DeepSI: Interactive Deep Learning for Semantic Interaction

YALI BIAN, Virginia Tech, United States
CHRIS NORTH, Virginia Tech, United States

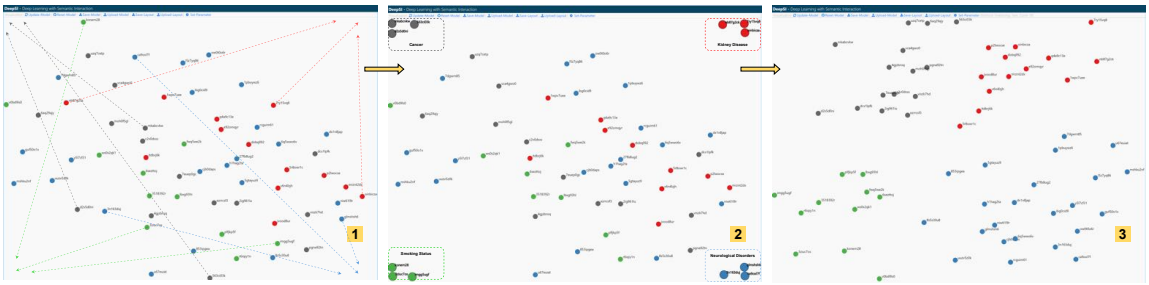


Fig. 1. Screenshots during the analysis of COVID-19 research articles about four risk factors (depicted in different colors) using our proposed model DeepSI$_{\text{finetune}}$: (1) the initial layout of all articles projected from pretrained BERT representations of the raw text data; (2) the analyst performs semantic interactions to provide visual feedback regarding articles about different risk factors; these interactions are then exploited to tune the underlying DL model BERT; (3) the resulting projection updated by the tuned BERT.

In this paper, we design novel interactive deep learning methods to improve semantic interactions in visual analytics applications. The ability of semantic interaction to infer analysts' precise intents during sensemaking is dependent on the quality of the underlying data representation. We propose the DeepSI$_{\text{finetune}}$ framework that integrates deep learning into the human-in-the-loop interactive sensemaking pipeline, with two important properties. First, deep learning extracts meaningful representations from raw data, which improves semantic interaction inference. Second, semantic interactions are exploited to fine-tune the deep learning representations, which then further improves semantic interaction inference. This feedback loop between human interaction and deep learning enables efficient learning of user- and task-specific representations. To evaluate the advantage of embedding the deep learning within the semantic interaction loop, we compare DeepSI$_{\text{finetune}}$ against a state-of-the-art but more basic use of deep learning as only a feature extractor pre-processed outside of the interactive loop. Results of two complementary studies, a human-centered qualitative case study and an algorithm-centered simulation-based quantitative experiment, show that DeepSI$_{\text{finetune}}$ more accurately captures users' complex mental models with fewer interactions.

CCS Concepts: • **Human-centered computing** → *Interaction techniques*; **Visual analytics**; • **Computing methodologies** → *Natural language processing*; *Learning from demonstrations.*

Additional Key Words and Phrases: Semantic Interaction, BERT, Visual Analytics, Interactive Deep Learning

ACM Reference Format:
Yali Bian and Chris North. 2021. DeepSI: Interactive Deep Learning for Semantic Interaction. In *26th International Conference on Intelligent User Interfaces (IUI '21), April 14–17, 2021, College Station, TX, USA.* ACM, New York, NY, USA, 17 pages. https://doi.org/10.1145/3397481.3450670

Manuscript submitted to ACM

1 INTRODUCTION

Semantic interaction (SI) [20, 21] is an interaction methodology that is commonly utilized to enhance visual analytics (VA) systems. SI-enabled systems let the analyst directly manipulate interactive projections of data [58]. The semantic meaning behind these projection interactions is the similarity relationships the analyst wishes to find within the data during the sensemaking process [49]. As shown in Fig. 1-2, the analyst drags 12 COVID-19 article points into four clusters to provide the visual feedback of grouping articles based on their perceived relevant risk factors. With these intuitive and natural interactions, the analyst can remain within the cognitive zone [25], thereby enhancing the analyst's efficiency in performing analytic tasks [65]. In the system, an interactive dimensionality reduction (DR) component [54, 65] plays a key role in capturing the analyst's intent behind these interactions by learning a new projection layout (Fig. 1-3). To determine the analyst's precise intent, increasingly powerful interactive DR models [65] have been proposed, from linear [29, 30, 37] to non-linear models [35, 40], and from single-model to multi-model approaches [10, 18, 19, 66].

However, the ability of semantic interaction to infer analysts' precise intents during sensemaking is dependent on the quality of the underlying data representation. Deep learning (DL) [36] is a state-of-the-art representation learning method [5], which can automatically extract abstract and useful hierarchical representations from raw data [6]. This offers the new opportunity to power SI in capturing the analyst's intent. We denote the DL-enhanced SI system as **DeepSI**. Previous researches have shown that even the usage of the pretrained DL representations as fixed data features have better performance than hand-crafted features in SI-enabled VA systems [7, 8]. We denote this straightforward DeepSI design with the basic use the pretrained DL as only a feature extractor in SI pipeline as $\text{DeepSI}_{\text{vanilla}}$.

In this paper, we aim to further improve semantic interaction inference by fine-tuning the model to obtain user- and task-specific representations from the pretrained DL model. Central to this design goal are two research questions:

- *How to exploit semantic interactions to accurately adapt the pretrained representations to current analytic tasks?*
- *How to make efficient adaptations, so that a small number of semantic interactions are enough for analysts to express their intents?*

To address these two questions, we propose a novel DeepSI framework, $\text{DeepSI}_{\text{finetune}}$, with the following two design goals. First, we insert the interactive DL training into the bidirectional structure of the semantic interaction pipeline, so that interactions trigger the DL adaptation. Thereby, new user- and task-specific representations are generated based on semantic interactions provided by analysts during their sensemaking process. Second, we employ the fine-tuning based DL adaption approach and the MDS-based interactive DR model to minimize the number of parameters that require training in the underlying model. Therefore, $\text{DeepSI}_{\text{finetune}}$ can tune the DL model efficiently from the analyst's interactions without information loss. Specifically, we use the pretrained BERT [16], a state-of-the-art DL model for NLP tasks, as the DL model representative inside $\text{DeepSI}_{\text{finetune}}$ for visual text analysis tasks.

To assess how well $\text{DeepSI}_{\text{finetune}}$ addresses these questions by integrating DL into the semantic interaction loop, we compare it with the well-evaluated baseline model $\text{DeepSI}_{\text{vanilla}}$ [7, 8], which uses DL outside of the interactive loop, in two complementary experiments: a human-centered qualitative case study about COVID-19 academic articles; and an algorithm-centered simulation-based quantitative analysis of three commonly used text corpora: Standford Sentiment Treebank (SST), Vispubdata, and 20 Newsgroups. The results of both experiments show that $\text{DeepSI}_{\text{finetune}}$ not only captures the analyst's precise intent more accurately, but also requires fewer interactions from the analyst.

Specifically, we claim the following contributions:

(1) The $\text{DeepSI}_{\text{finetune}}$ framework that integrates DL into the human-in-the-loop iterative sensemaking pipeline to improve semantic interaction inference.

(2) Two complementary studies, a user-centered qualitative case study and an algorithm-centered simulation-based quantitative experiment, that measure the performance of our method and reveal improvements.

2 RELATED WORKS

Four related components support our design: interactive DR models used in semantic interaction; basic knowledge of the DL model BERT; pretrained DL model adaptation approaches; and other work about user-centered interactive DL.

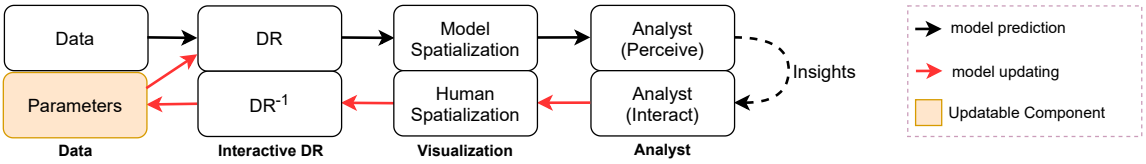


Fig. 2. SI pipeline showing the communication between the analyst and VA system, adapted from [21, 56]. The interactive DR component is responsible for capturing the analyst's intent from the human modified projection (denoted as human spatialization) and, consequently, updating the projection in response (denoted as model spatialization).

