Functional Components as a Paradigm for Neural Model Explainability

James Fiacco

CMU-LTI-23-017

Language Technologies Institute
School of Computer Science
Carnegie Mellon University
5000 Forbes Ave., Pittsburgh, PA 15213
www.lti.cs.cmu.edu

Thesis Committee:

Dr. Carolyn Rosé (chair), Carnegie Mellon University
Dr. Emma Strubell, Carnegie Mellon University
Dr. Rayid Ghani, Carnegie Mellon University
Dr. Yonatan Belinkov, Technion, (Israel Institute of Technology)

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy
In Language and Information Technologies

Abstract

Despite their ubiquity, trained neural models remain a challenging subject for explainability, with neural net researchers applying what might be considered esoteric and arcane knowledge and skills to understand what the models are learning and how the internal workings of the models change their learning outcomes. Understanding what these models are learning is a field of utmost importance as more and more production systems rely on neural models to provide more and more high-impact utilities.

This work lays out an interpretability methodology, built on a design philosophy for neural models that redefines the unit of analysis for these models from individual neurons to a set of interconnected functional components which we call neural pathways. These functional components, which are a consequence of the architecture, data, and training scheme, have the capacity to cut across structural boundaries. This enables a method of functionally-grounded, human-in-the-loop model understanding through increased transparency, encouraging a dialogue between the models and the researchers.

Over the course of this work for this thesis, we contribute to the literature in four ways: First, we provide the method for neural model interpretability at the subtask level, rigorously validating it against a suite of synthetic datasets. Second, we extend the method by providing a framework for aligning learned functional components to causal structures. This enables the comparison of the learned functions of a neural model with a theoretical causal structure allowing for rapid validation of our understanding of how a neural model is approaching a task. Third, we expand the method to compare and align functional components across models with differing architectures or training procedures. And lastly, we demonstrate the capabilities of the neural pathways approach in several domains of education technologies. This includes automatic essay feedback via rhetorical structure analysis, group formation via transactivity detection, and automated essay scoring.

This last contribution can be further specified into three facets separated by their domains and foci. First, neural pathways are employed to scaffold a neural discourse parser to more easily generalize to student writing. Next, we demonstrate that neural pathways can be used as a method for error analysis by exploring the discrepancy in performance between models trained on detecting transactivity in different domains. And lastly, we demonstrate the capability of tracking changes in problematic pathways across fine-tuning an AI writing detector.

With the broad applicability of the neural pathways approach, we are optimistic that the method can have a wide impact on the the design and development of neural models and we aim to provide a foundational work that has the capability of being extended far beyond the scope of the thesis.

Dedicated to my parents for all the questions you asked me,
and Gwen for telling me when my answers did not make sense.

Contents

List of Figures

List of Tables

1 Introduction

In recent years, the study of neural networks has witnessed a remarkable surge, revolutionizing benchmark modeling tasks across various fields. Yet, as we dive deeper into these advancements, a pivotal challenge emerges: the models' complexity often renders them inscrutable, even to seasoned researchers. The analogy of "fortune tellers reading tea leaves" captures the mystique of our attempts to decode these models. This obscurity is not merely academic; as our reliance on neural models in real-world applications grows, understanding their inner workings becomes crucial.

Presently, the intricate techniques employed to interpret neural networks primarily cater to those deeply versed in the field's mathematics, and while progress is being made across the interpretability field, this exclusivity not only limits accessibility but also poses a potential bottleneck to the technology's broader application. This thesis seeks to add to the body of literature new tools to illuminate the learned functions of neural models. By contributing towards closing the knowledge gap, we aspire not just to enhance researchers' ability to construct better models, but also to enable better choices regarding which model to begin with from the outset.

This work contributes to the body of interpretability of neural models in a way that redefines the unit of a neural model from a set of interconnected structural components, such as weights and architectures, to the set of interconnected functional components which we call, as a reference to the biological inspiration for this approach, neural pathways. These functional components, which are a consequence of the architecture, data, and training scheme, have the capacity to cut across structural boundaries as shown by the body of published work covered by this thesis. This enables a method of human-in-the-loop model selection through increased transparency. While the scope of this thesis is firmly targeted at increasing the capabilities of machine learning researchers, the broader aim is to offer the capability for a wide array of domain experts to be able to select from among models to achieve a desirable trade-off between task performance and acceptability of decision making processes (e.g., in terms of avoiding harmful biases). This neural pathways approach is a method to probe into complex neural models to recover functional groups of neurons and align them with reasonable task knowledge heuristics.

We explore this problem through the domain of education technologies including automatic essay feedback via rhetorical structure analysis, group formation via transactivity detection, and the automatic detection of machine generated writing. The field of education technologies provides a strong test bed for this work as it is a domain with many more domain experts as compared to machine learning researchers. Furthermore, being able to identify learned substructures within models may be able to provide a finer grained approach to identifying areas for feedback that were not explicitly trained for. The task of transactivity detection has shown a high affinity for transfer learning from generic NLP inference tasks to small datasets. Identifying transferred learned components may provide insight for identifying methods to identify good task candidates for transfer learning.

The challenge in explaining models erodes confidence in users, particularly in fields where accurate results are critical, such as medicine and education (Chitti et al., 2020; Fan et al., 2021), who must turn to models with lower performance but higher

transparency, like expert systems. In other cases, a lack of understanding for how the models works results in a misguided trust in the systems (Li and Suh, 2021) that we, as machine learning researchers, have the responsibility to ensure are safe.

It is of the author's opinion that building trust in neural networks *requires* a deeper insight into how they operate. While it important to note that this thesis can not and is not promising an instant or comprehensive solution to all trust issues surrounding neural models. Rather, our aim is to offer a fresh perspective that helps in better understanding these models, which can enable, through continued effort and future research, a path toward solving these grand challenges of interpretable AI.

1.1 Research Questions

Furthering this work, we consolidate the previous model analysis lessons of this work into a suite of tools to provide greater transparency for neural models that can allow for a human-in-the-loop selection process based not only on model accuracy, but also qualitative aspects of a model's functional pathways. This requires a new type of evaluation method as compared to other model selection techniques as prior automatic model selection has focused primarily on accuracy or other related performance metrics rather than the pathway aware selection we propose. Through this project we will be examining the research questions: Can we isolate connected modules within a neural network that can be transferred to other model architectures? Does this method provide a way to maintain desirable internal learning outcomes through the model interpretation process when compared to other interpretability methods?

To address the first research question and as an anchor point for future research, an evaluation of the feasibility of identifying sub-components of trained neural models and inserting those components into models for tasks for which the sub-component may be useful will need to be done. This would take the form of extracting components from models and selectively transferring components to tasks. Evaluating the performance of the selectively transferred components as compared to the whole embedding and other methods for selective knowledge embedding would give a strong indication of its feasibility.

To address the second question, we apply the the neural pathways approach to several domains and datasets to demonstrate how the method can advance the neural model design process. This takes the form of using pathways as auxiliary features, using pathways for error analysis, and using pathways for model comparison.

Answering these questions would open up an alternative approach to neural network design that enables researchers the flexibility to define what knowledge is to be used by the model to increase trust in the system through transparency of the functional components within. Furthermore, it may, in the long term, lead to an added benefit of enabling non-machine learning domain experts to join the conversation of neural model design, challenging the view that there is a best neural architecture and fostering the view that there should be a focus on how the architecture interacts with the data.

1.2 Thesis Overview

This thesis divides the work into two parts, each with three chapters. The first part introduces the core contribution of this work, neural pathways, and the theoretical foundation for their method of analysis and limitations. The second part applies the neural pathways approach to real world problems to demonstrate that pathway aware model selection and construction can provide benefits for expanding the state-of-the-art in numerous domains. After the two principal parts, we conclude a summary of contributions to the work and a discussion of the broad potential future work that can be enabled beyond the scope of the thesis.

1.2.1 Neural Pathways and Causality

Neural Pathways as Functional Components: Building on the concept of neural network probing tasks (Conneau et al., 2018), we present how abstracting away from the neurons to neural pathways allows for a competitive correlational analysis with a reduced quantity of dimensions. We compare these methods on two common natural language task: Recognizing Textual Entailment (RTE) (Dagan and Glickman, 2004) and Named-Entity Recognition (NER) (Sang and Meulder, 2003). In this chapter we present the mechanism used to extract neural pathways from a neural model.

Neural Pathway Alignment with Bayesian Networks: Neural probes and, by extension, the original formulation of neural pathways is a strictly correlational method for analysis. However, as our goal is to know how a neural model reaches a decision, we present a method for aligning pathways with causal structures. In order for this to function, we alter the neural pathways approach from a factor analysis method to a latent variable method. This allows us stricter theoretical guarantees when interfacing from the realm or correlation to that of causation. Furthermore we introduce a metric for alignment between a causal structure and the extracted neural pathways.

For this section we devise synthetic data whose causal structure can completely be known for our experiments. This allows us complete control and the ability to verify the method and discover its limitations, a key component for all of the following work that builds on this foundation.

Complete Neural Pathways Approach with Model Comparison: With the interest to make this method as accessible as possible and to further validate the process, we provide a detailed walk-through of the method, addressing each of the key decision points on the way to building a complete interpretation of a model. We rigorously define the boundaries within with the method can operate in and what questions can be answered.

Like the previous chapter, this section requires the use of synthetic data in order for all aspects of the causal structure to be available for exhaustive experimentation.

1.2.2 Applications of Neural Pathways

Utilizing High Salience Neural Pathways to Improve Generalizability of RST Parsing on Student Writing: For rhetorical analysis driven automatic essay feedback, we examine models for predicting rhetorical moves and steps as described by Swalesian

genre theory and for parsing trees defined by rhetorical structure theory. The model for predicting rhetorical moves and steps is trained on the Research Writing Tutor corpus and reaches near human performance on the task. The model is then dissected using several common neural interpretation approaches as a multifaceted approach to explore the inner workings of the model.

The efficacy of utilizing the inner workings of a model to develop a stronger model is explored in a neural rhetorical structure parser that, using knowledge of its inner working to influence added components, saw increased performance on the RST parsing task on both the standard dataset for RST parsing and a dataset composed entirely of student writing.

Neural Pathways in Transfer Learning: For transactivity detection we look into how attention based models handle transferable learning between generic language technologies tasks and specific domain specific datasets. By choosing pretraining tasks with sufficiently similar concepts required, we found we could apply deep learning approaches to datasets with sizes that would be generally unreasonable to use neural networks for and get significant improvement over simpler models that would ordinarily be used on smaller datasets.

Identifying Propagation of Problematic Pathways in Fine-Tuned AI-Writing Detectors:

Finally, we delve into the intricacies of foundation models, particularly in relation to AI writing detection, spotlighting the evolution of potential biases in a RoBERTa-based AI-writing detection model during fine-tuning. A significant stride made is the introduction of a novel AI-writing detection dataset tailored for student compositions, enriching the current pool of such datasets and aiding in pinpointing bias origins. Our detailed exploration of the fine-tuning dynamics within LLM's neuron groups has unearthed mechanisms and patterns crucial for understanding bias amplification during model training. Additionally, our findings underscore the inherent risks in relying too heavily on a limited set of pretrained foundation models for diverse tasks, highlighting the persistence of key functional components and their associated biases.

1.2.3 Conclusion and Future Directions

In the concluding chapter of this dissertation, we look ahead to potential developments in the field of neural network interpretability. It is the authors hope that this work serves as a starting point, prompting further research and refinement. The methods presented here could be expanded to encompass a wider range of models and datasets. Moreover, the core ideas have the potential to influence and assist in the advancements in other areas of machine learning. This next chapter outlines such future directions, linking our current findings to the opportunities that lie ahead.

2 Motivation and Background

The primary body of literature that we draw from is neural model interpretation. This is supplemented by a number of other domains where we apply the neural pathways approach including discourse parsing, textual entailment, transactivity, automatic essay feedback, which are described in Chapter 6. Our work on interpretation expands on this prior work while simplifying some procedures and introducing some of our own. Our goal is to address all three questions with an integrative approach, making use of multiple lenses and then integrating the disparate pictures each provides into a unified vision of network function.

In this chapter we examine the literature in neural model interpretation and causal modelling in neural networks that were foundational to this work and provide context and justification for decisions that are have been made. It is important to note that the field of interpretability has changed and evolved concurrently to the work performed for this thesis, and the ways with which these more recent works integrate with the neural pathways approach are discussed in Section 11.1. Each of the following sections of this chapter will cover what has been done in the field, where the gaps are that require further work, and how my work addresses a subset of that work.

2.1 Taxonomy of Neural Network Interpretability

This section delineates a taxonomy of the prevailing methodologies and techniques in neural network interpretability, drawing from seminal literature in the field. With the rapid expansion of interest and effort in the field of neural network explainability, it is impossible to enumerate or touch on every approach or paradigm that is used, but in an effort to situate this work within the broader research discourse, we present the top-level divisions that we have found to be present in the literature as well as some notable examples from each of those classifications.

Two salient demarcations in the interpretability literature are the distinction between *intrinsic* and *post-hoc* interpretability methods and the distinction between *high* and *low* abstraction interpretability methods. Intrinsic interpretability methods, as discussed by Räuker et al. (2023), focus on elucidating the internal mechanisms and representations by creating deep neural networks (DNNs) that are inherently interpretable. Conversely, post-hoc interpretability methods, as described by Madsen et al. (2022), furnish explanations subsequent to a model's training phase, operating predominantly in a model-agnostic manner. Moving into the dimension of abstraction, we observe that both intrinsic and post-hoc methods can employ high or low abstraction in their explanations. High-abstraction methods provide explanations using more abstract, human-comprehensible concepts, often articulated in sentences. These methods are tailored for easy human comprehension and often treat the neural network as a black box. On the other hand, low-abstraction methods encapsulate the model's specific behavior given an input, focusing on sub-networks or individual neurons.

Examples of intrinsic interpretability methods include continual learning methods (De-Arteaga et al., 2019; Smith et al., 2023; Ahn et al., 2019; Aljundi et al., 2019; Kirkpatrick et al., 2017; Li and Hoiem, 2017; Titsias et al., 2019; Zenke et al., 2017; Lee et al., 2019; Rusu et al., 2016; Yoon et al., 2018) where models are trained to

prevent catastrophic forgetting by maintaining salient weights; sparse networks (Moran et al., 2021; Wong et al., 2021; Lage and Doshi-Velez, 2017; Meister et al., 2021; Wang et al., 2020; Yeom et al., 2021) where models are trained to have sparsely activating neurons that are more likely to represent a human understandable concept; modular networks (Amer and Maul, 2019; Agarwala et al., 2021; Mittal et al., 2022) where networks are designed such that concepts will be learned by specific sub-networks; self-explaining models (Akata et al., 2018; Hendricks et al., 2016, 2018; Kim et al., 2018b; Patro et al., 2020; Camburu et al., 2018; Kumar and Talukdar, 2020; Lamm et al., 2021; Zhao and Vydiswaran, 2021) where models are trained to produce explanations in natural language; adversarial methods (Engstrom et al., 1906; Salman et al., 2020; Casper et al., 2022a,c; Santurkar et al., 2019) where models that are trained adversarial have been shown to yield more interpretable features in some cases; and disentanglement methods (Whitney, 2016; Siddharth et al., 2017; Esmaeili et al., 2018; Chen et al., 2020; Koh et al., 2020; Losch et al., 2019, 2021; Subramanian et al., 2018) where model representations are trained to align with interpretable concepts. Conversely, post-hoc interpretability methods pertain to methodologies that furnish explanations subsequent to a model's training phase. As these techniques operate with a fixed pre-trained model, they are predominantly model-agnostic and are designed to be retroactively applicable, offering an analysis of the model's decision-making processes (Madsen et al., 2022). In the remainder of this section we subdivide the space of post-hoc interpretability methods further.

Further granularity in the interpretability discourse can be achieved by categorizing the explanations into high-abstraction or low-abstraction paradigms (Madsen et al., 2022). High-abstraction methods, provide explanations using abstract concepts, often articulated in the form of sentences. These explanations are inherently human-grounded, meaning they are tailored for easy human comprehension. These methods often treat the neural network as a black-box. While intrinsic model can also have a high or low levels of abstraction, we focus on the spectrum for post-hoc interpretability methods as they are more related to our work. Post-hoc models that are also considered high abstraction include challenge sets and stress tests (Lehmann et al., 1996; Naik et al., 2018a) where alternative evaluation data that has been constructed specifically to verify the capabilities of the model to perform a task in a specific way is used, and other dataset-based methods (Zhou et al., 2014; Bau et al., 2020; Mu and Andreas, 2020; Hernandez et al., 2021; Oikarinen and Weng, 2022). Conversely, low-abstraction methods, which comprise sub-network methods for interpretability and individual neuron level interpretability encapsulate the model's specific behavior given an input.

Early of these attempts to understand the internal, low-abstraction functions of trained neural models limited themselves to investigations of the function of individual neurons or individual architectural components. An early way to probe the function of target components, as Karpathy et al. (2015) and Strobelt et al. (2016) have each proposed, is by visualizing patterns of activation through the target components, for example using heatmaps. There have also been approaches that made use of simpler classifiers to predict and then explain mistakes made by more complex models (Ribeiro et al., 2016; Krishnan and Wu, 2017). In a similar vein, linear classifier probes (Shi et al., 2016a; Adi et al., 2016; Conneau et al., 2018; Zhu et al., 2018; Kuncoro et al., 2018; Khandelwal et al., 2018; Gupta et al., 2015; Köhn, 2015; Lepori and McCoy,

2020; Li et al., 2022; Lindström et al., 2021; Miaschi et al., 2021; Niven and Kao, 2019; Perone et al., 2018; Saleh et al., 2020; Tamkin et al., 2020) have been used to co-train simple linear models to illustrate functions performed by particular layers in arbitrarily deep models, and then later by associating the learned patterns in the linear models with task or linguistic knowledge determined by hand or through some other means to be relevant or not instance-by-instance.

Additionally to these individual neuron focused methods, there has also been work in more broadly aligning representations in neural networks with understandable concepts. Concept vectors (Fong and Vedaldi, 2018; Kim et al., 2018a; Lucieri et al., 2020a,b; Reif et al., 2019; Zhou et al., 2018a; Abid et al., 2022; Yuksekgonul et al., 2022) are one such family of methods where the latent space of a neural network is treated as a landscape of meaningful concepts. Another family of methods use gradients to attempt to synthesize features that are represented by the latent space of the neural network (Mahendran and Vedaldi, 2015; Nguyen et al., 2016b; Olah et al., 2017; Casper et al., 2022a,c; Nguyen et al., 2016a, 2017). A closely related family of methods projects gradients onto inputs or neurons to identify which neurons or inputs contribute the most to a specific output, this family is aptly known as gradient-based methods (Adebayo et al., 2018, 2020; Ancona et al., 2019; Denain and Steinhardt, 2022; Dombrowski et al., 2019; Fokkema et al., 2022; Jeyakumar et al., 2020; Nielsen et al., 2022; Slack et al., 2020; Zhang et al., 2019; Ancona et al., 2018; Durrani et al., 2020; Lundstrom et al., 2022).

There are also two post-hoc methods that are roughly analogous to two intrinsic methods: sparse networks and module networks. Like sparse-network methods, weight-masking methods (Csordás et al., 2020; Wortsman et al., 2020; Zhao et al., 2020) leverage the concept that neural networks are often highly over-parameterized and only have a limited amount of important weights and neurons. However, because the models are not necessarily designed to be sparse, these methods can encounter issues with pruned models (Blalock et al., 2020; Frankle and Carbin, 2018; Vadera and Ameen, 2022). Analogous to module networks, partitioning methods (Watanabe et al., 2018, 2019; Liu and Arik, 2020; Casper et al., 2022b; Lange et al., 2022; Watanabe, 2019) attempt to cluster groups of related neurons together, but without any run-time analysis. Similar to partitioning methods, though leveraging run-time analysis to provide a more functional analysis of the activations of the models, there are neural circuit methods (Santurkar et al., 2021; Townsend et al., 2020; Zhang et al., 2018, 2020b). It is of note that Räuker et al. (2023) considered the original formulation of the neural pathways approach from Fiacco et al. (2019a) a member of the neural circuit family of methods. While this is a reasonable cataloging of the original approach, in Section 2.4, we explain how work in this thesis has drawn from various families of neural interpretation techniques to fit a new complementary niche.

2.2 Identifying Gaps in the Literature

Upon beginning the work in this thesis, we identified three primary gaps in the existing neural network interpretation literature that have substantial room for development:

- Historical approaches focus on understanding the roles of individual neurons in the greater neural network. It is our view that studying the interpretability of a

neural network at the individual neuron level can too easily lose the forest for the trees (see Section 2.4.3). A single neuron can simultaneously be too small of a unit to see a pattern and an entity that can affect too many different connected aspects of the network to fully see the breadth of its influence.

- Prior approaches focus on a single neural network at a time. In our view, an important utility of neural network interpretation is for choosing the correct model for a given task. By focusing interpretation on a single network at a time, it can be more difficult to see similarities between superficially different models (e.g. several different model architectures) or differences in superficially similar models (e.g. an identical model trained on different data).

- Current approaches do not take into account that any single interpretation technique will not tell one everything they want to know about a model. Each interpretation technique is a lens through which a piece of the whole picture can be seen. It is our view that there should be more focus on designing interpretation techniques that expect to be used concurrently with other complementary approaches.

Recent surveys of neural interpretation techniques have identified similar gaps (Sajjad et al., 2022) with the addition of three more limitations that are relevant to this thesis:

- Many approaches rely on human-defined concepts to analyze models, which can result in incorrect or incomplete analysis.

- The causal relation between neurons and a model's prediction is not fully understood.

- There is a lack of standard evaluation benchmarks, making it difficult to compare studies.

Throughout this thesis we primarily focus on addressing the first of our identified gaps; we propose a more abstract method for model interpretation that looks at functional group of neurons as the unit of analysis as opposed to individual neurons. While recently there has been considerable effort put into the field of neural network interpretability, such that these gaps are beginning to be filled, our method settles into a niche that connects an open questions in neural interpretation. *How to align sub-networks of a neural network to understandable feature in a manner that is functionally grounded?* We then explore this approach through a combination experiments on both simulated and real world data. Through correlation with the functional groups of neurons, we can furthermore reason about the role that those linguistic and task-level features have in the network's predictions.

Despite the primary focus being on the abstraction of neural interpretation, we are still mindful of the other two of our identified gaps to allow for a natural growth in those directions for future work. Specifically we demonstrate a method for using our interpretation approach to abstract even further from the core functional neuron groups to inter-model groups of related neuron groups. This allows if the identification of similar structures in different models even if the architectures are different.

We also generally keep in mind the idea that this interpretation approach is only one of many approaches that may be used to study the knowledge learned by a model. With this in mind we carve out its niche as a high level interpretation approach that can provide a good general understanding of what a model has learned and uses for its task as quickly as possible. Methods of interpretability should not be considered to supersede or to compete with other each other for supremacy of the field in the same way that some models can be objectively "better" than other at solving a task. Rather, we should see each interpretation method as one of many tools and instruments that each can answer a specific question or set of questions about a model. Only by using multiple approaches is it reasonable to claim understanding of a model.

2.3 Motivating Questions in Model Interpretation

In Fiacco et al. (2019b), we identified a need for three motivating questions that are central to the successful interpretation of a neural model:

- Which upstream inputs influence a neuron's activation level, and conversely, how do its activation levels affect downstream task performance?
- To what degree is activation related to input at a single time step as opposed to being context-sensitive (i.e., demonstrating influence from multiple time steps)? This is challenging to determine within recurrent neural models as there is a large amount of interdependence between neurons both within and across timesteps.
- Which neurons carry the most statistical influence over the final classification? A principal challenge of neural network interpretation is that there are too many neurons to attempt understanding what each contributes to the big picture. Narrowing to those that are most important would make interpretation more tractable.

2.4 Specifying the Niche along Axes of Interpretability Methods

This section situate the work in this thesis among related works along five axes: neural network explanation types, post-hoc versus intrinsic methods, level of abstraction, causality, and the measurement of interpretability.

2.4.1 Explanations in Neural Interpretation

The deceptively simple question *What is an explanation?* is still largely an open question in neural network interpretation, but necessary for the systematic evaluation of interpretation techniques (Wiegreffe and Marasović, 2021). We use the definition from Wiegreffe and Marasović (2021) that defines an explanation as a process of providing justification for or an understanding of the decision-making process of a neural network model. An explanation can be in the form of *highlights*, which are subsets of the input elements that explain a prediction; *free-text explanations*, which are free-form textual justifications that are not constrained to the input instance, or *structured explanations*, which are explanations that are not entirely free-form but are still written in natural language with constraints placed on the explanation-writing process.

For purposes of this thesis, we constrain out explanations to those defined as *highlights*, though we do so indirectly. Our method makes heavy use of factor analysis (Tipping and Bishop, 1999) which can be understood as carving out a subset of neurons that are deemed important for the predictions. The activations of the neurons are directly a result of inputs to the model and thus the method is, in effect, searching for highlights within the model that explain the predictions.

While our method is most reasonably classified as using highlights as explanation, a challenge arises in selecting datasets from prior work in this form of explanation. Specifically, many datasets that focus on highlights, require aligning decisions of a model to specific text in an input (Kwiatkowski et al., 2019; Chalkidis et al., 2021; Khashabi et al., 2018; Socher et al., 2013; Pang and Lee, 2005; Carton et al., 2020).

Furthermore, Madsen et al. (2022) posits a distinction between *local explanations* and *global explanations* as it applies to neural network interpretability methods. The former focuses on explaining why a model made a specific decision for a specific observation; the latter focuses on explaining the model as a whole. In this work, we take a global perspective towards our explanations, that is, we aim to explain the functions that the model has learned and applies to all inputs. The combination of the results of those functions manifests as the behavior of the network.

2.4.2 Post-hoc Interpretability Methods

As described in Section 2.1, the literature categorizes interpretability methods into intrinsic and post-hoc approaches. Intrinsic methods focus on models that are naturally interpretable, sometimes termed "white-box" models, though their claims of interpretability often require validation, as exemplified by debates over the attention mechanisms which were originally thought to be intrinsically interpretable (Bahdanau et al., 2014), but were later found out to be rather inconsistent (Jain and Wallace, 2019; Wiegreffe and Pinter, 2019; Serrano and Smith, 2019; Vashishth et al., 2019). On the other hand, post-hoc methods emphasize explaining models after training. While intrinsic methods are more specific in their application, post-hoc methods offer broader adaptability, contingent upon their accuracy. Both methods have unique contributions to model accountability, and their intersection can provide mutual validation.

Intrinsic methods are built into the model design and can thus be constrained by their inherent architecture. Post-hoc methods, on the other hand, allow for a more flexible analysis of any given model after its training. This flexibility facilitates broader applicability across various model types and domains, and was the driving reason for this work to focus on this type of method. There are potentials for complementary work on both intrinsic methods and post-hoc methods along with their interactions, but to limit the scope of this work, we focus only on the latter.

2.4.3 Abstraction

We also described in Section 2.1 an axis of distinction around interpretability models based on their level of abstraction. Low-abstraction methods, such as input features explanations or neurons, zoom into the specifics, often spotlighting individual input tokens or neural activations to indicate their contribution to a given prediction. While this

granularity affords detailed insights, it can remain limited in its ability to convey broader concepts that are human understandable (Madsen et al., 2022). On the other hand, high-abstraction techniques, exemplified by natural language explanations, operate at a more elevated level, providing overarching narratives that can encompass more abstract concepts in their elucidation. While the high-abstraction methods are typically more human-friendly and accessible, they might not always capture the nuanced behavior of a model as faithfully as their low-abstraction counterparts. Thus, choosing the right level of abstraction in interpretability is a delicate balance and is contingent upon the context and the intended recipients of the explanation.

The method presented in this thesis is primarily a low-abstraction method. We focus on identifying specific patterns of activation in the neural network to attribute meaning to. However, we take inspiration from more abstract methods to overcome some shortcomings of purely low-abstraction methods, especially those found in neuron-specific methods (Räuker et al., 2023). Specifically, the problems of polysemantic neurons, that are neurons which play multiple roles within the network, and frivolous neurons, that are neurons that do not play any meaningful role within the neural network, are directly addressed by our method as will be explained in detail throughout the remainder of the thesis.

2.4.4 Causal Modeling in Neural Networks

There has been increased interest in the literature for causal representations of neural models. Bengio et al. (2019) present a method for using a transfer objective to determine if there is a relatively sparse portion of the model that is resistant to change from the pretraining task to the post training task. Such sparsity could indicate learned causal relationships within the model. Elazar et al. (2021) propose an alternative method to traditional neural probes, called Amnesic Probing, which focuses on the influence of a causal intervention that removes information from the representation in order to assess the utility of a property for a given task. Furthermore, thee found that conventional probing performance is not correlated with task importance, leading them to call for increased scrutiny of claims that draw behavioral or causal conclusions from probing results.

There have also been an increase in perturbation and ablation based models (Hod et al., 2021; Zhou et al., 2018b; Morcos et al., 2018; Ravfogel et al., 2022) where parts of a model are modified to observe the effects. These counterfactual experiments allow researchers to observe causal effects within a neural network. On such example is in the analysis in Bengio et al. (2019) where they use changes of neurons in the network to determine if these causal relationships exist.

In this work, while strictly correlational in nature, we align neural models with arbitrary Bayesian networks to answer the question of *to what extent does the processes of the neural network resemble the causal processes defined by the Bayesian model*. As such, while it is not a causal method, it does make strides bridging the theoretical divide between correlational interpretability techniques and causal interpretability techniques by allowing researchers to ask questions about how well does their understanding of how a model works, reflect that actual behavior of the model.

2.4.5 Measurement of Interpretability

The measurement of interpretability in machine learning is a nuanced domain with little consensus on how best to measure it (Madsen et al., 2022). However, methods for evaluating such methods can be broadly categorized into three types (Doshi-Velez and Kim, 2017; Madsen et al., 2022) human-grounded, application-grounded, and functionally-grounded interpretability. Human-grounded interpretability is concerned with the resonance of explanations with human understanding, focusing on their immediate comprehensibility rather than their real-world applicability. On the other hand, application-grounded interpretability anchors its assessments in real-world scenarios, probing whether model explanations lead to tangible benefits in specific contexts, often contrasting machine-generated insights against those offered by humans. Though in machine learning research, this type of measurement is not often done in NLP research because it is highly application specific and incurs a significant cost (Madsen et al., 2022). Lastly, functionally-grounded interpretability, rather than gravitating towards human comprehension or real-world utility, centers on the fidelity of explanations to the models they represent, ensuring that the insights provided truthfully mirror the underlying model behaviors. Collectively, these categories furnish a multifaceted framework to rigorously assess the clarity, utility, and accuracy of model explanations in varied contexts.

While the resonance of explanations with human understanding and their real-world applicability are undeniably significant, a critical foundational aspect is to ensure that these explanations authentically encapsulate the intrinsic workings of the model. Absent this functional grounding, even the most lucid or pragmatically beneficial insights are susceptible to misconceptions or outright inaccuracies. Consider the propensity to anthropomorphise the behavior (Li and Suh, 2021) of machine learning models or selectivity bias in human raters (Miller, 2019). Functionally-grounded interpretability acts as a bulwark against these potential inaccuracies, serving as a cornerstone for the reliability of model explanations. For the purposes of this work, it is this form of grounding we use to ensure that the methods presented are not just readily comprehensible or opportune but are fundamentally reflective of the model's processes. Given the escalating reliance on machine learning models in both academic and practical spheres, maintaining the veracity of these explanations is imperative.

3 Neural Pathways as Functional Components

Interpretation of neural models is a difficult task because the knowledge learned within neural networks is distributed across hundreds of thousands of parameters. Interpreting the significance of any individual neuron is tantamount to reconstructing a forest based on a single pine needle. More specifically, the contribution of each individual neuron is a minuscule part in the overall representation of the learned solution, and the mapping between neurons and function may be many-to-many (Goodfellow et al., 2016). In this chapter we present the core contribution to this thesis which is the method of network interpretation that enables a more abstract view of what a network has learned which we refer to as neural pathways (Fiacco et al., 2019a). In this approach, inspired by the concept of biological neural pathways used in neuroscience research to understand physical brain function (Kennedy et al., 1975), a network is factored into functional groups of co-firing neurons that cut across layers in a complex network architecture. Rather than attempt interpretation of the activation pattern through a single neuron at a time, we instead attempt interpretation of a functional group of neurons where the activation pattern of the group, we argue, may be more effectively associated with human level task and linguistic knowledge. This enables understanding the neuron groups as working together to accomplish a comprehensible sub-task. These pathways help conceptualize what task and linguistic knowledge a model may be using in an approximate way, the benefit of which is that it does not depend on an isomorphism between network architectures. This chapter focuses on presenting the neural pathways technique while later chapters will expand on the theory and limitations of the method.

This method, which can be applied simply in a purely post-hoc analysis, independent of the training process, can enable both understanding of individual models and comparison across models. The interpretation process enables investigation of which identified functional groups correspond to linguistic or task level heuristics that may be employed in well understood non-neural methods for performing the task. Furthermore, it enables comparison across very different architectures in terms of the extent and the manner in which each architecture has approximated use of such knowledge. In so doing, the method can also be used to formulate explanations for differences in performance between models based on relevant linguistic or task knowledge that is identified as learned or not learned by the models. This approach builds on and extends prior work using linguistic and task knowledge to understand the behavior and the results of modern neural models (Shi et al., 2016b; Adi et al., 2016; Conneau et al., 2018).

In the remainder of this chapter we provide a detailed explanation of the neural pathways approach and apply it to previously published neural models, namely models for the task of named entity recognition (NER) (Ma and Hovy, 2016) on CoNLL 2003 data for English (Sang and Meulder, 2003) and recognizing textual entailment (Dagan and Glickman, 2004). We compare across different neural architectures through a shared lens comprising linguistic and task-level heuristics for the two target tasks and draw conclusions about learning outcomes on those tasks.

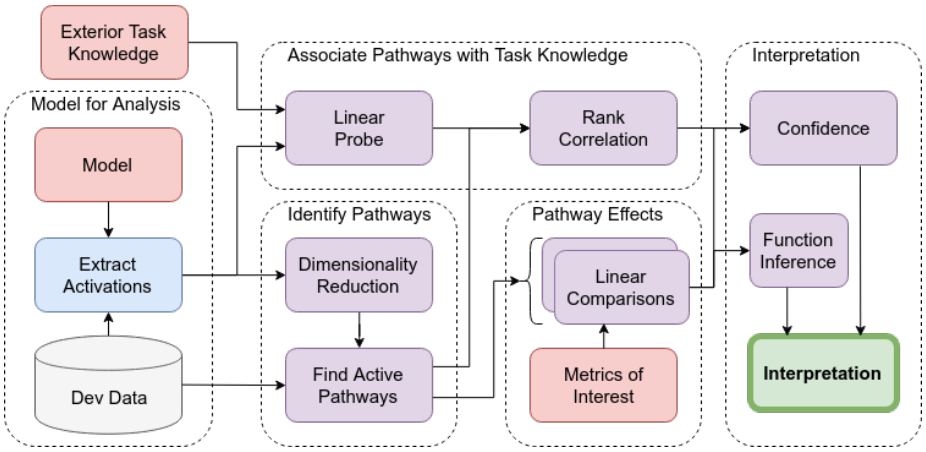


Figure 1: Flowchart representation of neural pathway based model interpretation.

3.1 Neural Pathways

Many previous approaches have analyzed individual neurons or architectures of specific neural networks with gradient methods (Karpathy et al., 2015; Bach et al., 2015; Arras et al., 2017). However, we propose an approach that enables abstraction above the surface structure of a network architecture, enabling a relaxation of the assumption of an direct link between structure and function. To accomplish this abstraction, we employ a simple approach to identify what we conceptualize as emergent neural pathways, which are specific sets of co-firing neurons that work together as the model makes predictions on the data. To understand the specifics of the function performed by the functional group, we align activation patterns through the group per instance with patterns of relevance for task and linguistic knowledge.

3.1.1 Prerequisites

As this is an interpretation method, there is an assumed set of information about the model, the dataset, and the task that must be known in order to apply the techniques effectively. Namely, there should be a reference set of heuristic knowledge, either at the linguistic or task level, that is associated with the dataset on an instance-by-instance level for at least some subset of the data.

Metrics of Interest: As our approach can be generalized across many tasks, the metrics that will be used to identify the salient pathways must be defined before the interpretation process. Section 3.2.1 and 3.2.2 provide specific examples of these metrics as applied to the entailment and NER models. Metrics are chosen to be able to be easily computed and will provide the target values for the statistical analysis outlined in Section 3.1.3, Linear Comparisons. Example metrics include disagreement between models, incorrectly predicted values, or other task specific metrics.

Model and Data: The neural pathways method is a post-training analytic approach, and thus it requires the existence of pretrained models, that will be the target of the

interpretation process. This stands in contrast to previous co-training approaches, where the mechanism for interpretation is trained simultaneously with the networks that are of interest.

Task Knowledge: This interpretation method is built on the assumption that the researcher has external knowledge of the task that their model is being applied to. This can be as straightforward as simply having a feature engineered baseline, as with our named entity recognition example (Section 3.2.2). However, it can also be as nuanced as having access to an analysis of the types of required knowledge to accurately predict certain instances in the data, as in our recognizing textual entailment example where we use an alternate validation set for the MultiNLI corpus where subsets have been earmarked as of interest for specific kinds of task and linguistic knowledge (Section 3.2.1). The external knowledge that is brought to the interpretation process will directly affect what conclusions can be drawn from the neural model as this method does not generate new knowledge, but validates the relevance of external knowledge for explaining network function. If the knowledge brought to the process is only partial, then only partial understanding of network function will be possible. However, as one iterates through the interpretation process, the potential relevance of additional knowledge may emerge, and the process can be repeated with the expanded set. This is an advantage of not requiring the interpretation mechanism to be trained along side the model in question.

Extracting Activations: As a preparatory step for the interpretation process, an activation matrix is constructed where the columns represent individual neurons, the rows represent instances, and the value of each cell is the activation of the associated neuron in the associated instance. Part of this method's flexibility is that the set of probed neurons can be arbitrarily large or small. This way, the sets can be specified to analyze the pathways within certain subsections of the model or in the model as a whole. This flexibility allows researchers to ignore parts of the model that may already be well explained by other neural interpretation techniques (e.g. low-level feature extraction in convolutional neural networks in image recognition, or attention heatmaps).

3.1.2 Identifying Pathways

Neural pathways are a distinct (though related) phenomenon from interconnectivity of a given network based on individual connection weights. While the weights describe the strength of connectivity between individual pairs of neurons, co-activation is an emergent property that arises through sets of connected neurons, and because of this, pathways can not be constructed through a simple graph partitioning of the network structure based on weights apart from the observation of the network in use.

For our analysis, we selected the number of pathways for each model so that they explain $\approx 75\%$ of the total variance in the model. This number was chosen arbitrarily as a balance between the total variance explained by the dimensionality reduction and the quantity of pathways required. Further experimentation may reveal an optimal balance.

Dimensionality Reduction: A dimensionality reduction is applied to the activation matrix to get a set of factors that will correspond to our neural pathways. While in principle, any form of dimensionality reduction can be used, Principal Component Analysis (PCA) (Hotelling, 1933) is used in this work for the dimensionality reduction

for its simplicity and transparency. Different methods for dimensionality reduction may prove better or worse for interpreting certain models for certain tasks, but the question of which specific dimensionality reduction technique works best is beyond the scope of this work.

Finding Active Pathways: For each data instance in the validation set, the pathways that are activated to produce the model predictions are identified. This is done by constructing an activation matrix, as explained above (Section 3.1.1), and applying PCA to it in order to define functional groups of neurons based on their coordinated behavior. The factors identified become the neural pathways and the factor loadings (DeCoster, 1998) become a means for understanding the activity of the pathways. These factor loadings are later used along with the weights learned by linear probes to align the extracted pathways with interpretable task information.

3.1.3 Evaluating Pathway Effects

With an approach similar to Radford et al. (2017), where it was found in a specific case that sentiment-related activations were encoded within single neurons, we abstract the concept of single neuron prediction up a level to examine single pathway prediction. Rather than operating at the level of a single neuron, where neurons typically play a minuscule part in many different functions, we operate at the level of a pathway, where a pathway represents neurons that demonstrate their relatedness through their coordinated behavior.

Linear Comparisons: This refers to the correlation between the activities associated with each pathway per instance to the pattern of relevance per instance of each metric of interest (e.g. each piece of linguistic or task knowledge). This yields a set of correlation coefficients which represent the importance of each PCA dimension (pathway) for explaining the use of each of the metrics of interest by the learned network.

3.1.4 Associating Task Knowledge with Pathways

Neural pathways are a way to abstract the problem of interpreting single neurons in a neural model to interpreting the functional groups of neurons. In isolation, the pathways are not meaningful, though grounded to task-related information via linear probes and rank correlation, the learned representations within the neural model can be evaluated.

Linear Probes: Like Conneau et al. (2018), a series of logistic regression models are trained to map a neural representation to a given linguistic phenomenon, though all of the neurons from parts of the network that are to be analyzed are included whether or not they come from the same layer. Logistic regression probes were used as opposed to the MLP probes in Conneau et al. (2018) to avoid the problem of attempting to interpret a model with another model that is comparably difficult to interpret. Additionally, concepts beyond surface features may also be used as the targets for the probes. This is demonstrated in Section 3.2.1, where we explore the types of knowledge required to solve a task rather than the surface features of the input. From each of the linear models, we store the weight vector, which represents the importance of each neuron for predicting the types of task-specific phenomena learned by the linear model and the

performance of the linear model which indicates the degree to which that information is embedded in the neural model.

Rank Correlation: Using both the factor loadings of the neurons from Section 3.1.2 and the weights from the linear probes discussed above, we can connect the pathways to known task information. Intuitively, if a neural pathway was approximating a function similar to one of the phenomena examined by the linear probes, then the loadings of each neuron in the pathway would be similar in relative shape to the weights of the relevant linear probe. That is, if the pathway and the probe are viewing the same phenomenon, the neurons with stronger weights in the probe should have higher loadings in the pathway and vice versa. To measure the relatedness of each pathway's loadings to each linear model's weights, we use Spearman's rank correlation coefficient (ρ) (Spearman, 1904), which assesses the monotonicity of two data sets giving a numerical comparison of the relative shapes of the weights and loadings.

Simplification: The original pathways approach (Fiacco et al., 2019a) used the above rank correlation between the weights of the knowledge probes and the loadings on the PCA to determine how closely aligned a type of external knowledge was to a pathway. However, this is approach does not make use of the result of the dimensionality reduction. Given the output of the PCA represents how strongly each factor is realized in a given data instance, this could be improved.

We use these values as the surrogate activations of the *pathways* rather than using the activations of the individual neurons for all of the linear probes. The weights of the probes then directly relate the pathways to to external knowledge. This further reinforces the idea of abstracting away from the individual neuron level to the pathway level while reducing the amount of effort aligning the pathways with external knowledge. We replace the validation step with an additional linear probe, a binary classification predicting the metric given the pathway activations. This also enables us to compare models with the meta-pathways approach detailed later in this chapter.

The rank correlation may still provide insight on how well the structure of that pathways matches a classifier that is directly trained on a given phenomena.

3.1.5 Interpretation

The above methods provide the foundation for a quantitatively backed interpretation of a neural model. With this foundation, inferences can be made about the model with a statistical indicator of the confidence or utility of the pathways.

Function Inference: From pathways that have high rank correlation with the linear probes, it can be inferred that the model contains a set of neurons in those pathways that perform the tasks provided to the probe. It is also known what metrics of interest that pathway has influence over from the linear comparisons. It is then possible to extrapolate whether the model has learned to use the knowledge examined by the probes in such a way that it can influence those metrics. This directly provides an insight into what knowledge the model has learned and in what cases it has learned to apply it.

