Cognitive Systems Research 83 (2024) 101184

Contents lists available at ScienceDirect

Cognitive Systems Research

journal homepage: www.elsevier.com/locate/cogsys

Transformability, generalizability, but limited diffusibility: Comparing global vs. task-specific language representations in deep neural networks

Yanru Jiang [a,*], Rick Dale [a], Hongjing Lu [b]

[a] Department of Communication, University of California Los Angeles, USA
[b] Department of Psychology, University of California Los Angeles, USA

ARTICLE INFO

Keywords:
Transformer
RNN
Autoencoder
Hybrid cognitive model
Deep learning

ABSTRACT

This study investigates the integration of two prominent neural network representations into a hybrid cognitive model for solving a natural language task, where pre-trained large-language models serve as global learners and recurrent neural networks offer more "local" task-specific representations in the neural network. To explore the fusion of these two types of representations, we employ an autoencoder to transform them between each other or fuse them into a single model. Our exploration identifies a computational constraint, which we term *limited diffusibility*, highlighting the limitations of hybrid systems that operate with distinct types of representation. The findings from our hybrid system confirm the crucial role of global knowledge in adapting to a new learning task, as having only local knowledge greatly reduces the system's transferability.

1. Introduction

The goal of the present study is to investigate how two prominent neural network representations can be integrated in one unitary framework to solve a particular natural language task. The overall model fuses two types of learning. Prominent pre-trained large-language models can be thought of as *global* learners, extracting large-scale latent structure from massive input. Recurrent neural networks (RNNs), on the other hand, are often deployed on particular training data, to carry out very specific tasks. This yields more "local" *task-specific* representations in the neural network. Integrating these two distinct representations, global vs. task-specific, is likely important both for natural cognition and applied artificial intelligence. To explore how this integration could take place, we use an autoencoder to test how the two types of representations can be fused into a single model or transformed between each other. This allows us to investigate the computational benefits and limitations of hybrid systems of this sort. One outcome of our exploration is identifying a kind of computational constraint which we term *limited diffusibility*, which highlights the limitations of hybrid systems that transact in very distinct kinds of representation. We end with implications both for modeling and relevance to theories of human cognition.

2. Background

Numerous advances in artificial intelligence (AI) in the past decade have been focused on specific tasks, such as natural language processing (NLP), computer vision, robotics, and autonomous driving (Dutta, 2018). In contrast, the field has recently devised AI systems that can perform more general tasks, incorporate world knowledge, and exhibit human-like reasoning (Haenlein & Kaplan, 2019). For example, transformers are a sophisticated deep learning architecture and a significant breakthrough in developing generalizable AI applications, especially for language processing tasks (Vaswani et al., 2017). One way that transformers integrate world knowledge in its pre-trained model is by utilizing external knowledge bases, such as Wikipedia or other structured data sources, and incorporating these sources into their representations (Devlin et al., 2019).

It could therefore be said that these are two broad and distinct approaches to AI, namely task-specific and general AI. These approaches can be framed in terms of the internal representations these AI systems are learning, such as in the embedding vectors that can numerically represent meaning in NLP tasks. On the one hand, AI systems can have *task-specific representations* that optimize the performance of a particular situation or goal; but they can also have more *global representations* that encode general knowledge from a massive pre-trained text corpus.

Though an AI system can be designed to focus on one or another type

* Corresponding author at: Dept. of Communication, Rolfe Hall, Los Angeles, CA 90095, USA.
E-mail address: yanrujiang@g.ucla.edu (Y. Jiang).

https://doi.org/10.1016/j.cogsys.2023.101184
Received 30 April 2023; Received in revised form 3 October 2023; Accepted 5 November 2023
Available online 7 November 2023

of learned representation, it is unlikely that the human mind would employ just one of these types alone. Indeed, there is evidence across various domains that memory, attention and other capacities of human cognition utilize integrated representations (and corresponding processes) of distinct kinds. These representations can be quite distinct in their function. For example, in the so-called "complementary learning systems" framework of O'Reilly and others, the cerebral cortex and subcortical limbic system are taken to subserve two distinct but complementary subprocesses (O'Reilly & Norman, 2002). The cortical system may more slowly encode and preserve broad associative knowledge about the world; whereas subcortical hippocampal processes may deploy that knowledge, transforming it in particular situations or tasks. So human cognition, and perhaps sophisticated AI, could make use of multiple complementary representations or processes. Across various domains, there is strong evidence that distinctive representational formats or processes occupy many aspects of cognition such as short-term and working memory (Baddeley et al., 2019), attention and learning (Conway, 2020), long-term memory (Tulving, 1985; Squire, 2004) and more.[1]

The goal of the present study is not to investigate any one of these particular proposals. Instead, we use relatively recent language models to examine how distinctive representational formats can be integrated into a single model. Though our work has goals similar to prominent hybrid cognitive models (Nason & Laird, 2005; Sun et al., 2001; Jilk et al., 2008), our aim is to show how both task-specific and global representations can participate together in a particular cognitive task. The advent of these large-scale models permits new examination of how computational systems, construed generally, can comprise distinctive representational formats yet generate singular coherent responses under particular tasks. The present work thus informs continuing theorization about multiple systems in memory and learning by sketching out some computational implications and limitations of multiple representational types.

2.1. Applying the typology to current neural networks for NLP

Under the typology above, RNN and Bidirectional Encoder Representations from Transformers (BERT) can be viewed as two classic language model architectures to compare the characteristics of the two representational systems. An RNN is typically designed to process sequential data by using feedback loops to pass information from one time step to the next (Rumelhart et al., 1986; Jordan, 1986), making them well-suited for processing a sequence of input tokens from a sentence (Mikolov et al., 2010). Its capacity to process sequential dependencies conceptually aligns with how humans process and derive meaning from a sentence (Elman, 1990). However, the vanilla form of RNN suffers from the vanishing gradient problem and has difficulty capturing long-term dependencies, which are important for many NLP tasks. Long Short-Term Memory (LSTM) was then introduced as an extension to RNNs. LSTMs include an additional state variable, called the cell state, to control specific information that needs to be kept or updated while processing the whole sequence (Sherstinsky, 2018). Before the introduction of transformer-based models, RNNs were the most commonly used architecture for NLP. The embeddings extracted at the final hidden state of RNNs can represent the entire input sequence and are optimized for the downstream language task and thus can be considered task-specific representations.

On the other hand, BERT is a state-of-the-art NLP and transformer-based learning model pre-trained on a large text corpus, including Wikipedia pages and books (Devlin et al., 2019). Different from the sequential dependency in RNNs, BERT generates high-quality contextualized embeddings by using a self-attention mechanism that allows the model to capture the relationships between different words and phrases in a sentence and generate embeddings that reflect the context in which they appear. Being trained on a large corpus of text data, its pre-trained model "learns" world knowledge from general patterns and relationships in language (Rogers et al., 2020). In practice, BERT's pre-trained model can be further fine-tuned on specific NLP tasks by adjusting its final layers to generate task-specific embeddings and optimize the model performance (Devlin et al., 2019). The embeddings generated from the BERT-pre-trained model capture the sentence-level semantic and contextual information that is derived from BERT's inherent knowledge and thus can be considered as global representations.

Compared to transformer-based models, RNNs are considered to be more transparent and easier to interpret. Particularly, RNNs capture the emergent behavior of the input data without any external training corpus, meaning they can learn the underlying structure of the data sequentially on their own. Transformers, on the other hand, are typically composed of multiple self-attention layers and billions of parameters to capture the complex relationships between different parts of the input sequence (Baan et al., 2019). Additionally, transformers are often pre-trained on massive amounts of data, which can make it difficult to pinpoint and understand the connection between the training corpus and the resulting model predictions.

Given the substantive differences between these two computational models and representations in terms of architecture, transparency, emergent nature, and sparsity, this study aims to shed light on the distinct representational systems in deep learning by applying the RNN and BERT models to the same set of data—a collection of emotionally charged tweets—and contextualizing the findings based on the ordinal nature of emotion categorization (Yannakakis et al., 2021). As noted above, this contrast between task-specific and global representations serves as a context for investigating how two distinct representational schemes coordinate.

2.2. Transforming and diffusing representational systems

There are several choices available to simulate the interaction between two distinct representational subsystems. One common practice of hybrid models is to train different tasks under distinct systems (e.g., sequential, spatial, transformer, etc.), concatenate these different representations, and use a fully connected (FCN) layer to computationally integrate them for the downstream task (Lu et al., 2019).

Alternatively, an autoencoder has the ability to transform one representational system into another (Hinton & Zemel, 1994). An autoencoder is a neural network that contains three components: encoder, bottleneck, and decoder (Michelucci, 2022). The model learns to reconstruct the input data by compressing (encoder) it into a lower-dimensional embedding (bottleneck), then reconstructing it back into its original form or any target form (decoder). If the output is the same as the input, this process allows the network to learn a compressed, latent representation of the input data that captures the most salient features of the original data and is commonly used for dimensionality reduction and self-supervision. With the output being a different representational system from the input, the autoencoder could be used to transform the input system into the target system. For instance, a previous study used an autoencoder to bridge the human and robotic haptic representations in a common space (Edmonds et al., 2019).

Given the architectural advantages of an autoencoder, the present study uses it to explore the potential transformation (i.e., "system switching") and diffusion (i.e., "intermediate stage") between the two representational systems: global vs. task-specific representations. We

[1] Many readers will know that "representation" is sometimes a fraught theoretical term, especially as it pertains to the human cognitive system (e.g., Markman & Dietrich, 2000; Chemero, 2001). Our focus here is on numerical representations of concepts in embeddings, a familiar format in neural network models. We do not wish to imply the findings are easily generalizable to the human case, as it is outside the scope of the present paper. The models we explore here simply provide a platform to test distinct representational formats in a model system and how they might diffuse and transform their knowledge.

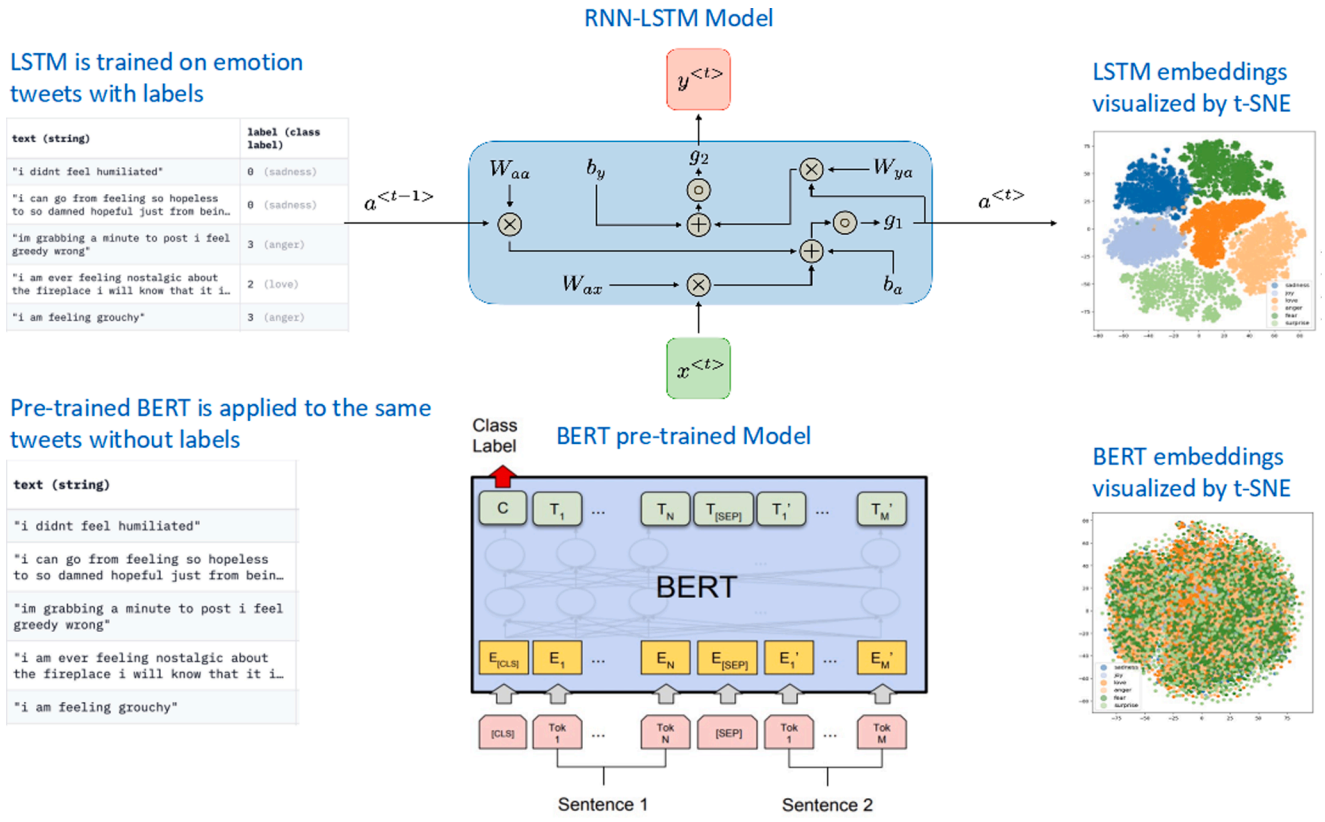

Fig. 1.1. Step 1 of the multi-step simulation: Generating Representational Systems. The same emotion tweets are used to generate LSTM and BERT embeddings, visualized with t-SNE.

