EG UK Computer Graphics & Visual Computing (2024), pp. 1–9
A. Slingsby and D. Hunter (Editors)

Visual Interpretation of Tagging: Advancing Understanding in Task-Oriented Dialogue Systems

Yazhuo Zhou[1], Y. Xing[1], A. Abdul-Rahman[1], and R. Borgo[1]

[1]King's College London, United Kingdom

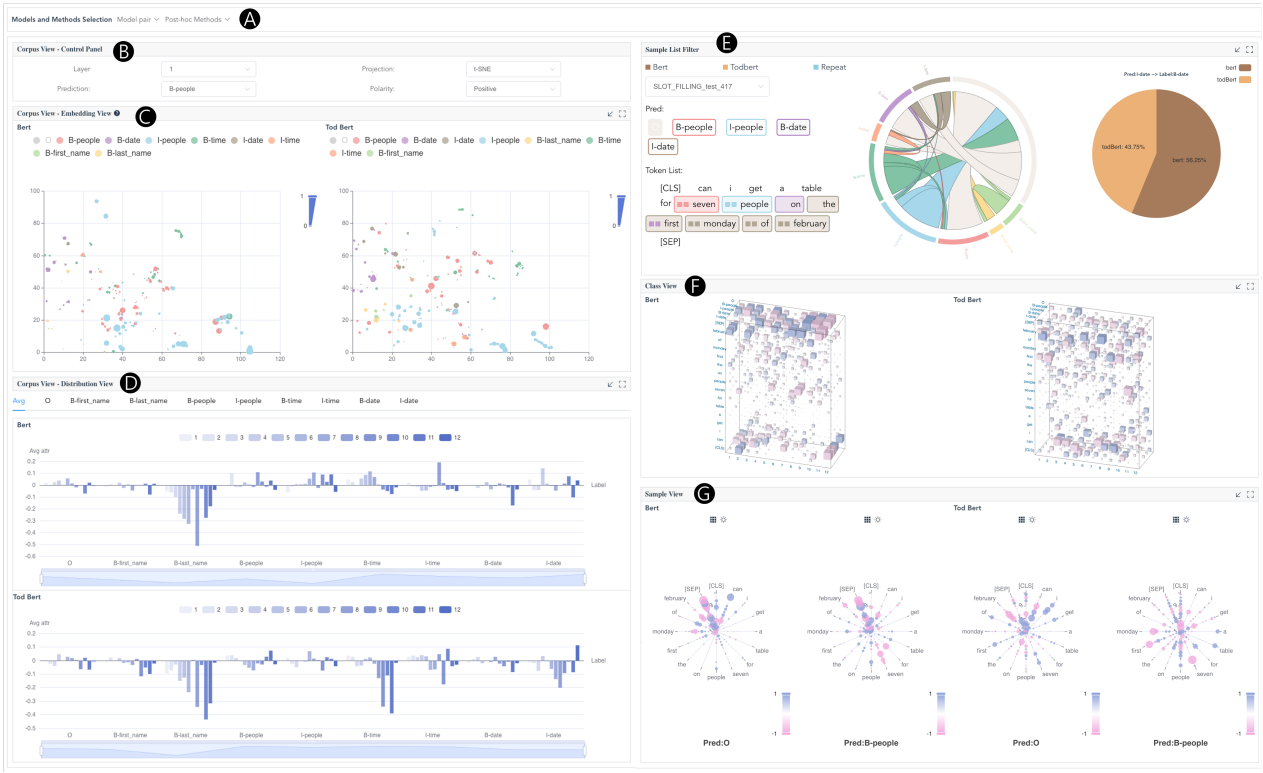

Figure 1: *The interface of Visual Tagging Interpretation System*

Abstract

In task-oriented dialogue systems, tagging tasks leverage Large Language Models (LLMs) to understand dialogue semantics. The specifics of how these models capture and utilize dialogue semantics for decision-making remain unclear. Unlike binary or multi-classification, tagging involves complex multi-to-multi relationships between features and predictions, complicating attribution analyses. To address these challenges, we introduce a novel interactive visualization system that enhances understanding of dialogue semantics through attribution analysis. Our system offers a multi-level and layer-wise visualization framework, revealing the evolution of attributions across layers and allowing users to interactively probe attributions. With a dual-view for streamlined comparisons, users can effectively compare different LLMs. We demonstrate our system's effectiveness with a common task-oriented dialogue task: slot filling. This tool aids NLP experts in understanding attributions, diagnosing models, and advancing dialogue understanding development by identifying potential sources of model hallucinations.

CCS Concepts

• ***Human-centered computing*** → ***Visual analytics;***

1. Introduction

Large Language Models (LLMs) have significantly improved task-oriented dialogue tasks but remain opaque, complicating transparency. With increasing dialogue complexity, the need for high-level semantic explanations has grown. Tagging tasks, such as slot filling and intent recognition, are crucial within the dialogue pipeline, enhancing understanding and response accuracy. Slot filling is a task where specific words or spans are tagged as a class of concept, such as identifying "Paris" as the "destination" slot in the sentence "Book a flight to Paris". Unlike binary and multi-class classification, tagging tasks require independent predictions for each feature, leading to a complex many-to-many relationship that complicates interpretability. It's crucial to consider how each feature, through interactions with others, influences multiple predictions, revealing intricate interaction patterns.

Current methodologies use feature attribution to explain model behaviors post-hoc [DK21], but interpreting these attributions in tagging tasks is challenging. Attribution assigns a score to each input feature (e.g., individual words) to indicate its influence on a prediction. A positive score indicates that the feature contributes positively to the prediction, meaning the prediction depends on that feature to some extent, while a negative score implies the opposite. In tagging tasks, a single feature can influence multiple predictions, requiring analysis beyond one-to-one relationships. Each feature's contribution creates complex patterns, with combined effects that may differ from individual impacts. The high dimensionality of embeddings adds to the difficulty of understanding feature influences, complicating attribution and meaningful model interpretation. While some studies provide quantitative evaluations, they often lack detailed explanations of model processes across layers.