Confidence: The confidence of the claim that the model has learned such information can be assessed by using the rank correlation coefficient and the performance metrics of the linear probe and the linear comparisons. The rank correlation coefficient measures

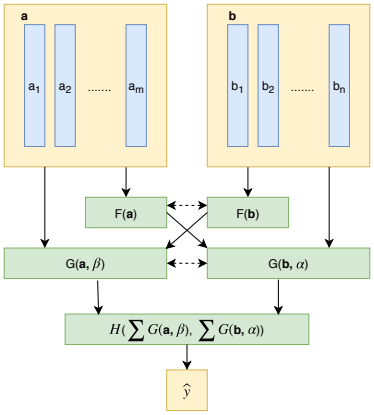


Figure 2: Decomposable Attention Model. Dotted arrows indicate networks with shared weights.

how well the knowledge stored within the network aligns with the function that the pathway is performing. The linear probe and linear comparison performance are likewise related to how likely the information is stored within the pathway and how influential that pathway is on the metric respectively.

3.2 Interpreting Neural Models for NLP Tasks

To evaluate our interpretation technique on real world data, we applied our method on four trained models over two tasks: recognizing textual entailment using the Multi-genre Natural Language Inference corpus (Williams et al., 2018) and named entity recognition using the CoNLL 2003 data (Sang and Meulder, 2003) for English NER. The analysis was implemented using Scikit-Learn (Pedregosa et al., 2011) and SciPy (Jones et al., 2001–) and unless otherwise noted used default hyperparameters.

3.2.1 Recognizing Textual Entailment

Recognizing textual entailment is a task comprised of deciding whether the concepts presented in one text can be determined to be true given some context or premise in a different text (Dagan and Glickman, 2004). The Multi-genre Natural Language Inference (MultiNLI) corpus (Williams et al., 2018) follows this definition and contains annotated pairs of sentences which are labeled as *entailment* if the hypothesis sentence is definitely true given the premise sentence, *contradiction* if the hypothesis is definitely false given the premise, and *neutral* if the hypothesis could be true, but is not guaranteed to be given the premise.

Models and Data: We implemented two neural models for this task: a bidirectional version of the simple LSTM classifier from Bowman et al. (2015) and the decomposable attention model (DAM) (Figure 2) from Parikh et al. (2016a). We use Keras (Chollet et al., 2015) with the TensorFlow (Abadi et al., 2015) backend for our implementations of both of the entailment models.

Metrics of Interest: For purposes of this chapter, the metric of interest used is simply the class value for each data instance. For this task, the activations in the representations for each text segment learned by the model just prior to the classification step are used in the analysis.

Task Knowledge: Our external knowledge for this task comes from a stress test dataset developed for models trained on the MultiNLI corpus (Naik et al., 2018b). There are nine categories and subcategories, each of which contains data instances that require a specific type or reasoning to correctly identify the entailment relationship. We combine all of the data instances in the stress test and tag each with the category or subcategory it belongs to. The entailment models' representations are analyzed in terms of the type of reasoning they can perform. While we acknowledge that recent work by Liu et al. (2019a) has found limitations in this dataset with respect to the reasoning that is required for the models to achieve, we use it as a foundation for interpretation that can be expanded as new resources become available.

Identifying Pathways: For the entailment models, the total variance explained for the decomposable attention model was 76.9% over 15 pathways and for the BiLSTM encoder model variance explained was 76.5% over 175 pathways. This result clearly shows that the representation learned by the decomposable attention model has significantly more internal coherence as compared to the BiLSTM encoder.

Evaluating Pathway Effects: From the linear comparisons for the decomposable attention model, three pathways had a correlation coefficient greater than 0.25 ($p < 0.001$). However, in the LSTM model, there were 14 pathways that correlated with the model prediction, but none of them individually had a correlation coefficient greater than 0.2 ($p < 0.05$). Higher coefficient indicate the pathways that have stronger effect on the model prediction. It also indicates that individual pathways in the decomposable attention model are more informative for understanding why the model makes certain predictions than the LSTM model.

Associating Pathways With Task Knowledge: The results from the linear probes are presented in Table 2 with the F1 score of each probe on the given piece of external task information. For the entailment task, 55% of the instance types can be predicted with high precision and recall for the decomposable attention model, though only 44% with the BiLSTM encoder. There are two stand-out instance types that have major differences between models: Antonyms and Swapped Content Words. Both of these are related to word meanings indicating that the decomposable attention model may be storing more information about meaning than the BiLSTM encoder.

Presented in Table 3 are the results for correlating the neural pathways with the information extracted via the linear probes. The pathway numbers are ordered by variance explained, with lower pathway indexes indicating that the pathway explains more variance in the activations. For the entailment task, the largest difference between the models is that the decomposable attention model has pathways which are correlated well with antonyms and numeric types of data instances even where the antonym pathway represents a relatively small amount of the model variance. Contrasted to this, the BiLSTM encoder model has the best correlations with data instances that display large length differences between the hypothesis and premise sentences. Despite having well over 100 different pathways to explain the variance in the model, the pathways that correlate well with high level instance types also explain more variance on average.

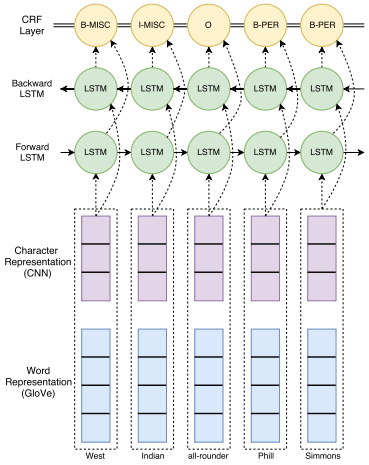


Figure 3: End-to-end model architecture for neural SOTA described in Ma and Hovy (2016). The character representation is computed by a CNN over the characters of the word. This is concatenated with the word embedding (initialized with GloVe) and fed into a BiLSTM. A CRF layer does a sequential decoding to predict the NER tags using the BiLSTM hidden layer vector.

Interpretation of Models: For the entailment models, the experiment was designed to explore the predictive behavior of each model for the task. The linear probes indicate that the information about what type of reasoning is required for a task, which is hypothesized to be encoded in the models, was distinctly encoded in each model, but to a greater extent in the decomposable attention model. The connection between the pathways and the linear probes was less strong, however. This indicates that despite the models having an encoding of the knowledge observed by the probe, it is likely a byproduct of a different function that is being approximated by the neural network. The pathways were created by analyzing which neurons behave cohesively, indicating a subprocess within the network. However, these subprocesses do not correspond strongly to any of the tested features. Consequences of this finding could be an indication that the model is 'cheating' on the task and has some inductive bias that is beneficial to the task independent from the task as envisioned by the creators. Otherwise, if many models demonstrate this behavior, the task or dataset may be insufficient to induce the desired learning behavior in neural models. This is consistent with recent highly domain specific analyses of this task (Gururangan et al., 2018; Glockner et al., 2018; Poliak et al., 2018).

3.2.2 Named Entity Recognition

Given an input sequence, the NER task involves predicting a tag for each token in the sequence that denotes whether the token is an entity or not, as well as what type of entity it is. An example of such a tag might be `PER` for a "person" entity or `ORG` for an "organization" entity.

Models and Data: We implemented two neural models for our experiments: the first (Figure 3) is a well performing neural model that uses a CNN over characters, word embeddings, a Bidirectional LSTM, and a CRF layer for decoding (Ma and Hovy, 2016). Our second model has the same architecture as above only with a BiLSTM over the characters instead of a CNN. The neurons chosen for analysis were the resulting activations for each character encoding sub-network, the word embeddings, and the resulting activations from the sentence level BiLSTM. Implementations of each of the NER models was done using DyNet (Neubig et al., 2017a).

We used the CoNLL 2003 dataset (Sang and Meulder, 2003) for training. For the analysis we sampled the data to get a dataset with a balanced number of classes. The sampling procedure is inexpensive and can be repeated to maintain statistical power.

Metrics of Interest: The differences in predictions for the task are used as the metric of interest. This is a binary value for each data instance where it is 1 if the two models did not produce the same response and 0 otherwise (correct or not). Neurons from across layers were used for the NER task analysis.

Task Knowledge: For our external knowledge, we use a set of features inspired by Tkachenko and Simanovsky (2012) who describe a comprehensive set of traditionally used and linguistically informed features for the NER task. These can be sorted into three categories: *'Local Knowledge Features'* that refer to the features that can be extracted from a particular word; *'External Knowledge Features'* are those that use external information such as part-of-speech tags (extracted using *nltk*[1]); and *Other* which includes miscellaneous features like End-of-Sentence markers, hyphenated words, among others.

Identifying Pathways: For the NER models, 74.5% of the variance was explained for the CNN-BiLSTM-CRF with 40 pathways and 75.1% of the variance was explained by 35 pathways in the BiLSTM-BiLSTM-CRF. This shows a that both models have similar amounts of observable structure within them.

Evaluating Pathway Effects: Similarly, for the NER task, the differences in predictions for the CNN based character encoder model and the BiLSTM based character encoder via the linear comparisons, were explained by several pathways. For the CNN-BiLSTM-CRF, the top 5 predictive pathways for the differences between the two models' predictions have an average of 0.025 higher correlation coefficient ($p < 0.001$) than the BiLSTM-BiLSTM-CRF.

Associating Pathways With Task Knowledge: For the NER task linear probes, 13 out of 50 features are almost perfectly predicted by the activation probes (i.e. greater than 0.90 F1) and there are no significant differences between higher performing probes for the BiLSTM-CRF with the CNN character encoder versus the BiLSTM character encoder. The main difference seen in the results is that the CNN trades off storing information about plural nouns and adjectives for storing clearer representations for parentheses and digits.

For the NER analysis, the pathways that correspond with the surface features represent a very small amount of the variance within the model (with few exceptions). A notable difference between the two models is that the BiLSTM character encoder seems to have a considerably more organized pathway corresponding to title case than

[1] http://www.nltk.org/api/nltk.tag.html

Task	Model	Dev F1
ENTAILMENT	BILSTM ENCODER	57.4
	DECOMPOSABLE ATTENTION	72.8
NER	BILSTM-BILSTM-CRF	83.7
	CNN-BILSTM-CRF	94.4

Table 1: F1 score for each model on the development set for the entailment task and the NER task.

the CNN based character encoder.

Interpretation of Models: The NER model analysis was set up to understand the factors contributing to the differences between the two models rather than the factors influencing the prediction accuracy. Many of the surface features that were tested were present in the models, although there were not significant differences as to which of these features were encoded in one model or the other. Examination of the correlation of each pathway to the prediction differences between the models indicate that the differences were primarily explained by pathways that had high amounts of explained variance. Strong linear probe results, in conjunction with a mismatch between which pathways correlated to the metric of interest and which pathways correlated well to each surface feature that was probed, indicate that each of the models learned the surface features from the data and that other functions are responsible for differences. This can guide future examination of these models to pinpoint exactly what knowledge the model is using for the task. For example, a high variance pathway for the CNN-BiLSTM-CRF included some neurons from the CNN and some from the LSTMs and was typically activated by words with capital letters. However, it also activated on notable exceptions such as "van" and "de" that serve as a lowercase part of some names indicated that it had memorized those exceptions to the broader heuristic. No such pathway was identified in the BiLSTM-BiLSTM-CRF model.

3.2.3 Results

Table 1 shows the F1 score on the validation set for the models on both tasks. These models were not tuned to obtain the highest performance possible as they are simply the subject of the interpretation techniques, but their relative performance on the tasks provides some context for further analysis.

Instance Type	DAM	BiLSTM	Difference
ANTONYM	0.93	0.38	0.55
LENGTH.DIFFERENCE	0.98	0.98	0.00
NEGATION	1.00	0.93	0.07
NUMERIC	0.99	0.96	0.03
WORD.OVERLAP	1.00	0.94	0.06
CONTENT.WORD.SWAP	0.69	0.47	0.22
FUNCTION.WORD.SWAP	0.56	0.47	0.09
KEYBOARD.SWAP	0.59	0.50	0.09
SPELLING.SWAP	0.62	0.59	0.03

Feature	CNN	BiLSTM	Difference
WORD.CONTAINSCAPITAL	0.98	0.98	0.01
WORD.HYPEN	0.80	0.83	-0.03
WORD.ISDIGIT	1.00	0.99	0.01
WORD.ISTITLE	1.00	1.00	0.00
WORD.UPPER	0.92	0.93	-0.01
WORD.LOWER	0.73	0.71	0.01
WORD.POSTAG-(	0.94	0.95	-0.00
WORD.POSTAG-)	0.58	0.38	0.20
WORD.POSTAG-,	1.00	1.00	0.00
WORD.POSTAG-.	0.59	0.59	-0.00
WORD.POSTAG-IN	1.00	1.00	0.00
WORD.POSTAG-JJR	1.00	1.00	0.00
WORD.POSTAG-JJS	0.55	0.66	-0.11
WORD.POSTAG-MD	0.90	0.98	-0.08
WORD.POSTAG-NN	0.95	0.95	-0.00
WORD.POSTAG-NNP	0.95	0.95	-0.00
WORD.POSTAG-NNPS	0.11	0.21	-0.10
WORD.POSTAG-NNS	0.24	0.41	-0.17
WORD.POSTAG-PRP	0.44	0.62	-0.18
WORD.POSTAG-VB	0.17	0.21	-0.04
WORD.POSTAG-VBD	0.99	0.98	0.01
WORD.POSTAG-VBG	0.13	0.19	-0.06
WORD.POSTAG-VBN	0.98	0.98	-0.00
WORD.POSTAG-VBP	0.64	0.59	0.05
WORD.POSTAG-VBZ	0.56	0.64	-0.08

Table 2: Linear probe F1 score for the presence of provided external task knowledge given the neural activations and the difference between the two models. Top: entailment stress test data instance categories. Bottom: NER surface features. All performance metrics have $p < 0.05$.

Instance Type	DAM		BiLSTM	
	Pathway	ρ	**Pathway**	ρ
ANTONYM	12	0.19	16	0.10
LENGTH.DIFFERENCE	0	0.10	17	0.23
NEGATION	1	0.08	1	0.18
NUMERIC	2	0.29	4	0.13
WORD.OVERLAP	3	0.15	10	0.16
CONTENT.WORD.SWAP	8	0.08	32	0.11
FUNCTION.WORD.SWAP	8	0.11	31	0.11
KEYBOARD.SWAP	4	0.09	31	0.13
SPELLING.SWAP	8	0.10	12	0.09

Feature	CNN		BiLSTM	
	Pathway	ρ	**Pathway**	ρ
WORD.CONTAINSCAPITAL	35	0.11	30	0.11
WORD.HYPEN	38	0.09	26	0.07
WORD.ISDIGIT	18	0.11	6	0.16
WORD.ISTITLE	30	0.14	28	0.23
WORD.UPPER	38	0.12	0	0.14
WORD.LOWER	15	0.05	28	0.05
WORD.POSTAG-(	4	0.12	10	0.07
WORD.POSTAG-)	27	0.09	0	0.08
WORD.POSTAG-,	31	0.15	32	0.18
WORD.POSTAG-.	28	0.09	23	0.06
WORD.POSTAG-IN	27	0.13	22	0.15
WORD.POSTAG-JJR	13	0.11	34	0.18
WORD.POSTAG-JJS	0	0.11	8	0.07
WORD.POSTAG-MD	37	0.11	16	0.08
WORD.POSTAG-NN	0	0.07	22	0.06
WORD.POSTAG-NNP	35	0.10	3	0.09
WORD.POSTAG-NNPS	39	0.13	33	0.08
WORD.POSTAG-NNS	26	0.04	8	0.07
WORD.POSTAG-PRP	18	0.06	8	0.14
WORD.POSTAG-VB	0	0.10	25	0.07
WORD.POSTAG-VBD	25	0.08	34	0.13
WORD.POSTAG-VBG	39	0.06	14	0.04
WORD.POSTAG-VBN	38	0.07	17	0.12
WORD.POSTAG-VBP	17	0.05	24	0.10

Table 3: Most correlated neural pathway along with the rank correlation coefficient for each model for each task studied. Top: entailment stress test data instance categories. Bottom: NER surface features. All rank correlations have $p < 0.001$.

3.3 On Connectivity within Pathways

Based on the method to compute neural pathways from model activations, more specifically the dimensionality reduction step, it is reasonable to inquire about the possibility that neurons that end up with in a neural pathway may not connect directly to other neurons within the pathway. In our current formalism for the neural pathways approach we use the term pathway in a specific sense as described above that is no longer directly tied to the biological inspiration of chains of networks of biological neurons. However, we must entertain the possibility that there may be disconnected neurons within our pathways and address the implications should that occur. There are two distinct ways that neurons within a pathway can be disconnected from the others: the can be neurons within the same layer or they can be in different layers. Each of these cases has different considerations that will be discussed in the remainder of this section.

The first case where the neurons are on the same layer is the simpler case and is largely represents the purpose of the neural pathways approach. In general, all neurons within a artificial neural network layer are not directly connect to each other to enforce the assumptions that allow backpropagation and are thus independent from each other given the input to that layer from an information flow perspective. However, we may reasonably expect that, in aggregate, the neurons within the layer represent some intermediate computation for the neural network and the core assumption of the neural pathways approach is that sets of neurons within these intermediate representations contain representations of sub-tasks that the model has learned. It is thus expected and intentional that neurons within a layer are allowed to be within a pathway despite a lack of a direct connection for information to flow between them.

The other case can be broken further into two sub-cases: first, where the neurons are from entirely distinct layers (e.g. two different layers within a feed-forward neural network), and second, where the neurons are from the same layer applied to different information (e.g. from different time-steps in a recurrent neural network). Each of these cases are more complex than the previous and will require future work to characterize the full extent of the implications. It is for this reason that the experiments on each the NER and Entailment tasks earlier in this chapter chose specific sets of neurons to examine that avoided these cases.

We expect the first sub-case to be fairly rare, though we will need to more rigorously define how rare that is. This is because between layers, there are often many if not exhaustive connections between neurons. This means that a pair of neurons in adjacent layers cannot be disconnected and for them to be disconnected across more layers, they both need to be disconnected and no neuron in the layers between them can be in the pathway as well. It is possible to imagine such functions where the intermediate representation does not correlate with either input or output, but we expect in general,that those functions will not be learned regularly in practice. However, future work will need to verify this assumption.

We expect the second sub-case to be more common. This, like the first case, we determine to be reasonable behavior depending on the intended analysis goal one has. If we imagine the case with multiple time steps of a recurrent neural network, it has been reported that some neurons act like a state machine and represent the change in state within a model (Giles et al., 1995; Tiňo et al., 1998; Hudson and Manning, 2019). It

would be expected that these neurons would occur in a pathway with other neurons that activate in that state. In this way that pathway does not represent the flow of information, but a group of neurons that have a purpose which is the manner in which we defined neural pathways. This would be able to be aligned with a certain type of metadata that might be of a different form than pathways that one would find in a feedforward neural model. This highlights the importance of knowing what type of answers one wants from a model from the analysis when selecting neurons for the neural pathways approach.

3.4 Conclusions

In this chapter, we have introduced an approach for neural interpretation using neural pathways on recognizing textual entailment and named entity recognition. This serves as a foundation for addressing the issues laid forward in this thesis. By abstracting away from individual neurons and combining linear probes, task knowledge, and correlation techniques, insight into the knowledge learned by the neural models have been made more transparent. And furthermore, the technique can be extended to allow for rich comparisons between models even when they have dissimilar architectures. We find this general interpretation method draws similar conclusions to highly domain-specific analyses, and while it will not replace the need for deep analysis, it provides a simple and effective starting point for a broad class of models. Chapter 4 will expand the theoretical guarantees of the method along with a metric for evaluating the interpretability of a model with the method, and Chapter 5 discusses the limitations of the method along with a practical walk-through of the method.

4 Neural Pathway Alignment with Bayesian Networks

Like all neural probe-derived techniques, the neural pathways approach in the previous chapter is a purely correlation based approach and thus can only hint at the presence of knowledge encoded within a neural network. We extend the theory of the neural pathways approach to show that under certain assumptions, the method can be adapted to optimize for alignment with causal variables within a Bayesian model. Causal approaches have recently become of interest in the neural network interpretation community (Hu and Tian, 2022; Geiger et al., 2022; Lin et al., 2022), but have remained a challenge for post-hoc intrepretability methods that cannot rely on model affecting interventions (Fan et al., 2021).

Causal modeling is a method for representing the causal relationships within a system, with a causal relationship defined as a specific type of connection between two or more variables where a change in one variable (often termed the "independent" variable) directly results in a change in another variable (often termed the "dependent" variable). Neural networks, on the other hand, are learned functions that map an input to an output. They have no guarantee of causal reasoning, even when such reasoning is desirable. From our experiments in the previous chapter, we demonstrated that coordinated sets of neurons can be identified, and furthermore we showed that the activations of these pathways can be reasonably aligned with external knowledge such as task metadata and implicit features (e.g. parts-of-speech, capitalization, etc.).

If a specific task that a neural network is trained to perform has an underlying causal structure, we may suppose that a neural network, despite the lack of a guarantee to do so, may learn functions approximating those underlying causal components. Verifying that a neural model operates with a similar causal structure to those developed by theory can provide confidence that the model is performing the task in a manner consistent with human expectation and not finding invalid shortcuts within the data. As confidence in neural model output validity is a common barrier to entry for adoption of this technology, this is a step of utmost importance.

The primary contribution in this chapter and thesis is a method for verifying the presence of and aligning with theoretical causal structures with the flow of data observed within a neural model. Because of the ability of the neural pathways approach to reduce the effective number of activations that must be interpreted, we use that as a foundation for our experiments. We first provide a theoretical overview of how the pathways approach can be related to causal models, then we provide a new metric to evaluate the alignment between the pathways extracted by a neural network and the variables within a causal model. Lastly, we provide a set of experiments to validate the approach.

Real-world problems have complex causal structures that are difficult to exactly define in their entirety. We therefore build the foundation of this research via synthetic data generated from a defined Bayesian model where can know *exactly* how the causal structure is defined. In later chapters we apply the method on multiple natural language tasks to demonstrate its applicability to real-world problems. In the remainder of the chapter, we validate the ability of the method correlate with causal variables and for the precision and recall-like metrics that we present, accurately represent the quality of alignment between the pathways and the Bayesian network.

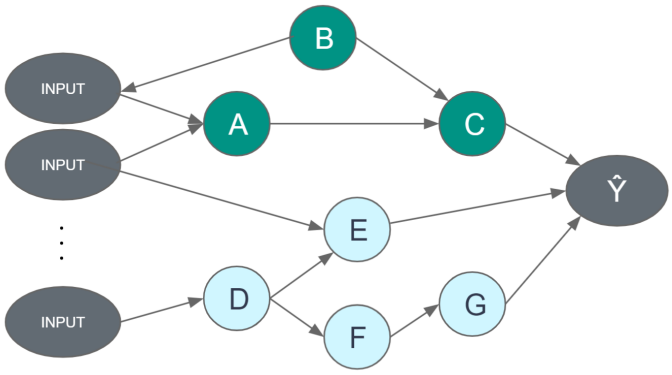


Figure 4: An example Bayesian network defining some causal structure.

4.1 Correlation of Neural Pathways with Causal Variables

Ideally, it would be possible to create a direct mapping from all of the variables within a Bayesian network to the parts of a neural network that model the same information. However, this is non-trivial during a post-hoc analysis because, without interventions, there are limitations on what information is obtainable.

Consider a neural network that is modeling the process defined by the example Bayesian network in Figure 4. The only guaranteed overlap between the neural model and the Bayesian network are the inputs and the output (darkly shaded). Superficial logic would seem to dictate that in order to correctly model the process of predicting $\hat{Y}$ given the INPUTs, the neural network must also model some or all of the intermediate variables part of the casual structure, A to G. However, it is not reasonable to make this assumption as it has been repeatedly shown that neural networks find "shortcuts" to make predictions (Du et al., 2021; Eisenschlos et al., 2021). This propensity of neural networks to find unexpected methods to solve tasks is precisely the reason that neural network interpretability is such an important area of research.

Furthermore, even if the neural model is actually representing the causal structure defined by the Bayesian network, it may not be discoverable with the resolution that the Bayesian network may indicate. If one is given only the inputs and outputs, there may be sets of variables within the causal structure that cannot be statistically separated from each other. From Figure 4, within each lightly shaded region (e.g. $\{A, B, C\}$ and $\{D, E, F, G\}$) it is impossible to disentangle the effects of each variable from each other. At best, one could only conclude that a functional component is correlated with the set of variables. Separable variables can easily be found by converting the Bayesian network into a factor graph (Cowell et al., 1999) whereupon separable groups of variables are visually separated (Figure 5). In practice, this can result in the apparent correlations between functional components and intermediate causal variables appearing weaker than expected. It also enhances the importance of choosing good Bayesian networks to plausibly explain the neural model.

It is thus required to treat the independent sub-graphs of the Bayesian network as

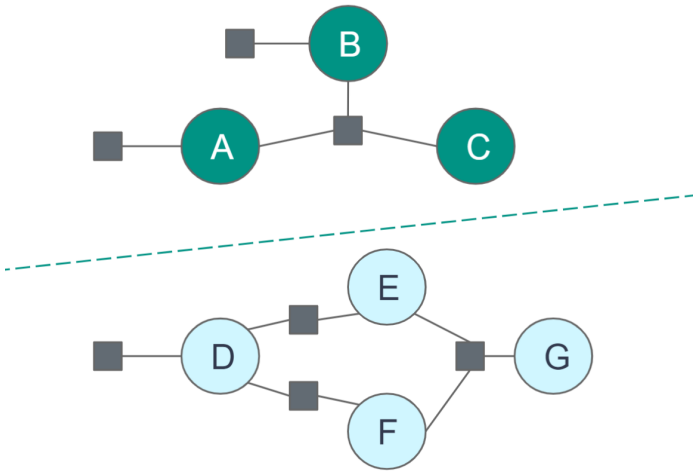


Figure 5: Factor graph of Bayesian model from Figure 4.

the unit of analysis when it comes to alignment.

PCA, the method of dimensionality reduction used in Fiacco et al. (2019a) does not guarantee independence between components; it only indicates that variance is maximized in the orthogonal projection of the data onto the principal subspace (Hotelling, 1933). These orthogonal dimension can be helpful to identify groups of correlated neurons, but they do not implicitly have any reason to correspond to the intermediate latent variables in the Bayesian network.

Fortunately, factor analysis (Tipping and Bishop, 1999), a closely related, though latent variable based, dimensionality reduction technique to PCA, ensures the independence of its factors and can be used as a drop in replacement to PCA. The resulting change to the procedure simply replaces mentions of components for PCA with factors. Like for PCA, a Varimax rotation should be performed on the factor analysis to yield more consistent and interpretable factors (Kaiser, 1958).

The reason this simple change allows for a reasonable alignment with the latent variables in a Bayesian network is as follows: given a neural network, NN, with activation matrix, A (as above), a factor analysis is performed yielding a set of factors, F. For each $f_i, f_k \in F$, $f_i \perp\!\!\!\perp f_k | X, Y$, where X is the set of inputs to the neural network and Y is the set of predictions from the neural network. With a sufficient number of factors such that factors, F, contain all independent factors in A, if there exists a common latent variable in both NN and factorized Bayesian network, G, with factors $g_i \in G$, then there will be some $f_i \overset{\propto}{\sim} g_j$. This forms the basis of our alignment.

Excess functional components may represent exploitable biases or artifacts specific to the dataset while too few functional components may represent an incomplete Bayesian representation of the task.

4.1.1 Correlation vs. Causation for Neural Pathways

An important challenge to this work arises from the mingling of the concept of causation with neural network interpretation. It is important, therefore, to be specific on where causal relationships exist within this approach. The only causal relationships in this work are between causally linked variables within the Bayesian network. In this work, we *cannot* make any conclusions about causality within the neural network nor can we make any conclusions involving causality between the Bayesian network and the neural network.

Like other neural probe techniques, functional components are a purely correlation based approach and thus can only hint at the presence of knowledge encoded within a neural network. Our extension extends the approach to allow interpretation via a structure of connected concepts rather than a set of disparate attributes. We extend the capability of probing techniques by defining subsections of the model and the theoretical causal structure of the task that can be reasonably be compared.

4.1.2 Precision and Recall for Evaluating Alignment

In order to compare one model's ability to align with a causal network with another, we define precision-like quantity and a recall-like quantity that are meaningful in context of the alignment of factors of a factor analysis and a Bayesian network. Classically, precision measures the proportion of correct, positive identifications over the total number of positive identifications made. Recall, on the other hand, measures the proportion of correct, positive identifications over the total number of expected positive identifications. These metrics are designed to measure the *correctness* and *completeness* of a given model, respectively. While the definitions of precision and recall in terms of ratios of correct positive identifications do not apply to the alignment problem, we nevertheless want a measure of correctness and completeness.

To satisfy the requirement of correctness, we present the question *Does the Bayesian network explain the activations of the neural network?* Our precision metric, therefore, must indicate that the factors identified as related to the variables in the Bayesian network do, in fact, correlate with the neural model's activations. This can be computed as the average of the best aligned correlations between the factor activations and the state of the intermediate sub-graph of the Bayesian network. The quantity of best alignment in this case is the set of one-to-one mappings of factors to states that maximizes the weighted average. More formally, we can define our precision metric, P_{align}, with Equation 1:

$$P_{align}(F, G) = \frac{\sum_{(f,g)\in \text{best}(F,G)} |r_{f,g}|}{\min(|F|, |G|)} \tag{1}$$

where F is the set of factors from the factor analysis, G is the set of factors from the factorized Bayesian network, *best* is a function producing the tuples of the best aligned set of factors, $EV(f)$ is the percent variance explained by factor, f, and $r_{f,g}$ is the Pearson's correlation coefficient between f and g over the validation set.

For completeness, we instead present the question *Can all of the neural model's activations be explained by the Bayesian network?* Our recall measure, can thus be

defined as the weighted average over all the factors from the factor analysis of the correlation coefficients between a given factor and the state of the sub-graph of the Bayesian network with which it best aligns. We can define our recall metric, R_{align}, with Equation 2:

$$R_{align}(F, G) = \sum_{f \in F} \mathrm{EV}(f) \times \max_{g \in G}(|r_{f,g}|) \tag{2}$$

One can see that $P_{align}(F, G) = 1$ when all of the uniquely aligned factors are perfectly correlated with each other and that $R_{align}(F, G) = 1$ when every factor from the factor analysis correlates exactly with an observed intermediate state of the Bayesian network.

It is also important to note that we are using the absolute value of the correlation coefficient because it does not matter for purposes of alignment whether there is a positive or negative correlation; all that matters is the degree to which they correlate.

4.2 Experiments

To experimentally validate our approach, we designed two sets of experiments. The first set of experiments is designed to empirically evaluate the ability to extract pathways from a neural network that align with underlying latent variables in the data. Earlier in the paper, we provide a theoretical argument for why factor analysis should be a more appropriate dimensionality reduction technique than PCA for this purpose, and we are interested in whether that holds in practice. The second set of experiments is designed to evaluate whether the metrics that we proposed in Section 7.4.2 provide a good measurement of the alignment between the functional components of the neural network and the underlying Bayesian network.

For the following experiments, we use synthetic datasets so we can precisely control the data distributions so we can confidently use a Bayesian that aligns task to test how well the neural network aligns with it. With natural datasets, it would be incumbent on the researcher to use their theoretical knowledge of the task to devise a Bayesian network that represents their understanding or expectation for the underlying process behind the task.

4.3 Constructing the Synthetic Datasets

For each experiment, a Bayesian network (e.g. Figure 6) and the conditional probability tables for each of its variables are defined. The specifics of the structure and the choice of conditional probabilities is dependent of the experiment. From those defined structures, datasets were generated via forward sampling (Henrion, 1988) using the PGMPY (Ankan and Panda, 2015) python library for probabilistic modeling. The number of samples for each dataset were chosen to be approximately the minimum number of data instances required for neural models used in the experiment to show a convergence. Through the forward sampling algorithm, we could record the states of all of the variables in the Bayesian network for each sample. These states could either be used for input information for the neural network, prediction targets, or latent states

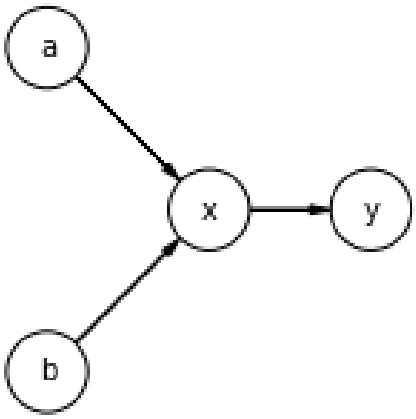


Figure 6: PGM graph used to generate the synthetic datasets for the experiments in Section 4.3.1

to attempt to align with learned functional components. For each synthetic dataset, we performed a 50/50 split into a training and validation set. We did not use any of the generated data for a separate evaluation set as this is a method for error analysis of a neural model, and thus, like tuning, it would be invalid to use on dedicated test data.

4.3.1 Comparing Factor Analysis with PCA for Interpretable Functional Components

To compare the viability of the two dimensionality reduction techniques for extracting functional components, we consider how robust each technique is to recovering alignable functional components when varying the coupling strength between variables of the underlying distribution. Coupling strength, in this context refers to the certainty that a dependent variable in the Bayesian will be in a specific state given the states of its parents.

Experimental Setting: To implement this condition, defined a set of values to represent high, low, and inverse coupling (0.8, 0.5, and 0.2 respectively). These values were used to populate the conditional probability tables (CPT) of the PGM in Figure 6 as values for $p(* = 1|...)$ where $*$ refers to each variable in the Bayesian network (the corresponding $p(* = 0|...)$ was set to one minus the value ensure the validity of the probability distribution). We generated a full dataset of 1,000 samples for each permutation of settings, yielding 6,561 separate synthetic datasets.

We defined two tasks to train a separate neural network on for each of the synthetic datasets generated (i.e. 13,122 uniquely trained neural networks). For the first task, we used the sampled states of *x* as the input for the neural network and the states of *y* as the predicted class for the neural network. For the second task, we used the sampled states

of *a* and *b* as the input to the neural network with the same predicted class. The values of states of the remaining variables that were sampled from the model for each task were reserved for analysis. These values are the states of the variables that the models sere not given explicitly, therefore if there is observed a correlation between the activations (abstracted via functional components) and the variable values, the neural network learned to encode at least some of the information contained within that variable.

Each trained neural network had functional components extracted such the the percent of explained variance was at least 99% via each dimensionality reduction technique. It was defined that a functional component had aligned with an unobserved variable if a trained logistic regression model given the functional component as input could predict the unobserved variable with a Cohen's κ of greater than 0.6 indicating at least "substantial agreement" (Warrens, 2015).

Measuring Robustness We define robustness as the ability of a dimensionality reduction technique to identify valid functional components in a consistent and repeatable manner. To evaluate this quantity, we train a classifier for each technique to predict whether or not it will find an aligned functional component from the neural model given the settings of for the CPT of the Bayesian network. A high accuracy in this classifier would indicate that there are more well defined boundaries with which the method is reliable. This is an important quality to have in order to trust the interpretation.

Furthermore, it is important that there is not an absolute loss of performance when switching to the new method. In this respect the difference in percent of conditions in which a functional component has been successfully identified should be small or in the favor of the new method.

Model Architecture for Analysis We used a simple feed forward neural network with two hidden layers for both tasks. The network for the first task had one input, *x*, while the network for the second task had two, *a* and *b*. Both networks had two, 8 unit hidden layers. An excess of neurons were used to provide leeway for multiple pathways to arise if necessary and overfitting was not a concern as we could generate as much data as necessary to mitigate it. We expect there to be very few pathways given the simplicity of the task.

4.3.2 Evaluation Metric Validation

In this set of experiments, we validate the reasonableness of the alignment precision and recall metrics we defined in Section 7.4.2. As our interest, in the long term, is in understanding neural models for NLP, we specifically designed our synthetic dataset to have some similar, though considerably simplified, properties to text. Specifically, we explore sequential data where each time step is dependent on a previous time step. This type of data furthermore allows for a more stark comparison between models as we can use neural architectures that have a different capacity for modeling sequential dependencies. Because the proposed metrics are primarily meaningful in the comparison of values between different models or Bayesian networks, the ability to have a clear difference between the models that we use in this experiment is advantageous.

Experimental Setting: To realize the sequential dataset, we defined the Dynamic Bayesian network (Koller and Friedman, 2009) depicted in Figure 7. We generated 1,000 sequences from the PGM, split into 500 sequence train and validation sets. Each

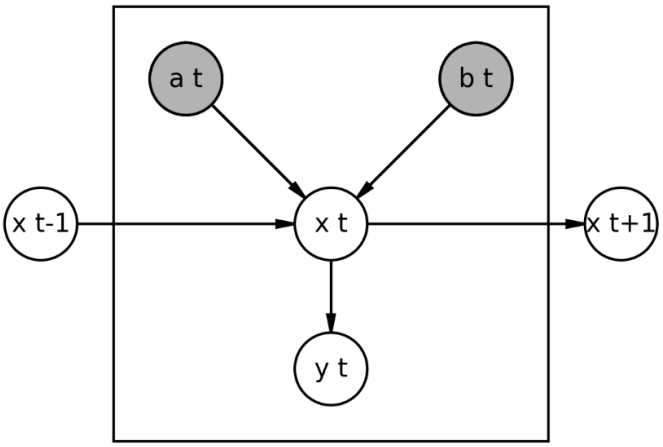


Figure 7: Dynamic Bayesian network used to generate the synthetic datasets for the experiments in Section 4.3.2

sample is a sequence containing 10 elements where each element has 2 input features (a_t and b_t) and a binary class output (y_t). The intermediate variable, x_t, is dependent on both a_t and b_t as well as x_{t-1}.

We specifically designed the CPT of the Bayesian network to have a certain set of behaviors such that we could easily identify them within the functional components. Depending on the state of x_{t-1}, x_t would represent a soft version of one of the following Boolean functions of a and b: XNOR if $x_{t-1} = 1$, NAND if $x_{t-1} = 0$, or XOR if $t = 0$. Furthermore y_t was simply set to be a soft NOT of x_t, and the priors of a and b were set to $p(* = 1) = 0.5$.

Functional components were extracted from each model for each instance in the validation set and their alignment scores were computed with respect to the states of the unobserved variables, x_t and x_{t-1}. We then performed a number of qualitative tests to verify the difference in scores were reasonable. First of these tests is to graph the average correlation between the functional component and the attributes over the 10 steps in the sequence. If a functional component is well aligned with an attribute, the correlation should stay high throughout all of the elements in the sequence approximately equally. The second test is to graph the average factor activation for each step of the sequence for instances where a certain state was present. This test can help qualitatively identify what function a functional component is performing, by showing potential combinations of states that indicate activity in the factor.

Models for Analysis: We use two models for this set of experiments with differing capabilities for modeling sequential dependencies: a feed-forward neural network (FFNN) that only looks at each element of the sequence independently, and an LSTM (Hochreiter and Schmidhuber, 1997) which can use context for its predictions. The FFNN takes in 2 inputs and has 2 fully connected hidden layers with 8 hidden units each and a sigmoid activation. The LSTM has 8 hidden units and makes a prediction at each time step (i.e. not seq2seq (Wiseman and Rush, 2016)).

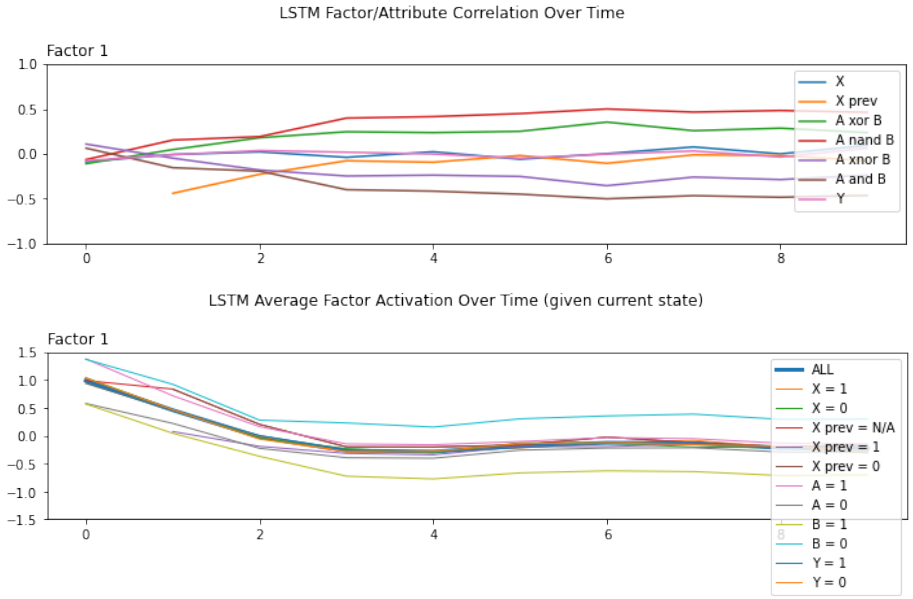


Figure 8: Example of the graphs generated for a given factor for qualitative analysis in Section 4.3.2.

4.4 Results and Discussion

Task	Unobserved Variable	PCA	FA
X $\longrightarrow$ Y	A	45	**62**
	B	50	**57**
A, B $\longrightarrow$ Y	X	**117**	97

Table 4: Number of CPT settings (out of 6,561) where the given unobserved variable was alignable with a factor extracted from the neural network trained on the given task.

4.4.1 Comparing Factor Analysis with PCA for Interpretable Functional Components

For both PCA and factor analysis, only a very small fraction of settings for the conditional probability table could be aligned with the activations of the neural network (Table 4). This is not unexpected as when there is very low coupling between the variables (i.e. a coin-flip for whether a state in influenced by its parent), it is not reasonable to expect that the model will be learn that connection. Qualitatively, we observe this behavior. Furthermore, because a is not conditionally independent with b given x, in

cases where we are given x as the input, it is likely that there will be some uncertainty about a and b if they both contribute significantly to x. This finding should be a warning to all probing tasks as there is a narrow band of relationships between variables that can reliably be detected. However, in practice, there is usually very strong coupling between variables that are of interest. PCA and factor analysis perform similarly in this respect so there is not a considerable loss in performance by using the different dimensionality reduction technique.

Task	Unobserved Variable	PCA	FA
X $\longrightarrow$ Y	A	0.907	**0.920**
	B	0.674	**0.921**
A, B $\longrightarrow$ Y	X	0.350	**0.801**

Table 5: Accuracy of the decision tree model for predicting whether or not a setting for the CPT would be alignable with a factor.

On the other hand, factor analysis significantly outperforms PCA when it comes to systematically identifying under what conditions alignment can happen. Table 5 shows that while the ability to distinguish between settings where a can be aligned via factors are relatively similar for both PCA and factor analysis, the accuracy of the decision tree classifier was significantly higher ($p < .001$) for all of the attributes. We attribute this performance increase to the probabilistic formulation of factor analysis that can be a better analogue to the variables within the Bayesian network. It is also notable in that the decision trees for the PCA factors are considerably simpler than for the factor analysis. This is likely because only simpler rules could be made to approximate the division as there was insufficient consistency within settings as compared to factor analysis.

4.4.2 Evaluation Metric Validation

Model	Num. Factors	Task κ	P_{align}	R_{align}
FFNN	2	0.000	**0.342**	**0.579**
LSTM	7	**0.348**	0.264	0.400

Table 6: Performance (Cohen's κ) and alignment metrics for FFNN and LSTM from Section 4.3.2.

At first, there seems to be a contradiction in the results for the experiment to validate the alignment metrics. In Table 6 the LSTM reaches fair alignment with the synthetic task while the feed-forward neural network fails entirely (in fact it only predicted TRUE for all instances on the validation set). However, in both of the alignment metrics, the FFNN out performed the RNN. This was highly unexpected as it was our hypothesis that better model would likely show better alignment. If we do the qualitative analysis to try to examine why the alignment is better for the FFNN, we do, in fact, find that it

the activations of the FFNN do more accurately reflect the Bayesian model, validating in a quite unexpected way, that the metrics are reasonable.

