Improving Large-Scale Conversational Assistants using Model Interpretation based Training Sample Selection

Stefan Schroedl Manoj Kumar Kiana Hajebi Morteza Ziyadi
Sriram Venkathapaty Anil Ramakrishna Rahul Gupta Pradeep Natarajan
Amazon Alexa AI, USA
{schroedl, abithm, hajkiana, mziyadi, vesriram, aniramak, gupra, natarap}@amazon.com

Abstract

Natural language understanding (NLU) models are a core component of large-scale conversational assistants. Collecting training data for these models through manual annotations is slow and expensive that impedes the pace of model improvement. We present a three stage approach to address this challenge: First, we identify a large set of relatively infrequent utterances from live traffic where the users implicitly communicated satisfaction with a response (such as by not interrupting), along with the existing model outputs as candidate annotations. Second, we identify a small subset of these utterances usings *Integrated Gradients* based importance scores computed with the current models. Finally, we augment our training sets with these utterances and retrain our models. We demonstrate the effectiveness of our approach in a large-scale conversational assistant, processing billions of utterances every week. By augmenting our training set with just 0.05% more utterances through our approach, we observe statistically significant improvements for infrequent tail utterances: a 0.45% reduction in semantic error rate (SemER) in offline experiments, and a 1.23% reduction in defect rates in online A/B tests.

1 Introduction

Large-scale, voice-based conversational assistants such as Alexa, Siri, Google Assistant and Cortana process each utterance through a multi-stage pipeline that includes wakeword detection, automatic speech recognition (ASR), natural language understanding (NLU), entity resolution, and text-to-speech. This is a well-understood sequence (Sarikaya, 2017) and each of these steps leverage multiple machine learning models. The NLU system is often modularized into a number of *domains* that handle distinct classes of utterances such as Music, Weather, etc. (Su et al., 2018). The assistant system comprises models for *domain classification (DC), intent classification (IC),* and *named entity recognition (NER).*

A key challenge in building, extending and maintaining such a system is that the underlying models need annotated training data. Collecting large volumes of such data through manual labeling is expensive and does not scale. Our work aims at improving the efficiency of this process. In contrast to previous approaches which identify utterances with defective responses, we instead focus on identifying cases that were processed successfully by the conversational assistant, and automatically retraining models with the additional data. However, this introduces two challenges. First, the vast majority of utterances are already processed correctly by the deployed system, resulting in an overwhelmingly large set of augmentation candidates. Secondly, implicit signals for satisfaction are noisy, as users might frequently ignore incorrect responses without making the effort to reformulate their query or provide a response that reflects dissatisfaction with the experience. Thus, simply adding all utterances from the full candidate pool (potentially billions/week) is infeasible and might actually degrade performance due to noise. We present a novel approach to address this based on *Integrated Gradients (IG)* (Sundararajan et al., 2017), a technique for understanding model behavior through feature importance scores. We propose a *sample importance score* that aggregates word scores and ranks the utterances in our initial candidate set, followed by training set augmentation with a small fraction of the top utterances.

Our experiments on live traffic data from a large scale conversational assistant indicate that retraining models with training sets, augmented by as little as 0.05% in size, produces a statistically significant ($p-\text{value} < 0.05$) improvement in semantic error rate (SemER) in offline test sets – 0.27% overall and 0.45% on a more challenging set of less frequent "tail" utterances. In online A/B tests, we

Proceedings of EMNLP 2022 Industry Track, pages 381–388
December 9–11, 2020.

observe a 1.23% and 0.96% improvement in defect rates for all and tail utterances, respectively. In contrast, simply adding *all* utterances from our initial candidate set *degrades* SemER by 1.74% and 3.13% on the full and tail data sets, respectively. Finally, we demonstrate the repeatability and generalizability of our approach on public benchmark data sets. Despite the lack of label noise, we see small but consistent accuracy gains of 0.21% resp. 0.65% on the Snips and AGNews data sets.

