NJM-Vis: Applying and Interpreting Neural Network Joint Models in Natural Language Processing Applications

by

David Johnson

BCIS, University of the Fraser Valley, 2016

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

in

THE FACULTY OF GRADUATE AND POSTDOCTORAL STUDIES

(Computer Science)

The University of British Columbia

(Vancouver)

October 2019

© David Johnson, 2019
The following individuals certify that they have read, and recommend to the Faculty of Graduate and Postdoctoral Studies for acceptance, the thesis entitled:

NJM-Vis: Applying and Interpreting Neural Network Joint Models in Natural Language Processing Applications

submitted by David Johnson in partial fulfillment of the requirements for the degree of MASTER OF SCIENCE in Computer Science.

Examinining Committee:
Giuseppe Carenini, Computer Science
Supervisor

Gabriel Murray, Computer Science
Supervisor
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 thesis 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).
Lay Summary

A deep neural network is a machine learning algorithm in which layers of artificial neurons are trained to learn features of the neural network input which are informative for the target prediction task. Although this algorithm can have strong predictive power, it is often seen as a blackbox in which the results of the neural network are often difficult to explain or interpret. Multitasking is a manner of training deep neural networks in which the neural network learns more than one task at the same time with the intention that learning one task will benefit learning the other task and vice versa. The added complexity of multitask learning means the results from deep neural networks may be even more difficult to interpret. This thesis describes an interface intended to allow users to interpret and explore the results from their deep multitask neural networks, making the blackbox that lies between the users and their deep neural network output transparent.
Preface

This thesis is an original work of the author, David Johnson, under the supervision of Dr. Giuseppe Carenini and Dr. Gabriel Murray. This work is unpublished at this time.
# Table of Contents

Abstract ................................................................. iii

Lay Summary ............................................................. iv

Preface .......................................................................... v

Table of Contents .......................................................... vi

List of Tables ............................................................... viii

List of Figures ............................................................... ix

Acknowledgments .......................................................... xvi

1 Introduction ............................................................... 1

2 Related Work ............................................................. 5
   2.0.1 Neural Joint Models .............................................. 5
   2.0.2 Interfaces to Interpret Neural Models (Visual Data) .......... 5
   2.0.3 Interfaces to Interpret Neural Models (Textual Data) .......... 6
   2.0.4 Saliency Interpretation Method .................................. 8
   2.0.5 Word Cloud Visualization Techniques ......................... 8

3 Datasets ...................................................................... 18

4 Data Model .............................................................. 21
   4.1 Model Description .................................................... 21
List of Tables

Table 4.1  Single & Joint Task Results for Extractive Summarization . . .  24
Table 4.2  Single & Joint Task Training Results for Dialogue Act . . . . .  25
Table 4.3  Single & Joint Task Results for Bot Detection . . . . . . . . .  25
Table 4.4  Single & Joint Task Results for Fake News Detection . . . . .  25
List of Figures

Figure 1.1  *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.

Figure 2.1  *CNNVis*, a system for visualizing CNNs for understanding, diagnosing non-convergence problems, and refining CNNs. The visualization shows a CNN as a directed acyclic graph, with an aggregation of neurons and layers. [25]
Figure 2.2 Visualization tool for live convnet activations from [43]. For clarity, the original caption is slightly revised as: “The bottom shows a screenshot from the software. Webcam input is shown, along with the whole layer of conv5 activations. The selected channel pane shows an enlarged version of the 13x13 conv5_{151} channel activations. Below it, the deconv starting at the selected channel is shown. On the right, three selections of nine images are shown: synthetic images produced using regularized gradient ascent methods, the top 9 images patches from the training set (the images from the training set that caused the highest activations for the selected channel), and the deconv of those top 9 images. All areas highlighted with a green star relate to the particular selected channel, here conv5_{151}; when the selection changes, these panels update. The top depicts enlarged numerical optimization results for this and other channels. conv5_2 is a channel that responds most strongly to dog faces, but it also responds to flowers on the blanket on the bottom and halfway up the right side of the image (as seen in the inset red highlight). conv5_{151} detects different types of faces. The top nine images are all of human faces, but here we see it responds also to the cat’s face. Finally, conv5_{111} activates strongly for the cat’s face, the optimized images show cat like fur and ears, and the top nine images (not shown here) are also all of cats.”

Figure 2.3 Multifaceted Feature visualization. Original caption revised as: “The top images show 8 types of images that activate the ‘grocery store’ class neuron. The bottom images show training set images that activate the same neuron, as well as resemble the synthetic images in the top panel” [35].
Figure 2.4  *deepViz* interface. The middle of the interface shows the bitmap representations of filter banks in the selected CNN layer. Above are nodes displaying the network overview. The user is able to click one of the nodes to view the selected layer’s filters. The left menu allows users to search for images labeled with a certain class (“plane” in this figure). The side view is a histogram of the model’s predicted classes for the selected image [5]. . .

Figure 2.5  *ActiVis* interface [20]. For clarity, the original caption is slightly revised as: “*ActiVis* integrates several coordinated views to support exploration of complex deep neural network models, at both instance-and subset-level. 1. The user Susan starts exploring the model architecture, through its *computation graph* overview (at A). Selecting a *data node* (in yellow) displays its *neuron activations* (at B). 2. The *neuron activation matrix view* shows the activations for instances and instance subsets; the *projected view* displays the 2-D projection of instance activations. 3. From the *instance selection* panel (at C), she explores individual instances and their classification results. 4. Adding instances to the matrix view enables comparison of activation patterns across instances, subsets, and classes, revealing causes for misclassification.” . . . . . . . . . . . . . . . . . . . . .

