NJM-Vis: Interpreting Neural Joint Models in NLP

David Johnson
davewj@gmail.com
University of British Columbia

Giuseppe Carenini
carenini@cs.ubc.ca
University of British Columbia

Gabriel Murray
gabriel.murray@ufv.ca
University of the Fraser Valley

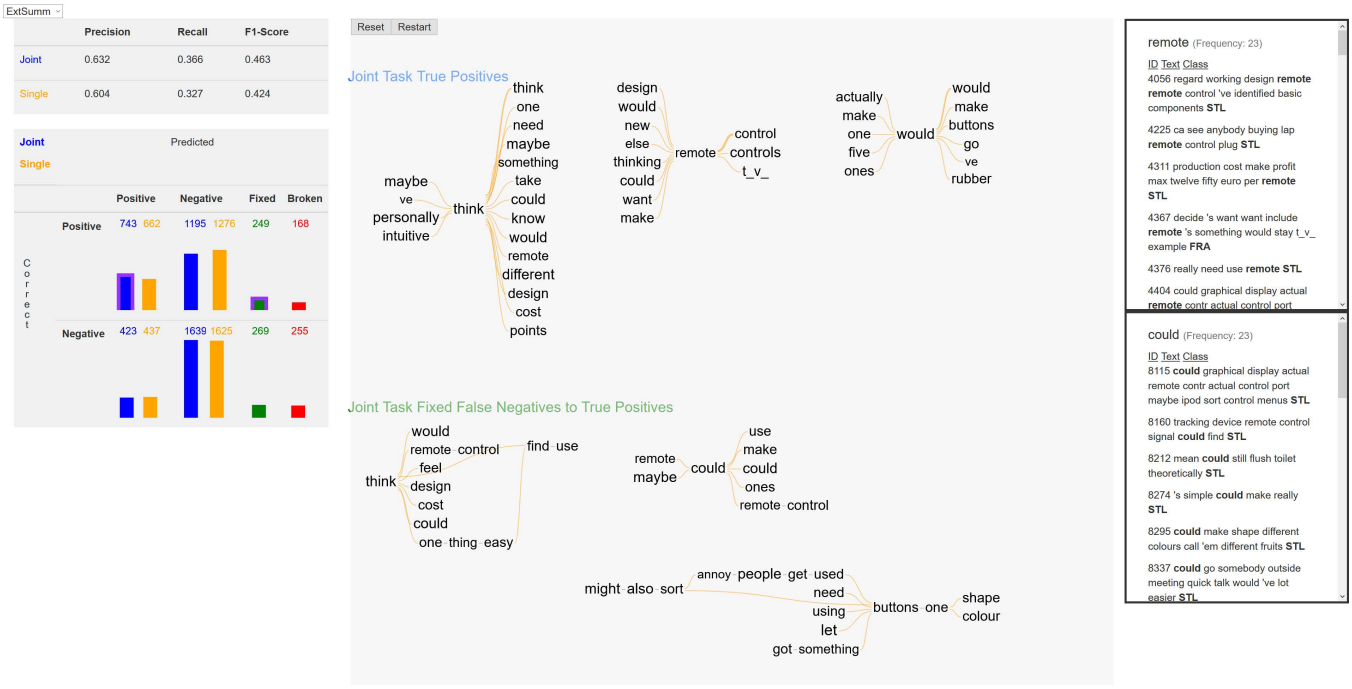

Figure 1: *NJM-Vis* interface. The left of the interface shows the score panel, displaying individual model performance. The middle of the interface is the vis panel, displaying a word graph visualization indicating words which are relevant to model prediction. The right side of the interface is the sentence browser panel. After clicking words that appear in the vis panel, the sentence browser panel is populated with sentences from the dataset which contain the selected word.

ABSTRACT

Neural joint models have been shown to outperform non-joint models on several NLP and Vision tasks and constitute a thriving area of research in AI and ML. Although several researchers have worked on enhancing the interpretability of single-task neural models, in this work we present what is, to the best of our knowledge, the first interface to support the interpretation of results produced by joint models, focusing in particular on NLP settings. Our interface is intended to enhance interpretability of these models for both NLP practitioners and domain experts (e.g., linguists).

CCS CONCEPTS

• **Human-centered computing** → *Information visualization*; *Visualization toolkits.*

KEYWORDS

explainable artificial intelligence, deep learning, information visualization

ACM Reference Format:
David Johnson, Giuseppe Carenini, and Gabriel Murray. 2020. NJM-Vis: Interpreting Neural Joint Models in NLP. In *25th International Conference on Intelligent User Interfaces (IUI '20), March 17–20, 2020, Cagliari, Italy.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3377325.3377513

IUI '20, March 17–20, 2020, Cagliari, Italy

ACM ISBN 978-1-4503-7118-6/20/03…$15.00
https://doi.org/10.1145/3377325.3377513

1 INTRODUCTION

Deep learning approaches have recently shown great potential on a large number of key prediction problems. However, while this progress has been achieved by mainly focusing on one specific task at a time, it is clear that more powerful solutions can be developed by building joint models, where dependencies between multiple tasks can be effectively exploited [30]. These joint models are already outperforming non-joint models on several important tasks and

have become a thriving area of research in AI and Machine Learning, especially when applied to Natural Language Processing (NLP) and Computer Vision. For instance, in Computer Vision, jointly learning histogram of oriented gradient features, deformation handling, and occlusion handling can improve a pedestrian detection system over one that learns each task individually [40]. In NLP, named-entity recognition can use part-of-speech tags as features, so improving the accuracy of a part-of-speech tagger can improve the results of a named-entity recognition model, and vice versa [8]. Similarly, discourse parsing can often be combined with other NLP tasks in which improvements in learning discourse parsing can improve learning in a joint task such as sentiment analysis [39].

Although a strength of deep learning is in learning effective representations of data, the complexity and distributed nature of such representation makes explanation for deep neural networks notoriously difficult [15]. This is arguably even more challenging for joint models, which tend to be much more complex given that the models have shared layers. It may not be clear how much one task contributes to the learning of shared layers over another task, and therefore not clear how much impact one task had on the output of another task. Several researchers have worked on enhancing the transparency/interpretability of single-task neural models typically by visualizing feature optimization. In Computer Vision this can be accomplished by continually synthesizing images which cause higher and higher neuron activations, eventually finishing with an image synthesized to maximally activate neurons. Though these preferred input images rarely look like natural images, they can be used to determine what a neuron layer has learned to detect [46][12][16]. In addition to feature optimization methods, there are also attribution methods such as Layerwise Relevance Propagation (LRP) [3] and Saliency attribution [27]. These methods try to attribute a neuron's relevance to the neural model's output, and are more appropriate for domains other than Computer Vision. Attention mechanisms [4] have also been considered for supporting the interpretation of neural models [26], but they have been recently criticized for being rather unreliable predictors of input relevance (e.g., [19])

In this work we present what is, to the best of our knowledge, the first visual interface to support the interpretation of results produced by Neural Joint Models (*NJM-Vis*). In particular, we focus on supporting the understanding of the benefits that one task is bringing to the other in NLP settings by relying on the LRP attribution method. *NJM-Vis*, shown in Figure 1, comprises two views in a multiform overview/detail design [35], in which one view shows an overview of the results as confusion matrices, while the other allows the user to explore details of the results through an adaptation of a sentence visualization tool [18].