2 Related Work

Several approaches have been proposed recently to use *distant* or *weak supervision* to address sparsity of labeled data (see e.g. the survey in (Hedderich et al., 2020)). A number of works identify utterances with processing errors through offline analysis (Sethi et al., 2021; Gupta et al., 2021; Chada et al., 2021; Khaziev et al., 2022). These approaches however still need human annotation in an active learning loop to improve production models. Query rewriting based approaches (Ponnusamy et al., 2020; Sodhi et al., 2021; Su et al., 2019; Hao et al., 2020) aim to address this limitation and enable self-learning without the need for human annotation. They detect instances where a user reformulates a query due to an unsatisfactory response and learn to map the failed utterance to a subsequent successful one. However, such approaches do not generalize to other utterance shapes. Falke et al. (2020) leverage user paraphrasing behavior in dialog systems to automatically collect annotations for long-tail utterances. Moerchen et al. (2020) present an approach where implicit negative feedback from the user is used to train a re-ranker that is then applied to pick correct annotations for under-performing utterances.

A range of post-hoc *model interpretability methods* for machine learned models has been developed in recent years (see e.g., (Madsen et al., 2021; Sundararajan et al., 2017; Ribeiro et al., 2016; Lundberg and Lee, 2017). Local black-box methods typically measure the influence of individual features of an input example (e.g., individual words in a sentence) on the output prediction. Other techniques aim to score complete examples with respect to prototypicality (Carlini et al., 2019), influence on test predictions (Garima et al., 2020), and difficulty (Agarwal et al., 2022). Our word-occurrence based approach can be seen as a computationally scalable linear approximation to such measures.

Bhatt et al. (2019) conducted a survey on how organizations use model interpretability in practice. They identified model debugging as one of the primary uses of model explainability, seeking explicit human feedback on gathering more data for improving model performance.

Our approach differs from previous work as follows:

- There has been no prior work on the use of model interpretability in the context of data augmentation (though inversely, Chen and Ji (2019) proposed data augmentation to improve model explainability).
- Instead of detecting failed user interactions, we focus on utterances with implicit *positive* feedback.
- We leverage interpretability techniques in an automated way, without the need for human inspection.

3 Implicit User Feedback based Data Augmentation

NLU services which cater to a large number of users such as voice controlled agents typically collect implicit feedback metrics for each interaction. As a simplistic example, the absence of any negative feedback from the user to the agent's action (no interruption, no repetition, etc.) can suggest that the agent successfully served the user's request. In this paper, we propose a mechanism that relies on successful user interactions to identify additional data for building NLU models.

Oftentimes, an unsuccessful action from our virtual agent is followed by the user rephrasing their request. If the rephrase is successfully served by the agent, then this indicates that the NLU hypothesis for the rephrased request is likely correct. We believe that a correctly served rephrased turn is a stronger positive feedback when compared to a single-turn interaction with an implicit positive feedback (i.e no negative feedback from the user). We create a new training sample using the ASR transcript of the rephrased user request and the NLU hypothesis. We call this data set *weak signal labeled (WSL)* data since we rely on weak supervision from the user to obtain NLU labels. We score these utterances using integrated gradient technique as described in Section 4 before using them as additional data source for building NLU models.

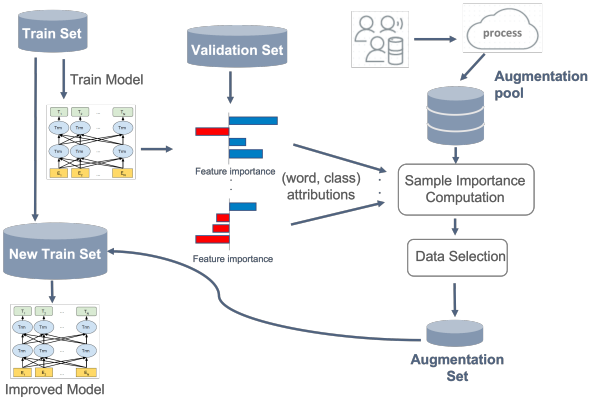

Figure 1: Model improvement based on feature attribution.

