Towards Explainable Dialogue Systems: Explaining Intent Classification using Saliency Techniques

Ratnesh Kumar Joshi[1], **Arindam Chatterjee**[12] **and Asif Ekbal**[1]
[1]Department of Computer Science and Engineering, IIT Patna, Patna, India
[2]Wipro AI Research Labs, Bangalore, India
(ratnesh_1921cs28, arindam_2021cs02, asif)@iitp.ac.in

Abstract

Deep learning based methods have shown tremendous success in several Natural Language Processing (NLP) tasks. The recent trends in the usage of Deep Learning based models for natural language tasks have definitely produced incredible performance for several application areas. However, one major problem that most of these models face is the lack of transparency, *i.e.*, the actual decision process of the underlying model is not explainable. In this paper, first we solve a very fundamental problem of Natural Language Understanding (NLU), *i.e.*, intent detection using a Bidirectional Long Short Term Memory (BiLSTM). In order to determine the defining features that lead to a specific intent class, we use the Layerwise Relevance Propagation (LRP) algorithm to find the defining feature(s). In the process, we conclude that saliency method of ϵLRP (epsilon Layerwise Relevance Propagation) is a prominent process for highlighting the important features of the input responsible for classification of intent, which also provides significant insights into the inner workings, such as the reasons for misclassification by the black box model.

1 Introduction

Chatbots or conversational agents have been gaining immense popularity in recent years. This is one of the most widely used Artificial Intelligence (AI) applications that has a market value of USD 190.8 millions, and is expected to grow upto USD 1,250.1 million by the year 2025[1]. These chatbots are being used in almost every vertical of our society, such as travel, healthcare, judiciary *etc.* With the rapid adaptation of chatbots as digital assistants, it is important that these chatbots should be very robust, as many of these domains (*e.g.*, health, judiciary *etc.*) are very sensitive, and minor inaccuracies in information can lead to significant damage. The model as a whole can be made robust if all its individual components are also accurate. The very first step of most modular dialogue systems is the Natural Language Understanding (NLU) phase, that comprises of *dialogue act classification*, *intent detection* and *slot filling*. This part of the dialogue system plays an important role of deciphering the syntax and semantics of the user input, to aptly produce the bot's reply. While the intent classification focuses on the semantic meaning of the input, slot filling focuses on extracting the relevant information like named entities *etc.*

These systems are not perfect, and even the *state-of-the-art* models very often fail to classify the intent correctly. In order to understand what went wrong in these misclassifications, the features of the text that led to the incorrect classification can provide some helpful information. Due to the rise of deep neural network based architectures the transparency of such models is low. This leads to the requirement of eXplainable Artificial Intelligence (XAI) methods that determine the important features of the input text. There are 2 major methods that highlight the feature importance, namely *saliency* based methods and *attention* based methods. Saliency based methods (Section 2.2) are adhoc techniques that explain individual inference done by the model. This is done after the model training process, hence the cost depends on the number of explanations required. The additional cost of these XAI models tend to make the architectural framework more expensive. Since most of the XAI methods explain each prediction individually, the processing cost keeps on increasing during the model deployment. Attention, on the other hand, calculates the feature importance over the entire training data, and seems like a cost effective alternative to saliency in figuring out the

[1]https://www.grandviewresearch.com/industry-analysis/chatbot-market

Proceedings of the 18th International Conference on Natural Language Processing, pages 120–127
Silchar, India. December 16 - 19, 2021.

relevant features of the input text. Whether attention is actually a good alternative for explanation is still a matter to be explored (Grimsley et al., 2020). When compared to the saliency based techniques, due to the attention's focus on gradient descent as the weight updating criteria, attention has shown high correlation to the gradient based saliency techniques (Jain and Wallace, 2019).

