Deep Neural Model Inspection and Comparison via Functional Neuron Pathways

James Fiacco
Language Technologies Institute
Carnegie Mellon University
jfiacco@cs.cmu.edu

Samridhi Choudhary*
Alexa Machine Learning
Amazon
samridhc@amazon.com

Carolyn P. Rosé
Language Technologies Institute
Carnegie Mellon University
cprose@cs.cmu.edu

Abstract

We introduce a general method for the interpretation and comparison of neural models. The method is used to factor a complex neural model into its functional components, which are comprised of sets of co-firing neurons that cut across layers of the network architecture, and which we call neural pathways. The function of these pathways can be understood by identifying correlated task level and linguistic heuristics in such a way that this knowledge acts as a lens for approximating what the network has learned to apply to its intended task. As a case study for investigating the utility of these pathways, we present an examination of pathways identified in models trained for two standard tasks, namely Named Entity Recognition and Recognizing Textual Entailment.

1 Introduction

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). As a response to this, the contribution of this paper is a new method of network interpretation that enables a more abstract view of what a network has learned, which we refer to as neural pathways. 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 can then be more effectively associated with 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 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 the paper we review common techniques for network interpretation followed by a detailed description of the neural pathways approach. Next, we apply the neural pathways approach to previously published neural models,

* Work was done as a graduate student at Carnegie Mellon University.

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.

2 Related Work

Our work falls under the broad topic of neural network interpretation. Recently, in this area of research a wide variety of models have been the target of investigation, including additive classifiers (Poulin et al., 2006), kernel-based classifiers (Baehrens et al., 2010), hierarchical networks (Landecker et al., 2013), and many others that are too numerous to list. As our work focuses on interpretation, we are not presenting new state-of-the-art performance on a given task, but rather a new method to understand and compare neural models. Our evaluation is a demonstration that focuses on models trained for the Named Entity Recognition and Recognizing Textual Entailment tasks. The specific goal of our evaluation will be to demonstrate the broad applicability of the approach, and position it as building on and extending the existing body of work exploring interpretability of previously defined neural models (Glockner et al., 2018; Mudrakarta et al., 2018).

We observe that neural interpretation approaches fall within several broad categories: visualizations and heatmaps (Karpathy et al., 2015; Strobelt et al., 2016), gradient-based analyses (Potapenko et al., 2017; Samek et al., 2017b; Bach et al., 2015; Arras et al., 2017), learning disentangled representations during training (Whitney, 2016; Siddharth et al., 2017; Esmaeili et al., 2018), and model 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). Our work uses linear probes as a method to identify the function of groups of neurons that are correlated with linguistic and task-level features, rather than for interpretation of individual neurons. Through correlation with the pathway analysis, we can furthermore reason about the role that those linguistic and task-level features have in the network's predictions.

Recent attempts to understand the functioning of trained neural models have 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. However, making meaningful patterns apparent in these visualizations can be highly dependent on the artful arrangement of the data presented within them, and it is easy to overlook patterns that are not immediately obvious. 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 have been used by Alain and Bengio (2016) 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.

More recently, Montavon et al. (2017) published a detailed tutorial on the recent approaches and techniques of interpreting deep neural networks. They identified cross-cutting techniques that have been applied to explain the behavior of a wide range of models. A notable contribution of this tutorial is an approach for sensitivity analysis capable of identifying important input features to a network. The technique observes the magnitude of the gradient for each input feature for each data point, giving relevance scores per data point for each feature. Analogous methods for accomplishing similar goals include *layer-wise relevance propagation* (Bach et al., 2015) and its derivatives (Samek et al., 2017a; Arras et al., 2017).

While these approaches have mainly focused on explaining the predictions and performances of a single network at a time, few if any prior attempts have been made to use these techniques for comparison across different network architectures, as we do in this paper.