4 Importance Score Computation

We describe our approach for scoring utterances according to their importance to the performance of a classification model (e.g., a domain or intent classifier). Let T be the original training data, A be a pool of augmentation data, V be a validation data set, and Y be the test set. The model trained on T typically makes some mistakes on V. The objective is to use word attribution scores computed with a local black-box interpretability technique (Sundararajan et al., 2017; Ribeiro et al., 2016) on V to score utterances in A; then, by adding some of them to T, we hope to train a model that is more robust against failures on Y that are similar to those observed on V. Thus, our approach can be roughly subdivided into three steps:

1. Calculation of an attribution score for each word in a misclassified utterance in V (with respect to the target and/or predicted class).
2. Aggregation of word scores over all instances.
3. Scoring of utterances in A based on the occurrences of important words.

See Figure 1 for a high level flow chart of our approach. We describe the details of each of these steps in the following sections.

4.1 Model interpretability

In this paper, we conduct experiments using the *Integrated Gradients* (Sundararajan et al., 2017) method. It is a local interpretation technique that addresses the problem of attributing a prediction of a deep network to its input features. Our approach is not restricted to it and it could be replaced with other methods such as LIME (Ribeiro et al., 2016) or SHAP (Lundberg and Lee, 2017). However, integrated gradients has several advantages:

word	True	Pred	T-Attr	P-Attr
tell	Books	Information	-0.61	+3.29
us	Books	Information	-1.01	+0.64
a	Books	Information	+0.37	+0.93
bedtime	Books	Information	-2.85	+3.27
story	Books	Information	+7.82	-3.80

Table 1: Feature attributions for true and predicted classes.

- It is scalable to large volumes of data.
- Its computed attributions are deterministic.
- It satisfies the desirable axioms of *linearity*, *implementation invariance*, and *sensitivity* (Sundararajan et al., 2017), which facilitate comparability of attributions across features and instances.

Integrated gradients require a non-informative baseline input. In the context of text processing, a natural choice is a sequence formed of padding tokens of the same length as the input. We interpret the words of an utterance as the features to be attributed, by averaging over token embedding vectors. Our implementation makes use of the PyTorch Captum package (Kokhlikyan et al., 2020).

Table 1 illustrates an example utterance in the validation set of the domain classifier along with the feature attributions of the words towards the true class (Books) as well as the predicted class (Information). We can see that the word `story` has positively influenced the model towards the true class but was not able to influence enough to make a right prediction. The word `tell` has positively influenced the utterance towards an incorrect prediction, while the word `bedtime` has negative influence towards the true class and high positive influence towards the predicted class. The objective, therefore, is to alter the training data so that the words `tell` and `bedtime` become more strongly associated with the class Books, especially in the context of the anchor word `story`.

4.2 Aggregation of word scores

The interpretability method produces an attribution mapping ρ: $(u, w, c, M) \rightarrow \mathbb{R}$, where u is an utterance in the validation set, w is a word in u, c is the class label, and M is the interpreted model.

Let the aggregated word scores be

$$g(w,c) = \sum_{(u,c)\in V, c'\in C} \max(0, -\rho(u,w,c',M)\cdot\delta(c,c',M)) \quad (1)$$

The function δ indicates the direction of the influence gap based on the true label of the utterance and the model prediction. Negative attributions with respect to the true class ($c = c'$), and positive attribution towards wrongly predicted classes ($c \neq c' = M(u)$) are summed over the validation set. Our objective is to enrich the training set with examples for the true class containing these words.

$$\delta(c,c',M) = \begin{cases} 1 & c = c' \\ -1 & c \neq c' = M(u) \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

4.3 Score an utterance using word attributions

Let $G = \{(w,c)\}$ be the set of all word attributions computed according to Eqn. 1. The most straightforward way of computing the score of utterance u of class c is the *greedy method* of summing up the importance scores of all word occurrences:

$$h(u,c) = \sum\{g(w,c) \| w \in u, (w,c) \in G\} \quad (3)$$

Then, we select the top n utterances from A according to this score.