In this paper, we investigate into an explainable deep learning based intent classifier. We compute the features responsible for misclassification of the utterances, in order to get a better idea of why actually the trained model incorrectly predicted these test inputs. This leads to a much better understanding of the limitations of Long Short Term Memory (LSTM) based models. One such limitation is when we go over the ATIS dataset, where we find out that the model misclassifies an Intent class(as shown in figure 2) '*meal*' even though it learns to identify the '*meal*' token as the most important feature as cumulative weight of tokens pointing to the '*flight*' intent is higher. Another such instance can be the '*day_name*' intent being misclassified as '*flight*'. The model does not even learn to pay attention to token(s) like '*day*' or '*day of the week*' as the total instances of '*day_name*' in the entire dataset is less than 0.1% of the dataset.

2 Related Work

In this section, we present a very brief literature survey that starts with a intent classification followed by saliency based explainable models.

2.1 Basic Components of any Conversational System

In practice, two forms of chatbot architectures are prevalent. One being the modular architecture that we focus here in this paper. This procedure breaks the conversational process into a pipeline structure where the upcoming module uses the information gathered from the previous step to build a functional agent. The process includes intent classification, slot filling/entity extraction for the language understanding phase. Dialogue Management (DM) uses the intent and entities extracted to formulate the next action. In this phase we can also employ rules to direct the functioning to a specific action. For example- one rule could be that if the input is exactly the same, use the reply in the training data directly. Of course, this depends on the task at hand. Finally, the information of the DM module is actualized into human understandable text using the Natural Language Generation (NLG) phase. This text generation can be either template based or a neural based, trained on the data available. The second prevalent architecture is the end-to-end architecture which trains a single deep learning model which takes the user input and gives reply utterance directly in one go. Since the entire process of conversation is condensed into one single model this kind of architecture generally requires much more data and since we cannot explicitly impart rules on such a model, it can perform poorly on seemingly simple tasks for a similar amount of data.

2.2 Intent Classification

Intent classification is a highly informational step of any modular dialogue system. It is the initial process of the Natural Language Understanding (NLU) pipeline, which focuses on the prediction of the task the user wants the current input to focus on, from the variety of tasks the model has been trained to perform. The Cambridge dictionary defines intention as 'something you want or plan to do'. Similarly in NLP the intent refers to the task/goal the user wants to accomplish by the conversation. For example, in the user utterance 'what meals are available in flight from Milwaukee to Seattle' the goal/intent of the user is to enquire about the food options. The structure of this utterance is similar to that of a flight search query like 'what flights are available from Milwaukee to Seattle', we want the model to be able to aptly distinguish between these intents. A good intent classifier can bypass poorly directed user queries and correctly processes user intents leading to the smooth conversational flow. Bi-directional Long Short Term Memory (BiLSTM) (Huang et al., 2015) models are a decent baseline in NLP tasks including classification, generation, summarization etc. However, due to the innate opaqueness of neural network based models it leaves a lot to be desired in terms of making the users understand the decision making process. This leads to people using adhoc post-processing steps (saliency techniques) to find out the features/tokens in our text most responsible for the classification output of the model. However, since this adds an overhead to the model, it results in increased cost for providing feature importance.

2.3 Saliency Techniques

Saliency is used in psychology and other fields with subtly varying meanings. For NLP tasks, we refer to saliency as the process of finding the most important features/token(s) responsible for the model. For example, in the utterance "The service was bad.", the token(s) 'service' and 'bad' are responsible for the utterance to be classified with intent 'complaint'. There are multiple classes of these models with the focus, ranging from token combinations, to game theory concepts (Section 2.3.2) and propagation based (Section 2.3.3). We discuss some techniques for saliency and try to highlight the issues pertaining to these models for adhoc explanations.