As running examples, we use two NLP joint models: Summ-DiaAct and Bot-FakeNews. In Summ-DiaAct, an extractive summarization task [36][37] is jointly performed with a dialog act prediction task [10] on conversational data. The goal of extractive summarization is to classify each sentence as important (extract-worthy) or not, while the goal of dialog act prediction is to predict the speaker intention associated with an utterance, e.g. question, answer, inform. In Bot-FakeNews, a bot detection task [24] is jointly trained with a fake news detection task [44]. For bot detection, the goal is simply to classify a tweet as coming from either a bot Twitter account, or real user account. The goal of the fake news detection task is to classify a tweet as either verifiable fact, or rumour. In this work, we have implemented a joint neural model for both Summ-DiaAct and Bot-FakeNews and used the results produced by such models as inputs to our interface.

Users of visualization tools for deep learning can be categorized into three overlapping groups [17]: model developers, model users, and non-experts. Our system is designed to assist an overlap of model developers and model users. Model developers understand deep learning thoroughly and use systems like Tensorboard [1], Deep Eyes [42], and Blocks [5] to interpret the underlying neural model, to debug or improve it. In contrast, model users may have less or no experience implementing deep learning solutions, but employ neural networks as a means of developing domain-specific applications. Systems built for these users include ActiVis [21], and LSTMVis [43]. In this work we will specify the terms "model developers" or "model uers" when speaking of only one or the other, and use the term "users" to refer generally to users that could be either model developers or model users.

Our main goal is to enhance the ability of users to interpret the benefits of a joint task model compared to a single task model by allowing them to inspect the predictions of the joint task model; to assess how the joint task models differ from the single task models; and, more importantly, to evaluate the reasons why these predictions are different.

To assess the strengths and weaknesses of our initial prototype, we have run a formative evaluation as a case study with four model user participants. In these studies, we have used our two joined models: Summ-DiaAct and Bot-FakeNews.

2 RELATED WORK

In this section, we discuss neural joint models, interfaces for interpreting both single and joint task neural models, saliency interpretation methods, and word cloud style visualizations.

2.1 Neural Joint Models

Neural joint models come in two alternative forms: multi-tasking and pre-training. Pre-training completes the training of one task and then uses the learned weights to initialize the weights for a second task. This has been shown by Erhan et al. [11] to result in better generalization and better performance than the typical manner of random weight initialization. Another style of joint model is multi-tasking [8], where the training process proceeds by feeding training examples from alternating tasks allowing the neural model to jointly learn multiple tasks. Multi-tasking has been successfully applied in multiple areas, such as NLP [29] and computer vision [22]. In this work, we use multi-tasking, as it tends to outperform pre-training (e.g., [39] in NLP, joining discourse parsing and sentiment).

2.2 Interfaces to Interpret Neural Models (Single Task):

Much of the previous work creating interfaces for visualizing deep neural networks DNNs has been done on computer vision tasks using Convolutional Neural Networks (CNN). Yosinski et al. [46] describes an interface allowing visualization of plotted convolutional layer activation values. Similarly Liu et al. [28] presents an

interface for visualizing CNNs by converting a CNN to a directed acyclic graph and clustering neurons in each layer of the network before adding an edge-bundling visualization to show an overview of the whole network. Both of these do support interpretation of the underlying neural models, but they are both focused on CNNs and vision tasks, unlike our goal of using feed-forward DNNs on textual tasks. Some recent work has explored visual interfaces for understanding neural NLP (e.g., [27], [17], [38]), but they are limited to single-task models, while our goal is supporting the interpretation of joint-models.

Zhang et al. [47] is likely the interface for interpreting neural models most similar to ours. However, it is not intended to directly compare joint task models as our interface does, so it does not directly display to the user data instances which were fixed or broken by the process of joint training.

2.3 Saliency Interpretation Method

In Li et al. [27] salience is used to measure the amount a neural unit contributes to the meaning compositionality (building sentence meaning from the meaning of words or phrases) using first-order derivatives. In this way, the authors are able to show explanations for the difference in performance on sentiment analysis tasks between a recurrent neural network (RNN), long short-term memory network (LSTM), and bi-directional LSTM. Although saliency methods contribute to understanding of neural models, it was shown in Arras et al. [2] that they are less effective than LRP methods, as they are unable to establish when words are inhibiting a prediction decision as LRP is capable of doing. More generally, LRP has been successfully applied in several domains [33], most recently to healthcare [6]. Another popular option for interpreting deep learning results is using attention mechanisms [26]. However, they have been recently shown to be rather poor predictors of input relevance (e.g., [19]). In this work, we rely on LRP to explain both single task and joint task predictions.

2.4 Word Cloud Visualization Techniques

In their original form, word clouds summarize the content of a text or a set of documents as an image of words, where the size of each word corresponds to its frequency or any other measure of importance [23]. While a large number of variations have been proposed (e.g., [20]), the most relevant to our work is SentenTree [18], a visualization that includes links between words in the word cloud, indicating that words co-occur in sentences together. Our work uses the underlying SentenTree algorithm, but builds on top of the original SentenTree by changing SentenTree to allow word size to indicate the score of a contribution measure, such as LRP. Additionally, we split the SentenTree visualization to allow two sets of SentenTree visualizations, supporting our users in comparing two selections of subsets of data at the same time (for instance, our users can compare true positives from the joint task, and true positives of the single task at the same time).

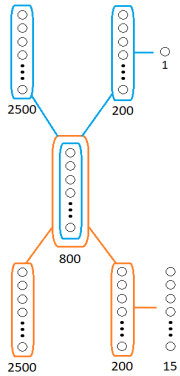

Figure 2: Network architecture for Summ-DiaAct. The blue layers corresponds to the extractive summarization task, the orange layer corresponds to the dialogue act task, and the overlapping color shows the layer which is shared and jointly learned between the two tasks.

3 NLP DATASETS, EMBEDDINGS, MODELS AND RESULTS

Dataset: For the Summ-DiaAct joint model, we use the AMI corpus as our dataset [7]. AMI is a multi-modal dataset containing transcripts of group meetings. AMI has annotations for multiple tasks such as dialogue act, topic segmentation, abstractive and extractive summarization, named entities etc. We use the annotations for the dialogue act (15 types) and extractive summarization (binary) tasks, as these two were shown to benefit from joint training [41].