The greedy method has the drawback that it can become too narrowly focused on just a few high-scoring words, thus leading to heavily imbalanced augmentation data sets. We introduce a *diversity method* as a remedy. The idea is to incrementally discount a score pair after selecting an utterance containing it. One simple way of doing so is by dividing the word importance by the number of such utterances, as outlined in Algorithm 1.

Algorithm 1 Diversity method for utterance scoring.

```
function SELECT_DIVERSITY(n, G, A)
    for all (w, c) ∈ G do
        n(w, c) ← 1
    end for
    S ← {}
    for i ← 1, ..., n_augment do
        for all (u, c) ∈ A \ S do
            h(u, c) ← Σ{g(w, c)/n(w, c) ||
                        w ∈ u, (w, c) ∈ G}
        end for
        (u', c') ← arg max_{u,c} h(u, c)
        A ← A ∪ {(u', c')}
        for all w' ∈ u', (w', u') ∈ G do
            n(w', c') ← n(w', c') + 1
        end for
    end for
    return S
end function
```

5 Experimental Setup

We first present initial results of our approach on the open source intent classification (IC) data sets (Snips (Coucke et al., 2018) and AGNews (Del Corso et al., 2005)), and then demonstrate the impact of our approach on a joint intent classification and named entity recognition (IC-NER) task on live traffic of a commercial conversational assistant.

5.1 Data sets

5.1.1 Open source data sets

Snips (Coucke et al., 2018) is a natural language understanding benchmark data set of over 16 000 crowdsourced queries distributed among 7 NLU intents. It is pre-split into a training set (13 084 utterances), validation set, and test set (700 utterances each with 100 queries per intent).

With **AGNews** (Del Corso et al., 2005), we chose a data set with a slightly different but related task (news topic classification), due to its sufficient size. It contains 4 classes each containing 30 000 training samples and 1 900 testing samples. The total number of training and testing data is 120 000 and 7 600, respectively. We apply stratified random sampling to subdivide the training further for a validation set of 7 600 instances.

Apart from the usual partitions of data into train, validation and test sets, our experiments consider a further sub-partition of the train set into base set and augmentation set. The base set is a randomly down-sampled version of the full train set to a desired target size (e.g., 50% of the data). The rest of the training data is the augmentation set from which additional samples are selected. The validation set is used for computing the feature attributions.

5.1.2 Proprietary data from conversational assistant

In these experiments, we work with logs of user interactions with our conversational assistant. This data is prepared in accordance with our general strict privacy protection procedures. All production data is de-identified so that it is not personally identifiable.

We evaluate our approach on the utterances from a random partition of live traffic as well as on a set of low frequency *tail* utterances. Tail utterances constitute a significant portion of the overall traffic, and measure the statistical model's ability to generalize to a wide gamut of real-world utterances of users. Improving a machine-learned model's performance on rare utterances is of increasing interest among industrial and academic applications, as defects in frequently recurring head utterances can often readily be addressed using rule based systems. We compare models in terms of offline NLU performance, but also run live traffic A/B experiments to directly measure the user impact.

Our weak signal labelled data stems from unique utterances with implicit positive user feedback across all domains over a period of time. For improved precision, we remove utterances with ASR confidence scores below an empirically determined threshold. We rank the utterances within each domain using the scores obtained using interpretability methods, greedy and diversity, as described in Section 4.1. The WSL data set thus prepared represents $\approx 8.5\%$ of the total training data size. For each domain, we rank WSL utterances in the order of decreasing relevance: we favor utterances which are likely to influence the domain classification model predictions the most. We select a small fraction of the most influential utterances (0.05% of the training data) and fine-tuned IC-NER models. The amount of data that can be augmented is limited by engineering constraints (e.g., model build times, storage capacity), hence the interpretability-based scores are useful to identify an optimum subset of utterances that provide the most utility.