Figure 2.6  *DeepCompare* [33] interface showing a comparison between CNN and LSTM deep neural models. A) A neuron weight detail panel showing weights of one layer color-coded (green high, red low) for each model. B) Neuron Activation Distribution panel showing the number of data instances binned into an activation scale from -1 to 1. C) Test results panel showing data instances and a glyph displaying whether each model predicted the test instance correctly or incorrectly. D) Test Result Summary panel showing a treemap of the model performance on the entire test dataset color-coded to show positive and negative data instances (purple and yellow respectively). . . . .
Figure 2.7  *Manifold* interface. Original caption: “Manifold consists of two interactive dialogs: a model comparison overview (1) that provides a visual comparison between model pairs using a small multiple design, and a local feature interpreter view (2) that reveals a feature-wise comparison between user-defined subsets (c) and provides a similarity measure (b) of feature distributions. The user can sort based on multiple metrics (a) to identify the most discriminative features among different subsets, i.e., sort based on the selected subset in red (2) or a specific class such as $C_1$ (3).” [44].

Figure 2.8  Original caption slightly revised as: “Saliency map for ‘I hate the movie’. Each row corresponds to saliency scores for the correspondent word representation. The x-axis corresponds to the word embedding dimensions (each word is set to 60 dimensions)” [24].

Figure 2.9  SentenTree visualization on Twitter data in which words are nodes and nodes connected by an edge are words which co-occur in tweets together. The size of words indicates a word’s frequency in a dataset [18].

Figure 2.10  MultiCloud visualization. The visualization contains small dots around the word cloud which indicate “anchor points”. “Anchor points” are those which indicate documents. The anchor points pull words in the visualization towards the point if the word is in the document. Anchor points, and words which belong to the document represented by the anchor point, are colour-coded. Words which are in multiple documents are grey in colour. The right image shows a word cloud built containing only words which are from individual documents, while the left image shows a word cloud built from words belonging to both multiple documents (words in grey) and words in individual documents (words in colour) [19].
Figure 4.1 Neural network architecture for the bot detection & fake news detection tasks. Bot detection is in blue, fake news detection is in orange, and the overlap is the shared layer which is jointly learned during joint training. The numbers in the graphics indicate the dimensions of each layer. For instance, the input for both networks is set to 2000 dimensions.

Figure 4.2 Neural network architecture for the extractive summarization & dialog act prediction tasks. Extractive summarization is in blue, dialog act prediction is in orange. The numbers in the graphics indicate the dimensions of each layer. For instance, the input for both networks is set to 2500 dimensions.

Figure 6.1 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.

Figure 6.2 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.
Figure 6.3  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 6.4  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.  

Figure 6.5  Early interface mock-up. The top of the interface contains confusion matrices showing positive/negative subsets (true positive, false positive, etc.) The confusion matrix also includes the number of fixed/broken data instances in green and brown. Below the confusion matrices is a table containing the scores of precision, recall, and f-score. Below, is a sentence browser where each sentence is color-coded with contribution score in which the darker the blue is the stronger the contribution score toward prediction is, and the darker the red is the stronger the contribution score against prediction is. Below is the word-graph visualization which we continue to use on the current interface.  

Figure 6.6  Prototype interface. This is the first workable prototype interface. The interface contains confusion matrices at the top with the number of data instances classified into each positive/negative subset, as well as the word-graph visualization below.  

Figure 6.7  A diagram of possible aligned bar chart designs drawn during the iterative design process.  

Figure 6.8  Another possible stacked bar chart design drawn during the iterative design process.  

Figure 6.9  Interface design created prior to the current design. This version of the interface contained only single user selections, and did not allow direct comparisons between multiple user selections.
Figure 8.1 A mock-up to potentially expand the number of user selections at once. This shows a way of potentially splitting the vis panel into sections by dynamically decreasing the number of graph visualizations in such a way as to fit the most number of graphs into each selection window. 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 as explained in Figure 8.2.

Figure 8.2 A mock-up to address case study feedback. In the vis panel, the size scale has been increased to make it easier to differentiate between high contributing words (large font) and low contributing words (small font). Additionally, words which have a high frequency in only their one selected subset have a highlighted background, as seen with the word ”points”.

Figure A.1 A screenshot of the visualization produced by the toy dataset used for demo purposes during the training portion of the case study.
Acknowledgments

I would like to extend my gratitude to Dr. Giuseppe Carenini and Dr. Gabriel Murray for supporting me, guiding me, and teaching me every step of the way throughout this work.

To my family and to my friends I can say with certainty that I wouldn’t have ever succeeded in reaching this point without you. My thanks and my love to you for being there for me to rely on, and for consistently bringing joy into my life.
Chapter 1

Introduction

Deep learning approaches have recently shown great potential on a large number of key prediction problems. 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 [27]. 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 [36]. 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 [7]. 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 [34].

Although a strength of deep learning is in learning representations of data, the complexity of the representation makes explanation for deep neural networks notoriously difficult. Deep neural networks often learn multiple representations of the input, and we do not know which part of the input they capture [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

---

**Figure 1.1:** 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.
look like natural images, they can be used to determine what a neuron layer has learned to detect [43][11][16]. In addition to feature optimization methods, there are also attribution methods such as Layerwise Relevance Propagation (LRP) and Saliency attribution. 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.

In this thesis 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 1.1, comprises two views in a multiform overview/detail design [30], 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 [31][32] is jointly performed with a dialog act detection task [9] 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 detection is to predict the speaker intention associated with an utterance, e.g. question, answer, inform. In Bot-FakeNews, a bot detection task [22] is jointly trained with a fake news detection task [40]. 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 thesis, 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 (e.g., Tensorboard [1], Deep Eyes [38], and Blocks [4]) 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 Ac-
tiVis [20], and LSTMVis [39]. In this work we will specify the terms “model
developers” or “model users” 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.
Chapter 2

Related Work

2.0.1 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 [12] to result in better generalization and better performance than the typical manner of random weight initialization. Another style of joint model is multi-tasking [7], 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 [26] and computer vision [21]. In this work, we use multi-tasking, as it tends to outperform pre-training (e.g., [34] in NLP, joining discourse parsing and sentiment analysis).