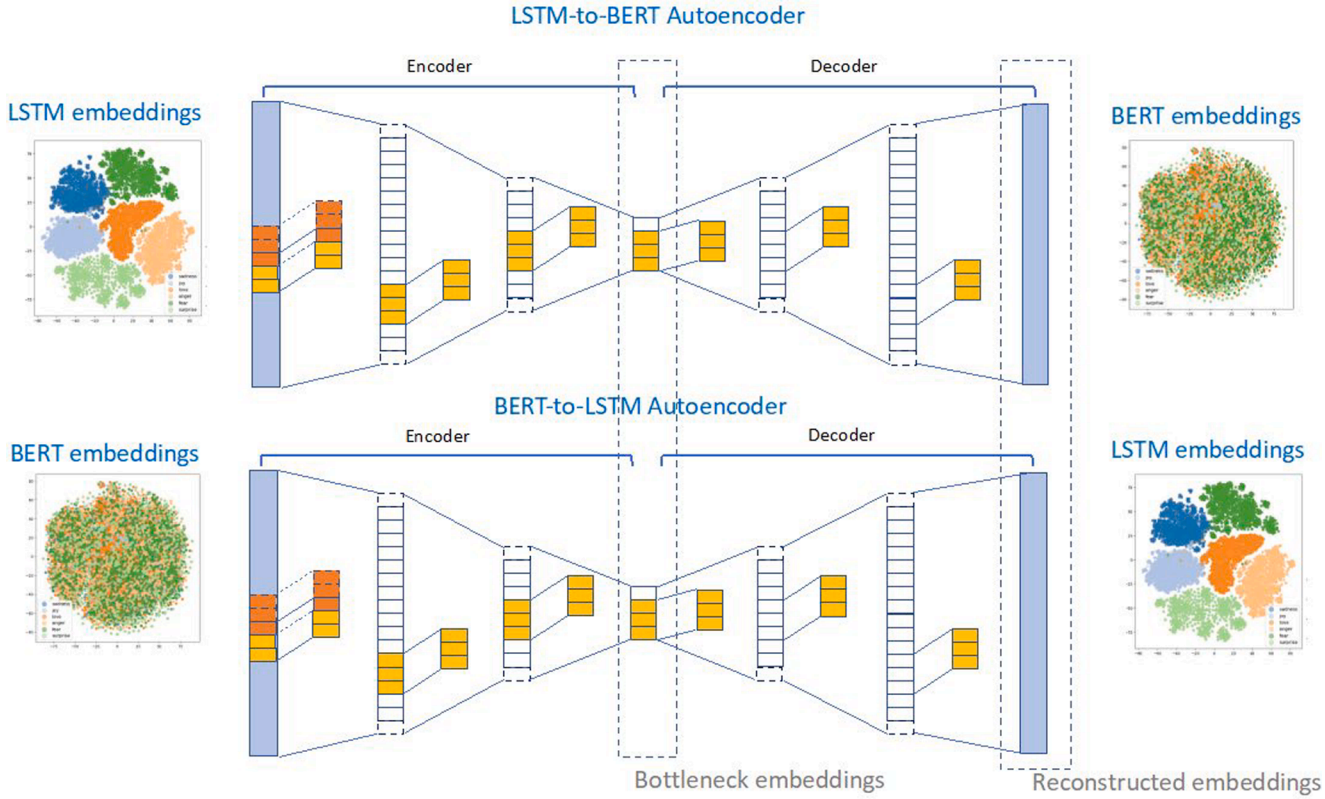

Fig. 1.2. Step 2 of the multi-step simulation: Transforming Representational Systems. The autoencoder LSTM-to-BERT transforms LSTM embeddings into BERT embeddings (upper model), and the autoencoder BERT-to-LSTM transforms BERT embeddings into LSTM embeddings (lower model).

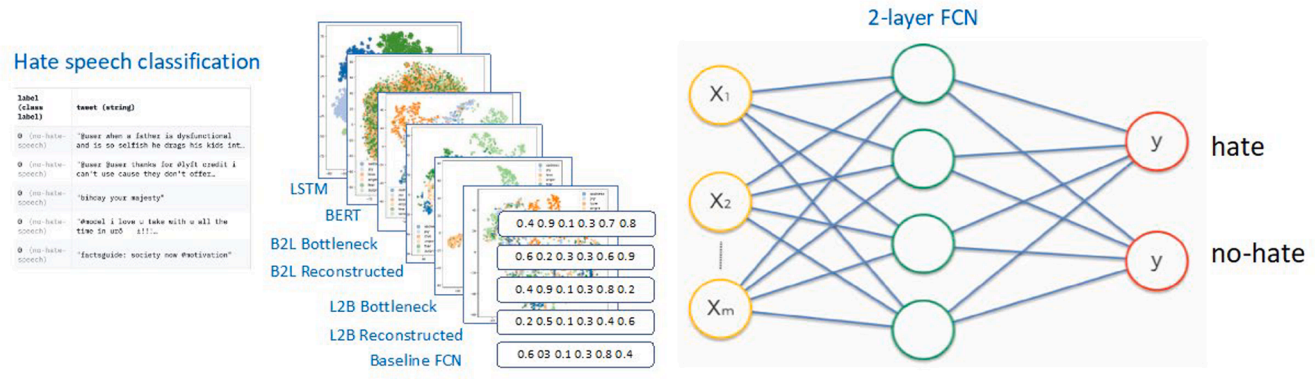

Fig. 1.3. Step 3 of the multi-step simulation: Generalizing to a new task. Hate speech tweets are used to generate seven representations, LSTM, BERT, BERT-to-LSTM (B2L) bottleneck and reconstructed, LSTM-to-BERT (L2B) bottleneck and reconstructed, and baseline fully connected network (FCN) embeddings. Each of these representations is fed into a 2-layer FCN for classifying hate and non-hate tweets.

investigate whether the intermediate stage of the transformation can carry characteristics of both global and task-specific information and whether these intermediate representations will be more transferable and generalizable to other tasks. Particularly, the bottleneck and reconstructed embeddings of the autoencoder can be viewed as diffused and transformed representations. This examination not only offers a theoretical contribution to cognitive computing but also sheds light on the possibility of diffusing two representational systems that exhibit different advantages and information into a common space and utilizing the diffused system to optimize the downstream task.

As noted above, we apply the RNN and BERT models to a set of emotionally charged tweets for emotion categorization. Once the LSTM (i.e., RNN-LSTM) and BERT representations of emotionally charged tweets have been generated, this study constructs two autoencoders, BERT-to-LSTM and LSTM-to-BERT, to examine the transformability and diffusibility between the systems. After obtaining the bottleneck and reconstructed embeddings from the two models, these new representational systems (transformed and diffused), in addition to the previous global and task-specific systems, will be applied to a different emotion-related task, hate-tweet classification, to examine which systems exhibit better transferability and generalizability.

3. Simulation

This study selected two deep learning models: the transformer model BERT (Devlin et al., 2019) for generating global representations, and RNN-LSTM (Hochreiter & Schmidhuber, 1997; *a.k.a* LSTM) for generating task-specific representations for the same tweets from a tweet-emotion classification task. Next, to examine the transformability and diffusion of these representations, BERT-to-LSTM and LSTM-to-BERT autoencoders were constructed. The study aimed to determine whether these representations could be transformed into each other through reconstruction and whether they could diffuse to the same embedding space using a similarity measurement. After generating the six representational systems (namely LSTM, BERT, BERT-to-LSTM bottleneck and reconstructed, and LSTM-to-BERT bottleneck and reconstructed), they are employed in a different emotion-related task, hate speech classification. This application assesses the generalizability and information retention capabilities of each system. Appendix I provides a table documenting all models, their corresponding inputs and outputs, along with formulas and descriptions.

Figs. 1.1–1.3 provide a holistic overview of this multistage simulation, including (1) the generation of representational systems, (2) the transformation between systems, and (3) generalization to a new task. This multi-step simulation has been executed end-to-end for 25 runs to illustrate the consistency and robustness of the results. For illustrative purposes, models with metrics falling within the range of ± 1 standard deviation were selected for the figures. All models and simulations were implemented using Pytorch on a NVIDIA GeForce RTX 3090 GPU under CUDA version 12.0.

3.1. Generating representational systems

3.1.1. Dataset for training deep learning networks

This study used the Emotion dataset from Hugging Face (Saravia et al., 2018) to generate both global and task-specific tweet representations for both the BERT and LSTM models. This dataset consists of 20,000 English Twitter messages with six basic emotions (e.g., anger, fear, joy, love, sadness, and surprise) by adopting Plutchik's wheel of emotions (Plutchik, 2001), Ekman's six basic emotions (Ekman, 1992), and hashtags in tweets. Tweets were annotated through noisy labels and distant supervision introduced by Go et al. (2009). We divided 20,000 tweets into 16,000 for training, 3,200 for validation, and 800 for test to train the LSTM model (LSTM embeddings will be generated during the training process). The same 16,000 tweets in the training set were used for generating BERT and LSTM embeddings (i.e. representations), and these embeddings were further split into training and test sets at a 9:1 ratio to train autoencoders.

3.1.2. Transformer model and global representation

The BERT model can take any input sentence and return a sentence embedding plus a set of word embeddings for every token in that sentence. When applying BERT to a sentence, it generates word and sentence embeddings based on context, and the sentence embedding can be extracted at the initial [CLS] token (which stands for "classification") from the output of the last layer of transformers. When BERT undergoes the pre-trained phase that is trained on the next sentence prediction objective, the model is given sentence pairs < current sentence, next sentence>, wherein the [CLS] token is used to separate the two sentences. As the [CLS] token is the first token in the input sequence, it captures information from both the left and right context, effectively encoding the context of the entire sentence, making it a powerful representation of the sentence as a whole. Due to this characteristic, the [CLS] token has been widely used to represent the entire input sequence in various downstream tasks such as sentiment analysis, text classification, and question answering (Chen et al., 2022; Koroteev, 2021).

This study leveraged the "bert-base-uncased" pre-trained model from Hugging Face (Wolf et al., 2020) to generate a contextualized summary representation, which takes into account the relationships between words and the overall meaning of the sentence, for each tweet sample. This uncased-based version of BERT requires fewer computational resources compared to larger BERT models. No fine-tuning was performed as this study relies on the pre-trained BERT *only* to produce a global representation of each tweet in a 786-dimensional language space. The

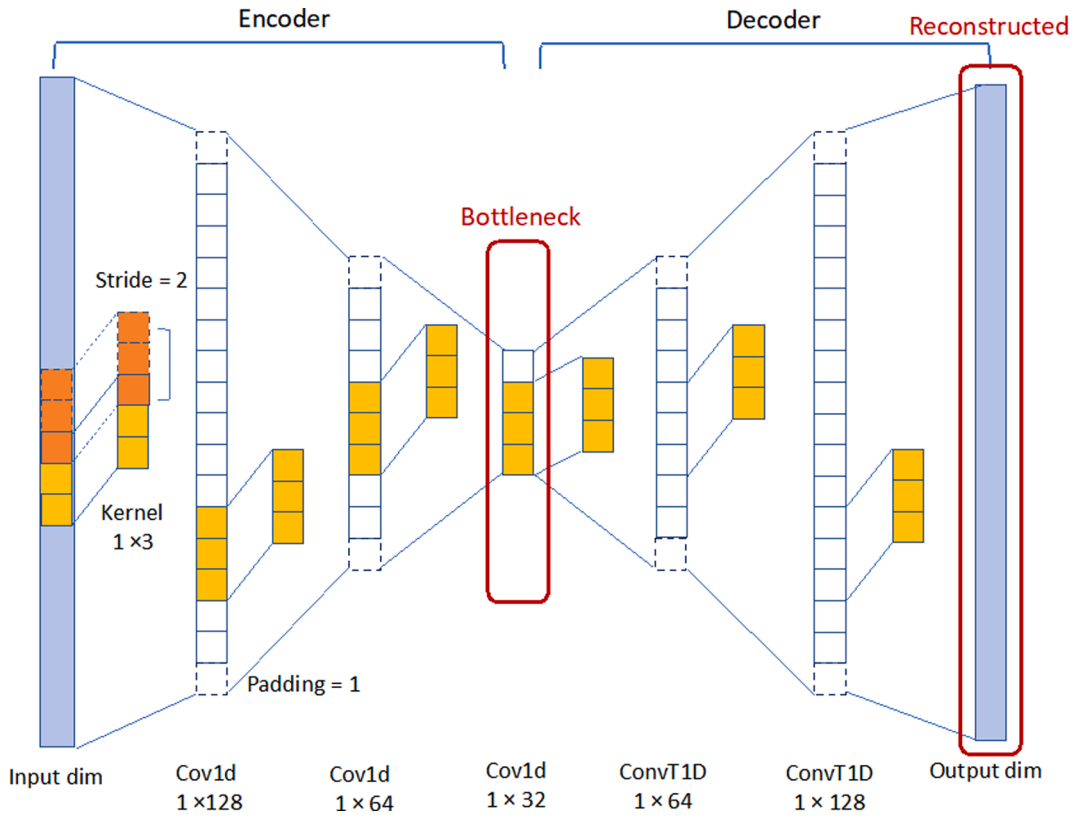

Fig. 2. Conv1D-Autoencoder architecture for BERT-to-LSTM and LSTM-to-BERT models, with stride of 2, padding of 1, and kernel size of 3. Hidden layers of bottleneck and reconstructed embeddings are highlighted in red.