To address the challenges of interpreting tagging tasks in task-oriented dialogue systems, we developed a comprehensive, interactive visualization tool that enables deep, multi-level analysis. This tool transforms complex multi-to-multi relationships and high-dimensional attributions into intuitive visual formats, enhancing the clarity of data presentation. Key features include a dual-view capability, which simplifies direct model comparisons and guides model selection, and layer-wise visualization that assists in exploring detailed model behaviors and tracking linguistic feature evolution. Our tool employs a multi-level visualization framework, spanning corpus, class, and sample levels, that allows users to progressively delve deeper into data according to the granularity of information. This approach not only lowers the threshold for understanding complex Large Language Models (LLMs) but also significantly improves their explainability. By providing users with tools to explore and interpret nuanced feature interaction and behaviors within models, our system advances the interpretability of tagging tasks in dialogue systems.

2. Related Work

LLMs in Dialogue The integration of Large Language Models (LLMs) into conversational AI has enhanced the conversational capabilities and engagement of dialogue systems. For instance, DialoGPT, BlenderBot, Lamda, ChatGLM3 and Llama 2-Chat [ZSG*19, SXK*22, TDFH*22, ZLD*22, TMS*23] are trained or fine-tuned on a dataset of dialogues, for understanding and engaging in complex conversational scenarios, and generate human-like coherent responses in specific tasks. Tod-BERT [WHSX20] is a BERT-based model pre-trained on task-oriented dialogue datasets which shows great performance on the Natural Language Understanding (NLU) module of task-oriented dialogue. It remains unclear why they outperform general models to a significant extent. Specifically, it is unknown whether they have acquired additional dialogue skills or dialogue semantics knowledge.

Post-hoc Attribution Methods Explainable AI (XAI) techniques is critical to improve model transparency and trustworthy. Existing works attempt to enhance the interpretability of LLMs by deploying various post-hoc methods and examining their performance [DJR*19, DK21, YSHC21, WSP*22, JG20]. These methods are used to quantify the contribution of each feature to the model output. Then evaluate the explanation quality from two perspectives, faithfulness and plausibility. Above works provided final evaluation results and lack of intermediate information complicates the human understanding of explanations. They also focus on representing low-level language syntactic interpretation of general classification tasks, our focus is more inclined towards the specific semantics inherent in dialogue context. Our system provides Input x Gradient [LCHJ16], and Layer Gradient x Activation [web] to output each layer attribuition for users to explore.

Visualization for Explainable LLM Visualizations can effectively help understand LLMs. The most common model-specific visualization to reveal the model inner working is attention map [Vig19b, Vig19a, **?**, PNJ*19, DWB20, YCW*23]. LLM interpretation through interactive visualization tools [LWY*22, LLL*18, TWB*20, **?**, WTC21, DAB*22] for exploring and diagnosing models, focusing on explaining low-level linguistic features by designing auxiliary classification tasks to understand linguistic knowledge in NLP models. However, these tools prove insufficient for supporting complex conversational semantics comprehension. Some works focusing on visual analysis of representation contextualization [HSG19, SKB*21, SKB*22]. Some visual interpretation tools for specific tasks [SFSvdW23, SSZ*23, WHJ*23], such as QA and Machine translation, which has some alignment pattern between input and output, reveals some higher level semantics. To mitigate these challenges, we propose an interactive visualization tool tailored for comprehensively understanding post-hoc interpretations within dialogue models. We aim to reveal the role of dialogue knowledge in the model's reasoning process.

3. Requirements and Tasks

Our approach is guided by the need of NLP experts to develop interpretability methods that enhance the understanding of model behaviors. A significant challenge in this domain is that users often struggle to comprehend the attributions generated by current methods, as the models' reasoning may differ from human logic and involve complex multi-factor interactions and influences on predictions. This discrepancy can lead to biases in model understanding and offer limited support for model selection. Therefore, an approach that provides more intuitive and straightforward explanations is necessary.

3.1. Model, Data and Methods Selection

To analyze attributions in LLMs, we encode input tokens into embeddings, generate predictive outputs, and apply attribution methods to these outputs for attribution data. Our selection spans BERT-like autoencoders, GPT-like autoregressive models, general models like BERT and GPT-2, and specialized dialogue models like TOD-BERT and DialoGPT, ensuring comprehensive comparison. Dialogue model means the models are pre-trained by dialogue corpus not general text. For our study, we chose a slot filling task within task-oriented dialogues, using the RESTAURANT-8K dataset [CFG*20]. This dataset focuses on slot predictions. For the attribution methods, we used Gradient x Input [LCHJ16], which assigns importance based on multiplying the gradient by the input embedding vector at the last layer. We calculated the intermediate layer explanations using Layer Gradient x Activation, measuring feature contributions quantitatively. These attributions were implemented using the Captum library [web].

3.2. Design Requirements and Tasks

Our analysis commenced with informal interviews with two domain experts, focusing on their routine interactions with interpretability methods. Additionally, we reviewed prevalent techniques used by researchers to decode model behavior through attribution analysis and visualization (refer to section 2). Our overarching aim is to simplify the analysis of attributions, unravel the intricate relationships between features and predictions, enhance the comprehension of model mechanisms, and facilitate detailed comparisons between diverse models.

R1 - Intuitive Visualization of Attribution Users need visual tools to intuitively understand how multiple features impact multiple predictions simultaneously in tagging tasks, with interactive exploration capabilities. The gap between model logic and human intuition requires attributions to be in a straightforward visual format.

R2 - Generalization Across Models and Attribution Methods Users need a system that generalizes across various LLMs and interpretation methods for tagging tasks. It reduces biases from different attribution methods, and enhances the faithfulness of attributions. Additionally, it facilitates model diagnostics and improvement.

R3 - Multi-Level Visualization for Attribution Data Effective tagging task analysis requires multi-level visualization. Users need to identify global patterns or biases at the corpus level and spot anomalies at the class or token level to understand how models handle specific semantic roles. Due to the complexity of many-to-many feature-to-prediction mappings, a hierarchical approach to decomposition is essential.

