InterpreT: An Interactive Visualization Tool for Interpreting Transformers

Vasudev Lal[1], **Arden Ma**[1], **Estelle Aflalo**[1], **Phillip Howard**[1],
Ana Paula Q Simoes[1], **Daniel Korat**[2], **Oren Pereg**[2], **Gadi Singer**[1], **Moshe Wasserblat**[2]
[1]Intel Labs, Cognitive Computing Research, USA
[2]Intel Labs, Artificial Intelligence Lab, Israel
{firstname.lastname}@intel.com

Abstract

With the increasingly widespread use of Transformer-based models for NLU/NLP tasks, there is growing interest in understanding the inner workings of these models, why they are so effective at a wide range of tasks, and how they can be further tuned and improved. To contribute towards this goal of enhanced explainability and comprehension, we present InterpreT, an interactive visualization tool for interpreting Transformer-based models. In addition to providing various mechanisms for investigating general model behaviours, novel contributions made in InterpreT include the ability to track and visualize token embeddings through each layer of a Transformer, highlight distances between certain token embeddings through illustrative plots, and identify task-related functions of attention heads by using new metrics. InterpreT is a task agnostic tool, and its functionalities are demonstrated through the analysis of model behaviours for two disparate tasks: Aspect Based Sentiment Analysis (ABSA) and the Winograd Schema Challenge (WSC).

1 Introduction

In recent years, Transformer-based models (Vaswani et al., 2017) such as BERT (Devlin et al., 2019), GPT-2 (Radford et al., 2019), XLNET (Yang et al., 2019) and RoBERTa (Liu et al., 2019) have demonstrated state-of-the-art performance in many NLP tasks and have become the gold standard. However, there are many open questions regarding the behavior of these models. Phenomena such as why Transformers perform well on specific examples but not others, as well as how their internal mechanisms facilitate their ability to generalize to new tasks and settings (or lack therof) are not yet fully understood. Observations and insights which help answer these questions will be pivotal in driving the construction of more powerful and robust models.

The pursuit of such answers have spurred the development of a wide variety of analytical studies and tools to enable the visualization of information encapsulated in Transformer-based models. Clark et al. (2019), studied the attention mechanisms of a pre-trained BERT model to find that certain heads correspond to specific linguistic patterns. Jawahar et al. (2019) investigated the distribution of phrase-level information throughout the layers of BERT using t-SNE (van der Maaten and Hinton, 2008). The visualization tools of Aken et al. (2020) and Reif et al. (2019) perform a layer-wise analysis of BERT's hidden states to understand the internal workings of Transformer-based models that are fine-tuned for question-answering tasks. Other tools, such as Vig (2019), focus on visualizations of the attention matrices of pre-trained Transformer models. In the work of Tenney et al. (2020), the authors introduce an interactive platform for the visualization and interpretation of NLP models. The tool includes, among other capabilities, attention visualizations, embedding space visualizations, and aggregate analysis. Other related tools include those by Wallace et al. (2019) and Hoover et al. (2020). The increasingly large body of work on the interpretability and evaluation of Transformer-based models reveals the growing need for the development of tools and systems to aid in the fine-grained analysis and understanding of these models and their performance on complex language understanding tasks.

With this goal in mind, we present **InterpreT**[1], a tool for **interpre**ting **T**ransformers. A key contribution of InterpreT is that it is a single system that enables users to track hidden representations

[1]The source code for InterpreT, along with a live demo and screencast describing its functionality is available at https://github.com/IntelLabs/nlp-architect/tree/master/solutions/InterpreT

Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations, pages 135–142
April 19 - 23, 2021.

of tokens throughout each layer of a Transformer model, as well as visualize and analyze attention head behaviors. Similarly to Tenney et al. (2020), InterpreT enables dynamic point selection, aggregation of attention head statistics, visualization of attention head matrices, and the ability to compare models. Novel contributions made in InterpreT include the ability to track and visualize token embeddings through each layer of a Transformer (Section 3.2), highlight distances between certain token embeddings through illustrative plots (Section 3.6), and identify task-related functions of attention heads by using new metrics (Section 3.3).