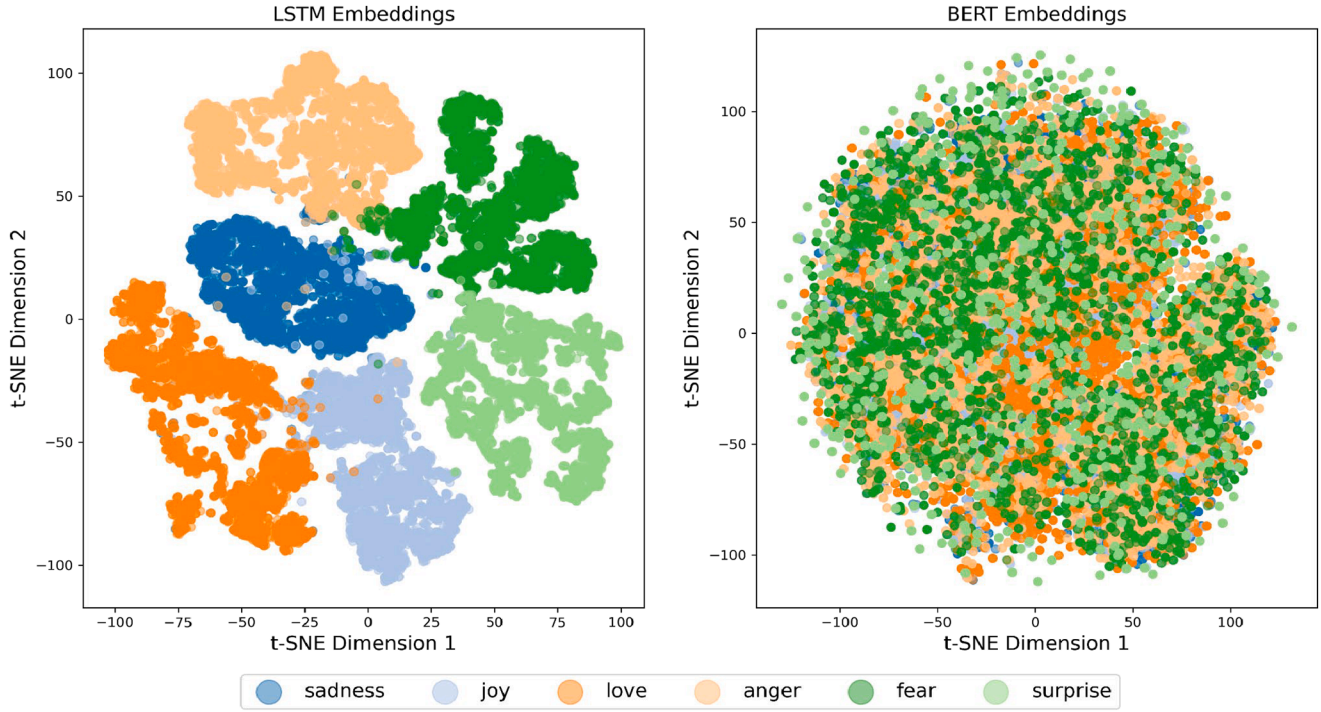

Fig. 3. t-SNE visualizations for LSTM (left) and BERT (right) embeddings.

default tokenizer of the bert-base-uncased model was used and the batch size was set to 32. No additional pre-processing was necessary as the tweets in the Emotion dataset had already been processed and did not contain any hashtags, emojis, mentions, or other extraneous symbols (Saravia et al., 2018).

3.1.3. *RNN model and task-specific representation*

Before the introduction of transformer-based models, RNN-LSTMs

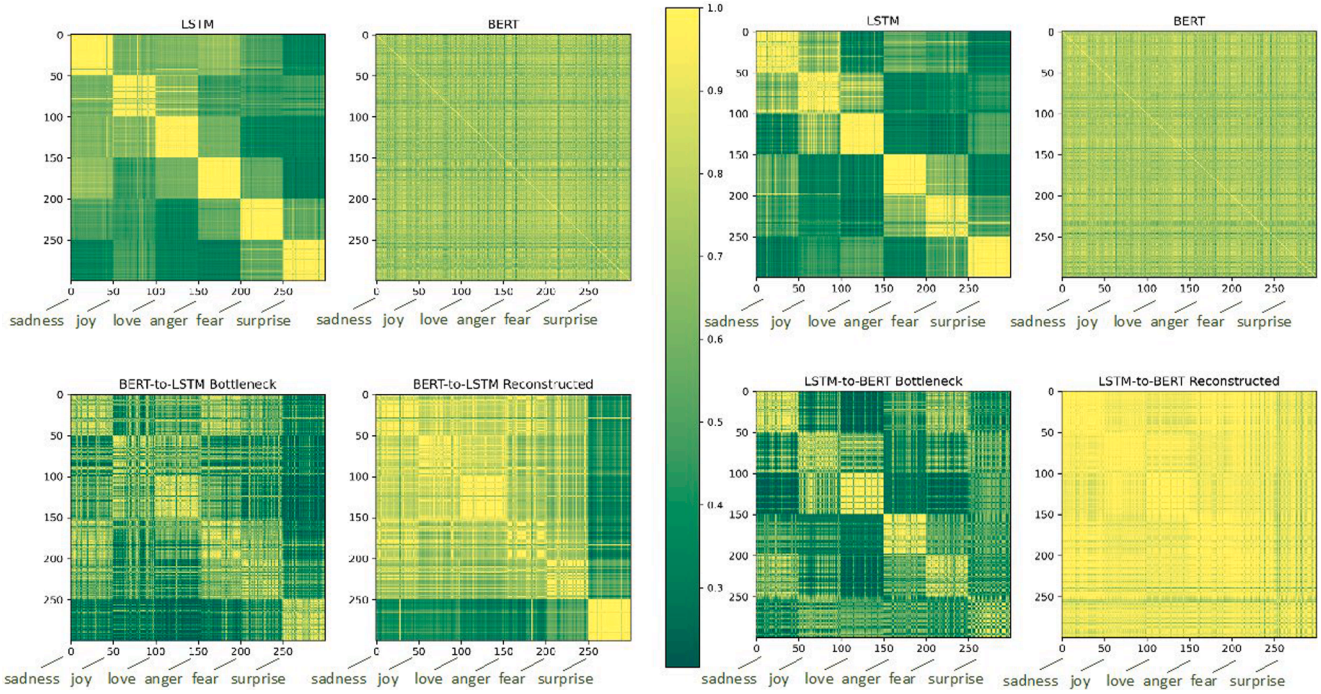

Fig. 4. Similarity matrices of LSTM, BERT, bottleneck, and reconstructed embeddings from the BERT-to-LSTM (left) and LSTM-to-BERT (right) models. Each matrix has a shape of 300 × 300, constructed from the pairwise cosine similarity of 300 sampled tweets' embeddings.

(or LSTMs) were the most commonly used architecture for language processing. To avoid contextualization in word embeddings, GloVe (Global Vectors for Word Representation), a word vectorization approach that does not rely on local statistics (local-context information of words) was used to vectorize each token in a tweet (Pennington et al., 2014). The "basic English" tokenizer from torchtext was used to tokenize each tweet. The GloVe embedding with 300-dim semantic features has been utilized because it strikes a balance between capturing sufficient information and maintaining computational efficiency. To optimize the computation in PyTorch for RNN, all tokenized tweets were padded to the same length of 66 tokens, which is the number of tokens that the longest tweet sample contains.

The construction of RNN-LSTMs in this study followed the common practice used in deep learning. A batch normalization was first applied to the data input, with the size batch_size × time_step × input_dimension (32 × 66 × 300). The normalization was conducted across the 66 time steps of each data point to reduce variance within a data unit. Then an RNN-LSTM was applied to the normalized input to convert temporal information to a dense embedding at each time step. The final time step (i.e., hidden state) of RNN was used as sentence-level embedding, which incorporates information from all previous time steps (i.e., tokens) and compresses the entire input to be fed to the fully connected network (FCN) layer. Finally, an FCN layer, without any activation, was applied to map the embeddings in the last time step to desired output classes for multiclassification prediction.

The training was conducted using the standard cross-entropy losses for multiclassification. The batch size was set to 32 and the model was trained for 10 epochs. An Adam optimizer was used to learn the network parameters, with default parameters and learning rate = 0.001.

In a classification task, the final time step is expected to contain information about the classification label through iterations of back-propagation. Therefore, once the model has been successfully trained with a satisfactory accuracy after 10 epochs, the final hidden state of RNN-LSTM from the last epoch was extracted and used as a sentence embedding to resemble a task-specific representation of each tweet.

3.2. Transforming representational systems

Once the LSTM and BERT embeddings were generated, this study employed an autoencoder architecture (see Fig. 1.2 and Fig. 2) for both the BERT-to-LSTM and LSTM-to-BERT models to explore the potential transformation between global and task-specific representations. The bottleneck embeddings of the two models are considered the intermediate diffusion stage of the two systems, which is substantially smaller than the input and output dimensions.

While an FCN autoencoder would have been a theoretically meaningful benchmark for testing the transformation of representations, this study opted for a 1D convolutional (Conv1D) autoencoder for its ability to handle data scarcity more effectively than an FCN autoencoder (Xie et al., 2023; Lee et al., 2022). The FCN autoencoder contains significantly more parameters due to its fully connected architecture, whereas the Conv1D layer can pool local information efficiently through its sliding kernel, thus substantially reducing the size of hidden layers at each step.

The representational dimensions for BERT and LSTM are 768 and 300, respectively. When implementing an autoencoder, to utilize the Conv1D design, each input data is reshaped from 2D to 3D (e.g., data_size × input_dim to data_size × input_dim × 1). Both models are composed of a three-layer Conv1D encoder to pool input embeddings in the order of input_dim × 128, 128 × 64, 64 × 32, followed by a three-layer ConvTranspose1D decoder to decompress the bottleneck embeddings in the order of 32 × 64, 64 × 128, 128 × output_dim. Kernel is set to 3, and stride = 2, with padding = 1 for both encoder and decoder.

Both autoencoders were trained using the sum of the mean standard errors of all output dimensions. The batch size was set to 32 and both models were trained with 50 epochs.