For the Bot-FakeNews joint models, we used two separate datasets, both of which are comprised of tweets from Twitter. For bot detection, we used a dataset from Cresci et al. [9], which contains both tweets from genuine accounts, as well as from accounts identified as bots in [45]. For the fake news task, we used a dataset from Zubiaga et al. [48], in which the authors enlisted a team of journalists to identify when a newsworthy event was occurring, at which case the authors collected tweets associated with the event. The journalists then went through the collected tweets and identified them as either factual or rumour.

Word Embeddings: Both models use Word2Vec [31] at 100 dimensions. Each sentence is tokenized into individual words, and stop-words are removed. The Summ-DiaAct model has sentences pruned at 25 words maximum length. Any sentence less than 25 words is padded with 0s. This gives embeddings for each sentence of 2500 dimensions. The Bot-FakeNews model has sentences pruned at 20 words maximum, or padded with 0s for sentences less than 20 words, giving 2000 dimensions.

Model Architecture: Both networks were developed using Tensorflow [1] with an LRP implementation adapted from Lapuschkin et al. [25]. The network for Summ-DiaAct is shown in Figure 2, with the intermediate layers using ReLU activation functions. The output layer for the extractive summarization task applies a sigmoid activation function, while the dialogue act task applies a softmax activation function. The network for Bot-FakeNews is shown and

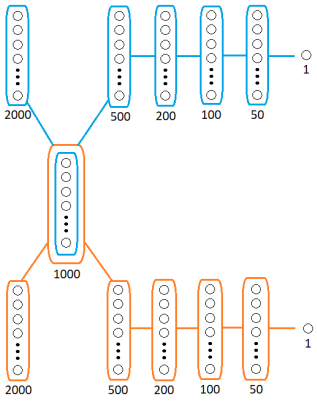

Figure 3: Network architecture for Bot-FakeNews. The blue layers corresponds to the bot detection task, the orange layer corresponds to the fake news detection task, and the overlapping color shows the layer which is shared and jointly learned between the two tasks.

Model	F-score	Precision	Recall
Single	.424	.604	.327
Joint	.463	.632	.366

Table 1: Single & Joint Task Results for Extractive Summarization

Micro Average (Single)	Micro Average (Joint)
.711	.714

Table 2: Single & Joint Task Training Results for Dialogue Act

described in Figure 3. The intermediate layers use ReLUs, while both output layers use sigmoid activation functions.

Model Learning: a classifier learns a hidden layer as a shared representation for the two tasks. The weights are initialized with a Xavier initialization [14] since activations chosen from a random normal distribution tended to cause neuron saturation within our DNN. To train, the network uses a multitasking approach [39] by randomly choosing which task to train in each epoch. Once the training is completed, the shared layer will have learned weights beneficial for both tasks.

Results: As shown in Table 1, the results for extractive summarization improve when trained in the Summ-DiaAct joint model. In contrast, Table 2 indicates that for the dialogue act task the improvement is negligible. When looking at the Bot Detection and Fake News detection tasks, the Bot-FakeNews joint model outperforms the single model for both tasks as shown in tables 3 and 4 .

4 *NJM-VIS* DATA AND TASK ABSTRACTIONS

By following the standard methodology for designing visual interfaces as in Munzner [34], we base the design of *NJM-Vis* on

Model	F-score	Precision	Recall
Single	.75	.88	.66
Joint	.96	.97	.93

Table 3: Single & Joint Task Results for Bot Detection

Model	F-score	Precision	Recall
Single	.76	.86	.70
Joint	.81	.90	.72

Table 4: Single & Joint Task Results for Fake News Detection

abstracted data and task models. The data model describes information about sentences and words that we need to compute and store, while the task model outlines key analysis tasks to support the interpretation of joint models and their comparison with single models.

4.1 Data Model

The data model for the four classification tasks comprises tables containing information associated with sentences and with words. For each sentence in the datasets, we need to store all its words and their corresponding embeddings. Additionally, for each sentence and for each task we need the prediction of the joint model, of the single model, and the gold-standard label. Moving to words, for each word we need a measure of its contributions to each possible prediction for the sentence containing that word (across models and tasks). So for instance, for the word "remote" in the sentence "we do not include a remote" in the AMI corpus, we would need a measure of its contribution to the prediction of that sentence being summary-worthy and of its dialog act type, in the single and in the joint models.

With respect to computing how much a word contributes to a neural prediction, there are multiple possible methods. Our goal is to explain the prediction for a classification problem such that given an input vector x we would like to know how the features of x (the words in our tasks) contribute toward our classification prediction, and in what way they contribute to our prediction.

Predictions of DNNs can be explained by decomposing the output of the network on the input variables. *NJM-Vis* uses this form of explanation through a method known as Layerwise-Relevance Propagation (LRP)[3] to explain the output of the model. LRP propagates the relevance of the output backward through the network, distributing the relevance layer by layer in proportion to how much each neuron in the layer contributed to the output, until reaching the input layer where the relevance is finally distributed among the input neuron (the words in our tasks) in proportion to how much each contributed to the output, giving us how relevant each part of the input was on the output from the network. Using this relevance, we can determine whether a particular part of the input was able to contribute for or against (and whether the contribution was weak or strong). Let the neurons of the network be:

$$a_k = g(\sum_j a_j w_{jk} + b) \quad (1)$$

where a_k is the neuron activation, g is an activation function which is positive and monotonically increasing, a_j are the activations from

the previous layer, w_{jk} is the weights of the neuron, and b is the bias parameter. The rule to propagate relevance [32] is:

$$R_j = \sum_k \frac{a_j w_{jk}^+}{\sum_j a_j w_{jk}^+} \hat{R_k} + \frac{a_j w_{jk}^-}{\sum_j a_j w_{jk}^-} \check{R_k} \quad (2)$$

where $\hat{R_k} = \alpha R_k$ and $\check{R_k} = -\beta R_k$ with α and β chosen to constraints $\alpha - \beta = 1$ and $\beta >= 0$

Although in this work we have used LRP as a means of determining each word's relevance to the predicted output, our interface does not require that LRP is used; an alternative importance measure could be applied.

4.2 Task Model

The high-level goal of *NJM-Vis* is to support model developers and model users [17] in interpreting the benefits of joint task neural model predictions, when compared to single task models. Given that this is a comparison task, we referred to existing literature on visualizing comparisons. For example, in Gleicher [13], comparison tasks are grouped abstractly as the following actions: Identify, Measure, Dissect, Connect, Contextualize, and Communicate. We also went through an informal iterative collection of user requirements from NLP experts (including the authors). The following tasks are intended to be supported by the interface.