2.0.2 Interfaces to Interpret Neural Models (Visual Data)

Much of the previous work creating interfaces for visualizing deep neural networks (DNN) has been done on computer vision tasks using Convolutional Neural Networks (CNN). [43] describes an interface allowing visualization of plotted convolutional layer activation values as seen in Figure 2.2. Similarly [25] (Fig. 2.1) 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. In [35] (Fig. 2.3) the authors state that a limit of current explanation techniques is that they assume each neuron detects only one type of feature, but neurons must be multifacted, i.e., that they fire in response to different types of features. Previous activation maximization techniques constructed images without regard for multiple facets of a neuron. The authors explicitly uncover multiple facets of each neuron by separately producing a synthetic visualization of each of the types of images that activate a neuron. The authors accomplish this by calculating the derivative of the target neuron activation with respect to each pixel (this describes how to change pixel color to increase activation of that neuron). Another work for visualizing CNNs is deepViz [5] (Fig. 2.4). In deepViz, the interface allows exploration of convolution neural networks as the network changes over training time steps. Users select an image which is then passed through a particular network filter. Visualization is done using decaf [10] which is used to retrieve and visualize the activation values of a particular layer, selected by the user. With an included slider to allow the user to move through the training time steps, the interface visualizes how the selected layer’s filter changes over time. Like other discussed interfaces, deepViz focuses only on visual image data, not text, and does not have any functionality in place to account for joint neural models.

2.0.3 Interfaces to Interpret Neural Models (Textual Data)

Some recent work has explored visual interfaces for understanding neural NLP (e.g., [24]) (Fig 2.8), but they are limited to single-task models, while our goal is supporting the interpretation of joint-models.

In [33] (Fig. 2.6), the authors present a visual analytic system for comparing the results of two deep neural models. Their interface allows users to compare neuron activations by layer, as well as the number of data instances which fall into each activation bin from values -1 to 1. In addition, the interface shows users details of data instances such as a juxtaposition showing which of the two neural models correctly classified the data instance. The interface shows a treemap of the overall performance of both models. Although this interface compares two neural
Figure 2.1: CNNVis, a system for visualizing CNNs for understanding, diagnosing non-convergence problems, and refining CNNs. The visualization shows a CNN as a directed acyclic graph, with an aggregation of neurons and layers. [25]

models as we do in our work, they do not compare joint neural models. Additionally, the intention behind our work is to explain the results of joint deep neural networks, whereas the intention behind DeepCompare is to compare the performance of models, not to explain the results. Another similar interface is ActiVis [20] (Fig. 2.5). In ActiVis the authors present an interface intended to be effective for industry scale data. The interface allows users to see neuron activations over a dataset by means of viewing classes, a 2d projection of the neuron activations by class, and a visual representation of the data instances by class which allows the user to explore the data instances and compare data instance neuron activations. Though this interface does allow neural models trained on textual data, it focuses only on single-task networks. Manifold [44] (Fig. 2.7) is another interface for interpretation of neural models, although Manifold also incorporates machine learning models beyond just neural models (linear regression, support vector machines,
etc.). The interface allows comparison between positive and negative subsets (true positive, false negative, etc.) of predictions between models. The interface additionally provides a way to view the number of features appearing in each of the user’s selected instances, allowing the user to see the feature distribution. There are similarities to our work, such as the comparison between positive and negative subsets of predictions between models, though Manifold is not intended to directly compare joint task models as our interface does, so Manifold does not directly display to the user data instances which were fixed or broken by the process of joint training.

2.0.4 Saliency Interpretation Method

In [24] (Fig. 2.8) 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 [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. In this thesis, we rely on LRP to explain both single task and joint task predictions.