R4 - Layer-wise Attribution Analysis for LLMs LLMs' multi-layered structures require observing attribution changes at each layer. This enables understanding where the model captures or overlooks specific dialogue semantics and tracks the evolution of its behavior.

We now discuss the specific tasks our system is designed to facilitate, based on the needs of NLP researchers and data scientists who require a deep understanding of attributions and models.

T1 - Attribution analysis Understanding and interpreting attributions remains challenging. Statistical analysis of attribution distributions and gradient magnitudes helps identify influential tokens, measure their contributions, and pinpoint regions of the input space where the model's predictions are most sensitive. This process can reveal key features or noise sources and aids in semantic analysis by identifying tokens with substantial explanatory power, especially for specific concepts like dialogue system slots (requires **R1**, **R3** and **R4**).

T2 - Model behavior tracing Attribution assigns importance scores to input token embeddings, providing detailed token-level attribution that is essential for understanding the model's decision-making process. This enables users to identify how the model prioritizes linguistic features and patterns, which is critical for both optimizing the model and diagnosing issues. Users could track changes in attribution scores in layer-wise manner and examine how the model reacts to features in the input (requires **R2**, **R3** and **R4**).

T3 - Model Comparison Post-hoc attribution methods offer a unified view for comparing token contributions across models. Key tasks include identifying differences between models, understanding the factors driving these differences, and assessing their impact on model behavior and predictions. This comparison is crucial for model optimization, selection, and development, ensuring model suitability for specific tasks (requires **R2**, and **R4**).

4. System Overview and Application

Modules	Description
Model and Methods Selection	Display the options of model pairs and attribution methods
Corpus View - Embedding View	Visualizes the dataset by projecting layer-wise embeddings down to two dimensions, incorporating attribution values. It provides options to select the layers, attribution target prediction, polarity, and dimension reduction methods.
Corpus View - Distribution View	For a selected prediction, display the average token contributions (attribution values) from each slot type across all 12 layers. Include a tab to select a specific slot category and show the detailed attribution distribution of tokens within that slot to the chosen prediction across 12 layers.
Sample List Filter	This module includes a chord diagram that filters instances of misclassification between two models by clicking on directed arcs. Additionally, a pie chart displays the distribution of these misclassified samples across different models. When a user selects an instance ID, the system displays detailed labels and predictions for each token by both models, along with all unique predictions generated by them.
Class view	Presents the attribution distribution of all tokens within an individual instance across all unique predictions and all layers using a 3D time-space cube. This visualization allows users to interactively explore multi-feature and multi-prediction interactions by zooming and rotating the display.
Sample view	Displays polar and traditional heatmaps for token-based feature attribution, illustrating the evolution of information for an individual instance towards a specific prediction in a layer-wise manner. This visualization is activated upon the selection of a prediction in Sample List Filter.

Table 1: *Built-in modules in the system.*

In the following section, we detail the visualization and interaction design of our system. The system's design aligns with the requirements specified in section 3, ensuring it meets user needs effectively. For clarity, we have summarized the system's modules

in Table 1, providing a quick guide to its functionalities and features, along with an overview of its design rationale.

Design Paradigm Our visualization system, schematically represented in Figure 2, is designed to meet the complex requirements of experts working with the explainability of LLMs. It was developed through a rigorous collaborative process with domain experts and has undergone iterative refinements. Enhancements were directly influenced by detailed expert feedback focusing on usability, functionality, and interperability.

In terms of visualization encoding, several consistent elements across all views reinforce clarity and understanding: 1) Color Encoding: Pink and blue are used to represent negative and positive attribution values, respectively, facilitating quick identification of attribution polarity. The color scheme also categorizes the 9 slot types consistently across views. 2) Magnitude Representation: Attribution magnitudes are encoded with a sequential color scale and element size; darker colors and larger sizes indicate higher magnitudes, providing instant visual cues about the impact of attributions. 3) Dual View for Model Comparison: This feature allows easy comparison between models like BERT and TOD-BERT, which have the same architecture but different pre-training corpora, by juxtaposing them for effective analysis. 4) Layer Representation: All views include a display of the model's 12 layers, from the initial layers at the bottom (e.g. layer 1-3) to the final layers at the top (e.g. layer 10-12).

Workflow This system supports multiple workflows for exploratory analysis, allowing users to initiate their exploration from any primary view according to their analytical goals. A typical workflow begins with users selecting model pairs and attribution methods from the Model and Methods Selection dropdown menu. Exploration starts at the Corpus View for a global dataset overview, then transitions to the Sample List Filter to focus on specific types of misclassification instances, utilizing a chord diagram for diagnosing model vulnerabilities. Users can then move to the Class View for a 3D visualization of token attributions across model layers and all predicted classes, highlighting dynamic token interactions. For a detailed exploration of specific predictions, users proceed to the Sample View, which provides an in-depth token-level analysis for each model-instance-prediction combination.

4.1. Corpus View

The Corpus View of our system is designed to provide a comprehensive overview of the dataset, showcasing attribution of all instances and their tokens. The Corpus View incorporates a set of four dropdown menus for data filtering, enhancing the user's ability to tailor the visualization to their analytical needs in Control Panel (Figure 1B), these options enable further refinement within the Corpus View. The `Layer` dropdown, ranging from 1 to 12, facilitates layer-specific exploration, while the `Prediction` is utilized to filter tokens from all instances with a selected prediction in their prediction sequence. The `Projection` menu allows users to select the dimensionality reduction technique for data presentation. We provide users with three dimensionality reduction technique options: PCA [LU20], UMAP [MHM18], and t-SNE [VdMH08]. The `Polarity` options are available for finding tokens that contribute positively or negatively to a specific prediction. Then the