- **(T1) Measure predictions of the two models quantitatively**
  Example: Measure Precision, Recall, and F-score for both models, or how many predictions are "fixed"/"broken" by the models.
  Our elicited user requirements determined that both model users and model developers want to determine whether the joint task training actually improves predictive performance over the single task training by a measurable amount.
- **(T2) Identify key words in subsets of predictions**
  Example: Identify that "dollars" is often appearing in true positive predictions, and is therefore a key word. Alternatively, a word may be considered a key word if it has a high contribution measure in a subset of predictions, or if it often co-occurs with other words in subsets of predictions.
  It was determined from our user requirements that the ability to identify words which are important to prediction subsets could be a valuable first step in understanding the predictive difference between joint and single task models. Identifying key words allows the user to gain an overview of potentially important differences between the models which the user can then begin to explore more in-depth (such as in T3 and T5).
- **(T3) Dissect linguistic similarities/differences between single and joint predictions**
  Example: Determine that key words appearing in true positive predictions are often pronouns.
  Once the user has identified which words are key words, they may want to move to more in-depth analysis of those words in an attempt to discover linguistic similarities/differences and, through that analysis, gain an improved understanding of the compared tasks. For example, perhaps a user finds that by jointly training extractive summarization with dialog act prediction, their extractive summarization performance improves, and through the dissection of linguistic properties of key words between the models they discover that words which are pronouns often show up in true positives for the joint task but not single task trained model. This would indicate that pronouns may have predictive power to both tasks, and that by jointly training the two tasks, the network is better able to learn a representation which accounts for the importance of pronouns. The user then is able to gain knowledge about the tasks themselves, i.e. that pronouns may be important to predicting whether a sentence is extract worthy, but only when the sentence is expressing particular dialog acts.
- **(T4) Identify possible errors in results**
  Example: Identify that "ve" has a high frequency, which may be an error left over from pre-processing words like "i've", "should've", etc.
  In our elicitation of requirements from model developers it was determined that developers, since they often manually build model architectures, want to be able to easily identify possible errors in the pre-processing and training phase. Though this is also useful for model users, it may be more difficult for model users to understand the technical details of errors than the more experienced model developers, and therefore the means of identifying errors to model users may need to be more intuitive than measures like gradient values, etc.
- **(T5) Identify key relationships between predicted class labels**
  Example: Identify that a particular predicted class label for the input task 1 is often appearing in subsets of predictions for the input task 2.
  Identifying key relationships between predicted class labels is an important user requirement for understanding the difference between the two compared models. Consider a user with the tasks extractive summarization and dialog act prediction in which the user finds that sentences with the dialog act "suggest" label are appearing more often in the jointly trained extractive summarization model than in the single trained model. The user could then infer that perhaps spoken suggestions have predictive power for whether a sentence should be included in an abstract, and the act of joint training helped the extractive summarization network learn a representation which accounts for this linguistic property. Similar to T3, the user then gains understanding about both the tasks themselves, as well as the predictive differences between the joint and single task models.
- **(T6) Contextualize predictions at the granularity of sentences**
  Example: Learning that the key word "schedule" in true positives often appears in sentences with the modal verb "must", potentially indicating that "must" may also have predictive power when with "schedule" for extractive worthy sentences.
  It was discovered through our user requirement elicitation that strictly showing a key word may not always be enough context to understand the word's importance to predictions.

It was determined that an important user task is to be able to view a key word in its full sentence allowing the user to analyze the complete context of the input to the network. This context could lead the user to a deeper understanding of the linguistic properties which caused the model prediction.

5 DESIGN SOLUTION

NJM-Vis is faceted into multiple views of coordinated visualizations. As seen in Figure 1, the left side of the interface, the score panel (Fig. 4), supports T1 by summarizing and comparing the predictions of the joint (blue) and single (orange) neural models. In the top of the score panel Precision, Recall, and F-Scores are shown in a table format for both the joint and single task versions of the model. The rows of the table list the joint and single task, in which joint is in blue and single is in orange. The bottom of the score panel shows a confusion matrix with aligned bar-charts. True positive, false positive, false negative, and true negative subsets of dataset examples comprise the aligned bar-charts. The aligned bar-charts also include green bars for examples which were fixed by the joint training process (i.e., from false negative to true positive and from false positive to true negative). Similarly, examples which were broken by the joint training process (i.e., from true positive to false negative and from true negative to false positive) are shown in red. If the user clicks a bar the bar will be highlighted with a purple outline, as seen by the purple highlight on the joint task true positive bar in (Fig. 4).

The middle view allows comparison of two selections by juxtaposition, placing the first selected subset along the top of the view, and the second selected subset along the bottom of the view, allowing a user to view and compare selections such as true positive from the joint task and true positive from the single task.

Clicking any of the subsets in the bar chart view, such as true positive for the single task, or false negative for the joint task, brings up a word cloud style visualization in the Vis Panel (Fig. 5), adapted from Hu et al. [18]. This sentence browser is structured as a node-link graph diagram in which nodes are words and links represent words that co-occur in a sentence as shown in Fig. 1. The visualization also supports the following tasks:

- **T2:** Since words which appear in the visualization are words which have a high frequency in the selected subset, the visualization intrinsically identifies words which could be key words. Additionally, the visualization uses the size of words to encode a measure of how strongly a word contributed to the selected subset of predictions in which a larger word indicates it contributed more strongly to a prediction than a smaller word. This is also an indication that a word may be a key word. For instance, in Fig. 5 the word "new" appears in the middle word cloud in the joint task true positive subset, and is larger than other words in the subset. This indicates to the user that the word "new" is a key word for this subset. We use LRP as our measure of a word's contribution to prediction.
- **T3:** The visualization panel allows the user to make two selections and compare them directly, allowing the comparison of subsets of data instances for both the single and joint task at the same time. With the ability to have this juxtaposed comparison, the user can view and compare linguistic similarities or differences. The user can easily see if, for example, pronouns were often appearing in the joint task true positive predictions, but not the single task true positive predictions.
- **T4:** Because the visualization is built on frequent words, it's possible for the user to see cases in which a commonly appearing error is occurring in a subset of their data. For example, in Fig. 1 in the joint task true positive subset, in the right-most word cloud centering on "would" we see the word "ve" which seems to be potentially an error in the data pre-processing.

Figure 4: Score panel. The top of the panel shows Precision, Recall, and F-score for both the joint and single task. The bottom of the score panel shows a confusion matrix with aligned bar charts. Each bar chart represents the number of data instances which are categorized into each positive/negative subset (true positive, false positive, etc.). The bar charts also include fixed and broken subsets (i.e., from false negative in the single task, to true positive in the joint task and from false positive in the single task, to true negative in the joint task). As seen in the true positive subset, when a user clicks a subset the bar is highlighted in purple

Clicking any of the words in the middle view visualization brings up a scrollable list of sentences in the Sentence Panel on the right side of the view (Fig. 6), all of which are sentences from the user's dataset containing the word which was clicked on by the user from the selected subset in the middle view. The top right sentence view appears when a user clicks a word in a node tree in the top half of the middle view, and the bottom right sentence view appears when a user clicks a word in a node tree in the bottom half of the middle view. This allows users to directly compare sentences containing selected words between two selected subsets. This sentence panel supports the following tasks:

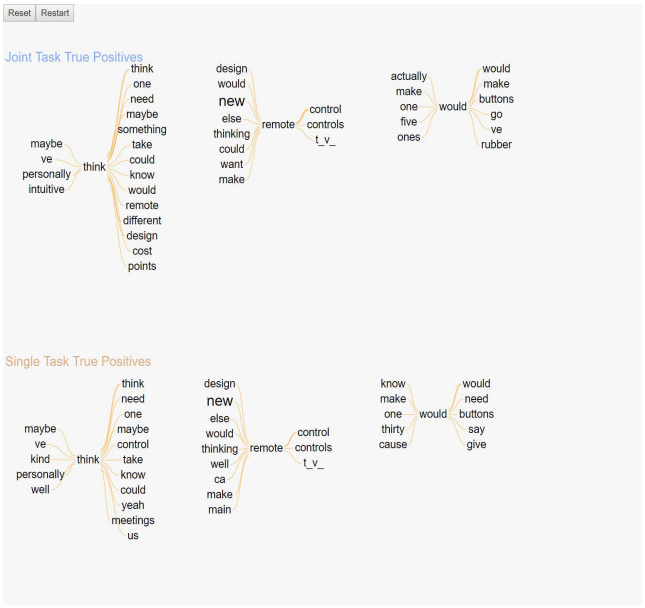

Figure 5: Vis panel. The vis panel contains two selections for direct comparison between user subset selections. The top shows Joint Task True Positives in blue, while the bottom shows Single Task True Positives in gold.

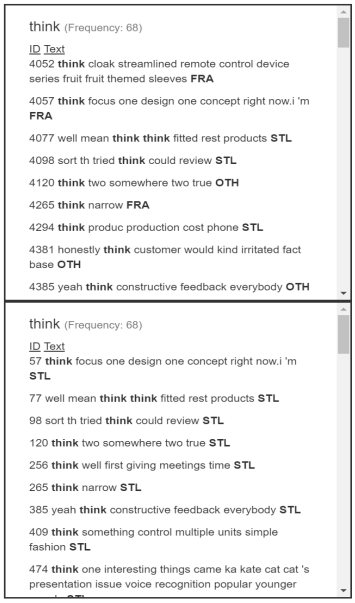

Figure 6: Sentence browser panel. The panels allow two user selections for direct comparisons. The sentences are those which contain the selected word, which is bolded in each sentence. At the end of each sentence is the name of the secondary task label written in bold uppercase text.

- **T5:** By including all of the secondary task class labels in the sentence panel, the user is able to see whether certain class labels for the user's secondary task are appearing often in their selected primary task subset. For example, in Fig. 6 in the bottom panel it appears many of the data instances have the secondary class label (in this case, dialogue act class label) of "STL" (shorthand for "stall"). This indicates to the user that perhaps the class label of "stall" has important predictive power for the selected subset of primary task predictions.
- **T6:** Since the sentence panel allows the user to scroll through the full sentences of the key words appearing in the vis panel, it allows the user to directly compare full sentences of their selected subsets. For example, users may choose to select true positive for joint task and true positive for single task and compare similarities and differences between the selections.

After running our case studies, we received feedback from our participants of how we may better be able to support our task model. The feedback is discussed in section 7 and potential interface additions and mock-ups are presented in section 8.

6 CASE STUDY

To assess the efficacy of the design, we ran a case study with a set of participants. The case study is intended to be part of an iterative design process. Future versions of the interface will be influenced directly by the case study feedback, at which point further case studies could be run, allowing additional feedback, and so on.

6.1 Method

The case study involved four participants. One participant was a postdoctoral researcher, while the others were graduate students. All of the participants were from a Computer Science background. The participants were split into two groups with participants 1 and 3 being assigned to the Summ-DiaAct joint tasks, and participants 2 and 4 being assigned to Bot-FakeNews.

After an initial explanation of the purpose and intention behind the interface, participants were walked through a short training session on a toy dataset of predictions. Participants were asked to answer simple questions (e.g., identify one high frequency word appearing in the joint model true positive subset, but not appearing in the single model true positive subset) by using the interface on the toy dataset. Once the users were able to correctly answer all the simple questions indicating that they understand the basic encodings and functions of the interface, they moved on to using the interface to explore their assigned dataset of predictions.

Participants were told to explore the predictions however they saw fit. They were told to write down any general insights that they gained from using the interface, as well as any insights gained about specifically why the joint task outperformed the single task.

6.2 Participant Results

In this section we provide the results for each of the four participants from the case study, categorized by common topics. The results include observations made about the participants use of the interface during the case study, as well as participant's feedback given through post-study questionnaires.

6.2.1 Interpretation and Comparison Facilitation. Participant 1 was able to make some inferences about the two tasks using the interface, for instance that modal verbs were often big, indicating that they contributed strongly to predictions. The participant commented that this "...seemed intuitive given that the dataset was a dialog dataset."

Participant 2 came up with multiple insights about the dataset, such as the fact that many of the tweets in the dataset were about the stock market, trading, money, and these were mostly predicted as true positives (i.e. that they are tweets from bots). Additionally, they commented that many of the true positives are tweets that mention blogs and other posts. They also determined that many of the tweets predicted to be from people did not cover a single overarching topic. Some observations about the word graph visualizations made by participant 2 were that: "The joint task true positives had deeper word cloud trees than the single task true positives" and that, "The single task false negatives had a lot of word overlap with the single task true positives, but that this was not the case for the joint task."

Participant 3 noted that through using the score panel and viewing the positive/negative confusion matrix they discovered that the joint task outperformed the single task in both the true positive and true negative categories.

Participant 4 mentioned that the interface enabled them to see that currency words such as "forex" were important for correctly classifying whether a tweet was from a bot. They also stated that because there were significantly more false negatives for the single task than for the joint task, that the single task model has trouble identifying what words are more "human".

6.2.2 Interaction Observations. Participant 1 began their task by comparing each subset directly between single and joint tasks, such as comparing true positives for both the joint and single tasks, and then false positives for both the joint and single task. Throughout this process, the participant made notes on paper (an indication that our interface should include a notepad functionality) about which words were large. Participant 1 focused primarily on the vis panel, not using the sentence browser panel at all until reminded of the functionality during the study.

Participant 2 started exploring much of the interface and compared many combinations of subsets against each other, both single versus joint, as well as single versus single and joint versus joint. They clicked around all aspects of the interface often, including clicking on many words to see the sentences in which they appeared. The participant quickly recorded insights about the datasets after only a few minutes of use. By the 10 minute mark of the study the participant had already recorded multiple insights about the datasets. The participant spent nearly the entire allotted time (35 of 40 allotted minutes for exploration) using the interface and continuously writing new insights on the datasets every few minutes.

Participant 3 first looked at strictly the bar charts first, spending time studying the charts before moving on to clicking the charts to view the visualizations. When looking at the visualizations, the participant began first by looking at True Positives and comparing both joint and single, and then moving to False Positives and comparing joint and single, and continued in that fashion until having looked at all the subsets. After fully exploring the whole interface carefully, the participant moved to recording all of their insights.

