2 compares the metrics of the pSLIQ model, where the BERT module is pre-trained through MLM head with historical conversational data, with the metrics obtained by considering the variant where the common BERT module is used, pre-trained with uncased English language from BookCorpus and English Wikipedia ('bert-base-uncased' in [5]). We can see that pre-training the BERT module with conversational data largely improves the defect prediction performance as pSLIQ can better adapt to the user requests made to a conversational assistant, and those data are significantly different compared to books and wikipedia.

	bert-base-uncased	pSLIQ	Diff. (%)
AUC	0.74	**0.81**	+9.5%
F1	0.49	**0.61**	+24.5%
Precision	0.37	**0.49**	+32.4%
Recall	0.72	**0.81**	+12.5%
Accuracy	0.52	**0.82**	+57.7%

Table 2: Model performance for the defect classification task by pre-training the BERT model with different training sets.

4.3 Importance of the text feature and its decomposition

An utterance text can be decomposed in carrier phrase and entities. The carrier phrase is obtained by replacing in the text all the tokens for which the NER provides a non-"other" classification with the corresponding label. For instance, for the utterance "play Madonna", where NER=("play:other Madonna:ArtistName"), the corresponding carrier phrase is "play ArtistName", and the separate entity itself is "Madonna" (see Section 1.1).

We have investigated the impact of carrier phrase and entities on the defect. The need to understand this decomposition arises as defects may be due to the utterance shape, as some shapes are more difficult to handle, or from the entities, as newer ones may not be included in the system's internal catalogs. Given the different inputs, we have used the 'bert-base-uncased' model in Section 4.2 for the BERT module, pre-trained with uncased English language but fine-tuned in the second stage of training.

Table 3 shows the results comparing the case where no text is input in pSLIQ (Baseline) vs. only the carrier phrase (Carrier) vs. only the entities (Entities) or when both carrier phrase and entities are input (Both). We can see that not including any text degrades

	Baseline	Carrier	Entities	Both
AUC	0.71	0.725	0.734	0.74
F1	0.47	0.48	0.49	0.49
Precision	0.39	0.39	0.40	0.37
Recall	0.60	0.58	0.63	0.72

Table 3: Model Performance for the defect classification task with the decomposition of the text feature. We compare the model performance with no text (Baseline), only the carrier phrase (Carrier), only entities (Entities), both carrier phrase and entities (Both).

the performance, while adding carrier phrase and entities increases the AUC from 0.71 to 0.74. Note that the benefit from adding entities is larger than carrier phrases, for instance in the utterance "play Madonna", the entity feature "Madonna" is more helpful than the carrier phrase feature "play ArtistName" for defect prediction.

In addition, we have investigated whether it is more beneficial in the training to input (a) carrier phrase and entities separately as text features (b) the utterance text as in pSLIQ. Table 4 shows that manual decomposition in input is not necessary and including the text shows overall better performance.

	Carrier Phrase + Entities	Utterance Text (pSLIQ)
Factorized	Yes	No
AUC	0.81	0.81
F1	0.56	0.61
Precision	0.43	0.49
Recall	0.81	0.81
Accuracy	0.68	0.82

Table 4: Model Performance for the defect classification task by decomposing the text in carrier phrase and entities vs. the utterance text as in pSLIQ.

4.4 Prediction of aggregate defect metrics for a new SLU model deployment

The most important application of pSLIQ is the ability to predict the IQ defect (rate) due to a new SLU model to be deployed into production by leveraging SLU features only. To this purpose, we have selected an A/B experiment and pulled recent traffic data before it for prediction, as described in Section 3.2. We have input those data into the new SLU model and used the resulting SLU output features as pSLIQ input for prediction. Regarding the ASR confidence score, for utterance texts already contained in previous traffic, we have matched the confidence score distribution, while for new texts not present before we have considered a value of 0.5. We have then aggregated the predicted defect by intent and correlated it, using both Spearman and Pearson correlation, with the measured defect aggregated at intent level during the A/B experiment. The results are shown in Table 5. For comparison, we have computed the same correlations by considering accuracy metrics calculated by testing the new SLU model standalone. These metrics are obtained by considering a labelled test set and comparing the output of the new SLU model with the reference DC, IC and NER. In commercial settings, three metrics are widely used: Intent Classification Error Rate (ICER) which is given, for each intent, by the fraction of utterance with correct SLU hypothesized intent among all utterances with reference that intent, the Information Retrieval Error Rate (IRER) which is given, for each intent, by the fraction of utterance with correct SLU hypothesized intent and labels among all utterances with reference that intent, and finally the SEMantic Error Rate (SEMER), given by SEMER(intent) = $\frac{(\text{number of label errors within intent + number of intent errors within intent})}{(\text{number of reference labels within intent + number of data within intent})}$ [9, 16, 25, 28].