3.2.1. Representational similarity analysis

Representational similarity analysis (RSA) is a technique used to compare the similarity of representations in different models, human behaviors, or brain regions (Kriegeskorte et al., 2008). It is commonly used to compute the correlation between human dissimilarity judgments and model-derived distances (i.e., distances between embeddings) as

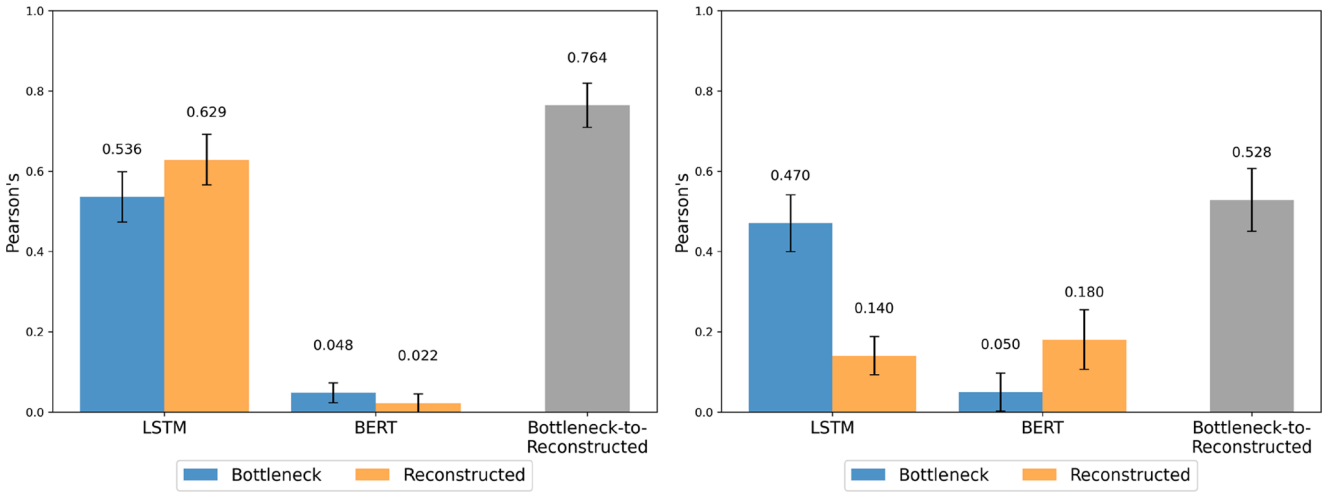

Fig. 5. Pearson correlation between LSTM, BERT, bottleneck, and reconstructed embeddings from the BERT-to-LSTM (left) and LSTM-to-BERT (right) models.

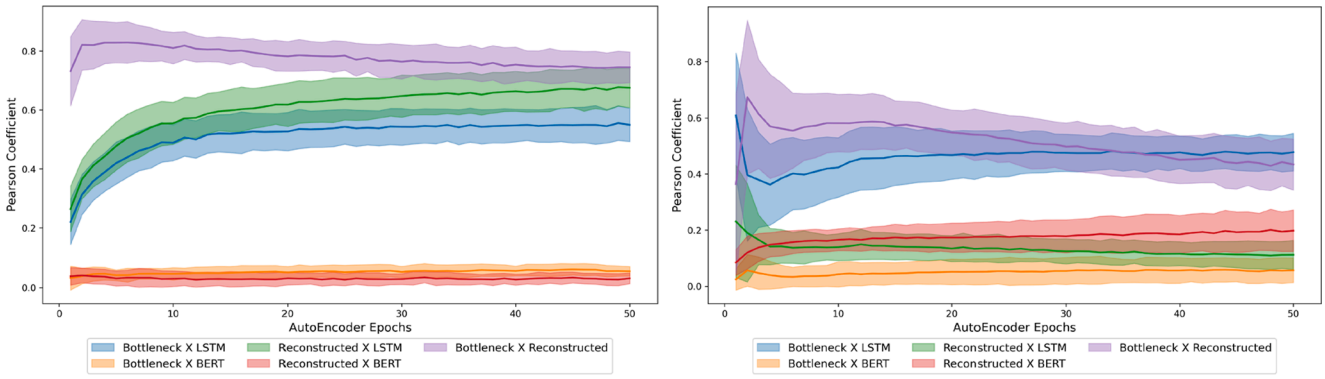

Fig. 6. Pearson correlation between LSTM, BERT, Bottleneck and Reconstructed embeddings from the BERT-to-LSTM (left) and LSTM-to-BERT (right) models for all epochs.

Table 1
Model performances for the baseline model and different representational systems.

Model	Representation	Train Loss	Test Loss	Test Acc
Baseline	–	0.684(0.003)	0.685(0.002)	0.574(0.069)
LSTM	Task-specific	0.618(0.028)***	0.640(0.025)***	0.631(0.034)***
BERT	**Global**	**0.077 (0.011)*****	**0.558 (0.056)*****	**0.818 (0.013)*****
BERT-to-LSTM	Diffused	0.596(0.018)***	0.621(0.019)***	0.657(0.024)***
	Transformed	0.648(0.012)***	0.655(0.014)***	0.615(0.025)**
LSTM-to-BERT	Diffused	0.654(0.014)***	0.665(0.014)***	0.591(0.029)
	Transformed	0.673(0.011)***	0.675(0.013)***	0.568(0.031)

*** $p < 0.001$ ** $p < 0.01$ * $p < 0.05$.

well as to "compare representations between stages of processing within a given brain or model, and between brain and behavioral data" (Nili et al., 2014, p.1).

Introduced by psychologists and neuroscientists, RSA has recently been applied to understanding the representational learning of deep learning models and visualizing the representational space, such as categorical information, of a layer in a DNN (Groen et al., 2018). RSA is more advantageous than the common high-dimensional visualization techniques as it provides a finer-grained quantitative measure of similarity that is not limited to visualizing embeddings in a 2D or 3D space. Additionally, these embedding visualization techniques use dynamic sampling that incorporates randomness into their visualizations, which can make it challenging to compare different runs or evaluate the stability of the results.

In this study, the RSA approach was used to calculate the correlation of distance matrices among embeddings from different models and/or layers, including the original global and task-specific representations (i.e., BERT and LSTM) and transformed and diffused representations from autoencoders (i.e., bottleneck and reconstructed). Therefore, this study asks how textual information is represented differently across four distinct schemes: global, task-specific, diffused, and transformed.

For each autoencoder model, first, a model that is saved at half of the total epoch was applied to the test data to generate bottleneck and reconstructed representations while avoiding overfitting. Then, we sampled 50 tweets for each emotion category and the representations of the same 300 (50 × 6) tweets were used to construct four similarity matrices based on pairwise cosine similarity between each sampled tweet. Each matrix has a shape of 300 × 300 (see Fig. 4). Next, pairwise Pearson correlation was calculated between four representations by flattening the lower triangular of each cosine matrix (see Fig. 5).

3.3. *Generalizability of global, task-specific, diffused, and transformed embeddings*

To investigate the transferability and generalizability of these four representational systems, the study treats them as input embeddings and

Table A1
Details on the input, output, and models used in the multi-step simulation.

Phases	Model Id	Model	Input	Output
(1) Generate Representational Systems	1.1	RNN-LSTM $x_{emotiontweet} = [tok_1, tok_2, \cdots, tok_n] y_{lstm} = LSTM(GloVE_{Embedding(x_{emotiontweet})})$	Sequence of GloVe embeddings of emotion tweets	LSTM embeddings
	1.2	Pre-trained BERT $x_{emotiontweet} = [tok_1, tok_2, \cdots, tok_n] y_{bert} = BERT_{LayerN}(\cdots(BERT_{Layer1}(x_{emotiontweet})))$	Emotion tweets tokenized by the default BERT tokenizer	BERT embeddings
(2) Transform Representational Systems	2.1	BERT-to-LSTM $y_{bert} = Conv1D - Autoencoder(y_{lstm})$	BERT embeddings	Reconstructed LSTM embeddings Bottleneck BERT-to-LSTM embeddings
	2.2	LSTM-to-BERT $y_{lstm} = Conv1D - Autoencoder(y_{bert})$	LSTM embeddings	Reconstructed BERT embeddings Bottleneck LSTM-to-BERT embeddings
(3) Generalize to a new task	3.1	Baseline 2-layer FCN $x_{hstweet} = [tok_1, tok_2, \cdots, tok_n] y_{hs} = FCN(nn.Embedding(x_{hstweet}))$	Pre-trained embeddings loaded from GloVe for hate speech (HS) tweets	Predictive labels for the hate speech classification task (0 and 1)
	3.2	LSTM + 2-layer FCN $y_{hs} = FCN(MODEL1.1(x_{hstweet}))$	HS embeddings generated by the trained LSTM	
	3.3	BERT + 2-layer FCN $y_{hs} = FCN(MODEL1.2(x_{hstweet}))$	HS embeddings generated by the pre-trained BERT	
	3.4	Bottleneck BERT-to-LSTM + 2-layer FCN $y_{hs} = FCN(MODEL2.1 - B2L_{bottleneck(x_{hstweet})})$	HS embeddings extracted from the bottleneck layer of Model 2.1	
	3.5	Reconstructed BERT-to-LSTM + 2-layer FCN $y_{hs} = FCN(MODEL2.1(x_{hstweet}))$	HS embeddings reconstructed in Model 2.1	
	3.6	Bottleneck LSTM-to-BERT + 2-layer FCN $y_{hs} = FCN(MODEL2.2 - L2B_{bottleneck(x_{hstweet})})$	HS embeddings extracted from the bottleneck layer of Model 2.2	
	3.7	Reconstructed LSTM-to-BERT + 2-layer FCN $y_{hs} = FCN(MODEL2.2(x_{hstweet}))$	HS embeddings reconstructed in Model 2.2	

*FCN stands for fully-connected network.

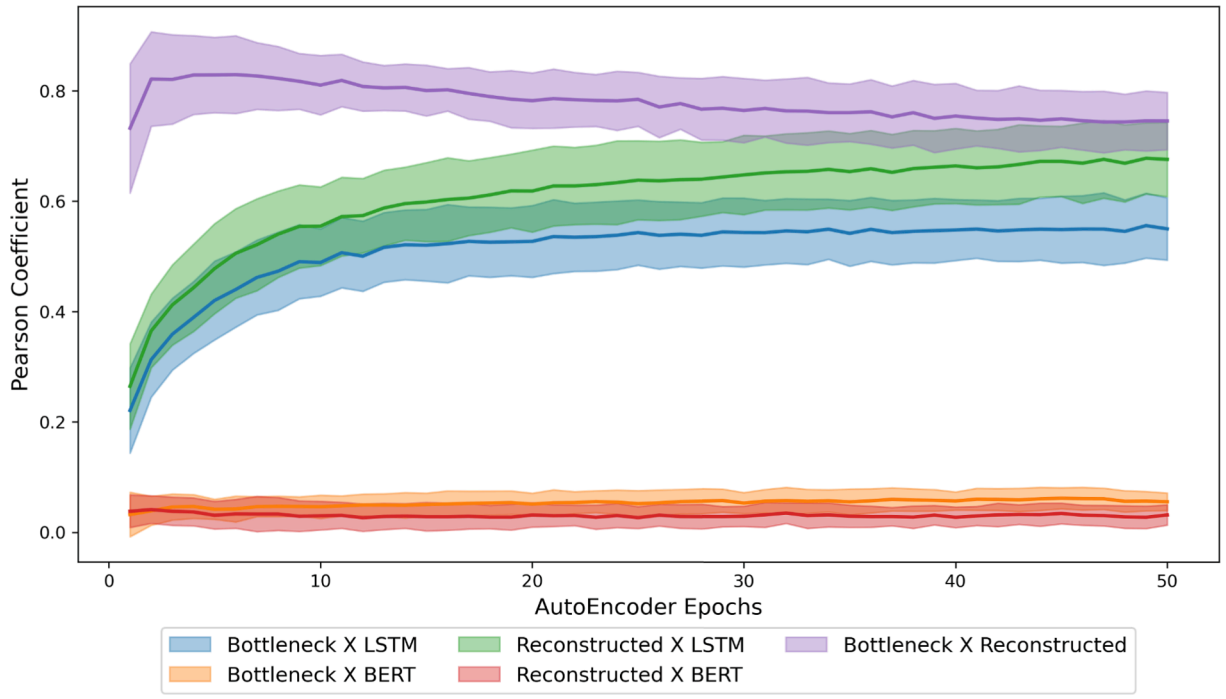

Fig. A1. Pearson correlation for all epochs in BERT-to-LSTM. Bottleneck dimension = 32.

applies them to a different emotional-related task, namely hate speech classification (Sharma, 2018), which is based on a different set of tweets.

3.3.1. Dataset and preprocessing

To examine generalizability, this study used a "Tweets-hate-speech-detection" dataset from HuggingFace (Sharma, 2018), which contains 31,926 tweets annotated as either "hate" or "no-hate". To achieve a smaller and more balanced dataset, we randomly sampled 2,200 tweets from each class. 3,520 tweets were used for training data, and 880 tweets were used for test data.

