Using AI to Analyze Personality

“Transforming” Personality Scale Development: Illustrating the Potential of State-of-the-Art Natural Language Processing

2023 *S. Fyffe, P. Lee, S. Kaplan*

Regular Submission

Organizational Research Methods
1–36
© The Author(s) 2023
Article reuse guidelines:
sagepub.com/journals-permissions
DOI: 10.1177/10944281231155771
journals.sagepub.com/home/orm
SSAGE

"Transforming" Personality Scale Development: Illustrating the Potential of State-of-the-Art Natural Language Processing

Shea Fyffe, Philseok Lee, and Seth Kaplan

Abstract

Natural language processing (NLP) techniques are becoming increasingly popular in industrial and organizational psychology. One promising area for NLP-based applications is scale development; yet, while many possibilities exist, so far these applications have been restricted—mainly focusing on automated item generation. The current research expands this potential by illustrating an NLP-based approach to *content analysis*, which manually categorizes scale items by their measured constructs. In NLP, content analysis is performed as a text classification task whereby a model is trained to automatically assign scale items to the construct that they measure. Here, we present an approach to text classification—using state-of-the-art *transformer* models—that builds upon past approaches. We begin by introducing transformer models and their advantages over alternative methods. Next, we illustrate how to train a transformer to content analyze Big Five personality items. Then, we compare the models trained to human raters, finding that transformer models outperform human raters and several alternative models. Finally, we present practical considerations, limitations, and future research directions.

Keywords

personality, scale development, machine learning, natural language processing, text classification, transformers

Researchers have spent decades echoing the challenging nature of scale development (e.g., Condon et al., 2020; Hinkin, 1998; McCrae & Costa, 1987; Norman, 1963); often describing it as a process of "trial and error" (Tellegen & Waller, 2008, p. 262)—whereby subject matter experts (SMEs) write, review, and revise items to best measure psychological attributes (Clark & Watson, 2016, 2019). If not carefully developed, scales could include unrelated items or items that underrepresent the construct(s) of interest (Hattie, 1985; Rosellini & Brown, 2021; Smith et al., 2022). So, best practices

Department of Psychology, George Mason University, Fairfax, VA, USA

Corresponding author:
Shea Fyffe, Department of Psychology, George Mason University, 4400 University Drive, MSN 3F5, Fairfax, VA 22030, USA.
Email: sfyffe@gmu.edu

suggest performing a *content analysis*, a process that evaluates the degree to which item content relates to the psychological construct(s) of interest (i.e., content validation; Anderson Gerbing, 1991; Colquitt et al., 2019; Worthington & Whittaker, 2006).

Content analysis has had a significant and historic role in scale development (e.g., Allport, 1937; Loevinger, 1957). This role is likely to grow in importance as researchers and practitioners deal both with more nuanced constructs and with the need to avoid construct proliferation. Most recently, researchers have begun proposing various machine-learning techniques to streamline the scale development process, for example, automated item generation, automated scoring, and automated test assembly (e.g., Campion et al., 2016; Hommel et al., 2022; Jiao & Lissitz, 2020; Lee et al., 2023; von Davier, 2018). Yet, the standard approach to content analysis is mostly a manual process that demands considerable time, cognitive effort, and decision-making from SMEs (Krippendorff, 2018; Short et al., 2018). After considering the recent growth of machine-learning techniques, automated item generation in particular (e.g., Götz et al., 2021; Hommel et al., 2022; Lee et al., 2023; von Davier, 2018), manually conducting content analysis becomes untenable since it would be virtually impossible for SMEs to quickly evaluate a large number of (e.g., a few thousand) generated items. These dynamics suggest a need for more efficient ways to conduct content analysis.

In response to this need, researchers have recently begun applying *natural language processing* (NLP)[1] techniques to automate the content analysis process (e.g., Ilmini & Fernando, 2017; Kobayashi et al., 2018a; Pandey & Pandey, 2019). The particular NLP strategy used for content validation is known as *text classification* (Kobayashi et al., 2018a). In text classification, an NLP model learns to automatically classify text documents into predetermined categories or *classes* by identifying similar patterns among text documents within a class (Kowsari et al., 2019). Here, we use the term "document" to encompass the various units of natural language text composed of at least one word (e.g., words, phrases, sentences, or paragraphs). To train a classification model, researchers must collect enough *labeled* documents. A "label" is a specific tag representing a class or collection of documents with similar content.[2]

In training a text classification model for content analysis, one can treat scale items as documents and their respective constructs as the labels. For example, the personality item "I am the life of the party" can be treated as a text document representing the class "extraversion." A model that could effectively perform such a task would dramatically increase the efficiency of a researcher's scale development process and provide several additional advantages. For example, a text classification model trained to perform content analysis could flag potentially problematic items (i.e., "blended" or cross-loading items) before collecting response data, preselect the most content-representative items among a large pool of pilot items before SME review, and instantly provide data-driven feedback to item writers regarding how well-constructed items align with the construct or constructs of interest.