2.1 Interactive Dimensionality Reduction

While using the human-in-the-loop sensemaking SI pipeline (Fig. 2), analysts gain insights from the model projection (spatialization) and express their preferences by repositioning data points in the projection. It is the interactive DR model's responsibility to learn new model parameters that capture the analyst's intent behind the modified projection and, in response, use the learned parameters to update the projection. Therefore, increasingly powerful DR models have been adapted in a semi-supervised manner to improve SI inference. VA frameworks V2PI [37] and BaVA [29] adapted linear DR models, including principal component analysis (PCA) [67] and weighted multidimensional scaling (WMDS) [55], to the bidirectional SI pipeline. To support more complex tasks and interactions, multiple models were chained together as a single interactive DR model, which is called multi-model SI [10, 19, 66]. Recently, to adapt more powerful but complex non-invertible DR algorithms, such as t-SNE [39] and UMAP [41], Zexplorer [40] used the invertible neural encoder [22] to emulate these models as the interactive DR model in SI applications.

Similarly, $\text{DeepSI}_{\text{finetune}}$ also aims to improve SI inference. However, $\text{DeepSI}_{\text{finetune}}$ highlights the importance of finding user- and task-specific data representations instead of more powerful DR. Therefore, we use the simple but commonly used WMDS as the default DR [11, 17, 56, 57] and focus on extracting meaningful DL representations.

2.2 BERT

BERT (Bidirectional Encoder Representations from Transformers) [16] is a DL language representation model. BERT is first pretrained on raw text data to learn general language representations. The pretrained BERT then can be easily adapted to downstream NLP tasks such as sentiment analysis and semantic textual similarity [13]. The adapted BERT model is able to provide task-specific representations and shows state-of-the-art performance in these downstream tasks. Technically, BERT is a Transformer encoder [62], containing a stack of transformer layers. The transformer layer learns token-level representations. For input token sequences, the transformer layer learns a new vector for each token based on all other tokens, using the self-attention mechanism [3]. Through the stack of transfer layers, BERT can convert a sequence of tokens into deep representations. In this paper, we use the pretrained BERT model as the default DL model in the DeepSI pipeline to provide text representations for visual text analytic tasks.

Table 1. A list of variables used throughout this paper and their descriptions.

Variable	Description
d	A set of documents for analysis
N, M	Number of samples in d, number of dimensions of d
n	Number of samples moved by the analyst, $n \ll N$
x	High-dimensional feature of d. DL representations (768 dimension-size BERT embeddings)
y	Coordinates in the 2D visual spatialization of d. Set either by analysts' interactions on the visualization, or by the underlying SI model, which maps x to y
$w_{\text{dimension}}$	Parameters of dimension weights of x. (a 768 dimension-size vector)
w_{BERT}	Internal parameters of the pretrained BERT model. BERT_{base} is used in this paper, which contains 110 million parameters
$dist$	Euclidean distance between data samples in d. Weighted Euclidean distance is used if $w_{\text{dimension}}$ applied. $dist_H$ defines the high-dimensional similarity. $dist_L$ defines the low-dimensional similarity

2.3 Pretrained Representation Adaptation

There are two main paradigms to adapt the pretrained DL representation model to downstream tasks: feature extraction and fine-tuning [48]. The feature extraction approach uses a task-specific architecture to adapt the pretrained representations to downstream tasks (e.g. ELMo [47]). In this approach, parameters inside the task-specific architecture are trained on the downstream tasks. In contrast, instead of using a new architecture, the fine-tuning method appends one additional output layer to the pretrained DL and tunes the whole pretrained model with the downstream tasks. The fine-tuning approach requires relatively less training data because it introduces minimal task-specific parameters and does not need to learn randomly initialized task-specific parameters from scratch. Therefore, we use the fine-tuning approach in the $\text{DeepSI}_{\text{finetune}}$ framework to adapt the pre-trained DL model to visual analytic tasks.

2.4 Human-centered Deep Learning

There are other human-centered DL techniques proposed to assist users in complex data analytic tasks. Hsueh-Chien et. al. [14] used CNN techniques to assist users in volume visualization designing through facilitating user interaction with high-dimensional DL features. In RetainVis [34], an interactive and interpretable RNN model was designed for electronic medical records analysis and patient risk predictions, which can be steered interactively by domain experts. Gehrmann et al. [24] proposed a framework of collaborative semantic inference that enables the visual collaboration between humans and DL algorithms. Sharkzor [50] is an interactive deep learning system for image sorting and summary, based on users' semantic interactions. Of these, Sharkzor is the most similar to our $\text{DeepSI}_{\text{finetune}}$. Both works provide users with semantic interactions to tune the DL model interactively. However, our work emphasizes a general solution to integrate DL models into SI systems to improve inference. While Sharkzor is only designed for image analysis, $\text{DeepSI}_{\text{finetune}}$ can be applied to other data analytic tasks and relevant DL models.

3 BACKGROUND

In order to frame our discussion of our model $\text{DeepSI}_{\text{finetune}}$, this section briefly describes DeepVA, the state-of-the-art SI model with pretrained DL [8]. For the purpose of comparison, we implement a specific version of DeepVA that uses BERT, which we denote as $\text{DeepSI}_{\text{vanilla}}$. For reference, Table 1 describes frequently used variables throughout this

paper. We use the pretrained BERT model as a representative DL model in DeepSI system designs. Note that WMDS is used as the default interactive DR in DeepSI frameworks for three reasons. First, the WMDS is a simple linear DR algorithm, so that we can focus on assessing the effects of data representations on the model performance. Second, WMDS is agnostic to the choice of the weighted distance function. Third, WMDS enables analysts to express their synthesis process by manipulating data point proximities to reflect their perceived similarity [58].

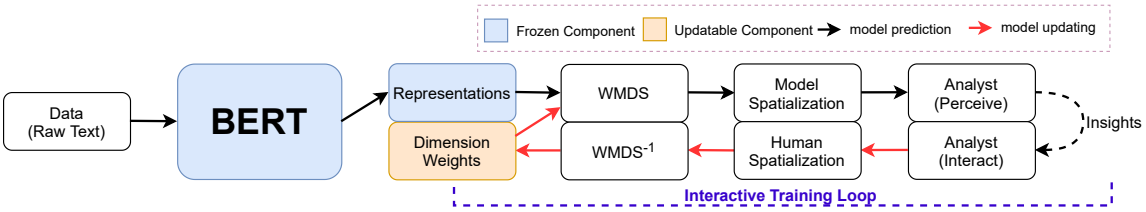


Fig. 3. DeepSI$_{\text{vanilla}}$ pipeline, adapted from [7]: using the pretrained BERT as only a feature extractor pre-processed outside of the interactive loop in SI pipeline. All parameters inside the pretrained BERT model are frozen and the output data representations are fixed. WMDS is the interactive DR, which is responsible to tune the dimension weights $\boldsymbol{w}_{\text{dimension}}$ to capture the analyst's intent.

DeepSI$_{\text{vanilla}}$ uses the DL model as only a feature extractor in the SI pipeline. As shown in Fig. 3, the pretrained parameters inside the BERT model are frozen. Thereby, for an input, BERT provides a fixed general-purpose representation, which is then used as the data features in the interactive training loop. The BERT model is outside of the interactive loop. Therefore, the interactive DR model, WMDS, is responsible for updating dimension weights $\boldsymbol{w}_{\text{dimension}}$ to capture the analyst's intent as a weighting of the BERT features. The complete process of the pipeline is as follows.