Participant 4 was focused on one selection at a time, clicking on many of the words and looking at sentences. They came up with insights within the first minute of using the interface. After continuing to use the interface the participant came up with multiple insights over the next few minutes. As the participant continued to use the interface, they clicked around often and used many aspects of the interface, quickly switching their attention between aspects of the interface. The participant made insights throughout their allotted time.

6.2.3 Participant Feedback: Strengths. In the post-study questionnaire participant 1 commented that they liked the concept of being able to compare the two models directly, and they found the vis design of using size to indicate importance useful. They commented that the size difference should be more notable so that there's a wider difference between small and large words. Participant 1 concluded the questionnaire by stating they would use this interface or a similar interface for their multi-task problems in the future if development was continued and the interface was further enhanced with their desired features

Participant 2 stated that the visual aspects of the interface, including the colours and clear organization of the score panel were, "very useful" and "...provided a fast and easy way to organize the results." Additionally the participant commented that being able to compare selections in juxtaposition was extremely useful. They concluded that they would use this or similar interfaces in the future for multi-task problems since the interface, "...is very good, easy to use, and convenient interface to see and do several things at once."

Participant 3 indicated in the post-study questionnaire that they liked the bar charts, and commented that they especially liked the fixed and broken columns. They commented that it took them a while to get used to the word graphs, but they appreciated that size was used to distinguish importance. They also liked that they were able to compare two sets of word groupings together to try to perform inference, though they felt the current design did lay a "fair bit" of cognitive load on the user, given that the design requires the user to click between subsets often to compare two at a time, and that they sometimes wanted to remember what the previous pairs of visualizations looked like while also viewing their current pairs of visualizations. Participant 3 said that they would want to use the interface in the future for other multi-task problems. They felt that it may have taken them longer than it should have to get used to the word graphs, but they felt that the graphs did eventually give them more insight into the model performance beyond what they would have got from simply looking at a confusion matrix. As a last note, the participant commented that they felt the interface might have use as a "sanity check" tool to determine, "...whether the dataset and annotations are any good."

Participant 4 stated in their post-study questionnaire that they liked that the interface showed influential words that contributed to each task for the positive and negative subsets. They also said they appreciated how the word graph denoted the importance of a word from its size, and they found that this design choice made looking at the visualization easy to breakdown and to understand. They felt that "most importantly" they liked they could compare the results of the joint and single task in a way that is more than "just a number

(i.e. accuracy)". They felt the word visualisations made it easier to understand what the deep models were looking at when making classification decisions. The participant commented that they would "absolutely" use this or similar interfaces for multi-task problems in the future. They felt that, "The visualization made it easier to peer into the models 'black box' and gain a better understanding of what is actually going on in the network."

6.2.4 Participant Feedback: Weaknesses. During the study, participant 1 commented that they wanted some sort of indication, such as colour, of which words were common between selections. The participant also commented that it would be useful to show which words are important in only the selected subset, i.e. show if a word is frequent in only the selected subset since many words show up in multiple subsets. Additionally, the participant wanted the interface to indicate words which have similar linguistic characteristics to the ones that appear in the vis; for example if a key word is a pronoun, then the participant wanted an option to see other pronouns. Lastly, the participant commented that they would prefer if the sentence browser panel showed all the sentences from the dataset, and used some kind of indication like highlighting to indicate which sentences below to the user selection.

Participant 2 suggested that the visualization of the trees could be improved by making the difference in the size scaling more noticeable. The participant also mentioned that perhaps the colour scheme could be changed for which colours indicate which model, as they found that the yellow colour of the arc connections between words in the word graphs conflicted with the gold colour used to indicate model type.

Participant 3 mentioned that the size differential between small and large words could be more pronounced. They also mentioned that connecting words based on whether they are in the same sentence may not be as useful as connecting them if they're in a more immediate context (in the case of long sentences)

Participant 4 stated that they found that they wanted more ways to know if a word was important in multiple subsets, or just the selected subset that they're viewing. They also expressed a desire to manually move the word cloud visualizations around the vis panel.

6.3 Summary of Results

All participants gained some insights to into why the joint task outperformed the single task. For instance, participant 4's comment that, "There are significantly more false negatives for the single task than the joint task. This indicates to me that the single task has trouble identifying what words are more 'human'. For example the single task indicated that 'offline' highly contributed to a tweet being classified as posted by a bot, in contrast the joint task does not have 'offline' included at all." Three of the four participants were also able to gain insights about their datasets in general. For instance, participant 2 stated "Many of the tweets are about the stock market, trading, auctions, and money, and these were mostly predicted as true positives (from bots)."

In the post-study questionnaire all participants commented that they liked the concept of being able to compare two models, as well as the encodings and overall organization of the interface. Participants additionally liked being able to see details on demand by clicking on words, as well as the ability to see visually which words are influential on predictions. Participant 4 commented that, "Most importantly I liked that I could compare results of the joint task vs the single task in a way that is more than just a number (i.e. accuracy, etc.). The words visualized made it easier to understand what the deep models were looking at when making classification decisions."

All four participants suggested that to improve the visualization the interface could include an indication for when a word is important and appearing in only one subset as opposed to multiple subsets. Three of the four participants also commented that the size difference between words should be more pronounced when indicating word contribution to prediction. Two participants mentioned that they would like to see more than two subset selections at the same time. Finally, all four participants said that they would use this or similar interfaces for multi-task problems in the future.

In the following section we present a mock-up of what the interface could look like after accounting for the feedback received from the case study participants.

7 FUTURE WORK

Although the current interface only supports training two tasks jointly, there may be benefit to training more than two tasks. We would like the interface to allow for these neural models jointly trained on more than two tasks, which would involve further development on multiple aspects of the interface. Fig. 7 contains a mock-up design of what the vis panel might look like if the ability to account for more than two tasks was incorporated. The vis panel is changed to allow for more than two user selections to be made at the same time by splitting the panel into as many evenly distributed sections as there are tasks. It was noted during the case studies that three of the four participants did wonder why the vis panel only allowed two selections at once instead of more, so this change could also satisfy some of the case study feedback. Further work would need to be done deciding how many tasks this design could support in total. The sentence browser panel could potentially allow for more than two tasks by removing the two panel layout and instead having a one panel layout with tabs at the top of the layout allowing the user to switch between multiple open selections. It may be that users want to visually see both open selections at the same time, and for that purpose the user tabs could be popped out of the interface and moved freely around the screen. Although this interface design change would allow more than two joint tasks, there may be scalability issues with the design as the number of tasks continues to increase.

We will also incorporate one piece of feedback received from all four study participants, which was that the scale between word size needs to be changed to make a more noticeable size difference.

Furthermore, future development of the interface should address feedback which was received from all four study participants that the interface could be improved with an indicator that a word is important in only one subset as opposed to multiple subsets. As seen in Fig. 7, an addition to the interface which could satisfy this requirement is the yellow highlighted background of words which appear in only one subset.