3 Methodology

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

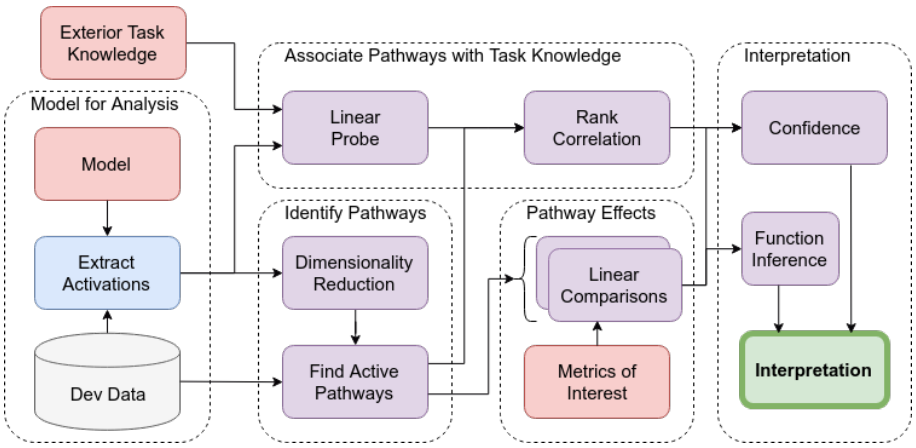

Figure 1: Flowchart representation of neural pathway based model interpretation.

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 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 4.1 and 4.2 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 3.3, Linear Comparisons. Example metrics include disagreement between models, incorrectly predicted values, or other task specific metrics.

Model and Data: The proposed 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: Our 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 4.2). 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 4.1). 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).

3.2 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.

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 not of interest in this foundational 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), 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.

3.3 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.

3.4 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 4.1, 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 3.2 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.

3.5 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 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.

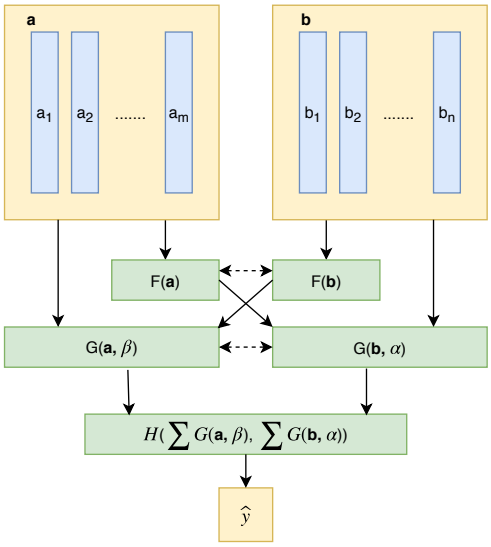

Figure 2: Decomposable Attention Model. Dotted arrows indicate networks with shared weights.

4 Experiments

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.

4.1 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. (2016). 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 work, 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., 2018). 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. (2019) 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.

4.2 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., 2017).

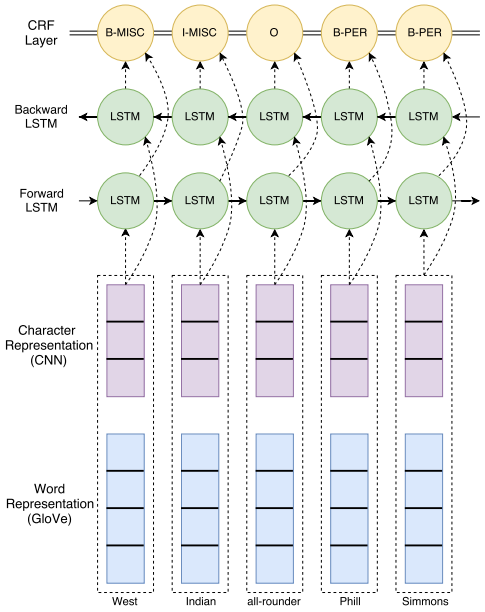

Figure 3: End-to-end model architecture for neural SOTA described in Ma and Hovy (2016). The character representation is computed by a CNN over the characters of the word. This is concatenated with the word embedding (initialized with GloVe) and fed into a BiLSTM. A CRF layer does a sequential decoding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

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-

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.

of-speech tags (extracted using *nltk*[1]); and *Other* which includes miscellaneous features like End-of-Sentence markers, hyphenated words, among others.