In Table 5 we can see than pSLIQ results are significantly better correlated with the online measured metrics in both linear (Pearson) relationship as well as Spearman correlation, where the latter measures the monotonicity of the relation between two variables. This is because pSLIQ learns the relationship between the SLU model features and the other system components, relationship which is failed to be captured by testing the SLU model standalone. Moreover, we can see that both Spearman and Pearson achieve a correlation slightly higher than 0.75. As pSLIQ attempts to predict the IQ defect leveraging SLU features only, a 100% correlation would not be possible. However, the results in Table 5 indicate that leveraging SLU features allows to capture more than 75% of the correlation, showing the impact of the SLU model in the end-to-end voice assistant.

Correlation	**pSLIQ**	ICER	SEMER	IRER
Spearman	**0.764**	0.283	0.241	0.087
Pearson	**0.773**	0.464	0.450	0.325

Table 5: Comparison of correlations between pSLIQ predicted aggregate defect rate vs. standalone SLU model testing metrics with measured aggregated defect rate on A/B test.

5 CONCLUSIONS AND FUTURE WORK

We present an effective machine learning system to predict interaction quality defect due to SLU model changes in a conversational assistant. We leverage a multi-modal transformer architecture with a gating mechanism to combine text embeddings, obtained by a BERT model pre-trained on conversational data, together with numerical and categorical SLU features. The model predicts with more than 76% correlation the aggregate defect rate of a new SLU model in production, enabling the ability to evaluate SLU model changes without running expensive A/B tests. This allows us to overcome the issues of testing the SLU model standalone, as standalone testing does not consider the complex interaction between the SLU model and the other system components, leading to poor correlation with the real metrics.

This model has two main limitations: 1) it can only be trained on a single defect at a time; 2) it considers each request individually, without taking into consideration the underlying dialog. As future work, we want to extend pSLIQ to a Multi-task transformer to predict several defects simultaneously, as well as trained on conversations instead of single requests.

REFERENCES

[1] Praveen Kumar Bodigutla, Spyros Matsoukas, Longshaokan Marshall Wang, Kate Ridgeway, Joshua Levy, Swanand Joshi, and Alborz Geramifard. 2019. Domain-Independent turn-level Dialogue Quality Evaluation via User Satisfaction Estimation. In *SIGDIAL 2019 Workshop on Implications of Deep Learning for Dialog Modeling*.

[2] Praveen Kumar Bodigutla, Aditya Tiwari, Josep Valls-Vargas, Lazaros Polymenakos, and Spyros Matsoukas. 2020. Joint turn and dialogue level user satisfaction estimation on multi-domain conversations. In *EMNLP 2020*.

[3] Rakesh Chada, Pradeep Natarajan, Darshan Fofadiya, and Prathap Ramachandra. 2021. Error Detection in Large-Scale Natural Language Understanding Systems Using Transformer Models. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*. Association for Computational Linguistics, Online, 498–503. https://doi.org/10.18653/v1/2021.findings-acl.44

[4] Corinna Cortes, Mehryar Mohri, Michael Riley, and Afshin Rostamizadeh. 2008. Sample selection bias correction theory. In *Algorithmic Learning Theory: 19th International Conference, ALT 2008, Budapest, Hungary, October 13-16, 2008. Proceedings 19*. Springer, 38–53.

[5] 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)*. Association for Computational Linguistics, Minneapolis, Minnesota, 4171–4186. https://doi.org/10.18653/v1/N19-1423

[6] Sebastian Farquhar, Yarin Gal, and Tom Rainforth. 2021. On statistical bias in active learning: How and when to fix it. *arXiv preprint arXiv:2101.11665* (2021).

[7] Huifeng Guo, Ruiming TANG, Yunming Ye, Zhenguo Li, and Xiuqiang He. 2017. DeepFM: A Factorization-Machine based Neural Network for CTR Prediction. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*. 1725–1731. https://doi.org/10.24963/ijcai.2017/239

[8] Saurabh Gupta, Xing Fan, Derek Liu, Benjamin Yao, Yuan Ling, Kun Zhou, Tuan-Hung Pham, and Edward Guo. 2021. RoBERTaIQ: An efficient framework for automatic interaction quality estimation of dialogue systems. In *KDD 2021 Workshop on Data-Efficient Machine Learning*.

[9] Rinat Khaziev, Usman Shahid, Tobias Röding, Rakesh Chada, Emir Kapanci, and Pradeep Natarajan. 2022. FPI: Failure Point Isolation in Large-scale Conversational Assistants. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Industry Track*. Association for Computational Linguistics, Hybrid: Seattle, Washington + Online, 141–148. https://doi.org/10.18653/v1/2022.naacl-industry.17

[10] Young-Bum Kim, Dongchan Kim, Joo-Kyung Kim, and Ruhi Sarikaya. 2018. A Scalable Neural Shortlisting-Reranking Approach for Large-Scale Domain Classification in Natural Language Understanding. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 3 (Industry Papers)*. Association for Computational Linguistics, New Orleans - Louisiana, 16–24. https://doi.org/10.18653/v1/N18-3003

[11] Jannik Kossen, Sebastian Farquhar, Yarin Gal, and Tom Rainforth. 2021. Active testing: Sample-efficient model evaluation. In *International Conference on Machine Learning*. PMLR, 5753–5763.