2.0.5 Word Cloud Visualization Techniques

In our work we use a word cloud style visualization adapted from [18]. In [18] (Fig. 2.9) the authors present SentenTree, a technique for visualizing social media text. The visualization uses a word cloud to present words which are common in social media datasets. The SentenTree technique builds on typical word clouds by including 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). Another work using word clouds as visualization techniques is MultiCloud [19]. MultiCloud is a visualization technique that expands word clouds to allow visualization of multiple documents within a single word cloud. MultiCloud has fixed points along the border of the visualization layout, and distributes words in locations which indicate to which document or documents the word belongs. For example, if the fixed point for document 1 is in the top left corner of the layout, and a word in the word cloud comes from document 1, it is pulled closer to the top left corner of the word cloud. Although this visualization technique is intended to show from which documents words in the word cloud are from, it is possible this technique could be applied to our work, in which we could instead have a single word cloud and use the fixed points on the border to indicate from which neural model (joint or single) the word is from. Exploring this alternative is left as future work.
Figure 2.2: Visualization tool for live convnet activations from [43]. For clarity, the original caption is slightly revised as: “The bottom shows a screenshot from the software. Webcam input is shown, along with the whole layer of conv5 activations. The selected channel pane shows an enlarged version of the 13x13 conv5\textsubscript{151} channel activations. Below it, the deconv starting at the selected channel is shown. On the right, three selections of nine images are shown: synthetic images produced using regularized gradient ascent methods, the top 9 images patches from the training set (the images from the training set that caused the highest activations for the selected channel), and the deconv of the those top 9 images. All areas highlighted with a green star relate to the particular selected channel, here conv5\textsubscript{151}; when the selection changes, these panels update. The top depicts enlarged numerical optimization results for this and other channels. conv5\textsubscript{2} is a channel that responds most strongly to dog faces, but it also responds to flowers on the blanket on the bottom and half way up the right image (as seen in the inset red highlight). conv5\textsubscript{151} detects different types of faces. The top nine images are all of human faces, but here we see it responds also to the cat’s face. Finally, conv5\textsubscript{111} activates strongly for the cat’s face, the optimized images show cat like fur and ears, and the top nine images (not shown here) are also all of cats.”
Figure 2.3: Multifaceted Feature visualization. Original caption revised as: “The top images show 8 types of images that activate the ‘grocery store’ class neuron. The bottom images show training set images that activate the same neuron, as well as resemble the synthetic images in the top panel” [35].
Figure 2.4: *deepViz* interface. The middle of the interface shows the bitmap representations of filter banks in the selected CNN layer. Above are nodes displaying the network overview. The user is able to click one of the nodes to view the selected layer’s filters. The left menu allows users to search for images labeled with a certain class ("plane" in this figure). The side view is a histogram of the model’s predicted classes for the selected image [5].
Figure 2.5: ActiVis interface [20]. For clarity, the original caption is slightly revised as: “ActiVis integrates several coordinated views to support exploration of complex deep neural network models, at both instance-and subset-level. 1. The user Susan starts exploring the model architecture, through its computation graph overview (at A). Selecting a data node (in yellow) displays its neuron activations (at B). 2. The neuron activation matrix view shows the activations for instances and instance subsets; the projected view displays the 2-D projection of instance activations. 3. From the instance selection panel (at C), she explores individual instances and their classification results. 4. Adding instances to the matrix view enables comparison of activation patterns across instances, subsets, and classes, revealing causes for misclassification.”
Figure 2.6: DeepCompare [33] interface showing a comparison between CNN and LSTM deep neural models. A) A neuron weight detail panel showing weights of one layer color-coded (green high, red low) for each model. B) Neuron Activation Distribution panel showing the number of data instances binned into an activation scale from -1 to 1. C) Test results panel showing data instances and a glyph displaying whether each model predicted the test instance correctly or incorrectly. D) Test Result Summary panel showing a treemap of the model performance on the entire test dataset color-coded to show positive and negative data instances (purple and yellow respectively).
Figure 2.7: *Manifold* interface. Original caption: “Manifold consists of two interactive dialogs: a model comparison overview (1) that provides a visual comparison between model pairs using a small multiple design, and a local feature interpreter view (2) that reveals a feature-wise comparison between user-defined subsets (c) and provides a similarity measure (b) of feature distributions. The user can sort based on multiple metrics (a) to identify the most discriminative features among different subsets, i.e., sort based on the selected subset in red (2) or a specific class such as $C_1$ (3).” [44].
Figure 2.8: Original caption slightly revised as: “Saliency map for ‘I hate the movie’. Each row corresponds to saliency scores for the correspondent word representation. The x-axis corresponds to the word embedding dimensions (each word is set to 60 dimensions)” [24].

Figure 2.9: SentenTree visualization on Twitter data in which words are nodes and nodes connected by an edge are words which co-occur in tweets together. The size of words indicates a word’s frequency in a dataset [18].
Figure 2.10: MultiCloud visualization. The visualization contains small dots around the word cloud which indicate “anchor points”. “Anchor points” are those which indicate documents. The anchor points pull words in the visualization towards the point if the word is in the document. Anchor points, and words which belong to the document represented by the anchor point, are colour-coded. Words which are in multiple documents are grey in colour. The right image shows a word cloud built containing only words which are from individual documents, while the left image shows a word cloud built from words belonging to both multiple documents (words in grey) and words in individual documents (words in colour) [19].
Chapter 3

Datasets

In this thesis, we develop two NLP joint models: Summ-DiaAct and Bot-FakeNews. In Summ-DiaAct, an extractive summarization task \[31\][32] is jointly performed with a dialog act detection task \[9\] on conversational data. In Bot-FakeNews, a bot detection task \[22\] is jointly trained with a fake news detection task \[40\]. For the Summ-DiaAct joint model, we use the Augmented Multi-party Interaction (AMI) corpus as our dataset \[6\]. AMI is a multi-modal dataset created from 100 hours of meeting recordings. In creating this dataset, participants played the various roles in a design team in which their goal was to take a project from kick-off to finish throughout the course of a day. The AMI dataset has annotations for multiple tasks such as dialog act, topic segmentation, abstractive and extractive summarization, named entities etc.. We use the annotations for the dialog act (15 types) and extractive summarization (binary) tasks, as these two were shown to benefit from joint training \[37\]. The extractive summarization labels represent whether a sentence is extract worthy or not. The dialog act annotations used are as follows:

- **Backchannel**: Someone listening to the speaker says something in the background without stopping the speaker.

- **Stall**: Speaker starts speaking before they are ready and uses “filled pauses” such as “uh”, “um”.

- **Fragment**: Speaker started saying something but stopped before they got far enough to finish their intention.
- **Inform**: Speaker spoke with the intention to give information

- **Elicit-Inform**: Speaker requested that someone else gives information

- **Suggest**: Speaker gives a suggestion to the listeners

- **Offer**: Speaker expresses an intention relating to their own actions

- **Elicit-Offer-Or-Suggestion**: Speaker requests that someone else makes an offer or a suggestion

- **Assess**: Speaker expresses an evaluation of something that is being discussed by the group

- **Comment-About-Understanding**: Speaker indicates that they did or did not understand what a previous speaker said

- **Elicit-Assessment**: Speaker attempts to elicit an assessment about what was said or done previously

- **Elicit-Comment-About-Understanding**: Speaker elicits from listeners about whether what has been said has been understood

- **Be-Positive**: Social acts intended to make the individual or group happier

- **Be-Negative**: Social acts expressing negative feelings to the individual or group

- **Other**: Any acts conveyed by the speaker which do not fit into the other act categories

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 [8], which contains both tweets from genuine accounts, as well as from accounts identified as bots in [41]. Dataset [8] included and built on work from [41] in which they identified “spambots” by crawling Twitter profiles through the Twitter API and determining which accounts have posted at least one malicious URL. A URL was identified as “malicious” via two methods: Google Safe Browsing
(GSB) and a URL honeypot. GSB is a blacklist for identifying malicious or phishing URLs. The honeypot was developed by [41] as well in order to visit the URLs using a browser inside a virtual machine and then to detect creation/modification of sensitive data. After identifying an account as potentially belonging to a bot, [41] manually went through each of the accounts and decided whether their tweets were useful and meaningful. Genuine real user accounts were identified from a random sample of Twitter users which [8] randomly contacted and asked a simple question in natural language. The replies to the question were manually verified, and accounts which properly answered the question were verified as human. For the fake news task, we used a dataset from [45], 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.