Section 4 demonstrates how the new features introduced in InterpreT can be used to obtain novel insights into the underlying mechanisms used by Transformers to tackle diverse tasks such as Aspect-Based Sentiment Analysis (ABSA) and the Winograd Schema Challenge (WSC). More generally, these demonstrations illustrate how such features enable rich, granular analysis of Transformer models.

2 System Design and Workflow

The system flow consists of two main stages: offline extraction of model specific and task specific information such as targets, predictions, relevant hidden states, and attention matrices (henceforth referred to as "collateral") and running the application itself. During the offline stage, the extracted hidden states are processed using t-SNE before being saved to a file. The collateral generated for a specific model and task is independent of collateral from other models and tasks, which enables the user to either run the app to examine a single model or to compare two different models that were evaluated on the same task and data. In this latter case, the collateral files for the two models are linked at runtime. A detailed specification for the collateral, along with the source code used to run InterpreT can be found in our GitHub.

3 Application Features

3.1 Overview

Key features of InterpreT include plots for the visualization and tracking of t-SNE representations of hidden states through the layers of a Transformer, a plot presenting summary statistics, custom metrics to quantify attention head behavior, and attention matrix visualizations. In addition, InterpreT includes a multi-select feature that enables groups of examples to be selected in the t-SNE plot and used as input to other plots in the application, as well as the flexibility to be used both for analyzing a single model and for visualizing the differences in behaviors between two models. In general, the core functionalities present in InterpreT are model and task agnostic, working for a wide-variety of architectures, sequence lengths, and tasks.

3.2 t-SNE Embeddings

A central component of InterpreT is the ability to visualize the contextualized embeddings of specific tokens throughout the layers of a Transformer. Following van Aken et al. (2019) and Jawahar et al. (2019), we use t-SNE to project hidden representations of tokens after each Transformer layer onto a two-dimensional space, creating disjoint t-SNE spaces for each layer of each model. In the resulting t-SNE plot, token embeddings can be visualized for a specific model and layer, and colored using various color schemes (Figure 1d). An example selected in the t-SNE plot is tracked and continues to be highlighted in the new t-SNE space when the model or the layer is changed.

3.3 Head Summary

InterpreT includes a head summary plot that displays attention head summary statistics for each head and layer (Figure 1b). For a given sentence, all attention weights are obtained in a matrix of size $(num\_layers \times num\_heads \times sentence\_length \times sentence\_length)$ and compute statistics over the final two dimensions, yielding a summary plot of size $(num\_layers \times num\_heads)$. The following statistics are currently supported:

The **Standard Deviation** of an attention head is generated by calculating the standard deviation of the corresponding attention matrix weights. Intuitively, the standard deviation of an attention head increases as the attention patterns become less uniform, allowing a user to easily identify heads that exhibit interesting behaviors.

The **Attention Matrix Correlation** is obtained by computing the correlation between an attention matrix and an arbitrary, same-size matrix. In Section 4.1.2, this correlation is computed using a binary matrix that encodes syntactic dependency relations, analogous to the parse matrix used in