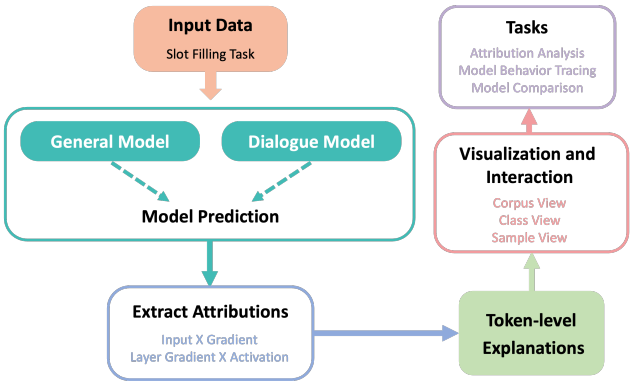

Figure 2: *Schematic representation of the system components: We have fine-tuned both general and dialogue-specific models on slot-filling data, subsequently utilizing these fine-tuned models for slot prediction on the test dataset. Following this, we calculated attributions for all predictions in a layer-wise manner and converted these attributions into token-level explanations. Finally, we loaded the attribution data into our system to support users in achieving the three designated tasks.*

Corpus view is divided into two parts: the Embedding View shown as Figure 1C and the Distribution View as Figure 1D , each offering unique insights through visual design and interactive features.

Embedding View This component is presented as a dual scatter plot, where each point represents an individual token as show in Figure 1C. The position of these points is determined by the word embeddings of each token. An interactive slot type multi-selector, represented by a series of colored circles at the top of the scatter plot, allows users to filter tokens by their assigned labels. Each circle, when filled with color, signifies an active selection of that category; users can toggle the selection of a label, rendering the circle gray to exclude that label category from the visualization. Tokens are plotted as 2D points, colored by their labeled slot classes, with the point size indicating the absolute attribution value of the token, as the attribution values range from -1 to 1. A bidirectional value slider refines the display to show points within a specified range of attribution values. Detailed token information, including attribution value, layer, and prediction, is disclosed when users hover over individual points.

Distribution View The Distribution View (Figure 1D) visualizes the distribution of attributions across slot classes for a selected target prediction. Under the Avg tab, two bar charts show the average attribution of tokens within each slot category across 12 layers. This average, denoted as $\bar{A}^i_{P,L}$, for a specific prediction class P and true label class L at layer i, is computed by summing the attribution scores $\text{attr}^i_{P,j}$ for all tokens of type L towards the prediction class P and dividing by the total number n of such tokens in the layer. This provides an averaged layer-specific view into how token classes influence predictions P. Mathematically, the formulation is expressed as: $\bar{A}^i_{P,L} = \frac{1}{n}\sum_{j=1}^{n} \text{attr}^i_{P,j}$. We also provide a variety of interactive features to facilitate detailed exploration by users. Hovering over a slot category on the x-axis displays a summary of the average attribution across the 12 layers, offering a layer-by-layer comparative

analysis to identify influential slot categories. Selecting a slot from the other tabs updates the view to show a combined scatter and violin plot (Figure 4), which details the distribution and scale of attributions for each class. Above each bar chart, a series of toggles labeled from 1 to 12 correspond to the model's 12 layers. Users can click these toggles to control the visibility of bars associated with each layer, enabling them to isolate or compare attributions across specific layers. Additionally, a slider located below each bar chart allows users to zoom in and out, providing a closer look at specific categories. The Distribution View enables users to explore the distribution and impact of each token class across the entire corpus on model predictions.

4.2. Sample List Filter

The Sample List Filter module (Figure 1E) features a directed chord diagram crucial for identifying misclassifications. Each arc represents a misclassification type, with direction from predicted to actual label, indicating tokens mispredicted from one slot to another. The width of each arc shows the volume of misclassified tokens. Hovering over an arc displays details such as predicted and true labels and the number of affected tokens. Clicking an arc filters and displays instance IDs with that error in a dropdown, with colored squares indicating errors unique to or shared by both models. This interaction also updates a pie chart on the right showing the error distribution across models. After selecting an instance, the Class View and Sample View visualize relevant diagrams. The interface allows multi-selection of unique predictions which made by the two models for the chosen instance to customize the Sample View display. Below, the Token List shows each token's true label via background color, and both models predictions with two colored squares. Tokens of Class `O`, which indicate no slot, are uncolored for clarity. The Sample List Filter thus streamlines the detection of prediction errors and enhances analysis with detailed token-level insights.

4.3. Class View

The Class View in our system (Figure 1F) draws inspiration from the space-time cube framework traditionally used in social geography to represent 3D Euclidean space [BPF14]. We adapt this by treating model layers as sequential time steps. In our 3D visualization, the x-axis represents the 12 layers of the model, the y-axis enumerates all tokens of a selected instance, and the z-axis displays all unique predictions, effectively mapping the complex relationships within the data. Each cube within this 3D matrix varies in color and size based on the attribution value and its polarity, highlighting the impact of each token across multiple predictions. This visualization helps users understand the complex multi-to-multi relationships inherent in tagging tasks, providing insights into how features dynamically interact across layers and predictions. To mitigate challenges like complex navigation and visual occlusion on a 2D screen, we incorporate zoom and rotate interactions to enhance user experience. Additionally, the Sample View can be considered as a collapsed version of the Class View along its prediction axis. This bridges the gap between the global perspective offered by the Class View and the detailed analysis facilitated by the Sample View.

4.4. Sample View

The Sample View, as shown in Figure 1G, extends the analytical capabilities of the Class View with two heatmap options, selectable via the Sample List Filter. The traditional Heatmap expands the Class View along the prediction axis for detailed layer-by-layer analysis of an instance's attributions for specific predictions. Conversely, the Polar Heatmap, with its radial layout, excels in handling long instances by increasing the density of radii rather than extending the visual space. This design keeps all information visible within a single view without the need to scroll. Both options offer interactive hover functionality for detailed insights and are supported by a bidirectional slider for filtering attribution values, enhancing user control. A tab system with `Grid` and `Sun` options allows toggling between layouts. When multiple predictions are selected, the Sample View adapts to display corresponding heatmaps for both models, providing a comprehensive view of the attributions across the model's layers, enriching understanding of the tagging process.