Although text classification has a large potential when applied to content analysis, two issues still exist in the literature. First, research demonstrating text classification techniques in organizational and psychological scale development is scarce. As such, there is little understanding of the advantages, concerns, and limitations of applying text classification for content analyzing scale items. A second issue arises from previous demonstrations of text classification in organizational and psychological research. While impressive in many ways, early studies often illustrate techniques that involve collecting a large number of text documents which require a significant amount of time to clean, preprocess, and label (e.g., Ilmini & Fernando, 2017; Kobayashi et al., 2018a; Pandey & Pandey, 2019). Recent developments in computer science and computational linguistics have led to significant improvements in text classification (Wolf et al., 2020); however, organizational and psychological researchers are still relatively unfamiliar with these techniques (Boyd & Schwartz, 2021; Eichstaedt et al., 2020; Kennedy et al., 2021).

In light of these issues, the present research aims to (a) introduce organizational researchers to state-of-the-art *transformer* models (see Vaswani et al., 2017); (b) illustrate how to use text classification to automate the content analysis process; (c) provide reproducible code and data for training such models; (d) compare the effectiveness of various text classification techniques with human raters when performing content analysis; and (e) discuss practical, methodological, and substantive concerns when using the proposed method in scale development.

The Present Research

In pursuit of our objectives, we first describe how transformers fit into the text classification process. Second, we present the factors that led to the emergence of transformer models in text classification, then we discuss their advantages over other NLP approaches. Third, we outline the steps to train and fine-tune a transformer-based text classification model. We provide a step-by-step tutorial illustrating how to automate the content analysis process. Specifically, we illustrate several ways to train transformers to classify personality items by their trait label to assess the *content relevance* (Haynes et al., 1995).[3] Fourth, we evaluate the efficacy of our proposed approach by comparing its accuracy to human raters and several other NLP models when performing content analysis. Lastly, we discuss methodological considerations and recommendations while progressing through each step of our text classification approach. In summary, this research strives to clarify the role of NLP in a critical scale development process—content analysis—in hopes of presenting NLP as an essential yet accessible element in the future of organizationally relevant scale development.

An Introduction to Transformers in Text Classification: Concepts and Developments

Text classification aims to train a *classification model* to assign text documents to predefined classes or categories (Kobayashi et al., 2018a). A classification model takes in a piece of text and outputs a label representing a predicted class or category. A classification model combines two components—the text representation method and the classification algorithm (see Figure 1). Although these components are distinct—allowing researchers to use various techniques for each—they combine to determine the overall accuracy of the classification model (Domingos, 2012; Kobayashi et al., 2018a). The text representation method converts raw text documents into a numeric form, for instance, a vector of 0's and 1's representing the presence or absence of particular words in a document. These numeric representations, called *encodings* or *embeddings*, are used as input to train the classification algorithm. When training, the classification algorithm aims to learn a function that most accurately predicts class labels predetermined by the researcher (Kobayashi et al., 2018a). While researchers perform several substeps during text classification (e.g., Kobayashi et al., 2018a; Kowsari et al., 2019; Mirończuk & Protasiewicz, 2018), these steps all seek to improve the quality of the embeddings (i.e., text representations) and classification algorithm.

The Emergence of Transformer Models

Researchers have developed increasingly sophisticated classification algorithms in hopes of better classifying text (see Kowsari et al., 2019). However, advancements in text representation are the primary source of the recent improvements in text classification (see Pilehvar & Camacho-Collados, 2020 for an overview)—most notably *transformer models* (a.k.a., transformers, Vaswani et al., 2017). Transformers are a type of deep neural network (i.e., neural network architecture) used to convert text into numeric representations (i.e., embeddings). Since transformers are primarily a text representation method, researchers can use them to perform a broad array of NLP tasks (not just text