2.3.1 Occlusion/Perturbation based

The occlusion or perturbation based methods (Zeiler and Fergus, 2014) compute the feature importance by removing parts of the input and recalculating the classification output and measuring the deviation from the original classification. This deviation from the original prediction then serves as a measure of the importance of the feature with respect to the current model classification. Though these methods are easy to execute for the Natural Language Processing (NLP) tasks, these add a high computation overhead in order to find the important features. For example - for just a text of 10 tokens there can be hundreds of such perturbation based subtexts resulting in a tedious prediction phase. The number of such perturbation based combinations of tokens increases exponentially with the size of the input text. Even though you can use meaningful combination techniques(Fong and Vedaldi, 2017) (here the overhead can be reduced by using many techniques like stopword removal, named entity removal, merging adjective with adverbs such as treating 'very good' as a single occlusion candidate etc.) the substantial overhead still exists.

2.3.2 Mathematical Model based

GradientxInput(Denil et al., 2014) calculates saliency of the input text as a function of input sequence vs individual input tokens. Integrated Gradient(Sundararajan et al., 2017) is another gradient based method that extends upon GradientxInput techniques and deals with the sensitivity and implementation invariance. Even though both IG and GradientXInput focus on the sensitivity of the features, it is taken as a measure of the saliency of the input features. SHAP (Lundberg and Lee, 2017) uses the concept of shapley values from game theory to calculate the feature importance.

2.3.3 Propagation based

Layerwise Relevance Propagation (LRP)(Bach et al., 2015) uses an additional backward pass that calculates the relevance of the nodes of our network at each layer. It then uses the weights of the nodes to redistribute the relevance of each layer with respect to the prediction. So, when it finally arrives at the input layer it has the relevant information for each input with regard to the prediction. Since the backward pass flows over the entire network, the cost of saliency is directly proportional to the size of the network trained (example, overhead for a model with 10 layers of depth is more than a model with 2 layers).

2.3.4 Sampling based

LIME(Ribeiro et al., 2016) adopts a local approach to the saliency problem. For a specific input at hand it calculates a locality around the input and then uses that local sample space to train an inherently interpretable model. This newly trained model is then used to make an explanation regarding feature importance for the input. However, in some cases like image classification, even these localities might be too much to be represented by a linear model, Anchors(Ribeiro et al., 2018) is a method which counters this issue by instead forming conditions for prediction. This rule/condition fixes the prediction at local level so changes ant global level. Thus highlighting the parts in the input that are enough to classify it. However, since the technique involves sampling from the training data and also training a new model (both for each explanation to any input), such methods are some of the most expensive saliency methods available.

2.4 Global vs Local Methods

Global methods describe the average behavior of a machine learning model. Global methods estimate expected values based on the distribution of the data. For example, the partial dependence plot (Friedman, 2001), a feature effect plot, is the most expected outcome when all other features are turned insignificant i.e. it shows the marginal effect one or two features have on the predicted outcome of a machine learning model. Since global interpretation methods describe average behavior, they are particularly useful when the user wants to understand the general mechanisms in the data or debug

a model.

The counterpart to global methods are local methods. Local interpretation methods explain individual predictions. LIME(Ribeiro et al., 2016) and SHAP(Lundberg and Lee, 2017) are attribution methods, so that the prediction of a single instance is described as the sum of feature effects. Other methods, such as counterfactual explanations(Wachter et al., 2017), are example-based.

For this paper, we focus on local explanations,i.e. look at the individual instances and try to figure the reason for misclassification for each group of instances instead of figuring a general trend for all the misclassification.

3 Methodology

We implement ϵLRP (epsilon Layerwise Relevance Propagation) model over a BiLSTM trained model to find the important features for a particular prediction. This is done with the aim of finding the reason behind the misclassification in incorrectly predicted utterances, as that can potentially help us deduce the reason for misclassification and improve our understanding of the model.