Mentions, hashtags, emojis and other foreign characters were removed to match with the cleaned tweets in the Emotion dataset. The same pipelines including generating BERT pretrained embeddings, extracting LSTM embeddings from the last timestep of the model based

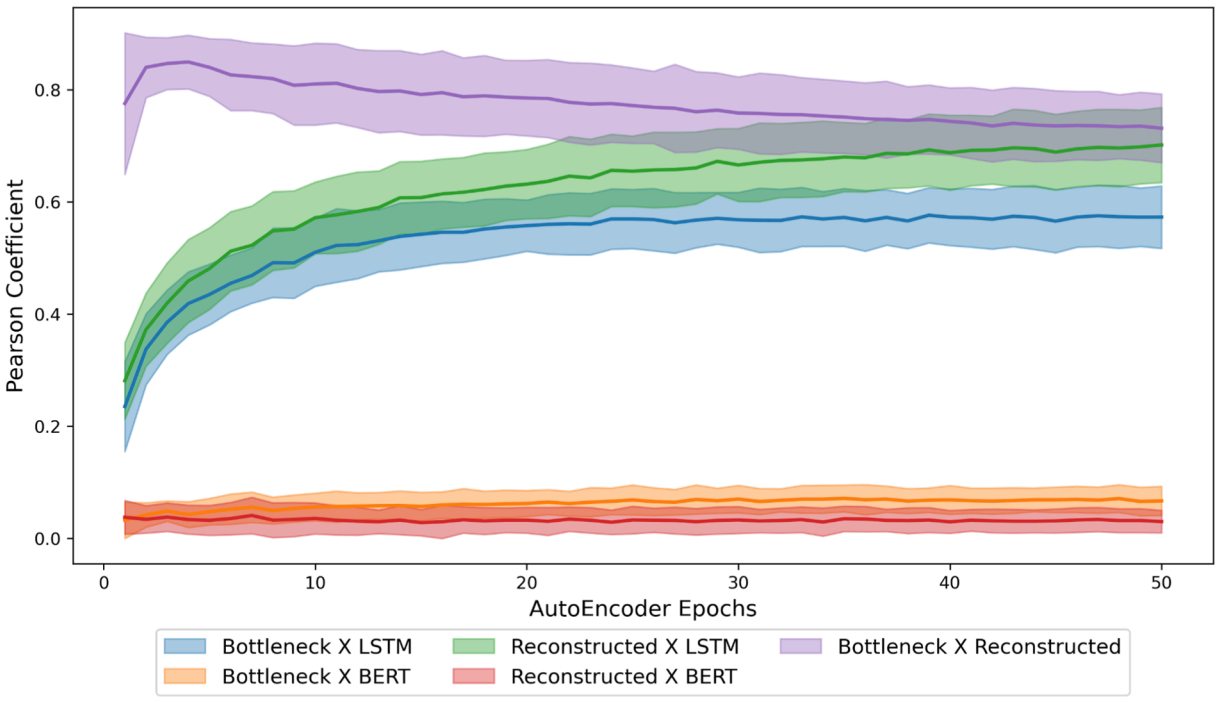

Fig. A2. Pearson correlation for all epochs in BERT-to-LSTM. Bottleneck dimension = 64.

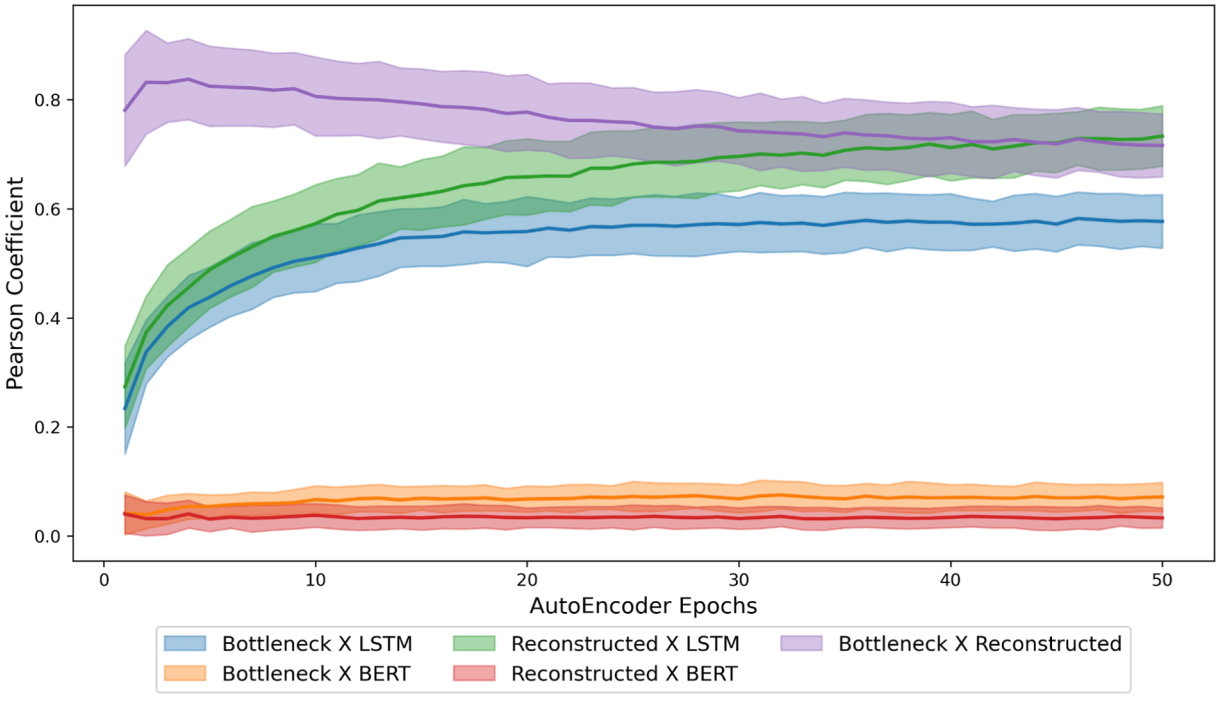

Fig. A3. Pearson correlation for all epochs in BERT-to-LSTM. Bottleneck dimension = 128.

on a sequence of GloVe input, and obtaining bottleneck and reconstructed embeddings from the two autoencoders were applied to the new cleaned tweets. Both the LSTM and autoencoder models were reusing previously trained models in the emotion classification task.

3.3.2. Model selection

A simple two-layer FCN has been used for testing the performance of each representation. Hidden layer size is standardized to 32 across all models to match the smallest embeddings. A baseline FCN with the same architecture was also conducted with the tweet input processed by the "basic-English" tokenizer and torchtext and an nn.embedding layer for performance comparison.

4. Results

4.1. Comparing BERT and LSTM representations

t-Distributed Stochastic Neighbor Embedding (t-SNE), a popular

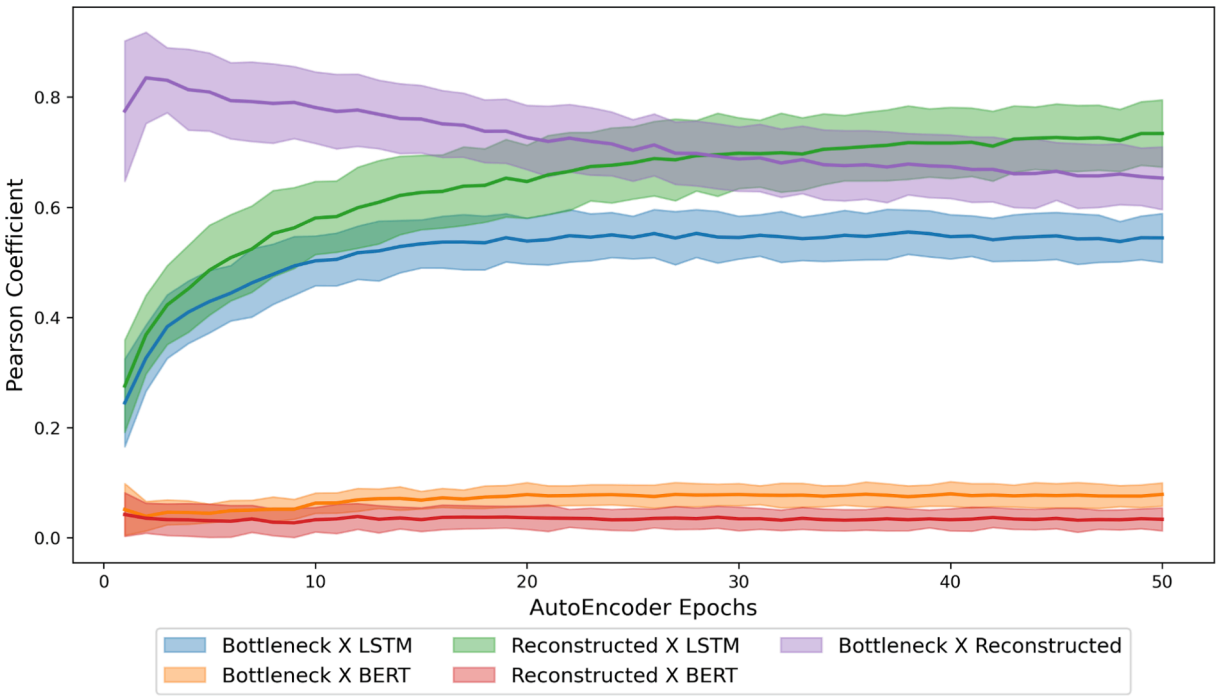

Fig. A4. Pearson correlation for all epochs in BERT-to-LSTM. Bottleneck dimension = 256.

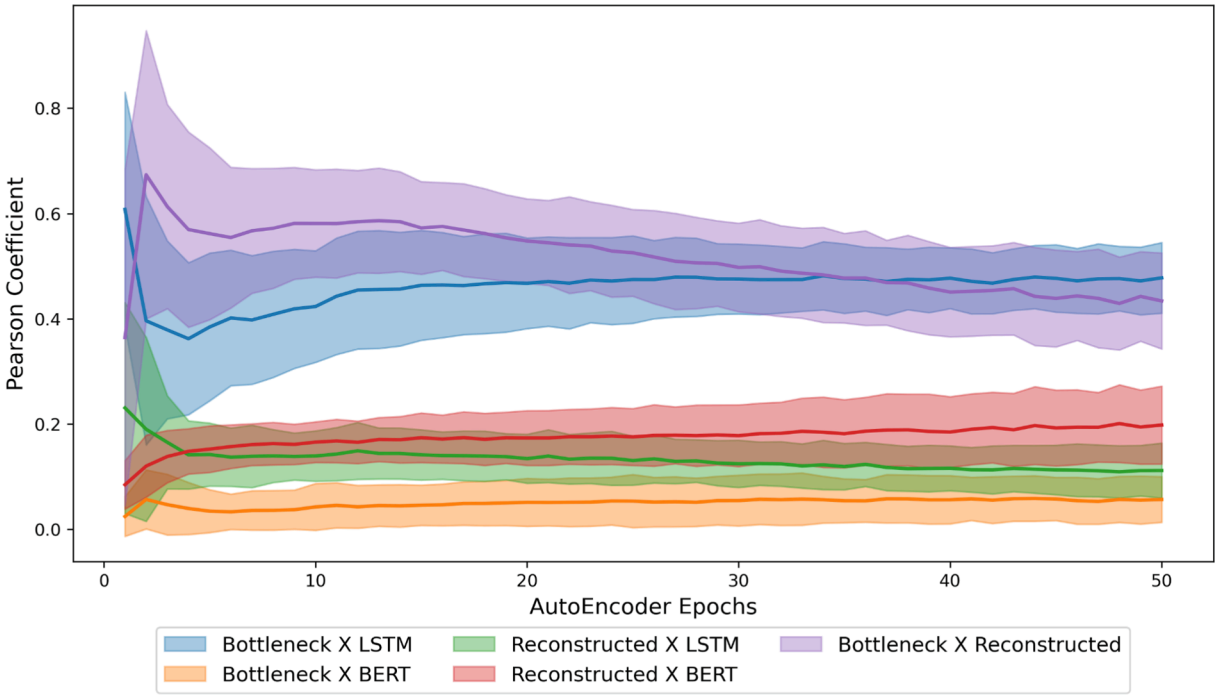

Fig. B1. Pearson correlation for all epochs in LSTM-to-BERT. Bottleneck dimension = 32.

technique for visualizing high-dimensional data in a lower-dimensional space (van der Maaten & Hinton, 2008), has been applied to the sentence embeddings generated by both the LSTM and BERT models. The resulting visualization in Fig. 3 shows that while the LSTM classifier is effective at categorizing tweets into six distinct emotion categories, the BERT pre-trained model captures the semantic meaning of each input without any explicit categorization, confirming that BERT and LSTM embeddings empirically align with global and task-specific representational systems.