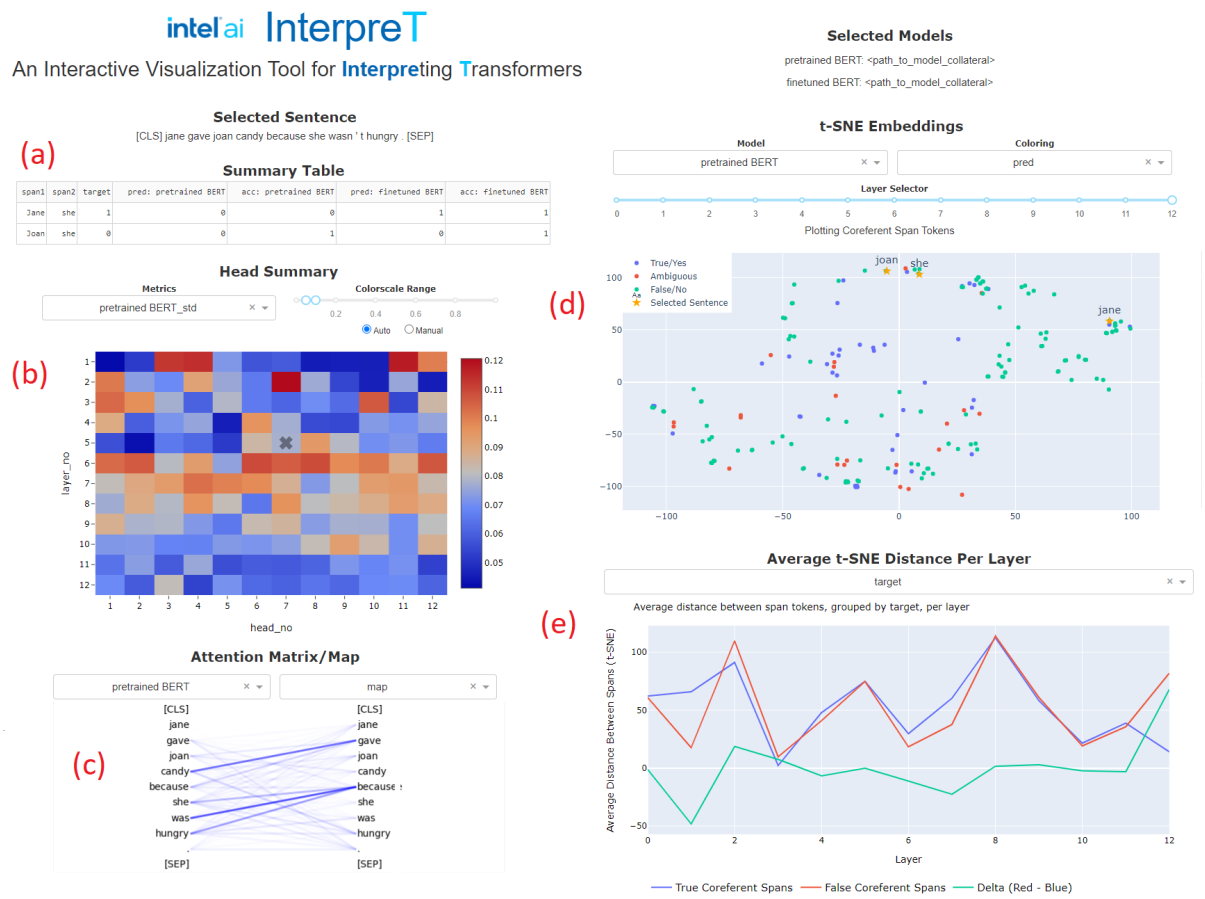


Figure (1) The InterpreT user interface (rearranged for print) for the task of coreference resolution (see Section 4.2). The UI includes a short description of the currently selected models and example at the top, along with the main features (a-e) described in Section 3.

Pereg et al. (2020). This formulation of a "grammar correlation" metric provides an indicator of an attention head's ability to identify syntactic relations in a sentence.

The **Task-Specific Attention Intensity** option allows a user to define and display custom metrics that highlight specific attention patterns. In Section 4.2.2, a "coreference intensity" metric is devised to pinpoint attention heads with an affinity for identifying coreference relationships. For this metric, each entry in the summary plot represents the attention weight between the coreferent spans being evaluated (if the span contains more than one token, the maximum is taken), for each head of each layer.

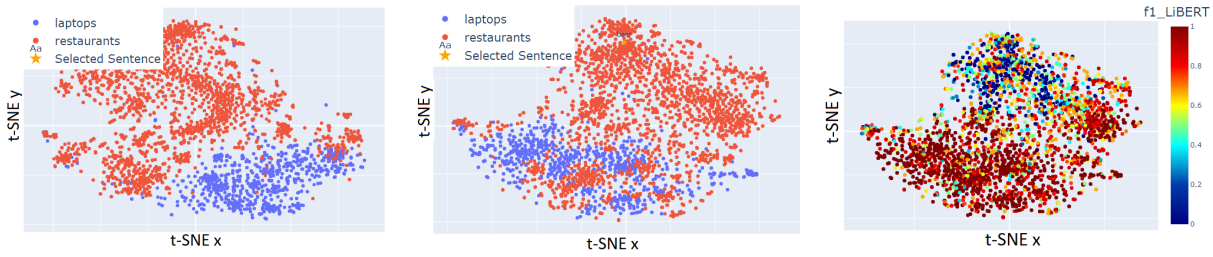
When running InterpreT with two models, the head summary plot can be used to visualize differences in the summary statistics between both models. As mentioned previously, the multi-select feature can be used with any of the summary statistic options. When using multi-select, the statistics are averaged over the selected examples, enabling the user to analyze general trends in attention behavior.