3.1 Intent Classification

For the base model, we use a BiLSTM based architecture. Bidirectional LSTM (BiLSTM) is used to model dependencies on the next time step in the input utterance. These are a combination of a recurrent layers that propagate the sequence forward through blocks and a recurrent module that propagates the sequence backwards through a different block. The tail model uses a concatenation operation on the penultimate two hidden states as input for the final layer.

$$
\begin{aligned}
i_{0,t} &= sigmoid(W_i x_t + b_i) \\
\acute{c}_{0,t} &= tanh(W_c x_t + b_c) \\
c_{0,t} &= i_{0,t} \times \acute{c}_{0,t} \\
o_{0,t} &= sigmoid(W_o x_t + V_o c_{0,t} + b_o) \\
h_{0,t} &= o_{0,t} \times tanh(c_{0,t})
\end{aligned}
$$

For training, the Adam optimizer (Kingma and Ba, 2014) and categorical cross-entropy loss(Zhang and Sabuncu, 2018) were used. This model had a depth of 2 with each layer having 256 hidden nodes and a dropout of 0.5. We went with a batch size of 24 due to memory restrictions.

3.2 Layer-wise Relevance Propagation

LRP works as an adhoc over the final trained model to calculate the explanation based on the domain of its inputs. LRP takes the weightage of the final classification and distribute this value over the previous layer depending on the contribution/influence of the neuron in the previous layer. This backward pass of sorts recursively distributes the classification weight to the input features, quantifying their importance to the task at hand. For example- if a model predicted the intent to 'flight' with a confidence of 0.8, this 0.8 is then distributed to the neurons of the previous model layer depending on their influence as per equation 1. Recursively this weight/relevance of 0.8 reaches to the feature input layer and the relevance is distributed across the input tokens. For better comprehension we normalize the input token relevance for clarity.

$$\sum_k \frac{a_j w_{j_k}}{\sum_0^j a_j w_{j_k}} R_k \tag{1}$$

Here, j and k denote 2 neurons of consecutive layers, w is weight, and a is the activation. Finally, R denotes the relevance of each neuron. $a_j w_{j_k}$ models the extent of influence of the neuron j in making the neuron k relevant. This influence is then used to distribute the relevance of neuron k to the neurons in the previous layer.R_k is the relevance of the k^{th} neuron at current layer. In this paper, we use ϵLRP which is a modification of base LRP (equation 2) that includes ϵ term in the denominator.

$$\sum_k \frac{a_j w_{j_k}}{\epsilon + \sum_0^j a_j w_{j_k}} R_k \tag{2}$$

The role of ϵ is to still accumulate some relevance even when the influence of the activation of neuron k are weak or contradictory i.e. if R_k is very small then each of the relevance it provides to the neuron(s) j is negligible. The ϵ term helps to maintain mathematical cohesion in case the relevance reaches zero. As ϵ becomes larger, only the most salient explanation factors survive the absorption. This typically leads to explanations that are sparser in terms of input features i.e. the weight distributed is more focused on the important features and all the irrelevant features(stopwords etc) get near zero weightage.This makes the weight distribution less noisy as we can easily focus on the relevant input features. So in conclusion we can say ϵLRP results in sparser and less noisy relevances.

3.3 Model

The model used in this paper uses BiLSTM to train the model for the intent classification task and then employs a LRP model for explanation of the inferences done by the model at testing. The backward pass implemented is a single step process that goes over all the layers of the trained model (from final prediction to the input features) and distributes the weight of the prediction over the input features. Finally, the input feature(s) with highest relative weight are considered responsible for the output of the model.

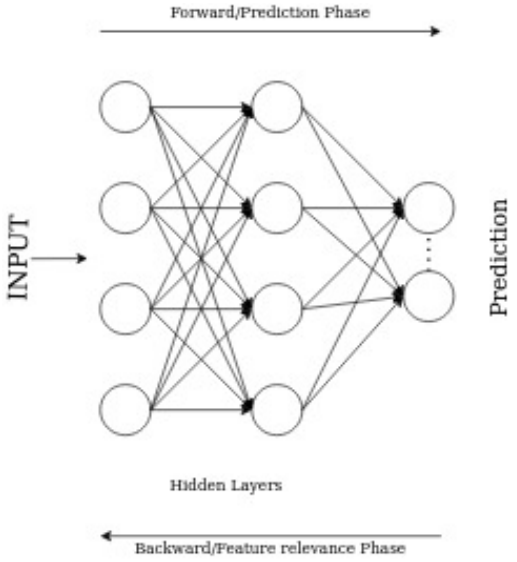

Figure 1: Basic representation of prediction phase (single forward pass) and feature relevance phase (single backward pass) of the BiLSTM + ϵLRP architecture

4 Dataset and Experimental Setup

In this section, we present the details of the datasets and experimental setup.