Since the development of *NJM-Vis* is guided by an iterative design process, future work will also include running more case studies

to evaluate the proposed interface design changes, and the use of more formal user studies to compare alternative versions of the interface.

We also plan to explore further joint neural network architectures. In our work, we used joint neural network architectures which shared only one layer. It may be that different architectures produce more accurate predictions than the two architectures that we use. It's possible that with better performing neural network architectures, our relevance contribution measure may produce stronger signals, and therefore our visualization may better explain predictions to users (eg. there may be a larger visual difference between words which contribute strongly to predictions versus those that contribute weakly to predictions if the network is more confident about predictions).

Finally, we chose to use LRP as our means of calculating a relevance measure for the input. However, it may be the case that there are better means of calculating a relevance for the network input. Though we did explore using attention mechanisms instead of LRP, we ultimately decided on LRP for reasons discussed in section 1. Conceivably, further research may improve attention mechanisms, giving a possible edge over LRP, or other means of relevance contribution may be developed by further research which will outperform LRP.

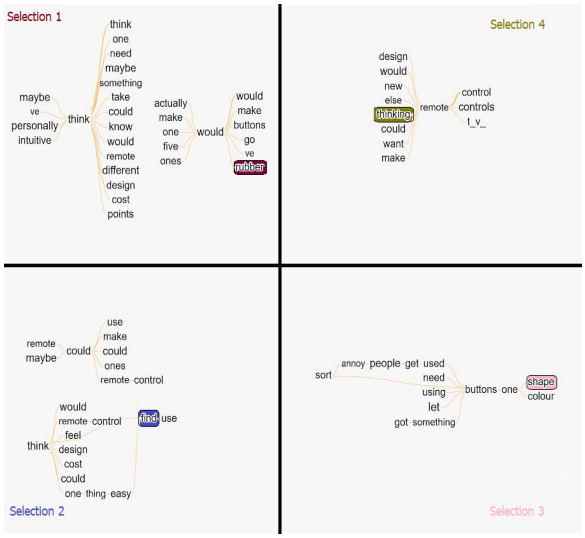

Figure 7: A mock-up to potentially expand the number of user selections at once. Although this could work for four selections, it could have scalability issues as the number of selections grows, since it will be increasingly difficult to dynamically adjust the graphs to fit into the smaller and smaller selection window sizes. Notice also that the background of some words is highlighted which indicates they have high frequency in only their selected subset

8 CONCLUSION

The contribution of this work is a novel interface for exploring and interpreting joint task neural network models. Neural joint models have been shown to outperform non-joint models on several NLP and Vision tasks and constitute a thriving area of research in AI and ML. However, to the best of our knowledge, there is not previous work to support their interpretability. *NJM-Vis* fills this gap by supporting the interpretation of NLP joint task neural model predictions, when compared to single task models. We designed two joint task neural networks using Tensorflow in Python, using two different datasets as the input to our joint task neural networks. In both of our joint neural networks the joint task network improved on the score from the single task network. We chose to use Layerwise Relevance Propagation as a means of explaining the relevance of the input to our neural network predictions. The generated neural network output and prediction scores, and our relevance scores of the input to the neural network, were all used as input to the interface.

As a means of determining how our interface could satisfy user goals, we developed a user task model, which includes the following tasks: measure predictions of two models quantitatively, identify key words in subsets of predictions, dissect linguistic sim/diff between single and joint predictions, identify possible errors in results, identify key relationships between predicted class labels, contextualize predictions at the granularity of sentences.

The design of our visual interface is built to support our user task model. It combines tables and aligned bar-charts with sentence browsers, and word cloud style visualizations. Our interface breaks down neural network predictions into subsets of positive and negative predictions (true positive, false negative, etc). The word cloud visualization is used with the relevance scores to allow users to quickly determine which words contributed to their predictions.

To assess the efficacy of our design, we ran a case study with four participants, who used the interface to explore two different datasets. The first AMI dataset contained labeled data for two NLP tasks: extractive summarization and dialog act prediction. The second dataset was comprised of labeled Twitter data for two tasks: fake news detection and bot detection. All four participants stated that they would use our interface for exploring and interpreting results from their joint tasks, providing preliminary evidence for the usefulness of our prototype.

The case studies also presented valuable feedback from which further versions of our prototype will be informed. Users noted that the interface should have an indication for which words were important in only one subset. Users also recommended minor design changes for general usability: increasing the word size scale so that high relevance words appear larger, and that the color choices of the interface are changed so that the gold color of the arcs between words looks clearly different from the golden color of the font in the interface. We consider this interface to be an early exploration of explaining joint neural models to users, and we believe that the user case study indicates that our interface and the concepts underlying its design are effective. With the iterative process of refining the design with user feedback we will continue to improve our contribution to the new area we have established of explainable joint neural models.

REFERENCES

[1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al.

2016. Tensorflow: a system for large-scale machine learning.. In *OSDI*, Vol. 16. 265–283.

[2] Leila Arras, Grégoire Montavon, Klaus-Robert Müller, and Wojciech Samek. 2017. Explaining recurrent neural network predictions in sentiment analysis. *arXiv preprint arXiv:1706.07206* (2017).

[3] 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 (2015), e0130140.

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

[5] Alsallakh Bilal, Amin Jourabloo, Mao Ye, Xiaoming Liu, and Liu Ren. 2017. Do convolutional neural networks learn class hierarchy? *IEEE transactions on visualization and computer graphics* 24, 1 (2017), 152–162.

[6] Moritz Böhle, Fabian Eitel, Martin Weygandt, and Kerstin Ritter. 2019. Layer-wise relevance propagation for explaining deep neural network decisions in MRI-based Alzheimer's disease classification. *Frontiers in aging neuroscience* 11 (2019), 194.

[7] Jean Carletta. 2007. Unleashing the killer corpus: experiences in creating the multi-everything AMI Meeting Corpus. *Language Resources and Evaluation* 41, 2 (2007), 181–190.

[8] Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*. ACM, 160–167.

[9] Stefano Cresci, Roberto Di Pietro, Marinella Petrocchi, Angelo Spognardi, and Maurizio Tesconi. 2017. The paradigm-shift of social spambots: Evidence, theories, and tools for the arms race. In *Proceedings of the 26th International Conference on World Wide Web Companion*. International World Wide Web Conferences Steering Committee, 963–972.

[10] Alfred Dielmann and Steve Renals. 2008. Recognition of dialogue acts in multiparty meetings using a switching DBN. *IEEE transactions on audio, speech, and language processing* 16, 7 (2008), 1303–1314.

[11] Dumitru Erhan, Yoshua Bengio, Aaron Courville, Pierre-Antoine Manzagol, Pascal Vincent, and Samy Bengio. 2010. Why does unsupervised pre-training help deep learning? *Journal of Machine Learning Research* 11, Feb (2010), 625–660.