5. Evaluation

Throughout the development process, we engaged with domain experts and conducted iterative testing and case studies on the prototype. This section presents three cases from our research, illustrating practical applications and insights. We then detail interviews with two domain experts after refining the tool, evaluating its usefulness and usability. These assessments provide a comprehensive view of the tool's effectiveness in real-world scenarios and its value to NLP practitioners and researchers.

5.1. Case Study

We conducted case studies involving three tasks with experts E1 and E2. Both of them are interested in attributions analysis. E1 would also like to explore the model behavior in slot filling task and E2 is curious about the difference of general model and dialogue model. Due to the time-consuming nature of computing and rendering attributions across every layer, we precomputed and stored intermediate values. This allowed for efficient loading and processing during the studies, demonstrating our system's ability to efficiently disentangle the mechanisms of sequence tagging tasks using LLMs.

5.1.1. Attribution analysis

After E1 and E2 selecting the BERT & TOD-BERT model pair and the Input X Gradient method, both E1 and E2 conducted attribution analysis. E1 selected t-SNE projection method and `B-people` as a prediction of interest. E1 observed that tokens actually labeled as `B-people` and `I-people` were not only more prevalent but also displayed significantly larger point sizes, indicating greater attribution as shown in Fig 3. E2 also observed this pattern across other categories such as `B-date`, where tokens labeled `B-date` and `I-date` similarly dominated the visualization in both number and scale. The visualization effectively highlights the semantic alignment between tokens and their predictive impact, as seen in the clear clustering and sizing of related semantic types. E1 concluded that tokens of each category demonstrate significant explanatory

power for predictions corresponding to related semantic types, indicating that slot semantics can serve as a robust explanation.

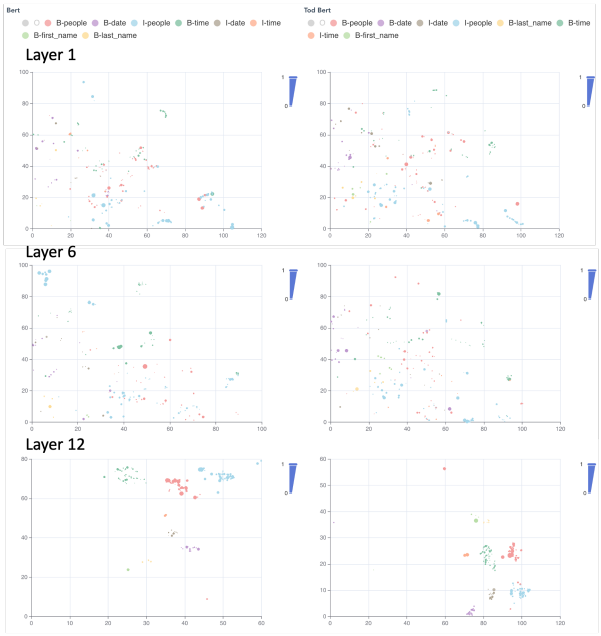

Figure 3: *The embedding scatter with attribution value of BERT and TOD-BERT towards B-people prediction.*

Figure 4: *Combined scatter and distribution plot present attribution value distribution of tokens labelled as B-last_name towards B-people prediction.*

After obtaining a global view of the attribution distribution across each category, E1 accessed the Corpus - Distribution View to examine detailed numerical values. Across the test dataset, E1 observed that tokens labeled as `B-last_name` showed significant negative average attributions toward the prediction of `B-people`. This pattern was consistent in both the Bert and TOD-Bert models, as depicted in Fig 1D, where `B-last_name`, `B-time`, and `I-date` in the TOD-Bert model also demonstrated notable negative contributions. Such negative attributions imply that the presence or high values of these features may decrease the likelihood of correctly predicting the `B-people` category. Subsequently, E1 analyzed the detailed attribution distribution of `B-last_name` tokens towards the `B-people` prediction, as shown in Fig 4. E1 noticed that the negative outliers are primarily exhibited in the model's bottom (layers 1-3) and top layers (layers 10-12). This pattern is more pronounced in the TOD-BERT model, which is specifically trained on conversational data. E1 inferred that dialogue-oriented models excel at capturing not just positive conversational semantics but also negative semantic cues due to their training on diverse conversational nuances and contextual complexities. E1 hypothesizes that this exposure enhances the models' sensitivity to negative cues. It is crucial for refining the models by identifying misleading features.

E1 believes that the findings obtained using our tool indicate a potential solution to address a key challenge in model interpretability. As identified by [JG20], one of the challenges in model interpretability is the lack of human-annotated explanations for evaluating the plausibility of explanations. E1's observations suggest that attributions could be indicative of human-understandable concepts, such as slots in this case, potentially bridging the gap between raw attributions and human interpretability. The ability of the system to reveal these correlations paves the way for developing more intuitive methods for annotating and evaluating explanations in alignment with human knowledge.

5.1.2. Model behavior tracing

With an enhanced insight into the distribution of model attributions, E1 embarked on tracking the behavior of the models through a comprehensive examination of their layer-wise capabilities. This in-depth analysis enabled the experts to meticulously track how attributions varied across each layer, providing a clearer understanding of the models' interpretative processes and the underlying mechanisms that contribute to their decision-making abilities. From Fig 3, depicting Layers 1, 6, and 12, it is evident that moving from lower to higher layers within the models, there is a noticeable increase in the formation of distinct clusters by semantic slot types. In the initial layers, tokens are more dispersed, indicating a preliminary stage of feature recognition where the model is beginning to differentiate between various types of information but has not yet formed clear associations. As E1 advanced to Layer 6, the scatter points begin to show a tendency toward clustering, suggesting that the models are developing a deeper semantic understanding and starting to group tokens by their functional roles within the dialogue. In Layer 12, the clusters become significantly more defined. This demonstrates that by the top layers, the models have effectively captured advanced dialogue semantics, enabling them to make more nuanced predictions about token roles. E1 observed that tokens labeled as `B-people` and `I-people`, both referring to the semantic category of people, form clusters that are close yet

clearly distinct, highlighting the model's ability to differentiate between semantically similar tokens.