5.2 Models

On the live traffic data set, we use a joint IC-NER model with a distilled version of BERT encoder pretrained with MLM objective on a combination of public and internal data sets. The total parameter count of the encoder is $17M$. We use a sentencepiece tokenizer of size $150K$ sub-word units. For each of IC and NER tasks, the model uses feed-forward layers of hidden size 256 followed by softmax layer. We train with early stopping, up to 10 epochs, at a learning rate of $5\mathrm{e}{-5}$ and a batch size of 32. For the simpler public domain datasets, we fine-tune the DistilBERT model from Huggingface[1] ($65M$ parameters) as an intent classifier.

6 Experiments and Results

6.1 Snips

For the experiments with data augmentation on Snips, we use the model trained on $\approx 50\%$ of the training data (T) – constituting $8\,192$ out of $13\,084$ samples – as the base model. The remaining data is considered the augmentation set (A). We investigate the accuracy impact of augmenting a small fraction of examples to the training set with our importance scoring approach. Specifically, we explore the greedy and diversity methods as explained in Section 4.3, with a set of 82 ($\approx 1\%$) augmentation utterances selected from A. As shown in Table 2, the diversity method improves performance to 0.976 compared to the baseline accuracy of 0.974, while a model trained with the full augmentation data set has 0.978 accuracy.

6.2 AGNews

For our interpretability experiments on the AGNews data set, we choose a base model trained on $\approx 25\%$ of data set, achieving an accuracy of 0.923. According to Table 2, the diversity method achieves a test accuracy of 0.929, an improvement over the baseline by 0.6%.

6.3 Data augmentation with weak signal data

We evaluate the utility of WSL data augmentation using model interpretability scores for NLU models of the conversational assistant. We build a domain-specific IC-NER model using the same training data as in the production setting. All IC-NER models share a common encoder as described in Section 5. The output dimension for each model depends on the number of intents and slot labels for each domain. We use similar training parameters (epochs, learning rate, optimizer, etc.) as production settings and defer any hyper-parameter tuning experiments for future work. We refer to this model as *baseline.*

For each domain, we build a second model *(WSL)* using the same architecture and training parameters as the baseline model. We augment all

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

data set	Full size	Baseline size	Modification size	Baseline accuracy	Augmentation		Full accuracy
					Diversity	Greedy	
Snips	13,084	8,192	82	0.974	0.976	0.971	0.978
AGNews	112,400	32,768	64	0.923	0.929	0.926	0.942

Table 2: Accuracy of intent classification on Snips and topic classification on AGNews data sets, comparing different approaches with random selection baseline. Each number is the average over 5 runs with different seeds.

Table 3: Relative semantic error rates (SemER) for IC-NER models trained on all WSL data (WSL), and WSL data filtered with interpretability-based scores, greedy and diversity, (WSL-IG). All metrics are reported relative to baseline model ($p < 0.05^*$).

Model	All	Tail
Baseline	0%	0%
WSL (no filtering)	1.74%*	3.13%*
WSL-IG (greedy)	-0.27%	-0.45%*
WSL-IG (diversity)	-0.13%	-0.33%*

Table 4: Relative defect rate from online A/B experiment comparing NLU models built with WSL data. The defect rates are reported for low-frequency utterances *(Tail)* and all utterances *(All)* relative to the control model ($p < 0.05^*$).

	Overall	General	Information
All	-1.23%*	-0.27%	-1.04%*
Tail	-0.96%*	-1.32%*	-1.64%*

the WSL data described in Section 3 to the training data, before applying importance scores for utterance selection. Finally, we build a third model *(WSL-IG)* which uses interpretability-based scores to select the most relevant utterances.

We report the IC-NER task performance using weighted semantic error rate (SemER; (Makhoul et al., 1999; Su et al., 2018)) metric. We construct a label sequence for each utterance by concatenating the intent and slots (in order). Given the total count of erroneous insertion (I), erroneous deletions (D), substitutions (S) and correct labels (C), SemER is computed as: $S = \frac{(I+D+S)}{(C+D+S)}$. In Table 3, we report the weighted mean of SemER relative to the baseline model and weighted by the domain's test utterance count. We compare the baseline and proposed models on two test sets: (i) *All* contains user queries from the entire traffic; (ii) *Tail* contains only low-frequency requests.