4.1 Dataset

We use the following benchmark datasets for the experiments.

ATIS: ATIS (Airline Travel Information System) (Hemphill et al., 1990) is a dataset of airline customer service with multiple user utterances and corresponding Intents. The dataset includes 4,478 utterances in training, 500 utterances for validation and 893 utterances in the test set. The data is annotated with 17 intent classes, *viz.* 'flight', 'airfare', 'airline', 'ground_service', 'quantity', 'city', 'abbreviation', 'aircraft', 'distance', 'ground_fare', 'capacity', 'flight_time', 'meal', 'flight_no', 'restriction', 'airport', 'cheapest'. We removed 23 instances labeled with more than one intent.

MultiDoGO: MultiDoGO dataset (Peskov et al., 2019) comprises of six domains, *viz.* airline, fastfood, finance, insurance, media and software. The dataset has two formats, annotated and unannotated. The unannotated version contains 86K conversations, while the annotated version contains 15,000 conversations with 2,500 for each domain. We focus on the user utterances of two sub-domains of airline and finance for intent classification with 38 classes. We use the training, validation and test sets, comprising of 29,742, 4,260 and 8,488 utterances, respectively.

4.2 Experimental Setup

For the experiment, we train the BiLSTM model. We train the model with epochs set to 5, 10 and 20. This is done to avoid any possible overfitting scenario. Adam optimizer and categorical cross-entropy loss were used. To represent the word vectors, a 256 dimensional (non-pretrained) vectors were used. Inference is then generated on the test data, where we primarily focus on the misclassification. The ϵLRP method is then executed on these misclassification to find why these utterances were predicted incorrectly.

4.3 Results and Analysis

The BiLSTM model is trained on the above mentioned datasets (ATIS airline dataset, MultiDoGO airline subdomain dataset and MultiDoGO finance subdomain dataset). We demonstrate the results in Table 1. Then, ϵLRP is used as an adhoc model to gain insights on misclassification. The interesting cases are highlighted.

Table 1: Results of BiLSTM trained on 3 datasets (ATIS airline dataset, MultiDoGO airline subdomain dataset and MultiDoGO finance subdomain dataset)

Dataset	Accuracy	Precision	Recall
ATIS	0.93	0.93	0.93
MultiDoGO Airline	0.91	0.91	0.91
MultiDoGO Finance	0.89	0.89	0.89

While we closely look at the misclassified cases, we see that most of the misclassifications in the **ATIS dataset** are a result of being predicted as belonging to the 'flight' class instead of the actual

class. This can be attributed to the disproportionate training data where 3388 of the 4478 training utterances are for the 'flight' class. This misclassification seems to be due to named entities like locations being taken as a shortcut to classify utterance as intent 'flight'. For example, in Figure 2 one can see that the presence of location tokens (in blue) collectively lead to misclassification as intent 'flight' even though the model knows that the tokens 'meal' and 'cities' (in red) play important role in the classification process (tokens highlighted with blue are responsible for current prediction, the ones with red highlight the second most likely class). This structure of sentences comprise of 4 of the 6 test examples for intent 'meal', all 4 of which are misclassified. The only 2 correctly classified utterances are the ones that do not mention the cities i.e. 'are meals ever served on tower air' and 'are snacks served on tower air'.