5 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.

5.1 Identifying Pathways

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.

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.

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.

5.2 Evaluating Pathway Effects

Entailment: 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

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

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 $p < 0.05$.

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.

NER: 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 be-

tween the two models' predictions have an average of 0.025 higher correlation coefficient ($p < 0.001$) than the BiLSTM-BiLSTM-CRF.

5.3 Associating Pathways With Task Knowledge

Linear Probes: 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.

For the NER task, 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.

Rank Correlation: 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.

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 the CNN based character encoder.

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 $p < 0.001$.

5.4 Interpretation

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).

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.

6 Conclusions

In this paper, we have demonstrated an approach for neural interpretation using neural pathways on recognizing textual entailment and named entity recognition. 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. 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 much simpler starting point for a broad class of models.

Future work can improve this method further by examining the effects of different dimensionality reduction methods with varying properties on extracting the most informative pathways from the activations.

Acknowledgements

This work was funded in part by NSF grant IIS 1822831. We would also like to thank Shruti Rijhwani for her help with the implementations for the NER models.

References

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. 2015. TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

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

Guillaume Alain and Yoshua Bengio. 2016. Understanding intermediate layers using linear classifier probes. *arXiv preprint arXiv:1610.01644*.

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

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

David Baehrens, Timon Schroeter, Stefan Harmeling, Motoaki Kawanabe, Katja Hansen, and Klaus-Robert MÃžller. 2010. How to explain individual classification decisions. *Journal of Machine Learning Research*, 11(Jun):1803–1831.

Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015. 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.

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

Alexis Conneau, Germán Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. 2018. 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.

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

Jamie DeCoster. 1998. Overview of factor analysis.

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

Max Glockner, Vered Shwartz, and Yoav Goldberg. 2018. 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.

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

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A Smith. 2018. 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.

Harold Hotelling. 1933. Analysis of a complex of statistical variables into principal components. *Journal of educational psychology*, 24(6):417.

Eric Jones, Travis Oliphant, Pearu Peterson, et al. 2001–. SciPy: Open source scientific tools for Python. [Online; accessed ¡today¿].

Andrej Karpathy, Justin Johnson, and Li Fei-Fei. 2015. Visualizing and understanding recurrent networks. *arXiv preprint arXiv:1506.02078*.

C Kennedy, MH Des Rosiers, JW Jehle, M Reivich, F Sharpe, and L Sokoloff. 1975. Mapping of functional neural pathways by autoradiographic survey of local metabolic rate with (14c) deoxyglucose. *Science*, 187(4179):850–853.

Urvashi Khandelwal, He He, Peng Qi, and Dan Jurafsky. 2018. Sharp nearby, fuzzy far away: How neural language models use context. *arXiv preprint arXiv:1805.04623*.

Sanjay Krishnan and Eugene Wu. 2017. Palm: Machine learning explanations for iterative debugging. In *Proceedings of the 2nd Workshop on Human-In-the-Loop Data Analytics*, page 4. ACM.

Adhiguna Kuncoro, Chris Dyer, John Hale, Dani Yogatama, Stephen Clark, and Phil Blunsom. 2018. 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.

Will Landecker, Michael D Thomure, Luís MA Bettencourt, Melanie Mitchell, Garrett T Kenyon, and Steven P Brumby. 2013. Interpreting individual classifications of hierarchical networks. In *Computational Intelligence and Data Mining (CIDM), 2013 IEEE Symposium on*, pages 32–38. IEEE.