3.4 Attention Matrix/Map

Similarly to other systems, InterpreT provides the ability to display the attention patterns and weights exhibited by specific attention heads, which can be selected by clicking on a specific head and layer in the head summary plot. These attention patterns can be displayed as either a heatmap ("matrix" view) or a token "map" ("map" view) visualization used in Clark et al. (2019). There is an option to switch between the two views in-app (Figure 1c). These visualizations can become unwieldy when using large sequence lengths, but this will not affect the functionality of the rest of the system.

3.5 Summary Table

A short summary table is provided, which contains task-specific information such as predicted token classifications and the gold (target) labels for the selected sentence/example (Figure 1a).



(a) (b) (c)

Figure (2) Baseline (a) and LIBERT (b,c) final layer t-SNE embeddings of aspect terms colored by domain (a,b) and aspect extraction sentence level F1 score (c) as seen in InterpreT.

3.6 Average t-SNE Distance Per Layer

To complement t-SNE visualization of the hidden states, InterpreT also introduces a novel plot showing the average t-SNE space distance between specific groups of terms across all of the Transformers' layers (Figure 1e). Section 4.2.1 demonstrates how information conveyed in this plot contributes towards novel interpretations of the inner workings of BERT.

4 Use Cases

The examples presented in this section focus on the analysis of bidirectional encoders using InterpreT, however the system can be applied to generative models or encoder-decoder architectures as well, so long as the appropriate collateral can be generated. Further examples of use cases along with instructions on how to use InterpreT for custom applications is detailed in our GitHub.

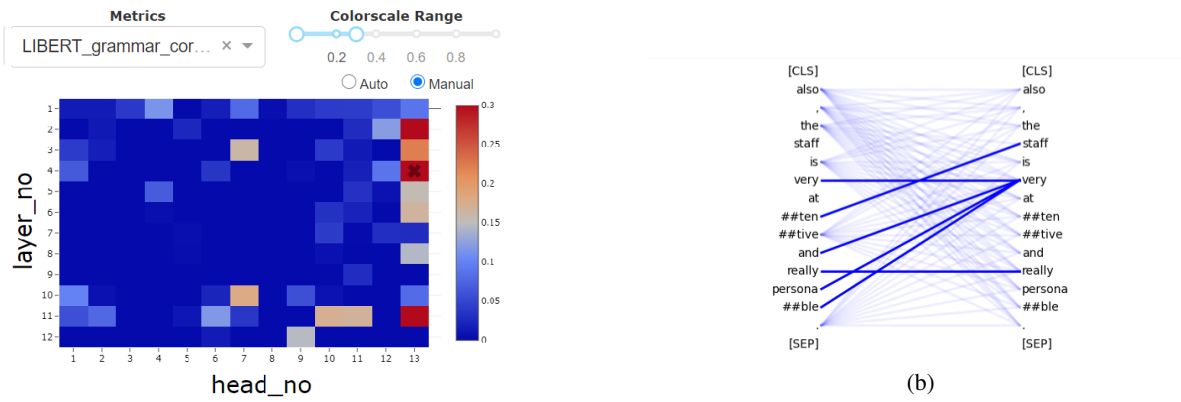
4.1 Cross-Domain Aspect Based Sentiment Analysis (ABSA)

A fundamental task in fine-grained sentiment analysis is the extraction of aspect and opinion terms. For example, in the sentence *"The chocolate cake was incredible"*, the aspect term is *chocolate cake* and the opinion term is *incredible*. Supervised learning approaches have shown promising results in single-domain setups where the training and the testing data are from the same domain. However, these approaches typically do not scale across domains, where only unlabeled data is available for the target domain. It has been shown that syntax, which is a basic trait of language and is therefore domain invariant, can help bridge the gap between domains (Ding et al., 2017; Wang and Jialin Pan, 2018).