[12] Dumitru Erhan, Yoshua Bengio, Aaron Courville, and Pascal Vincent. 2009. Visualizing higher-layer features of a deep network. *University of Montreal* 1341, 3 (2009), 1.

[13] Michael Gleicher. 2017. Considerations for visualizing comparison. *IEEE transactions on visualization and computer graphics* 24, 1 (2017), 413–423.

[14] Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*. 249–256.

[15] Yoav Goldberg. 2017. Neural network methods for natural language processing. *Synthesis Lectures on Human Language Technologies* 10, 1 (2017), 1–309.

[16] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572* (2014).

[17] Fred Matthew Hohman, Minsuk Kahng, Robert Pienta, and Duen Horng Chau. 2018. Visual analytics in deep learning: An interrogative survey for the next frontiers. *IEEE transactions on visualization and computer graphics* (2018).

[18] Mengdie Hu, Krist Wongsuphasawat, and John Stasko. 2017. Visualizing social media content with sententree. *IEEE transactions on visualization and computer graphics* 23, 1 (2017), 621–630.

[19] Sarthak Jain and Byron C Wallace. 2019. Attention is not explanation. *arXiv preprint arXiv:1902.10186* (2019).

[20] Markus John, Eduard Marbach, Steffen Lohmann, Florian Heimerl, and Thomas Ertl. [n.d.]. MultiCloud: Interactive Word Cloud Visualization for Multiple Texts. ([n. d.]).

[21] Minsuk Kahng, Pierre Y Andrews, Aditya Kalro, and Duen Horng Polo Chau. 2017. Activis: Visual exploration of industry-scale deep neural network models. *IEEE transactions on visualization and computer graphics* 24, 1 (2017), 88–97.

[22] Takuhiro Kaneko, Kaoru Hiramatsu, and Kunio Kashino. 2016. Adaptive visual feedback generation for facial expression improvement with multi-task deep neural networks. In *Proceedings of the 2016 ACM on Multimedia Conference*. ACM, 327–331.

[23] Owen Kaser and Daniel Lemire. 2007. Tag-cloud drawing: Algorithms for cloud visualization. *arXiv preprint cs/0703109* (2007).

[24] Sneha Kudugunta and Emilio Ferrara. 2018. Deep Neural Networks for Bot Detection. *arXiv preprint arXiv:1802.04289* (2018).

[25] Sebastian Lapuschkin, Alexander Binder, Grégoire Montavon, Klaus-Robert Müller, and Wojciech Samek. 2016. The LRP toolbox for artificial neural networks. *The Journal of Machine Learning Research* 17, 1 (2016), 3938–3942.

[26] Jaesong Lee, Joong-Hwi Shin, and Jun-Seok Kim. 2017. Interactive visualization and manipulation of attention-based neural machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. 121–126.

[27] Jiwei Li, Xinlei Chen, Eduard Hovy, and Dan Jurafsky. 2015. Visualizing and understanding neural models in NLP. *arXiv preprint arXiv:1506.01066* (2015).

[28] Mengchen Liu, Jiaxin Shi, Zhen Li, Chongxuan Li, Jun Zhu, and Shixia Liu. 2017. Towards better analysis of deep convolutional neural networks. *IEEE transactions on visualization and computer graphics* 23, 1 (2017), 91–100.

[29] Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. 2016. Recurrent neural network for text classification with multi-task learning. *arXiv preprint arXiv:1605.05101* (2016).

[30] Bryan McCann, Nitish Shirish Keskar, Caiming Xiong, and Richard Socher. 2018. The natural language decathlon: Multitask learning as question answering. *arXiv preprint arXiv:1806.08730* (2018).

[31] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* (2013).

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

[33] Grégoire Montavon, Wojciech Samek, and Klaus-Robert Müller. 2018. Methods for interpreting and understanding deep neural networks. *Digital Signal Processing* 73 (2018), 1–15.

[34] Tamara Munzner. 2009. A nested model for visualization design and validation. *IEEE transactions on visualization and computer graphics* 15, 6 (2009), 921–928.

[35] Tamara Munzner. 2014. *Visualization analysis and design*. AK Peters/CRC Press.

[36] Gabriel Murray and Giuseppe Carenini. 2008. Summarizing spoken and written conversations. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 773–782.

[37] Gabriel Murray, Steve Renals, and Jean Carletta. 2005. Extractive summarization of meeting recordings.

[38] Sugeerth Murugesan, Sana Malik, Fan Du, Eunyee Koh, and Tuan Manh Lai. 2019. DeepCompare: Visual and Interactive Comparison of Deep Learning Model Performance. *IEEE computer graphics and applications* (2019).

[39] Bita Nejat, Giuseppe Carenini, and Raymond Ng. 2017. Exploring Joint Neural Model for Sentence Level Discourse Parsing and Sentiment Analysis. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*. 289–298.

[40] Wanli Ouyang and Xiaogang Wang. 2013. Joint deep learning for pedestrian detection. In *Proceedings of the IEEE International Conference on Computer Vision*. 2056–2063.

[41] Tatsuro Oya and Giuseppe Carenini. 2014. Extractive summarization and dialogue act modeling on email threads: An integrated probabilistic approach. In *Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*. 133–140.

[42] Nicola Pezzotti, Thomas Höllt, Jan Van Gemert, Boudewijn PF Lelieveldt, Elmar Eisemann, and Anna Vilanova. 2017. Deepeyes: Progressive visual analytics for designing deep neural networks. *IEEE transactions on visualization and computer graphics* 24, 1 (2017), 98–108.

[43] Hendrik Strobelt, Sebastian Gehrmann, Hanspeter Pfister, and Alexander M Rush. 2017. Lstmvis: A tool for visual analysis of hidden state dynamics in recurrent neural networks. *IEEE transactions on visualization and computer graphics* 24, 1 (2017), 667–676.

[44] William Yang Wang. 2017. " liar, liar pants on fire": A new benchmark dataset for fake news detection. *arXiv preprint arXiv:1705.00648* (2017).

[45] Chao Yang, Robert Harkreader, and Guofei Gu. 2013. Empirical evaluation and new design for fighting evolving twitter spammers. *IEEE Transactions on Information Forensics and Security* 8, 8 (2013), 1280–1293.

[46] Jason Yosinski, Jeff Clune, Anh Nguyen, Thomas Fuchs, and Hod Lipson. 2015. Understanding neural networks through deep visualization. *arXiv preprint arXiv:1506.06579* (2015).

[47] Jiawei Zhang, Yang Wang, Piero Molino, Lezhi Li, and David S Ebert. 2018. Manifold: A model-agnostic framework for interpretation and diagnosis of machine learning models. *IEEE transactions on visualization and computer graphics* 25, 1 (2018), 364–373.

[48] Arkaitz Zubiaga, Maria Liakata, and Rob Procter. 2016. Learning reporting dynamics during breaking news for rumour detection in social media. *arXiv preprint arXiv:1610.07363* (2016).