4.2. *Representational similarity analysis*

Similar to the t-SNE visualization, the LSTM matrix shows clear distinctions across different emotions, while the semantic information in BERT does not resemble that illustration. The bottleneck and reconstructed representations in BERT-to-LSTM did a great job reconstructing the LSTM embeddings, although they did not entirely replicate the LSTM representation, depending on the task difficulty. For example, the boundary between fear and anger, as well as love and joy, cannot be

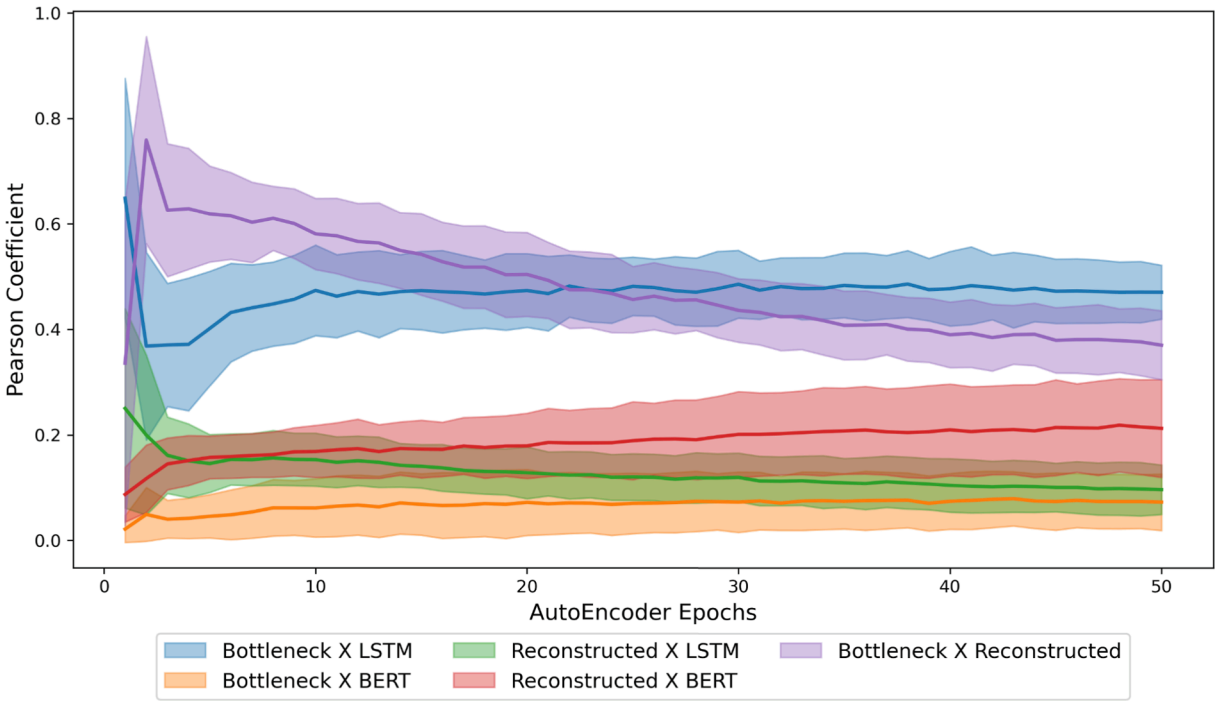

Fig. B2. Pearson correlation for all epochs in LSTM-to-BERT. Bottleneck dimension = 64.

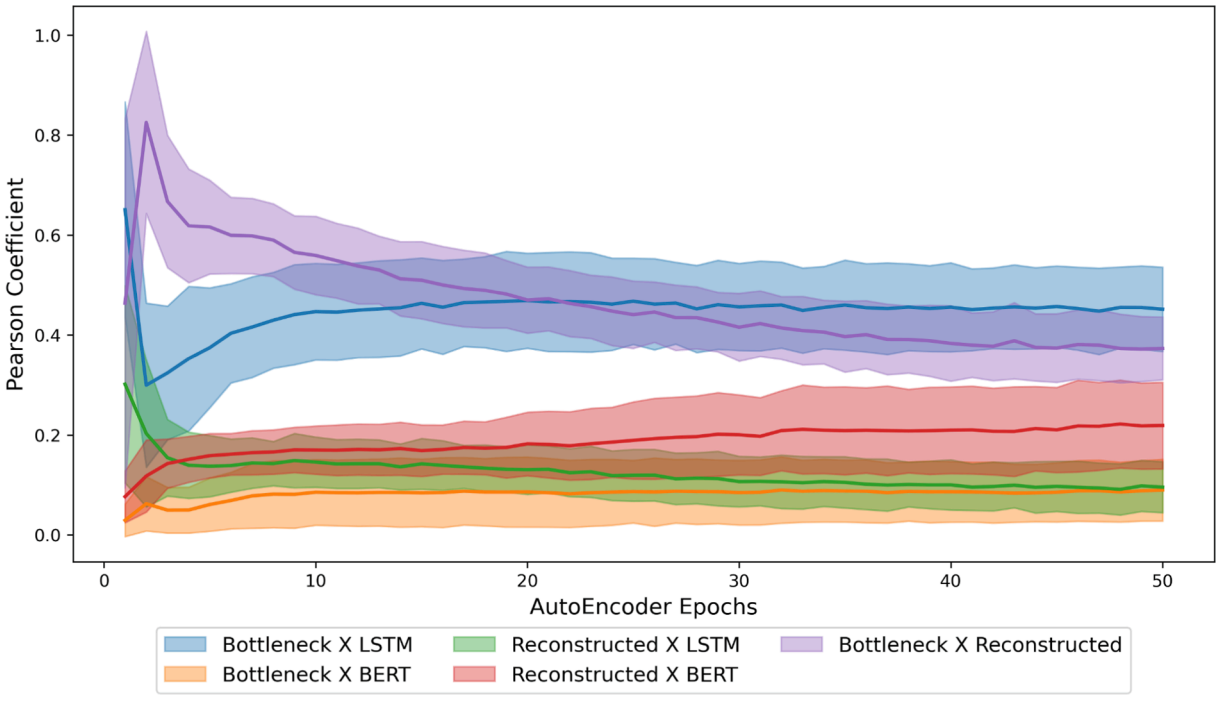

Fig. B3. Pearson correlation for all epochs in LSTM-to-BERT. Bottleneck dimension = 128.

fully recovered. This aligns with the valence-arousal model in emotions, where fear and anger are negative emotions, and love and joy are low arousal, making them more difficult to distinguish than other categories. In the LSTM-to-BERT model, both the bottleneck and reconstructed embeddings struggled to forget the LSTM representation or recover signals from BERT due to the sparsity in the LSTM representation.

Calculating the Pearson correlation between each representational system illustrates similar patterns to the similarity matrices—there is an asymmetry in the system transformation under the two autoencoders. While the BERT-to-LSTM model effectively transforms BERT to LSTM in both its bottleneck and reconstructed embeddings, the LSTM-to-BERT model struggles to erase LSTM signals or generate BERT embeddings. The bottleneck still contains weak signals from LSTM, even when the model has been sufficiently trained.

4.2.1. *Manipulating bottleneck layer and epoch number*

Since the epoch number represents the level of intuition a model has gained from iterating over an entire dataset, it is natural to expect that

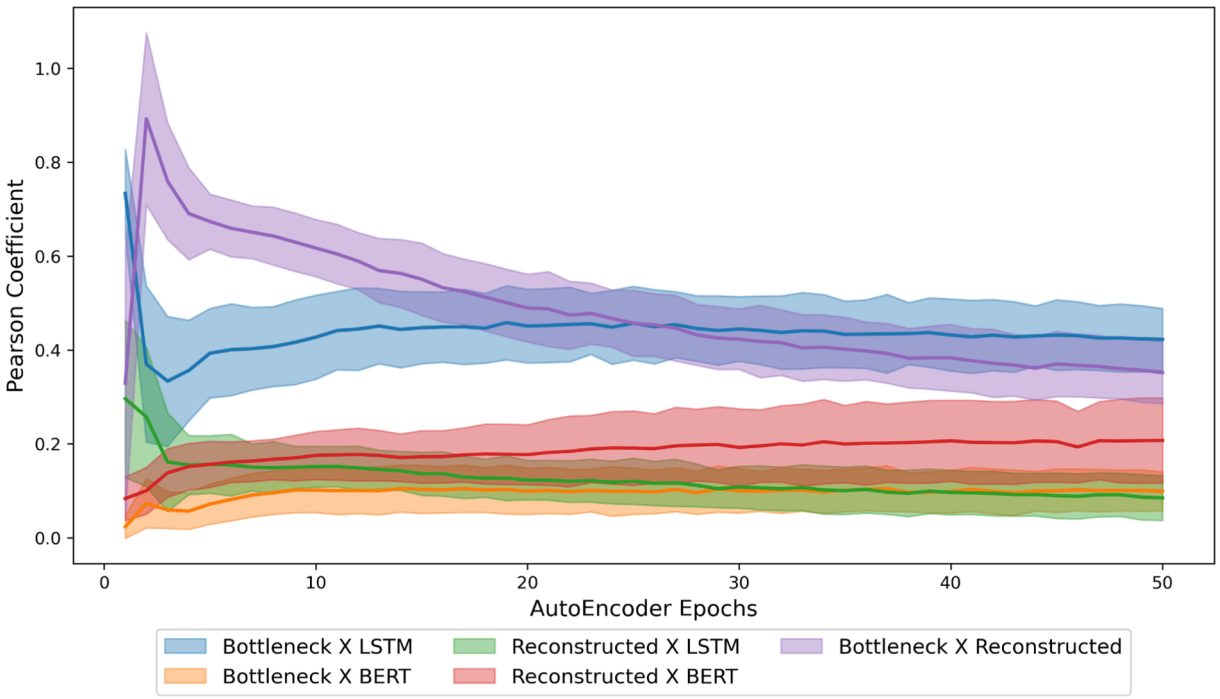

Fig. B4. Pearson correlation for all epochs in LSTM-to-BERT. Bottleneck dimension = 256.

the correlation between the bottleneck and inputs, as well as the reconstructed and inputs, will decrease, and the correlation between the bottleneck and outputs, as well as the reconstructed and outputs, will increase. The transformation trajectory between epoch and correlation for both models (on the test data) is visualized in Fig. 6.

In the BERT-to-LSTM model, both bottleneck and reconstructed embeddings are trained to approximate LSTM embeddings on test data. However, even at the beginning of training, the bottleneck and reconstructed embeddings barely contain any signals from the BERT embeddings. In the LSTM-to-BERT model, both bottleneck and reconstructed embeddings had difficulty recovering signals for BERT or even forgetting information in LSTM. At epoch 50, the reconstructed embeddings contained information from both LSTM and BERT, but they were still unable to sufficiently capture the information from BERT.

One concern that arises is the difference in size between BERT and LSTM (768 vs. 300), as well as the extremely small bottleneck (e.g. dim = 32), might make it challenging to retain information from BERT. Therefore, bottleneck embeddings with different dimensions have been examined (dim = 32, 64, 128, 256; see Appendix II). Although one might expect that a larger bottleneck would better resemble the larger embeddings (i.e. BERT), the pattern of transformation trajectories remains unaffected by the bottleneck sizes. Larger bottleneck sizes can shift the upper bound of learning potential in both models, though the upper potential of the LSTM-to-BERT model has been limited. For example, the Pearson *r* between the reconstructed and BERT embeddings at the final epoch increases from 0.20 to 0.23 when the bottleneck dim increases from 32 to 256, which is still much smaller than the correlation in the opposite direction in the BERT-to-LSTM model (which is around 0.62 – 0.75).

4.3. Generalizability performance

By inheriting representations from models that were trained on emotion classification tasks and further fine-tuned on hate speech classification, this study compares the generalizability performance across the four different systems by test loss and accuracy. An independent *t*-test was conducted across the output performance from 25 runs of the simulation between each model and the baseline FCN. (See Table 1).

As expected, the baseline model exhibits one of the lowest test accuracies and the highest test losses. When benchmarked against the baseline model, nearly all representations were able to meaningfully classify the new task with significantly lower test loss and significantly lower test accuracy.[2] Notably, BERT achieved substantially higher performance with extremely low training loss. The BERT-to-LSTM embeddings, which include BERT's information, demonstrate better performance by leveraging BERT's global knowledge compared to the LSTM-only task-specific embeddings. This is true even when the bottleneck is limited to only 32 dimensions.

5. Discussion

The study identifies two neural network representation systems introduced by recent advancements in deep learning (i.e., global and task-specific representations) and examines the possibility of integrating them into one hybrid model. In accordance with the initial conceptualization, the embeddings generated by the RNN-LSTM and BERT models effectively approximate task-specific and global representational systems, as intended by their respective model architectures as well as the visualization of the embeddings' distribution from similarity matrix and t-SNE.