In a recent work (Pereg et al., 2020), externally generated dependency relations are integrated into a pre-trained BERT model through the addition of a 13th attention head which incorporates the dependency relations into its Syntactically-Aware Self-Attention Mechanism. This model is referred to as Linguistically Informed BERT (LIBERT). InterpreT is used to analyze LIBERT and a Baseline model that shares the same size and structure as LIBERT but does not incorporate syntactic information for the cross-domain ABSA task, where both models are fine-tuned on laptop reviews and are evaluated on restaurant reviews (Pontiki et al., 2014, 2015; Wang et al., 2016). LIBERT and the Baseline model achieved aspect extraction F1 scores of 0.5143 and 0.4254 respectively on validation data from the restaurant domain.

4.1.1 Visualizing the Domain Gap

InterpreT is used to visualize how the incorporation of dependency relations in LIBERT contributes to bridging the gap between domains. Figure 2 depicts the final layer aspect term t-SNE embeddings from the restaurant and laptop domains produced by LIBERT and Baseline. The plot of the Baseline embeddings (2a) gives a prototypical depiction of the "domain gap" challenge present in cross-domain setups, through the clear separation of in-domain (blue) and out-of-domain (red) aspects. Conversely, the plot of LIBERT's embeddings (2b) demonstrates how LIBERT has learned to push the embeddings of some aspect terms from the out-of-domain region into the in-domain region, effectively overcoming the "domain gap" challenge for these examples. Furthermore, in the plot colored by the aspect extraction F1 score (2c), it is seen that LIBERT achieves a high F1 score on the out-of-domain examples that now overlap with in-domain examples, highlighting the usefulness of such visualizations for analyzing model performance and extensibility.



(a) (b)

Figure (3) InterpreT's Head Summary plot displaying aggregated grammar correlation using multi-selection for LIBERT (a) along with an example of the the attention matrix of selected attention head (head 13 in layer 4) (b).

4.1.2 Grammar Correlation

A key feature of InterpreT is the addition of metrics to help identify attention heads which carry out specific functions. For analyzing LIBERT, the "grammar correlation" metric described in Section 3.3 is used to identify attention heads with an affinity for detecting syntactic relations. Figure 3a demonstrates the result of using multi-selection to compute the average grammar correlation in each of LIBERT's attention heads aggregated over multiple examples.

As expected, the Syntactically-Aware Self Attention head (head 13) tends to show much higher grammar correlation than the regular Self Attention heads. Utilizing the granularity provided in the head summary plot, it is observed that LIBERT's 13th head seems to only express an affinity for parsing syntactic relations in layers 2,3,4, and 11. This is unexpected behavior, as the syntax information is relayed identically to the 13th head across all layers. To investigate further, InterpreT can be used to display attention matrices from head 13 in layers that have high grammar correlation. One such attention matrix, for an out-of-domain example, is displayed in Figure 3b. In this attention matrix visualization, it can be seen how LIBERT's 13th head identifies syntactic relations such as the *adjectival modifier* relation between "staff" and "attentive", and how this can be useful for the cross-domain ABSA task where "staff" and "attentive" are aspect and opinion terms (respectively) in an out-of-domain example.