The primary reason that the LSTM performs better than the FFNN on the task is best illustrated by the two graphs in Figure 24. In the top graph, each line indicates the Pearson's correlation coefficient between Factor 1 and attributes derived from the functions that the Bayesian network approximates for each time step in the sequence. The story that this tells is that, over the course of the sequence, Factor 1 gets more correlated with the function a NAND b, the top red line. As mentioned in Section 4.3.2, a changing correlation over time is a good indicator that the factor is *not* aligned with the attribute. In this case, we can use the bottom graph to see that early on, this factor is the most active in the beginning and continues to be somewhat active when the input $b = 0$. Because of how the Bayesian network was defined to generate the data, sequences generally trend such that x will be 1 more often than not, especially if either of the inputs are 0. So what this factor is likely doing is tracking how far along the sequence is as it means that y will more likely be 1. That is, the alignment metrics enabled us to find that the recurrent neural network was finding a shortcut, even on this extremely simple task.

The FFNN, on the other hand, had a factor that was consistently highly correlated with a NAND b which turns out to be the most common function that the Bayesian network performed while generating the data.

4.5 Conclusion

Neural networks, with their intricate operations and architecture, pose an intriguing challenge when attempting to align their behaviors with structures such as Bayesian networks. Neural networks' propensity to find unconventional prediction routes emphasizes the necessity for interpretative tools that map functional components to features, grounded in human understanding.

Our exploration into this alignment problem highlighted the merits of factor analysis over more simpler dimensionality reduction techniques, such as PCA. While PCA is very fast and scalable, factor analysis, with its emphasis on factor independence, offers a closer alignment with the variables of Bayesian networks. This is not to suggest that factor analysis is the ultimate tool as other dimensionality reduction techniques that optimize factor independence (e.g. Independent Component Analysis) share this property. However, it does represent a step toward theoretically sound component based neural network interpretability.

In the realm of neural probing, the distinction between correlation and causation remains a crucial aspect. Although Bayesian networks offer a structured perspective on causality, making definitive causal claims within the complex web of neural networks is not straightforward. Yet, the methodologies detailed in this work have extended our interpretative capabilities, allowing for a richer understanding of networks through interlinked concepts, as opposed to isolated attributes.

Precision and recall, reconceptualized for this unique challenge, provide an insightful lens to gauge alignment success. The conceptual metrics, $P_{align}(F, G)$ and $R_{align}(F, G)$ bring clarity to the alignment task. They offer a methodical way to

quantify the alignment between the complex behavior of neural networks and the deterministic structure of Bayesian networks.

In wrapping up this chapter, it's evident that optimizing pathways for independence is a meaningful step forward in the broader context of neural interpretation. The insights and methodologies laid out will likely serve as invaluable reference points for upcoming endeavors in the domain of neural network interpretability. In later chapters we explore the limitations of this method and enforcing similar independence conditions on the human understandable features used in the analysis.

5 Complete Neural Pathways Approach with Model Comparison

The overarching goal for the neural pathways approach is to be able to use the abstractions of neural pathways to identify causal connections learned by the network and distinguish these connections from spurious correlations or proxy variables. Because of the considerable differences between the concepts of a neural network and a causal model, we must rigorously define the progression of steps from the inputs and outputs of the network to the knowledge of a causal structure for a task that an expert would have. We visualize the path via a comprehensive flowchart (Figure 9) where the givens of the process (rhomboid boxes) follow a series of steps to the potential outcomes of the process (rectangle boxes marked in purple). Later in this chapter we examine each of the steps in the flowchart (circular boxes) in more detail.

In order for this procedure to be successful in practice, there are two sets of requirements that must be met: the first set is simply the required data that one must have to perform the process, the other is a set of assumptions we make regarding the neural network and the data which through this chapter and proposed work demonstrate are necessary and sufficient to result in the outcomes provided.

There are two key assumptions that underlie this method: first, causal connections exist within the data with some detectable strength, there may be confounds, irrelevant knowledge, or noise, but they must exist; second, there can be many pathways, but only some of them matter. Pathways that either contribute little to the model outcome or contribute strongly to a very small number of cases, can be safely ignored.

Given the assumptions, the neural pathways method has four possible outcomes for each pathway/attribute pair:

1. The pathway does not have sufficient influence over the outcome of the model to warrant further analysis.
2. There is no evidence of a causal relationship between the the pathway and the attribute.
3. The attribute is likely a proxy for the function that the pathway is performing.
4. The function that the pathway is performing likely includes the attribute.

These outcomes can be grouped into two categories: the first two narrow down the pathways that can be analyzed, while the latter two distinguish between the possible reasons that there exists a correlated pathway/attribute pair. In the remainder of this chapter, we walk-through an example of the process to provide intuition on how these outcome arise and the decisions that must be made to be confident in the results.

5.1 Example Walk-through of the Flowchart

To aid in understanding the intuitions behind the flowchart, we present a simple example based in a real-world task to illustrate where the decision points in the chart arise and how one could arrive out each outcome.

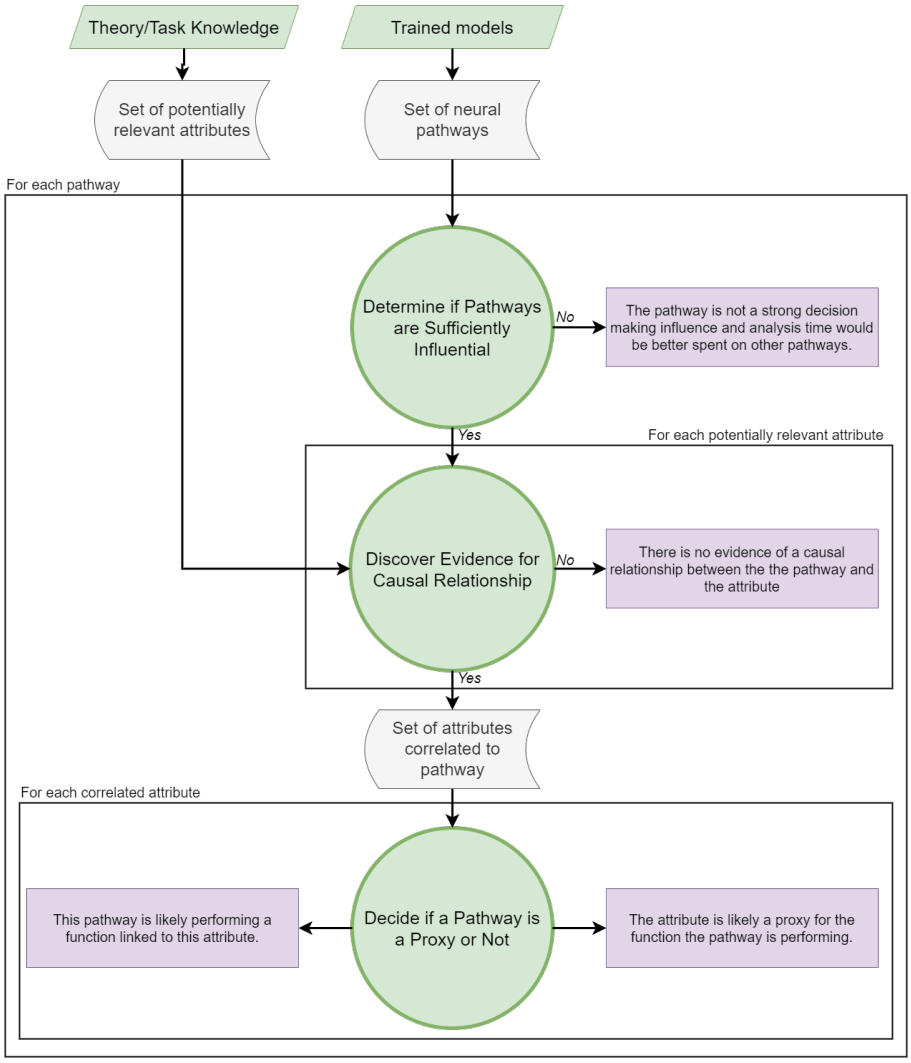


Figure 9: Flowchart illustrating the process of determining causal connections within a neural net via neural pathways.

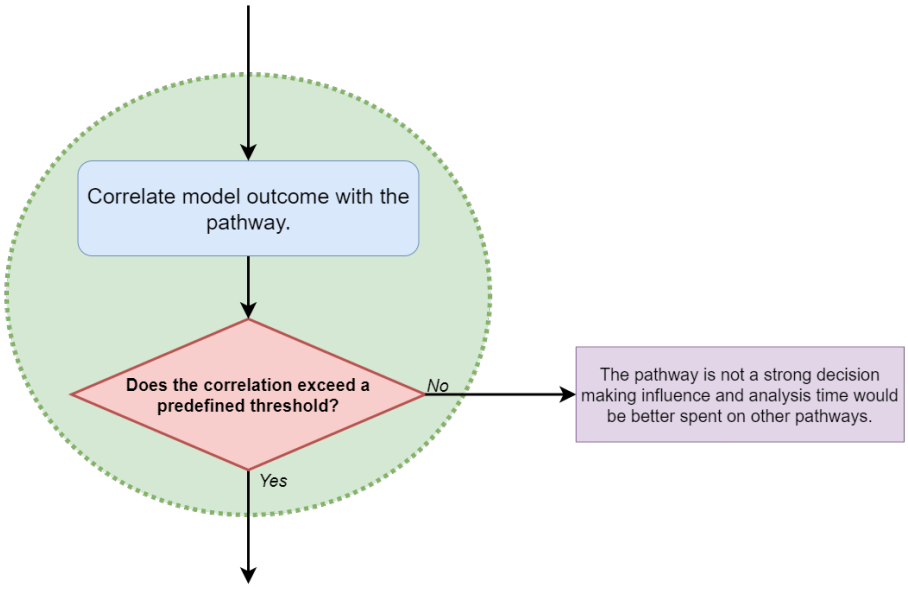


Figure 10: Detailed flowchart for determining sufficient influence of pathways. The input to the chart is a set of pathways extracted from a neural model and the output is a set of pathways that are worth investigating.

The task that we use for this walk-through is Named Entity Recognition (NER) where a model needs to identify the named entities within a text. For this specific example, we assume that we are given a complex, probe-able, neural network that was trained on data derived from a hypothetical story called *The Wacky Adventures of Xavier and Xander*. For illustration purposes, we have selected three attributes that we hypothesize are potentially being utilized by this network: *(A) Word is in a list of names*, *(B) Word begins with a capital letter*, and *(C) Word follows a '.'*. The text set for this task where we perform our example analysis uses data derived from a hypothetical book called *The Unabridged History of the Xylophone*. The process for the neural pathways approach can be organized into three sections based on what outcomes are being distinguished: Determining sufficient influence, discovering evidence of a causal relationship, and deciding if that relationship is a proxy or not.

5.1.1 Determining Sufficient Influence of Pathways

The first phase of the process, broken down into more detail in Figure 10, is to select neural pathways for analysis. With complicated models with many potential pathways and attributes, it can be expensive to perform an analysis for every possible pathway/attribute pair. However, it is important to ensure that the pathways we choose to perform the analysis on will be sufficiently relevant to the questions we want to ask. In our example, we run the model on the test set to extract the activations which we can use to generate

pathways via the procedures from earlier chapters. For purposes of our example, we assume that many pathways are generated at this step thus requiring a method to select the most relevant.

This is the first decision point in the process. We correlate each pathway with the outcome of the model to get a rank for how influential the pathway is within the model. We must then choose a threshold for the correlation, below which we will consider the pathway not sufficiently important to continue analysis. It is not obvious what this threshold could be as it balances the number of pathways that will need to be analyzed with the amount of information retained. If it is set too high, there is a possibility that notable causal connection could slip by unnoticed; if it is too low, than considerable effort will be allocated to inconsequential pathways. We devise the experiment in Section 5.2.1 to provide an intuition on how varying this threshold can influence the outcomes of the procedure. It is thereafter at the discretion of the researcher for where along the scale they desire to set their threshold.

5.1.2 Discovering Evidence for Causal Relationships

To continue the example, we determine if there is any potential connection between the pathways and the attributes available. Let us assume that from the previous step, three pathways had correlations with the outcome that exceeded our threshold. We then compute the correlation of each of the three pathways that we identified with the three attributes that we are interested in. we find that one pathway (**Pathway A**) correlates to *words that are in a list of names*, one pathway (**Pathway B,C**) correlates with both *words that begin with a capital letter* and *words that follow a '.'*, and the last pathway does not correlate with any of the attributes we have. We can therefore determine that we have no evidence that the last pathway has a causal relationship within the model. It is possible that there exists a causal relationship that it corresponds to, but we are limited by what attributes we include in our analysis.

What is meant when it is said that the pathways correlate with an attribute must be specified further. It is another threshold that must be defined. The higher the threshold, the more clearly the pathways would appear to be related to the attribute. However, too high and noise within the data can obfuscate meaningful connections. Fortunately, the boundaries of what amount of causal connection can be detected via correlation is the subject of the experiments in Section 4.2, and we can apply those lessons towards choosing this threshold.

5.1.3 Deciding if a Pathway is a Proxy

Finally, for the remaining two pathways that were determined to be correlated with one or more attributes, we must distinguish between the case where the model learned and is using the attribute manner expected from a theoretical understanding of the task, versus the case where the model learned a proxy for the attribute. In our example, we find that **Pathway A** correlates with the error cases of the model while **Pathway B,C** does not. We conclude that **Pathway A** is likely a proxy for the *Word is in a list of names* attribute. We examine the error cases and find that in the test set, "Xylophone" is often present when the model makes a mistake when **Pathway A** is active. We might then be able to

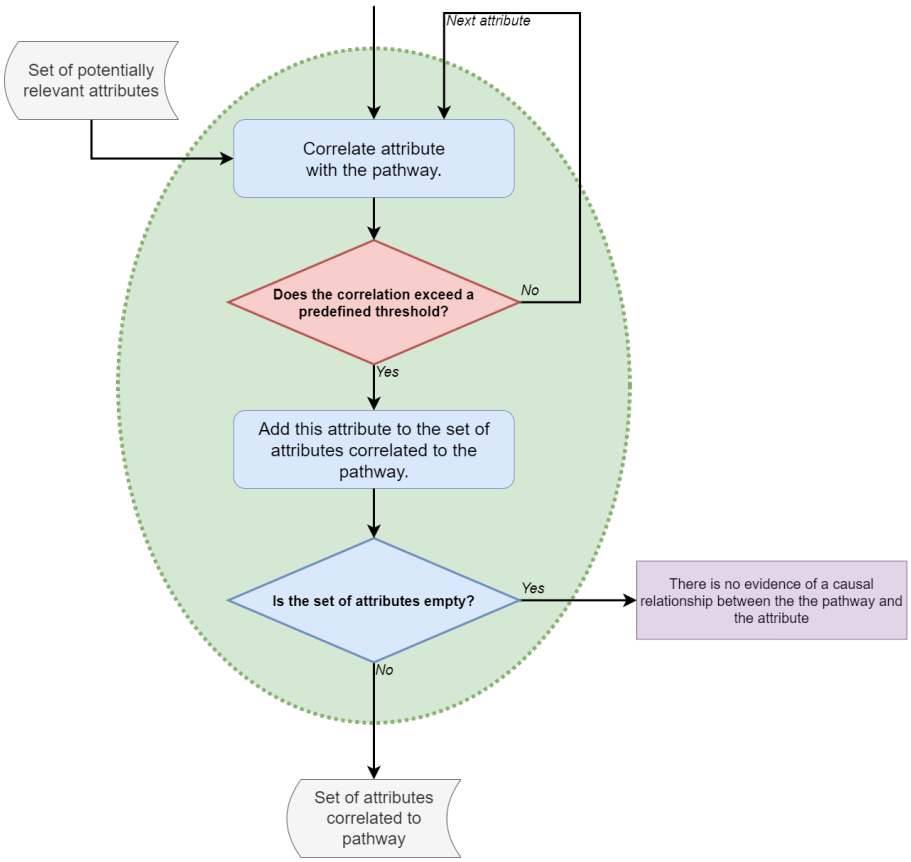


Figure 11: Detailed flowchart for discovering evidence for causal relationships in pathways. The input to the chart is the set of pathways that are worth investigating and the set of potentially relevant attributes. The output is the sets of attributes correlated to each pathway.

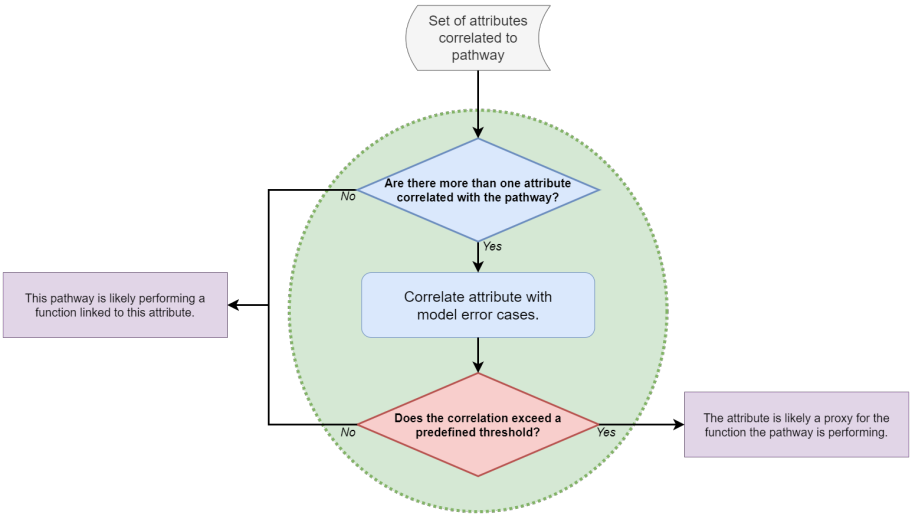


Figure 12: Detailed flowchart for deciding if a pathway is a proxy or not. The input to the chart are the sets of attributes correlated to each pathway.

intuit that the pathway has actually learned to identify words the begin with the letter 'X' as in the training data, the two most common words that started with 'X' were the names of the characters in the story ("Xavier" and "Xander"). That a word begins with 'X' is not a reasonable argument for deciding whether a word is a named entity from the theoretical perspective and so on the test set where there were many instances of the non-named entity "Xylophone," the model made mistakes. On the other hand, because **Pathway B,C** is performing the function that is equivalent to the more theoretically sound attributes, it does not tend to contribute to errors in the test set.

As data outside this example can be full of noise and confounds, the threshold for how correlated a pathway must be with the error cases of the model for it to be considered a proxy must be defined precisely. We construct the experiment described in Section 5.2.2 to examine the extent to when confounds can be identified via this method. This allows for a clear limit to the capabilities of this method providing the parameters within which it is valid.

5.2 Synthetic Alignment Experiments

In line with the intuition provided in Section 5.1, two additional experiments are proposed, each building upon each other with increasing complexity. Because the purpose of these experiments is to get precise measurements the of correlations between a neural model and latent causal variables, we use a series of synthetic datasets. Through these datasets we can precisely control the causal structure of the data in a way that is impossible for natural data.

A similar method is employed as for the experiments in Section 4.2 to generate the synthetic datasets. A PGM is defined and Gibbs sampling is used to get values for the

variables. With this method we can manipulate the extent to which there is a causal connection between variables, the strength of that connection, and whether there are additional variables that can be used as proxies.

Furthermore, we can include additional data in the synthetic datasets that can be generated from a different distribution or set of rules that can serve as data points that are exceptions to the general rules defined in the PGM. Along with the ability to add noise to sampling procedure, this allows us to validate the robustness of the method to extract valuable information from the neural network and set a reasonable threshold to maintain the most signal.

5.2.1 Threshold for Influential Pathways

To more precisely define the what the optimal threshold is for a pathway's correlation with an attribute to indicate that a pathway is influential enough to be relevant for analysis, we repeat the steps from the experiment described in Section **??** several times. On each repetition, we vary the threshold for which pathways we analyze. We record for the purposes of this experiment the resulting band of interactions between the random variables that the method could discover.

The measure of correlation that we use for this experiment is the Pearson's correlation coefficient between the the pathway activation and the model prediction. If the model is a multi-class classifier, we use the maximum absolute values of the correlations between each classes. We do this so we can be confident that we do not miss an important positive or negative correlation, erring on the side of including more pathways for analysis.

Changes in the boundaries of the method from truncating the neural pathways analyzed can be used to determine an acceptable threshold for choosing which pathways to explore in detail. The results of this experiment can inform future users of this experiment on what information or discovery potential is lost by reducing the number of pathways explored. This would allow them to be confident that they are not missing major connections within their model while using their analysis time most effectively.

Potential Challenges: It is possible that with experimental settings from Section **??**, the resulting model will have insufficient unique pathways to make the threshold meaningful. Potential modifications to the method, should this be the case, may include an increase in the complexity of the PGM used to generate the synthetic data. This may enable the model to learn more independent functional components.

5.2.2 Threshold for Proxy Connection

Our second proposed set of experiments is designed to determine what degree of correlation between an attribute and the mistakes a model is sufficient to determine that the model's knowledge of the attribute is resulting in errors in the task. We vary the underlying PGMs used to generate the synthetic data that for a similar experimental structure as before. This enables us to exhaustively test for types of connections found in causal structures including confounds that affect either or both of the inputs and outputs visible to the neural model.

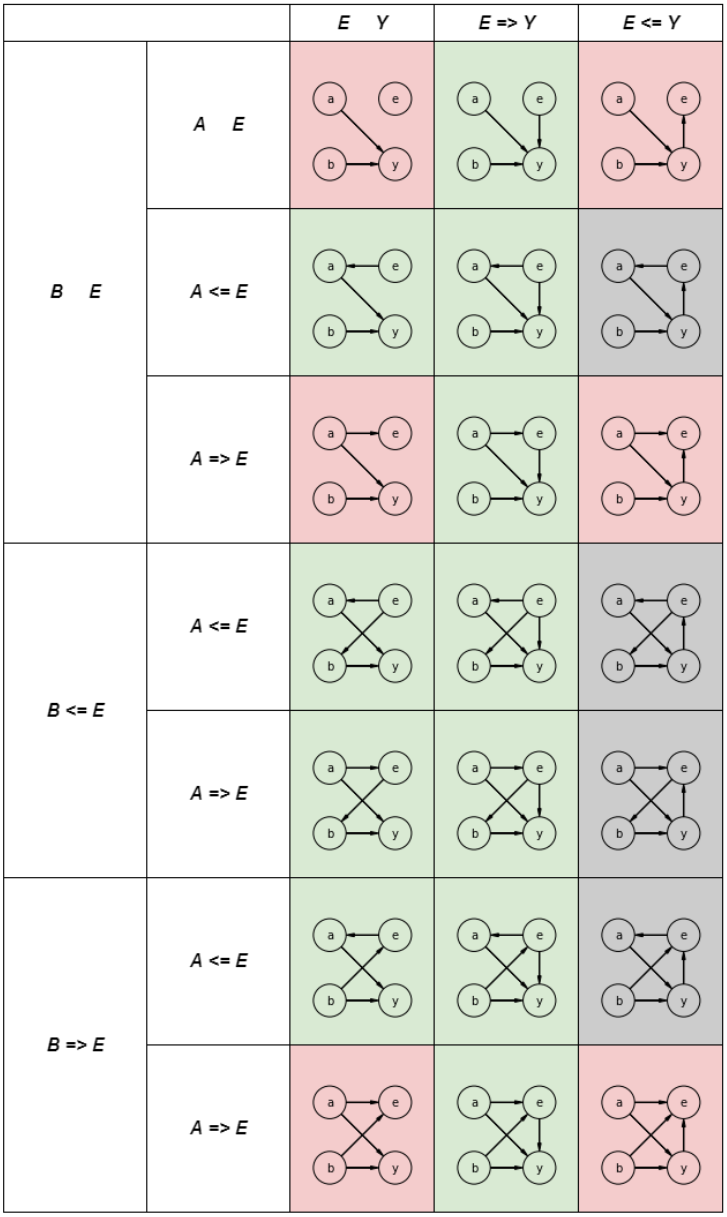


Figure 13: Exhaustive set of PGM structures used to determine when our method can distinguish between causal networks where there is a causal link from e to y (green) and those that do not (red). Graphs that have gray shading have dependency cycles and are excluded from the experiments.

Types of Causal Connections: In our experimental setting a and b correspond to the inputs into the neural model and the sampled value of y is the value that the neural model is trained to predict. Figure 13 shows all of the perturbations of this causal structure to include some other variable e which we will use as the attribute of interest for the neural pathways analysis.

This list is exhaustive subject to two constraints: (1) there is never to be a direct connection between a and b and (2) b will never have a connection to a variable that a does not have a connection to. The first constraint arises from the requirement of independence for pathways to serve as analogs to causal variables that we explore in Chapter 4. The second constraint is enforced because we ignore symmetries in the structure as this is a synthetic dataset that is only an analog of possible structures found in natural data. Furthermore, our experiments also do not use several of the causal networks represented in Figure 13 colored in gray. These networks have cycles and cyclic causal dependencies are out of scope for this work.

We organize these causal networks into cases where the causal connection is valid for the model to directly use (colored green), that is, there is a causal path from e to y, and cases where it would be considered a proxy (colored red).

Expected Results: First we determine if the sampled value of e is correlated with a pathway in the neural model using the threshold discovered in the previous experiment. We will the measure the correlation between the sampled value of e with the error cases in the model. As the data was generated probabilistically, the model should not be able to obtain a perfect prediction on the test set despite the causal structure being so simple.

Because we know what the causal connection is between e and y, we can determine if there is a difference in the recorded error case correlation between cases where there is a causal path from e to y and cases where there is not causal path.

Results: From Figure 13, only a subset of the causal relationships yielded data from which a pathway could be correlated to e when the neural network was trained on y given a and b. A qualitative analysis of these failure modes lead to the following conclusion: for a specific pathway, p, that will be correlated to attribute, e, to be able to be learned by a feed-forward neural network, the following condition appears to hold: e must be an ancestor of at least one element of the set of input variables, $\mathbf{X}$. If we repeat the experiment without a direct connection between a and y, we see that this qualitative observation holds even when e is a mediator between a and y.

Furthermore, if we swap the direction of the causal influence of the edges between a and y and b and y, we find that there only is a pathway correlated to e when e is a direct cause of y. It is notable, however, in this case that the pathway correlated with e only so far as e correlated with y. In fact, with our Bayesian network definition, the pathway almost perfectly correlated with $a \not\leftrightsquigarrow b$. This further supports the previous conclusion in this section.

Challenges: During our analysis, we witnessed a surprising amount of shortcuts learned by the neural model even when the underlying causal structure is incredibly simple. This means that it may not be feasible to generate data that would guarantee the presence of a specific pathway in the neural model. Because of this, getting a well defined threshold for what is required to witness a pathway in general terms may be elusive.

It is also possible the the selection of conditional probability tables in the PGM may

a	b	e	Is e causal?	NN κ	$max(\|r(pathway, e)\|)$
A | E	B | E	E | Y	No	0.828	0.031
A | E	B | E	E $\longleftarrow$ Y	No	0.804	0.212
A $\longrightarrow$ E	B | E	E | Y	No	0.862	0.324
A $\longrightarrow$ E	B | E	E $\longleftarrow$ Y	No	0.845	0.220
A $\longrightarrow$ E	B $\longrightarrow$ E	E | Y	No	0.876	0.276
A $\longrightarrow$ E	B $\longrightarrow$ E	E $\longleftarrow$ Y	No	0.833	0.338
A | E	B | E	E $\longrightarrow$ Y	Yes	0.761	0.028
A $\longleftarrow$ E	B | E	E | Y	Yes	0.869	**0.719**
A $\longleftarrow$ E	B | E	E $\longrightarrow$ Y	Yes	0.814	**0.784**
A $\longleftarrow$ E	B $\longleftarrow$ E	E | Y	Yes	0.830	**0.847**
A $\longleftarrow$ E	B $\longleftarrow$ E	E $\longrightarrow$ Y	Yes	0.804	**0.768**
A $\longleftarrow$ E	B $\longrightarrow$ E	E | Y	Yes	0.884	**0.664**
A $\longleftarrow$ E	B $\longrightarrow$ E	E $\longrightarrow$ Y	Yes	0.825	0.**736**
A $\longrightarrow$ E	B | E	E $\longrightarrow$ Y	Yes	0.758	0.372
A $\longrightarrow$ E	B $\longleftarrow$ E	E | Y	Yes	0.884	**0.775**
A $\longrightarrow$ E	B $\longleftarrow$ E	E $\longrightarrow$ Y	Yes	0.770	**0.761**
A $\longrightarrow$ E	B $\longrightarrow$ E	E $\longrightarrow$ Y	Yes	0.714	0.216

Table 7: Number of CPT settings (out of 6,561) where the given unobserved variable was alignable with a factor extracted from the neural network trained on the given task.

have an out-sized effect on the outcome of this experiment. To test this, one would need to do an exhaustive search over different settings for the conditional probability tables. Because, even at the scale of this experiment, an exhaustive search has an intractable number of possibilities, we require a different way to test this.

5.2.3 Expanding Complexity in Underlying Causal Graphs

From the first part of this set of experiments, the results imply that for the simplest forms of causal relationships in the underlying data, a simple feed forward neural network does not learn spurious correlations in the data. This held true even when there was an indirect causal path and a dataset constructed with a sampled bias. However, from multitudes of research in this field () we know that neural models *do* learn spurious correlations that are often undesirable for the tasks they are being used to solve. We thus devise an experiment to explore the question: how complex do they systems underlying a dataset need to be for these spurious correlations in learned neural models to appear?
Systematically Adding Complexity to Causal Graphs: There are three locations we can add influence from new causal variables to expand the complexity in the Bayesian networks from Section 5.2.2: the new variables can influence the inputs, the new variables can influence the class value, and the new variables can be mediators between the inputs and the class value.

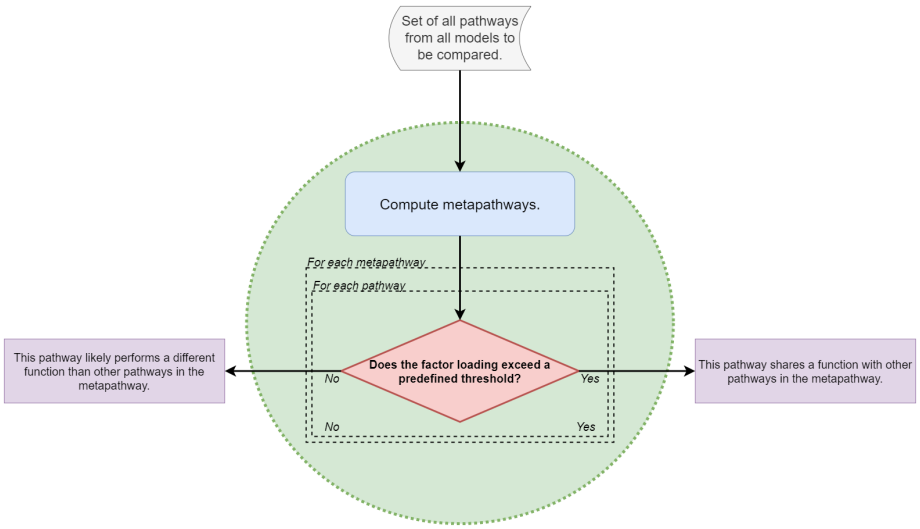


Figure 14: Flowchart addition for meta-pathway based model comparison.

5.3 Independent Meta-pathways for Model Comparison

We introduce the idea of meta-pathways as a method to compare models to determine if one or more neural models are approaching a task in the same way or in a different way. To formalize this method, we add on to the flowchart introduced in this chapter, creating a new set of outcomes and a new threshold, and define the key concept of *functional groups* which serves as the core mechanism for constructing meta-pathways. Furthermore,, *feature*, and *feature group*.

The new section of the flow chart, Figure 14, takes as input the extracted pathways of all of the different models that are being compared and results in a set of meta-pathways. The loadings that a model's pathways have on the meta-pathway indicate how correlated those pathways are with the function that the meta-pathway has extracted. Therefore, there is a threshold for the loading that determines if above the threshold, the pathways are performing the same function.

5.3.1 Walk-through

Building upon the Named Entity Recognition example earlier in the chapter we can walk through the method for model comparison. We assume that we have two neural models, trained on the same data as before. One of these models is a simple feed-forward neural network that is given a word and the characters within the word and predicts if it is a named entity or not. The second model is a bidirectional sequence model that takes in each word in the text as a sequence and predicts the sequence of named entities in the text.

Each one of these models have different information that they have access to: the first model has sub-word information, and the second has context around each word. We

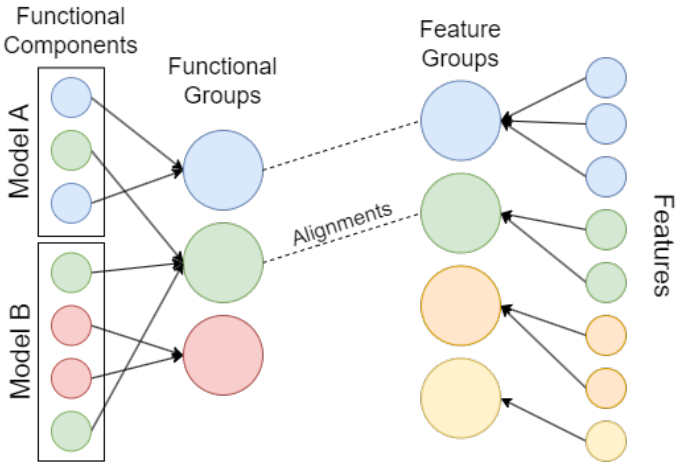


Figure 15: Diagram visualizing the structure of the meta-pathways alignment methodology. Nodes of each color represent correlated values.

extract pathways from each model on the test dataset and merge the resulting matrices so the meta-pathways can be extracted.

When performing the meta pathways, we three meta pathways, one of which includes has high loadings on only pathways from the first model, one that has pathways from only the second, and the third has a mix of both. We can then conclude that while there is overlap between the models each is solving the task differently. And upon performing a pathways analysis, we may find that only the sub-word model pathway seems to capture *Word that begins with a capital letter*, while the sequence model pathway captures Word follows a '.'. The pathway that they shared may not align with an attribute that we have elected to use, but indicates that there is an attribute that we have yet to find that both models leverage to make predictions.

5.3.2 Functional Components and Groups

From Chapter 4 functional components, i.e. neural pathways refer to the learned functions of a neural network. We might draw an analogy to how one can define and describe the common features between mammals by comparing their common and unique characteristics. In this analogy, a particular component of a dog may be a "dog leg". In a neural Automatied Essay Scoring (AES) system, these would be a group of neurons that have correlated activations when varying the input essays. The approach to extracting these pathways from a neural network consists of finding the sets of coordinated neuron activations, summarized by the following steps:

1. Save the activations of neurons for each data instance in the validation dataset into an activation matrix, A of size $M \times N$, where M is the number of data instances in the validation set and N is the number of neurons being used for the analysis.

2. Perform a dimensionality reduction, such as Principal Component Analysis

(PCA) (Hotelling, 1933), on A to get component activation matrix, T_{model} of size $M \times P$, where P is the number of principal components for a given model.

The functional groups that form the meta-pathways are collections of similar functional components. Continuing the analogy, they would be compared to the more general concept of a "leg". We compute functional groups by concatenating the dimensionality reduced matrixes, T_{model}, of the two models that are to be compared and performing an additional dimensionality reduction over that matrix to get a matrix of group activations, T. The functional components that are highly loaded onto each functional groups are considered members of that group. An important departure from Fiacco et al. (2019a), stemming from the limitation that does PCA does not guarantee independence between components, is that we use Independent Component Analysis (ICA) (Comon, 1994) instead. ICA is a dimensionality reduction technique that maximizes the independence between components, resulting in more validity in the technique's resulting alignments.

To determine if a functional group is influential in the performance of the model (designating it an *important functional group*), we can still compute the Pearson's correlation coefficient between each column of the group activation matrix and the predictions of the model, the errors of the model, and the differences between the compared models.

5.3.3 Independent Feature Groups

Features are human understandable attributes that can be extracted from an analysis dataset. In the analogy the features would represent potential descriptors of a components of a mammal, e.g. "hairy". In an AES context, these features may manifest as "no capitalization after a period". Ideally, it would be possible to create a direct mapping from each of the functional components to each of the features for which the functional component is related. However, this is non-trivial during a post-hoc analysis because, without interventions, there are limitations on what information is obtainable. Specifically, because features are not necessarily independent from each other, their correlations cannot be separated from each other, yielding imprecise interpretations. It is thus required for only independent features to be used as the unit of analysis when it comes to alignment with functional components. Unfortunately, in practice, this is a prohibitive restriction and most features that would be interesting are going to have correlations.

Fortunately, much in the same way that we can use ICA to extract independent functional components from a neural network's activations, we can use it to construct independent feature groups that can be reasonably be aligned with the functional groups of the neural networks. In the analogy, these independent feature groups can therefore, be thought of as collections of descriptive terms that can identify a characteristic of the mammal, such as "an appendage that comes in pairs and can be walked on" which would align with the "leg" functional group. In AES, an example feature group may be "uses punctuation improperly". It would be expected that this feature group would align well with a functional group in a neural AES system that corresponds with a negative essay score. Furthermore, feature groups for AES can be thought of as being roughly analogous to conditions that would be on an essay scoring rubric (as well

as potentially other features that may be intuitive or obvious to human scorers but contribute to accurate scoring).

The specific process used to define these groups is to perform a dimensionality reduction on each set of feature types that may have significant correlations and collecting them into a feature matrix. We do this process for each feature type rather than over all features at once because spurious correlations between some unrelated features may convolute the feature groups, making them far more difficult to interpret.

5.3.4 Alignment

Using ICA as the dimensionality reduction, the independent functional groups of the neural model can reasonably align with the independent feature groups using the following formal procedure: given a neural network, N, with activation matrix, A (as above), a independent component analysis is performed yielding a set of functional components, F. For each $f_i, f_k \in F$, $f_i \perp\!\!\!\perp f_k | X, Y$, where X is the set of inputs to the neural network and Y is the set of predictions from the neural network. With a sufficient number of components such that F contains all independent functional components in A, if there exists a common latent variable in both N and the set of independent feature groups, G, with components $g_i \in G$, then there will be some $f_i \overset{\infty}{\sim} g_j$.

Throughout the following sections, we explore the use of meta-pathways in the comparison of AES models.

5.4 Automatic Essay Scoring Experiments

In this section, we delve into the specific methodology used to analyze the activations of the four transformer models for AES, as well as the steps taken to prepare the data and features for this analysis.

5.4.1 Datasets

Although scoring rubrics are specific to the genre and grade level of a writing task, there are commonalities between each rubric that allow their traits to be reasonably combined for modeling. All our rubrics, for example, include LANGUAGE (and style) and ORGANIZATION traits, though their expectations vary by genre and grade level. The generic MAIN IDEA trait corresponds to "Claim" and "Clarity and Focus" traits, and SUPPORT corresponds to "Support and Development" as well as "Analysis and Evidence." Rubrics and prompts were developed for validity, and essays were rigorously hand-scored by independent raters in the same manner as described in West-Smith et al. (2018b).

For each generic trait, the training set was sampled down from over 50,000 available essays, responding to 95 writing prompts. Essays from 77 prompts were selected for the training set, and another 18 were held out for evaluation. Within each split, essays were sampled to minimize imbalance between essay score, genre, grade level, In the un-sampled data, longer essays tend to be strongly correlated with essay score, risking overfitting to this surface feature. Similarly, among the subset of data where school district data was available, districts with predominantly Black enrollment were

under-represented among essays with a score of ”4” across all traits. To counteract these potential biases, the available data was binned by length and district demographic information for each score, genre, and grade level, and essays were under-sampled from the largest bins. In addition to these balanced essays, about 800 “off topic” essays representing nonsense language or non-academic writing were included in the dataset, with a score of zero.

5.4.2 Essay Scoring Criteria

Essays were scored separately by human scorers based on four distinct traits, namely *Organization*, *Main Idea*, *Support*, and *Language*. The criteria are described below:

Organization: The overall structure and coherence of an essay. A well-organized essay presents ideas in a logical and easy-to-follow manner, with clear transitions between paragraphs and a clear introduction and conclusion that frame the essay’s argument.

Main Idea: The presence of a central thesis or argument in an essay. A strong essay will have a clear, focused main idea that is supported throughout the essay by relevant evidence and analysis.

Support: The presence of evidence and examples to back up the main idea of an essay. Effective support is relevant, convincing, and logically connected to the main idea, and may include quotations, statistics, personal anecdotes, or other types of evidence.

Language: The overall quality of writing in an essay, including grammar, vocabulary, sentence structure, and style. Strong language use enhances clarity and precision, engages the reader, and conveys the author’s tone and purpose.

5.4.3 Models

Longformers are a transformer-based neural network architecture that have gained prominence in various NLP tasks (Beltagy et al., 2020). In the context of AES, each generic trait’s model is a Longformer with a single-output regression head, fine-tuned on the trait’s balanced dataset: For the remainder of this paper, the model fine-tuned on a given trait will be referred to as “the TRAIT model” (e.g. the ORGANIZATION model) for simplicity.

Although ordinal scores from 0 to 4 were used for sampling and evaluation, the training data labels were continuous, averaged from rater scores. Essays were prefixed with text representing their genre (e.g., ”Historical Analysis”) and prompt’s grade range (e.g., ”grades 10-12”) before tokenization, but no other context for the writing task (e.g., the prompt’s title, instructions, or source material) was included. In addition to Longformer’s sliding attention window of 512 tokens, the first and last 32 tokens received global attention.

Scores were rounded back to integers between 0 and 4, before evaluation. On the holdout prompts, overall Quadratic Weighted Kappa (QWK) ranged from 0.784 for MAIN IDEA to 0.839 for LANGUAGE, while correlation with word count remained acceptably low: 0.441 for LANGUAGE up to 0.550 for SUPPORT.

The activations of the Longformer model were saved for each instance in the analysis set at the “classify” token to create a matrix of activations for the functional component

extraction.

Model *A*	Model *B*	# Essays	Extracted Features	# Independent Feature Groups	# Aligned IFG
ORGANIZATION	MAIN IDEA	407	148	114	24
ORGANIZATION	LANGUAGE	275	118	86	39
ORGANIZATION	SUPPORT	144	90	63	37
LANGUAGE	MAIN IDEA	341	129	95	26
LANGUAGE	SUPPORT	72	67	38	23
SUPPORT	MAIN IDEA	260	127	94	27

Table 8: Comparing analysis dataset size and numbers of extracted features for each of the model comparisons, identified by the Model *A* and Model *B* columns.

5.4.4 Features

The features employed in this analysis encompass statistical properties of the essays, tree features generated from Rhetorical Structure Theory (RST) parse trees of the essays, essay prompt and genre, a combination of algorithmically derived and human-defined style-based word lists, and certain school-level demographic features. A description of each feature type is provided below:

Statistical Features: While statistical features such as *essay word count* are often good indicators of essay score, they are not intrinsically valuable to the different traits that our models are scoring. We thus want to see lower alignment with these features to indicate that the model is not overly relying on rudimentary shortcuts scoring an essay. We also include *average word length*, *essay paragraph count*, *essay sentence count*, *average sentence length*, and the *standard deviation of the sentence length* for completeness.

RST Tree Features: These features were integrated to capture the rhetorical structure of the text, such as the hierarchy of principal and subordinate clauses, the logical and temporal relations between propositions, and the coherence of the argument. These concepts have a high validity for scoring essays (Jiang et al., 2019), especially for ORGANIZATION, so high alignment between functional groups would be expected. To generate RST trees for each essay, we utilize a pretrained RST parser specifically fine-tuned for student writing (Fiacco et al., 2022). We include the presence of an RST relation as a feature as well as relation triplets $(\text{REL}_{parent}, \text{REL}_{child_1}, \text{REL}_{child_2})$ as tree-equivalent n-gram-like features.

Essay Prompt and Genre: Categorical representations of the essay prompt and genre were employed as features to examine if components of the AES model were preferentially activated based on the content or topic of the essay, a low validity feature.