Before entering the interactive training loop, the data representations are initialized by the BERT model with the pretrained parameters $\boldsymbol{w}_{\text{BERT}}$:

$$\boldsymbol{x} = \text{BERT}(d, w_{\text{BERT}}) \tag{1}$$

In the forward model-prediction direction, WMDS is performed to project high-dimensional data points ($\boldsymbol{x}$) into the two-dimensional spatialization ($\boldsymbol{y}$), with current dimension weights $w_{\text{dimension}}$ (initially, all weights are equal). This provides a new projection for the analyst to perceive and interact.

$$\boldsymbol{y} = \arg\min_{y} \sum_{i<j\leq N} \left(dist_L(y_i, y_j) - dist_H(x_i, x_j, w_{\text{dimension}}) \right)^2 \tag{2}$$

In the backward model-updating direction, the analyst provides visual feedback by repositioning n data points within the projection. WMDS^{-1} uses the low-dimensional pairwise distances between the moved n data points as input, to learn new dimension weights $\boldsymbol{w}_{\text{dimension}}$ to make sure these moved data points have similar relationships in the high-dimensional space, based on the following optimization criterion:

$$\boldsymbol{w}_{\text{dimension}} = \arg\min_{w} \sum_{i<j\leq n} \left(dist_L(y_i, y_j) - dist_H(x_i, x_j, w_{\text{dimension}}) \right)^2 \tag{3}$$

Therefore, through this loop, the dimension weights $\boldsymbol{w}_{\text{dimension}}$ are trained interactively and incrementally based on analysts' interactions to capture their intents.

DeepSI$_{\text{vanilla}}$ has been well-evaluated previously. DeepVA [8] used ResNet [27] as an image data feature extractor in the SI system that assists users performing visual concepts analysis using DL representations. In [7], Bian et al. compared SI systems that use embedding vectors as features and those that use bag-of-words as features in visual text analysis tasks.

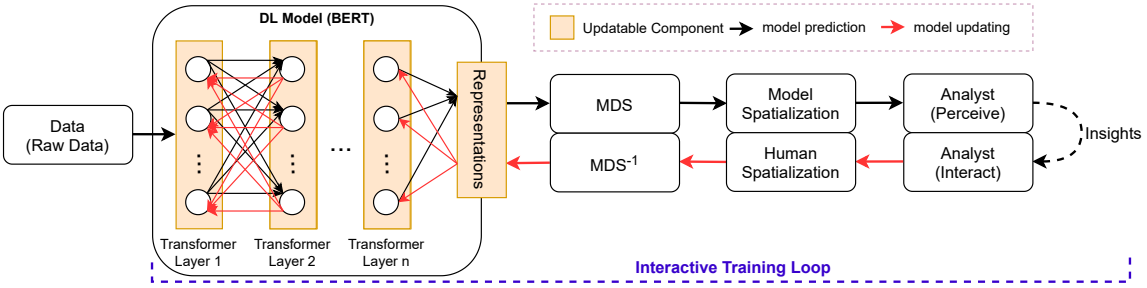


Fig. 4. DeepSI$_{\text{finetune}}$ pipeline: embedding BERT within the SI loop. Semantic interactions are exploited to fine-tune BERT interactively through backpropagation. The tuned BERT is responsible for generating new representations, so as to capture the analyst's intent. Thereby, no external parameters are needed.

Experiments in both works show that even the general-purpose representations of pretrained DL models can enable SI to better capture the analyst's intent than hand-crafted features. However, using the general-purpose pretrained representations still restricts SI inference. In the next section, we propose DeepSI$_{\text{finetune}}$, which exploits fine-tuned representations to further improve SI inference. As the best-performing model from previous studies, DeepSI$_{\text{vanilla}}$ is the baseline model for comparison.

4 MODEL DESCRIPTION

This section outlines the main design, model pipeline, and implementation details of DeepSI$_{\text{finetune}}$.

4.1 Model Design

We propose two main design goals to address the two research questions discussed in Sec. 1.

Design goal 1 - Integrating DL into the human-in-the-loop interactive sensemaking pipeline. To get user- and task-specific representations, it is necessary to iteratively train the DL model with semantic interactions during the human-in-the-loop process. Inspired by multi-model SI systems [10, 19, 66], we inserted the DL model update and prediction process into the bidirectional semantic interaction loop, as shown in Fig. 4. The DL model update and prediction process occurs before the interactive DR model. The interactive DR model passes the analyst's visual feedback from the human spatialization to the DL model. The visual feedback is then used to update the parameters inside the DL model ($\boldsymbol{w}_{\text{BERT}}$) through the DL backpropagation [53] (red arrows inside the DL component). With updated parameters $\boldsymbol{w}_{\text{BERT}}$, the BERT model calculates new representations for input data by the forward propagation [70] through the internal transformer layers (black arrows inside the DL component). Through the interactive sensemaking process, the DL model is trained by semantic interactions in an interactive machine learning setting [2, 23]. Thereby, improved representations are generated to accurately capture the analyst's intent.

Design goal 2 - Introducing minimal parameters into the interactive DL training pipeline. To solve analytic tasks efficiently, the analyst prefers to perform fewer interactions in each sensemaking loop. However, DL model training typically needs a relatively large amount of training data. To reduce the number of interactions needed for training, we should introduce minimal parameters into the pipeline while integrating the DL model training. For this design goal, we made specific modifications to both the DL and the interactive DR components. First, we used the fine-tuning approach to adapt the pretrained BERT model with semantic interactions. Unlike the feature-based method, fine-tuning approach introduced minimal task-specific paramters [26]. This drastically reduced the required training

data. Further, we used MDS/MDS^{-1} as the interactive DR component. During the interactive BERT model training, representations are updated to capture analysts' intents. It is unnecessary to tune extra parameters for the same purpose in the interactive DR model. Therefore, we used MDS/MDS^{-1} without dimension weights $\boldsymbol{w}_{\text{dimension}}$ as the interactive DR component. There are no parameters to tune in this component. Therefore, users' interactions can be passed directly to DL training without information loss.

4.2 Model Pipeline

We illustrate the DeepSI$_{\text{finetune}}$ pipeline (Fig. 4) in detail through the human-in-the-loop sensemaking process. In the forward model-prediction direction, new representations are generated for the dataset d through the forward propagation calculation of the BERT model, with current BERT parameters ($\boldsymbol{w}_{\text{BERT}}$):

$$\boldsymbol{x} = \text{BERT}(d, w_{\text{BERT}}) \tag{4}$$

The high-dimensional DL representations x are then projected to the 2D spatialization (model spatialization) by MDS through the following equation:

$$\boldsymbol{y} = \arg\min_{y} \sum_{i<j\leq N} \Big(dist_L(y_i, y_j) - dist_H(x_i, x_j)\Big)^2 \tag{5}$$

In contrast to Eq. 2, the high-dimensional distance function is not explicitly weighted. Instead, the updates to $\boldsymbol{y}$ in each loop are captured by the fine-tuned representation x itself. The analyst perceives the updated spatialization and gains insight.

In the backward model-updating direction, the analyst modifies the visual layout (human spatialization) by repositioning some samples to express the preferred similarities between them. Then, MDS^{-1} uses the human-defined similarities between n moved data points, $dist_L(y_i, y_j)$), to steer the BERT model parameters to generate better high-dimensional representations x, such that the similarity of the representations reflects the proximity of the points in the modified projection, as follows:

$$w_{\text{BERT}} = \arg\min_{\boldsymbol{w}_{\text{BERT}}} \sum_{i<j\leq n} \Big(dist_L(y_i, y_j) - dist_H(\text{BERT}(d_i, w_{\text{BERT}}), \text{BERT}(d_j, w_{\text{BERT}}))\Big)^2 \tag{6}$$

The optimization objective is to fine-tune BERT weights $\boldsymbol{w}_{\text{BERT}}$ to minimize the difference between low-dimensional and high-dimensional distances of n moved data points through backpropagation. All internal parameters of the BERT model ($\boldsymbol{w}_{\text{BERT}}$) are updated in order, from last transformer layers to previous layers, by a gradient descent optimization algorithm [52]. After the backpropagation, the updated $\boldsymbol{w}_{\text{BERT}}$ is used in the forward propagation to calculate new representations $\boldsymbol{x}$ in Eq. 4.