4.2 Coreference Resolution in the Winograd Schema Challenge (WSC)

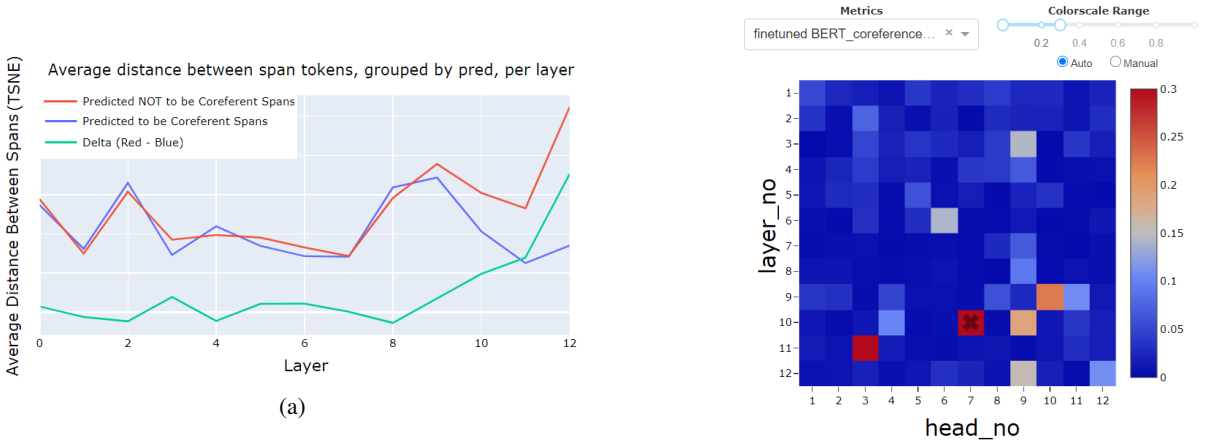
In this section, the utility of InterpreT is showcased for a markedly different task: coreference resolution. Coreference resolution is a challenging NLP task that often requires a nuanced understanding of context and sentence semantics. This task is the basis of the Winograd Schema Challenge (WSC) from the SuperGLUE benchmark (Alex Wang, 2020), where the goal is to determine whether or not a pronoun is the correct referent of a given noun phrase. In this analysis of WSC, InterpreT demonstrates how information in the attention matrices and the hidden states of a Transformer can be used to understand the implicit mechanisms contributing to its ability to identify coreferent terms. BERT-base (uncased) is chosen for this analysis and is fine-tuned using the WSC task training set.

Example	Coreference Candidates (Fred, he)	(George, he)
"... got back"	**False**	**True**
"... got up"	**True**	True

Table (1) Predictions of the fine-tuned BERT model for the two examples. The values in bold are correct predictions.

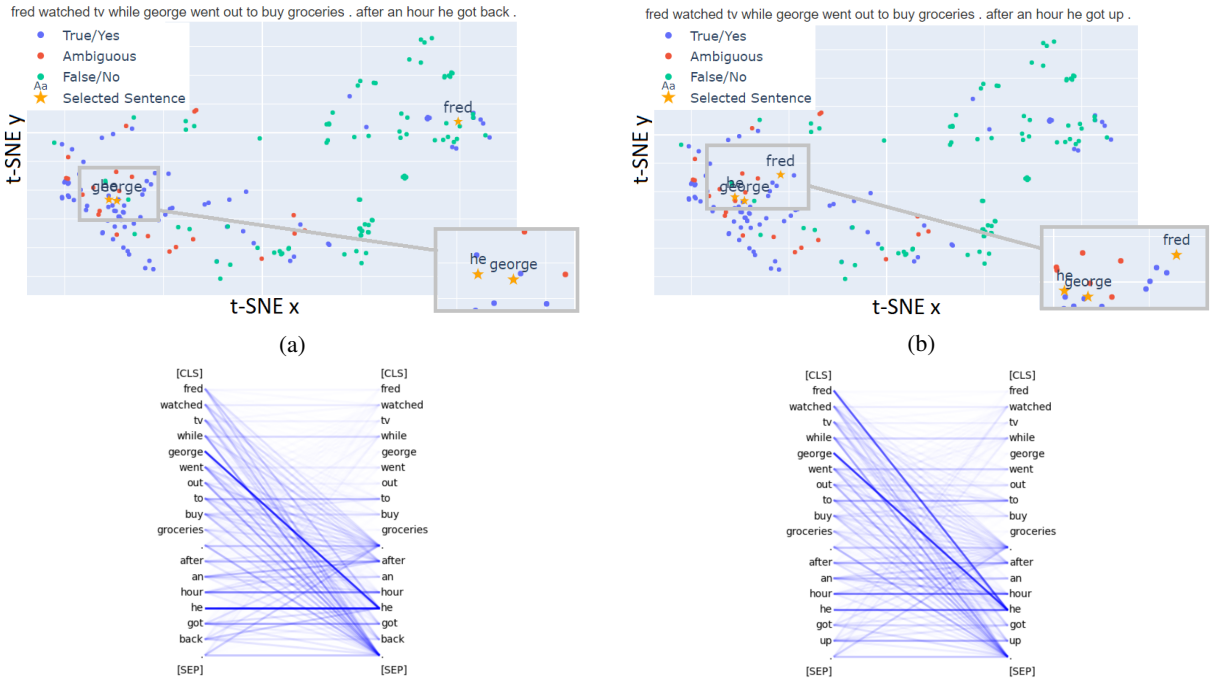
4.2.1 Spatial Convergence of Coreferent Terms