The author proceeded in this manner over five different newsworthy events:

- Ferguson unrest: citizens of Ferguson in MI, USA protest a fatal shooting of an 18-year-old African American Michael Brown by a white police officer on August 9, 2014
- Ottawa shooting: shootings on Parliament Hill in Ottawa, Canada result in the death of a Canadian soldier on October 22, 2014
- Sydney siege: a gunman held hostage ten customers and eight employees of a Lindt chocolate cafe located at Martin Place in Sydney, Australia on December 15, 2014
- Charlie Hebdo shooting: two brothers force their way into the offices of French satirical weekly newspaper Charlie Hebdo in Paris, killing 11 people and wounding 11 more on January 7, 2015
- Germanwings plane crash: a passenger plane from Barcelona to Dusseldorf crashes in the French Alps on March 24, 2015, killing all passengers and crew. The plane was found to have been deliberately crashed by the co-pilot
Chapter 4

Data Model

By following the standard information visualization methodology [29], we base the design of *NJM-Vis* on abstracted data and task models. In this chapter, we present the data model, which describes information about sentences and words that we need to compute and store. The task model, which outlines key analysis tasks to support the interpretation of joint models and their comparison with single models, will be presented in the following chapter.

4.1 Model Description

The data model for the four classification tasks comprises tables containing information associated to sentences and to 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)\cite{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:

\[
a_k = g(\sum_j a_j w_{jk} + b)
\]  

(4.1)

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. As shown in [28] a rule that works to propagate relevance is:

\[
R_j = \sum_k \frac{a_j w_{jk}^+}{\sum_j a_j w_{jk}^+} R_k^+ + \frac{a_j w_{jk}^-}{\sum_j a_j w_{jk}^-} R_k^-
\]  

(4.2)

where \( R_k^+ = \alpha R_k \) and \( R_k^- = -\beta R_k \) with \( \alpha \) and \( \beta \) chosen to constraints \( \alpha - \beta = 1 \) and \( \beta >= 0 \).

Although in this thesis 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. Another possible method would be an attention mechanism which could similarly give an importance score to each input word. For the purposes of this thesis we chose to use LRP over an attention mechanism since, as shown in [42], introducing an attention mechanism adds additional complexity to neural networks.
which can possibly require longer training time and more labeled data (which are rather limited for our prediction tasks).

4.2 Model Architecture

Both networks were developed using Tensorflow [1] in Python with an LRP implementation adapted from [23]. The network for Summ-DiaAct is shown in Figure 4.2. Both the extractive summarization and dialog act prediction networks use 2500 input dimensions, since the input to each network is a sentence containing 25 words total, with 100 word embedding dimensions per word. Sentences with more than 25 words are trimmed at 25 words, and sentences with less than 25 words are padded with zeroes until they hit 2500 dimensions. The intermediate layers for both tasks using ReLU activation functions, as this tended to result in better performance through empirical testing. The output layer for the extractive summarization task applies a sigmoid activation function since the task is binary classification, while the dialogue act task applies a softmax activation function since the task is multi-class classification.

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.

The network for Bot-FakeNews is shown and described in Figure 4.1. Both the bot detection and fake news detection networks use 2000 input dimensions, since the input to each network is a tweet containing 20 words total, with 100 word embedding dimensions per word. Using 2000 input dimensions performed better than 2500 input dimensions for these tasks from empirical testing. Tweets with more than 20 words are trimmed at 20 words, and tweets with less than 20 words are padded with zeroes until they hit 2000 dimensions. The intermediate layers use ReLU activation functions, while both output layers use a sigmoid activation function.

4.3 Results Comparing Single and Joint Models

As shown in Table 4.1, the results for extractive summarization improve when trained in the Summ-DiaAct joint model. In contrast, Table 4.2 indicates that for
Figure 4.1: Neural network architecture for the bot detection & fake news detection tasks. Bot detection is in blue, fake news detection is in orange, and the overlap is the shared layer which is jointly learned during joint training. The numbers in the graphics indicate the dimensions of each layer. For instance, the input for both networks is set to 2000 dimensions.

the dialogue act task the improvement is negligible.

<table>
<thead>
<tr>
<th>Model</th>
<th>F-score</th>
<th>Precision</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single</td>
<td>.424</td>
<td>.604</td>
<td>.327</td>
</tr>
<tr>
<td>Joint</td>
<td>.463</td>
<td>.632</td>
<td>.366</td>
</tr>
</tbody>
</table>

Table 4.1: Single & Joint Task Results for Extractive Summarization

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 4.3 and 4.4.
Figure 4.2: Neural network architecture for the extractive summarization & dialog act prediction tasks. Extractive summarization is in blue, dialog act prediction is in orange. The numbers in the graphics indicate the dimensions of each layer. For instance, the input for both networks is set to 2500 dimensions.

<table>
<thead>
<tr>
<th>Micro Average (Single)</th>
<th>Micro Average (Joint)</th>
</tr>
</thead>
<tbody>
<tr>
<td>.711</td>
<td>.714</td>
</tr>
</tbody>
</table>

Table 4.2: Single & Joint Task Training Results for Dialogue Act

<table>
<thead>
<tr>
<th>Model</th>
<th>F-score</th>
<th>Precision</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single</td>
<td>.75</td>
<td>.88</td>
<td>.66</td>
</tr>
<tr>
<td>Joint</td>
<td>.96</td>
<td>.97</td>
<td>.93</td>
</tr>
</tbody>
</table>

Table 4.3: Single & Joint Task Results for Bot Detection

<table>
<thead>
<tr>
<th>Model</th>
<th>F-score</th>
<th>Precision</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single</td>
<td>.76</td>
<td>.86</td>
<td>.70</td>
</tr>
<tr>
<td>Joint</td>
<td>.81</td>
<td>.90</td>
<td>.72</td>
</tr>
</tbody>
</table>

Table 4.4: Single & Joint Task Results for Fake News Detection
Chapter 5

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 comparison [13]. As in [13] where comparison tasks are grouped abstractly as the following actions: Identify, Measure, Dissect, Connect, Contextualize, and Communicate.