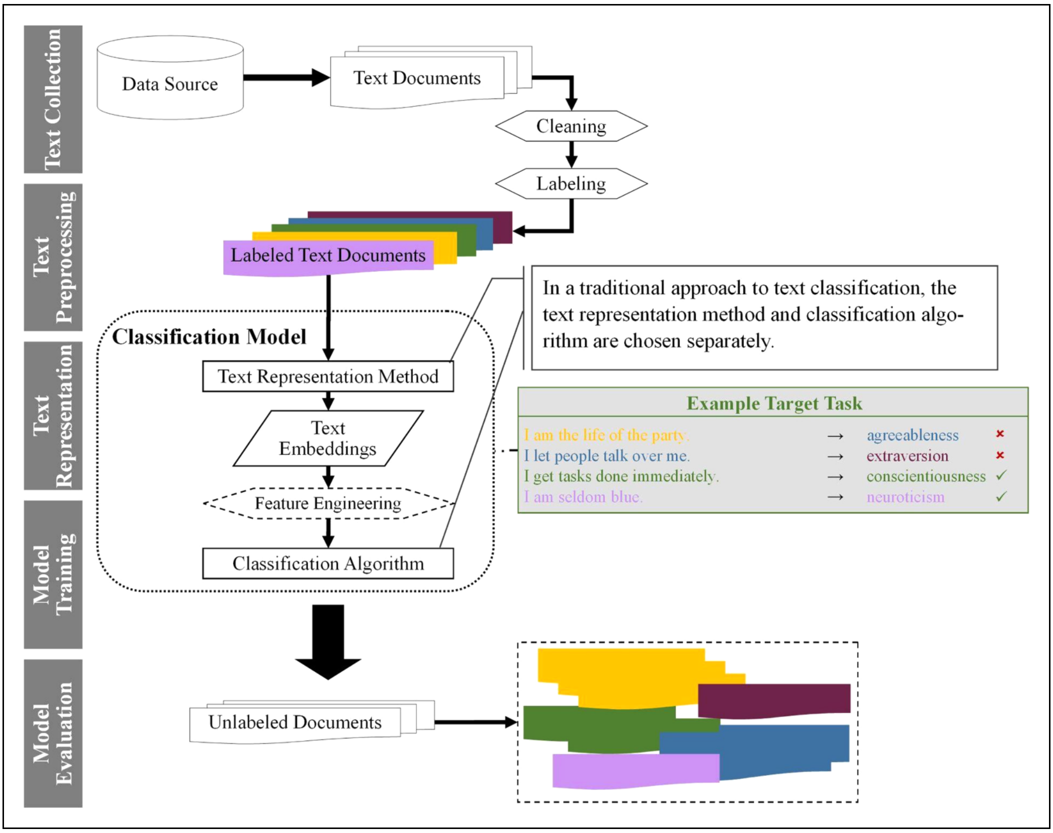
Text Collection
Text Preprocessing
Text Representation
Model Training
Model Evaluation
Data Source
Text Documents
Cleaning
Labeling
Labeled Text Documents
Classification Model
Text Representation Method
Text Embeddings
Feature Engineering
Classification Algorithm
In a traditional approach to text classification, the text representation method and classification algorithm are chosen separately.
Example Target Task
I am the life of the party. → agreeableness ×
I let people talk over me. → extraversion ×
I get tasks done immediately. → conscientiousness ✓
I am seldom blue. → neuroticism ✓
Unlabeled Documents

Figure 1. Traditional approach to text classification.

classification). As such, the transformer models that have grown in popularity (e.g., Bidirectional Encoder Representations from Transformers [BERT]; Devlin et al., 2019; Generative Pretrained Transformer 3 [GPT-3]; Brown et al., 2020; Sentence-BERT [SBERT]; Reimers & Gurevych, 2019) are just specialized types of transformers—developed to accelerate in a particular area of NLP. Nonetheless, the key similarity between these models is that they all use transformer architecture for text representation.

The development of the transformer architecture arose from the need for an efficient but effective way to represent complexities in human language (Vaswani et al., 2017). Unlike early "count-based" approaches to text representation (e.g., bag-of-words and bag-of-n-grams; Harris, 1954) and even well-established shallow and deep neural network approaches (e.g., *ELMo*; Peters et al., 2018; *word2vec*; Mikolov et al., 2013), transformers are highly efficient when producing embeddings (Kennedy et al., 2021; Liu et al., 2020). Specifically, *contextual embeddings* account for both the meaning of a word and the context in which the word is used (Eichstaedt et al., 2020).