While analyzing WSC with InterpreT, the system's wide-ranging capabilities gave rise to a novel observation, wherein it was discovered that a fine-tuned BERT model pushes closer together the embeddings of terms it predicts to be coreferent. Figure 4a displays the average distance per layer



(a) (b)

Figure (4) InterpreT summary plots for WSC. These plots display summary statistics for the average predicted span token distance per layer (a) and coreference intensity metric (b) for fine-tuned BERT aggregated over the full dataset.



(a) (b)

(c) (d)

Figure (5) InterpreT plots tracking specific examples in WSC. These plots depict the final layer t-SNE embeddings and attention map visualizations of head 10 layer 7 for the following examples: "Fred watched TV while George went out to buy groceries. After an hour he got **back**" (a,c), and "Fred watched TV while George went out to buy groceries. After an hour he got **up**." (b,d). In (a) and (b), the yellow stars indicate candidate mention spans, and "He" and "George" are almost overlapping.

between terms which BERT predicts to be coreferent (blue) and terms which BERT predicts to not be coreferent (red), aggregated over the full WSC dataset. It is observed that in BERT's final layers, the model learns to modify the hidden representations of terms to increase or decrease the distance between them based on whether or not it predicts they are coreferents. This behavior can also be seen in the green trace, which measures the difference in the average distance of terms predicted to be coreferent and those that are not predicted to be coreferent.

Additionally, Figures 5a and 5b show a specific example of this phenomenon with the sentences:

"Fred watched TV while George went out to buy groceries. After an hour he got **back**" (Figure 5a and Table 1) and "Fred watched TV while George went out to buy groceries. After an hour he got **up**." (Figure 5b and Table 1). These two examples show how changing a single token ("**back**" became "**up**") significantly alters the sentence semantics, as in the first example, "he" refers to "George", and in the second example "he" refers to "Fred". InterpreT enables us to visualize this behavior using the t-SNE plots. Figure 5a show how for the first example, "he" and "George" are much closer together than "he" and "Fred" are. Figure 5b shows how in the second example, the change from "he got back" to "he got up" is reflected in BERT's behavior, where the representation of "Fred" to be pushed much closer to "he" than in the first example.

4.2.2 Attention Patterns between Coreferent Terms

Another feature of InterpreT is the ability to utilize custom metrics, such as the "coreference intensity" metric described in Section 3.3. Coreference intensity is visualized using the head summary plot in Figure 4b. The figure shows that the fine-tuned model highlights attention heads that seem to perform well as coreferent predictors. Darker shades of red correspond to higher attention between the two coreferents being evaluated. It appears that the heads which are the most involved in the coreference resolution task after fine-tuning are the 7th head of layer 10 and the 3rd head of layer 11.

This new metric is used to examine the example previously presented with "Fred", "George", and "he". Figures 5c and 5d show the attention matrix visualizations for the head selected in Figure 4b (head 7 in layer 10). The token map visualization depicts how "he" attends heavily to "George" in the first example (5c) while attending to both "Fred" and "George" in the second example (5d).