Nelson F Liu, Roy Schwartz, and Noah A Smith. 2019. Inoculation by fine-tuning: A method for analyzing challenge datasets. *arXiv preprint arXiv:1904.02668*.

Xuezhe Ma and Eduard Hovy. 2016. 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.

Grégoire Montavon, Wojciech Samek, and Klaus-Robert Müller. 2017. Methods for interpreting and understanding deep neural networks. *Digital Signal Processing*.

Pramod Kaushik Mudrakarta, Ankur Taly, Mukund Sundararajan, and Kedar Dhamdhere. 2018. Did the model understand the question? In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1896–1906.

Aakanksha Naik, Abhilasha Ravichander, Norman Sadeh, Carolyn Rose, and Graham Neubig. 2018. Stress test evaluation for natural language inference. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2340–2353.

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. 2017. Dynet: The dynamic neural network toolkit. *arXiv preprint arXiv:1701.03980*.

Ankur Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. 2016. A decomposable attention model for natural language inference. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2249–2255.

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. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Adam Poliak, Jason Naradowsky, Aparajita Haldar, Rachel Rudinger, and Benjamin Van Durme. 2018. Hypothesis only baselines in natural language inference. In *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*, pages 180–191.

Anna Potapenko, Artem Popov, and Konstantin Vorontsov. 2017. Interpretable probabilistic embeddings: bridging the gap between topic models and neural networks. In *Conference on Artificial Intelligence and Natural Language*, pages 167–180. Springer.

Brett Poulin, Roman Eisner, Duane Szafron, Paul Lu, Russell Greiner, David S Wishart, Alona Fyshe, Brandon Pearcy, Cam MacDonell, and John Anvik. 2006. Visual explanation of evidence with additive classifiers. In *Proceedings Of The National Conference On Artificial Intelligence*, volume 21, page 1822. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999.

Alec Radford, Rafal Jozefowicz, and Ilya Sutskever. 2017. Learning to generate reviews and discovering sentiment. *arXiv preprint arXiv:1704.01444*.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. Why should i trust you?: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1135–1144. ACM.

Wojciech Samek, Alexander Binder, Grégoire Montavon, Sebastian Lapuschkin, and Klaus-Robert Müller. 2017a. Evaluating the visualization of what a deep neural network has learned. *IEEE transactions on neural networks and learning systems*, 28(11):2660–2673.

Wojciech Samek, Thomas Wiegand, and Klaus-Robert Müller. 2017b. Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models. *arXiv preprint arXiv:1708.08296*.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. 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.

Xing Shi, Kevin Knight, and Deniz Yuret. 2016a. Why neural translations are the right length. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2278–2282.

Xing Shi, Inkit Padhi, and Kevin Knight. 2016b. Does string-based neural mt learn source syntax? In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1526–1534.

Narayanaswamy Siddharth, T Brooks Paige, Jan-Willem Van de Meent, Alban Desmaison, Noah Goodman, Pushmeet Kohli, Frank Wood, and Philip Torr. 2017. Learning disentangled representations with semi-supervised deep generative models. In *Advances in Neural Information Processing Systems*, pages 5925–5935.

Charles Spearman. 1904. The proof and measurement of association between two things. *The American journal of psychology*, 15(1):72–101.

Hendrik Strobelt, Sebastian Gehrmann, Bernd Huber, Hanspeter Pfister, and Alexander M Rush. 2016. Visual analysis of hidden state dynamics in recurrent neural networks. Technical report, Harvard University OpenScholar.

Maksim Tkachenko and Andrey Simanovsky. 2012. Named entity recognition: Exploring features. In *KONVENS*, pages 118–127.

William Whitney. 2016. *Disentangled representations in neural models*. Ph.D. thesis, Massachusetts Institute of Technology.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. 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.

Xunjie Zhu, Tingfeng Li, and Gerard Melo. 2018. 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.