Algorithmically Generated Word List Features: We calculate the frequency of usage of words within algorithmically derived sets of words in the essays as a group of features to probe the AES model's consideration for stylistic language. To generate these word lists, we obtain Brown clusters (Brown et al., 1992) from essays. We generate separate Brown clusters for each prompt in our dataset and subsequently derive final word lists

based on the overlaps of those clusters. This approach emphasizes common stylistic features as opposed to content-based clusters.

Human Generated Word List Features: In addition to the algorithmically defined word lists, we devise our own word lists that may reflect how the AES model scores essays. We created word lists for the following categories: simple words, informal language, formal language, literary terms, transition words, and words unique to African American Vernacular English (AAVE).

Demographic Features: We used the percent to participants in the National School Lunch Program (NSLP) at a school as a weak proxy for the economic status of a student. Also as weak proxies for economic status of essay authors, we include the school level features of *number of students* and *student teacher ratio*. Furthermore, we use a school level distribution of ethnicity statistics as a weak proxy for the ethnic information of an essay's author. These features were employed to investigate the model's perception of any relationship between the writer's background and the quality, content, and style of the essay, in order to gain insight of the equity of the AES model.

		Functional Group Extraction			**Important Functional Group Alignment**			
Model *A*	**Model *B***	**# Comp. *A***	**# Comp. *B***	**# FG**	**# Aligned FG**	**# A Only**	**# B Only**	**# Mixed**
ORGANIZATION	MAIN IDEA	119	55	125	22	12	0	10
ORGANIZATION	LANGUAGE	96	66	110	29	11	0	18
ORGANIZATION	SUPPORT	66	36	68	22	9	1	12
LANGUAGE	MAIN IDEA	78	55	93	23	8	3	12
LANGUAGE	SUPPORT	34	28	38	13	2	2	9
SUPPORT	MAIN IDEA	45	49	64	25	2	2	21

Table 9: Comparing number of functional groups extracted for each model comparison and presenting the number of functional groups that were both deemed important (Section 5.3.2) and sufficiently aligned with at least one feature group. Also specified is the number of functional groups that are unique to a particular model and the number that are shared between the models of given a comparison pair.

5.4.5 Analysis Settings

To choose the number of components for ICA, a PCA was performed to determine how many components explained 95% of the variance of the activation (or 99% of the variance for the features) to be used as the number of components of the ICA. To determine that a functional group was important, it needed to have an absolute value of Pearson's r value of greater than 0.2. This threshold was also used to determine if a functional group should be considered aligned with a feature group.

5.4.6 Results

In this section, we present aggregate statistics for each model comparison when it comes to computing features and independent feature groups (Table 8), extracting functional

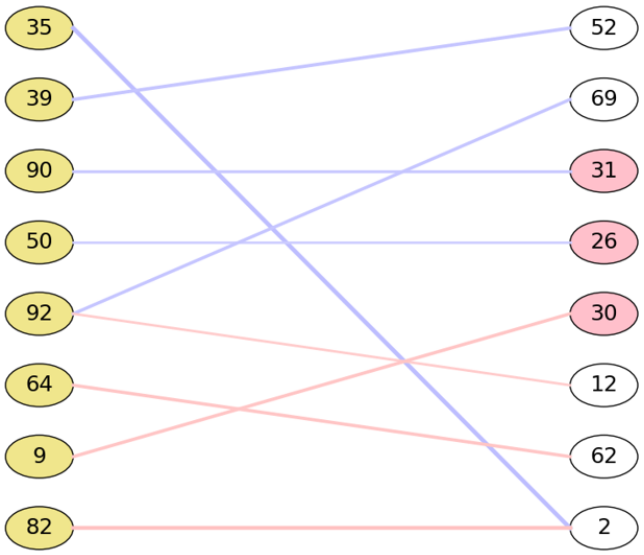


Figure 16: Alignment diagram for functional groups (*left*) that are specific to the LANGUAGE model with their alignment to feature groups (*right*). Only functional groups and feature groups are shown if they have a positive correlation greater than 0.25 *(blue edges*) or a negative correlation less than 0.25 (*red edge*s). The numbers correspond to the IDs of the functional group or feature group that the node represents (*see* Table 10).

groups and aligning important functional groups (Table 9), and lastly, we provide examples taken from the model comparison between the LANGUAGE model and the MAIN IDEA model. Due to length constraints, we present detailed examples of this comparison only. Similar figures and correlation statistics can can be found on Github[2].

Independent Feature Groups: Since each trained model held out a different set of prompts from its training set, common prompts between analysis sets needed to be identified, and thus the number of features extracted and the resulting independent feature groups vary between model comparisons. Computing the independent feature groups for each model comparison (Table 8) yielded between 70% and 77% of the original extracted features for all comparisons, except LANGUAGE V SUPPORT, which only yielded 57% as many independent feature groups compared to original features. Despite high variability in the number of independent feature groups identified during the process, a much more narrow range of independent feature groups was aligned during the analysis. The types of feature groups that were aligned varied considerably between different comparisons.

Functional Component Groups: The initial extraction of functional components for each model elicited numbers of functional components between 28 and 119. Table 8

[2] https://github.com/jfiacco/aes_neural_functional_groups/tree/main/supplementary_results

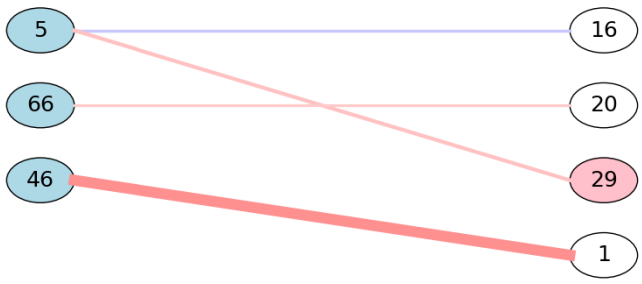


Figure 17: Alignment diagram for functional groups (*left*) that are specific to the MAIN IDEA model with their alignment to feature groups (*right*). Only functional groups and feature groups are shown if they have a positive correlation greater than 0.25 *(blue edges*) or a negative correlation less than -0.25 (*red edge*s). The numbers correspond to the IDs of the functional group or feature group that the node represents (*see* Table 10).

and 9 show that for a given model, fewer functional components will be extracted given a fewer instances in the analysis dataset. Despite this noise, a clear pattern emerges where the ORGANIZATION model has the most functional components, followed by the LANGUAGE model. The MAIN IDEA model has fewer functional components, with the SUPPORT model having the fewest.

When performing the dimensionality reduction to compute the functional groups, there is a consistent reduction to approximately 61-71% of the combined total functional components.

Important Functional Groups: Despite the variance in the number of feature groups and functional groups extracted per comparison, there is a remarkably consistent number of important functional groups that have at least one sufficient alignment to a feature group (Table 9). With the exception of the LANGUAGE V SUPPORT comparison, all other comparisons had between 21 and 29 aligned functional groups.

As a visual aid for the important functional groups, see the left sides of Figures 17 and 18. Each Figure is derived from the functional groups and feature groups of the LANGUAGE V MAIN IDEA comparison. The numbers on each node are the identifiers of a given functional group, a subset of which are represented in Table 10.

Alignment of Functional Groups: The entirety of findings from the alignments for all of the comparisons would be too numerous to present in a conference paper format. However, we will present the major trends we found in our analysis. The first main trend is that all models had functional groups that we correlated with the statistical features of the essay. Furthermore, by computing the correlations between the individual features within that type, it was determined that *number of paragraphs* is likely the most salient contributor.

The second set of trends is presented in Table 11, where the percent of the total aligned feature groups per model was computed. This revealed that the ORGANIZATION model had considerably more aligned RST-based features than the other models, while the MAIN IDEA model had the least proportion. The LANGUAGE model had the most aligned word list features, which is the combination of the algorithmically and

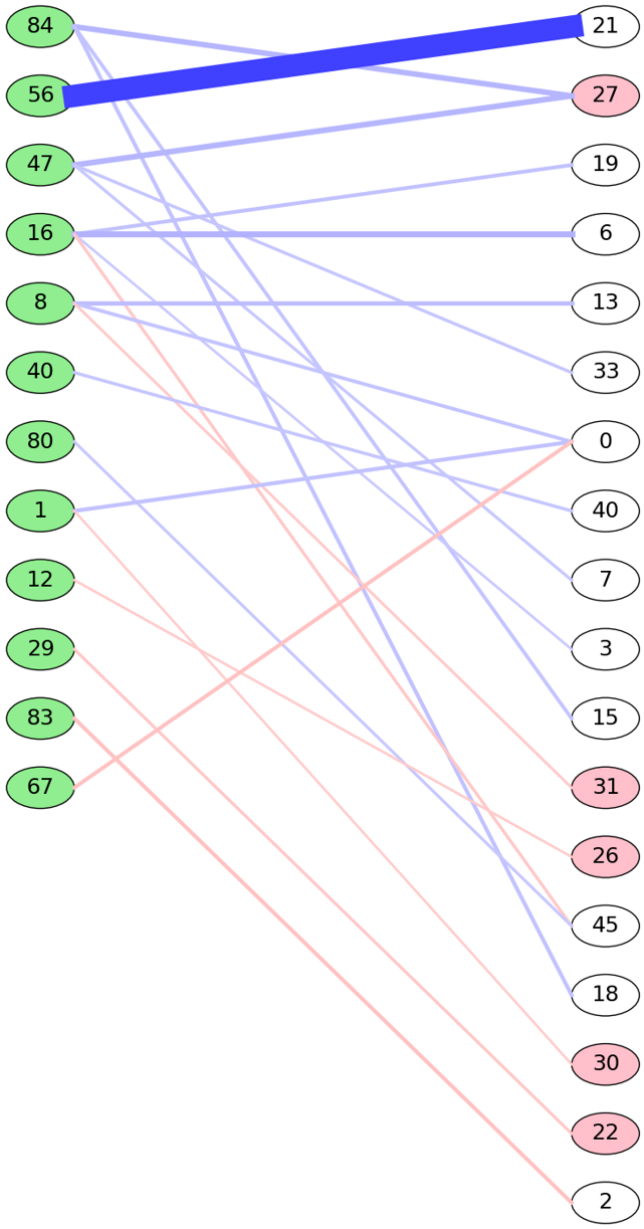


Figure 18: Alignment diagram for functional groups (*left*) that are common to both the LANGUAGE and MAIN IDEA models with their alignment to feature groups (*right*). Only functional groups and feature groups are shown if they have a positive correlation greater than 0.25 *(blue edges)* or a negative correlation less than -0.25 (*red edges*). The numbers correspond to the IDs of the functional group or feature group that the node represents (*see* Table 10).

Functional Group 46
Diff:LANGUAGEVSMAININDEA $r = -0.39 (p < 0.001)$
Independent Feature Group 1 $r = -0.43 (p < 0.001)$
ModelErrors:MAININDEA(+), ModelPairDifference(+), ModelErrors:LANGUAGE(-)
Functional Group 56
Predictions:MAININDEA $r = -0.13 (p < 0.05)$
Independent Feature Group 21 $r = 0.75 (p < 0.001)$
EssayStats:STDDEVSENTENCELENGTH(+), EssayStats:NUMSENTENCES(+), EssayStats: MEANWORDLENGTH(+), EssayStats:NUMWORDS (-), EssayStats:NUMPARAGRAPHS(-), EssayStats: MEANSENTENCELENGTH(-)
Functional Group 92
Predictions:LANGUAGE $r = -0.13 (p < 0.05)$
Independent Feature Group 12 $r = -0.20 (p < 0.001)$
WordCluster:PRIORITIES(+), WordCluster:POPULATIONCOMPARISION(+), WordCluster:EFFICIENCY(+), WordCluster:TEENVALUES(-), WordCluster:STORYTELLING(-), WordCluster:SCHOOL (-), WordCluster:PARENTALDECISIONS(-), WordCluster:INFORMAL(-), WordCluster:HISTORICALCONFLICT(-)
Independent Feature Group 69 $r = 0.22 (p < 0.001)$
RST:NN—CONTRAST(+), RST:SN—EVALUATION(NS—ELABORATION, LEAF)(+), RST:SN—BACKGROUND(LEAF, NS—ELABORATION)(+), RST:NS—EVIDENCE(LEAF, NN—CONJUNCTION)(+), RST:NN—JOINT(NN—CONJUNCTION, NN—JOINT)(+), RST:NN—CONTRAST(LEAF, LEAF)(+), RST:NN—CONJUNCTION(NS—ELABORATION, NN—CONJUNCTION)(+), RST:SN—EVALUATION(NN—CONJUNCTION, LEAF)(-), RST:NN—CONJUNCTION(LEAF, LEAF)(-)

Table 10: Selected examples of correlated functional group/feature groups. Pearson's R values for relevant importance metric (model difference, model predictions) and feature group alignment are presented with p-values.

human-created word list features. For the last percentage, we combine the prompt and demographic features and find that the SUPPORT model tended to align with fewer of these types of features. The reason for combining the demographic and prompt features is discussed in Section 5.4.7.

Qualitative Analysis: While the method that we presented can quickly advance one's

Model	%RST	%Word List	%Demo. & Prompt
ORGANIZATION	**41**	13	21
LANGUAGE	30	**26**	19
SUPPORT	36	19	13
MAIN IDEA	23	21	**23**

Table 11: % of aligned feature groups for a given model by feature type.

understanding of a model from the black-box neural network to aligned feature groups directly, understanding what function a feature group represents can be more difficult. It is thus necessary to resolve what a feature group represents to form a strong statement on what the model is doing. For instance, we found it concerning that so many of the models were connected with feature groups that contained demographic features (colored red in Figures 17 and 18). However, a qualitative look at the datasets for which prompts were included, we found that the distribution of prompts over the different schools, when controlling for essay length, were such that certain schools (with their demographic features) were the only source of certain prompts. It, therefore, becomes likely that many of these feature groups are more topic-based rather than the potentially more problematic demographic-based. This interpretation was reinforced by many of the feature groups with demographic information also including prompts (e.g. *"Independent Feature Group 29"* from Table 10) and by examining essays that present those feature groups.

5.4.7 Discussion

The results presented in the preceding section demonstrate the efficacy of the proposed method in extracting salient feature groups and functional groups from the neural models, particularly when applied to the dataset under consideration. The true potential of this method, however, lies in its capacity to be broadly applied to any neural AES system, thereby facilitating a deeper understanding of the models and the underlying processes they employ. In the following discussion, we will delve further into the results, emphasizing the prominent trends observed in the alignment of functional groups and their correlation with essay features, as well as the implications of these findings for enhancing the interpretability and transparency of neural AES systems.

Functional Component and Feature Groups: The proposed method successfully extracted meaningful functional groups from the analyzed neural models. Notably, the LANGUAGE V SUPPORT comparison emerged as an outlier in several of our analyses. This discrepancy is likely attributable to the considerably fewer essays shared by both models' analysis sets, which may result in a noisier analysis and expose a limitation of the method. As the size of the analysis increases, one would expect the extraction of feature groups and function groups to approach their ideal independence characteristics. Despite this limitation, the method managed to condense the analysis space from thousands of activations to fewer than 125 while still accounting for over 90% of the model's variance.

Interestingly, the ORGANIZATION model exhibited the highest number of functional groups. This observation suggests that capturing the ORGANIZATION trait is a more intricate process, necessitating the learning of additional features. This notion is further corroborated by the comparisons between ORGANIZATION and other models; models which displayed very few, if any, functional groups exclusively present in the non-organization models.

Alignment of Important Functional Groups: In line with our expectations, the ORGANIZATION model demonstrated the greatest alignment with the RST tree features, while the LANGUAGE model displayed the most significant alignment with the word list features. It was postulated that ORGANIZATION would necessitate the model to possess knowledge of how ideas within essays are structured in relation to each other, a type of knowledge encoded by rhetorical structure theory. Although the RST parse trees recovered from the parser are considerably noisy (RST parsing of student essay data has been shown to be markedly more challenging than standard datasets (Fiacco et al., 2022)), the signal remained significant. Furthermore, we anticipated that the LANGUAGE model would have a greater reliance on word choice, a concept mirrored by the word list-based feature groups.

Contrary to our expectations, the MAIN IDEA model exhibited the highest number of prompt-based feature groups. Our most plausible explanation for this observation is that certain prompts might have clearer expectations for thesis statements than others, a notion generally supported by a qualitative examination of the essays from prompts that score higher on MAIN IDEA.

5.4.8 Conclusion

The neural network interpretation technique presented in this paper demonstrates significant promise in learning the implicit rubrics of neural automated essay scoring models. By effectively mapping the intricate relationships between feature groups and the functional groups of the underlying scoring mechanism, the technique provides a step towards an understanding of the factors contributing to a transformer's evaluation of essay quality. This enhanced understanding enables researchers and educators to not only identify potential biases in scoring models, but also to refine their models to ensure a more reliable and fair assessment of student performance.

The code for this method will be released and incorporated into an analysis tool for application to neural models not limited to the ones examined in this work with the goal to pave the way for the development of more transparency in neural AES models. These advancements can contribute to the overarching goal of promoting ethical and responsible AI in education by facilitating the examination and comprehension of complex neural models.

6 Applying Neural Pathways to Real-world Tasks and Datasets

In the next three chapters, we leverage the neural pathways approach from Fiacco et al. (2019a) to guide advances in three natural language processing tasks: automatic transactivity detection (Joshi and Rosé, 2007), rhetorical structure parsing (Mann and Thompson, 1988), and AI writing detection (Solaiman et al., 2019; Yan et al., 2023). Each of these tasks provide an important feature for evaluating how the neural pathways approach can be applied to tasks and dataset beyond the synthetic tasks and data that we have previously covered.

The automatic transactivity detection task makes use of transfer learning to leverage model learning done on a large, well studied dataset on a much smaller dataset. The similarities between the pretraining task and the automatic transactivity detection task allow the model to adapt the functions that it learned to the new domain. This provides a good test bed for the neural pathways approach as we can probe the commonalities and differences between the models before and after the transfer learning has been applied. We can thus make model design decisions that can make the most out of the transferred learning while covering gaps of previous iterations of the models.

In a similar vein, we can use the neural pathways approach to understand models designed to perform rhetorical structure parsing such that we can choose improvements that directly address existing models' shortcomings. This is particularly useful for rhetorical structure parsing for student writing where the goal is not only to study how students construct essays, but also to provide automatic feedback. Providing feedback requires knowing where disjoint structures occur, but can be augmented should one be able to know *why* the mistake was made.

Lastly, we chose to explore the identification of pathways within AI writing detection models as our final task, recognizing its significance in the current landscape. While our focus is primarily on understanding what the neural model is learning, we framed our investigation such that its methods and insights may be applicable and interesting bias and fairness in neural networks community. However, it's crucial to mention that our study does not directly evaluate the bias or fairness of the these models.

In the remainder of this chapter, we provide the foundational knowledge for each of these tasks along with the reasons why they are instrumental in demonstrating the capabilities of the neural pathways approach. The following three chapters go into much greater detail about the experiments performed with their respective tasks.

6.1 Transferable Pathways for Automatic Transactivity Detection

The concept that transfer learning is able to carry over knowledge learned from one task to another provides an excellent test-bed for examining neural pathways. It stands to reason that if knowledge is being transferred from a model trained on a pretraining task to a post training task there should be overlapping neural pathways insofar as there is overlapping knowledge requirements for the task. We hypothesize that this overlap will be detectable and that knowing what information transfers can lead to better decisions for choosing pretraining tasks and architectures that can carry forward the

desired information.

6.1.1 Transactivity

The concept of Transactivity originally grows out of a Piagetian theory of learning where this conversational behavior is said to reflect a balance of perceived power within an interaction (Berkowitz and Gibbs, 1983; De Lisi and Golbeck, 1999). It is a property of discourse in an educational context that is associated with interactions that are beneficial for learning (Azmitia and Montgomery, 1993), and thus it has been of great interest within the learning sciences in the area of discussion based learning.

Transactive contributions demonstrate consideration of the earlier expressed ideas. Thus, it makes sense that recent work has demonstrated that automated models for Transactivity detection can be used as a foundation for highly effective assignment of students to project teams in MOOCs by estimating the collaborative potential of pairs of participants based on the exchange of Transactive contributions (Wen et al., in press). Even before this recent work, there was much interest in automated detection of transactivity in educational applications (Joshi and Rosé, 2007; Rosé et al., 2008; McLaren et al., 2007; Ai et al., 2010; Gweon et al., 2013). However, where there are reported successes, past work has failed to produce models that generalize well to new domains (Mu et al., 2012), which we address in this work.

A Transactive contribution to a discourse must meet two requirements(Gweon et al., 2013). First it must display reasoning, in other words revealing how a speaker thinks something works, which can be accomplished through an expressed evaluation, comparison, or reference to a causal mechanism. For example, ”Use of coal increases pollution” displays a causal mechanism and ”Use of wind power may not be reliable throughout the year” expresses an evaluation. But something like ”I prefer coal power” does not express reasoning. A Piagetian perspective on learning would suggest that students display their reasoning more when they are in a safe environment where they feel their ideas are valued and respected (De Lisi and Golbeck, 1999; Azmitia and Montgomery, 1993).

The second requirement for a Transactive contribution is that it references and idea expressed earlier in a discourse. Students reference the ideas of another student when they are listening to that student. It is a sign that the student takes the other student seriously enough to consider their ideas and how their respective ideas relate to one another (De Lisi and Golbeck, 1999; Azmitia and Montgomery, 1993). If earlier a speaker said, ”Wind is my choice because it is sustainable”, a Transactive reply would be ”Wind is sustainable, but it fails to be reliable throughout the year”. On the other hand, ”Use of coal is cheap and reliable” would not be a Transactive reply. In one case, the speaker shows consideration of another student’s ideas, while in the second case we do not see this consideration. From our technical perspective, an important aspect of the operationalization that we leverage in the work reported in this chapter is the *idea relatedness* of the Transactive contribution and the earlier contribution it refers transactively to.

6.1.2 Transfer Learning

Transfer learning, the process of transitioning learning from one task to another, has long been studied in the context of reinforcement learning and robotics (Taylor and Stone, 2009), but has more recently began have strong influences in other domains (Pan and Yang, 2010). In natural language processing, transfer learning has been shown to support a variety of basic tasks including chunking, named entity recognition, and semantic role labeling (Collobert et al., 2011; Peng et al., 2017; Peters et al., 2017). More recently, deep learning models in the paradigm of sequence-to-sequence modeling have been shown to be able to leverage multi-task learning (Luong et al., 2015; Yang et al., 2017). Many of these multi-task transfers have both the initial task and the transfer task made use of very large datasets. Here we approach a transfer task in two domains where there is not a very large corpus in either domain.

6.1.3 Entailment as a Pretraining Task

In our work, we use the Entailment task as the more fundamental task that forms a foundation for Transactivity detection. The Entailment task, specifically, comprises of deciding whether the concepts presented in one text can be determined to be true given some context or premise given in a different text (Condoravdi et al., 2003). For example, if an object is a shoe, then we can assume it was made to be worn on the foot. Therefore, *shoe* entails *made to be worn on the foot*. Because the task requires inferring abstract connections between ideas within two snippets of text, we considered it a good candidate for transferring learning to more specific applied discourse tasks where it is important to identify forms of *idea relatedness*, such as Transactivity.

One text entails another text if there is a conceptual link via an inference that associates those two texts. Similarly, a Transactive contribution to a discussion is one that displays reasoning and uses that reasoning display to evaluate, extend, transform, or refer substantively to an assertion made earlier in the discourse. The simple way of thinking about what constitutes a reasoning display is that it has to communicate an expression of some causal mechanism or express an evaluation or comparison. Transactive contributions are reasoning displays where the contribution either explicitly refers linguistically in some way to a prior statement, such as through the use of a pronoun or deictic expression, or implicitly by referring to a related idea. Thus, both Transactivity detection and entailment detection share the notion of concepts linked via inference.

What makes detection of Transactivity challenging in a domain general way is identification of the relevant conceptual links between ideas related by inference. Instead, using state-of-the-art approaches to Transactivity detection, such as linear Support Vector Machine models with n-gram features (Rosé et al., 2008) is that rather than learn the general task of identifying idea relatedness, the models tend to learn which concepts in the training domain are related to one another, and to identify them from their associated words. Thus, the learned associations are not useful anymore in a different domains since the set of related concepts that are relevant in the new domain will be different. Our work is based on the premise that networks trained to perform the Entailment task may need to learn internal text encoding representations that enable measurement of

"closeness by inference" rather than "closeness in meaning", in other words identification of abstract connections between expressed ideas. Since Transactive contributions build on or evaluate assertions made earlier in a discourse, the sub-problem of detecting idea relatedness is a foundational task. Note that the concept of *idea relatedness* used here as in the operationalization of Transactivity goes beyond text similarity. The idea is not that the two concepts are rephrases of one another, but that they are related to one another through some inference.

6.2 Discourse Parsing

Discourse parsing, more specifically discourse parsing with Rhetorical Structure Theory (RST), allows us to examine models that operate over a well defined hierarchical structure. Furthermore, unlike syntactic parsing, the elementary unit of the tree structures in RST is not only a word, but a sentence that contains potentially complex meaning on its own. This allows us to use neural pathways to examine both the neural parser state at various parsing decisions, but also at the representation for the units of discourse that it develops to facilitate the parsing.

6.2.1 Rhetorical Structure Theory

Rhetorical Structure Theory decomposes a document into basic units of analysis called elementary discourse units (EDU) that can be combined through rhetorical relations between units into larger composite units (Mann and Thompson, 1988). Thus, the rhetorical relations combine to build a hierarchical tree structure that represents the overall structure of the document (Figure 19a). Each relation has one (mononuclear) or more (multinuclear) nuclei where a nucleus is an essential span which, if deleted, would leave the remaining text incoherent. Mononuclear relations have satellites that are related to the nucleus by means of a rhetorical relation. They play a supporting role, and are therefore not necessary for coherence of the document. Each node of the tree represents a relation tuple $\langle S, N, R \rangle$ where S is the span, N is the direction of nuclearity, and R is the relation label. This is more readily seen in Figure 19b which depicts an alternate representation of the RST tree structure.

RST has a long history (Mann and Thompson, 1988), and its original formulation continues to be treated as authoritative. However, for some types of writing, especially student writing, additional and combined relations have been proposed in order to bring the set of used relations in line with the writing practices that are applicable to the corpus (Jiang et al., 2019).

6.2.2 Parsing Rhetorical Structures for Automatic Essay Feedback

Datasets for RST are time consuming to annotate and require high degrees of expertise to achieve reliability (Jiang et al., 2019). Available public corpora of data with RST annotations are small relative to the magnitude of annotated data available for training syntactic parsers. Considering both the dearth of annotated data and the challenges of decision making for discourse relations based on local context, it is not surprising that

RST parsing has remained a challenging task that has had only incremental improvements even since the deep learning revolution (Li et al., 2014; Ji and Eisenstein, 2014; Li et al., 2016; Braud et al., 2017; Yu et al., 2018; Mabona et al., 2019).

6.3 AI Writing Detection

Large Language Models (LLMs) have gained unprecedented prominence in a diverse range of Natural Language Processing (NLP) tasks, from creative text generation to sophisticated question answering (Christiano et al., 2017; Elkins and Chun, 2020; Stiennon et al., 2020; Lu et al., 2022; Ouyang et al., 2022; Guo et al., 2023; Huang and Tan, 2023). Yet, as reliance on these LLMs burgeons, so does apprehension over the biases inherent in these models. The community acknowledges that pretrained models are heavily influenced by their training data, inherently absorbing biases (Bommasani et al., 2021; Talboy and Fuller, 2023; Venkit et al., 2023) and a quickly growing body of research is aimed at mitigating these effects (Saleiro et al., 2018; Barikeri et al., 2021; Lin et al., 2021; Lee et al., 2023; Ungless et al., 2022; Thakur et al., 2023). However, there is a lack of understanding about the specific patterns within the ubiquitous process of fine-tuning that give rise to these biases.

Furthermore, as the proficiency and ubiquity of LLMs advance, there emerges a pressing need for effective AI detection tools and datasets to train them. The transformative potential of LLMs has led to their expanding use in automated content generation and essay writing (Fuchs, 2023; Sharples, 2022; Yeadon et al., 2023). As a result, differentiating between human and AI-generated content has become a crucial research frontier (Solaiman et al., 2019; Yan et al., 2023).

Detection of AI-generated writing is rapidly becoming a more difficult task as improvements in text-generation make leaps and bounds of progress (Ippolito et al., 2020; Clark et al., 2021). The development of AI-writing detection techniques is thus a growing area of research with two primary approaches: deep learning and neural network based approaches (Solaiman et al., 2019; Adelani et al., 2020; Uchendu et al., 2020) which can work exceptionally well, though are inherently black-boxes leading to instances where the model can fail catastrophically for potentially unforeseen reasons (Solaiman et al., 2019); and statistical and feature-based approaches (Fröhling and Zubiaga, 2021; Gallé et al., 2021) which intend to overcome the black-box limitation of deep learning based approaches by using inherently more interpretable methods. This paper focuses on deep learning based AI-writing detection models as they are currently the most effective style of AI-detection system (Jawahar et al., 2020) and highly flexible in their trainability. This means they are highly relevant to the community and have a high capacity for future research to adapt to the findings of this work.

This area of research is quickly evolving, and the background presented in this chapter is only a brief snapshot of the work that is ongoing in the field.

6.4 Fairness in Neural Models

While the prominence of neural networks has exponentially grown across diverse domains, impacting even policy decisions, the imperative for these models to make fair predictions is undeniable. We interpret fairness as ensuring consistent expected

outcomes across varied populations, devoid of influences from sensitive attributes such as race, gender, or socio-economic status.

Our engagement in the AI Writing detection task has allowed us to explore themes connected to fairness in neural models. However, it is crucial to clarify that, within this scope, our intention is not to make direct contributions to the broader discourse on bias and fairness evaluation methodologies. Our objective is more nuanced, focused on the intersections between our research, that is understanding what functions a model uses to perform a task, and the identified gaps in fairness literature.

In interviews with machine learning practitioners, Holstein et al. (2019) found five general gaps in the research being done for fairness in machine learning. First, the literature focused on "de-biasing" when interviewees indicated that they would prefer more work towards curating datasets that will more naturally yield more fair models. Second, practitioners wanted to see resources, metrics, processes, and tools that can be domain-specific as the very definition of fairness can change between contexts and applications. Third, they found that many fairness auditing techniques required individual-level demographic information despite that type of information often being unavailable to the practitioners. Their penultimate finding was that there was a dearth of tools for fairness-focused debugging of machine learning models. Lastly, there was a great desire for prototyping tools for evaluating potential fairness issues prior to deployment.

These findings were further corroborated by a later interview study conducted by Law et al. (2020) on professional modelers at a large technology firm. Specifically, in addition to the general finding that the professionals expressed concern at how performance between demographic groups could be considerably different, there was expression of a great need for conducting fairness audits without individual-level access to demographic information that is either unobtainable, unreliable, or a cause for privacy concerns. The interviewees further commented on the difficulty of foreseeing bias within models and then a difficulty in formulating the caveats that apply to the models. The last theme of the responses centered around the difficulty in understanding the root causes of the biases in to both resolve the bias and to communicate what the bias is to stakeholders.

While it would be exceptionally difficult to attempt to address all of the issues raised in these studies, there are some overlaps between the two studies that the neural pathways approach can provide some insight. Specifically, we view that neural pathways can join and augment other fairness auditing frameworks to allow for debugging neural models with respect to fairness and illuminate the causes of the biases within a model.

In the remainder of this section we review other fairness auditing frameworks and provide background on the types of discrimination that serves as the target of our use of neural pathways.

6.4.1 Discrimination

There are many types of discrimination and there is a deep body of literature on discrimination theory (Marshall, 1974; Romei and Ruggieri, 2014; Willborn, 1984). However, for purposes of fairness in deep learning, it is possible to generalize the categories of discrimination to (1) *Explainable Discrimination* and the two subtypes of *unexplainable*

discrimination: (2) *direct discrimination* and (3) *indirect discrimination* (Mehrabi et al., 2021). In this chapter we focus on distinguishing between *explainable discrimination*, an apparent discriminatory outcome that can be explained by non-sensitive attributes (e.g. a difference in wages in a population of men and women that can be explained by a difference in hours worked between the two groups) (Kamiran and Žliobaitė, 2013), and *indirect discrimination*, a discriminatory outcome wherein an apparently neutral treatment becomes discriminatory based on interactions between sensitive attributes and treatment (e.g. the use of ZIP code to determine credit worthiness, as ZIP codes can correlate with ethnicity in residential areas) (Rice, 1996).

Treating a model as a black box for purposes of fairness makes it difficult if not impossible to distinguish between *explainable discrimination*, which is considered legal and acceptable (Kamiran and Žliobaitė, 2013), and *indirect discrimination*, which is considered a unjustified and illegal (Kamiran and Žliobaitė, 2013), as the difference between the two is dependent on what attributes a model is using to make its decision. We propose that our work on neural pathways can provide a way to differentiate between these forms of discrimination in existing trained neural models to provide researchers the information required to make fair modeling decisions.

7 Utilizing High Salience Neural Pathways to Improve Generalizability of RST Parsing on Student Writing

Neural models, in innumerable studies, exhibit a tendency to overfit to features that are unique to their training dataset, resulting in models that may not perform well on unseen, varied data. In more specific terms, overfitting to dataset-specific features refers to the model's inclination to learn intricate details, patterns, and noise from the training data that do not generalize well to new, unseen data, compromising the model's ability to perform the desired task in diverse real-world scenarios (Caruana et al., 2000). This is predominantly attributed to the high-dimensional and highly flexible nature of neural models, which, while enabling them to learn complex representations, also makes them susceptible to fitting to the peculiarities and idiosyncrasies of the training data while overwriting the heuristics that may be more beneficial to the general task.

Results from Caruana et al. (2000) paint a picture of overfitting in large models as a heterogeneous process that affects various regions of the neural network differently, describing a landscape of regions within the neural model, some of which correspond to generalizazble features and others corresponds to dataset artifacts. This property is exploited in early-stopping during training of neural networks by freezing the network when the neural model exhibits the maximum generalizable behavior. Through the use of neural pathways, we seek to exploit the generalizable regions of a neural model further by identifying which components of a model are aligned with the features essential for performing the task, freezing those components, then performing another round of learning to encourage the model to learn another set of generalizable features.

In this chapter, we demonstrate the ability of neural pathways to isolate the clusters of neurons that are responsible for the more general functions of a task, and inject them into a secondary model to improve the generalizability of the model. Specifically, our focus is on a neural transition parser designed for the task of discourse parsing, which is a complex yet important task that has applications in several NLP areas including text categorization, authorship attribution, and automated essay feedback (Feng and Hirst, 2014b; Ji and Smith, 2017; Jiang et al., 2019). We use the Rhetorical Structure Theory (RST) to represent discourse structures (Mann and Thompson, 1988), a widely recognized theory in the field of Computational Linguistics.

Our approach to improving model design is twofold. First, we perform a pathways analysis on a baseline RST parser to understand what kinds of information the model is using for its decision-making and assess how effectively it is doing so. Based on this analysis, we then augment the model to address its identified weaknesses. Second, we extract general structures from a pre-trained transition parser and construct a new version that uses these general structures for better adaptability to different datasets. We apply these techniques in the context of RST parsing for the purpose of enhancing Automatic Essay Feedback mechanisms.

7.1 Parsing Rhetorical Structures with Neural Models

For neural architectures applied to RST, neural transition based parsers have been making headway (Yu et al., 2018; Mabona et al., 2019), however, at their core, transition

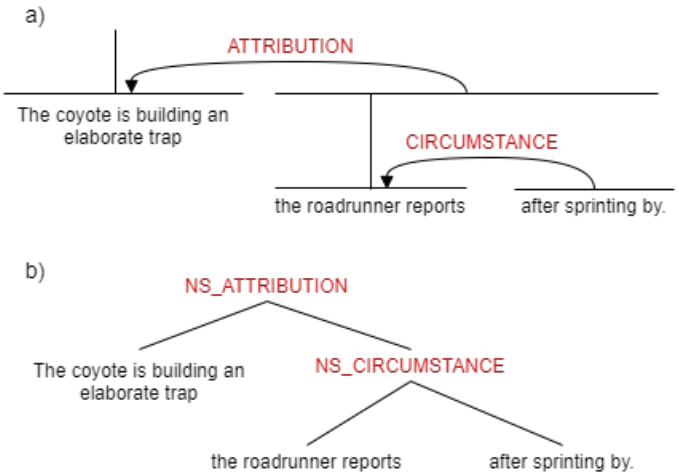


Figure 19: Example RST tree fragment with nuclearity and relations. a) The traditional depiction of an RST tree structure. b) The RST tree form corresponding to the labeled attachment decisions of (a).

parsers make parsing decisions locally. While many of these use recurrent models to construct their stacks and buffers, in practice, recurrent models have been shown to primarily to use very near context (Khandelwal et al., 2018). This is a limitation for discourse parsing where knowledge about the document as a whole may provide essential context for judging relations.

7.1.1 RST Datasets

The English RST Discourse Treebank includes 385 articles from the Wall Street Journal (Carlson et al., 2003). These articles represent approximately 180,000 words of texts and cover a wide range of topics, such as finance and arts. These articles were created by professional writers, thus, it is the most frequently used data set for well-written, copy-edited sentences (Palmer et al., 2010).

The student writing corpus, which was provided by Turnitin, and annotated and made public by Jiang et al. (2019), includes 274 essays from students' responses to standards-aligned (Valencia and Wixson, 2001) formative writing tasks (West-Smith et al., 2018a). These tasks cover a range of genres, including literary analysis, argumentative, historical analysis, and informative writing. As an example of a writing task, students write an essay to the head of the school board to argue whether sports are more helpful or harmful to young people. These essays represent a diverse set of secondary classrooms across the United States, representing a broad range of writing skills and student backgrounds. We separate out 25 documents for a development set and reserve 28 documents as a test set.

Comparison of Datasets: In addition to errors related to grammar, spelling, and usage, common patterns in less proficient students' writing could make the modeling task on

our selected second corpus featuring student writing more challenging than with the relatively clean RST-DT dataset. Common practices in the corpus include (1) sentences lacking transition words, such as words to express contrasts (e.g., however), or transition words used inappropriately; (2) pronouns exhibiting reference ambiguity; (3) paragraphs where the topic sentence is not clearly indicated, or where there are multiple main ideas (and sometimes contradictory ideas) in one paragraph or across paragraphs; (4) sentences not presented in a logical progression. These areas of focus for developing writers are also highlighted in the literature (de Jong and Harper, 2005). Ambiguous and weakly structured essays may indicate an opportunity for automated feedback, but they also pose challenges for the parsing task.

The use of the JOINT relation captures some of the difference between RST-DT and the Turnitin essays. JOINT, as defined by Jiang et al. (2019), indicates a lack of rhetorical relations between nuclei. It indicates that there is no relation that could describe the connection between sentences. In newspaper articles, this lack of connection is very rare, however, in student essays the lack of coherent rhetorical relations is common because of the high variability in writing ability among authors.

7.2 Neural Transition Parsing Model

Transition parsers are common among state-of-the-art models for discourse parsing with RST in the past several years. Their power lies in their ability to make strong local decisions about the next action the parser must take given an embedding that, because of recurrent neural models, has the capacity to contain features from the whole document. However, recurrent neural networks often do not in practice retain sufficient context for long range dependencies (Bahdanau et al., 2014; Khandelwal et al., 2018). We address this by providing an additional embedding for the predicted most nuclear sentence of the document to provide a reference point for the parsing decisions. Furthermore, inspired by neural interpretation techniques, we further augment the model with a two stage parsing approach that allows the second stage of the model to learn from mistakes made by the first.

The model presented in this work is based on the parser presented in Yu et al. (2018). For the benefit of the reader this subsection provides an overview of that model, however, for a full mechanical description see their paper. Our augmentations of the model follow in Sections 7.3.1 and 7.3.2.

The model constructs a neural representation that is used to decide whether to make a SHIFT or REDUCE action analogous to those in a simple LR-parser (Knuth, 1965). Furthermore, the model maintains a neural analogue to a stack and buffer to track progress through the parse, which is illustrated in the unshaded regions of Figure 20.

EDU Embedding: Each sentence in the document is embedded using a BiLSTM over word embeddings for each word in the EDU. The final states of the forward and backward LSTMs are used as the EDU representation.

Dependency Parse Embedding: In addition to the embedding generated by the BiLSTM, an embedding of syntactic information was included (Braud et al., 2017; Mabona et al., 2019). The information was integrated via concatenating the produced arc embedding from the dependency parse obtained from the parser described in Dozat and

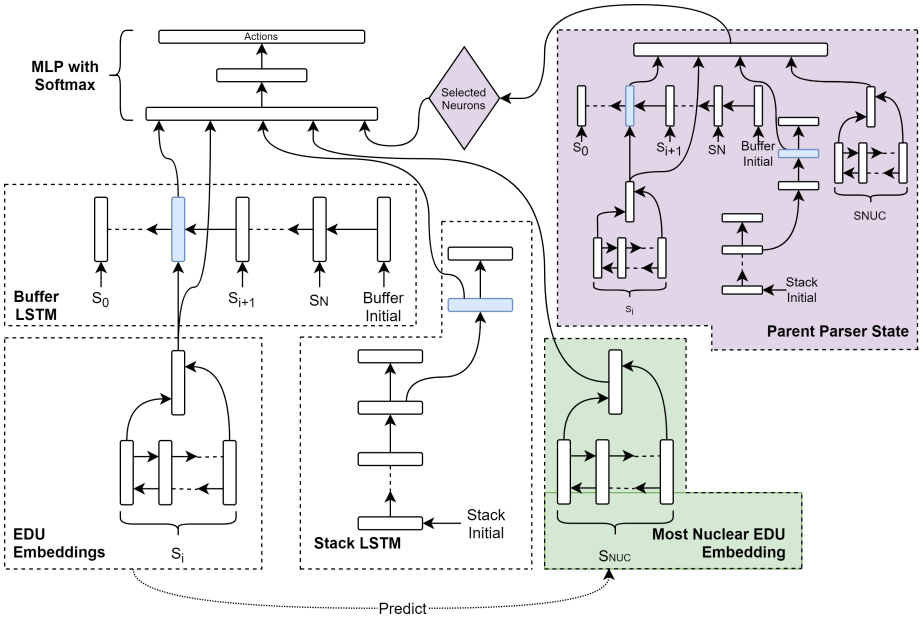


Figure 20: Diagram of neural transition parser model architecture for RST parsing augmented with our changes (shaded purple and green). The parent parser state (purple) has the same basic architecture as the rest of the diagram with the exception of having another parent parser state component. The dotted line from EDU Embedding to Most Nuclear EDU Embedding (green) indicates choice made by the model for which EDU to use.

Manning (2017) with the output from the BiLSTM above.

Buffer: The buffer is an LSTM that inputs each EDU embedding from the end of the document to the beginning. Each state is stored in memory such that it can be accessed sequentially as items are removed from the buffer. Each state of the buffer is therefore an aggregate representation of all of the EDUs from the current EDU to the end of the document.

Stack: The stack is a Stack LSTM (Swayamdipta et al., 2016). The stack state is updated via the result of an MLP given the two stacks states popped off the stack during a REDUCE action procedure. If an item is popped off the stack, the stack state is updated to the output state of the LSTM of the previous cell.

Action and Relation Prediction: At each time-step the parser either predicts a SHIFT action or one of the many REDUCE actions. Each REDUCE action has an associated relation label and predicting the correct REDUCE action amounts to choosing the correct relation for the current subtree. The prediction is made by a multi-layer perceptron (MLP) that is provided a concatenation of the EDU embedding, the current neural state of the buffer, the current neural state of the stack, and additional neural representations that will be described in depth in Sections 7.3.1 and 7.3.2. The input layer to the MLP will be referred to as the parser state at a given time. For each action, a deterministic procedure is executed in line with the transition parsing paradigm. In the case where there is only one possible action, the model is forced to use that action without choice.

7.3 Augmentations to the Baseline

We propose two improvements to the neural transition parser paradigm that can provide better performance and utilize limited data more efficiently:

1. By adding a co-task of predicting the most nuclear unit of the RST tree, we can increase the model's performance with the intuition that it may incentivize the model to maintain a broader document context that it can use for predicting individual tree spans and nuclearity.