In addition to referring to the aforementioned existing works, 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 which model performs better quantitatively since this allows a user 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, therefore we could say that “dollars” is a key word. Alternatively, a word may be considered a key word if it has a high contribution measure in a subset of predictions. Lastly, a word may be a key word 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: Dissect that key words appearing in true positive predictions are often pronouns.

Once the user/developer 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, perhaps 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 “sug-
gest” label are appearing more often in the jointly trained extractive summa-
rization model than in the single trained model. The user could then infer
that perhaps spoken suggestions have predictive power for whether a sen-
tence 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: Contextualize 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.
Chapter 6

Design Solution

6.1 NJM-Vis Design

NJM-Vis is faceted into multiple views of coordinated visualizations. As seen in Figure 6.1, the left side of the interface, the score panel (Fig. 6.2), 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. 6.2).

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. This
Figure 6.1: 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.

facilitates direct comparison between selected subsets.

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. 6.3) adapted from [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 Figure 6.1. 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. 6.3 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 words which often appear, 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. 6.1 in the joint task true positive subset, and 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 in which it is likely the intended word was “i’ve” or perhaps “would’ve”.

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. 6.4), 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:

- **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. 6.4 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. This allows the user to directly compare sentences for the joint and single task systems. 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.

### 6.2 Iterative Design Process

The design of the interface took place through an iterative design process in which we refined the design of the interface as we refined the clarity of the user tasks and data model. An early mock-up version of the interface can be seen in Fig 6.5. The interface included tables showing model prediction subsets such as true positive, false positive, etc as well as precision, recall, and f-scores scores for both the single and joint task models. The table also included a “sentence browser” with color-coded words which were attributed to relevance contribution values. The darker the color-coded words, the stronger the contribution score towards the prediction (i.e. darker blue contributes stronger to prediction than lighter blue). This was removed in the final version, since the lighter colors tended to be harder to view, and since the relevance contribution measure was already being captured by the size of the words in the word graph. In a panel below was the word cloud visualizations.
Though this early mock-up version did include some of the concepts which were included in the final version, such as using a word cloud visualization and showing tables with the number of true positives, false positives, etc. this version of the interface did not fully support all of the user tasks which we ended up deriving. For example, T3 is not supported by this interface, as the word cloud visualization only supports viewing one subset selection at a time.

A prototype interface was implemented as seen in Fig 6.6 with the design based off of the mock-up version in Fig 6.5. The prototype version was similar in concept to the mock-up, including the tables counting the number of positive and negative subsets of predictions, as well as the word cloud visualization underneath. To better suit user comparison, the tables showing positive and negative subsets of predictions was enhanced with aligned bar-charts [30]. The bar charts went through several mock-ups as seen in Fig 6.7 and Fig 6.8.

During the process of adding the new aligned bar-chart design, it was also decided that adding a sentence browser would help satisfy our user task model by supporting what would become our tasks T5 and T6. With both the sentence browser and the aligned bar-chart design implemented, the interface design was as seen in Fig 6.9.

As our user task model became more refined we realized that the interface design in Fig 6.9 would not fully support a key task in our user task model, task T3, and that to support a task focused on comparing both models we needed to amend the vis panel to allow for a juxtaposed visualization view in which the user is able to view two selections at the same time. After completing the implementation of the juxtaposed vis panel, we arrived at the current design of the interface which supported our entire task model.

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 Chapter 6 and potential interface additions and mock-ups are presented in Chapter 7.
Figure 6.2: 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.
Figure 6.3: 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 6.4:** 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.
Figure 6.5: Early interface mock-up. The top of the interface contains confusion matrices showing positive/negative subsets (true positive, false positive, etc.) The confusion matrix also includes the number of fixed/broken data instances in green and brown. Below the confusion matrices is a table containing the scores of precision, recall, and f-score. Below, is a sentence browser where each sentence is color-coded with contribution score in which the darker the blue is the stronger the contribution score toward prediction is, and the darker the red is the stronger the contribution score against prediction is. Below is the word-graph visualization which we continue to use on the current interface.
Figure 6.6: Prototype interface. This is the first workable prototype interface. The interface contains confusion matrices at the top with the number of data instances classified into each positive/negative subset, as well as the word-graph visualization below.
Figure 6.7: A diagram of possible aligned bar chart designs drawn during the iterative design process.
Figure 6.8: Another possible stacked bar chart design drawn during the iterative design process.
Figure 6.9: Interface design created prior to the current design. This version of the interface contained only single user selections, and did not allow direct comparisons between multiple user selections.
Chapter 7

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 in which future versions of the interface could be influenced directly by the case study feedback, at which point further case studies could be run, allowing additional feedback, and so on.

7.1 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 two participants being assigned to the Summ-DiaAct joint tasks, and the other two 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. An example of the appearance of the interface during the demo can be seen in Appendix A.2. 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 see 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.

7.2 Participant Results

In this section we provide the results for each of the four participants from the case study. 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. The full participant post-study questionnaire results can be seen in the Appendix.

7.2.1 Participant 1

Participant 1 was assigned to the Summ-DiaAct joint tasks. The participant began the 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.

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.

In the post-study questionnaire the participant commented that they liked the concept of being able to compare the two models directly, and they found the visual design of using size to indicate importance useful. The participant commented that the size difference should be more notable so that there’s a wider difference between small and large words.

Using the interface the participant was able to make some inferences about the two tasks, for instance that modal verbs were often big indicating that they contributed strongly to predictions, which the participant commented, “…seemed intuitive given that the dataset was a dialog dataset”.