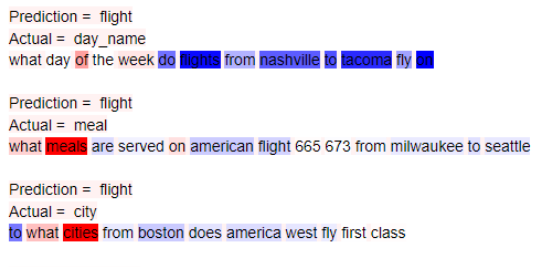

Figure 2: Examples of misclassifications on the ATIS dataset

For **MultiDoGO airline sub domain**, majority of misclassifications seem to arise from the model paying heavy weightage to the tokens 'ok' as 'confirmation', and thanks for 'thankyou' as the intent class, as shown in Figure 3. There are also a few cases of model being confused between the intents 'getseatinfo' (asking for seat details, ex- I want to know seat no) and 'changeseatassignment' (change the curreent seating, ex- I want to get window seat) due to having similar tokens in the training data. Also for the utterance 'thank you sir but i would like to have a middle seat as i do not like a window seat' this direct alignment of the word thank to the intent 'thankyou' leads to misclassification even though the model pays attention to the tokens relating to the correct intent 'changeseatassignment'. We found that this association can be lowered by introducing more examples of similar structure to above utterance but that leads to some instances of 'thankyou' intent to 'rejection' intent(ex- thank you so much nothing more bye). For the first example shown the misclassification is negligible in the context that the same utterance 'ok thanks' is labelled as both 'thankyou' and 'confirmation' in the training data.

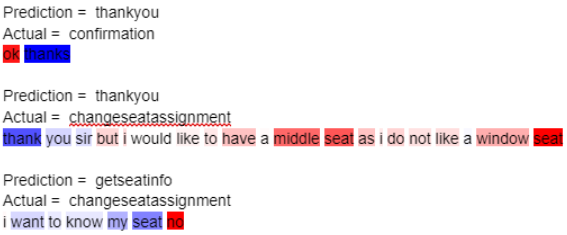

Figure 3: Examples of misclassifications for the MultiDoGO data

For **MultiDoGO finance dataset**, on the other hand, is filled with misclassifications that seem to be right when going through human evaluation. For example, in Figure 4, examples 1 and 3 can be said to be somewhat correctly predicted (since we are looking at them as individual utterances instead of entire dialogue) even though the actual intents are different. For example 2, the evaluation could go either way depending on preferences of the annotator and the evaluators as all the closing greeting examples have the word thanks in them and are very overlapping in their intention.

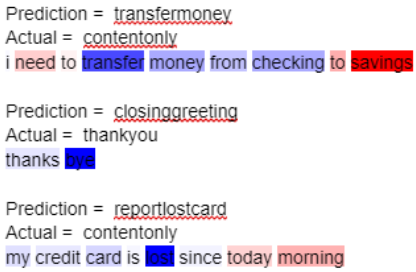

Figure 4: Examples of misclassification on MultiDoGO finance intents

Going through all these datasets, we summarize that there are a lot of inferences that can be drawn with respect to the incorrect classification. The issues arising due to the unbalanced dataset results in forcing the model to pay high attention to some specific tokens. We see the benefits of saliency based methods as it highlights the tokens responsible for the classification. This not only helps us understand the reason for misclassification but also can highlight cases where the data might be incorrectly annotated, resulting in the possibility to improve the quality of dataset along with the classification process.

5 Conclusion and Future work

In this paper, we have attempted to build an explainable intent detection model with the saliency based methods. The model is able to identify the appropriate and relevant features used for intent classification. We also discussed some issues with these approaches, most of which deal with the fact that the saliency techniques calculate the feature importance (which constitute an explanation) as an adhoc measure.