E1 analyzed a specific misclassification shown in the chord diagram in the Sample List Filter, focusing on tokens labeled as `B-date` but predicted as `I-date`. Notably, in the sample with ID `SLOT_FILLING_test_417`, the token "on" was incorrectly predicted by both models as `O` rather than `B-date`, highlighting a systemic issue with contextualizing time-related prepositions. This error led to subsequent tokens like "the", "first", "Monday", and "February", which should have been part of a continuous date sequence as `I-date`, being inaccurately labeled as `B-date`. E1 observed a cascading effect of this initial misrecognition, with errors extending to the labeling of following tokens. Further examination in the Class view revealed that tokens following "on" had significant attribution impacts, as shown by larger and denser cubes in the top half of the visualization (Figure 1F), indicating the model's difficulty in recognizing slots of tokens following "on". E1 also carefully checked the detailed attribution from the Sample View by selecting the prediction `O` and `B-date` which are two incorrect predictions (Figure 1G). Despite deeper layers capturing more advanced semantics, they still fail to recognize the preposition as a trigger for the date. E1 concluded that enhancing the models' grasp of contextual cues in time expressions could improve accuracy, pointing out that detailed error analysis could help pinpoint specific layers and behaviors for adjustment to boost overall performance in natural language understanding tasks.

5.1.3. Model Comparison

After exploring all tasks within the workflow, E2 summarized observations regarding the distinct behaviors of the two models. Through comparative analysis in the Corpus View, E2 noted that TOD-BERT exhibits a more dynamic attribution pattern across its layers and certain slot classes, suggesting a potentially more nuanced understanding of context. In terms of attribution distribution, BERT demonstrated a relatively stable learning pattern; although it exhibited some limitations in capturing connections between certain tokens with a dispersed distribution, it generally showed fewer extremes. Further investigations using the Class View and Sample View on specific instances revealed differences in how the two models focus on the initial token. BERT primarily relies on the semantics integrated by the [CLS] token for predictions, whereas TOD-BERT shows a more even distribution of attention across all tokens, suggesting that it makes decisions by synthesizing information from all tokens rather than depending primarily on the first token. Inspired by the insights gained through our system, E2 devised a strategy that both models could benefit from training data specifically designed to address observed deficiencies, such as adding examples that involve prepositional phrases and complex slot structures.

5.2. Expert Interview

After several refinements, we assessed the system usefulness and usability through expert interviews combined with the think-aloud protocol. We observed the domain experts' interactions and specific usage of the tool, conducting a qualitative analysis based on the interview transcripts.

Table 2: *Interview Procedure and Duration.*

Order of Procedure	Activities	Duration
Preliminary Preparation	1) Introductory questioning 2) Tool walkthrough	10 min
Task Scenarios	1) Test via predefined tasks	20 min
Reflective Discussions	1) Reflection on the tool 2) Future direction	20 min

5.2.1. Methodology

We interviewed two domain experts E1 & E3, one of whom (E1) had been actively involved in previous discussions during the development phase. Both participants have over eight years of experience in the NLP domain. The interview sessions were conducted remotely via MS Teams, with each session lasting between 30 and 60 minutes. The procedures followed during these sessions are detailed in Table 2.

The sessions began with preliminary preparations where we clarified the objectives and explained the think-aloud protocol. This was followed by a tool overview, introducing the primary panels and their functions, along with a tutorial demonstrating essential interactions and use cases. The core session focused on task scenarios where participants, thinking aloud, shared their screens while exploring a familiar document using the tool. Participants evaluated each visualization for usability—assessing effectiveness, intuitiveness, and user-friendliness—and provided feedback on strengths and weaknesses. They also validated the workflow by completing tasks related to attribution analysis, model behavior tracing, and model comparison using integrated visualizations. This approach aimed to assess how well the visualizations worked together, identify usability issues, and gather suggestions for improving the overall user experience. The sessions concluded with reflective discussions on users' experiences with specific aspects of the tool. Participants shared their overall impressions, highlighting the tool's strengths and areas for improvement, and provided suggestions for additional features or modifications to enhance its usefulness and usability.

5.3. Results

Throughout the evaluation process, participants provided valuable insights into the usability, effectiveness, and areas for improvement for the system. The feedback is summarized based on the tool's components both individually and as a whole, highlighting how they can help in meeting the requirements.

Overall, the system was found to be intuitive and user-friendly. Both participants appreciated the helpful tooltips and clear visual elements that made navigation and understanding easier. The embedding view, distribution view, sample list filter, class view, and sample view each offered unique insights that contributed to understanding model behavior and performance. To enhance learnability for new users, it was suggested that user guides with practical examples be provided for each functionality. This would help users, especially those unfamiliar with specific tasks, quickly understand and utilize the tool effectively.

Corpus Views: The embedding view was considered intuitive and user-friendly, with helpful tooltips enhancing the user experience. Participants could easily observe model attention shifts and identify influential tokens. The size of points representing attribution values was noted for its interpretability. E3 especially praised the interaction for providing a comprehensive understanding of model behavior and mentioning this visualization is insightful: "*in the final layer, different categories converged and clustered effectively.*" Both participants appreciated the distribution view for its detailed comparison feature and error analysis capabilities, which are essential for understanding model behavior. Displaying average values aided in a more accurate understanding of the embedding view. The specific distribution unfolded the details of each bar, making detailed exploration easier (E3). However, performance limitations when handling multiple models and large datasets were noted, indicating a need for scalability improvements (E1, E3).

Sample List Filter: The sample list filter provided a clear and straightforward display of token-level details, predictions, and labels for two models, making it easier to understand prediction deviations. However, the chord diagram was seen as complex by both participants, and E1 recommended adding guidance or arrow effects to improve understanding. E3 noted, "*Displaying the number of instances for each type of error in a comprehensive table would greatly facilitate improved model diagnostics.*" E1 also commented, "*It would be better if the error statistics were separated for each model rather than aggregated in a single chord diagram.*"

Class View: Based on the feedback, the class view was praised for effectively showing the overall distribution of predictions across multiple categories within an instance, aiding in model diagnosis. Participants found the 3D interaction highly useful for rotating, dragging, and zooming to view both overall and detailed information. This component was particularly noted for its ability to facilitate the observation of mutual influences of features on multiple predictions. "*The 3D interaction makes it easy to diagnose the model by observing the feature influences*" (E1).