Finally, the participant concluded that 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 as described above.

### 7.2.2 Participant 2

Participant 2 was assigned to the Bot-FakeNews dataset. The participant began by 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.

The participant was able to come 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 the participants 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”.

The visual aspect of the interface, including the colours and clear organization of the score panel were commented on by the participant as being “very useful” and “…provided a fast and easy way to organized the results”. Additionally the participant commented that being able to compare selections in juxtaposition was extremely useful.

The participant 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.

The participant concluded by 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”.

7.2.3 Participant 3

Participant 3 was assigned to the SummDia-Act tasks. The participant began the study by first looking 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 made at once.

While using the interface, the participant 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, so the joint task outperformed the single task in each category. The
participant felt they could look at word groups that appear in the true positive but
not the false negative for the joint task and not in the true positive for the single
task, (presumably to find words which were important in only the true positive
subset indicating that the neural model learned through joint training that the word
can be informative in predictions) though they felt that might take a lot of cognitive
load to accomplish.

In the post-study questionnaire the participant indicated 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.

As suggestions for improvements to the interface, the participant 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).

The participant said that they thought 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”.

47
7.2.4 Participant 4

Participant 4 was assigned the Bot-FakeNews dataset. They began the study by looking at 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 (see Participant 4 questionnaire in Appendix A). 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. As the study allotted time continued, the participant made insights throughout their allotted time.

The participant mentioned that because of the interface they were able 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”.

During the post study questionnaire the participant stated 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 did find that they wanted more ways to know if a word was important in multiple subsets, or just the selected subset that they’re viewing. As a smaller interface improvement suggestions, such as a way to move the word cloud visualizations around the vis panel manually.

Finally, 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”.

48
7.3 Summary of Results

All participants gained some insights to 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 chapter we present a mock-up of what the interface could look like after accounting for the feedback received from the case study participants.
Chapter 8

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. 8.1 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 may also continue developing the interface to address 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 between what is a small word and what is a large word. A mock-up of the interface in Fig. 8.2 shows what this change could look like.

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. 8.2, an addition to the interface which could satisfy this requirement is the yellow highlighted background of words which appear in only one subset, indicating clearly to the user that the word is important in only the selected subset.

Since the development of NJM-Vis is guided by an iterative design process, future work would also include running more case studies to evaluate the proposed interface design changes; and more long term to run formal user studies, possibly comparing 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, such as ones that share more than one layer, or networks which are deeper or wider, may 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 used may produce stronger signals, and therefore our visualization may better explain predictions to users (e.g. 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 Chapter 4. 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 may give better results than LRP.
Figure 8.1: A mock-up to potentially expand the number of user selections at once. This shows a way of potentially splitting the vis panel into sections by dynamically decreasing the number of graph visualizations in such a way as to fit the most number of graphs into each selection window. 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 as explained in Figure 8.2.
Figure 8.2: A mock-up to address case study feedback. In the vis panel, the size scale has been increased to make it easier to differentiate between high contributing words (large font) and low contributing words (small font). Additionally, words which have a high frequency in only their one selected subset have a highlighted background, as seen with the word “points”.
Chapter 9

Conclusion

The contribution of this thesis 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. For the purposes of analyzing with our interface, we designed two joint task neural networks using Tensorflow in Python. We used 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. The task model included tasks: measure predictions of two models quantitatively, identify key words in subsets of predictions, dissect linguistic similarities/differences 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. The
design 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. The participants had an opportunity to use the interface to explore two different datasets. The first dataset was comprised of data from the AMI corpus and 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 useful feedback from which further versions of our prototype will be informed. Users noted that the interface could have an indication for which words were important in only one subset, i.e. that the interface could indicate if a word had a high relevance score or high frequency of occurrence 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 could continue to improve our contribution to the emerging field of explainable deep learning.
Bibliography


58


Appendix A

Supporting Materials

A.1 Case Study Questionnaires

A.1.1 Participant 1

Please use the interface to explore the data however youd like. If you gain any insights about the dataset from using the interface, please write them below

Could informative

Did you gain any insights on why the jointly trained model has outperformed the single trained model on this dataset? If so, please write them below

POST-STUDY QUESTIONNAIRE

Please answer the following questions. What did you like about the interface? What did you find to be useful? I liked the idea of comparing two different models. Being able to look at two models at the same time was useful. Different sizes of the words that indicate predictive power were useful.

What did you dislike about the interface? How can the visualization be improved? - It was difficult to know what size was considered big and what size was considered small. - similar sizes made me difficult to understand whats going on in the cell. - I gave some feedback during the study. - not only frequencies, but I also want to know the words that are frequent in that subset, but not common in other subsets. - I wanted to see a full text and which sentences were selected for a summary or not. - I wanted to see more subsets at the same time so that I dont need
to remember my observations. It was not easy to identify differences between two selections.

Do you have any further comments on what you learned about the data from using the interface? - light verbs ... Modal verbs. (e.g., would, could) were big, which was surprising to me. Dave said it was a dialog dataset, then now it makes sense. - for the false positive subset the most frequent words were very different for the two models. I'm not sure why, but I wanted to explore it more because the words were very similar for true positive and false negative...

Would you use this or similar interfaces in the future for your multi-task problems? Yes, if it is enhanced a little more.

A.1.2 Participant 2

Please use the interface to explore the data however you'd like. If you gain any insights about the dataset from using the interface, please write them below

Many of the tweets are about the stock market, trading, auctions, and money, and these were mostly predicted as true positives (from bots).

The tweets predicted to be from people (true negatives) had no real overarching topic, though they seem to be more personal such as I know or the name, Aidan.

The true positives focus on tweets that mention blogs and other posts.

The false positives about fitbits seem like they're from ads. The false positives seem like casual comments/replies.

Did you gain any insights on why the jointly trained model has outperformed the single trained model on this dataset? If so, please write them below

Some of the false negatives that the joint model fixed to true positive were still present in the single task true positives, e.g., gold, silver, etc.