[12] Chin-Yew Lin. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In *Text Summarization Branches Out*. Association for Computational Linguistics, Barcelona, Spain, 74–81.

[13] Ryan Lowe, Michael Noseworthy, Iulian Vlad Serban, Nicolas Angelard-Gontier, Yoshua Bengio, and Joelle Pineau. 2017. Towards an Automatic Turing Test: Learning to Evaluate Dialogue Responses. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Vancouver, Canada, 1116–1126. https://doi.org/10.18653/v1/P17-1103

[14] Phuc Nguyen, Deva Ramanan, and Charless Fowlkes. 2018. Active testing: An efficient and robust framework for estimating accuracy. In *International Conference on Machine Learning*. PMLR, 3759–3768.

[15] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Philadelphia, Pennsylvania, USA, 311–318. https://doi.org/10.3115/1073083.1073135

[16] Enrico Piovano, Dieu-Thu Le, Bei Chen, and Melanie Bradford. 2022. Online adaptive metrics for model evaluation on non-representative offline test data. In *2022 26th International Conference on Pattern Recognition (ICPR)*. IEEE, 4464–4470.

[17] Pragaash Ponnusamy, Alireza Ghias, Yi Yi, Benjamin Yao, Chenlei Guo, and Ruhi Sarikaya. 2022. Feedback-based self-learning in large-scale conversational ai agents. *AI magazine* 42, 4 (2022), 43–56.

[18] Wasifur Rahman, Md Kamrul Hasan, Sangwu Lee, AmirAli Bagher Zadeh, Chengfeng Mao, Louis-Philippe Morency, and Ehsan Hoque. 2020. Integrating Multimodal Information in Large Pretrained Transformers. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Online, 2359–2369. https://doi.org/10.18653/v1/2020.acl-main.214

[19] Alexander Schmitt and Stefan Ultes. 2015. Interaction quality: assessing the quality of ongoing spoken dialog interaction by experts—and how it relates to user satisfaction. *Speech Communication* 74 (2015), 12–36.

[20] Alexander Schmitt and Stefan Ultes. 2015. Interaction Quality: Assessing the quality of ongoing spoken dialog interaction by experts—And how it relates to user satisfaction. *Speech Communication* 74 (2015), 12–36. https://doi.org/10.1016/j.specom.2015.06.003

[21] Alexander Schmitt, Stefan Ultes, and Wolfgang Minker. 2012. A Parameterized and Annotated Spoken Dialog Corpus of the CMU Let's Go Bus Information System. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*. European Language Resources Association (ELRA), Istanbul, Turkey, 3369–3373.

[22] Stefan Schroedl, Manoj Kumar, Kiana Hajebi, Morteza Ziyadi, Sriram Venkatapathy, Anil Ramakrishna, Rahul Gupta, and Pradeep Natarajan. 2022. Improving Large-Scale Conversational Assistants using Model Interpretation based Training Sample Selection. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: Industry Track*. Association for Computational Linguistics, Abu Dhabi, UAE, 371–378. https://aclanthology.org/2022.emnlp-industry.37

[23] Pooja Sethi, Denis Savenkov, Forough Arabshahi, Jack Goetz, Micaela Tolliver, Nicolas Scheffer, Ilknur Kabul, Yue Liu, and Ahmed Aly. 2021. AutoNLU: Detecting, root-causing, and fixing NLU model errors. *arXiv preprint arXiv:2110.06384* (2021).

[24] Koustuv Sinha, Prasanna Parthasarathi, Jasmine Wang, Ryan Lowe, William L. Hamilton, and Joelle Pineau. 2020. Learning an Unreferenced Metric for Online Dialogue Evaluation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Online, 2430–2441. https://doi.org/10.18653/v1/2020.acl-main.220

[25] Chengwei Su, Rahul Gupta, Shankar Ananthakrishnan, and Spyros Matsoukas. 2018. A re-ranker scheme for integrating large scale nlu models. In *2018 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 670–676.

[26] 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 *Advances in Neural Information Processing Systems*, I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.), Vol. 30. Curran Associates, Inc.

[27] Tong Wang, Jiangning Chen, Mohsen Malmir, Shuyan Dong, Xin He, Han Wang, Chengwei Su, Yue Liu, and Yang Liu. 2021. Optimizing NLU Reranking Using Entity Resolution Signals in Multi-domain Dialog Systems. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Industry Papers*. Association for Computational Linguistics, Online, 19–25. https://doi.org/10.18653/v1/2021.naacl-industry.3

[28] Verena Weber, Enrico Piovano, and Melanie Bradford. 2021. It is better to Verify: Semi-Supervised Learning with a human in the loop for large-scale NLU models. In *Proceedings of the Second Workshop on Data Science with Human in the Loop: Language Advances*. Association for Computational Linguistics, Online, 8–15. https://doi.org/10.18653/v1/2021.dash-1.2