5 Conclusion and Future Work

InterpreT is a generic system for interpreting Transformers, as evident through its suite of tools for understanding general model behaviors and for enabling granular analysis of attention patterns and hidden states for individual examples. The capabilities provided by InterpreT empower users with new insights into what their models are learning, as illustrated in the visualization of the mitigation of the "domain gap" for ABSA and in the novel discovery of the spatial convergence of coreferent terms in WSC. These examples showcase how the fine-grained analysis enabled by InterpreT affords a higher level of insight that is indispensable for interpreting model behavior for complex language understanding tasks.

InterpreT is an ongoing development effort. Future work will include support for additional use cases as well as additional analysis and interactivity features, such as the ability to dynamically add and modify examples while the app is running.

6 Acknowledgements

We thank the anonymous reviewers for their comments and suggestions.

References

Betty van Aken, Benjamin Winter, Alexander Löser, and Felix A. Gers. 2019. How does bert answer questions? *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*.

Betty van Aken, Benjamin Winter, Alexander Löser, and Felix A. Gers. 2020. Visbert: Hidden-state visualizations for transformers. In *Companion Proceedings of the Web Conference 2020*, WWW '20, page 207–211, New York, NY, USA. Association for Computing Machinery.

Nikita Nangia Amanpreet Singh Julian Michael Felix Hill Omer Levy Samuel R. Bowman Alex Wang, Yada Pruksachatkun. 2020. Superglue: A stickier benchmark for general-purpose language understanding systems.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019. What does bert look at? an analysis of bert's attention. In *BlackBoxNLP@ACL*.

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

Ying Ding, Jianfei Yu, and Jing Jiang. 2017. Recurrent neural networks with auxiliary labels for cross-domain opinion target extraction. In *Association for the Advancement of Artificial Intelligence*, pages 3436—3442.

Benjamin Hoover, Hendrik Strobelt, and Sebastian Gehrmann. 2020. exBERT: A Visual Analysis Tool to Explore Learned Representations in Transformer Models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 187–196, Online. Association for Computational Linguistics.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What does BERT learn about the structure of language? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3651–3657, Florence, Italy. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86):2579–2605.

Oren Pereg, Daniel Korat, and Moshe Wasserblat. 2020. Syntactically aware cross-domain aspect and opinion terms extraction. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1772–1777, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Suresh Manandhar, and Ion Androutsopoulos. 2015. SemEval-2015 task 12: Aspect based sentiment analysis. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 486–495, Denver, Colorado. Association for Computational Linguistics.

Maria Pontiki, Dimitris Galanis, John Pavlopoulos, Harris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. 2014. SemEval-2014 task 4: Aspect based sentiment analysis. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 27–35, Dublin, Ireland. Association for Computational Linguistics.

A. Radford, Jeffrey Wu, R. Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Emily Reif, Ann Yuan, Martin Wattenberg, Fernanda B Viegas, Andy Coenen, Adam Pearce, and Been Kim. 2019. Visualizing and measuring the geometry of bert. In *Advances in Neural Information Processing Systems*, volume 32, pages 8594–8603. Curran Associates, Inc.

Ian Tenney, James Wexler, Jasmijn Bastings, Tolga Bolukbasi, Andy Coenen, Sebastian Gehrmann, Ellen Jiang, Mahima Pushkarna, Carey Radebaugh, Emily Reif, and Ann Yuan. 2020. The language interpretability tool: Extensible, interactive visualizations and analysis for nlp models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need.

Jesse Vig. 2019. Visualizing attention in transformer-based language representation models.

Eric Wallace, Jens Tuyls, Junlin Wang, Sanjay Subramanian, Matt Gardner, and Sameer Singh. 2019. AllenNLP interpret: A framework for explaining predictions of NLP models. 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): System Demonstrations*, pages 7–12, Hong Kong, China. Association for Computational Linguistics.

Wenya Wang and Sinno Jialin Pan. 2018. Recursive neural structural correspondence network for cross-domain aspect and opinion co-extraction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pages 1—-11.

Wenya Wang, Sinno Jialin Pan, Daniel Dahlmeier, and Xiaokui Xiao. 2016. Recursive neural conditional random fields for aspect-based sentiment analysis. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 616–626, Austin, Texas. Association for Computational Linguistics.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. Cite arxiv:1906.08237Comment: Pretrained models and code are available at https://github.com/zihangdai/xlnet.