Sample View: Both participants agreed that the sample view provided a detailed analysis of token-based feature attribution, which helped participants understand model decisions.

Overall Tool: This tool was found to be intuitive and effectively supported understanding model attributions (R1) by both participants. E1 remarked, "*The tool provides significant information about the internal workings of models, which are typically considered black boxes.*" The tool's capabilities in providing generalization across different models and attribution methods (R2) were recognized, although performance limitations highlighted the need for improved scalability. The multi-level visualization approach was particularly praised for its detailed error analysis and layer-wise attribution capabilities (R3 & R4). However, including practical examples and additional context in user guides would enhance the tool's usability and effectiveness, especially for new users (R1). Additionally, improvements to better distinguish between model errors and correct predictions, including clarifying the sources of errors (e.g., misclassification vs. model issues), were suggested by E1. As E1 pointed out, "*You need to know whether a mistake is a misclassification or due to the model's interpretation.*"

Overall, the feedback highlights system's strengths in providing detailed and intuitive visualizations, particularly for error analysis (R3) and layer-wise attribution (R4). Addressing scalability issues (R2) and including practical examples (R1) will enhance the tool's usability and broaden its applicability, ensuring it remains an efficient and effective resource for NLP researchers and practitioners.

Reflections: During interviews, participants praised the system for its effectiveness in visualizing up to varies models and methods across approximately 2000 data instances, while also highlighting scalability challenges in managing extensive attribution comparison analyses, especially for tagging tasks. They noted the critical need for improved scalability in handling larger volumes of data and more complex models, which involves enhancing both visual and computational performance. One participant (E1) stated, "*If scaling can be fixed, then people can easily upload their models and quickly identify what's going on. Within seconds or minutes, you could use this tool across other tasks and datasets, not just slot filling in dialog systems.*" This underscores its potential for broader application across various NLP tasks.

Currently, the system mitigates some performance challenges by pre-calculating and storing results of attribution and dimension reduction, which streamlines the computational process and optimizes web interface rendering, allowing users to interact with the data more fluidly and efficiently. Participants appreciated these efficiencies, noting how they improved usability, but suggested that expanding the system's capabilities would require further improvements. Specifically, they emphasized the need for integrating this system more seamlessly into the workflow pipelines of LLMs, such as automating data preprocessing steps and enabling real-time analysis. Enhancing this integration would improve the tool's utility, ensuring it remains effective for comprehensive model analysis and interpretation across diverse tasks and larger datasets.

6. Conclusion and Future work

In this work, we have identified three key user tasks and met four critical requirements for NLP experts, designed to facilitate their understanding of attributions, model comprehension, selection and diagnosis. To achieve these objectives, we've implemented tailored visual encodings within our system. Notably, the multi-level visualization facilitates a layered, detailed exploration of models, while the dual-view feature supports intuitive, side-by-side comparisons of model pairs. Additionally, our system accommodates diverse analytical methods through its support of various workflows.

Our evaluations demonstrate that the system provides unique insights, particularly in revealing advanced semantic information, such as slot semantics in dialogue contexts. Looking ahead, we plan to further explore higher-level dialogue-specific features for generative LLMs and address scalability challenges to expand the system's capabilities. We remain committed to enhancing the tool's utility, ensuring it continues to be a valuable resource for experts navigating the complexities of LLMs. Through these efforts, we aim to make substantial contributions to the field of Explainable AI, pushing the boundaries of model interpretation and analysis.

References

[BPF14] BACH B., PIETRIGA E., FEKETE J.-D.: Visualizing dynamic networks with matrix cubes. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems* (2014), pp. 877–886. 5

[CFG*20] COOPE S., FARGHLY T., GERZ D., VULIĆ I., HENDERSON M.: Span-convert: Few-shot span extraction for dialog with pretrained conversational representations. *arXiv preprint arXiv:2005.08866* (2020). 3

[DAB*22] DELAFORGE A., AZÉ J., BRINGAY S., MOLLEVI C., SALLABERRY A., SERVAJEAN M.: Ebbe-text: Explaining neural networks by exploring text classification decision boundaries. *IEEE Transactions on Visualization and Computer Graphics* (2022). 2

[DJR*19] DEYOUNG J., JAIN S., RAJANI N. F., LEHMAN E., XIONG C., SOCHER R., WALLACE B. C.: Eraser: A benchmark to evaluate rationalized nlp models. *arXiv preprint arXiv:1911.03429* (2019). 2

[DK21] DING S., KOEHN P.: Evaluating saliency methods for neural language models. *arXiv preprint arXiv:2104.05824* (2021). 2

[DWB20] DEROSE J. F., WANG J., BERGER M.: Attention flows: Analyzing and comparing attention mechanisms in language models. *IEEE Transactions on Visualization and Computer Graphics 27*, 2 (2020), 1160–1170. 2

[HSG19] HOOVER B., STROBELT H., GEHRMANN S.: exbert: A visual analysis tool to explore learned representations in transformers models. *arXiv preprint arXiv:1910.05276* (2019). 2

[JG20] JACOVI A., GOLDBERG Y.: Towards faithfully interpretable nlp systems: How should we define and evaluate faithfulness? *arXiv preprint arXiv:2004.03685* (2020). 2, 6

[LCHJ16] LI J., CHEN X., HOVY E., JURAFSKY D.: Visualizing and understanding neural models in NLP. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (San Diego, California, June 2016), Association for Computational Linguistics, pp. 681–691. URL: https://aclanthology.org/N16-1082, doi:10.18653/v1/N16-1082. 2, 3