The joint task true negatives captured more words than the single task true negatives.

The joint task true positives has deeper trees than the single task true positives.

The single task connected words that were not supposed to (e.g., single task false negatives has one large tree vs. joint task true positives has similar words in two trees).

The single task false negatives has a lot of overlap with its true positives, but it
is not the case for joint task.

**POST-STUDY QUESTIONNAIRE**

*Please answer the following questions.* What did you like about the interface? What did you find to be useful?

I like the visual aspect, the colours, and the clear organization of the left-hand menu. The summary on the left-hand side was very useful, and provided a fast and easy way to organize the results and know the difference in the results between the two models.

I also like how I can click on any word and see all its appearances on the same page. The side-by-side comparison of two things was extremely useful as well.

**What did you dislike about the interface? How can the visualization be improved?**

The trees can be improved; for example, the colour and text size can be made more obvious. It was sometimes not immediately obvious that one word was larger than another. Perhaps having a different colour than the ones used to indicate the model for the tree would be better. Currently, the single task model and the connection between words share the same yellow colour.

A bug already mentioned is that the words may overlap if there are too many of them to fit.

**Do you have any further comments on what you learned about the data from using the interface?**

It was obvious what kind of tweets were classified as fake (from bots) but not obvious what kind of tweets are real.

**Would you use this or similar interfaces in the future for your multi-task problems?**

Yes, having everything fit on one page—the summary (left-hand), the visualization (center), and the details (right-hand)—is a very good, easy to use, and convenient interface to see and do several things at once.

**A.1.3 Participant 3**

Please use the interface to explore the data however youd like. If you gain any insights about the dataset from using the interface, please write them below
Expect to try and understand *why* the joint fixed what it did and *why* it broke what it did - eg. look at why joint fixed false negatives to true positives by looking at single-task false negatives and fixed - looking at word groupings that appeared in both - implication: single-task considered it important for negative label, but joint fixed it because of that word grouping - eg. remote control - looking at word groupings that appear in fixed but not in single-false-negative - this would indicate that the single-task model did not consider it significant enough to warrant a positive - presumably that would also be the case for fixing false positives as well as broken

I think I was overthinking things - can look at why the joint fixed things by looking only at fixed, for example

The joint seemed to use remote-control to both fix false negatives and break true negatives (meaningful words that appeared in both sets)

Did you gain any insights on why the jointly trained model has outperformed the single trained model on this dataset? If so, please write them below

Bar charts are better for the true pos/neg and worse for the false pos/neg, so outperforms in every category

Might be able to look at word groupings that appear in the true positive but not false negative for the joint and not in the true positive for the single, but that would require a fair bit of cognitive load - might be useful to perform such an inference automatically and visualize that

**POST-STUDY QUESTIONNAIRE**

*Please answer the following questions. What did you like about the interface? What did you find to be useful?*

- The bar charts, especially the fixed and broken columns - it took me a while to get used to the word graphs (I kept thinking of the pairings as phrases instead of just words in the same sentence, despite Dave telling me otherwise), but the size differential is useful - being able to compare two sets of word groupings to try and perform further inference (though that is a fair bit of cognitive load)

**What did you dislike about the interface? How can the visualization be improved?**

- Multiple copies of words in the same graph might make it harder to make connections - the size differential could be more pronounced - Connecting words
based on whether they're 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) - have the displayed sentences include the removed stopwords for readability - being able to click on a sentence and see it in the broader context might be useful - having the label for the other task shouldn’t appear to be part of the sentence; a separate column for that may be more useful

Do you have any further comments on what you learned about the data from using the interface?
- many of the important word groupings seem to be from words that don't seem that important (yeah, oh, one, etc.) - since the TP/TN/FP/FN scores were so similar for the two models, the fixed and broken columns seemed to be the most informative

Would you use this or similar interfaces in the future for your multi-task problems?
I think so. I feel that it took me longer than it should have to get used to the word graphs, and the graphs seem to include many words that I wouldn’t consider important, but that in itself is a useful insight that I wouldn’t have had from just looking at a confusion matrix. I’d like to be able to see how it handles a better dataset. The tool may have a use as a sanity check to see if the dataset and/or annotations are any good.

A.1.4 Participant 4

Please use the interface to explore the data however you'd like. If you gain any insights about the dataset from using the interface, please write them below

The word fiddling appears to me some kind of code for some bots' tweets
There are more influential words that contribute to true positives (bots), than to true negative (humans)
forex is important for false negatives joint and single, as well as true positive joint and single. (i.e. the vis helped me see that forex is an important word for classifying a bot)

Did you gain any insights on why the jointly trained model has outperformed the single trained model on this dataset? If so, please write them below
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, contrastingly the joint task does not have offline included at all.

**POST-STUDY QUESTIONNAIRE**

*Please answer the following questions.*  **What did you like about the interface? What did you find to be useful?**

I liked that the interface showed influential words that contributed to each task for true/false positives and true/false negatives. I also appreciated how the word graph denoted the importance of a word from its size. This made looking at the visualization easy to breakdown and understand.

Most importantly I liked that I could compare to 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.

**What did you dislike about the interface? How can the visualization be improved?**

- way to know if something was posted by the same for a given word (i.e. how many of the fiddling sentences were posted by the same user) - way to move the graphs around (i.e. maybe I want one on top of the other for some reason) - way to know if the words visualised are important across the different categories (i.e. forex is important for false negatives joint and single, as well as true positive joint and single) - indicate the importance cut off for a word to be visualized

**Do you have any further comments on what you learned about the data from using the interface?**

**Would you use this or similar interfaces in the future for your multi-task problems?**

Absolutely. 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.
A.2 Demo Dataset

As seen in Fig. A.1 the demo of the interface used during the training portion of the case study. The demo was done on a very simple toy dataset.

**Figure A.1:** A screenshot of the visualization produced by the toy dataset used for demo purposes during the training portion of the case study.