The autoencoder demonstrates the ability to transform between two systems, as well as diffuse them into a more efficient cognitive representation that embodies characteristics of both. However, the hybrid model performs better when the input system is more informationally dense than the output system. This aligns with practical applications in deep learning that an autoencoder would be more effective at compressing images rather than enhancing their resolution. Similarly, BERT, a transformer model trained on a massive corpus with denser embeddings and prior knowledge, is expected to capture the signal in LSTM

[2] The discrepancy in significance levels and effect sizes between loss and accuracy results could potentially be attributed to the nature of accuracy as a more oscillating metric in deep learning. While accuracy is calculated through binary classification accuracy, the training and test losses gradually decrease through gradient descent when the model is properly trained.

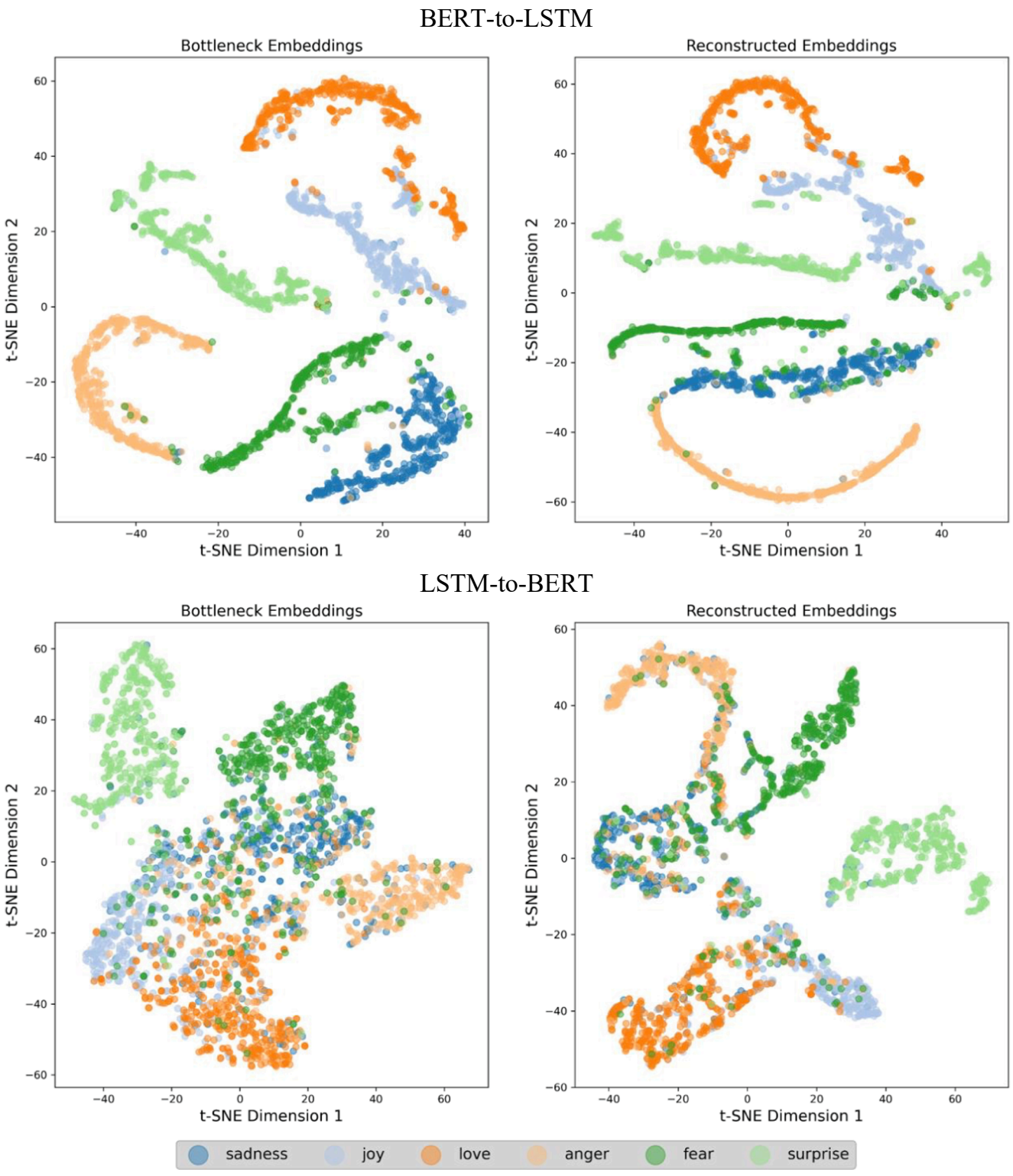

Fig. C1. T-sne visualizations for bottleneck (left) and reconstructed (right) embeddings from the BERT-to-LSTM (top) and LSTM-to-BERT models (bottom).

that is trained solely on signals emphasized by the emotion classification task, but not vice versa.

This finding has practical implications for deep learning research. While autoencoders have the ability to transform or merge representational systems into a shared space, it is important to note that the two systems should have a relatively equivalent amount of information and the underlying tasks should be meaningful for the transformation. Expanding the bottleneck size could potentially alleviate the information constraint, as increasing the size of the bottleneck embeddings aids in preserving information in both directions of transformation; however, the overall pattern of the transformation trajectory typically remains unchanged, and the potential upper bound is still restricted when converting a sparse representation into a denser one.

An intriguing observation from the similarity matrices of the LSTM-to-BERT model is that despite the autoencoder being adequately trained to reconstruct BERT, the bottleneck layer struggled to forget information

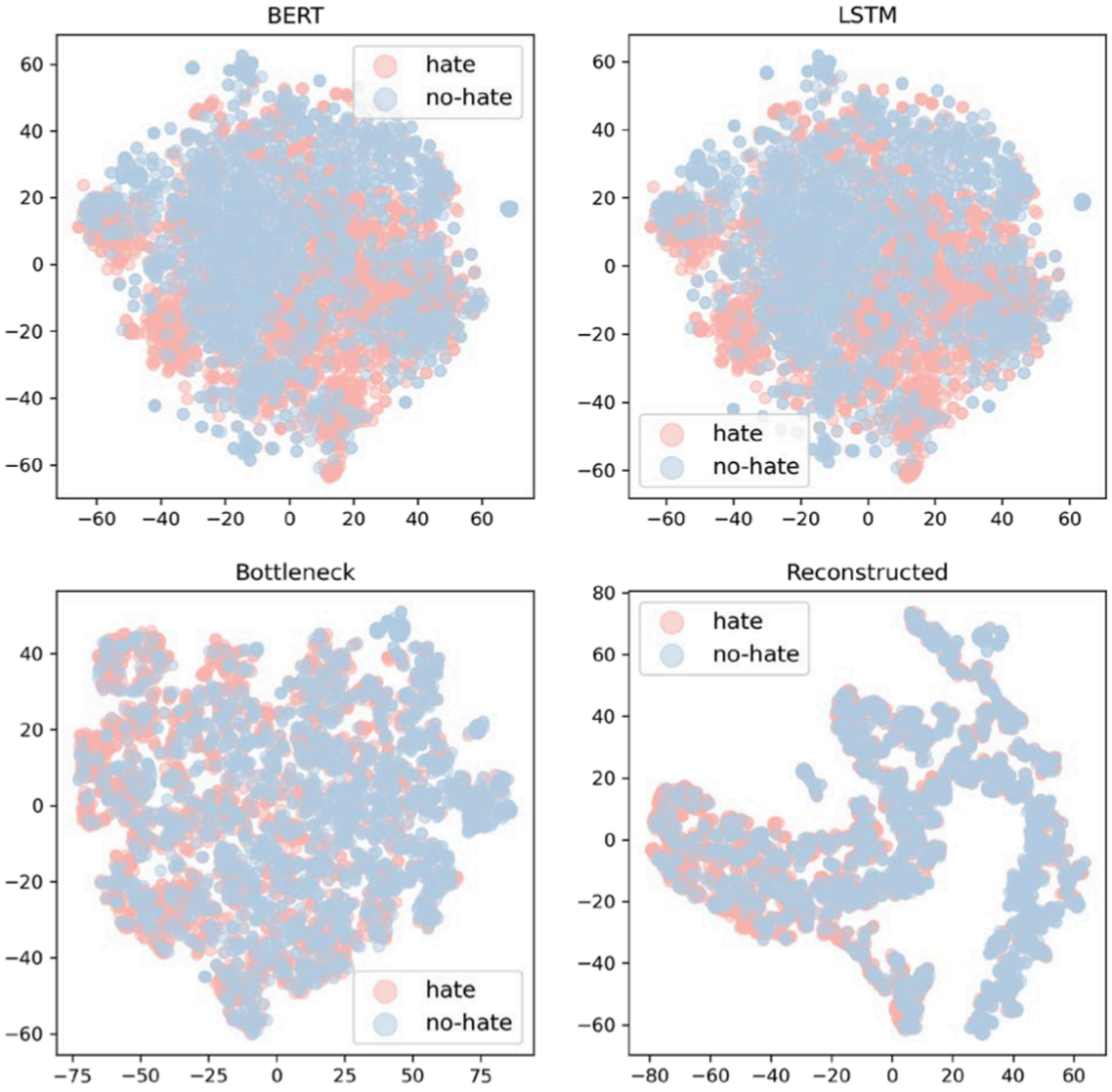

Fig. D1. t-SNE visualizations for all four representations of tweets from the hate speech classification task.

in LSTM, likely because the pooling Conv1D layer in the autoencoder was spatially aligned with how emotional information is structured in a sequence of tokens. This might be because emotions in a tweet are likely stored in phrases that consist of multiple tokens, and the pooling process efficiently preserves these signals in the bottleneck embeddings. In contrast, Conv1D is likely to extinguish the dense information representation in BERT, which could clarify why the BERT-to-LSTM bottleneck rapidly forgets a substantial amount of information in BERT. While these assumptions could be explored using an FCN instead of a Conv1D autoencoder, this study could not test this scenario due to the lack of substantial training samples to adequately train an FCN-based autoencoder.

Finally, when applying the four representational systems to a new dataset that could benefit from emotional signals, all embeddings, including LSTM which was generated from an emotion classification task, demonstrated a better performance in terms of test loss than a baseline FCN model, albeit the gain was not significant in LSTM or any representations that inherit some information from LSTM. Despite having a very low dimension, the bottleneck (dim = 32) still performed significantly better than the baseline FCN model, indicating that semantic signals were efficiently captured in the bottleneck layer. As anticipated, the global representations generated by a powerful model outperformed any other approach by a significant margin. It is worth noting that this exceptional BERT performance was achieved without fine-tuning the transformer BERT but merely by feeding the pre-trained BERT to an FCN output layer, which is a testament to the remarkable performance of BERT even as a contextual vectorization model. Additionally, any representations that contained some level of BERT information benefitted from its world knowledge, as indicated by the smaller test losses of diffused and transformed embeddings. This is intriguing from an RSA standpoint, as even though the bottleneck and reconstructed representations exhibit little resemblance to the BERT global representations, BERT still appeared to have a significant impact on the downstream generalizability task. These observations further confirm that global knowledge is crucial for adapting to a new learning task.

Although the study was unable to construct bottleneck embeddings that represented a successful diffusion of two representational systems and demonstrate the advantages of a successfully merged system, it sheds light on the directional implementation that could aid in successful transformation and diffusion, such as adjusting bottleneck sizes or determining the theoretical compatibility of the representational systems. Furthermore, the model performance on the generalizability task reinforces the confidence that once an autoencoder can achieve satisfying transformed and diffused representations, downstream tasks could significantly benefit from this successfully diffused hybrid model.