Through this human-in-the-loop interactive DL process, the BERT model is tuned properly to generate user- and task-specific representations, so as to capture analysts' precise intents: samples that should be closer to each other in the visualization obtain similar features, while more distant samples gain differing features.

4.3 Prototyping Detail

Here, we describe the implementation details of DeepSI prototypes used in our experiments, including model settings and visualization design. These implementations are applicable for both DeepSI$_{\text{finetune}}$ and DeepSI$_{\text{vanilla}}$.

4.3.1 Model Settings. We use Pytorch [44], a well-known Python DL framework, to implement the DeepSI system. For the forward DR component, MDS is adapted from Scikit-Learn [45]. The MDS^{-1} is implemented in Pytorch as a neural network layer. The pretrained BERT model is adapted from the publicly available Python library, Transformers [68]. Transformers provides two sizes of pretrained BERT models: $\text{BERT}_{\text{BASE}}$, and $\text{BERT}_{\text{LARGE}}$. We used the small BERT model ($BERT_{BASE}$) (bert-base-uncased, 12-layers, 768-hidden, 12-heads, 110M parameters), because it is more stable on small datasets. For a document containing a list of tokens, $\text{BERT}_{\text{BASE}}$ can convert each of the tokens into a 768-dimensional vector. To generate fixed-length encoding vectors from documents of different lengths, we appended a MEAN pooling layer to the last transformer layer of the BERT model, such that the output representation for a document was a 768-dimensional vector. Therefore, the $w_{\text{dimension}}$ used in $\text{DeepSI}_{\text{vanilla}}$ is also a 768 dimension vector. We also tested other pooling strategies, such as MAX pooling and CLS pooling [51]. However, there was no obvious performance difference, and the MEAN pooling showed slightly better performance. In addition, we used the Adam optimizer [33] to optimize the DeepSI model parameters in the model-updating direction. We also explored other optimizers provided by PyTorch. Across all our experiments, we found that Adam optimizer performed the best. Further, we found that the suggested learning rate ($3e^{-5}$) for finetuning BERT models in [16] led to optimal $\text{DeepSI}_{\text{finetune}}$ performance in experiments.

4.3.2 Visualization Design. We drew inspiration for visualization design from SI-enabled VA applications, including Andromeda [58], ForSPIRE [21], and Dis-Function [11]. As shown in Fig. 1, the visual interface mainly uses a scatterplot as the projection layout. This scatterplot not only displays the relationships between data updated by the underlying projection model, but also allows the analyst to intervene and modify the layout. Specifically, in the forward model-prediction direction, the positions between data points on the scatterplot reflect the points' relative similarity learned by underlying models, either by the projection method or by the fine-tuned BERT model, shown in Fig. 1-1 (model spatialization). In the backward model-updating direction, the user can drag several data points to new positions to modify similarities between points based on their preference, shown in Fig. 1-2 (human spatialization). Having both the underlying models and analysts work on the same visualization provides direct and effective communication between humans and computation. In addition to the scatterplot view, the prototype also provides a sidebar view to help analysts review the content of a selected document when exploring in the scatterplot view. In this paper, we intentionally focus on the scatterplot view in the screenshots, to focus on the analysis of the model performance.

5 EXPERIMENTS

To evaluate $\text{DeepSI}_{\text{finetune}}$, we conducted the following experiments. To examine how well $\text{DeepSI}_{\text{finetune}}$ addresses the goals, we measured its performance in two respects:

- **Accuracy**: How accurately can $\text{DeepSI}_{\text{finetune}}$ capture the analyst's intent?
- **Efficiency**: How many interactions does $\text{DeepSI}_{\text{finetune}}$ need to capture the analyst's intent properly?

We use $\text{DeepSI}_{\text{vanilla}}$, described in Sec. 3, as the baseline model to evaluate the advantage of $\text{DeepSI}_{\text{finetune}}$'s task-specific, instead of general-purpose, representations. Boukhelifa et al. [9] proposed the complementary evaluation of interactive machine learning systems by using both algorithm-centered and human-centered evaluation methods. We perform both evaluation methods in our experiments: the case study in Sec. 5.1 is the human-centered qualitative analysis, and the simulation-based evaluation method in Sec. 5.2 is the algorithm-centered quantitative analysis.

5.1 Case Study: COVID-19

Recently, COVID-19 [61] has become a global pandemic. It is essential that medical researchers quickly find relevant documents about a specific research question, given the extensive coronavirus literature. We used an analysis task on academic articles related to COVID-19 in this case study to examine our proposed $\text{DeepSI}_{\text{finetune}}$, compared with the baseline model $\text{DeepSI}_{\text{vanilla}}$. In this study, we performed the same task with the help of both DeepSI prototypes and then measure the model performance in the following two perspectives:

- **Accuracy**: the quality of the projection updated by the underlying model given the task's ground truth.
- **Efficiency**: how many interactions are needed for the underlying model to provide a useful projection.

5.1.1 Dataset and Task. The COVID-19 Open Research Dataset (CORD-19) [1] contains a collection of more than 200,000 academic articles about COVID-19. CORD-19 also proposes a series of tasks in the form of important research questions about the coronavirus. One of the research tasks focuses on identifying COVID-19 risk factors [2]. In this case study, we selected a task that requires identifying articles related to specific risk factors for COVID-19. We asked an expert to choose as many research papers as possible about risk factors from CORD-19. We found four main risk factors: cancer (15 articles), chronic kidney disease (13 articles), neurological disorders (23 articles), and smoking status (11 articles). We used these four risk factors as the ground truth for the test task, and loaded all these 62 articles into our DeepSI prototypes. Therefore, the test task was to organize these 62 articles into four clusters with our DeepSI prototype such that each cluster represented articles of a specific risk factor.

This particular ground truth is just one possible way an analyst might want to organize this group of documents. So our goal is to see if this particular set of expert knowledge can be easily injected using SI to help re-organize the documents in this particular way. To help judge the quality of the visual layout in organizing this particular ground truth, we color the dots according to the ground-truth risk factors in the visualization: **cancer** (black dot •), **chronic kidney disease** (red dot •), **neurological disorders** (blue dot •), **Smoking status** (green dot •). It should be noted that the underlying model was not provided with the ground truth or color information. The ground truth is only injected via semantic interaction from the human in the form of partial groupings of only a few of the documents.

5.1.2 Study Procedure. To compare the projection layouts updated by two models based on the same input interactions, semantic interactions based on the ground truth are performed in the shared visual projection and then applied to the two models separately. Fig. 1 and Fig. 5 show the process of interactions applied separately to $\text{DeepSI}_{\text{finetune}}$ and $\text{DeepSI}_{\text{vanilla}}$ prototypes. In both figures, frame 1 and frame 2 are from the shared visual projection. Frame 1 in both figures (Fig. 1-1 and Fig. 5-1) shows the same initial layout updated by the default pretrained BERT model. In the initial projection layout, all the articles are combined. This means that the pretrained BERT model cannot distinguish these articles by their related risk factors. Interactions were performed within the projection based on the ground truth to reflect the perceived connections between articles: grouping three articles about cancer to the top-left region of the projection, indicated by the black arrows; three articles about chronic kidney disease to the top-right region indicated by red arrows; three articles about smoking status to the bottom-left part indicated by green arrows; and three articles about about neurological disorders to the bottom-right part indicated by blue arrows.

Frame 2 (Fig. 1-2 and Fig. 5-2) is the same human spatialization and shows four clusters created by the ground truth. After we clicked the 'model update' button on the menu bar to start the model training process. Then, the same human

[1] https://www.kaggle.com/allen-institute-for-ai/CORD-19-research-challenge
[2] https://www.kaggle.com/allen-institute-for-ai/CORD-19-research-challenge/tasks?taskId=558

spatialization was used to train both models. After the two models had been updated, the updated projections of these two models showed the performance difference in frame 3 (Fig. 1-3 and Fig. 5-3). Subsequently, the performance of the model could be assessed based on how reasonable the layout was in comparison with the ground truth.