2. By selectively introducing parser states from a previously trained parser into a new model during training, we can guide the training of the new model towards better performance on noisy datasets.

The first improvement builds on the general concept of multitask learning in NLP (Bingel and Søgaard, 2017; Peng et al., 2017) and the intuition that a topic-like sentence, as a common key component in many writing assessments and rubrics (Aull, 2015), may provide important contextual information to aid local parser decision-making. The second improvement builds on an intuition provided by a neural network interpretation technique that suggests a potential for neural component reuse.

We evaluate our model on both the standard English RST Discourse Treebank (RST-DT) (Carlson et al., 2003) and a more recently introduced RST dataset that comprises student essays (Jiang et al., 2019). The second dataset was created for use as development data for automatic essay feedback where the RST structure can signal places where the structure and flow of student writing might need improvement.

7.3.1 Most Nuclear EDU Embedding

To provide the model a reference for making parsing decisions for a given document, we include in the parser state an EDU embedding of the predicted most nuclear EDU. Formally, we consider the most nuclear EDU the leaf node of the RST tree that is reached when, starting at the root node, one follows the direction of nuclearity at each branch. For multinuclear nodes, we arbitrarily take the left branch. In Figure 19, the most nuclear EDU would be "The coyote is building an elaborate trap."

The most nuclear EDU S_{NUC} is selected by the model by choosing the EDU with the maximum score computed by an MLP given the EDU embedding and choosing the highest scoring sentence. This can be formalized as:

$$S_{NUC} = \arg\max_{\mathbf{s} \in S} W \cdot \mathbf{s}$$

Where S is a set of all of the sentence EDU computed by the neural transition parser.

The most nuclear EDU embedding is constructed via a BiLSTM in much the same manner as the EDU embeddings in the neural transition parser. This BiLSTM has its own set of learned parameters, though it uses shared word embeddings as those used for the EDU embeddings.

Because there is only one predicted most nuclear EDU for a document, the effective training samples for this embedding is equal to the number of documents in the training set rather than the number of EDUs. Because of this, it is necessary to restrict the size of the embedding to prevent overfitting. Furthermore, the error from the RST parsing task cannot backpropagate to W through the argmax so we include a separate error signal for predicting the correct most nuclear EDU. The most nuclear EDU of a document can be trivially obtained from the gold trees.

7.3.2 Parent Parser State

From prior work using neural pathways for the NER task, we found some evidence that neural models may be learning general heuristics and memorizing exceptions to those heuristics that increase performance on a given task. Assuming this is the case, we attempt to exploit this behavior to offload some of the complexity of learning the RST discourse parsing task into multiple phases of training. A fully trained parent model, which includes all of the features in Sections 7.2 and 7.3.1, is executed concurrently to the child model and a subset of the parser state of the parent model is concatenated with the parser state of the child model.

The parser state for the parent model is updated along with the child model using the action chosen by the child model, though with its own stack and buffer representations. This ensures that even if the parent and child models diverge in their predicted actions, the parser states are consistent. Maintaining this consistency is important for the neural transition parser as the representation of the stack can contain a representation of a larger segment of the document than just a single EDU.

Neuron Selection via Pathways: To improve the generalizability of the model on noisy data, we prune the parser state from the parent model to only use the dimensions of the state that correspond to the neurons that are part of the neural pathways that explain the

most variance of the model. The intuition for this pruning is that the groups of neurons that explain the largest amount of variance in the model will regularize the model via eliminating overfitted parameters.

These neurons are obtained by extracting the parser state for each training instance and constructing an activation matrix with the dimensions of the parser state as columns and the training instances as rows. A PCA is performed over the matrix, and the subset of resulting factors that cumulatively explain more than a tunable threshold of the variance are chosen as the subset of pathways of interest. For each selected factor, the factor loadings of each neuron are computed and the N neurons with the highest loadings are added to the set of neurons to be transferred. The value of N can be tuned by optimizing performance on a validation set.

7.3.3 Training

There are three phases to the training of the model: parent model training, neuron selection, and child model training. The procedure for training the parent and child models are identical except for the usage of the parent neurons as features for the child model. The neuron selection phase is only applicable for the noisier Turnitin data and is described in Section 7.3.2.

There are three objectives that are optimized using negative log likelihood loss during the model training. The first training objective (L_m) is predicting the most nuclear EDU at the document level (Section 7.3.1). The second objective (L_n), at the action level, is to predict the nuclearity of each relation given the parser state. This objective affects how the model composes the embeddings when combining via a REDUCE action. The final training objective, (L_a), is to choose the correct action given the parser state. We do not fine tune the embedding from the dependency parser during training. The third phase of training follows the same procedure as the first phase with selected neurons from the parent parser state included. The final loss for a document is described as:

$$L = \alpha_m L_m + \alpha_n \sum_A L_n + \alpha_a \sum_A L_a$$

where A is the set of all actions required for the parse and each α is a scaling factor that can be tuned for each loss.

For noisy datasets, an additional step is required for the training procedure; the neurons that will be used by the child model must be selected. This is performed by computing the neural pathways of the parent model using the parser state via PCA. The pathways that explain the most variance are chosen and the heaviest loaded neurons on those pathways are selected. During training, no gradient is passed back to the parent model so the neuron selection process need not be continuous nor differentiable. Training the child model thereby uses the parser state of the parent model as though it were a fixed input.

7.4 Experiments

7.4.1 Experimental Design

We provide three quantitative evaluations of our method: first, in order to compare our parser to previous RST parsers, we train and evaluate our parser on the English RST-DT corpus (Section 7.5.1). Second, we provide an ablation study of the added components of our model along with the model we used as a base (Section 7.5.2). The ablation study uses the same test set as the first experiment, so results are directly comparable. Lastly, we train another version of our model on the Turnitin dataset, which has a very different set of properties when compared to the RST-DT corpus. This last set of experiments is designed to test the ability of the model to handle noisier, less structured text (Section 7.5.3). The model is compared to the strongest baseline from the RST-DT corpus retrained on the Turnitin dataset.

Model	**S**	**N**	**R**	**F**
JI AND EISENSTEIN (2014)*	64.1	54.2	**46.8**	**46.3**
FENG AND HIRST (2014A)*	68.6	55.9	45.8	44.6
LI ET AL. (2016)*	64.5	54.0	38.1	36.6
BRAUD ET AL. (2016)*	59.5	47.2	34.7	34.3
BRAUD ET AL. (2017)*	62.7	54.5	45.5	45.1
MABONA ET AL. (2019)	67.1	57.4	45.5	45.0
ZHANG ET AL. (2020A)	67.2	55.5	45.3	44.3
OUR MODEL	**71.7**	**60.3**	44.5	44.3
-DEPENDENCY PARSE EMBEDDINGS	71.2	58.4	43.6	43.6
-PARENT PARSER STATE	70.2	57.2	43.0	42.9
-MOST NUCLEAR EDU EMBEDDINGS	68.4	57.2	42.7	42.4
TRANSITION PARSER ONLY	67.2	53.7	39.9	39.8

Table 12: RST-DT test set micro-averaged F1 scores for labeled attachment decisions for our model with varying components removed. Parsers from previous work are reported as they appear in their original publication, with the exception of those marked with an * where the reported results come from the Morey et al. (2017) replication study.

7.4.2 Evaluation Metrics

The evaluations of this work follow the setup described by Morey et al. (2017) and, for consistency, only compare to models that were included in that replication study or use the same evaluation method. The reason for this restriction is that it was found that RST Parseval, the previous standard evaluation metric, artificially raised scores and had been used inconsistently (Morey et al., 2017). Our models are therefore evaluated using micro-averaged F1 scores on labeled attachment decisions for the four standard metrics: span attachments (S), span attachments with nuclearity (N), span attachments with relations (R), and span attachments with both nuclearity and relation labels (F).

7.4.3 Implementation Details

The models were implemented using the DyNet neural network toolkit (Neubig et al., 2017b). Training was performed on a NVIDIA GTX 1080. Early stopping was performed based on the F1 scores of the model without an oracle on the development set, with a patience of 3. The ADAM optimizer (Kingma and Ba, 2014) is used for training with a learning rate of 0.001. Dropout (Srivastava et al., 2014) is used for regularization and a dropout of 0.3 is applied to each hidden layer. All tunable α hyperparameters were left at 1.

For the RST parsing models, word embeddings for both the parent and child models were randomly initialized with 128 dimensional vectors. Each LSTM in the parent model had 256 dimensions while in the child model, each LSTM had 512 dimensions. For neuron selection, the 16 neurons with the highest factor loadings from the PCA were chosen for each pathway that explained more than 1% of the model variance. The number of dimensions for the PCA was tuned to explain 90% of the variance in neuron activations.

The dependency parser was pretrained on Universal Dependencies v1 (Nivre et al., 2016) derived from the Penn Treebank 3 (Marcus et al., 1999) using version 3.9.2 of the Stanford Universal Dependency Converter. Word embeddings and label MLP dimensions were set to 64 while the recurrent layers and the arc MLP layers were set to 128. Choice of optimizer, dropout, and early stopping criteria were the same for the dependency parser pretraining.

Model	**S**	**N**	**R**	**F**
RST-DT				
JI AND EISENSTEIN (2014)*	64.1	54.2	**46.8**	**46.3**
OUR MODEL	**71.7**	**60.3**	44.5	44.3
OUR MODEL (W/ NEURON SELECTION)	70.6	59.7	44.4	44.3
Turnitin Corpus				
JI AND EISENSTEIN (2014)*	**56.1**	**33.4**	1.2	1.1
OUR MODEL	44.1	22.9	14.0	12.4
OUR MODEL (W/ NEURON SELECTION)	47.6	28.4	**18.0**	**17.0**

Table 13: Test set micro-averaged F1 scores for labeled attachment decisions for our model on the RST-DT corpus and the Turnitin dataset. The models were evaluated on each dataset both with and without pruning the parent parser state (W/ NEURON SELECTION).

7.5 Evaluation

7.5.1 Parsing Results

Table 12 shows the performance across parsers on the labeled attachments metrics for the RST-DT test set. We include reported metrics for several models beyond the best baseline in order to provide a comprehensive view of recent work in the field, including

other neural based models. The best version of our model gains a 4.5% increase in F1 score for the span metric (S) and a 7.9% increase in F1 score for combined span and nuclearity metric (N) in comparison with the Feng and Hirst (2014a) model, the next best model for those metrics. The increase was gained with a competitive, albeit 2.8% lower span and relation metric (R).

Furthermore, we achieve these results with only the dependency parser as external data. Pretrained embeddings of any kind were not required for either the dependency parser nor the final RST parser and were found to not contribute empirically. Using pretrained GloVe embeddings (Pennington et al., 2014) do not significantly improve the performance over random initialization.

7.5.2 Ablation

We evaluated the model with key components removed to evaluate the effects of each of those components on the final performance of the model. The components ablated were the dependency parser embedding, the most nuclear EDU embedding, and the parent parser state. These results are presented in the lower section of Table 12.

From the results we see that the largest contributor to our model's performance was the inclusion of the most nuclear EDU co-task without which, the parser does not outperform the previous state-of-the-art on any metric. The parent model's parser state as a feature for action and relation prediction had the next largest effect with the span and nuclearity metric (N) falling to the same level as when the most nuclear EDU embedding was not used. Lastly, the syntactic information carried in the dependency parser embedding contributed the least, but still had a significant effect on all metrics.

We also present the performance of the base model, our implementation of the neural transition parser from Yu et al. (2018) with the same settings as each of the other models from the ablation study. While it has competitive performance to prior work on the span only metric (S), all of the metrics are considerably lower than the final model. All ablation conditions were significantly different from the final model with $p < 0.05$.

7.5.3 Model Robustness with Neuron Selection

In addition to the evaluation on the standard RST-DT corpus, we evaluated our model on the Turnitin dataset to test the robustness of the model against noisy data. Table 13 shows a comparison of the model performance on both the RST-DT corpus and the Turnitin dataset. For each dataset, we include versions of the model that use neuron selection as described in Section 7.3.2 and without. Performing the neuron selection significantly ($p < 0.001$) increased performance on the Turnitin dataset with only a minimal reduction of performance on the RST-DT corpus.

7.6 Discussion

We presented two principal augmentations to neural transition parsers for RST that resulted in a 7.9% increase in span prediction and a 4.5% increase in nuclearity prediction. These improvements were made while remaining competitive on relation prediction, though no improvement was observed for that metric. Furthermore, we evaluated our

model on an alternate, noisier dataset. We found that on this dataset our model had more accurate relation predictions than past approaches from the inclusion of a neuron selection step between the training of parent and child models in a boosting-like neural ensemble enhancement.

The use of pathways to carry forward the most critical data from a parent model to a child model allowed the model to stay robust against a far noisier type of data as compared to the standard Wall Street Journal articles that the RST-DT dataset is comprised of. Combined with the ability to analyze the function of neural pathways, the ability to use the more general pathways to give a model a gerneralizable foundation can be a powerful tool for developing models for other small or noisy datasets.

8 Neural Pathways in Transfer Learning

The lack of understanding about knowledge learned by neural networks has traditionally constrained model selection across tasks by limiting the insights obtainable through error analysis. Often conceptualized as feature extractors, neural networks inherently develop specific pathways (i.e. features) that influence their predictive outputs. Error analysis of these models has primarily focused on the role of training data and architectural elements in dictating what features a neural network acquires. Researchers use their intuitions on their findings to make modifications to their models. This chapter aims to extend the capabilities of researchers in the field by providing a method to provide insights into the effects that a specific type of these selection decisions have on the task performance. Specifically, the task we opt to explore this method with is the transactivity detection task, which is defined later in this chapter and benefits strongly from transfer learning.

This chapter first establishes a foundational task that substantiates the feasibility of shared neural pathways across models trained for different objectives. Situated within the context of transfer learning and specifically exploring the transferability from Recognizing Textual Entailment tasks to Transactivity detection tasks, the chapter then examines the impact of altering the nature of the training data, particularly data requiring differing computational characteristics for transactivity detection, on the neural pathways that the model ultimately learns.

8.1 Neural Transfer Learning in Small Datasets: Transactivity Detection Task

Over the past decade, increasing interest in automated analysis of online discussion for learning, sometimes referred to as Discourse Analytics, has been featured in research on environments like Massive Open Online Courses (MOOCs). In particular, prior work in MOOCs has demonstrated that students can benefit from discussion encounters with other students (Ferschke et al., 2015). Much of this prior work has targeted short synchronous collaborative discussion assignments or informal and unstructured discussion in asynchronous discussion forums (Nelimarkka and Vihavainen, 2015). More recently the topic of supporting team based project learning in MOOCs has emerged (Wen et al., in press).

Prior work has demonstrated the value in automated analysis of discussion for enabling effective assignment of students to project teams (Wen et al., in press), for triggering dynamic support of group learning (Kumar et al., 2007), and for assessment of learning processes (McLaren et al., 2007; Dascalu et al., 2015). Though a plethora of frameworks for analysis of discussion for learning are in operation, many include a dimension for collaborative knowledge construction where a valued conversational behavior is one where students explicitly make their reasoning visible in a way that connects back to ideas and reasoning expressed earlier in the encounter (Hmelo-Silver, 2013). One popular and long standing such construct is that of Transactivity.

In the remainder of the section we describe our approach. We then present an evaluation that demonstrates that the novel approach beats a state-of-the-art baseline both within the domain in which it was trained and a separate domain, without any

drop in performance when moving to the separate domain. We discuss implications for practice in at-scale learning environments. We conclude with limitations and directions for continued research.

8.1.1 Transferable Attention Model from Entailment to Transactivity

In our work, we employ a transfer method that used the recognizing textual entailment task as pretraining task. In particular, for the entailment pretraining we specifically consider the simple attention model proposed in the language technologies community (Parikh et al., 2016b). It is notable that despite the presented model's simple structure and relatively small number of parameters, it performs comparably with far more complex models that have orders of magnitude more parameters. As we are looking to work with small datasets, models that are both simple and effective are a natural choice.

In recent years, large datasets for the textual entailment task have been developed and made available for researchers (Bowman et al., 2015). State of the art performance on these datasets have been rising steadily with use of complex recurrent neural networks (Sha et al., 2016), neural attention models (Parikh et al., 2016b), tree based neural models (Munkhdalai and Yu, 2016), and hybrid methods using both of those approaches (Wang et al., 2017; Chen et al., 2016). Models trained on such a corpus to identify concepts linked through inference across a plethora of domains are required through the training process to build conceptual representations for words that make identification of conceptual links possible. The idea behind our computational approach is to leverage this tendency in a pretraining step for training to detect Transactivity in one topic domain so that rather than learn just the associations between specific pairs of concepts, the model would learn to leverage the entailment representation space that enables computation of idea relatedness of texts across domains. The hope is that a model trained to detect transactivity in one domain but building on this general purpose representation space would be able to transfer to another domain where the relevant set of linked concepts is different but still within the broad range of topics covered inside the very broad and diverse entailment corpus.

We refer to our adaptation of the original Decomposable Attention Model as the Transferable Attention Model. Specifically, we adapted the model described above for the purpose of transfer learning. We started by separating the model into two modules. The fist module includes both the attention and comparison components, which generate sentence representations from the input representations referred to in deep learning work as word embeddings. The second module includes the classification step, which takes in the two text segment comparison vectors and makes a prediction for the text pair's class.

The reason we needed to separate these components of the model is that, while performing transfer learning, we need to be able to dynamically manipulate the weights or structures of the classification stage while maintaining the integrity of the parameters learned in the representation stage. This allowed us the flexibility to have varying numbers of classes between our source task and our target task. The modularity also allows for varying types of classifiers or bindings to other models that we consider for future work.

8.1.2 Decomposable Attention Model

For our modeling work, we adapt the previously published Decomposable Attention Model presented by (Parikh et al., 2016b) for the purpose of transfer learning from the Entailment task to Transactivity detection. This model was chosen as a starting point for this work by virtue of it demonstrating benefits including using an order of magnitude fewer trainable parameters than other common methods in for approaching textual entailment while maintaining a high level of performance. Furthermore, the model was not bound by a task specific architecture or feature set that makes it a good candidate for multi-task learning with pairwise comparisons. The Decomposable Attention Model operates in four stages: input, attention, comparison, and aggregation. We will provide an overview of their model to provide context for our adaptations and an intuitive explanation for the benefit of the reader. See Figure 21 for a visualization of the structure of the model.

Input: The model is defined with input of two text segments, $\mathbf{a} = (a_1, ..., a_m)$ and $\mathbf{b} = (b_1, ..., b_n)$ where m and n are the lengths of the respective segments. Each vector a_i and b_j are real value, d dimensional vector embeddings for each word in their respective text segments. For words not in the vocabulary, an embedding is assigned randomly based on the word's shape. The output of the model is defined as $y = (y_1, ...y_C)$ where C is the number of output classes for the dataset.

Attend: At the first stage of the model, each input is passed into network F where a soft alignment between word embeddings is computed via a type of neural attention (Bahdanau et al., 2014). The attention mechanism weights the importance of each word in each sentence for how it will be used in the subsequent computations. The network, F is a simple feed forward neural network with rectified linear activation (Glorot et al., 2011). This results in a matrix of dimension $m \times n$, $e_{ij} = F(a_i, b_j)$, where each cell contains a score of how important each given word in a text segment is, given that it co-occurs with another word in the other text segment.

The matrix is then normalized for each direction to obtain two vectors, α and β, to represent the aligned subphrases from **b** to **a** and **a** to **b**:

$$\begin{aligned} \beta_i &= \sum_{j=1}^{n} \frac{\exp(e_{ij})}{\sum_{k=1}^{m} \exp(e_{ik})} b_j, \\ \alpha_j &= \sum_{i=1}^{m} \frac{\exp(e_{ij})}{\sum_{k=1}^{n} \exp(e_{kj})} a_i. \end{aligned} \tag{3}$$

Compare: In the next stage, each aligned phrase is compared separately by an additional feed forward neural network, G:

$$\begin{aligned} v_{1,i} &= G([a_i, \beta_i]) \quad \forall i \in [1, ..., m], \\ v_{2,j} &= G([b_j, \alpha_j]) \quad \forall j \in [1, ..., n]. \end{aligned} \tag{4}$$

There are now two sets of vectors that encode a comparison between the input and the aligned subphrases of each input text segment.

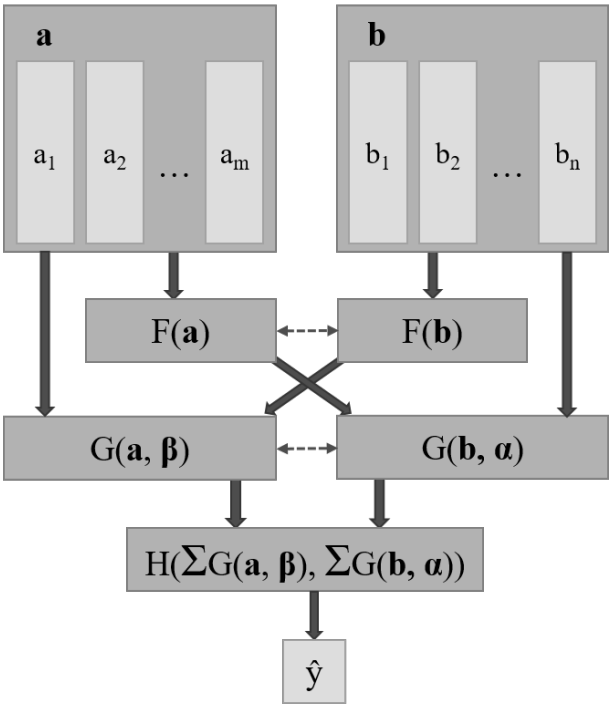


Figure 21: Decomposable Attention Model. Arrows with dotted lines indicate networks with shared weights.

Aggregate: The final stage of the model compresses the two sets of vectors via summation, giving a vector representation for each text segment with respect to the other.

$$\mathbf{v}_1 = \sum_{i=1}^{m} v_{1,i} \quad , \quad \mathbf{v}_2 = \sum_{j=1}^{n} v_{2,j}. \tag{5}$$

These two vector representations are then concatenated and fed into a final feedforward neural network with softmax activation, H, to predict probabilities of class values: $\hat{\mathbf{y}} = H([\mathbf{v}_1, \mathbf{v}_2])$. The predicted class is thus $\hat{y} = \arg\max_i \hat{\mathbf{y}}_i$

8.1.3 Datasets for Domain Generality

Throughout the experimental work reported in this chapter, we used five datasets to demonstrate first task transfer and then domain generalizability. Short descriptions of each are provided here. We will refer to two main tasks: the Entailment task, which is our source task, and Transactivity Detection, which is our target task. We also refer to two domains in which we perform the Transactivity task. The source domain, which is a Power Plant domain, is where the training for the Transactivity task is performed. And the target domain, which is the Superheroes domain, is the domain for the Transactivity task where we do the test of domain generality of the trained Transactivity task model.

Stanford Natural Language Inference Corpus: As our primary dataset for the Entailment task, we selected the Stanford Natural Language Inference Corpus (SNLI), version 1.0 (Bowman et al., 2015). This corpus contains over 570 thousand annotated text pairs for the recognizing textual entailment task. Pairs consist of a premise and a hypothesis, and are labeled as *entailment* if the hypothesis is definitely true given the premise, *contradiction* if the hypothesis is definitely false given the premise, and *neutral* if the hypothesis could be true, but is not guaranteed to be given the premise. The premises were captions from the Flickr30k corpus (Young et al., 2014) and the hypotheses were generated via an Amazon Mechanical Turk task where workers were asked to write three alternate captions that followed certain rules to create appropriate hypotheses for the entailment task.

Multi-Genre NLI Corpus: The Multi-genre Natural Language Inference Corpus (MultiNLI), version 0.9 (Williams et al., 2016) consists of over 390 thousand text pairs annotated in the same way as the SNLI corpus described above. However, this dataset includes text segments from four different categories: fiction, government texts, magazine articles about popular culture, and transcripts of telephone speech. As this is a comparable dataset to SNLI, we determined that it was a valid alternative for pretraining in our experiments.

Power Plant Transactivity Corpus: Our larger annotated Transactivity dataset, which is a shared dataset we used as a target task to transfer our Entailment model, comprises 426 annotated text segments (Wen et al., in press). These text segments come in the form of posts made by participants from Amazon's Mechanical Turk working in teams where they needed to determine which type of power source(s) a city should make use of given a set of characteristics that the city possesses. For each instance, the labeled post is in reply to a previous post which is also included in the representation of the

instance for reference. Each instance was annotated as *Transactive* or *not Transactive* with respect to the context.

Superhero MOOC Transactivity Corpus: This set of annotated Transactivity data consists of 57 annotated text segments from a Massive Open Online Course in which students design superheroes and discuss them with other members of the course (Wen, 2016). The data are collected conversations between students. Each contribution was annotated as *Transactive* or *not Transactive* with respect to the conversation. Each instance was annotated as *Transactive* or *not Transactive*, just like in the previous corpus.

Microsoft Research Paraphrase Corpus: The Microsoft Research Paraphrase corpus (Dolan and Brockett, 2005), has 5801 annotated sentence pairs that are either labeled as *paraphrase* or *not a paraphrase*. This will be used in one of our validation experiments.

8.1.4 Training and Implementation Details

Training the Transferable Attention Model is performed in three stages: first training the model on the source task for a given number of iterations, then dynamically changing the classification module to match the target task, and finally training the new model on the target task until convergence.

During the training of the target task, error is propagated backwards through both modules of the model to allow for fine tuning of the attention and comparison networks for the Transactivity task. Input word embeddings are held fixed throughout the training. This backpropagation method is a standard training approach for neural network models.

We implemented the Transferable Attention Model using the Keras deep learning library (Chollet et al., 2015) with the Theano tensor library (Theano Development Team, 2016) as a foundation. Each network, F, G, and H were 2 layer feed forward densely connected networks with 200 hidden units per layer. The structure of H was the same for both target and source task with the exception that the output dimension of the target task was 2 while in the source task the output dimension was 3. Text segments were fixed at 100 tokens with zero vectors left padding the text segments if the length was shorter and truncating if the length was longer. Word embeddings were 300 dimension pretrained GloVe (Pennington et al., 2014) embeddings.

Our model was trained with the Adam optimizer (Kingma and Ba, 2014) with a learning rate of 0.001. Training was done on a NVIDIA GeForce GTX 760 with CUDA 8.0 (Nickolls et al., 2008) and CuDNN 5 (Chetlur et al., 2014). One iteration of pretraining on the source task was performed, classification weights were reset with a random Gaussian distribution, and ten iterations of training on the target task were performed per fold during the experiments. Metrics for the tenth iteration of target dataset training were reported in all cases.

Beyond demonstrating the performance of our model on the given task, we also motivated our experiments with validating that our model operated as our intuitions predicted.

The metrics that we collect throughout our experiments are accuracy, to see the percent correct of the predictions each model makes, and Cohen's kappa, to evaluate the

models' accuracy in a way that controls for agreement by chance. Results are reported in the Results and Discussions section and in each of the corresponding subsections.

8.1.5 Domain Generality Experiments

Cross Domain Generality

In order to evaluate our method on the task of Transactivity detection we test our method of transfer learning against several baselines, which are described below.

After pretraining the model on the SNLI corpus, we perform a standard ten-fold cross validation over our Transactivity training corpus, in each fold beginning with the model weights generated by the pretraining. After each fold, we evaluate the trained model on the held out Transactivity data from the source domain (i.e., the Power Plan data). We also apply the model trained in each fold to the data in the target domain (i.e., the Superheroes data). As is a standard practice for evaluation by cross-validation, results for all the folds are averaged together for our final metric, reported in the Cross Domain Generality subsection.

Baselines

Logistic Regression with Unigrams: Previous work by Joshi et al. (Joshi and Rosé, 2007) on predicting transactivity used a simple unigram model (Pang et al., 2002) with logistic regression, trained on the Power Plant Transactivity Corpus. We therefore use this as a baseline to connect our method with previous work in that field.

Basic Neural Network: As our model consists of only feed forward neural networks, we evaluated the performance of a basic neural net architecture without an attention mechanism using the same GloVe word embeddings. We use a 2 layer feed forward neural network with 200 hidden units per layer, as that is the equivalent structure for the classification step of our model. We allowed this model to be pretrained as with our Transferable Attention Model.

Bidirectional Long Short-Term Memory: Many systems in entailment use LSTM (Hochreiter and Schmidhuber, 1997), and the bidirectional variant, BLSTM (Graves and Schmidhuber, 2005) based models with word embeddings (Bowman et al., 2016). We also evaluated our model with sentence embeddings generated by single layer BLSTMs with 128 hidden units each direction, then classified with a densely connected layer. This model was also pretrained on the SNLI corpus.

Lexical Overlap In early work with textual entailment, it was shown that simple word overlap is a strong predictor of entailment (Bos and Markert, 2005). Because of the similarity between entailment and transactivity, we hypothesized that this may hold for our task as well so we investigated to ensure our model was making inferences beyond that naïve method. To eliminate this possibility, we removed all overlapping words between target and context sentences for both the entailment dataset and the transactivity datasets during test and training. We then report the results of our model, trained and evaluated as in our first experiment above.

This makes the task considerably more difficult as the model loses access to a large amount of content based context. It therefore must rely on non-overlapping structural information in the texts, synonyms, or more abstractly connected words.

Dataset Alignment In the SNLI dataset, there are three classes, *entailment*, *contradiction*, and *neutral*, one of which can be applied to each text pair. However, in the

transactivity dataset, each pair can only be identified by either *transactive* or *not transactive*. When arranging the data between pretraining on entailment and training on transactivity data, we need to decide how these classes map to one another to give the pretraining the most impact. *Entailment* and *neutral* are easy to correspond to *transactive* and *not transactive* respectively given that the former indicates a logical connection between the two while the latter indicates there is not. *Contradiction*, on the other hand, is more difficult to determine. The hypothesis can either be considered connected to the premise through logic that makes the hypothesis impossible or it can be considered not connected as it is not entailment.

We tested applying the contradiction component of the pretraining data differently to evaluate which performed the best for the transfer learning. The conditions that we evaluated were relabeling the contradiction cases as either entailment (*contradiction positive*) or neutral (*contradiction negative*) before evaluating as in the Cross Domain Generality experiment. We also pretrained with all three entailment classes and just ignored the contradiction label while training and evaluating on transactivity. Discussion of these results can be found below in the Results section.

Ablation This set of experiments was designed to make sure that the transfer learning was having sufficient impact to warrant their inclusion in the model. We first tested to ensure that the pretraining was being utilized by the model and not simply being overwritten by the training that the model performs over the transactivity dataset. To accomplish this, we executed the experiment as in the Cross Domain Generality case without pretraining the model on the entailment dataset. We then evaluated on only the in domain data.

To ensure that the model was not simply applying textual entailment to our transactivity dataset and that it learned something meaningful from the small dataset, we ran the experiment with only weights learned on the entailment task and evaluating on the in domain transactivity test data. Both of these experiments are reported below in the Results section.

Alternative Datasets The last set of experiments were motivated by the possibility that the entailment task was not necessarily the explanation for the performance of the model. We considered two alternative explanations: that the SNLI corpus may be particularly suited for transfer leaning in this domain, or that any sentence comparison task would transfer sufficiently for transactivity to be predicted.

To evaluate the first consideration, we tested the model using an alternate source dataset, the MultiNLI corpus. In this evaluation, our source task was the same as before, but the data used to pretrain was different.

To evaluate the second consideration, we evaluated our model when the source task was changed to the similar, though not identical, task of paraphrase detection using the MSRP corpus. Key differences between paraphrase detection and entailment is that entailment represents a directed relationship between text pairs, while paraphrase detection is undirected. Paraphrase detection also has only two output classes compared to entailment's three.

One issue that we needed to control for when pretraining with paraphrase detection was that the dataset was significantly smaller than either entailment corpus. To provide a fair comparison, we randomly selected an equivalent number of SNLI and MultiNLI examples to pretrain with and reported those results as well.

Data Set Size

One of the most frequent questions asked about automated approaches to discussion analysis that require training is how much data is required. Thus we include one additional experiment that manipulates the amount of training data and shows how performance varies as a result.

	Accuracy		**Cohen's Kappa**	
Models	*In domain*	*Out of domain*	*In domain*	*Out of domain*
Unigrams with LR	0.795	0.667	0.510	0.376
Basic Neural Network	0.798	0.721	0.498	0.305
Bidirectional LSTM	0.814	0.782	0.543	0.472
Transferable Attention (TA)	**0.840**	**0.832**	**0.607**	**0.611**

Table 14: Model performance in domain versus out of domain compared to baselines.

	Accuracy		**Cohen's Kappa**	
Models	*In domain*	*Out of domain*	*In domain*	*Out of domain*
Unigrams with LR	0.781	0.667	0.476	0.363
Basic Neural Network	0.761	0.733	0.412	0.309
Bidirectional LSTM	0.812	0.772	**0.524**	0.442
Transferable Attention (TA)	**0.828**	**0.810**	0.475	**0.551**

Table 15: Model performance in domain versus out of domain compared to baselines with no lexical overlap between target and context.

Models	**Accuracy**	**Kappa**
TA	**0.840**	**0.607**
- pretraining	0.700	0.035
- transactivity training	0.307	0.005

Table 16: Model performance with varying training stages removed.

8.1.6 Results

Cross Domain Generality Table 14 shows the results for our comparison of our model's performance on the in-domain transactivity dataset to the out of domain transactivity data set after pretraining on the SNLI corpus for the entailment task. We find that our model outperforms the baselines in all metrics, with over 80% accuracy and a kappa of over 0.6 indicating good agreement with annotators. When comparing accuracy between tests, we can see that our model loses less than one percentage point, while the unigram baseline drops over 12 percentage points when evaluating on the out of domain set. The

	Accuracy		Cohen's Kappa	
Models	*In domain*	*Out of domain*	*In domain*	*Out of domain*
TA with contradiction negative	**0.848**	0.824	0.542	0.586
TA with contradiction positive	0.828	0.791	0.598	0.511
TA with three classes	0.840	**0.832**	**0.607**	**0.611**

Table 17: Model performance with respect to how contradiction was treated in task transfer.

	Accuracy		Cohen's Kappa	
Models	*In domain*	*Out of domain*	*In domain*	*Out of domain*
TA with full SNLI training set	0.840	**0.832**	0.607	**0.611**
TA with full MultiNLI training set	**0.869**	0.804	**0.647**	0.544
TA with both SNLI and MultiNLI	0.833	0.828	0.536	0.585
TA with truncated SNLI	**0.781**	**0.786**	**0.328**	**0.464**
TA with truncated MultiNLI	0.764	0.761	0.255	0.383
TA with MSRP training set	0.752	0.751	0.210	0.345

Table 18: Model performance with respect to dataset used for pretraining.

simple word embedding based baselines also appeared to drop across domains, though not as dramatically as the unigram model.

From this, we can infer that learning to operate over general semantic vectors can influence the domain generality of classification models. We also demonstrate that transferring learned representations from a deep model trained on a general source task can improve performance on multiple domains of a target task even if the model was only trained on a single domain of the target task.

Lexical Overlap A similar story is seen in Table 15 with lexical overlap between target and context text segments is removed. All of the tested models dropped performance modestly, though our model still managed to get an accuracy of over 80%. This provides compelling results that the reasoning our modeling is doing between the two text segments is more abstract that simply measuring word overlap.

Dataset Alignment Because the source task is a three class classification and the target task is a two class classification, we considered alternative alignments between categories, which we found to have different implications for performance in the two transactivity datasets. The results presented in Table 17 make sense when the data is examined qualitatively.

In the condition in which contradiction was used as a positive example, the model obtained a notably higher kappa on the in domain dataset that contained more competitive transacts, demonstrating disagreement. However, when contradictions were treated as negative examples, the model performed much better on the out of domain dataset which contains a lower percentage of competitive transacts. When contradiction is given a separate class during source task training and not used in target task training,

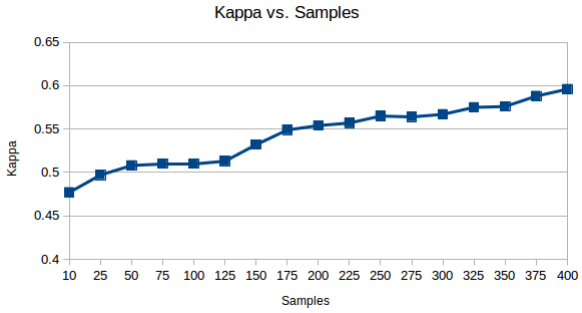


Figure 22: Graph of the change in kappa score over varying number of transactivity training instances.

the kappa is higher for both target task datasets indicating that the model was free to make a determination on the role of learned contradiction-type relationships as it applies to the transactivity task.

Ablation

Table 16 reveals that the pretraining on the source task and the training on the target task are both critical for the performance of the model. This indicates that the model learned important representation structure from the large amount of data provided with the source task. It also can be seen to not only classify the target task as if it were the source task, but rather it learned about the difference between the tasks sufficiently to adapt to the new task.

Alternative Datasets In our final set of experiments as reported in Table 18, we can see that there are comparable results between using SNLI and MultiNLI for pretraining. An interesting observation is that pretraining on the MultiNLI corpus seemed to perform better for in domain transactivity detection while pretraining on the SNLI corpus had stronger results for out of domain prediction. This raises some interesting questions regarding how the domain of the source data sets can influence the generalizability of target datasets while transferring learning.

We can also see that with a smaller number of source task text pairs, it appears that SNLI provides the best performance, followed by MultiNLI, then MSRP performs the worst. This provides some evidence that the entailment task is providing more valuable pretraining as compared to paraphrase task.

Dataset Size Here we address the question of how much data is required for training in order to achieve the best performance. We ran a series of cross-validation experiments using the full Transfer Attention Model where we manipulated the number of training instances sampled from the maximal training set on each fold of the cross-validation. The results are displayed in Figures 22 and 23 for Kappa and Accuracy respectively. Here we see progressive improvement as more and more data is used, without a substantial plateau. Thus, it is possible even better performance could have been achieved had we provided more data, and a smaller training set size would have yielded poorer performance.

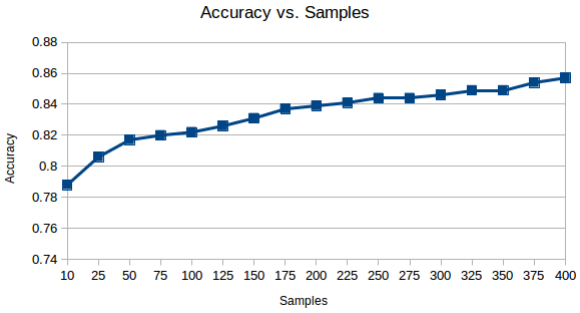


Figure 23: Graph of the change in accuracy score over varying number of transactivity training instances.

8.1.7 Discussion and Implications

The results presented in this chapter demonstrate that the novel neural approach to classification we present achieves an improvement in accuracy as well as generalizability over previously published work on automated Transactivity detection (Joshi and Rosé, 2007; Rosé et al., 2008; Ai et al., 2010; Mu et al., 2012; Gweon et al., 2013).

Automated Transactivity detection has a variety of applications in online learning environments, especially where discussion is part of the learning process. The presence of Transactivity is a significant predictor that a collaborative discussion is conducive to learning (Berkowitz and Gibbs, 1983; Azmitia and Montgomery, 1993; Joshi and Rosé, 2007; Gweon et al., 2013; Rosé et al., 2017). That makes Transactivity a construct that is particularly valuable to be able to detect.

In general, automated detection of discussion processes that are either positively or negatively related to learning can be applied to problems such as automated assignment of students to project teams (Wen et al., in press), for triggering dynamic support of group learning processes (Kumar et al., 2007; Ferschke et al., 2015), and for assessment of those learning processes (Rosé et al., 2017). Raising the level of accuracy at this detection increases the feasibility of offering these forms of automated support in massive online learning environments.

The generalizability result has particular implications for learning at scale. Scale is not just about reaching a large number of students in one course or offering the same course many times, but being able to apply a form of learning support broadly across courses. Without the ability to generalize a model's performance to new data sources, it would be necessary to train a new Transactivity detection model for every course, or maybe even every assignment where the model will be used. Clearly, a solution that requires retraining over and over is more costly to use than one that can be trained once and then reused many times in many different contexts.

8.2 Commonalities and Differences in Transactivity Models Across Different Types of Writing

In this study. we investigate the ability to adapt a model to a new student population, namely. masters students in a large North American business school, where we observe strikingly different patterns of transactive exchange than in prior studies. We found that this difference prompted required both a reformulation of the coding standards and innovation in the modeling approach (Fiacco et al., 2021).

8.2.1 Automated transactivity detection experiments

For our experimental exploration for this work, our goal is to find a model that can most accurately predict the various facets of transactivity that we have defined in our dataset. To this end, we started with the Transferable Attention Model for transactivity detection to evaluate the ability of it to detect our more nuanced definition of transactivity in our data. A data analysis was performed to attempt to explain the discrepancy in performance of the baseline model on each dataset. With the lessons learned from the data analysis, a new detector for transactivity was proposed to address the shortcomings of the baseline model. We provide an evaluation of the new transactivity detector.

Results for each experiment for transactivity detection were obtained via a 10 fold crossvalidation where each fold was randomly assigned but consistent throughout the different conditions.

Baseline: Transferable attention model for transactivity detection We use the Transferable Attention Model described above as the baseline for this section. Similarly to the previous section we create analogues for entailment in the transactivity task. While the entailment task takes in a premise and a hypothesis statement to train the model with the hypothesis statement being the statement to be determined if the entailment relation holds, in the transactivity task, the premise is replaced by the context and the hypothesis is replaced by the message. The message is the text that is to be labeled as transactive and the context is the text for which the message is responding to.

For experiments on our dataset, the message was the post that is to be determined to show one of the aspects of transactivity while the context is the post that message was a response to. Note that the message and context may not be temporally adjacent as determination for message response was made via the forum response tree and participants can respond directly to prior posts.

Comparisons of transactivity data with respect to transferable attention model The first research question we sought to address stems from a comparison between the data used previously to train the Transferable Attention Model and the new dataset from class discussions noting that previous datasets used far more concrete language as opposed to our new dataset. Concreteness of language is characterized by referring to specific objects, people, or actions while abstractness is defined as language referring to concepts and ideas.

In Table 19, we present the abstractness of each dataset based on the average abstractness of inputs using the methodology from Brysbaert et al. (2014). We evaluate the transferable attention model using an alternative entailment pretraining dataset, the Multi-genre Natural Language Inference corpus (MultiNLI) (Williams et al., 2018)

Datasets	Text Abstractness
SNLI (Bowman et al., 2015)	0.334
MultiNLI (Williams et al., 2018)	0.530
Powerplant Transactivity Corpus	0.538
MBA Student Corpus	0.583

Table 19: Abstractness for datasets relevant to transactivity detection; scale 0 (concrete) to 1 (abstract).

which we found to be considerably more abstract than the Stanford Natural Language Inference (SNLI) corpus (Bowman et al., 2015) which was the pretraining corpus for the original Transferable Attention Model. This pretraining corpus was hypothesized to improve the model's performance by better representing the more abstract text found in the MBA student data.

Transformer model for transactivity detection One of the key shortcomings of the Transferable Attention Model is its inability to take into account word order. This is especially relevant to the Challenging Views dimension as negation is common within examples of that dimension and the meaning of a negation is highly word order dependent. To address this, we propose to use a class of models from the Natural Language Processing literature called transformers (Vaswani et al., 2017). The benefit of this type of model is that it combines the capability for self-attention with sequential reasoning to build a numerical representation of a sequence of text that can be used to classify that sequence.

Specifically, we use the pretrained transformer model, RoBERTa (Liu et al., 2019) which incorporates some optimizations of the BERT transformer model (Devlin et al., 2019). This model, like the GloVe embeddings used in the Transferable Attention Model, was pretrained on an enormous amount of general text data and will be fine-tuned on both the entailment pretraining task and the transactivity task. The model was then fine-tuned on the Recognizing Textual Entailment task similarly to the Transferable Attention Model. This fine-tuned model was the based model for each of the crossvalidation folds. For each fold, the model was further fine-tuned on the transactivity data with a separate classification head as the entailment classifier.