From Table 3, we notice that the interpretability-based filtering plays an important role in improving the semantic error rate on both test sets. SemER reductions obtained with WSL-IG models are significant at $p < 0.05$ on the tail test set. The magnitude of SemER improvement is higher on the Tail test set, which is likely due to the similar nature of WSL utterances (sourced from low-frequency traffic). Interestingly, WSL models which are built using the largest training data sizes are significantly worse than the baseline, illustrating the noisy nature of implicit user feedback. In contrast to our Snips and AGNews results, the greedy method performs better than diversity – possibly due to the much larger training size, and suggesting a more diverse range of defect patterns.

We followed up with an online A/B experiment on our voice-controlled agent to test the impact of WSL data on live traffic. We experiment with two domains: *General* which serves generic requests such as turn on device, change volume, etc. and *Information*, which serves general knowledge related requests. For both domains, user requests on the treatment group were served by NLU models trained with additional WSL data which were filtered using interpretability-based scores, (WSL-IG). We measure the outcome of the A/B experiment using an internal business metric (referred to as defect rate) which estimates whether the agent was successful in serving the user's request. Success is estimated based on the user's perception following the agent's response. For example, it is likely that the agent has misinterpreted the user's request when the user rephrases/repeats their request or the agent communicates that it cannot serve the user's request: *"sorry I don't know the answer to that"*. We present the relative change in defect rate on both low-frequency utterances *(Tail)* and all utterances *(All)*. From Table 4, we observe improvements in the defect rate across both deployed domains and the overall traffic. For both domains, while defect reductions are observed on both All and Tail test sets improvements in the latter are more noticeable, which demonstrate the utility of interpretability-based filtering of implicit customer feedback for NLU model building.

7 Conclusions and Future Work

A key challenge in building state-of-the-art deep learning models is the cost and effort involved in obtaining large volumes of manually labeled data. Our work is part of a line of investigations into leveraging unlabeled or weakly supervised data at scale. We extract large amounts of user dialogs with a conversational assistant which are deemed successful according to implicit feedback. However, it is not sufficient to add all such examples indiscriminately to the training data – doing so does not improve the model, nor is it computationally scalable. We show how to leverage model interpretability techniques to prioritize the most important instances that should be added to the training set. Our approach leads to statistically significant error rate reductions of our live system. We also demonstrate transferability on public NLU data sets, Snips and AGNews.

In the future, we will apply our approach to other challenging public data sets which suffer from significant label noise and ambiguity. We will investigate other types of data set modifications, such as removal and replacement of examples.

References

Chirag Agarwal, Daniel D'souza, and Sara Hooker. 2022. Estimating example difficulty using variance of gradients. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10368–10378.

Umang Bhatt, Alice Xiang, Shubham Sharma, Adrian Weller, Ankur Taly, Yunhan Jia, Joydeep Ghosh, Ruchir Puri, José M. F. Moura, and Peter Eckersley. 2019. Explainable machine learning in deployment. *CoRR*, abs/1909.06342.

N Carlini, U Erlingsson, and N Papernot. 2019. Prototypical examples in deep learning: Metrics, characteristics, and utility. *arXiv*.

Rakesh Chada, Pradeep Natarajan, Darshan Fofadiya, and Prathap Ramachandra. 2021. Error detection in large-scale natural language understanding systems using transformer models. In *ACL-IJCNLP 2021*.

Hanjie Chen and Yangfeng Ji. 2019. Improving the interpretability of neural sentiment classifiers via data augmentation. *CoRR*, abs/1909.04225.

Alice Coucke, Alaa Saade, Adrien Ball, Théodore Bluche, Alexandre Caulier, David Leroy, Clément Doumouro, Thibault Gisselbrecht, Francesco Caltagirone, Thibaut Lavril, et al. 2018. Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces. *arXiv preprint arXiv:1805.10190*.