Fig. 5. Screenshots during the case study using DeepSI$_{\text{vanilla}}$: Frame 1 and 2 show the similar initial steps performed by the analyst in Fig. 1. Frame 3 shows the resulting projection updated by DeepSI$_{\text{vanilla}}$.

DeepSI$_{\text{finetune}}$ spatialization: The projection updated by DeepSI$_{\text{finetune}}$ is shown in Fig. 1-3. There are four clear clusters, and all articles are clearly grouped into the correct clusters. The top left cluster contains all the articles about cancer (●), the top right cluster contains articles about kidney disease (●), the bottom left contains articles about smoking status (●), and the bottom right contains articles about neurological disorders (●). This means the new representations generated by the fine-tuned BERT model are able to accurately capture the semantic meanings behind users' interactions.

DeepSI$_{\text{vanilla}}$ spatialization: With the same interactions as input, the updated DeepSI$_{\text{vanilla}}$ shows a different layout. As shown in Fig. 5-3, there are no clear clusters in the updated layout compared with Fig. 1-3. Articles about different risk factors still overlap. Even after continued interactions based on the ground truth, DeepSI$_{\text{vanilla}}$ is unable to properly capture the user's semantic intent and differentiate these articles.

Further study for DeepSI$_{\text{vanilla}}$: However, DeepSI$_{\text{vanilla}}$ did work well at separating articles into two clusters in two opposite positions in the projection. For example, in Fig 5-3, smoking status articles (●) are separated from kidney disease articles (●). Exploring further, after resetting the model as shown in Fig. 6-2, three neurological disorders articles (●) are dragged to the bottom-left and three chronic kidney disease articles (●) are dragged to the top-right on the scatterplot view. After the layout updates, articles from these two dragged clusters are well placed in two opposite sides of the visualization in Fig. 6-3, ignoring articles in the other two clusters (about cancer ● and smoking status ●).

Fig. 6. Further case study using DeepSI$_{\text{vanilla}}$ in grouping two clusters: Frame 1 is the initial projection layout, Frame 2 shows interactions performed within the projection, and Frame 3 shows the resulting projection updated by DeepSI$_{\text{vanilla}}$.

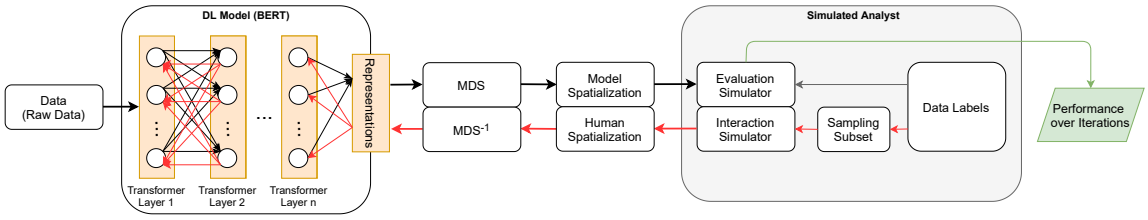


Fig. 7. Simulation-based evaluation pipeline. The analyst is replaced by the 'simulated analyst' component where: analyst perception is simulated by the kNN classifier and analyst interaction by sampling a subset of ground truth. In each SI loop, the kNN classification is employed to calculate the accuracy of the model spatialization, which is updated by the underlying model $\text{DeepSI}_{\text{finetune}}$ and reflects the model performance.

5.1.3 Qualitative Results. In terms of accuracy, $\text{DeepSI}_{\text{finetune}}$ grouped articles correctly based on the user-defined risk factors. In contrast, $\text{DeepSI}_{\text{vanilla}}$ did not provide a useful projection. The further study also confirmed that $\text{DeepSI}_{\text{vanilla}}$ can handle more straightforward tasks with only two clusters. The semantics of the ground truth knowledge provided by the contest organizers are more recognizable in the $\text{DeepSI}_{\text{finetune}}$ projection. Articles in each group are clearly clustered. However, $\text{DeepSI}_{\text{vanilla}}$ only partially separated in two separate directions, instead of into distinct clusters, which requires more cognitive effort to identify the boundary between the groups. In terms of efficiency, $\text{DeepSI}_{\text{finetune}}$ is more efficient than $\text{DeepSI}_{\text{vanilla}}$. $\text{DeepSI}_{\text{finetune}}$ needed a small number of interactions (moving three articles in each cluster, 12 dots movement in total) in one interactive SI loop to fine-tune the BERT model properly for this task. In contrast, in $\text{DeepSI}_{\text{vanilla}}$, the same amount of interactions only supported the simpler task with two clusters, and additional rounds of interaction still did not uncover all four clusters.

5.2 Simulation-based Evaluation

From the machine learning algorithm perspective, DeepSI systems are transductive models [71] that interactively learn projections provided by the analyst. Therefore, the performance of DeepSI systems can be measured by the predicted projections. To conduct the quantitative comparisons between the predicted projections from DeepSI systems, we replaced the analyst with a simulation component (simulated analyst). As shown in Fig. 7, the simulated analyst uses the interaction simulator to generate a training projection (human spatialization) based on data labels, and the evaluation simulator to evaluate the accuracy of the predicted projection. After training iteration, the simulated analyst outputs a current projection accuracy. The projection accuracy over iterations reflects the learning curve [46] of the DeepSI model. Therefore, performances of both DeepSI models could be compared through their learning curves in both accuracy and efficiency perspectives.

5.2.1 Simulated Analyst. As shown in the simulation pipeline (Fig. 7), data labels are the ground truth to support both interaction simulator and evaluation simulator. First, the interaction simulator uses these labels to calculate the pairwise distances between a subset of data samples, simulating the human-defined similarities between these samples. Further, the evaluation simulator uses these class labels to measure how well the predicted projection grouped data samples into correct classes based on their labels.

Interaction simulator: In each interaction, three samples from each class are selected using random sampling [60]. Then the interaction simulator calculates the pairwise distance $dist_L(y_i, y_j)$ of these selected samples based on:

$$dist_L(d_i, d_j) = \begin{cases} 0 & \text{if } d_i \text{ and } d_j \text{ have the same label} \\ \sqrt{2} & \text{otherwise} \end{cases}$$

As shown in the above Equation, if two selected samples have different labels, the distance between them is $dist_L(d_i, d_j) = \sqrt{2}$, because the analyst should move them away from each other to obtain the farthest distance on the 2D spatialization. If the two samples have the same label, the analyst should move them as close as possible ($dist_L(d_i, d_j) = 0$) in the projection, because they belong to the same cluster. Therefore, the interaction simulator provides the calculated pairwise distances between the selected samples as the training projection for DeepSI models.

Evaluation simulator: After the interaction simulator trains the DeepSI model, the trained model predicts a new projection. The predicted projection reflects the similarity relationships between samples in the low-dimensional spatialization. We used a kNN (K-nearest-neighbour) classifier [15] as the evaluation simulator to measure the predicted projection [7, 11]. The kNN classifier uses the neighbor information on the projection to train and predict the data classes. The performance of the learned kNN classifier can directly reflect the quality of the projection [64]. Concretely, we used the leave-one-out cross-validation [42] and set k = 5 closest training examples to predict the unlabelled sample. We also explored other values for k, such as 3, 7, 9, 11, but these did not produce significant changes in the results. We could thus obtain the trained kNN classifier accuracy by comparing the predicted output with the ground truth.

Performance over iteration: A new accuracy from the kNN classifier was returned from the simulation pipeline in each iteration loop. These are accumulated into a plot of kNN classifier performance over the iterations of the simulated interaction loop. This learning curve shows how rapidly the DeepSI model learned during the interactive process.