8.2.2 Evaluation

We evaluated the potential to automate analysis using the extended transactivity definition proposed here beginning with the best published approach from Fiacco & Rosé (2018), and comparing its approach to three other variants. From Table 20, it is evident that pretraining the Transferable Attention Model on the MultiNLI dataset had a large positive effect ($p < 0.05$) on all of the dimensions of transactivity. The increase was most notable for Active Listening while there were only modest improvements for Challenging Views and Idea Expansion.

Even more dramatic is the increase in performance from the redefinition of inputs for the Transferable Attention Model to make the model perform self-attention rather

	Cohen's Kappa		
Models	*Active Listening*	*Challenging Views*	*Idea Extension*
Transferable Attention Model (Fiacco & Rosé, 2018)	0.840	**0.832**	0.607
Transferable Attention Model+MultiNLI	**0.781**	**0.786**	**0.328**
Transferable Attention Model+MultiNLI+Self Attn.	**0.869**	0.804	**0.647**
RoBERTa+MultiNLI	0.833	0.828	0.536

Table 20: Cohen's kappa scores of transactivity detection models on 10 fold crossvalidation.

than attending between context and content. Furthermore, the RoBERTa model is able to significantly improve upon the performance on the Challenging Views dimension. However, it did not significantly improve on Idea Extension and underperformed on Active Listening. All differences between rows in Table 20 are significant ($p < 0.05$).

8.2.3 Discussion

The results of the experiment reveal data considerations that must be taken into account when modeling approaches are used for automated detection of constructs such as transactivity. The line of experimentation reported here was prompted by an observation that our previous work of demonstrating of domain generality could not be generalized to a substantially different student population with its own distinctive discourse practices. The investigation reported in this paper points to needed adjustments first at the level of operationalization of the construct and then at the level of modeling approach – with synergistic considerations between the two – in order to achieve success.

In particular, the results reported above reveal a larger increase in performance for the Active Listening dimension between the baseline Transferable Attention Model and the version that used the MultiNLI pretraining as compared to either Challenging Views or Idea Extension. We attribute this largely to the vocabulary of the NLI datasets as compared to the MBA student data. The MBA student data is far more abstract than the SNLI dataset as compared to MultiNLI dataset. Active Listening is a relatively simple task as compared to Challenging Views or Idea Extension as it is frequently signalled by agreement or disagreement. As the SNLI dataset is based off of image descriptions, there is little opportunity for language such as that to occur. The MultiNLI corpus pulls data from a far broader range of genres and may expose the model to more relevant sentence forms. For the other two transactivity dimensions, the limiting factor was not as much the vocabulary, but how the model was able to use the data it had.

There was a large jump in performance across all dimensions of transactivity by redefining the Transferable Attention Model as a self-attention model as opposed to attending between the content and its context. While in data with less abstract contributions, the important factor for detecting transactivity may be ensuring that there are aligned phrases between the content and the context, in our MBA student dataset, it appears to be more important for the model to understand what the responder is contributing. This result aligns with our qualitative observations that the MBA students

had deeper contributions and more structured responses as compared to the contributions in prior datasets. Detecting transactivity, in this case, is more about evaluating how well formed the response is, regardless of the context.

8.3 Conclusion

Broadly, this chapter seeks to make a small advancement to the area of model selection for transfer learning by matching the properties of the models we select to the features that the networks were induced to learn from the data. Specifically, we do this in terms of generalizing performance in detecting transactive interaction in electronically mediated communication. This serves to broaden our understanding of how variable task objectives and training data influence neural pathway development, ultimately offering a more flexible and insightful framework for neural network design and application.

9 Identifying Propagation of Problematic Pathways in Fine-Tuned AI-Writing Detectors

In this chapter, we examine foundation models and their application to AI writing detection in an effort to understand the evolution of potentially biased components of a RoBERTa-based (Liu et al., 2019b) AI-writing detection model (Solaiman et al., 2019) during the fine-tuning process. The first step towards this goal is the construction of a new AI-writing detection dataset within the context of student writing that complements existing datasets in AI-writing detection to assist in the identification of the source of bias within our analysis model.

Second, we undertake an in-depth examination of the fine-tuning process within functional groups of neurons in the LLM. This granular analysis allows us to uncover patterns and mechanisms that may otherwise remain obscure, offering valuable insights into how biases might be inadvertently introduced or amplified during this critical phase of model training.

Lastly, we present empirical evidence to show that important functional components are retained from pretrained models, leading to sub-optimal behavior even in cases when early stopping would culminate. This finding yields a cautionary note about the potential risks of heavily depending on a small pool of pretrained LLMs, often referred to as foundation models in the literature (Bommasani et al., 2021; Zhou et al., 2023), for a broad spectrum of tasks. It calls for further investigation into the implications of these inherited components and proactive measures to prevent the propagation of undesirable biases.

The contributions of this work can thus be summarized as follows:

- The introduction of a novel AI-detection dataset specifically designed for student writing.
- An analysis of the evolution of functional groups of neurons during the fine-tuning process for a RoBERTa based AI-writing detection model.
- Evidence demonstrating that the most important functional components for a model's decision making are inherited from the pretrained weights.

With the release of a new AI-writing detection dataset for student writing, this work advocates expanding an under-researched aspect of bias in machine learning, the risk of a foundation model monoculture. While understanding the functions of coordinated neuron groups is still nascent, it is imperative to address both bias detection and mitigation in LLMs while also avoiding a homogeneous pool of pretrained models. By addressing these concerns, we can foster a more diverse and responsible approach to AI development and contribute to the advancement of the field.

9.1 The Role of Neural Pathways in Model Decision Making

From Chapter 4, we have a method for aligning separable Bayesian networks with the functional components of a neural network. Each aligned pathway can therefore represent a subgraph of the equivalent Bayesian structure encoded by the neural model.

This can be viewed as an independent decision-making component of the neural network. This enables us to examine each component separately when determining if the model uses sensitive information.

There is a fundamental limit to the information that pathways can extract from a model wherein non-separable variables cannot be distinguished from each other (discussed in Chapter 4) and a set of defined outcomes from the approach (Chapter 5). These limitations may seem restrictive because variables in natural data, especially those of interest to the bias and fairness community, are often coupled and do not meet the separable criteria that idealized neural pathways would require.

Though this is the case, we can still extract useful insight from the pathways by utilizing external knowledge of the task to assert for how likely a pathway that correlates with protected information is to represent a problematic variable. Through the methods described in Chapter 5, we can distinguish between a proxy variable for sensitive information and an allowable variable that is only correlated to the sensitive attribute. The key mechanism from Chapter 5 that we use to make this distinction is the examination of error cases in the model predictions in light of available external task information.

At this time, it is of the utmost importance to stress that the ability of this method to distinguish reasonable from problematic pathways is limited by the external information that is brought to the analysis. A correlation with sensitive information cannot be explainable without having some knowledge of the proposed explanation. We therefore perform our analysis in a "fail-safe" manner, where a pathway that is correlated with a protected attribute is considered problematic until sufficient evidence demonstrates that it is not.

9.2 Aligning Functional Components Over Time

In previous work on extracting functional components, extraction was only performed on the final trained model (Fiacco et al., 2019a, 2022) (or multiple trained models for comparison (Fiacco et al., 2023)). However, there is no precedent for aligning functional components across time for the same model. In this section, we describe a procedure to extract functional components from snapshots of a model at each epoch during fine-tuning and correlate the resulting important components across adjacent epochs. The procedure has three phases: first extracting and saving the activations from the model during training; second, computing the functional components activations and determining the importance of each component; and third, correlating components within each temporally adjacent epoch.

In the initial phase, activations are extracted and saved from the model during its training process. For each epoch (including model before fine-tuning), these activations correspond to the responses of the neurons to every data instance in the analysis set. These activations are recorded and stored in an activation matrix, A, which forms the basis for subsequent computations. This technique provides a time-series of the model's neuron activations during the course of its training.

In the second phase, the functional components' activations are computed, and the importance of each component is determined. The computation of functional components is achieved by applying ICA to the activation matrix A. This step yields the

Dataset	Total Documents	Average Tokens/Document
GPT-WIKIPEDIA-INTROS	300,000	163
AES-ASAP-DETECT	3566	417
TOEFL ESSAYS	182	420

Table 21: Comparison of key statistics between human-authored and AI-generated essays in the AI-augmented corpus

functional component activation matrix, F_{model}. Each component's importance is then evaluated by calculating the Pearson's correlation coefficient between each column of F_{model} and the model's predictions. An important functional component is thus defined as above with the threshold for importance set to 0.05, which while it appears to be a very low correlation, in large models, each functional component often only contributes to a small portion of the models output.

The third phase involves correlating components within each temporally adjacent epoch. Up until this point, each time-step of the model was treated as an independent entity, however, in this phase components are correlated across consecutive epochs. This step involves tracking the correlation between the identified important components from one epoch to the next, producing a sequence of correlation matrices, C where each matrix has the shape of $P_k \times P_{k+1}$ where P is the number of components of the model at time-step k. This novel approach of aligning functional components across time allows for the observation of the model's learning behavior and the change in influence of these components on the model's predictions over the course of its training.

9.3 Datasets for Distinguishing Between Generated and Human Written Language

Three datasets are used in this work, our AI detection model was trained on the GPT-Wikipedia-Intro dataset (Aaditya Bhat, 2023) and evaluated with both the TOEFL essays dataset (Liang et al., 2023) and our newly constructed dataset (AES-ASAP-Detect).

9.3.1 GPT-Wikipedia-Intro Dataset

This dataset consists of 150,000 topic introductions from Wikipedia, along with corresponding introductions generated by the "Curie" GPT model (Aaditya Bhat, 2023). The generation process followed a specific prompt format, wherein a 200-word introduction in the style of Wikipedia was created for each topic. The prompt included the title of the Wikipedia page, and the initial seven words of the actual Wikipedia introduction are used as starter text. From this dataset, we randomly divided the topics into an 80% train and 20% evaluation set wherein both the human and GPT generated introductions for a given topic would both be in only one of the sets.

9.3.2 TOEFL Essays

This dataset (Liang et al., 2023) is comprised of 91 human-authored TOEFL (Test of English as a Foreign Language) essays obtained from a Chinese educational forum each of which has a corresponding essay that was revised by ChatGPT-3.5 (OpenAI, 2022). This dataset provides a corpus of essays from non-native English speakers which highlighted the challenges faced by several off-the-shelf AI detection tools in differentiating between them and AI generated text. Furthermore, it was found that it was also difficult for AI detection tools to determine that the revised essays were written by ChatGPT.

9.3.3 ASAP-AES Detect

While Liang et al. (2023) constructed a dataset derived from the Automated Essay Scoring dataset from the Hewlett Foundation's Automated Student Assessment Prize (AES-ASAP)[3], their approach was to use ChatGPT to simplify the existing essays to make them explicitly more difficult to distinguish from non-native English writing. However, we desired a corpus of fully AI-generated essays to evaluate the capability of AI-writing detectors to distinguish between the writing styles of students and ChatGPT when following an essays prompt.

With this goal, we present a supplement to the first essay set of the AES-ASAP dataset, called AES-ASAP-DETECT. By using the AES-ASAP dataset as a foundation, a complementary dataset is generated via prompting ChatGPT (OpenAI, 2022). The use of the dataset of human writing in conjunction with the LLM provided a suite of generated essays that are more diverse and similar in content and style to the human written essays. This makes for a more challenging dataset for detecting AI-generated writing and a better foundation for the further exploration of the learned functions of the detectors.

The first essay set of the AES-ASAP corpus consists of over 1,700 essays with a broad range of writing styles where each address the topic of the impact of computers on society. The authors of the essays were grade 8 students. The LLM was prompted to write the essays based on summaries of the original essays constructed from a previous LLM prompt, the qualities of a good essay from the rubric for grading the essays, and the original essay prompt. The essay summaries were prompted to be 3 sentences and were also generated using ChatGPT-3.5.

Furthermore, an additional pass was required for the essay generation was required because there was a preprocessing step performed on the original essays wherein proper nouns and numbers were replaced by numbered placeholder tokens (e.g. @ORGANIZATION3). While in this work, we are only using this dataset as an evaluation set, without this step, any model trained on it would yield a trivial, degenerate solution to differentiate between the human and AI generate texts by identifying the presence/absence of the placeholder tokens. Therefore, a new prompt was used to modify the generated essays to include those tokens in their text with the appropriate number scheme. The resulting modified essays we observed to have a reasonable distribution of the placeholder

[3] https://www.kaggle.com/c/asap-aes

	Epoch							
Dataset	0	1	2	3	4	5	6	7
GPT-Wikipedia-Intros	0.884	0.997	0.983	0.679	0.701	0.718	0.723	0.724
AES-ASAP-Detect	0.722	0.993	0.98	0.000	0.000	0.000	0.000	0.000
TOEFL	0.352	0.043	0.000	0.000	0.000	0.000	0.000	0.000
Number of important FCs	70	16	91	98	67	46	29	21

Table 22: Evaluation performance on varying datasets (F1 scores) and the number of functional components per epoch. Note that Epoch 0 refers to the pretrained model without additional fine-tuning.

tokens, making a more broadly applicable dataset. Altogether, the dataset creation required 5.5 million tokens of usage on the GPT API.

This AI-augmented corpus offers a novel contribution to the realm of NLP. Its primary function is to serve as resource for evaluating AI-generated writing detectors, specifically for the purposes of analyzing and understanding what is learned by the detection models. As this corpus represents an valuable resource for future NLP research in a rapidly emerging area of study, we have made the generated essays available as well as the construction details such as the prompts and scripts use to create the dataset[4]. However, it is important to note that results from the GPT API may be non-deterministic.

9.4 Experiments

This section presents the methodology employed to investigate the performance and behavior of a fine-tuned RoBERTa based AI writing detection model. The model was trained using pretrained weights from the RoBERTa-base GPT-2 AI detection model (Solaiman et al., 2019) and fine-tuned for this work with the GPT-WIKIPEDIA-INTRO dataset[5].

9.4.1 RoBERTa Base AI Detection Model

The RoBERTa-base OpenAI Detector model (Solaiman et al., 2019) is utilized as the pretrained model in our analysis experiments. Derived from the RoBERTa base model, it has been fine-tuned with the outputs from the GPT-2 model (OpenAI, 2021), a 1.5B-parameter language model LLM. This model was chose as the base for our experiments because, while it was fine-tuned to perform the AI-writing detection task, it was not trained on any data from the LLMs that generated the data used in this work. Furthermore, because of the availability of this model for AI-writing detection, it is relevant to the current discourse on such models. All of the hyperparameters for the model were fixed at their default value.

The GPT-Wikipedia-Intro dataset served as the training data for fine-tuning where the model was trained until it reached well past the point of over-fitting, ensuring that the evolution of the functional components would be fully discernible. Before fine-tuning

[4]Scripts and prompts: ¡REMOVED FOR ANONYMITY¿

[5]Code: ¡REMOVED FOR ANONYMITY¿

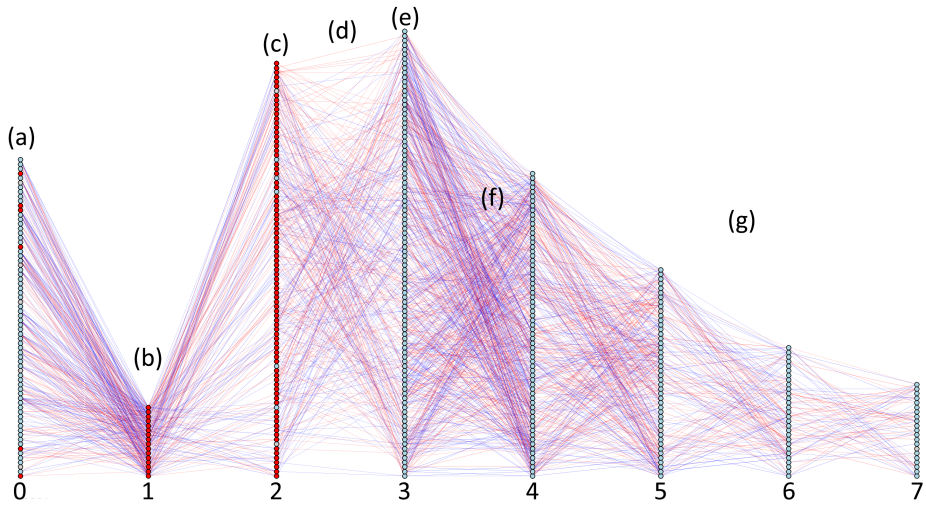


Figure 24: Correlations between important functional components of the neural network over epochs of fine tuning. The strength of the lines represents the strength of the correlations with the positive correlations colored blue and the negative correlations colored red. Nodes in the graph that are filled red correspond to important functional components that correlate strongly with both non-native English essays and AI-generated essays, but not native English essays. The relative variance explained in the model activations by each functional component is represented by the vertical ordering of the nodes, where nodes increase in variance explained as they approach the bottom of a column. Epoch 0 refers to the pretrained model without additional fine-tuning. Letters (a-g) highlight the following features of the figure: (a) important functional components in the pretrained model; (b) the contraction of important functional components in the first epoch of fine-tuning; (c) the expansion of important functional components in the second epoch of fine-tuning; (d) the reorganization of functional components resulting in test performance plummeting; (e) the difference in function of functional components in epochs 3+; (f) the 'X'-like pattern of correlations between functional components in later epochs; and (g) the decaying number of important functional components as the model over-fits.

and after each epoch, the model state was saved for subsequent analysis and the model activations for the held out analysis set were recorded[6].

The functional components were extracted for each time-step of the model using the held out evaluation data from the GPT-Wikipedia-Intro dataset (60,000 documents; 50% human written, 50% AI generated). Only the final embedding of the transformer model was used to compute the functional components and a new ICA model was computed and saved for each time-step. The dimensionality reduction was tuned to explain 99% of the variance of the activations of the model on the analysis set. The importance of each component for each time-step of the model was computed and recorded. A correlation analysis was performed to assess the similarity or dissimilarity between the functional components extracted from different epochs. Significantly correlated ($p < 0.05$) important functional components were graphed to visualize the evolution of the functional components across time.

9.4.2 Correlation of Functional Components with Biased Model Behavior

The experimental analysis was further extended to the two other datasets to assess a known biased behavior of many AI-writing detection models (Liang et al., 2023): a bias toward flagging writing by non-native English speakers as AI-generated content. To perform this experiment, the same ICA models that were fit on the activations from the GPT-WIKIPEDIA-INTRO dataset were utilized on the activations for each time step's model on these additional datasets. The neuron activations were collected and the resulting functional component activations were subsequently concatenated.

Each concatenated functional component activation vector was correlated with a vector that was constructed according to the following rule: 1 if an essay was either human-written from the TOEFL ESSAY dataset, or AI-generated from the AES-ASAP-DETECT dataset. This rule identifies functional components that do not contribute significantly to the differentiation between human and AI-written texts, but rather discriminate between native English speakers' writing and AI-revised essays, and non-native English speakers' writing and LLM-generated texts.

The identified important components that yielded a Pearson's r value greater than 0.5 or less than -0.5 were particularly noted. In the corresponding visualization, these components were highlighted in red to effectively mark their presence to track the propagation of such components through the fine-tuning process.

9.5 Results

In this section, we present the performance and our observations of the AI detection model. The results are structured into two subsections based on whether they are focused on the the model within a single epoch (i.e. a snapshot of the model) or if the results refer to the behavior of the model as it changes across epochs. In the subsection "Model Evaluation by Epoch," we provide the model's performance and highlight the important functional components of the model on a per-epoch basis. On the other hand, the subsection "Inter-Epoch Results" focuses on the interactions and comparisons between

[6]Model weights ($\sim$ 5GB) and activations ($\sim$ 29GB) are available upon request.

different epochs, shedding light on the model's behavior, overfitting tendencies, and dynamics of the important functional components.

9.5.1 Model Evaluation by Epoch

The performance of the model on the three evaluation datasets is presented in Table 22 which provides the F1 score across various epochs. It is notable that, without pretraining, the initial performance (Epoch 0) of the pretrained model varied significantly across the datasets. The model performed best on the GPT-WIKIPEDIA-INTROS dataset with an F1 score of 0.884, followed by the AES-ASAP-DETECT dataset with 0.722, and least effectively on the TOEFL ESSAYS dataset at 0.352 F1 score. These differences make sense as the pretrained model was exposed to Wikipedia style text via the GPT-2-OUTPUT dataset (Solaiman et al., 2019) and the ChatGPT revised TOEFL essays were generated using prompts that explicitly request the model to sound more native (Liang et al., 2023).

Furthermore, the pretrained model displayed 70 important functional components, 6 of which were correlated with both non-native English essays and AI-generated essays, but not native English essays (colored red in Figure 24a).

During the first epoch, the model's performance spiked significantly, achieving nearly perfect F1 scores on the GPT-WIKIPEDIA-INTROS dataset and the AES-ASAP-DETECT dataset with F1 scores of 0.997 and 0.993 respectively. However, it had a sharp falloff on the TOEFL ESSAYS dataset, reaching an F1 score of 0.043. This change in performance coincided with a sharp reduction in the number of important functional components to only 16, all of which correlated significantly to an inability to distinguish non-native English authored essays with AI generated essays (Figure 24b). This pattern continued into epoch 2 with the one change that the number of important functional components dramatically increased to 91 (Figure 24c).

The performance across subsequent epochs showed diverging trends between the in-domain GPT-WIKIPEDIA-INTROS and the other two evaluation datasets. For the GPT-WIKIPEDIA-INTROS dataset, there was a considerable decrease in performance during epoch 3 to 0.679, but a steady increase was observed from epoch 4 on-wards, reaching 0.724 by the termination of training. Contrarily, the performance on AES-ASAP-DETECT and TOEFL ESSAYS datasets dropped to zero at epoch 3 and epoch 2 respectively. The model did not show any improvement for either dataset in the following epochs. The number of important functional components increased to a maximum of 98 at epoch 3 (Figure 24e) which steadily declined until the end of training (Figure 24g).

9.5.2 Epoch-to-Epoch Behavior

During the initial phase of fine-tuning (Figure 24a-b), an higher density of correlations was observed between lower variance-explaining functional components (towards the top of the figure) of the pretrained model and the important functional components of the fine-tuned model. Some of the lower variance-explaining functional components correlate strongly with both non-native English essays and AI-generated essays, but not native English essays, a correlation that exists in all of the important functional

components in the model after one epoch of fine-tuning. These were further propagated to the important functional components of the model at epoch 2, though the correlations are stronger with between the important functional components of the model at epoch 1 and the lower-variance explaining functional components of the model at epoch 2.

Between the second and third epochs, there was a dramatic rearrangement of functional components, visible by the fairly random patter of relatively weak correlations between important functional components of the model at epochs 2 and 3 (Figure 24d). The resulting important functional components share none of the highlighted red correlations with the previous epoch.

From the third to the fourth epochs, an 'X' shaped region of high density of correlations is visible in Figure 24f. This 'X' connects components that explained a low amount of variance to components that explain a high amount of variance across the epochs. While this trend appears to weaken over time along with the general correlation between important functional components in the epochs (Figure 24g), it continued for the rest of the training.

9.6 Discussion

The results of our study provide valuable insights into the transformation of learned functional components in a RoBERTa-based AI-writing detection model during fine-tuning, shedding light on the origin and evolution of biases against non-native English speakers. Firstly, the initial performance of the pretrained model varied significantly across different evaluation datasets. This discrepancy, as discussed previously, can be attributed to the exposure of the pretrained model to the relevant types of text. These variations highlight the importance of creating a new dataset as a counterpoint to the existing TOEFL ESSAYS dataset that has both the properties of being essays written with varying degrees of writing skill and with human written text produced by native English speakers.

During the fine-tuning process, we observed a change in the model's performance and a rapid and substantial change in its important functional components. In the first epoch, the model experienced an improvement in performance on two of the datasets but exhibited a sharp decline on the third. This change coincided with a reduction in the number of important functional components. In addition, we observed correlations between lower variance-explaining functional components of the pretrained model and the important functional components of the fine-tuned model in this initial phase of fine-tuning. These correlations were linked to biases against non-native English speakers by being unable to differentiate them from AI-generated essays while having that ability for essays written by native English speakers. These biases were propagated to the important functional components of the model in subsequent epochs, indicating the persistence and amplification of biases during fine-tuning. This results in a detriment to the generalizable performance of the model, even while appearing to improve the performance on out-of-domain dataset. Furthermore, with fine tuning these large language models, this appears to occur very quickly in the fine tuning process.

Interestingly, the rearrangement of functional components while the model was transitioning to a state of overfitting was observed between the second and third epochs. This was characterized by weak correlations and an immediate drop-off of the biases

functional components that were present in the previous epoch. This may suggests that the model learned dataset artefacts present in the GPT-WIKIPEDIA-INTROS dataset that were not present in either alternate datasets. This is corroborated by the sharp decline of performance on the AES-ASAP-DETECT dataset in the same time-step.

Additionally, an 'X'-shaped region of high density correlations between functional components was identified clearly in the visualization presented in Figure 24 between the third and fourth epochs and more faintly between the subsequent epochs. The 'X' shape emerges from high correlation between low-variance explaining components and those explaining high variance. This is expected if we consider that the model is likely making small hops around its local minimum as it is forced to continue training. The re-odering of the functional components is likely a result of the training during the epoch nudging the model to a different part of the loss landscape near the local minimum and back in the next epoch. The weakening of the correlation between important functional components during the late stages of over-fitting (Figure 24g) may be a result of this occurring with functional components not considered important for a given epoch.

9.7 Conclusion

Throughout this work, we introduced a new AI-detection dataset tailored for student writing, and leveraged it to provide an analysis of the dynamic nature of the functional groups of neurons within a transformer based model during the fine-tuning process. This provided insight into the model's evolution through training and highlights the significance of inherited components from pretrained weights in decision-making processes.

Because these inherited components from pretrained models can rapidly influence down-stream fine-tuned models in ways that can appear helpful in someways while being detrimental in others demonstrate the risk of only having a small number of standard pretrained models that are used for the breadth of tasks for which NLP has proved effective. Much like diverse groups of people bring together varying perspectives to overcome their biases and solve complex problems, we need to avoid a machine learning monoculture. By uncovering the transformation of learned components during fine-tuning and tracking the origin and evolution of biases, another step is made toward increased transparency, fairness, and critical evaluation of LLMs. Future research should explore methods to mitigate these biases by improving the diversity of pretrained models, identifying the functions of a model's learned components, and taking into account the consequences of the dynamics observed in the fine-tuning process.

10 Practical Usage of Neural Pathways

The exploration of neural pathways in machine learning models is not just a theoretical exercise and in this chapter we delve into the pragmatic aspects of pathway analysis. Our focus is on equipping readers with both the knowledge and the tools to dissect and comprehend the neural networks. This chapter is designed as a bridge connecting the theories of neural pathways with their tangible application.

At the heart of this chapter is a detailed user guide for a tool designed specifically for conducting a thorough pathways analysis. The tool, equipped with a graphical interface and advanced analytical capabilities, simplifies the complex task of dissecting neural networks, making it an invaluable asset for anyone in the field of machine learning and potentially enabling even those new to the field to navigate the intricacies of a neural pathways analysis with ease.

Furthermore, the chapter contains commentary on practical considerations when performing a pathways analysis, including tips, best practices, and strategies to optimize the analysis process, ensuring that readers can make the most out of their investigative endeavors.

Lastly, the chapter includes a series of real-world instances where pathway analysis has been successfully implemented. These case studies serve as exemplars of how the method can be applied in various contexts, ranging from academic research to industry applications. They not only illustrate the versatility and utility of pathway analysis but also provide tangible evidence of its impact and effectiveness.

10.1 Neural Pathways Explorer Tool

The Pathways Explorer Tool[7] is an instrument developed to facilitate the expeditious and detailed examination of neural networks. It provides a platform for researchers and practitioners to input attribute tables and neural network activations. Central to this tool is the feature that allows for the extraction of neural pathways with tunable levels of variance explained. This feature affords users the discretion to modulate the depth of their analysis, ranging from a broader overview of neural patterns to a more granular exploration of network behavior. Complementing this is the tool's graphical interface, which visualizes correlation between pathways and attributes. This interface enhances the analytical process by simplifying the identification of complex relationships. Additionally, the tool's functionality extends to presenting exemplar data instances for each identified pathway, coupled with their respective attribute values. This capability is invaluable for contextualizing the pathways in practical scenarios and performing qualitative analysis of the pathways.

In essence, the Pathways Explorer Tool is crafted to bridge the gap between theoretical neural network models and their practical analysis, catering to the community's need for a detailed and accessible examination tool. It is optimized for conducting swift analyses on relatively modest-sized datasets and neural network models. Specifically, it is most effective when applied to datasets comprising fewer than 10,000 instances and models with less than 1,000 neurons. This scope ensures that the tool is sufficiently

[7] https://github.com/jfiacco/NeuralPathwaysEditor

responsive, making it suitable for exploratory analysis and preliminary investigations in both academic and applied settings.

This limitation in scalability, however, is an area of ongoing development, as acknowledged in the future directions. In Section 11.3, we discuss the planned enhancements aimed at expanding the tool's capabilities to handle larger datasets and more complex neural network architectures. The aspiration is to incrementally improve the tool's efficiency and adaptability, thereby extending its applicability to a broader range of research scenarios and larger-scale applications.

The current version of the Pathways Explorer Tool, therefore, serves as a foundational step towards more comprehensive analysis capabilities. It offers a valuable resource for immediate, in-depth exploration of neural pathways within a defined scope, while also laying the groundwork for future advancements in the domain of neural network analysis.

10.2 User Guide

In Chapter 5, we presented the procedure for performing a neural pathways analysis which is to be used as the procedure for performing a pathways analysis. In this section, we provide a guide for using the graphical tool to perform an analysis via the graphical user interface. This tool can be used to quickly analyze small to medium sized neural networks where one has extracted the activations of the network and has a table of attributes. This guide is structured to navigate users through the tool's functionalities, providing practical considerations for the four phases of the analysis process: choosing attributes, determining the number of pathways, determining pathway correlations, and performing a qualitative analysis.

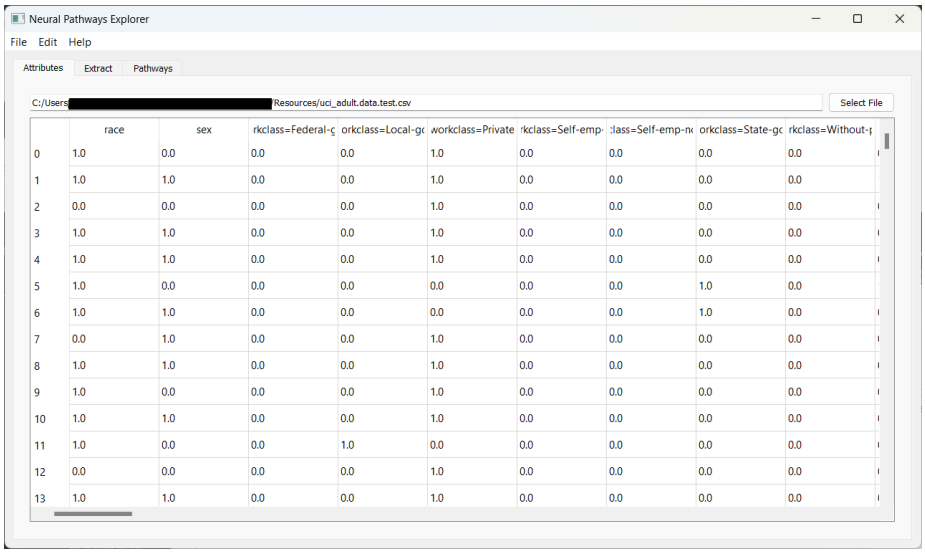
In this initial phase, the focus is on selecting attributes that are potentially significant for the model's decision-making process. This selection is crucial as it influences the subsequent analysis of neural pathways. Attributes can range from input features to more abstract model-specific characteristics. The choice of attributes should be guided by the specific objectives of the analysis and the theoretical underpinnings of the model.

The next phase involves setting a practical limit on the number of neural pathways to be analyzed. The decision is a balance between computational feasibility and the comprehensiveness of the analysis. A higher number of pathways might provide a more detailed picture but at the cost of increased complexity.

In the main analysis phase, the tool helps to uncover how the identified pathways correlate with the chosen attributes. This analysis is pivotal in understanding the interplay between different components of the model. Correlation analysis can reveal insights such as dependencies, redundancies, or unique contributions of specific pathways to the model's behavior.

The final phase involves a qualitative analysis, where the correlations from the previous phases are interpreted and contextualized by projecting them onto the data. This phase allows for a deeper understanding of the model, going beyond mere statistical relationships. It involves examining the pathways in the context of the analysis data to provide an intuition for the findings.

Throughout this guide, we will provide detailed walkthroughs, accompanied by screenshots from the tool, to illustrate each step of the process. By the end of this guide,



	race	sex	rkclass=Federal-g	orkclass=Local-gc	workclass=Private	rkclass=Self-emp-	class=Self-emp-n	orkclass=State-gc	rkclass=Without-p
0	1.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0
1	1.0	1.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0
3	1.0	1.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0
4	1.0	1.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0
5	1.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0
6	1.0	1.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0
7	0.0	1.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0
8	1.0	1.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0
9	1.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0
10	1.0	1.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0
11	1.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0
12	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0
13	1.0	1.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0

Figure 25: Example screenshot of the ATTRIBUTES tab of the Neural Pathways Explorer Tool.

users will be equipped with a thorough understanding of how to leverage this tool to analyze neural pathways in a range of models, leading to more informed interpretations and potentially more robust model development.

10.2.1 Choosing Attributes

This section of the user guide provides instructions on how to load your analysis dataset into the Pathways Explorer Tool and select appropriate attributes for your analysis. The process involves importing a CSV file of attributes and considerations for choosing relevant attributes.

Step 1 - Preparing Your CSV File: Before you begin, ensure that your CSV file is properly formatted. Each row in the file should represent a distinct data instance in your analysis dataset, and each column should correspond to either features of your model or attributes that may be correlated with the neural pathways. It is important that this data is clean and accurately represents the variables of interest for your analysis.

Step 2 - Loading the CSV File into the Tool:

1. *Open the Pathways Analysis Tool*: Launch the application and navigate to the ATTRIBUTES tab (Figure 25).

2. *Import the CSV File*: Look for the option to 'Select File'. Click on this option and navigate to the location of your CSV file on your computer.

3. *Select and Open the File*: Choose the CSV file you prepared and open it. The tool will process the file and load the data.

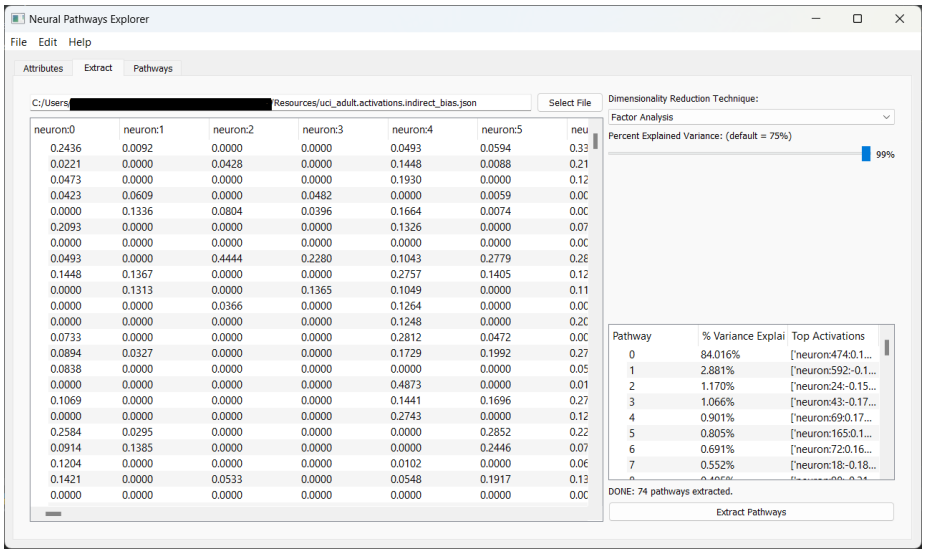


Figure 26: Example screenshot of the EXTRACT tab of the Neural Pathways Explorer Tool. In this case, too many pathways were extracted.

Step 3 - Confirm the Table of Attributes: Once the CSV file is loaded, the tool will display the table of attributes. This table will show all the columns from your CSV file, representing the features and attributes of your analysis dataset. Verify that this table is correct.

Tip - Choosing Attributes: Choosing the right attributes is crucial as it determines the perspective from which you will analyze the model. Attributes should be closely related to the specific task your model is designed to perform and should ideally be independent to provide a clear and unbiased view of the model's behavior. As it is often difficult to find attributes that are truly independent, performing an Independent Component Analysis on the features can group them into maximally independent groups.

Following these steps will successfully load your analysis dataset into the Pathways Explorer Tool and set the stage for a comprehensive analysis of neural pathways based on the attributes relevant to your specific research or application scenario.

10.2.2 Determining the Number of Pathways

This section of the user guide explains how to extract neural pathways from neuron activations using the Pathways Analysis Tool. The process involves loading neuron activations from a JSON file and choosing the appropriate method and parameters for pathway extraction.

Step 1 - Prepare Activation JSON File: The neuron activations should be in a JSON file with the following format:

```
{"<NAME OF LAYER/NEURON SET>":
```

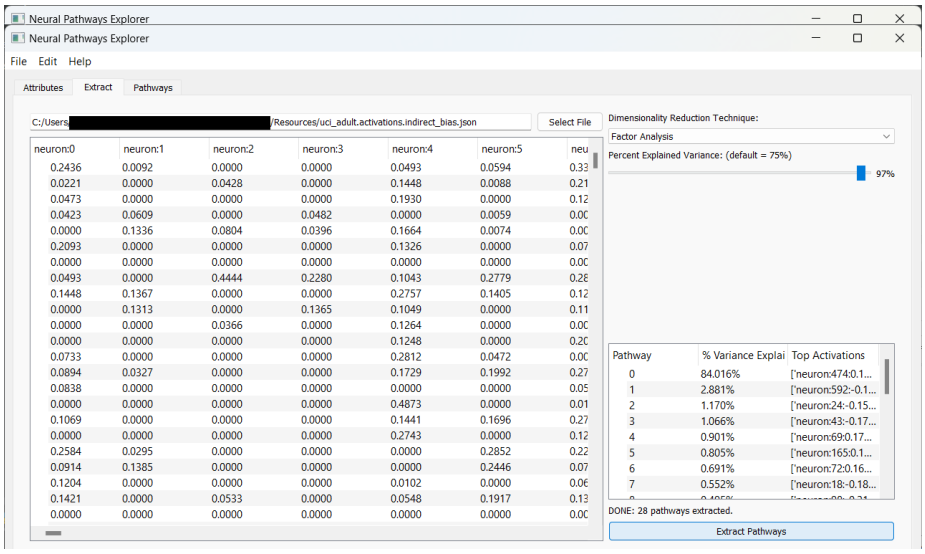


Figure 27: Example screenshot of the EXTRACT tab of the Neural Pathways Explorer Tool. In this case, the extraction parameters were tuned to provide a more reasonable amount of pathways while maintaining a high amount of variance explained.

```
[[<ACTIVATIONS FOR DATA INSTANCE 0>], ...,
 [<ACTIVATIONS FOR DATA INSTANCE N>]], ...}
```

Each key in the JSON object represents a layer or set of neurons, and the associated value is a list of activation values for each data instance.

Step 2 - Loading Neuron Activations:

1. *Navigate to the* EXTRACT *Tab*: Look for a tab or section labeled EXTRACT. Click on this tab to navigate to the pathway extraction section of the tool.

2. *Load the JSON File*: In the EXTRACT tab, find the option to 'Select File'. Select this option and navigate to your prepared JSON file.

3. *Confirm the File Selection*: Choose the JSON file and confirm to upload it. The tool will then process and display the neuron activations.

Step 3 - Choosing the Pathway Extraction Method: Select the pathway extraction method with the dropdown menu. The default method for pathway extraction is Factor Analysis. An alternative option available is Principal Component Analysis (PCA). Choose the method that best suits your analysis needs; factor analysis generally provides better quality pathways, though PCA is often faster for larger datasets or models.

Step 4 - Setting the Target Percent of Variance:

1. Determine the Target Percent of Variance: Decide on the percentage of variance that should be explained by the pathways. This is a crucial decision, as it affects

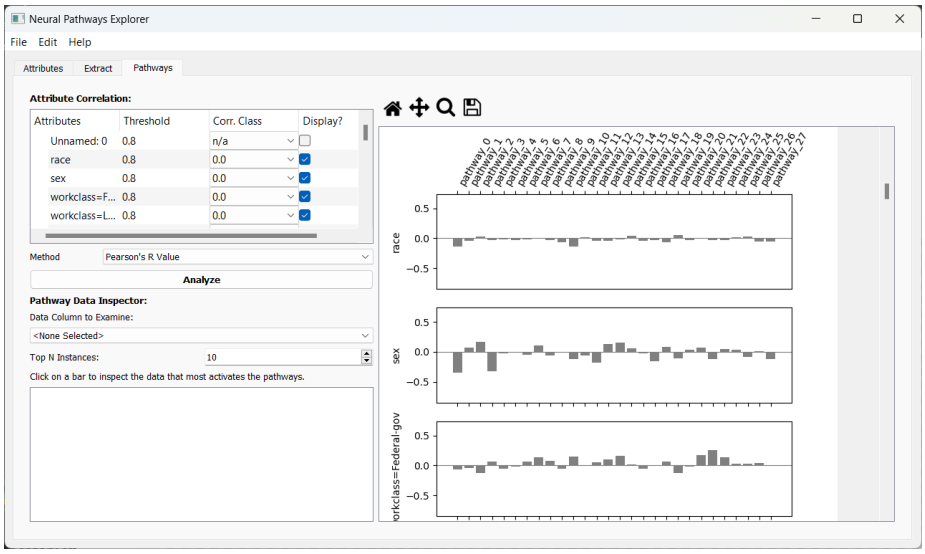


Figure 28: Example screenshot of the PATHWAYS tab of the Neural Pathways Explorer Tool after the 'Analyze' button has been clicked.

the complexity and quantity of the pathways extracted. A higher percentage means less information loss but results in more complex and numerous pathways.

2. Input the Target Percent of Variance: In the tool, locate the option to set the target percent of variance. Enter the value you have determined based on your analysis needs.

Tip - Determining the Number of Pathways As a guideline, it is recommended to aim for a percent variance that yields approximately one-tenth the number of pathways as there are neurons in your model. This ratio is suggested as a starting point and can be adjusted based on the specific requirements of your task.

Step 4 - Extracting Pathways: Once all settings are confirmed, proceed to extract the pathways by clicking the 'Extract Pathways' button. The tool will process the neuron activations using your specified method and variance target, resulting in a set of neural pathways for further analysis. The number of pathways and each of their percent variance explained will be displayed above the 'Extract' button (Figures 26 and 27). The percent variance explained and the method for extraction can be changed after extraction, but you must use the 'Extract' button to extract pathways with the new settings.

10.2.3 Determining Pathway Correlations

This section provides instructions on how to analyze the correlations between extracted pathways and loaded attributes using the Pathways Analysis Tool. The process involves selecting attributes for correlation computations, choosing a correlation method, and interpreting the results through graphical representations.

Step 1 - Navigating to the Pathways Tab:

1. *Verify Attributes and Pathways*: Before proceeding with this section, attributes must be loaded and pathways must be extracted.
2. *Locate the Pathways Tab*: Look for a tab or section labeled PATHWAYS. Click on this tab to access the correlation analysis section.

Step 2 - Selecting Attributes for Correlation:

1. *Review the Attributes Table*: In the top left section under the Pathways tab, you will find a table populated with attributes and features from the ATTRIBUTES tab.
2. *Toggle Attributes*: Next to each attribute in the table, there is a checkbox. By toggling the checkbox, you can include or exclude that attribute from the correlation computations.
3. *Confirm Your Selections*: Ensure that checkboxes are checked for all attributes you wish to analyze, and unchecked for those you want to exclude.