Gianna M Del Corso, Antonio Gulli, and Francesco Romani. 2005. Ranking a stream of news. In *Proceedings of the 14th international conference on World Wide Web*, pages 97–106.

Tobias Falke, Markus Boese, Daniil Sorokin, Caglar Tirkaz, and Patrick Lehnen. 2020. Leveraging user paraphrasing behavior in dialog systems to automatically collect annotations for long-tail utterances. In *COLING 2020*.

Garima, Frederick Liu, Satyen Kale, and Mukund Sundararajan. 2020. Estimating training data influence by tracing gradient descent. In *Advances in Neural Information Processing Systems*, volume 2020-Decem.

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

Jie Hao, Linfeng Song, Liwei Wang, Kun Xu, Zhaopeng Tu, and Dong Yu. 2020. Robust dialogue utterance rewriting as sequence tagging. *CoRR*, abs/2012.14535.

Michael A Hedderich, Lukas Lange, Heike Adel, Jannik Strötgen, and Dietrich Klakow. 2020. A survey on recent approaches for natural language processing in low-resource scenarios. *arXiv preprint arXiv:2010.12309*.

Rinat Khaziev, Usman Shahid, Tobias Röding, Rakesh Chada, Emir Kapanci, and Pradeep Natarajan. 2022. FPI: Failure point isolation in large-scale conversational assistants. In *NAACL-HLT*.

Narine Kokhlikyan, Vivek Miglani, Miguel Martin, Edward Wang, Bilal Alsallakh, Jonathan Reynolds, Alexander Melnikov, Natalia Kliushkina, Carlos Araya, Siqi Yan, et al. 2020. Captum: A unified and generic model interpretability library for pytorch. *arXiv preprint arXiv:2009.07896*.

Scott M Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. In *Neural Information Processing Systems*.

Andreas Madsen, Siva Reddy, and Sarath Chandar. 2021. Post-hoc interpretability for neural NLP: A survey. *arXiv preprint arXiv:2108.04840*.

John Makhoul, Francis Kubala, Richard Schwartz, and Ralph Weischedel. 1999. Performance measures for information extraction. In *In Proceedings of DARPA Broadcast News Workshop*, pages 249–252.

Fabian Moerchen, Patrick Ernst, and Giovanni Zappella. 2020. Personalizing natural-language understanding using multi-armed bandits and implicit feedback. In *CIKM 2020*.

Pragaash Ponnusamy, Alireza Roshan-Ghias, Chenlei (Edward) Guo, and Ruhi Sarikaya. 2020. Feedback-based self-learning in large-scale conversational AI agents. In *AAAI 2020*.

Marco Túlio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "Why should I trust you?": Explaining the predictions of any classifier. *CoRR*, abs/1602.04938.

Ruhi Sarikaya. 2017. The technology behind personal digital assistants: An overview of the system architecture and key components. *IEEE Signal Process. Mag.*, 34(1):67–81.

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. *CoRR*, abs/2110.06384.

Sukhdeep S. Sodhi, Ellie Ka In Chio, Ambarish Jash, Santiago Ontañón, Ajit Apte, Ankit Kumar, Ayooluwakunmi Jeje, Dima Kuzmin, Harry Fung, Heng-Tze Cheng, Jon Effrat, Tarush Bali, Nitin Jindal, Pei Cao, Sarvjeet Singh, Senqiang Zhou, Tameen Khan, Amol Wankhede, Moustafa Alzantot, Allen Wu, and Tushar Chandra. 2021. Mondegreen: A post-processing solution to speech recognition error correction for voice search queries. *CoRR*, abs/2105.09930.

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 2018, Athens, Greece, December 18-21, 2018*, pages 670–676. IEEE.

Hui Su, Xiaoyu Shen, Rongzhi Zhang, Fei Sun, Pengwei Hu, Cheng Niu, and Jie Zhou. 2019. Improving multi-turn dialogue modelling with utterance ReWriter. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 22–31, Florence, Italy. Association for Computational Linguistics.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. *CoRR*, abs/1703.01365.