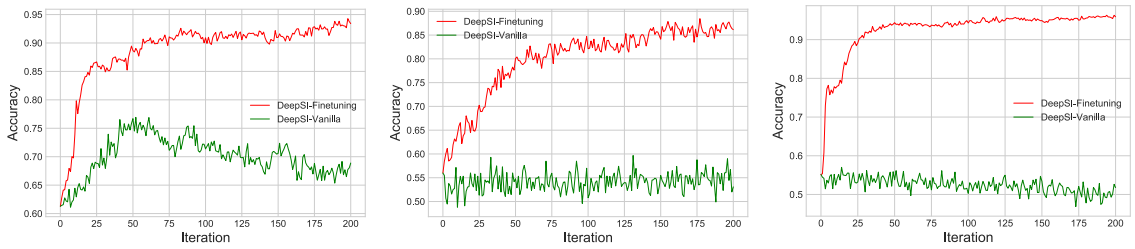
5.2.2 Dataset and Task. We explored three commonly used text corpora in natural language processing and visual text analysis tasks. These corpora contain different numbers of labels and are from different domains, providing a comprehensive evaluation of performance comparisons.

SST with two clusters: The SST dataset [59] is a collection of movie reviews with both fine-grained labels (out of five stars) and binary labels (positive and negative reviews). We used the binary version of the dataset, which contains 1821 reviews in total: 909 positive and 912 negative. The task, denoted as T_{sst}, used the SST dataset to train the DeepSI methods to obtain two clusters (positive and negative).

Vispubdata with three clusters: The Vispubdata dataset [31] contains academic papers published in the IEEE VIS conference series. These papers belong to one of the three conferences: InfoVis (Information Visualization), SciVis (Scientific Visualization), and VAST (Visual Analytics Science and Technology). We used the papers published between 2008 and 2018 (including 397 papers from InfoVis, 534 papers from SciVis, and 521 papers from VAST) in this task, denoted as T_{vis}. In T_{vis}, the simulated analyst need to iteratively drag papers into these three conferences clusters to evaluate the DeepSI.

20 Newsgroups with four clusters: The 20 Newsgroup dataset [3] is a collection of newsgroup posts on 20 topics. Based on this dataset, we create the task (T_{news}) to classify four topics into different clusters in the spatialization. We picked four topics from the same sub-category 'rec' including: 594 reports from 'rec.autos', 598 reports from 'rec.motorcycles', 597 reports from 'rec.sport.baseball', and 600 reports from 'res.sport.hockey'.

[3] http://qwone.com/ jason/20Newsgroups/



(a) T_{sst} (b) T_{vis} (c) T_{news}

Fig. 8. The accuracies of both $\text{DeepSI}_{\text{finetune}}$ and $\text{DeepSI}_{\text{vanilla}}$ updated projections over 200 iterations across the three tasks (T_{sst}, T_{vis}, and T_{news}) during the simulation-based experiment.

5.2.3 Quantitative Results. Fig. 8 shows the learning curves of both DeepSI methods in all three tasks. There is no crossing of the curves between these two models, and the performance curve of $\text{DeepSI}_{\text{finetune}}$ is above the curve of $\text{DeepSI}_{\text{vanilla}}$ through all iterations. This means $\text{DeepSI}_{\text{finetune}}$ showed better performance on all three tasks than $\text{DeepSI}_{\text{vanilla}}$. For model accuracy, $\text{DeepSI}_{\text{finetune}}$ converged to more than or nearly 90% accuracy in all three tasks. On the contrary, the best performance of $\text{DeepSI}_{\text{vanilla}}$ only showed slightly higher accuracy (less than 80%) than the initial performance in the first task (T_{sst}) with two clusters. For tasks with more than two clusters (T_{vis} in Fig. 8(b) and T_{news} in Fig. 8(c)), $\text{DeepSI}_{\text{vanilla}}$ did not show noticeable accuracy increases. This is consistent with our findings in the case study (Sec. 5.1). For model efficiency, the performance over iterations of $\text{DeepSI}_{\text{finetune}}$ in all three tasks showed steeper increases and quickly approximated peak accuracy. Furthermore, the performance of $\text{DeepSI}_{\text{finetune}}$ increased fairly consistently compared to $\text{DeepSI}_{\text{vanilla}}$. This provides analysts with more consistent feedback over iterations.

6 DISCUSSION

6.1 Generality and Applicability

In this paper, we use pretrained BERT as the specific DL model in $\text{DeepSI}_{\text{finetune}}$ to advance SI-enabled applications. Considering $\text{DeepSI}_{\text{finetune}}$ as a framework, it is general enough to apply other pretrained DL models into the semantic interaction pipeline for other VA tasks. First, other transformer-based models, such as RoBERTa [38], XLNet [69] and GPT-3 [12], can be applied directly in the DeepSI pipeline without any special configuration. In addition, fine-tunable DL models with other structures, such as CNN and RNN models [1], can also be integrated into a DeepSI pipeline by appending a special pooling layer to transform hidden states into proper representations. Further, other feature-based DL models could also be applied to the inteactive fine-tune process with specific designs. For example, ELMo [47] could be fine-tuned by using max-pool over the model's internal states and adding a softmax layer [48].

6.2 Scalability

Beyond measuring accuracy and efficiency, our two experiments also illuminated scalability. All our experiments were conducted on a desktop computer with an Intel i9-9900k processor, 32G Ram, and one NVIDIA GeForce RTX 2080Ti GPU, running Windows 10. In the case study, $\text{DeepSI}_{\text{finetune}}$ captured the analyst's intent and provided an accurate projection with 62 data points in real time. In the simulation-based experiment, $\text{DeepSI}_{\text{finetune}}$ also provided accurate projections that contains thousands of data points. During the simulation, MDS projection calculation ($O(n^2)$ algorithm)

consumed the majority of the time. The amount of time required for the DL model update and prediction was negligible in comparison. This highlights the potential for improving the DR method in DeepSI.

6.3 Interactive Deep Metric Learning

Dis-Function [11] describes the WMDS-based SI model as an interactive distance function learning model from the interactive machine learning perspective. Likewise, $\text{DeepSI}_{\text{finetune}}$ can be regarded as an interactive deep metric learning model [32]. As shown in Fig. 4, in the model-updating direction, the underlying pretrained BERT model is trained interactively to output better presentations that can capture the analyst-desired distance relationships. Deep metric learning methods usually use metric loss functions [32] for labelled data, such as contrastive loss [26], triplet loss [28] and angular loss [63]. In contrast, $\text{DeepSI}_{\text{finetune}}$ uses a metric loss function specially designed for semantic interactions based on MDS^{-1}.

6.4 Limitations and Future Work

Our $\text{DeepSI}_{\text{finetune}}$ proved effective in capturing analysts' precise intents and displaying intuitive projections. However, current $\text{DeepSI}_{\text{finetune}}$ prototypes could be extended in two directions. First, it is important to make the internal status of the underlying model interpretable to analysts in order to facilitate them making hypotheses and decisions during the sensemaking process, which is known as interpretable machine learning [43]. Other than the projection scatterplot, the current $\text{DeepSI}_{\text{finetune}}$ prototype does not provide any other visual hints about the status of the DL model. Therefore, we plan to add more specific visual designs in future work to better expose the effects of tuning the DL model.

In addition, $\text{DeepSI}_{\text{finetune}}$ uses the traditional metric learning method [4] as the interactive DR component to communicate between the DL model and the analyst. As discussed above (Sec. 6.3), a MDS-based loss function Loss_{SI} is used to interactively tune the DL model. Inversely, these deep metric learning loss functions, such as contrastive loss and triplet loss, could also be used as interactive DR components. We plan in future work to use deep metric learning loss functions as the interactive DR component in $\text{DeepSI}_{\text{finetune}}$.

7 CONCLUSION

In this work, we focused on DeepSI and the research question of how to integrate the DL model into the SI pipeline to leverage its capability to better capture the semantics behind user interactions. We identified two design requirements of effective DeepSI systems: the DL model is trained interactively in the SI pipeline and the DL model can be tuned properly with a small number of interactions. We presented $\text{DeepSI}_{\text{finetune}}$, which incorporates DL fine-tuning and MDS-based interactive DR methods into the DeepSI pipeline to meet these requirements. We performed two complementary experiments to measure the effectiveness of $\text{DeepSI}_{\text{finetune}}$, including a case study of a real-world task relating to COVID-19 and a simulation-based quantitative evaluation method on three commonly used text corpora. The results of these two experiments demonstrated that $\text{DeepSI}_{\text{finetune}}$ improves performance over the state-of-the-art alternative that uses DL only as a pre-processed feature extractor, indicating the importance of integrating the DL into the interactive loop.With a small number of semantic interactions as input, $\text{DeepSI}_{\text{finetune}}$ better captures the semantic intent of the analyst behind these interactions.