The general findings in this paper confirm the complementary

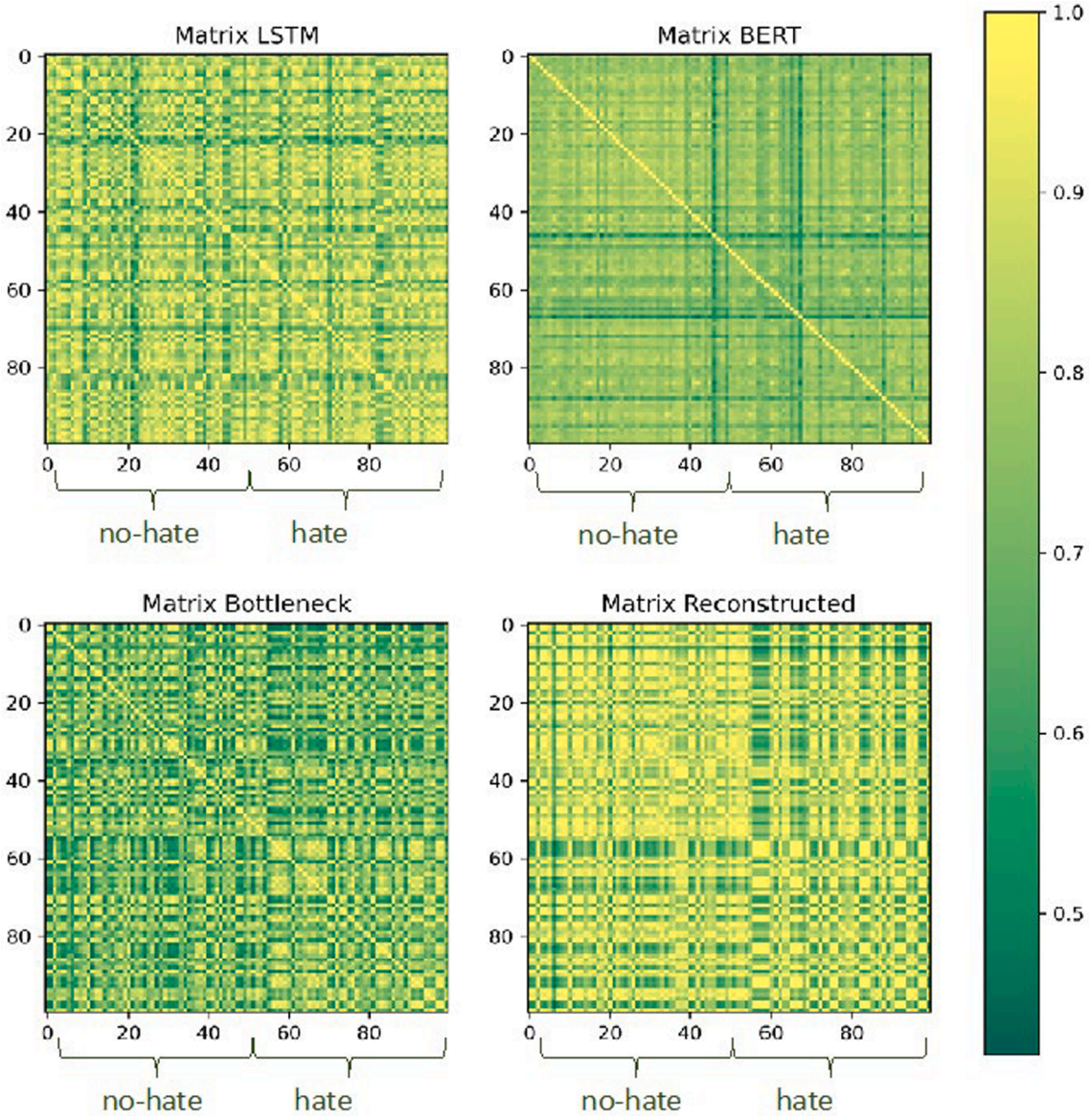

Fig. E1. Similarity matrices for all four representations of tweets from the hate speech classification task.

learning framework in natural cognition, in that they suggest that cognitive systems can use global knowledge to navigate through different specific situations (O'Reilly & Norman, 2002) – and that having only local knowledge greatly reduces the transferability of the system. Hybrid computational systems hold promise to expand the functionality of AI systems and also bring their performance into closer proximity to that of humans. Indeed, hybrid cognitive systems in cognitive science have shown impressive performance along a number of tasks (Hélie & Sun, 2010).

The present results offer some insights into these practical and theoretical developments. For example, our results suggest there is a strong constraint from information gradients (i.e. how information changes or varies across different data representations). These gradients are intrinsic to global vs. task-specific representations. Global representations may require a higher dimensionality with higher-entropy encoding; task-specific may usually be tuned to lower-dimensional and lower entropy formulations of a specific task. The information-theoretic gradient across representational schemes (regardless of their specific function) may be the reason for this constraint in what derived format is most useful in generalization and other tasks. Though this statement may seem intuitive, a modeling platform of the kind we show here affords the means to test and quantify the effect of these gradients, and to determine how they may be assessed in future models.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

I have shared the data/code in the comment and cover letter.

Appendix I

See Table A1.

Appendix II

BERT-to-LSTM

See Figs. A1–A4.

LSTM-to-BERT

See Figs. B1–B4.

Appendix III

See Fig. C1.

Appendix IV

To evaluate the generalizability task, the t-SNE visualization and similarity matrices of the four embeddings were further examined to ensure that the new classifiers were not simply relying on any naive or trivial patterns in the representations. Neither visualization indicated any explicit patterns in the distribution of the embeddings (See Fig. D1 and Fig. E1).

References

Baan, J., Hoeve, M.T., Wees, M.V., Schuth, A., & de Rijke, M. (2019). Do transformer attention heads provide transparency in abstractive summarization? *ArXiv*, abs/1907.00570.

Baddeley, A. D., Hitch, G. J., & Allen, R. J. (2019). From short-term store to multicomponent working memory: The role of the modal model. *Memory & Cognition, 47*(4), 575–588. https://doi.org/10.3758/s13421-018-0878-5

Chemero, A. (2001). Dynamical explanation and mental representations. *Trends in Cognitive Sciences, 5*(4), 141–142.

Chen, Z., Flemotomos, N., Singla, K., Creed, T. A., Atkins, D. C., & Narayanan, S. (2022). An automated quality evaluation framework of psychotherapy conversations with local quality estimates. *Computer Speech & Language, 75*, Article 101380. https://doi.org/10.1016/j.csl.2022.101380

Conway, C. M. (2020). How does the brain learn environmental structure? Ten core principles for understanding the neurocognitive mechanisms of statistical learning. *Neuroscience & Biobehavioral Reviews, 112*, 279–299. https://doi.org/10.1016/j.neubiorev.2020.01.032

Devlin, J., Chang, M. -W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. *In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 1, 4171–4186. https://doi.org/10.18653/v1/N19-1423.

Dutta, S. (2018). An overview on the evolution and adoption of deep learning applications used in the industry. *WIREs Data Mining and Knowledge Discovery, 8*(4). https://doi.org/10.1002/widm.1257

Edmonds, M., Gao, F., Liu, H., Xie, X., Qi, S., Rothrock, B., Zhu, Y., Wu, Y. N., Lu, H., & Zhu, S.-C. (2019). A tale of two explanations: Enhancing human trust by explaining robot behavior. *Science Robotics, 4*(37). https://doi.org/10.1126/scirobotics.aay4663

Ekman, P. (1992). An argument for basic emotions. *Cognition & Emotion, 6*(3–4), 169–200. https://doi.org/10.1080/02699939208411068

Elman, J. L. (1990). Finding structure in time. *Cognitive Science, 14*(2), 179–211.

Go, A., Bhayani, R., & Huang, L. (2009). Twitter sentiment classification using distant supervision. *Stanford Digital Library Technologies Project, 30*, 1–12.

Groen, I. I. A., Greene, M. R., Baldassano, C., Fei-Fei, L., Beck, D. M., & Baker, C. I. (2018). Distinct contributions of functional and deep neural network features to representational similarity of scenes in human brain and behavior. *ELife, 7*. https://doi.org/10.7554/elife.32962

Haenlein, M., & Kaplan, A. (2019). A brief history of artificial intelligence: On the past, present, and future of Artificial Intelligence. *California Management Review, 61*(4), 5–14. https://doi.org/10.1177/0008125619864925

Hélie, S., & Sun, R. (2010). Incubation, insight, and creative problem solving: A unified theory and a connectionist model. *Psychological Review, 117*(3), 994.

Hinton, G. E., & Zemel, R. S. (1994). Autoencoders, minimum description length, and Helmholtz free energy. In J. D. Cowan, G. Tesauro, & J. Alspector (Eds.), *Advances in Neural Information Processing Systems, 6*, 3-10, https://proceedings.neurips.cc/paper_files/paper/1993/file/9e3cfc48eccf81a0d57663e129aef3cb-Paper.pdf.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation, 9*(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

Jilk, D. J., Lebiere, C., O'Reilly, R. C., & Anderson, J. R. (2008). SAL: An explicitly pluralistic cognitive architecture. *Journal of Experimental & Theoretical Artificial Intelligence, 20*(3), 197–218. https://doi.org/10.1080/09528130802319128

Jordan, M. I. (1986). *Serial order: A parallel distributed processing approach*. Institute for Cognitive Science: University of California, San Diego.

Koroteev, M. V. (2021). BERT: A Review of Applications in Natural Language Processing and Understanding. *ArXiv, abs/2103.11943*.

Kriegeskorte, N., Mur, M., & Bandettini, P. (2008 Nov). Representational similarity analysis - connecting the branches of systems neuroscience. *Front Syst Neurosci., 24*(2), 4. https://doi.org/10.3389/neuro.06.004.2008. PMID: 19104670; PMCID: PMC2605405

Lee, J., Jeong, K., & Kim, W. (2022). Multivariate time series traffic anomaly detection with Prediction & AutoEncoder.

Lu, P., Bai, T., & Langlais, P. (2019). Sc-lstm: Learning task-specific representations in multi-task learning for sequence labeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 1*, 2396-2406.

Markman, A. B., & Dietrich, E. (2000). Extending the classical view of representation. *Trends in cognitive sciences, 4*(12), 470–475.

Michelucci, U. (2022). An introduction to autoencoders. arXiv preprint arXiv: 2201.03898.

Mikolov, T., Karafiát, M., Burget, L., Černockỳ, J., & Khudanpur, S. (2010). Recurrent neural network based language model. In *Proceedings of the 11th Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 1045-1048.

Nason, S., & Laird, J. E. (2005). Soar-RL: Integrating reinforcement learning with Soar. *Cognitive Systems Research, 6*(1), 51–59. https://doi.org/10.1016/j.cogsys.2004.09.006

Nili, H., Wingfield, C., Walther, A., Su, L., Marslen-Wilson, W., & Kriegeskorte, N. (2014). A Toolbox for Representational Similarity Analysis. *PLoS Computational Biology, 10*(4), e1003553.

O'Reilly, R. C., & Norman, K. A. (2002). Hippocampal and neocortical contributions to memory: Advances in the complementary learning systems framework. *Trends in Cognitive Sciences, 6*(12), 505–510. https://doi.org/10.1016/S1364-6613(02)02005-3

Pennington, J., Socher, R., & Manning, C. D. (2014). GloVe: Global vectors for word representation. In *In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (pp. 1532–1543). https://doi.org/10.3115/v1/D14-1162

Plutchik, R. (2001). The nature of emotions. *American Scientist, 89*(4), 344–350.

Rogers, A., Kovaleva, O., & Rumshisky, A. (2020). A primer in Bertology: What we know about how Bert Works. *Transactions of the Association for Computational Linguistics, 8*, 842–866. https://doi.org/10.1162/tacl_a_00349

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature, 323*(6088), 533–536. https://doi.org/10.1038/323533a0

Saravia, E., Liu, H.-C.-T., Huang, Y.-H., Wu, J., & Chen, Y.-S. (2018). CARER: Contextualized affect representations for emotion recognition. In *In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (pp. 3687–3697).

Sharma, R. (2018). Tweets Hate Speech Detection. Retrieved [March, 23, 2023], from https://huggingface.co/datasets/tweets_hate_speech_detection.

Sherstinsky, A. (2018). Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) Network. *ArXiv*, abs/1808.03314.

Squire, L. R. (2004). Memory systems of the brain: A brief history and current perspective. *Neurobiology of Learning and Memory, 82*(3), 171–177. https://doi.org/10.1016/j.nlm.2004.06.005

Sun, R., Merrill, E., & Peterson, T. (2001). From implicit skills to explicit knowledge: A bottom-up model of skill learning. *Cognitive Science, 25*(2), 203–244. https://doi.org/10.1207/s15516709cog2502_2

Tulving, E. (1985). How many memory systems are there? *American Psychologist, 40*(4), 385–398. https://doi.org/10.1037/0003-066X.40.4.385

van der Maaten, L., & Hinton, G. (2008). Visualizing data using t-SNE. *Journal of Machine Learning Research, 9*(11), 2579–2605.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Neural Information Processing Systems., 30*. https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., & Brew, J. (2020). Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 38-45. https://doi.org/10.18653/v1/2020.emnlp-demos.6.

Xie, F., Zhu, Y., Wang, B., Wang, W., & Jin, P. (2023). End-to-end underwater acoustic communication based on Autoencoder with dense convolution. *Electronics, 12*(2), 253. https://doi.org/10.3390/electronics12020253

Yannakakis, G. N., Cowie, R., & Busso, C. (2021). The ordinal nature of emotions: An emerging approach. *IEEE Transactions on Affective Computing, 12*(1), 16–35. https://doi.org/10.1109/taffc.2018.2879512