Step 3 - Choosing the Correlation Method: By default, the tool uses Pearson's R value for correlation. An alternative option available is Logistic Regression, where correlations reflect the weights learned by a logistic regression model trained to predict the attribute class with the pathways as inputs. For most cases, the default Pearson's R value is recommended. However, choose the method that aligns best with your analysis needs.

Step 4 - Analyzing the Correlations:

1. *Initiate the Analysis*: Click on the 'Analyze' button. The tool will compute correlations between each attribute and each pathway.
2. *View the Results*: The correlations will be displayed in bar graphs (Figure 28), with each graph representing an attribute. Within each graph, individual bars represent the correlation of a pathway with that attribute.

Tip - Interpreting the Bars: Bars that represent correlation above a certain threshold are highlighted for convenience. In many practical scenarios, a Pearson's correlation greater than 0.3 is generally indicative of a pattern that is qualitatively discernible in the data. The graphical representation of correlations provides a clear and intuitive understanding of how different pathways relate to each attribute. The highlight feature on the bars assists in quickly identifying significant correlations, streamlining the process of pinpointing relevant pathways for further investigation.

10.2.4 Qualitative Analysis

This section of the user guide describes the process of conducting a qualitative analysis on the pathways using the Pathways Analysis Tool. This analysis involves interacting with the correlation bar graphs to explore the data instances most associated with specific pathways and to understand the connection between these pathways and attributes.

Step 1 - Interacting with the Correlation Bar Graphs:

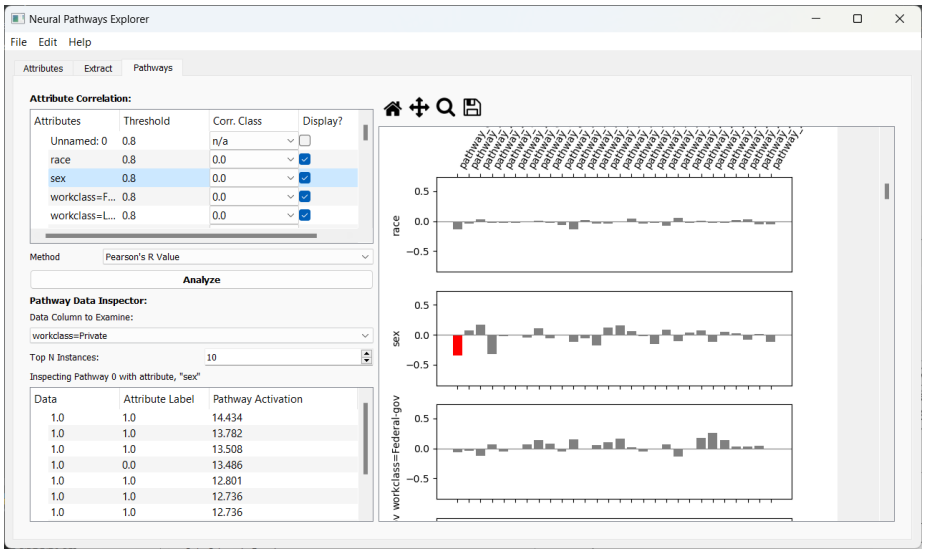


Figure 29: Example screenshot of the PATHWAYS tab of the Neural Pathways Explorer Tool after the correlation bar has been selected.

1. *Locate the Correlation Bar Graphs*: Ensure you are in the PATHWAYS tab where the correlation bar graphs are displayed following your analysis.

2. *Select a Pathway for Analysis*: Click on one of the bars in the correlation bar graphs. The bar you select represents a specific pathway and its correlation with an attribute.

Upon clicking, the selected bar will be highlighted, indicating it is the focus of your qualitative analysis (Figure 29).

Step 2 - Viewing Data Instances Related to the Selected Pathway:

1. *Examine the Bottom Left Table*: Once a bar is selected, look at the table located in the bottom left of the window. This table will automatically populate with data instances.

2. *Review the Displayed Data*: The data instances shown are those that most activate the pathway represented by the clicked-on bar.

3. *Note the Attribute Labels*: Alongside each data instance, the label of the attribute for that instance, corresponding to the attribute of the selected graph, will also be displayed.

Tip - Customizing the Data Display:

1. *Choose a Column from the Attribute Table*: For tasks involving textual data, it is recommended to display the raw text of the data instances. Select a column from the attribute table that you wish to view via the dropdown menu.

2. *Adjust the Number of Displayed Instances*: The tool allows you to change the number of data instances shown in the table. Depending on the complexity of the task and the strength of the correlation, you may need to adjust this number. More instances can help in discerning patterns, especially in cases of weaker correlations or more complex tasks.

Step 3 - Verifying Connections Between Pathways and Attributes: This qualitative analysis provides an opportunity to validate the quantitative findings from the correlation analysis and to gain a deeper, more nuanced understanding of the relationships within the model.

1. *Analyze the Data*: Use the information in the table to observe and analyze how the most activated data instances correlate with the selected pathway and its associated attribute.

2. *Confirm Relationships*: This step allows you to quickly verify the connections between specific pathways and attributes. Look for patterns or trends in the data that support the correlation indicated by the bar graph.

10.3 Illustrative Use Cases

This section transitions catalogues some practical applications of the neural pathways approach. We present three distinct instances where researchers, unaffiliated with the tool's design and implementation, have successfully utilized the pathways method to advance their work. These cases are drawn from diverse fields, showcasing the versatility and impact of the tool in real-world scenarios. Each use case not only exemplifies the practical utility of the pathways method but also illustrates how it can be adapted to address various challenges and objectives in the realm of artificial intelligence.

The first case delves into the field of AI education, where the tool was employed to enhance the learning experience of students in understanding neural network models. By providing a transparent view of neural pathways, the tool enabled educators to offer a more intuitive and interactive approach to teaching complex AI concepts. The second instance discusses its use in neural model debugging, highlighting how researchers leveraged the tool to identify and rectify inefficiencies in model architectures. This application underscores the tool's diagnostic capabilities, crucial for optimizing model performance. The final case focuses on bias detection in datasets, an increasingly pertinent issue in AI ethics. Researchers used the pathways method to uncover and address biases in training data, ensuring fairer and more equitable model outcomes. Each of these cases provides valuable insights into the practical benefits and adaptability of the pathways method across different areas of AI research and application.

In implementing these practical applications, our approach was deliberate and methodical. We selected researchers who had a specific and relevant context for using the neural pathways tool, ensuring that its application would be meaningful and challenging. Guiding these researchers through the application process, we provided support and insight into how the tool could be best utilized in their unique scenarios. This collaboration was a two-way street; as they applied the neural pathways approach to their work, we engaged in detailed discussions about their experiences and the conclusions they drew

from using the tool. These interactions were invaluable, as they not only allowed us to observe the tool's application in diverse fields but also to gather a range of user feedback and perspectives. From these engagements, we were able to extract key lessons and insights. These ranged from understanding the tool's versatility and user-friendliness in different contexts, to identifying areas for improvement and enhancement. This process was not just about validating the utility of the neural pathways approach but also about learning from its application in real-world scenarios, thereby enriching its development and refining its effectiveness for future users.

10.3.1 AI Education

The ongoing development of neural pathways as a teaching tool in AI education represents a significant expansion upon the work established by Chao et al. (2023). This curriculum module, called StoryQ, was originally designed for high school English/Language Arts classes and implemented through a web-based platform. It laid the groundwork for integrating AI concepts into secondary education and is currently being expanded to cover material in concurrent Social Studies and Math classes as well. As students engage with tasks like intent recognition and sentiment analysis, they gain first-hand experience with machine-learning language models. In its original formulation, the curriculum focused on logistic regression as a relatively simple model for students to understand. The ubiquity of neural networks and now generative models has applied pressure to include neural network concepts into this curriculum despite their complexity. And so, one part of the next phase of this educational design involves integrating the neural pathways approach to deepen students' comprehension of these complex models.

The neural pathways method aims to create a more intuitive bridge between advanced neural models and simpler logistic regression models. This approach is particularly beneficial in making AI concepts accessible and relatable to high school students. By visualizing the influence of input features on model outcomes, the pathways approach provides a clear and interactive means of understanding how neural networks process and interpret data. This visual mapping not only clarifies the mechanics of AI but also enriches students' perception of language as a multifaceted tool beyond simple communication. The positive feedback from the initial StoryQ module has set a promising precedent, and the integration of neural pathways analysis is poised to further enhance the educational value of the program. This initiative continues to evolve with revisions and further testing planned for Fall 2022.

In this next phase of educational design, the integration of the neural pathways approach into the curriculum is not a one-sided process; it thrives on the collaborative and iterative engagement between developers, educators, and students. This communication the curriculum designers has tailored and simplified the concepts of the neural pathways tool to suit the educational needs of high school students. As the curriculum gets tested in the field, educators can experiment with incorporating this tool into their lessons, where their feedback and observations become crucial in refining its design and functionality. Similarly, students' interactions with the tool will offer direct insights into how effectively the concepts are being communicated and comprehended. This ongoing dialogue can be instrumental in evolving the tool to be more user-friendly and

relevant to the curriculum. The process is dynamic, with each iteration informed by the experiences and perspectives of both teachers and learners. This collaborative effort not only ensures that the tool is pedagogically sound but also aligns it closely with the educational objectives of the StoryQ module. As a result, the neural pathways method is further adapted and refined, reflecting a deepening understanding of how to convey complex AI concepts in an educational setting.

10.3.2 Model Debugging

In recent unpublished work done by Nourbakhsh (2024), researchers conducted a detailed pathways analysis on LayoutLMv3 (Huang et al., 2022), using the FUNSD dataset (Jaume et al., 2019) as a basis for examination. The FUNSD dataset, known for its wide variety in document appearance and the complexity of form understanding tasks it presents, offered an ideal setting for this analysis. The primary objective was to ascertain how different modalities, particularly visual and spatial characteristics, influenced the model's decision-making processes. This involved a qualitative analysis of pathway activations across various document samples, aiming to identify whether documents with distinct features, such as a tabular structure, triggered different neural pathways. The initial findings revealed intriguing patterns in pathway activations, shedding light on how the model processes diverse document layouts.

Further deepening the analysis, clustering was performed on the pathways, drawing random samples from each cluster to uncover potential similarities within these groups. A standout observation was the clustering of documents based on textual density, indicating that LayoutLMv3 tended to group denser pages together, as showcased on one of the presentation slides. This clustering analysis was complemented by a comparative study across different versions of the LayoutLM model (versions 1, 2, and 3) and the basic RoBERTa embeddings, which lack multimodal information. The Adjusted Rand Index (ARI) scores obtained in this comparison revealed a gradual increase in reliance on multimodal features from LayoutLMv1 to v3. However, a critical insight was that despite these advancements, the text modality still predominantly influenced the model's semantics across all versions. This imbalance led to a subsequent investigation into the parameter norms of LayoutLMv2 and v3, revealing a significant number of dormant parameters in v2. These findings not only highlighted areas for model optimization but also underscored the nuanced role of different modalities in document understanding models. Through this multifaceted pathways analysis, the researchers were able to diagnose and pinpoint specific areas for improvement, demonstrating the tool's utility in neural model debugging and enhancement.

The research conducted by Nourbakhsh (2024) on LayoutLMv3 not only provided insights into the model's functioning but also exemplified a collaborative, iterative process between the interpretability work and the neural model design. As the neural model designers uncovered patterns and anomalies in the pathway activations, these findings and limitations of the method were shared with the interpretability researchers, fostering a dynamic dialogue. This exchange allowed the designers to gain a deeper perspective on how their model was being interpreted and utilized in real-world scenarios. It also provided them with valuable feedback on areas needing refinement, directly influencing subsequent iterations of the LayoutLM model. The work also revealed potential

scalability issues that will be discussed in the Future Work section of Chapter 11. This collaborative process ensured that the insights gained from the pathways analysis were not just theoretical observations but were actively used to enhance the model's design and functionality. As a result, this ongoing interaction between the researchers and the model designers became a critical component of the research, leading to more informed and effective model enhancements.

10.3.3 Dataset Analysis

The qualitative analysis of datasets using the pathways method, as implemented by Adamson (2023), offers a compelling illustration of how this approach can uncover insights into dataset characteristics. Provided with extracted pathways a qualitative analysis was performed, which sought to understand the functional significance of these feature groups. One of the initial concerns raised was the apparent association of many model pathways with demographic features. This observation posed potential ethical concerns about the influence of demographic factors on model decisions.

By examining the data that was activating the pathways identified by the higher-level analysis, it was revealed that what initially appeared as demographic bias was, in fact, more attributable to topic-specific patterns linked to certain schools. This was a direct result of the sampling approach used to create the dataset, which, while intending to provide a de-biased sample, introduced a different form of potential bias. This discovery exemplifies the importance of looking beyond surface-level data interpretations, especially in models where complex feature interactions are at play. By employing the pathways method, researchers and analysts can untangle intricate relationships within the data, moving from a preliminary understanding of feature groupings to a more profound comprehension of their practical implications. Such an approach is essential not only for refining model accuracy but also for ensuring ethical and responsible AI applications, particularly in areas where data sensitivity and integrity are paramount.

The qualitative analysis conducted by Adamson (2023) exemplifies the crucial collaborative relationship between dataset creators and interpretability researchers in the field of AI. In this instance, the interpretability method revealed a need to group non-separable features, see Section 5.4. This insight was pivotal in providing a robust interpretability tool that can handle the complex interactions within the dataset. Additionally, for the dataset creators, the feedback provided a valuable perspective on the unintended consequences of their sampling approach, highlighting areas where the dataset might inadvertently introduce biases. This two-way street of communication and collaboration led to a more comprehensive understanding of both the dataset's structure and the interpretability tool's capabilities.

10.4 Conclusion

This chapter, encompassing both the user guide for the Pathways Explorer Tool and the illustrative use cases, underscores the versatility and impact of the pathways method in diverse AI applications. From enhancing AI education and facilitating neural model debugging to ensuring ethical dataset analysis, the tool demonstrates its capability to render complex neural network behaviors into understandable and actionable insights.

By bridging the gap between theoretical AI concepts and practical implementations, the pathways method empowers users across various domains to delve deeper into the workings of AI models. As we have seen through the examples of StoryQ in education, LayoutLMv3 in model debugging, and automated essay scroing in dataset analysis, the pathways approach is not just a tool for analysis but a catalyst for innovation, ethical considerations, and advanced learning in the ever-evolving field of artificial intelligence. This chapter, therefore, serves as both a guide and an inspiration for future explorations and applications of neural pathway analysis.

11 Conclusion and Future Directions

11.1 On Changes in Neural Model Interpretablity Over the Lifetime of this Work

The field of neural network interpretability emerged to seek methods to explain model decisions, unveil representations within hidden layers, quantify uncertainty, and enable auditing by domain experts. Since the work that led to this thesis began, there has been monumental progress in the advancement of deep learning and neural network interpretability. In this work we have discussed a method for interpretability that focuses on a specific subset of the body of work, specifically in the post-hoc analysis of models that range from simple feedforward neural networks to transformer models. We have covered much of the work that has served as the foundation for this thesis in Chapter 2. However, concurrent to the work done in this thesis, researchers have introduced new techniques to elucidate model mechanics, evaluate explanation quality, expand scope to new data types, and embed trust and accountability within new models. The advent of large language models has also presented new challenges around decoding their knowledge and scaling explanations appropriately. This section chronicles some of the salient technical advances, philosophical shifts, and changing priorities within the interpretability field over the recent years. We survey rising approaches extracting post-hoc explanations, inherent techniques exposing model representations, metrics gauging quality, transition towards real-world impact, and outlook for demystifying ever-larger neural networks. In terms of the taxonomy from Chapter 2, a migration to a higher abstraction interpretability space is occurring as a response to larger and larger models being produced and deployed along with expanding and codifying the evaluation of such models.

The landscape of neural network interpretability research has evolved rapidly. Early interpretability methods focused primarily on assigning importance scores to input features (e.g. pixels for images or words for text). Approaches such as saliency maps, gradient-based localization, and intrinsic influence functions proliferated from 2015 to 2018 (Simonyan et al., 2013; **?**). However, these methods had several critical limitations - they failed to explain model reasoning, focused only on inputs, and were susceptible to adversarial attacks (Ghorbani et al., 2019).

The rigor of evaluation for explanation methods has matured. Properties such as robustness, fidelity, stability, and ground truth alignment are now quantified. Benchmark datasets specifically for testing interpretations have been released across vision, text, and tabular data (Jacovi et al., 2021). And visualization tools for interactive explanation analysis continue advancing (Ghani et al., 2023). Overall, recent advances now enable deeper insight into model reasoning, behavior across data distributions, failure modes, and alignment with human conceptualizations.

Alongside technical advances, the philosophical foundations and research priorities of the interpretability field have matured. As such, new lines of critique and analysis have developed. Researchers examine the assumptions, efficacy, and real-world impacts of explanations through an interdisciplinary lens spanning social science, psychology, and ethics (Miller, 2019). Trade-offs are weighed between accuracy, efficiency, and transparency. And interactive interfaces now enable non-technical stakeholders to

scrutinize explanations as well. Moving forward, calls continue growing to shift some research efforts away from technical novelty towards applications benefitting society - improving medicine, governance, criminal justice, accessibility, and more (Jacovi and Goldberg, 2020). This philosophical strand strives to unify scientific rigor with ethical responsibility - illuminating the true capacities and limitations of advanced AI through the language of detailed, personalized explanations. The next wave of neural network transparency will demand even greater integration of technical, social, and moral considerations.

Particularly, the next wave of neural network transparency will need to handle the advent of large pre-trained language models (many of which are based on a transformer architecture). These models have dramatically reshaped the neural network interpretability field over the past few years. Models such as BERT, GPT-3+, and PaLM have demonstrated impressive performance on language tasks once considered exceptionally human-like. However, their sheer scale - with hundreds of billions of parameters - poses massive challenges for explanation techniques. Attention layers now play a central role in unraveling their behavior, though, despite their name are an unreliable indicator for what the model is prioritizing (Grimsley et al., 2020; Chefer et al., 2021). Providing faithful explanations efficient enough to scale appropriately and quantifying how well these interpretations generalize across tasks remains an open problem. Researchers are only beginning to unpack the reasoning behind decisions in models like these. Promising directions include grounded LLM based information retrieval (Zhu et al., 2023), adversarial probing of knowledge representations(Kumar et al., 2023), and designing simplified performance-matched Networks amenable to explanation (Zheng et al., 2022). What representations these models learn, how they employ contextual knowledge, and whether their decision-making aligns with human rationales remain active debates within both the ML fairness and interpretability communities.

This raises the question, where does the neural pathways approach belong amongst this change in the field? Despite being founded on pre-LLM interpretability considerations, it can serve as an essential foundational layer that complements more recent and abstract interpretability techniques, such as those utilizing LLMs for model explanations. By providing empirical insights into the subset of the neural networks implicated in model decisions, especially in complex models like BERT or GPT-3+, it can anchor the narrative-like explanations generated by LLMs in concrete data-driven analysis. This approach ensures that high-level, human-readable interpretations are rooted in the actual computational behavior of the model, thus enhancing the accuracy and credibility of interpretations in AI. The integration of the neural pathways method with abstract interpretability techniques represents a balanced approach, combining in-depth technical analysis with accessible explanations.

In review, neural network interpretability research has rapidly evolved in recent years - both technically and philosophically. Explanation methods now provide multilayered insights into model mechanics and knowledge representations beyond mere input attribution. Rigor of evaluation continues improving to align with enhanced aims instilling trust, auditing systems, and ensuring fairness. The rise of transformers has unlocked unprecedented predictive power yet poses new challenges for explanation techniques to scale appropriately and unpack these massive models' reasoning, and the neural pathways approach presented in this thesis fills a specific niche between the

abstract concepts learned by neural models and the inscrutable neurons.

11.2 Summary of Contributions

The contributions of this thesis can be summarized as six specific items, listed below with the chapters that principally address each contribution.

1. **Method for Neural Model Interpretability at the Subtask Level (Chapters 3, 4, & 5):**

   The core contribution for this thesis is the introduction and validation of the neural pathways interpretability approach. Through this method, we show that we can analyze a neural model by examining groups of correlated neurons instead of individual neurons. This allows us to align the activations within a neural model with external knowledge that is too abstract to be represented by a single neuron activation.

2. **Framework for Aligning Learned Functions to Causal Structures (Chapters 4 & 5):**

   The second contribution of this work is an extension of the first where we show the we can align a limited type of causal information with pathways that a neural model has learned. We developed a metric that can summarize how closely aligned a neural model is with a causal model. This metric is conceptually based on precision and recall adapted for the alignment between pathways and casual graphs.

3. **Method for Comparing and Aligning Functional Components Across Models (Chapters 3 & 9):**

   We also contribute a extension of the neural pathways approach to align and compare learned functions between different models. This can include models of differing architectures or training procedures. This can allow for mapping the evolution of learned features and functions throughout the training process.

4. **Scaffolding Neural Model Learning with Functional Components (Chapter 7):**

   Applying the neural pathways approach to the neural model design process, we find that we can use fixed high level pathways as auxiliary features for a rhetorical structure parser that allows the parser to more easily handle student writing. Student writing is a difficult domain for discourse parsing as there is a wide variance of author writing ability and essay structure within the data.

5. **Employing Functional Component Transferability Through Fine-Tuning (Chapter 8):**

   We demonstrate the ability of the neural pathways approach for error analysis by analyzing the pathways correlated with the error cases for detecting transactivity in forum posts by Masters students. This is a different domain than was used to

train the original model for detecting transactivity. We further discuss the use of neural pathways approach to understand how the Recognizing Textual Entailment pretraining task informs the transactivity model providing insight on the specific reasons why there is such a dramatic difference in performance of the transactivity detection algorithm depending on the domain.

6. **Method to Describe Emergence of Problematic Features During the Fine-Tuning of Neural Models (Chapter 9):**

   Our last contribution explores how the neural pathways approach can be used to facilitate an analysis of the fine-tuning process for neural models. The target domain for this study coincides with a wide initiative in the field of interpretablity to improve fairness by identifying and reducing harmful bias. We show that neural pathways can be used effectively to determine how sensitive information may be reinforced during fine-tuning. This extends the concept of the meta-pathways approach beyond model comparison to mapping training process of a model over time.

11.3 Future Directions

Grounded in the comprehensive insights garnered from the research presented in the preceding chapters, this portion of the thesis seeks to illuminate potential avenues of exploration, challenges yet to be addressed, and novel opportunities that may arise in the both the near the far term. We postulate on the broader implications of this line of research and consider the wider impact. Furthermore, we posit the transformative potential of neural pathway interpretability tools, consider the ethical and societal ramifications of our innovations, and envisage a future where AI systems harmoniously collaborate with humans in two way dialogue.

11.3.1 Expanding on Neural Pathways

Domain Expansion *(Near-Term)***:** The exploration of neural pathways has been primarily situated within the context of educational technologies. This focus, while rich and instructive, merely represents the tip of the iceberg. When we consider the vast landscape of modern sectors, from the intricate diagnostics in healthcare to the nuanced decision-making in finance, the potential applications of neural pathway research can be multifarious. For instance, within the healthcare realm, understanding neural pathways can pave the way for transparent diagnostic AI tools, ensuring that medical professionals are equipped with both the predictions and the rationale behind them. Similarly, in the world of finance, deciphering the decision-making process of neural models could bolster trust in algorithmic trading or less structural bias in credit risk assessment. The overarching objective, across all these domains, remains consistent: to demystify the intricacies of neural models, thereby fostering transparency, trust, and widespread adoption.

Neural Pathways in Transformers and Large Language Models *(Near-Term)***:** The ascendance of transformer-based large language model architectures, epitomized by

models such as GPT-4 and LLAMA, have redefined the contours of state-of-the-art machine learning. These architectures, characterized by their ability to capture intricate contextual relationships, have become the bedrock of numerous applications. Yet, as our research indicates, a deeper probe into the application of neural pathways within transformers remains a largely uncharted territory. The extant methodologies have predominantly focused on the final embeddings, however, to truly harness the power of transformers, it would be worthwhile to explore the application of the neural pathways approach to the sequence-based structure inherent in their design. By doing so, we may better understand the computations and dependencies that underpin their predictions, thus offering a more comprehensive and granular understanding of their operations.

Pathway Pruning *(Long-Term)***:** The complexity of neural models enables them to capture nuanced patterns, but it also poses challenges in terms of computational efficiency and interpretability. As our understanding of neural pathways matures, a tantalizing prospect emerges: the possibility of pathway pruning. Pathway pruning is the process of identifying and excising redundant or non-essential pathways (or even fairness related problematic pathways) so that neural models that are leaner yet equally potent can be sculpted. Such pruning not only could result in significant computational savings but also aligns with the broader ethos of parsimony in computational modeling, an emerging concern tied to the environmental sustainability of ultra-large models. A well-pruned model, stripped of extraneous elements, can offer clearer insights into its decision-making process, marrying efficiency with transparency.

11.3.2 Enhancing Neural Pathway Interpretability Tools

User-Friendly Visualization Tools *(Near-Term)***:** Neural pathways, like any interpretability method, poses an inevitable challenge: accessibility to the those domain experts without a background in machine learning. To democratize the insights obtained from our research and to truly foster a broad-based understanding, there is an imperative need for intuitive visualization tools. Such tools serve as bridges, translating the abstract mathematical intricacies into tangible, comprehensible visuals that intuitively convey the logic behind a model's decision-making. Our preliminary endeavors[8] have resulted in the early stages of such a tool. However, as the domain of neural pathways continues to evolve and as we glean newer insights into their application, there is an undeniable need to iteratively refine and expand the visualization repertoire. Ultimately, the goal is to cultivate a toolset that evolves in tandem with the field, ensuring that the revelations of neural pathways remain accessible to all.

Automated Neural Pathway Analysis *(Long-Term)***:** In an age where rapid iteration is not just a luxury but a necessity, the neural pathways method, while a tool that can expedite and streamline its analytical processes, beckons for a system of automated neural pathway analysis. Such a system could autonomously identify and elucidate dominant neural pathways for a diverse array of tasks. Beyond mere identification, the true potency of this automation lies in its potential to offer hypotheses for reasonable explanatory structures that can be aligned with the internal reasoning of a neural model.

[8] https://github.com/jfiacco/NeuralPathwaysEditor

11.3.3 Ethical and Societal Implications

Bias Detection and Mitigation *(Near-Term)*: The proliferation of machine learning models in pivotal societal sectors, such as judicial and hiring mechanisms, accentuates the necessity to proactively discern and rectify biases. The inherent opacity of these models can often cloak deeply ingrained biases, making them inadvertent instruments of systemic prejudices; this is a known problem. However, with the lens of neural pathways, we possess an additional vantage point to dissect and diagnose these biases at their source. By understanding the specific pathways that culminate in biased outcomes, we may be able to orchestrate targeted interventions, recalibrating these pathways towards fairness. This not only ensures that our algorithms serve as enablers of equity but also underscores the profound responsibility we hold as stewards of these potent technological advancements.

Ethics in AI Curriculum *(Long-Term)*: As the realm of artificial intelligence continues to unfold and integrate into our daily lives, it is vital that the next generation is introduced to both its functions and limitations. Introducing students to the underlying principles of neural networks and machine learning is a critical goal for technological literacy. This foundational knowledge, coupled with real-world examples, can foster a balanced perspective, instilling both an understanding of the capabilities of AI and a healthy skepticism towards its limitations. By making these topics accessible and engaging for younger minds, we pave the way for a future population that not only benefits from AI-driven advancements but also critically engages with them.

11.3.4 Collaborative Scientific Progress with AI Systems

Human-AI Academic Dialogue *(Near-Term)*: The intersection of human domain expertise and the computational abilities of neural networks suggests a vibrant academic discourse, not as separate entities but as co-contributors to a shared body of knowledge. Human researchers, equipped with a nuanced understanding of specific domains, can impart this knowledge to guide and refine machine learning algorithms. Additionally, as these algorithms process vast datasets, they can unearth patterns and insights previously imperceptible to human analysis. Through neural pathways, we may be able to recover these insights and such revelations can, in turn, augment the prevailing domain knowledge, fostering a symbiotic loop. This iterative exchange signifies a paradigm where humans and AI partake in a dynamic academic dialogue, advancing the frontier of knowledge.

Causal Inference in Diverse AI Domains *(Near-Term)*: The potential for ideas from causal inference to augment the neural pathways approach extends beyond the work performed for this thesis. The infusion of causal methodologies into neural pathways presents a transformative avenue for decoding the complexities of neural network architectures. Instead of merely mapping relationships and associations between the learned functions, the causal perspective can allow us to ask questions about the causal chain of functions that the network uses to the transform its input into its output. By doing so, we transition from a surface-level understanding to an in-depth comprehension. A deep dive into the causal structures within neural networks holds the promise of enhancing their interpretability manifold, ultimately leading to better interpretability

methods.

11.4 Final Remarks

Throughout this dissertation, we have explored the concept of neural pathways as a means to improve the interpretability of neural network models. This approach has been applied across various applications, highlighting its potential in making complex models more transparent and understandable, especially for users who are not experts in machine learning. By focusing on the functional components of neural models, we have provided a new lens through which these systems can be examined and understood. However, it's important to recognize that this is an initial step in a much larger journey towards fully demystifying neural networks.

Looking ahead, the ambition is to further refine and expand the neural pathways approach. The goal is to make it more adaptable and applicable to a wider range of neural network architectures, including the increasingly complex large pre-trained models. We hope this work will contribute to the broader effort of making AI systems more transparent and accountable, not only for researchers but also for the general public who interact with these systems in their daily lives. Ultimately, the aspiration is for these efforts to lead to AI systems that are not only powerful and efficient but also trustworthy and understandable, aligning with ethical standards and societal needs.

Acknowledgements

This work was supported in part by NSF grants DRL-1949110, ACI-1443068, IIS-1546393, and IIS-1822831; and funding from the Schmidt foundation.

References

Aaditya Bhat. Gpt-wiki-intro (revision 0e458f5), 2023. URL https://huggingface.co/datasets/aadityaubhat/GPT-wiki-intro.

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL https://www.tensorflow.org/. Software available from tensorflow.org.

Abubakar Abid, Mert Yuksekgonul, and James Zou. Meaningfully debugging model mistakes using conceptual counterfactual explanations. In *International Conference on Machine Learning*, pages 66–88. PMLR, 2022.

David Adamson, April 2023.

Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. Sanity checks for saliency maps. *Advances in neural information processing systems*, 31, 2018.

Julius Adebayo, Michael Muelly, Ilaria Liccardi, and Been Kim. Debugging tests for model explanations. *Advances in Neural Information Processing Systems*, 33: 700–712, 2020.

David Ifeoluwa Adelani, Haotian Mai, Fuming Fang, Huy H Nguyen, Junichi Yamagishi, and Isao Echizen. Generating sentiment-preserving fake online reviews using neural language models and their human-and machine-based detection. In *Advanced Information Networking and Applications: Proceedings of the 34th International Conference on Advanced Information Networking and Applications (AINA-2020)*, pages 1341–1354. Springer, 2020.

Yossi Adi, Einat Kermany, Yonatan Belinkov, Ofer Lavi, and Yoav Goldberg. Fine-grained analysis of sentence embeddings using auxiliary prediction tasks. *arXiv preprint arXiv:1608.04207*, 2016.

Atish Agarwala, Abhimanyu Das, Brendan Juba, Rina Panigrahy, Vatsal Sharan, Xin Wang, and Qiuyi Zhang. One network fits all? modular versus monolithic task

formulations in neural networks. In *Proceedings of the Ninth International Conference on Learning Representations (ICLR 2021)*, 2021.

Hongjoon Ahn, Sungmin Cha, Donggyu Lee, and Taesup Moon. Uncertainty-based continual learning with adaptive regularization. *Advances in neural information processing systems*, 32, 2019.

Hua Ai, Marietta Sionti, Yi-Chia Wang, and Carolyn Penstein Rosé. Finding transactive contributions in whole group classroom discussions. In *Proceedings of the 9th International Conference of the Learning Sciences-Volume 1*, pages 976–983. International Society of the Learning Sciences, 2010.

Zeynep Akata, Lisa Anne Hendricks, Stephan Alaniz, and Trevor Darrell. Generating post-hoc rationales of deep visual classification decisions. *Explainable and Interpretable Models in Computer Vision and Machine Learning*, pages 135–154, 2018.

Rahaf Aljundi, Klaas Kelchtermans, and Tinne Tuytelaars. Task-free continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11254–11263, 2019.

Mohammed Amer and Tomás Maul. A review of modularization techniques in artificial neural networks. *Artificial Intelligence Review*, 52:527–561, 2019.

Marco Ancona, Enea Ceolini, Cengiz Öztireli, and Markus Gross. Towards better understanding of gradient-based attribution methods for deep neural networks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30-May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018.

Marco Ancona, Enea Ceolini, Cengiz Öztireli, and Markus Gross. Gradient-based attribution methods. *Explainable AI: Interpreting, explaining and visualizing deep learning*, pages 169–191, 2019.

Ankur Ankan and Abinash Panda. pgmpy: Probabilistic graphical models using python. In *Proceedings of the 14th Python in Science Conference (SCIPY 2015)*. Citeseer, 2015.

Leila Arras, Franziska Horn, Grégoire Montavon, Klaus-Robert Müller, and Wojciech Samek. " what is relevant in a text document?": An interpretable machine learning approach. *PloS one*, 12(8):e0181142, 2017.

Laura Aull. Connecting writing and language in assessment: Examining style, tone, and argument in the us common core standards and in exemplary student writing. *Assessing writing*, 24:59–73, 2015.

Margarita Azmitia and Ryan Montgomery. Friendship, transactive dialogues, and the development of scientific reasoning. *Social development*, 2(3):202–221, 1993.

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

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

Soumya Barikeri, Anne Lauscher, Ivan Vulić, and Goran Glavaš. Redditbias: A real-world resource for bias evaluation and debiasing of conversational language models. 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)*, pages 1941–1955, 2021.

David Bau, Jun-Yan Zhu, Hendrik Strobelt, Agata Lapedriza, Bolei Zhou, and Antonio Torralba. Understanding the role of individual units in a deep neural network. *Proceedings of the National Academy of Sciences*, 117(48):30071–30078, 2020.

Iz Beltagy, Matthew E Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*, 2020.

Yoshua Bengio, Tristan Deleu, Nasim Rahaman, Rosemary Ke, Sébastien Lachapelle, Olexa Bilaniuk, Anirudh Goyal, and Christopher Pal. A meta-transfer objective for learning to disentangle causal mechanisms. *arXiv preprint arXiv:1901.10912*, 2019.

Marvin W Berkowitz and John C Gibbs. Measuring the developmental features of moral discussion. *Merrill-Palmer Quarterly (1982-)*, pages 399–410, 1983.

Joachim Bingel and Anders Søgaard. Identifying beneficial task relations for multi-task learning in deep neural networks. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 164–169, 2017.

Davis Blalock, Jose Javier Gonzalez Ortiz, Jonathan Frankle, and John Guttag. What is the state of neural network pruning? *Proceedings of machine learning and systems*, 2:129–146, 2020.

Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

Johan Bos and Katja Markert. Recognising textual entailment with logical inference. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, HLT '05, pages 628–635, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics. doi: 10.3115/1220575.1220654. URL `http://dx.doi.org/10.3115/1220575.1220654`.

Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, 2015.

Samuel R Bowman, Jon Gauthier, Abhinav Rastogi, Raghav Gupta, Christopher D Manning, and Christopher Potts. A fast unified model for parsing and sentence understanding. *arXiv preprint arXiv:1603.06021*, 2016.

Chloé Braud, Barbara Plank, and Anders Søgaard. Multi-view and multi-task training of rst discourse parsers. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1903–1913, 2016.

Chloé Braud, Ophélie Lacroix, and Anders Søgaard. Cross-lingual and cross-domain discourse segmentation of entire documents. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 237–243, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-2037. URL https://www.aclweb.org/anthology/P17-2037.

Peter F Brown, Vincent J Della Pietra, Peter V Desouza, Jennifer C Lai, and Robert L Mercer. Class-based n-gram models of natural language. *Computational linguistics*, 18(4):467–480, 1992.

Oana-Maria Camburu, Tim Rocktäschel, Thomas Lukasiewicz, and Phil Blunsom. e-snli: Natural language inference with natural language explanations. *Advances in Neural Information Processing Systems*, 31, 2018.

Lynn Carlson, Daniel Marcu, and Mary Ellen Okurowski. Building a discourse-tagged corpus in the framework of rhetorical structure theory. In *Current and new directions in discourse and dialogue*, pages 85–112. Springer, 2003.

Samuel Carton, Anirudh Rathore, and Chenhao Tan. Evaluating and characterizing human rationales. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9294–9307, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.747. URL https://aclanthology.org/2020.emnlp-main.747.

Rich Caruana, Steve Lawrence, and C Giles. Overfitting in neural nets: Backpropagation, conjugate gradient, and early stopping. *Advances in neural information processing systems*, 13, 2000.

Stephen Casper, Kaivalya Hariharan, and Dylan Hadfield-Menell. Diagnostics for deep neural networks with automated copy/paste attacks. In *NeurIPS ML Safety Workshop*, 2022a.

Stephen Casper, Shlomi Hod, Daniel Filan, Cody Wild, Andrew Critch, and Stuart Russell. Graphical clusterability and local specialization in deep neural networks. In *ICLR 2022 Workshop on PAIR {\textasciicircum} 2Struct: Privacy, Accountability, Interpretability, Robustness, Reasoning on Structured Data*, 2022b.

Stephen Casper, Max Nadeau, Dylan Hadfield-Menell, and Gabriel Kreiman. Robust feature-level adversaries are interpretability tools. *Advances in Neural Information Processing Systems*, 35:33093–33106, 2022c.

Ilias Chalkidis, Manos Fergadiotis, Dimitrios Tsarapatsanis, Nikolaos Aletras, Ion Androutsopoulos, and Prodromos Malakasiotis. Paragraph-level rationale extraction through regularization: A case study on European court of human rights cases. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 226–241, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.22. URL https://aclanthology.org/2021.naacl-main.22.

Jie Chao, Rebecca Ellis, Shiyan Jiang, Carolyn Rosé, William Finzer, Cansu Tatar, James Fiacco, and Kenia Wiedemann. Exploring artificial intelligence in english language arts with storyq. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 15999–16003, 2023.

Hila Chefer, Shir Gur, and Lior Wolf. Transformer interpretability beyond attention visualization. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 782–791. IEEE, 2021.

Qian Chen, Xiaodan Zhu, Zhenhua Ling, Si Wei, and Hui Jiang. Enhancing and combining sequential and tree lstm for natural language inference. *arXiv preprint arXiv:1609.06038*, 2016.

Zhi Chen, Yijie Bei, and Cynthia Rudin. Concept whitening for interpretable image recognition. *Nature Machine Intelligence*, 2(12):772–782, 2020.

Sharan Chetlur, Cliff Woolley, Philippe Vandermersch, Jonathan Cohen, John Tran, Bryan Catanzaro, and Evan Shelhamer. cudnn: Efficient primitives for deep learning. *arXiv preprint arXiv:1410.0759*, 2014.

Manjari Chitti, Padmini Chitti, and Manoj Jayabalan. Need for interpretable student performance prediction. In *2020 13th International Conference on Developments in eSystems Engineering (DeSE)*, pages 269–272. IEEE, 2020.

François Chollet et al. Keras. https://keras.io, 2015.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30, 2017.

Elizabeth Clark, Tal August, Sofia Serrano, Nikita Haduong, Suchin Gururangan, and Noah A Smith. All that's 'human'is not gold: Evaluating human evaluation of generated text. 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)*, pages 7282–7296, 2021.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12(Aug):2493–2537, 2011.

Pierre Comon. Independent component analysis, a new concept? *Signal processing*, 36 (3):287–314, 1994.

Cleo Condoravdi, Dick Crouch, Valeria De Paiva, Reinhard Stolle, and Daniel G Bobrow. Entailment, intensionality and text understanding. In *Proceedings of the HLT-NAACL 2003 workshop on Text meaning-Volume 9*, pages 38–45. Association for Computational Linguistics, 2003.

Alexis Conneau, Germán Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. What you can cram into a single $ &!#* vector: Probing sentence embeddings for linguistic properties. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2126–2136, 2018.

Robert G Cowell, Steffen L Lauritzen, A Philip David, and David J Spiegelhalter. Probabilistic networks and expert systems, 1999.

Róbert Csordás, Sjoerd van Steenkiste, and Jürgen Schmidhuber. Are neural nets modular? inspecting functional modularity through differentiable weight masks. In *International Conference on Learning Representations*, 2020.

Ido Dagan and Oren Glickman. Probabilistic textual entailment: Generic applied modeling of language variability. *Learning Methods for Text Understanding and Mining*, 2004:26–29, 2004.

Mihai Dascalu, Stefan Trausan-Matu, Danielle S McNamara, and Philippe Dessus. Readerbench: Automated evaluation of collaboration based on cohesion and dialogism. *International Journal of Computer-Supported Collaborative Learning*, 10(4): 395–423, 2015.

Maria De-Arteaga, Alexey Romanov, Hanna Wallach, Jennifer Chayes, Christian Borgs, Alexandra Chouldechova, Sahin Geyik, Krishnaram Kenthapadi, and Adam Tauman Kalai. Bias in bios: A case study of semantic representation bias in a high-stakes setting. In *proceedings of the Conference on Fairness, Accountability, and Transparency*, pages 120–128, 2019.

Ester J de Jong and Candace A Harper. Preparing mainstream teachers for english-language learners: Is being a good teacher good enough? *Teacher Education Quarterly*, 32(2):101–124, 2005.

Richard De Lisi and Susan L Golbeck. Implications of piagetian theory for peer learning. 1999.

Jamie DeCoster. Overview of factor analysis. 1998.

Jean-Stanislas Denain and Jacob Steinhardt. Auditing visualizations: Transparency methods struggle to detect anomalous behavior. *arXiv preprint arXiv:2206.13498*, 2022.

William B Dolan and Chris Brockett. Automatically constructing a corpus of sentential paraphrases. In *Proc. of IWP*, 2005.

Ann-Kathrin Dombrowski, Maximillian Alber, Christopher Anders, Marcel Ackermann, Klaus-Robert Müller, and Pan Kessel. Explanations can be manipulated and geometry is to blame. *Advances in neural information processing systems*, 32, 2019.

Finale Doshi-Velez and Been Kim. Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*, 2017.

Timothy Dozat and Christopher D. Manning. Deep biaffine attention for neural dependency parsing. In *ICLR 2017*, 2017.

Mengnan Du, Varun Manjunatha, Rajiv Jain, Ruchi Deshpande, Franck Dernoncourt, Jiuxiang Gu, Tong Sun, and Xia Hu. Towards interpreting and mitigating shortcut learning behavior of nlu models. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 915–929, 2021.

Nadir Durrani, Hassan Sajjad, Fahim Dalvi, and Yonatan Belinkov. Analyzing individual neurons in pre-trained language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4865–4880, 2020.

Julian Eisenschlos, Bhuwan Dhingra, Jannis Bulian, Benjamin Börschinger, and Jordan Boyd-Graber. Fool me twice: Entailment from wikipedia gamification. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2021.

Yanai Elazar, Shauli Ravfogel, Alon Jacovi, and Yoav Goldberg. Amnesic Probing: Behavioral Explanation with Amnesic Counterfactuals. *Transactions of the Association for Computational Linguistics*, 9:160–175, 03 2021. ISSN 2307-387X. doi: 10.1162/tacl_a_00359. URL `https://doi.org/10.1162/tacl\_a\_00359`.

Katherine Elkins and Jon Chun. Can gpt-3 pass a writer's turing test? *Journal of Cultural Analytics*, 5(2), 2020.

Logan Engstrom, Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Brandon Tran, and Aleksander Madry. Adversarial robustness as a prior for learned representations (2019). *arXiv preprint arXiv:1906.00945*, pages 1–25, 1906.