ACKNOWLEDGMENTS

This work was supported in part by NSF I/UCRC CNS-1822080 via the NSF Center for Space, High-performance, and Resilient Computing (SHREC).

REFERENCES

[1] Oludare Isaac Abiodun, Aman Jantan, Abiodun Esther Omolara, Kemi Victoria Dada, Nachaat AbdElatif Mohamed, and Humaira Arshad. 2018. State-of-the-art in artificial neural network applications: A survey. *Heliyon* 4, 11 (2018), e00938. https://doi.org/10.1016/j.heliyon.2018.e00938

[2] Saleema Amershi, Maya Cakmak, William Bradley Knox, and Todd Kulesza. 2014. Power to the people: The role of humans in interactive machine learning. *AI Magazine* 35, 4 (2014), 105–120.

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

[4] Aurélien Bellet, Amaury Habrard, and Marc Sebban. 2015. Metric Learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning* 9, 1 (2015), 1–151. https://doi.org/10.2200/S00626ED1V01Y201501AIM030 arXiv:https://doi.org/10.2200/S00626ED1V01Y201501AIM030

[5] Yoshua Bengio, Aaron Courville, and Pascal Vincent. 2013. Representation Learning: A Review and New Perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35, 8 (2013), 1798–1828. https://doi.org/10.1109/tpami.2013.50

[6] Y. Bengio, A. Courville, and P. Vincent. 2013. Representation Learning: A Review and New Perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35, 8 (Aug 2013), 1798–1828. https://doi.org/10.1109/TPAMI.2013.50

[7] Yali Bian, Michelle Dowling, and Chris North. 2019. Evaluating Semantic Interaction on Word Embeddings via Simulation. *EValuation of Interactive VisuAl Machine Learning systems, an IEEE VIS 2019 Workshop.* (2019).

[8] Yali Bian, John Wenskovitch, and Chris North. 2019. DeepVA: Bridging Cognition and Computation through Semantic Interaction and Deep Learning. *Proceedings of the IEEE VIS Workshop MLUI 2019: Machine Learning from User Interactions for Visualization and Analytics.* (2019).

[9] Nadia Boukhelifa, Anastasia Bezerianos, and Evelyne Lutton. 2018. Evaluation of interactive machine learning systems. In *Human and Machine Learning*. Springer, 341–360.

[10] Lauren Bradel, Chris North, Leanna House, and Scotland Leman. [n.d.]. Multi-model semantic interaction for text analytics. In *2014 IEEE Conference on Visual Analytics Science and Technology (VAST)*. IEEE, 163–172.

[11] Eli T Brown, Jingjing Liu, Carla E Brodley, and Remco Chang. 2012. Dis-function: Learning distance functions interactively. *2012 IEEE Conference on Visual Analytics Science and Technology (VAST)* (2012), 83–92. https://doi.org/10.1109/VAST.2012.6400486

[12] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models are Few-Shot Learners. arXiv:2005.14165 [cs.CL]

[13] Daniel Cer, Yinfei Yang, Sheng yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Yun-Hsuan Sung, Brian Strope, and Ray Kurzweil. 2018. Universal Sentence Encoder. arXiv:1803.11175 [cs.CL]

[14] H. Cheng, A. Cardone, S. Jain, E. Krokos, K. Narayan, S. Subramaniam, and A. Varshney. 2019. Deep-Learning-Assisted Volume Visualization. *IEEE Transactions on Visualization and Computer Graphics* 25, 2 (Feb 2019), 1378–1391. https://doi.org/10.1109/TVCG.2018.2796085

[15] T. Cover and P. Hart. 2006. Nearest Neighbor Pattern Classification. *IEEE Trans. Inf. Theor.* 13, 1 (Sept. 2006), 21–27. https://doi.org/10.1109/TIT.1967.1053964

[16] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[17] M. Dowling, J. Wenskovitch, J. T. Fry, S. Leman, L. House, and C. North. 2019. SIRIUS: Dual, Symmetric, Interactive Dimension Reductions. *IEEE Transactions on Visualization and Computer Graphics* 25, 1 (2019), 172–182.

[18] Michelle Dowling, John Wenskovitch, Peter Hauck, Adam Binford, Nicholas Polys, and Chris North. 2018. A bidirectional pipeline for semantic interaction. In *Proc. Workshop on Machine Learning from User Interaction for Visualization and Analytics (at IEEE VIS 2018)*, Vol. 11.

[19] Michelle Dowling, Nathan Wycoff, Brian Mayer, John Wenskovitch, Scotland Leman, Leanna House, Nicholas Polys, Chris North, and Peter Hauck. 2019. Interactive Visual Analytics for Sensemaking with Big Text. *Big Data Research* 16 (2019), 49–58. https://doi.org/10.1016/j.bdr.2019.04.003

[20] A. Endert, R. Chang, C. North, and M. Zhou. 2015. Semantic Interaction: Coupling Cognition and Computation through Usable Interactive Analytics. *IEEE Computer Graphics and Applications* 35, 4 (July 2015), 94–99. https://doi.org/10.1109/MCG.2015.91

[21] Alex Endert, Patrick Fiaux, and Chris North. 2012. Semantic interaction for visual text analytics. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, 473–482.

[22] Mateus Espadoto, Nina Sumiko Tomita Hirata, and Alexandru C Telea. 2020. Deep learning multidimensional projections. *Information Visualization* (2020), 1473871620909485.

[23] Jerry Alan Fails and Dan R Olsen Jr. 2003. Interactive machine learning. In *Proceedings of the 8th international conference on Intelligent user interfaces*. 39–45.

[24] Sebastian Gehrmann, Hendrik Strobelt, Robert Krüger, Hanspeter Pfister, and Alexander M. Rush. 2020. Visual Interaction with Deep Learning Models through Collaborative Semantic Inference. *IEEE Transactions on Visualization and Computer Graphics* 26, 1 (2020), 884–894. https://doi.org/10.1109/tvcg.2019.2934595

[25] Tera Marie Green, William Ribarsky, and Brian Fisher. 2009. Building and applying a human cognition model for visual analytics. *Information visualization* 8, 1 (2009), 1–13.

[26] R. Hadsell, S. Chopra, and Y. LeCun. 2006. Dimensionality Reduction by Learning an Invariant Mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, Vol. 2. 1735–1742.

[27] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

[28] Elad Hoffer and Nir Ailon. 2015. Deep Metric Learning Using Triplet Network. In *Similarity-Based Pattern Recognition*, Aasa Feragen, Marcello Pelillo, and Marco Loog (Eds.). Springer International Publishing, Cham, 84–92.

[29] Leanna House, Scotland Leman, and Chao Han. 2015. Bayesian visual analytics: BaVA. *Statistical Analysis and Data Mining: The ASA Data Science Journal* 8, 1 (Jan. 2015), 1–13.

[30] X. Hu, L. Bradel, D. Maiti, L. House, C. North, and S. Leman. 2013. Semantics of Directly Manipulating Spatializations. *IEEE Transactions on Visualization and Computer Graphics* 19, 12 (2013), 2052–2059.