Saliency does have quite a few benefits of itself. The modular nature of the implementations provides a degree of model-agnostic behaviour which allows us to treat the black boxed nature of the deep learning models as an afterthought and focus entirely on the performance. After the model is trained and tuned, we applied the saliency techniques for determining the feature relevance. This also ensures that we can apply different saliency techniques for the same base model and the same saliency technique to different models allowing for a high degree of robustness.

However, even for saliency it is not necessary that the importance assigned to a feature is, in fact, due to the relevance of the features but could simply be a result of the underlying issues with the technique used. For example, in occlusion based methods, if we remove a feature, it is possible that the change in the prediction is just the result of the new input not being in the format the model expects(Kindermans et al., 2019).

For future work, we plan to use the feature importance information and use it to retrain the model in such a way to reduce misclassification. One such method could be to use the important features of misclassifications to help identify which kind of data to add, to improve the performance of the model further. However, such a method needs to be done in such a way that the incorrect predictions do not get corrected at the cost of misclassification of originally correctly predicted utterances. Another such method could be to identify the nodes which are more relevant to a highly misclassified intent and boost those neurons to improve model performance. However, this also needs to make sure the nodes we are associating with a particular intent do not have high influence on other intents as well, as that might lower the accuracy of some other intent.

6 Acknowledgement

Ratnesh Joshi gratefully acknowledge the UGC-JRF NET Fellowship (No. 3601/(NET-JULY2018)). Authors acknowledge the generous support of IMPRINT-2 Project "Sevak-An Intelligent Indian Language Chatbot" Grant No-IMP/2018/002072).

References

Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. 2015. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 10(7):e0130140.

Misha Denil, Alban Demiraj, and Nando De Freitas. 2014. Extraction of salient sentences from labelled documents. *arXiv preprint arXiv:1412.6815*.

Ruth C Fong and Andrea Vedaldi. 2017. Interpretable explanations of black boxes by meaningful perturbation. In *Proceedings of the IEEE international conference on computer vision*, pages 3429–3437.

Jerome H Friedman. 2001. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232.

Christopher Grimsley, Elijah Mayfield, and Julia R.S. Bursten. 2020. Why attention is not explanation: Surgical intervention and causal reasoning about neural models. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 1780–1790, Marseille, France. European Language Resources Association.

Charles T Hemphill, John J Godfrey, and George R Doddington. 1990. The atis spoken language systems pilot corpus. In *Speech and Natural Language: Proceedings of a Workshop Held at Hidden Valley, Pennsylvania, June 24-27, 1990*.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional lstm-crf models for sequence tagging. *arXiv preprint arXiv:1508.01991*.

Sarthak Jain and Byron C Wallace. 2019. Attention is not explanation. *arXiv preprint arXiv:1902.10186*.

Pieter-Jan Kindermans, Sara Hooker, Julius Adebayo, Maximilian Alber, Kristof T Schütt, Sven Dähne, Dumitru Erhan, and Been Kim. 2019. The (un) reliability of saliency methods. In *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, pages 267–280. Springer.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Scott M Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. In *Proceedings of the 31st international conference on neural information processing systems*, pages 4768–4777.

Denis Peskov, Nancy Clarke, Jason Krone, Brigi Fodor, Yi Zhang, Adel Youssef, and Mona Diab. 2019. Multi-domain goal-oriented dialogues (multidogo): Strategies toward curating and annotating large scale dialogue data. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4526–4536.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2018. Anchors: High-precision model-agnostic explanations. In *AAAI Conference on Artificial Intelligence (AAAI)*.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In *International Conference on Machine Learning*, pages 3319–3328. PMLR.

Sandra Wachter, Brent Mittelstadt, and Chris Russell. 2017. Counterfactual explanations without opening the black box: Automated decisions and the gdpr. *Harv. JL & Tech.*, 31:841.

Matthew D Zeiler and Rob Fergus. 2014. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer.

Zhilu Zhang and Mert R Sabuncu. 2018. Generalized cross entropy loss for training deep neural networks with noisy labels. In *32nd Conference on Neural Information Processing Systems (NeurIPS)*.