Babak Esmaeili, Hao Wu, Sarthak Jain, Alican Bozkurt, N Siddharth, Brooks Paige, Dana H Brooks, Jennifer Dy, and Jan-Willem van de Meent. Structured disentangled representations. *stat*, 1050:12, 2018.

Feng-Lei Fan, Jinjun Xiong, Mengzhou Li, and Ge Wang. On interpretability of artificial neural networks: A survey. *IEEE Transactions on Radiation and Plasma Medical Sciences*, 5(6):741–760, 2021.

Vanessa Wei Feng and Graeme Hirst. A linear-time bottom-up discourse parser with constraints and post-editing. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 511–521, Baltimore, Maryland, June 2014a. Association for Computational Linguistics. doi: 10.3115/v1/P14-1048. URL https://www.aclweb.org/anthology/P14-1048.

Vanessa Wei Feng and Graeme Hirst. Patterns of local discourse coherence as a feature for authorship attribution. *Literary and Linguistic Computing*, 29(2):191–198, 2014b.

Oliver Ferschke, Diyi Yang, Gaurav Tomar, and Carolyn Penstein Rosé. Positive impact of collaborative chat participation in an edx mooc. In *International Conference on Artificial Intelligence in Education*, pages 115–124. Springer, 2015.

James Fiacco, Samridhi Choudhary, and Carolyn Rose. Deep neural model inspection and comparison via functional neuron pathways. In *Proceedings of the 57th Conference of the Association for Computational Linguistics*, pages 5754–5764, 2019a.

James Fiacco, Elena Cotos, and Carolyn Rosé. Towards enabling feedback on rhetorical structure with neural sequence models. In *Proceedings of the 9th International Conference on Learning Analytics & Knowledge*, pages 310–319. ACM, 2019b.

James Fiacco, Ki-Won Haan, Anita Williams Woolley, and Carolyn Rosé. Taking transactivity detection to a new level. In *Proceedings of the 14th International Conference on Computer-Supported Collaborative Learning-CSCL 2021*. International Society of the Learning Sciences, 2021.

James Fiacco, Shiyan Jiang, David Adamson, and Carolyn Rose. Toward automatic discourse parsing of student writing motivated by neural interpretation. In *Proceedings of the 17th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2022)*, pages 204–215, 2022.

James Fiacco, David Adamson, and Carolyn Rose. Towards extracting and understanding the implicit rubrics of transformer based automatic essay scoring models. In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications*, 2023.

Hidde Fokkema, Rianne de Heide, and Tim van Erven. Attribution-based explanations that provide recourse cannot be robust. *arXiv preprint arXiv:2205.15834*, 2022.

Ruth Fong and Andrea Vedaldi. Net2vec: Quantifying and explaining how concepts are encoded by filters in deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8730–8738, 2018.

Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *International Conference on Learning Representations*, 2018.

Leon Fröhling and Arkaitz Zubiaga. Feature-based detection of automated language models: tackling gpt-2, gpt-3 and grover. *PeerJ Computer Science*, 7:e443, 2021.

Kevin Fuchs. Exploring the opportunities and challenges of nlp models in higher education: is chat gpt a blessing or a curse? In *Frontiers in Education*, volume 8, page 1166682. Frontiers, 2023.

Matthias Gallé, Jos Rozen, Germán Kruszewski, and Hady Elsahar. Unsupervised and distributional detection of machine-generated text. *arXiv preprint arXiv:2111.02878*, 2021.

Atticus Geiger, Zhengxuan Wu, Hanson Lu, Josh Rozner, Elisa Kreiss, Thomas Icard, Noah Goodman, and Christopher Potts. Inducing causal structure for interpretable neural networks. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 7324–7338. PMLR, 17–23 Jul 2022. URL `https://proceedings.mlr.press/v162/geiger22a.html`.

Rayid Ghani, Kit T. Rodolfa, Pedro Saleiro, and Sérgio Jesus. Addressing bias and fairness in machine learning: A practical guide and hands-on tutorial. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '23, page 5779–5780, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9798400701030. doi: 10.1145/3580305.3599180. URL `https://doi.org/10.1145/3580305.3599180`.

Amirata Ghorbani, Abubakar Abid, and James Zou. Interpretation of neural networks is fragile. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 3681–3688, 2019.

C Lee Giles, Bill G Horne, and Tsungnan Lin. Learning a class of large finite state machines with a recurrent neural network. *Neural Networks*, 8(9):1359–1365, 1995.

Max Glockner, Vered Shwartz, and Yoav Goldberg. Breaking nli systems with sentences that require simple lexical inferences. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 650–655, 2018.

Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier neural networks. In *Aistats*, volume 15, page 275, 2011.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. Deep learning (adaptive computation and machine learning series). *Adaptive Computation and Machine Learning series*, page 800, 2016.

Alex Graves and Jürgen Schmidhuber. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural Networks*, 18(5):602–610, 2005.

Christopher Grimsley, Elijah Mayfield, and Julia Bursten. Why attention is not explanation: Surgical intervention and causal reasoning about neural models. 2020.

Biyang Guo, Xin Zhang, Ziyuan Wang, Minqi Jiang, Jinran Nie, Yuxuan Ding, Jianwei Yue, and Yupeng Wu. How close is chatgpt to human experts? comparison corpus, evaluation, and detection. *arXiv preprint arXiv:2301.07597*, 2023.

Abhijeet Gupta, Gemma Boleda, Marco Baroni, and Sebastian Padó. Distributional vectors encode referential attributes. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 12–21, 2015.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A Smith. Annotation artifacts in natural language inference data. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 107–112, 2018.

Gahgene Gweon, Mahaveer Jain, John McDonough, Bhiksha Raj, and Carolyn P Rosé. Measuring prevalence of other-oriented transactive contributions using an automated measure of speech style accommodation. *International Journal of Computer-Supported Collaborative Learning*, 8(2):245–265, 2013.

Lisa Anne Hendricks, Zeynep Akata, Marcus Rohrbach, Jeff Donahue, Bernt Schiele, and Trevor Darrell. Generating visual explanations. In *14th European Conference on Computer Vision*, pages 3–19. Springer, 2016.

Lisa Anne Hendricks, Ronghang Hu, Trevor Darrell, and Zeynep Akata. Grounding visual explanations. In *Proceedings of the European conference on computer vision (ECCV)*, pages 264–279, 2018.

Max Henrion. Propagating uncertainty in bayesian networks by probabilistic logic sampling. In *Machine intelligence and pattern recognition*, volume 5, pages 149–163. Elsevier, 1988.

Evan Hernandez, Sarah Schwettmann, David Bau, Teona Bagashvili, Antonio Torralba, and Jacob Andreas. Natural language descriptions of deep visual features. In *International Conference on Learning Representations*, 2021.

Cindy E Hmelo-Silver. *The international handbook of collaborative learning*. Routledge, 2013.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

Shlomi Hod, Daniel Filan, Stephen Casper, Andrew Critch, and Stuart Russell. Quantifying local specialization in deep neural networks. *arXiv preprint arXiv:2110.08058*, 2021.

Kenneth Holstein, Jennifer Wortman Vaughan, Hal Daumé III, Miro Dudik, and Hanna Wallach. Improving fairness in machine learning systems: What do industry practitioners need? In *Proceedings of the 2019 CHI conference on human factors in computing systems*, pages 1–16, 2019.

Harold Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of educational psychology*, 24(6):417, 1933.

Yaojie Hu and Jin Tian. Neuron dependency graphs: A causal abstraction of neural networks. In *International Conference on Machine Learning*, pages 9020–9040. PMLR, 2022.

Jingshan Huang and Ming Tan. The role of chatgpt in scientific communication: writing better scientific review articles. *American Journal of Cancer Research*, 13(4):1148, 2023.

Yupan Huang, Tengchao Lv, Lei Cui, Yutong Lu, and Furu Wei. Layoutlmv3: Pre-training for document ai with unified text and image masking. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 4083–4091, 2022.

Drew Hudson and Christopher D Manning. Learning by abstraction: The neural state machine. *Advances in Neural Information Processing Systems*, 32, 2019.

Daphne Ippolito, Daniel Duckworth, Chris Callison-Burch, and Douglas Eck. Automatic detection of generated text is easiest when humans are fooled. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1808–1822, 2020.

Alon Jacovi and Yoav Goldberg. Towards faithfully interpretable nlp systems: How should we define and evaluate faithfulness? *arXiv preprint arXiv:2004.03685*, 2020.

Alon Jacovi, Ana Marasović, Tim Miller, and Yoav Goldberg. Formalizing trust in artificial intelligence: Prerequisites, causes and goals of human trust in ai. In *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency*, pages 624–635, 2021.

Sarthak Jain and Byron C Wallace. Attention is not explanation. In *Proceedings of NAACL-HLT*, pages 3543–3556, 2019.

Guillaume Jaume, Hazim Kemal Ekenel, and Jean-Philippe Thiran. Funsd: A dataset for form understanding in noisy scanned documents. In *2019 International Conference on Document Analysis and Recognition Workshops (ICDARW)*, volume 2, pages 1–6. IEEE, 2019.

Ganesh Jawahar, Muhammad Abdul-Mageed, and VS Laks Lakshmanan. Automatic detection of machine generated text: A critical survey. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2296–2309, 2020.

Jeya Vikranth Jeyakumar, Joseph Noor, Yu-Hsi Cheng, Luis Garcia, and Mani Srivastava. How can i explain this to you? an empirical study of deep neural network explanation methods. *Advances in Neural Information Processing Systems*, 33:4211–4222, 2020.

Yangfeng Ji and Jacob Eisenstein. Representation learning for text-level discourse parsing. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13–24, Baltimore, Maryland, June 2014. Association for Computational Linguistics. doi: 10.3115/v1/P14-1002. URL https://www.aclweb.org/anthology/P14-1002.

Yangfeng Ji and Noah A. Smith. Neural discourse structure for text categorization. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 996–1005, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1092. URL https://www.aclweb.org/anthology/P17-1092.

Shiyan Jiang, Kexin Yang, Chandrakumari Suvarna, Pooja Casula, Mingtong Zhang, and Carolyn Rose. Applying rhetorical structure theory to student essays for providing automated writing feedback. In *Proceedings of the Workshop on Discourse Relation Parsing and Treebanking 2019*, pages 163–168, 2019.

Eric Jones, Travis Oliphant, Pearu Peterson, et al. SciPy: Open source scientific tools for Python, 2001–. URL http://www.scipy.org/. [Online; accessed ¡today¿].

Mahesh Joshi and Carolyn Penstein Rosé. Using transactivity in conversation for summarization of educational dialogue. In *SLaTE*, pages 53–56, 2007.

Henry F Kaiser. The varimax criterion for analytic rotation in factor analysis. *Psychometrika*, 23(3):187–200, 1958.

Faisal Kamiran and Indrė Žliobaitė. Explainable and non-explainable discrimination in classification. In *Discrimination and Privacy in the Information Society*, pages 155–170. Springer, 2013.

Andrej Karpathy, Justin Johnson, and Li Fei-Fei. Visualizing and understanding recurrent networks. *arXiv preprint arXiv:1506.02078*, 2015.

C Kennedy, MH Des Rosiers, JW Jehle, M Reivich, F Sharpe, and L Sokoloff. Mapping of functional neural pathways by autoradiographic survey of local metabolic rate with (14c) deoxyglucose. *Science*, 187(4179):850–853, 1975.

Urvashi Khandelwal, He He, Peng Qi, and Dan Jurafsky. Sharp nearby, fuzzy far away: How neural language models use context. *arXiv preprint arXiv:1805.04623*, 2018.

Daniel Khashabi, Snigdha Chaturvedi, Michael Roth, Shyam Upadhyay, and Dan Roth. Looking beyond the surface: A challenge set for reading comprehension over multiple sentences. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 252–262, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1023. URL https://www.aclweb.org/anthology/N18-1023.

Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, et al. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In *International conference on machine learning*, pages 2668–2677. PMLR, 2018a.

Jinkyu Kim, Anna Rohrbach, Trevor Darrell, John Canny, and Zeynep Akata. Textual explanations for self-driving vehicles. In *Proceedings of the European conference on computer vision (ECCV)*, pages 563–578, 2018b.

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

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017.

Donald E Knuth. On the translation of languages from left to right. *Information and control*, 8(6):607–639, 1965.

Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. Concept bottleneck models. In *International conference on machine learning*, pages 5338–5348. PMLR, 2020.

Arne Köhn. What's in an embedding? analyzing word embeddings through multilingual evaluation. 2015.

Daphne Koller and Nir Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

Sanjay Krishnan and Eugene Wu. Palm: Machine learning explanations for iterative debugging. In *Proceedings of the 2nd Workshop on Human-In-the-Loop Data Analytics*, page 4. ACM, 2017.

Aounon Kumar, Chirag Agarwal, Suraj Srinivas, Soheil Feizi, and Hima Lakkaraju. Certifying llm safety against adversarial prompting. *arXiv preprint arXiv:2309.02705*, 2023.

Rohit Kumar, Carolyn Penstein Rosé, Yi-Chia Wang, Mahesh Joshi, and Allen Robinson. Tutorial dialogue as adaptive collaborative learning support. *Frontiers in artificial intelligence and applications*, 158:383, 2007.

Sawan Kumar and Partha Talukdar. Nile: Natural language inference with faithful natural language explanations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8730–8742, 2020.

Adhiguna Kuncoro, Chris Dyer, John Hale, Dani Yogatama, Stephen Clark, and Phil Blunsom. Lstms can learn syntax-sensitive dependencies well, but modeling structure makes them better. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1426–1436, 2018.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466, March 2019. doi: 10.1162/tacl_a_00276. URL `https://www.aclweb.org/anthology/Q19-1026`.

Isaac Lage and Finale Doshi-Velez. The neural lasso: Local linear sparsity for interpretable explanations. volume 4, 2017.

Matthew Lamm, Jennimaria Palomaki, Chris Alberti, Daniel Andor, Eunsol Choi, Livio Baldini Soares, and Michael Collins. Qed: A framework and dataset for explanations in question answering. *Transactions of the Association for computational Linguistics*, 9:790–806, 2021.

Richard D Lange, David Rolnick, and Konrad Kording. Clustering units in neural networks: upstream vs downstream information. *Transactions on Machine Learning Research*, 2022.

Po-Ming Law, Sana Malik, Fan Du, and Moumita Sinha. Designing tools for semi-automated detection of machine learning biases: An interview study. *arXiv preprint arXiv:2003.07680*, 2020.

Hwaran Lee, Seokhee Hong, Joonsuk Park, Takyoung Kim, Gunhee Kim, and Jung-Woo Ha. Kosbi: A dataset for mitigating social bias risks towards safer large language model application. *arXiv preprint arXiv:2305.17701*, 2023.

Soochan Lee, Junsoo Ha, Dongsu Zhang, and Gunhee Kim. A neural dirichlet process mixture model for task-free continual learning. In *International Conference on Learning Representations*, 2019.

Sabine Lehmann, Stephan Oepen, Sylvie Regnier-Prost, Klaus Netter, Veronika Lux, Judith Klein, Kirsten Falkedal, Frederik Fouvry, Dominique Estival, Eva Dauphin, et al. Tsnlp-test suites for natural language processing. In *COLING 1996 Volume 2: The 16th International Conference on Computational Linguistics*, 1996.

Michael A Lepori and R Thomas McCoy. Picking bert's brain: Probing for linguistic dependencies in contextualized embeddings using representational similarity analysis. *arXiv preprint arXiv:2011.12073*, 2020.

Jiaoda Li, Ryan Cotterell, and Mrinmaya Sachan. Probing via prompting. *arXiv preprint arXiv:2207.01736*, 2022.

Jiwei Li, Rumeng Li, and Eduard Hovy. Recursive deep models for discourse parsing. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2061–2069, 2014.

Mengjun Li and Ayoung Suh. Machinelike or humanlike? a literature review of anthropomorphism in ai-enabled technology. In *54th Hawaii International Conference on System Sciences (HICSS 2021)*, pages 4053–4062, 2021.

Qi Li, Tianshi Li, and Baobao Chang. Discourse parsing with attention-based hierarchical neural networks. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 362–371, 2016.

Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947, 2017.

Weixin Liang, Mert Yuksekgonul, Yining Mao, Eric Wu, and James Zou. Gpt detectors are biased against non-native english writers. 2023.

Stephanie Lin, Jacob Hilton, and Owain Evans. Truthfulqa: Measuring how models mimic human falsehoods. *arXiv preprint arXiv:2109.07958*, 2021.

Wanyu Lin, Hao Lan, Hao Wang, and Baochun Li. Orphicx: A causality-inspired latent variable model for interpreting graph neural networks. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13719–13728. IEEE Computer Society, 2022.

Adam Dahlgren Lindström, Suna Bensch, Johanna Björklund, and Frank Drewes. Probing multimodal embeddings for linguistic properties: the visual-semantic case. *arXiv preprint arXiv:2102.11115*, 2021.

Nelson F Liu, Roy Schwartz, and Noah A Smith. Inoculation by fine-tuning: A method for analyzing challenge datasets. *arXiv preprint arXiv:1904.02668*, 2019a.

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

Yu-han Liu and Sercan O Arik. Explaining deep neural networks using unsupervised clustering. *arXiv preprint arXiv:2007.07477*, 2020.

Max Losch, Mario Fritz, and Bernt Schiele. Interpretability beyond classification output: Semantic bottleneck networks. *arXiv preprint arXiv:1907.10882*, 2019.

Max Losch, Mario Fritz, and Bernt Schiele. Semantic bottlenecks: Quantifying and improving inspectability of deep representations. *International Journal of Computer Vision*, 129:3136–3153, 2021.

Pan Lu, Swaroop Mishra, Tanglin Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering. *Advances in Neural Information Processing Systems*, 35:2507–2521, 2022.

Adriano Lucieri, Muhammad Naseer Bajwa, Stephan Alexander Braun, Muhammad Imran Malik, Andreas Dengel, and Sheraz Ahmed. On interpretability of deep learning based skin lesion classifiers using concept activation vectors. In *2020 international joint conference on neural networks (IJCNN)*, pages 1–10. IEEE, 2020a.

Adriano Lucieri, Muhammad Naseer Bajwa, Andreas Dengel, and Sheraz Ahmed. Explaining ai-based decision support systems using concept localization maps. In *International Conference on Neural Information Processing*, pages 185–193. Springer, 2020b.

Daniel D Lundstrom, Tianjian Huang, and Meisam Razaviyayn. A rigorous study of integrated gradients method and extensions to internal neuron attributions. In *International Conference on Machine Learning*, pages 14485–14508. PMLR, 2022.

Minh-Thang Luong, Quoc V Le, Ilya Sutskever, Oriol Vinyals, and Lukasz Kaiser. Multi-task sequence to sequence learning. *arXiv preprint arXiv:1511.06114*, 2015.

Xuezhe Ma and Eduard Hovy. End-to-end sequence labeling via bi-directional lstm-cnns-crf. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1064–1074, 2016.

Amandla Mabona, Laura Rimell, Stephen Clark, and Andreas Vlachos. Neural generative rhetorical structure parsing. 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 2284–2295, 2019.

Andreas Madsen, Siva Reddy, and Sarath Chandar. Post-hoc interpretability for neural nlp: A survey. *ACM Computing Surveys*, 55(8):1–42, 2022.

Aravindh Mahendran and Andrea Vedaldi. Understanding deep image representations by inverting them. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5188–5196, 2015.

William C Mann and Sandra A Thompson. Rhetorical structure theory: Toward a functional theory of text organization. *Text-interdisciplinary Journal for the Study of Discourse*, 8(3):243–281, 1988.

Mitchell P Marcus, Beatrice Santorini, Mary Ann Marcinkiewicz, and Ann Taylor. Penn treebank-3. *Linguistic Data Consortium, LDC99T42, University of Pennsylvania*, 1999.

Ray Marshall. The economics of racial discrimination: A survey. *Journal of Economic Literature*, 12(3):849–871, 1974.

Bruce M McLaren, Oliver Scheuer, Maarten De Laat, Rakheli Hever, Reuma De Groot, and Carolyn Penstein Rosé. Using machine learning techniques to analyze and support mediation of student e-discussions. *Frontiers in Artificial Intelligence and Applications*, 158:331, 2007.

Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. A survey on bias and fairness in machine learning. *ACM Computing Surveys (CSUR)*, 54(6):1–35, 2021.

Clara Meister, Stefan Lazov, Isabelle Augenstein, and Ryan Cotterell. Is sparse attention more interpretable? In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 122–129, 2021.

Alessio Miaschi, Chiara Alzetta, Dominique Brunato, Felice Dell'Orletta, and Giulia Venturi. Probing tasks under pressure. In *CLiC-it*, 2021.

Tim Miller. Explanation in artificial intelligence: Insights from the social sciences. *Artificial intelligence*, 267:1–38, 2019.

Sarthak Mittal, Yoshua Bengio, and Guillaume Lajoie. Is a modular architecture enough? *Advances in Neural Information Processing Systems*, 35:28747–28760, 2022.

Gemma E Moran, Dhanya Sridhar, Yixin Wang, and David M Blei. Identifiable variational autoencoders via sparse decoding. *arXiv preprint arXiv:2110.10804*, 9, 2021.

Ari S Morcos, David GT Barrett, Neil C Rabinowitz, and Matthew Botvinick. On the importance of single directions for generalization. In *International Conference on Learning Representations*, 2018.

Mathieu Morey, Philippe Muller, and Nicholas Asher. How much progress have we made on RST discourse parsing? a replication study of recent results on the RST-DT. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1319–1324, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1136. URL https://www.aclweb.org/anthology/D17-1136.

Jesse Mu and Jacob Andreas. Compositional explanations of neurons. *Advances in Neural Information Processing Systems*, 33:17153–17163, 2020.

Jin Mu, Karsten Stegmann, Elijah Mayfield, Carolyn Rosé, and Frank Fischer. The acodea framework: Developing segmentation and classification schemes for fully automatic analysis of online discussions. *International journal of computer-supported collaborative learning*, 7(2):285–305, 2012.

Tsendsuren Munkhdalai and Hong Yu. Neural tree indexers for text understanding. *arXiv preprint arXiv:1607.04492*, 2016.

Aakanksha Naik, Abhilasha Ravichander, Norman Sadeh, Carolyn Rose, and Graham Neubig. Stress test evaluation for natural language inference. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2340–2353, Santa Fe, New Mexico, USA, August 2018a. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/C18-1198.

Aakanksha Naik, Abhilasha Ravichander, Norman Sadeh, Carolyn Rose, and Graham Neubig. Stress test evaluation for natural language inference. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2340–2353, 2018b.

Matti Nelimarkka and Arto Vihavainen. Alumni & tenured participants in moocs: Analysis of two years of mooc discussion channel activity. In *Proceedings of the Second (2015) ACM Conference on Learning@ Scale*, pages 85–93. ACM, 2015.

Graham Neubig, Chris Dyer, Yoav Goldberg, Austin Matthews, Waleed Ammar, Antonios Anastasopoulos, Miguel Ballesteros, David Chiang, Daniel Clothiaux, Trevor Cohn, Kevin Duh, Manaal Faruqui, Cynthia Gan, Dan Garrette, Yangfeng Ji, Lingpeng Kong, Adhiguna Kuncoro, Gaurav Kumar, Chaitanya Malaviya, Paul Michel, Yusuke Oda, Matthew Richardson, Naomi Saphra, Swabha Swayamdipta, and Pengcheng Yin. Dynet: The dynamic neural network toolkit. *arXiv preprint arXiv:1701.03980*, 2017a.

Graham Neubig, Chris Dyer, Yoav Goldberg, Austin Matthews, Waleed Ammar, Antonios Anastasopoulos, Miguel Ballesteros, David Chiang, Daniel Clothiaux, Trevor Cohn, et al. Dynet: The dynamic neural network toolkit. *arXiv preprint arXiv:1701.03980*, 2017b.

Anh Nguyen, Alexey Dosovitskiy, Jason Yosinski, Thomas Brox, and Jeff Clune. Synthesizing the preferred inputs for neurons in neural networks via deep generator networks. *Advances in neural information processing systems*, 29, 2016a.

Anh Nguyen, Jason Yosinski, and Jeff Clune. Multifaceted feature visualization: Uncovering the different types of features learned by each neuron in deep neural networks. *arXiv preprint arXiv:1602.03616*, 2016b.

Anh Nguyen, Jeff Clune, Yoshua Bengio, Alexey Dosovitskiy, and Jason Yosinski. Plug & play generative networks: Conditional iterative generation of images in latent space. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4467–4477, 2017.

John Nickolls, Ian Buck, Michael Garland, and Kevin Skadron. Scalable parallel programming with cuda. *Queue*, 6(2):40–53, 2008.

Ian E Nielsen, Dimah Dera, Ghulam Rasool, Ravi P Ramachandran, and Nidhal Carla Bouaynaya. Robust explainability: A tutorial on gradient-based attribution methods for deep neural networks. *IEEE Signal Processing Magazine*, 39(4):73–84, 2022.

Timothy Niven and Hung-Yu Kao. Probing neural network comprehension of natural language arguments. *arXiv preprint arXiv:1907.07355*, 2019.

Joakim Nivre, Marie-Catherine De Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajic, Christopher D Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, et al. Universal dependencies v1: A multilingual treebank collection. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 1659–1666, 2016.

Armineh Nourbakhsh, Jan 2024.

Tuomas Oikarinen and Tsui-Wei Weng. Clip-dissect: Automatic description of neuron representations in deep vision networks. In *The Eleventh International Conference on Learning Representations*, 2022.

Chris Olah, Alexander Mordvintsev, and Ludwig Schubert. Feature visualization. *Distill*, 2(11):e7, 2017.

OpenAI. Gpt-2 output dataset, 2021. URL https://github.com/openai/gpt-2-output-dataset. Accessed: 2023-06-1.

OpenAI. ChatGPT. https://chat.openai.com/, 2022. Accessed: 2023-6-1.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744, 2022.

Martha Palmer, Daniel Gildea, and Nianwen Xue. Semantic role labeling. *Synthesis Lectures on Human Language Technologies*, 3(1):1–103, 2010.

Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2010.

Bo Pang and Lillian Lee. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 115–124, Ann Arbor, Michigan, June 2005. Association for Computational Linguistics. doi: 10.3115/1219840.1219855. URL https://www.aclweb.org/anthology/P05-1015.

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pages 79–86. Association for Computational Linguistics, 2002.

Ankur Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. A decomposable attention model for natural language inference. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2249–2255, 2016a.

Ankur P. Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. A decomposable attention model for natural language inference. *CoRR*, abs/1606.01933, 2016b. URL http://arxiv.org/abs/1606.01933.

Badri Patro, Shivansh Patel, and Vinay Namboodiri. Robust explanations for visual question answering. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1577–1586, 2020.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Hao Peng, Sam Thomson, and Noah A Smith. Deep multitask learning for semantic dependency parsing. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2037–2048, 2017.

Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

Christian S Perone, Roberto Silveira, and Thomas S Paula. Evaluation of sentence embeddings in downstream and linguistic probing tasks. *arXiv preprint arXiv:1806.06259*, 2018.

Matthew E Peters, Waleed Ammar, Chandra Bhagavatula, and Russell Power. Semi-supervised sequence tagging with bidirectional language models. *arXiv preprint arXiv:1705.00108*, 2017.

Adam Poliak, Jason Naradowsky, Aparajita Haldar, Rachel Rudinger, and Benjamin Van Durme. Hypothesis only baselines in natural language inference. In *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*, pages 180–191, 2018.

Alec Radford, Rafal Jozefowicz, and Ilya Sutskever. Learning to generate reviews and discovering sentiment. *arXiv preprint arXiv:1704.01444*, 2017.

Tilman Räuker, Anson Ho, Stephen Casper, and Dylan Hadfield-Menell. Toward transparent ai: A survey on interpreting the inner structures of deep neural networks. In *2023 IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)*, pages 464–483. IEEE, 2023.

Shauli Ravfogel, Michael Twiton, Yoav Goldberg, and Ryan D Cotterell. Linear adversarial concept erasure. In *International Conference on Machine Learning*, pages 18400–18421. PMLR, 2022.

Emily Reif, Ann Yuan, Martin Wattenberg, Fernanda B Viegas, Andy Coenen, Adam Pearce, and Been Kim. Visualizing and measuring the geometry of bert. *Advances in Neural Information Processing Systems*, 32, 2019.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 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. ACM, 2016.

Willy E Rice. Race, gender, redlining, and the discriminatory access to loans, credit, and insurance: An historical and empirical analysis of consumers who sued lenders and insurers in federal and state courts, 1950-1995. *San Diego L. Rev.*, 33:583, 1996.

Andrea Romei and Salvatore Ruggieri. A multidisciplinary survey on discrimination analysis. *The Knowledge Engineering Review*, 29(5):582–638, 2014.

Carolyn Rosé, Yi-Chia Wang, Yue Cui, Jaime Arguello, Karsten Stegmann, Armin Weinberger, and Frank Fischer. Analyzing collaborative learning processes automatically: Exploiting the advances of computational linguistics in computer-supported collaborative learning. *International journal of computer-supported collaborative learning*, 3(3):237–271, 2008.

Carolyn Penstein Rosé, Iris Howley, Miaomiao Wen, Diyi Yang, and Oliver Ferschke. Assessment of discussion in learning contexts. In *Innovative Assessment of Collaboration*, pages 81–94. Springer, 2017.

Andrei A Rusu, Neil C Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive neural networks. *arXiv preprint arXiv:1606.04671*, 2016.

Hassan Sajjad, Nadir Durrani, and Fahim Dalvi. Neuron-level interpretation of deep nlp models: A survey. *Transactions of the Association for Computational Linguistics*, 10: 1285–1303, 2022.

Abdelrhman Saleh, Tovly Deutsch, Stephen Casper, Yonatan Belinkov, and Stuart Shieber. Probing neural dialog models for conversational understanding. *arXiv preprint arXiv:2006.08331*, 2020.

Pedro Saleiro, Benedict Kuester, Abby Stevens, Ari Anisfeld, Loren Hinkson, Jesse London, and Rayid Ghani. Aequitas: A bias and fairness audit toolkit. *arXiv preprint arXiv:1811.05577*, 2018.

Hadi Salman, Andrew Ilyas, Logan Engstrom, Ashish Kapoor, and Aleksander Madry. Do adversarially robust imagenet models transfer better? *Advances in Neural Information Processing Systems*, 33:3533–3545, 2020.

Erik F. Tjong Kim Sang and Fien De Meulder. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*, pages 142–147. Association for Computational Linguistics, 2003.

Shibani Santurkar, Andrew Ilyas, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Image synthesis with a single (robust) classifier. *Advances in Neural Information Processing Systems*, 32, 2019.

Shibani Santurkar, Dimitris Tsipras, Mahalaxmi Elango, David Bau, Antonio Torralba, and Aleksander Madry. Editing a classifier by rewriting its prediction rules. *Advances in Neural Information Processing Systems*, 34:23359–23373, 2021.

Sofia Serrano and Noah A Smith. Is attention interpretable? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2931–2951, 2019.

Lei Sha, Baobao Chang, Zhifang Sui, and Sujian Li. Reading and thinking: Re-read lstm unit for textual entailment recognition. 2016.

Mike Sharples. Automated essay writing: an aied opinion. *International Journal of Artificial Intelligence in Education*, 32(4):1119–1126, 2022.

Xing Shi, Kevin Knight, and Deniz Yuret. Why neural translations are the right length. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2278–2282, 2016a.

Xing Shi, Inkit Padhi, and Kevin Knight. Does string-based neural mt learn source syntax? In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1526–1534, 2016b.

Narayanaswamy Siddharth, T Brooks Paige, Jan-Willem Van de Meent, Alban Desmaison, Noah Goodman, Pushmeet Kohli, Frank Wood, and Philip Torr. Learning disentangled representations with semi-supervised deep generative models. In *Advances in Neural Information Processing Systems*, pages 5925–5935, 2017.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

Dylan Slack, Sophie Hilgard, Emily Jia, Sameer Singh, and Himabindu Lakkaraju. Fooling lime and shap: Adversarial attacks on post hoc explanation methods. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, pages 180–186, 2020.

James Seale Smith, Junjiao Tian, Shaunak Halbe, Yen-Chang Hsu, and Zsolt Kira. A closer look at rehearsal-free continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2409–2419, 2023.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA, October 2013. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/D13-1170`.

Irene Solaiman, Miles Brundage, Jack Clark, Amanda Askell, Ariel Herbert-Voss, Jeff Wu, Alec Radford, Gretchen Krueger, Jong Wook Kim, Sarah Kreps, et al. Release strategies and the social impacts of language models. *arXiv preprint arXiv:1908.09203*, 2019.

Charles Spearman. The proof and measurement of association between two things. *The American journal of psychology*, 15(1):72–101, 1904.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. Learning to summarize with human feedback. *Advances in Neural Information Processing Systems*, 33:3008–3021, 2020.

Hendrik Strobelt, Sebastian Gehrmann, Bernd Huber, Hanspeter Pfister, and Alexander M Rush. Visual analysis of hidden state dynamics in recurrent neural networks. Technical report, Harvard University OpenScholar, 2016.

Anant Subramanian, Danish Pruthi, Harsh Jhamtani, Taylor Berg-Kirkpatrick, and Eduard Hovy. Spine: Sparse interpretable neural embeddings. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Swabha Swayamdipta, Miguel Ballesteros, Chris Dyer, and Noah A. Smith. Greedy, joint syntactic-semantic parsing with stack LSTMs. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 187–197, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/K16-1019. URL `https://www.aclweb.org/anthology/K16-1019`.

Alaina N Talboy and Elizabeth Fuller. Challenging the appearance of machine intelligence: Cognitive bias in llms. *arXiv preprint arXiv:2304.01358*, 2023.

Alex Tamkin, Trisha Singh, Davide Giovanardi, and Noah Goodman. Investigating transferability in pretrained language models. *arXiv preprint arXiv:2004.14975*, 2020.

Matthew E Taylor and Peter Stone. Transfer learning for reinforcement learning domains: A survey. *Journal of Machine Learning Research*, 10(Jul):1633–1685, 2009.

Himanshu Thakur, Atishay Jain, Praneetha Vaddamanu, Paul Pu Liang, and Louis-Philippe Morency. Language models get a gender makeover: Mitigating gender bias with few-shot data interventions. *arXiv preprint arXiv:2306.04597*, 2023.

Theano Development Team. Theano: A Python framework for fast computation of mathematical expressions. *arXiv e-prints*, abs/1605.02688, May 2016. URL `http://arxiv.org/abs/1605.02688`.

Peter Tiňo, Bill G Horne, C Lee Giles, and Pete C Collingwood. Finite state machines and recurrent neural networks—automata and dynamical systems approaches. In *Neural networks and pattern recognition*, pages 171–219. Elsevier, 1998.

Michael E Tipping and Christopher M Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 61(3):611–622, 1999.

Michalis K Titsias, Jonathan Schwarz, Alexander G de G Matthews, Razvan Pascanu, and Yee Whye Teh. Functional regularisation for continual learning with gaussian processes. In *International Conference on Learning Representations*, 2019.

Maksim Tkachenko and Andrey Simanovsky. Named entity recognition: Exploring features. In *KONVENS*, pages 118–127, 2012.

Joe Townsend, Theodoros Kasioumis, and Hiroya Inakoshi. Eric: extracting relations inferred from convolutions. In *Proceedings of the Asian Conference on Computer Vision*, 2020.

Adaku Uchendu, Thai Le, Kai Shu, and Dongwon Lee. Authorship attribution for neural text generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8384–8395, 2020.

Eddie Ungless, Amy Rafferty, Hrichika Nag, and Björn Ross. A robust bias mitigation procedure based on the stereotype content model. In *The 5th Workshop on Natural Language Processing and Computational Social Science*, 2022.

Sunil Vadera and Salem Ameen. Methods for pruning deep neural networks. *IEEE Access*, 10:63280–63300, 2022.

Sheila W. Valencia and Karen K. Wixson. Commentary: Inside english/language arts standards: What's in a grade? *Reading Research Quarterly*, 36(2):202–217, 2001.

Shikhar Vashishth, Shyam Upadhyay, Gaurav Singh Tomar, and Manaal Faruqui. Attention interpretability across nlp tasks. *arXiv preprint arXiv:1909.11218*, 2019.

Pranav Narayanan Venkit, Sanjana Gautam, Ruchi Panchanadikar, Ting-Hao Huang, and Shomir Wilson. Nationality bias in text generation. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 116–122, 2023.

Yulong Wang, Xiaolu Zhang, Xiaolin Hu, Bo Zhang, and Hang Su. Dynamic network pruning with interpretable layerwise channel selection. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 6299–6306, 2020.

Zhiguo Wang, Wael Hamza, and Radu Florian. Bilateral multi-perspective matching for natural language sentences. *arXiv preprint arXiv:1702.03814*, 2017.

Matthijs J Warrens. Five ways to look at cohen's kappa. *Journal of Psychology & Psychotherapy*, 5(4):1, 2015.

Chihiro Watanabe. Interpreting layered neural networks via hierarchical modular representation. In *Neural Information Processing: 26th International Conference, ICONIP 2019, Sydney, NSW, Australia, December 12–15, 2019, Proceedings, Part V 26*, pages 376–388. Springer, 2019.

Chihiro Watanabe, Kaoru Hiramatsu, and Kunio Kashino. Modular representation of layered neural networks. *Neural Networks*, 97:62–73, 2018.

Chihiro Watanabe, Kaoru Hiramatsu, and Kunio Kashino. Understanding community structure in layered neural networks. *Neurocomputing*, 367:84–102, 2019.

Miaomiao Wen. *Investigating Virtual Teams in Massive Open Online Courses: Deliberation-based Virtual Team Formation, Discussion Mining and Support*. PhD thesis, Carnegie Mellon University, 2016.

Miaomiao Wen, Keith Maki, Xu Wang, Steven Dow, James D. Herbsleb, and Carolyn Penstein Rosé. Transactivity as a predictor of future collaborative knowledge integration in team-based learning in online courses. In *Proceedings of the 21st ACM Conference on Computer-Supported Cooperative Work and Social Computing*, in press.

Patti West-Smith, Stephanie Butler, and Elijah Mayfield. Trustworthy automated essay scoring without explicit construct validity. In *AAAI Spring Symposia*, 2018a.

Patti West-Smith, Stephanie Butler, and Elijah Mayfield. Trustworthy automated essay scoring without explicit construct validity. In *AAAI Spring Symposia*, 2018b.

William Whitney. *Disentangled representations in neural models*. PhD thesis, Massachusetts Institute of Technology, 2016.

Sarah Wiegreffe and Ana Marasović. Teach me to explain: A review of datasets for explainable nlp. In *Proceedings of NeurIPS*, 2021. URL `https://arxiv.org/abs/2102.12060`.

Sarah Wiegreffe and Yuval Pinter. Attention is not not explanation. 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 11–20, 2019.

Steven L Willborn. The disparate impact model of discrimination: Theory and limits. *Am. UL Rev.*, 34:799, 1984.

Adina Williams, Nikita Nangia, and Samuel R Bowman. A broad-coverage challenge corpus for sentence understanding through inference, 2016. URL `https://www.nyu.edu/projects/bowman/multinli/paper.pdf`.

Adina Williams, Nikita Nangia, and Samuel Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, volume 1, pages 1112–1122, 2018.

Sam Wiseman and Alexander M Rush. Sequence-to-sequence learning as beam-search optimization. *arXiv preprint arXiv:1606.02960*, 2016.

Eric Wong, Shibani Santurkar, and Aleksander Madry. Leveraging sparse linear layers for debuggable deep networks. In *International Conference on Machine Learning*, pages 11205–11216. PMLR, 2021.

Mitchell Wortsman, Vivek Ramanujan, Rosanne Liu, Aniruddha Kembhavi, Mohammad Rastegari, Jason Yosinski, and Ali Farhadi. Supermasks in superposition. *Advances in Neural Information Processing Systems*, 33:15173–15184, 2020.

Duanli Yan, Michael Fauss, Jiangang Hao, and Wenju Cui. Detection of ai-generated essays in writing assessment. *Psychological Testing and Assessment Modeling*, 65(2): 125–144, 2023.

Zhilin Yang, Ruslan Salakhutdinov, and William W Cohen. Transfer learning for sequence tagging with hierarchical recurrent networks. *arXiv preprint arXiv:1703.06345*, 2017.

Will Yeadon, Oto-Obong Inyang, Arin Mizouri, Alex Peach, and Craig P Testrow. The death of the short-form physics essay in the coming ai revolution. *Physics Education*, 58(3):035027, 2023.

Seul-Ki Yeom, Philipp Seegerer, Sebastian Lapuschkin, Alexander Binder, Simon Wiedemann, Klaus-Robert Müller, and Wojciech Samek. Pruning by explaining: A novel criterion for deep neural network pruning. *Pattern Recognition*, 115:107899, 2021.

Jaehong Yoon, Eunho Yang, Jeongtae Lee, and Sung Ju Hwang. Lifelong learning with dynamically expandable networks. In *International Conference on Learning Representations*, 2018.

Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics*, 2:67–78, 2014.

Nan Yu, Meishan Zhang, and Guohong Fu. Transition-based neural rst parsing with implicit syntax features. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 559–570, 2018.

Mert Yuksekgonul, Maggie Wang, and James Zou. Post-hoc concept bottleneck models. *arXiv preprint arXiv:2205.15480*, 2022.

Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *International conference on machine learning*, pages 3987–3995. PMLR, 2017.

Longyin Zhang, Yuqing Xing, Fang Kong, Peifeng Li, and Guodong Zhou. A top-down neural architecture towards text-level parsing of discourse rhetorical structure. *arXiv preprint arXiv:2005.02680*, 2020a.

Quanshi Zhang, Ruiming Cao, Feng Shi, Ying Nian Wu, and Song-Chun Zhu. Interpreting cnn knowledge via an explanatory graph. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Quanshi Zhang, Xin Wang, Ruiming Cao, Ying Nian Wu, Feng Shi, and Song-Chun Zhu. Extraction of an explanatory graph to interpret a cnn. *IEEE transactions on pattern analysis and machine intelligence*, 43(11):3863–3877, 2020b.

Yujia Zhang, Kuangyan Song, Yiming Sun, Sarah Tan, and Madeleine Udell. " why should you trust my explanation?" understanding uncertainty in lime explanations. *arXiv preprint arXiv:1904.12991*, 2019.

Mengjie Zhao, Tao Lin, Fei Mi, Martin Jaggi, and Hinrich Schütze. Masking as an efficient alternative to finetuning for pretrained language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2226–2241, 2020.

Xinyan Zhao and VG Vinod Vydiswaran. Lirex: Augmenting language inference with relevant explanations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 14532–14539, 2021.

Wenfeng Zheng, Yu Zhou, Shan Liu, Jiawei Tian, Bo Yang, and Lirong Yin. A deep fusion matching network semantic reasoning model. *Applied Sciences*, 12(7):3416, 2022.

Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Object detectors emerge in deep scene cnns. *arXiv preprint arXiv:1412.6856*, 2014.

Bolei Zhou, Yiyou Sun, David Bau, and Antonio Torralba. Interpretable basis decomposition for visual explanation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 119–134, 2018a.

Bolei Zhou, Yiyou Sun, David Bau, and Antonio Torralba. Revisiting the importance of individual units in cnns via ablation. *arXiv preprint arXiv:1806.02891*, 2018b.

Ce Zhou, Qian Li, Chen Li, Jun Yu, Yixin Liu, Guangjing Wang, Kai Zhang, Cheng Ji, Qiben Yan, Lifang He, et al. A comprehensive survey on pretrained foundation models: A history from bert to chatgpt. *arXiv preprint arXiv:2302.09419*, 2023.

Xunjie Zhu, Tingfeng Li, and Gerard Melo. Exploring semantic properties of sentence embeddings. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 632–637, 2018.

Yutao Zhu, Huaying Yuan, Shuting Wang, Jiongnan Liu, Wenhan Liu, Chenlong Deng, Zhicheng Dou, and Ji-Rong Wen. Large language models for information retrieval: A survey. *arXiv preprint arXiv:2308.07107*, 2023.