[31] Petra Isenberg, Florian Heimerl, Steffen Koch, Tobias Isenberg, Panpan Xu, Chad Stolper, Michael Sedlmair, Jian Chen, Torsten Möller, and John Stasko. 2017. vispubdata.org: A Metadata Collection about IEEE Visualization (VIS) Publications. *IEEE Transactions on Visualization and Computer Graphics* 23, 9 (Sept. 2017), 2199–2206. https://doi.org/10.1109/TVCG.2016.2615308

[32] Mahmut Kaya and Hasan Şakir Bilge. 2019. Deep Metric Learning: A Survey. *Symmetry* 11, 9 (2019), 1066. https://doi.org/10.3390/sym11091066

[33] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[34] Bum Chul Kwon, Min-Je Choi, Joanne Taery Kim, Edward Choi, Young Bin Kim, Soonwook Kwon, Jimeng Sun, and Jaegul Choo. 2018. RetainVis: Visual Analytics with Interpretable and Interactive Recurrent Neural Networks on Electronic Medical Records. *IEEE Transactions on Visualization and Computer Graphics* 25, 1 (2018), 299–309. https://doi.org/10.1109/tvcg.2018.2865027 arXiv:1805.10724

[35] B. C. Kwon, H. Kim, E. Wall, J. Choo, H. Park, and A. Endert. 2017. AxiSketcher: Interactive Nonlinear Axis Mapping of Visualizations through User Drawings. *IEEE Transactions on Visualization and Computer Graphics* 23, 1 (Jan 2017), 221–230. https://doi.org/10.1109/TVCG.2016.2598446

[36] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *nature* 521, 7553 (2015), 436.

[37] Scotland C Leman, Leanna House, Dipayan Maiti, Alex Endert, and Chris North. 2013. Visual to Parametric Interaction (V2PI). *PLOS ONE* 8, 3 (March 2013), e50474.

[38] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. arXiv:1907.11692 [cs.CL]

[39] Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research* 9, Nov (2008), 2579–2605.

[40] Alberto González Martínez, Billy Troy Wooton, Nurit Kirshenbaum, Dylan Kobayashi, and Jason Leigh. 2020. Exploring Collections of research publications with Human Steerable AI. (2020), 339–348. https://doi.org/10.1145/3311790.3396646

[41] Leland McInnes, John Healy, and James Melville. 2018. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426* (2018).

[42] Matthew Mullin and Rahul Sukthankar. 2000. Complete Cross-Validation for Nearest Neighbor Classifiers. In *Proceedings of the Seventeenth International Conference on Machine Learning (ICML '00)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 639–646.

[43] W James Murdoch, Chandan Singh, Karl Kumbier, Reza Abbasi-Asl, and Bin Yu. 2019. Interpretable machine learning: definitions, methods, and applications. *arXiv preprint arXiv:1901.04592* (2019).

[44] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (Eds.). Curran Associates, Inc., 8024–8035. http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

[45] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12 (2011), 2825–2830.

[46] Claudia Perlich, Foster Provost, and Jeffrey S. Simonoff. 2003. Tree Induction vs. Logistic Regression: A Learning-Curve Analysis. *J. Mach. Learn. Res.* 4, null (Dec. 2003), 211–255. https://doi.org/10.1162/153244304322972694

[47] Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365* (2018).

[48] Matthew E. Peters, Sebastian Ruder, and Noah A. Smith. 2019. To Tune or Not to Tune? Adapting Pretrained Representations to Diverse Tasks. *CoRR* abs/1903.05987 (2019). arXiv:1903.05987 http://arxiv.org/abs/1903.05987

[49] Peter Pirolli and Stuart Card. 2005. The sensemaking process and leverage points for analyst technology as identified through cognitive task analysis. (2005), 2–4. https://analysis.mitre.org/proceedings/Final_Papers_Files/206_Camera_Ready_Paper.pdf

[50] Meg Pirrung, Nathan Hilliard, Artëm Yankov, Nancy O'Brien, Paul Weidert, Courtney D. Corley, and Nathan O. Hodas. 2018. Sharkzor: Interactive Deep Learning for Image Triage, Sort and Summary. *CoRR* abs/1802.05316 (2018). arXiv:1802.05316 http://arxiv.org/abs/1802.05316

[51] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics. http://arxiv.org/abs/1908.10084

[52] Sebastian Ruder. 2016. An overview of gradient descent optimization algorithms. arXiv:1609.04747 [cs.LG]

[53] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. 1986. Learning representations by back-propagating errors. *nature* 323, 6088 (1986), 533–536.

[54] Dominik Sacha, Leishi Zhang, Michael Sedlmair, John A Lee, Jaakko Peltonen, Daniel Weiskopf, Stephen C North, and Daniel A Keim. 2016. Visual interaction with dimensionality reduction: A structured literature analysis. *IEEE transactions on visualization and computer graphics* 23, 1 (2016), 241–250.

[55] Susan S Schiffman, M Lance Reynolds, and Forrest W Young. 1981. *Introduction to multidimensional scaling: Theory, methods, and applications*. Emerald Group Publishing.

[56] Jessica Zeitz Self, Michelle Dowling, John Wenskovitch, Ian Crandell, Ming Wang, Leanna House, Scotland Leman, and Chris North. 2018. Observation-Level and Parametric Interaction for High-Dimensional Data Analysis. *ACM Trans. Interact. Intell. Syst.* 8, 2, Article 15 (June 2018), 36 pages. https://doi.org/10.1145/3158230

[57] Jessica Zeitz Self, Radha Krishnan Vinayagam, JT Fry, and Chris North. 2016. Bridging the gap between user intention and model parameters for human-in-the-loop data analytics. In *Proceedings of the Workshop on Human-In-the-Loop Data Analytics*. ACM, 3.

[58] Jessica Zeitz Self, Radha Krishnan Vinayagam, J. T. Fry, and Chris North. 2016. Bridging the Gap Between User Intention and Model Parameters for Human-in-the-loop Data Analytics. In *Proceedings of the Workshop on Human-In-the-Loop Data Analytics* (San Francisco, California) *(HILDA '16)*. ACM, New York, NY, USA, Article 3, 6 pages. https://doi.org/10.1145/2939502.2939505

[59] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*. 1631–1642.

[60] Yves Tillé. 2006. *Sampling algorithms*. Springer.

[61] Julio Torales, Marcelo O'Higgins, João Mauricio Castaldelli-Maia, and Antonio Ventriglio. 2020. The outbreak of COVID-19 coronavirus and its impact on global mental health. *International Journal of Social Psychiatry* 66, 4 (2020), 317–320. https://doi.org/10.1177/0020764020915212 arXiv:https://doi.org/10.1177/0020764020915212 PMID: 32233719.

[62] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. arXiv:1706.03762 [cs.CL]

[63] Jian Wang, Feng Zhou, Shilei Wen, Xiao Liu, and Yuanqing Lin. 2017. Deep Metric Learning With Angular Loss. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

[64] Kilian Q Weinberger and Lawrence K Saul. 2009. Distance metric learning for large margin nearest neighbor classification. *Journal of Machine Learning Research* 10, 2 (2009).

[65] John Wenskovitch, Michelle Dowling, and Chris North. 2020. With Respect to What? Simultaneous Interaction with Dimension Reduction and Clustering Projections. In *Proceedings of the 25th International Conference on Intelligent User Interfaces* (Cagliari, Italy) *(IUI '20)*. Association for Computing Machinery, New York, NY, USA, 177–188. https://doi.org/10.1145/3377325.3377516

[66] John Wenskovitch and Chris North. 2017. Observation-level interaction with clustering and dimension reduction algorithms. In *Proceedings of the 2nd Workshop on Human-In-the-Loop Data Analytics*. 1–6.

[67] Svante Wold, Kim Esbensen, and Paul Geladi. 1987. Principal component analysis. *Chemometrics and Intelligent Laboratory Systems* 2, 1 (1987), 37 – 52. https://doi.org/10.1016/0169-7439(87)80084-9 Proceedings of the Multivariate Statistical Workshop for Geologists and Geochemists.

[68] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2019. HuggingFace's Transformers: State-of-the-art Natural Language Processing. *ArXiv* abs/1910.03771 (2019).

[69] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2020. XLNet: Generalized Autoregressive Pretraining for Language Understanding. arXiv:1906.08237 [cs.CL]

[70] Aston Zhang, Zachary C. Lipton, Mu Li, and Alexander J. Smola. 2020. *Dive into Deep Learning*. https://d2l.ai.

[71] X. Zhu and A. Goldberg. 2009. *Introduction to Semi-Supervised Learning*. Morgan & Claypool. https://ieeexplore.ieee.org/document/6813505