[LLL*18] LIU S., LI Z., LI T., SRIKUMAR V., PASCUCCI V., BREMER P.-T.: Nlize: A perturbation-driven visual interrogation tool for analyzing and interpreting natural language inference models. *IEEE transactions on visualization and computer graphics 25*, 1 (2018), 651–660. 2

[LU20] LABRÍN C., URDINEZ F.: Principal component analysis. In *R for Political Data Science*. Chapman and Hall/CRC, 2020, pp. 375–393. 4

[LWY*22] LI Z., WANG X., YANG W., WU J., ZHANG Z., LIU Z., SUN M., ZHANG H., LIU S.: A unified understanding of deep nlp models for text classification. *IEEE Transactions on Visualization and Computer Graphics 28*, 12 (2022), 4980–4994. 2

[MHM18] MCINNES L., HEALY J., MELVILLE J.: Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426* (2018). 4

[PNJ*19] PARK C., NA I., JO Y., SHIN S., YOO J., KWON B. C., ZHAO J., NOH H., LEE Y., CHOO J.: Sanvis: Visual analytics for understanding self-attention networks. In *2019 IEEE Visualization Conference (VIS)* (2019), IEEE, pp. 146–150. 2

[SFSvdW23] SARTI G., FELDHUS N., SICKERT L., VAN DER WAL O.: Inseq: An interpretability toolkit for sequence generation models. *arXiv preprint arXiv:2302.13942* (2023). 2

[SKB*21] SEVASTJANOVA R., KALOULI A.-L., BECK C., SCHÄFER H., EL-ASSADY M., ZONG C., XIA F., LI W., NAVIGLI R.: Explaining contextualization in language models using visual analytics. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)* (2021), Association for Computational Linguistics, pp. 464–476. 2

[SKB*22] SEVASTJANOVA R., KALOULI A., BECK C., HAUPTMANN H., EL-ASSADY M.: Lmfingerprints: Visual explanations of language model embedding spaces through layerwise contextualization scores. In *Computer Graphics Forum* (2022), vol. 41, Wiley Online Library, pp. 295–307. 2

[SSZ*23] SHAO Z., SUN S., ZHAO Y., WANG S., WEI Z., GUI T., TURKAY C., CHEN S.: Visual explanation for open-domain question answering with bert. *IEEE Transactions on Visualization and Computer Graphics* (2023). 2

[SXK*22] SHUSTER K., XU J., KOMEILI M., JU D., SMITH E. M., ROLLER S., UNG M., CHEN M., ARORA K., LANE J., ET AL.: Blenderbot 3: a deployed conversational agent that continually learns to responsibly engage. *arXiv preprint arXiv:2208.03188* (2022). 2

[TDFH*22] THOPPILAN R., DE FREITAS D., HALL J., SHAZEER N., KULSHRESHTHA A., CHENG H.-T., JIN A., BOS T., BAKER L., DU Y., ET AL.: Lamda: Language models for dialog applications. *arXiv preprint arXiv:2201.08239* (2022). 2

[TMS*23] TOUVRON H., MARTIN L., STONE K., ALBERT P., ALMAHAIRI A., BABAEI Y., BASHLYKOV N., BATRA S., BHARGAVA P., BHOSALE S., ET AL.: Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288* (2023). 2

[TWB*20] TENNEY I., WEXLER J., BASTINGS J., BOLUKBASI T., COENEN A., GEHRMANN S., JIANG E., PUSHKARNA M., RADEBAUGH C., REIF E., ET AL.: The language interpretability tool: Extensible, interactive visualizations and analysis for nlp models. *arXiv preprint arXiv:2008.05122* (2020). 2

[VdMH08] VAN DER MAATEN L., HINTON G.: Visualizing data using t-sne. *Journal of machine learning research 9*, 11 (2008). 4

[Vig19a] VIG J.: Bertviz: A tool for visualizing multihead self-attention in the bert model. In *ICLR workshop: Debugging machine learning models* (2019), vol. 23. 2

[Vig19b] VIG J.: A multiscale visualization of attention in the transformer model. *arXiv preprint arXiv:1906.05714* (2019). 2

[web] Interpreting bert model. URL: https://captum.ai/tutorials/Bert_SQUAD_Interpret2. 2, 3

[WHJ*23] WANG X., HUANG R., JIN Z., FANG T., QU H.: Commonsensevis: Visualizing and understanding commonsense reasoning capabilities of natural language models. *arXiv preprint arXiv:2307.12382* (2023). 2

[WHSX20] WU C.-S., HOI S., SOCHER R., XIONG C.: Tod-bert: pre-trained natural language understanding for task-oriented dialogue. *arXiv preprint arXiv:2004.06871* (2020). 2

[WSP*22] WANG L., SHEN Y., PENG S., ZHANG S., XIAO X., LIU H., TANG H., CHEN Y., WU H., WANG H.: A fine-grained interpretability evaluation benchmark for neural nlp. *arXiv preprint arXiv:2205.11097* (2022). 2

[WTC21] WANG Z. J., TURKO R., CHAU D. H.: Dodrio: Exploring transformer models with interactive visualization. *arXiv preprint arXiv:2103.14625* (2021). 2

[YCW*23] YEH C., CHEN Y., WU A., CHEN C., VIÉGAS F., WATTENBERG M.: Attentionviz: A global view of transformer attention. *arXiv preprint arXiv:2305.03210* (2023). 2

[YSHC21] YIN F., SHI Z., HSIEH C.-J., CHANG K.-W.: On the sensitivity and stability of model interpretations in nlp. *arXiv preprint arXiv:2104.08782* (2021). 2

[ZLD*22] ZENG A., LIU X., DU Z., WANG Z., LAI H., DING M., YANG Z., XU Y., ZHENG W., XIA X., ET AL.: Glm-130b: An open bilingual pre-trained model. *arXiv preprint arXiv:2210.02414* (2022). 2

[ZSG*19] ZHANG Y., SUN S., GALLEY M., CHEN Y.-C., BROCKETT C., GAO X., GAO J., LIU J., DOLAN B.: Dialogpt: Large-scale generative pre-training for conversational response generation. *arXiv preprint arXiv:1911.00536* (2019). 2