A transformer's neural network performs several key operations to produce rich contextual embeddings. First, a transformer model splits text into several smaller pieces (usually words or subwords). Then the model gives each word an embedding—either randomly or using a pre-existing embedding learned during pre-training (see Figure 2 for more information on pre-training). Then, word embeddings combine with positional encodings, which modify the embeddings to account for each word's relative position in the text document. Accounting for word order is crucial when capturing a text's meaning (Boyd & Schwartz, 2021). For example, the words "job" and "fair" in

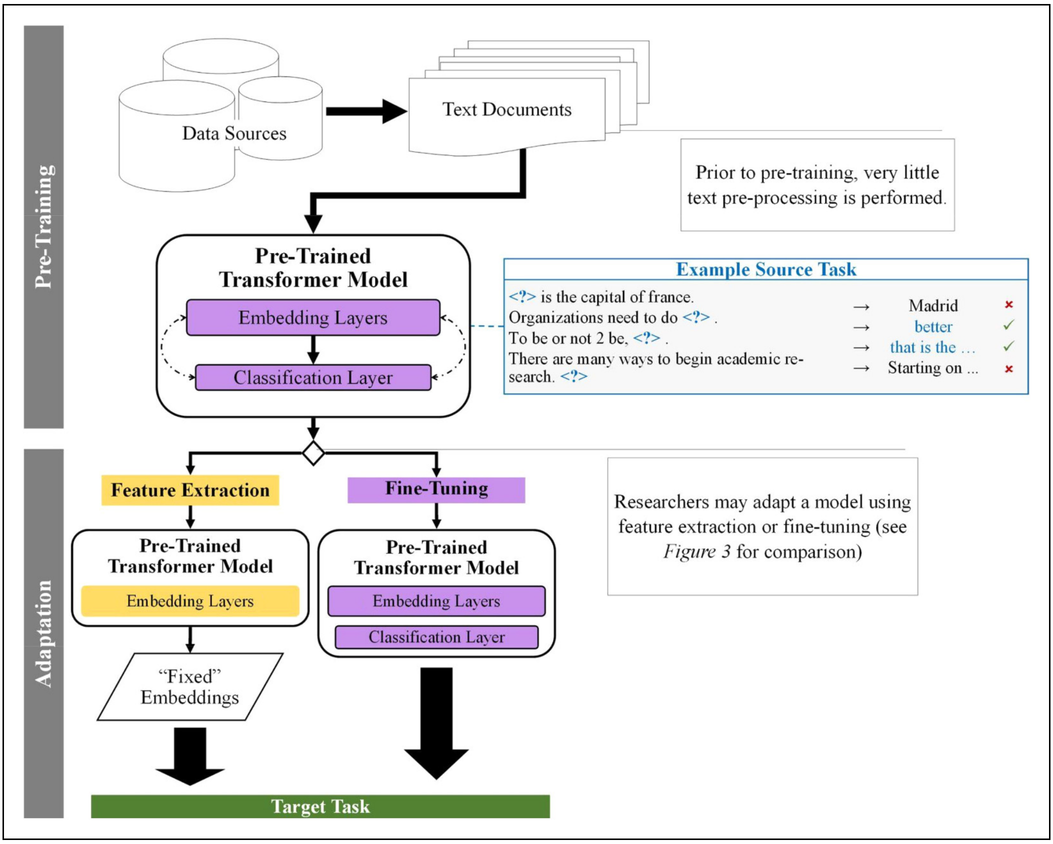
Pre-Training
Data Sources
Text Documents
Prior to pre-training, very little text pre-processing is performed.
Pre-Trained Transformer Model
Embedding Layers
Classification Layer
Example Source Task
<?> is the capital of france. → Madrid ×
Organizations need to do <?> . → better ✓
To be or not 2 be, <?> . → that is the ... ✓
There are many ways to begin academic research. <?> → Starting on ... ×
Adaptation
Feature Extraction
Fine-Tuning
Pre-Trained Transformer Model
Embedding Layers
"Fixed" Embeddings
Pre-Trained Transformer Model
Embedding Layers
Classification Layer
Researchers may adapt a model using feature extraction or fine-tuning (see Figure 3 for comparison)
Target Task

Figure 2. Transfer learning approach to text classification.

the text documents "job fair" and "fair job" will have different contextual embeddings in each document because they appear in distinct positions. The addition of positional encodings gives transformers advantages over early word embedding approaches (e.g., *gloVe*; Pennington et al., 2014; *word2vec*; Mikolov et al., 2013) since transformers can have multiple representations for the same word depending on the context. Next, in a series of steps, the model updates "naive" contextual embeddings with valuable information from surrounding words.

The core mechanism of the transformer architecture—known as *self-attention* (Vaswani et al., 2017)—updates the contextual embeddings. The self-attention mechanism determines which of the surrounding words the transformer should give more weight when updating a word's contextual embedding. That is, the attention mechanism determines which words the model should pay more attention to when considering a word's context. Transformers tend to outperform alternative deep neural network architectures, such as recurrent neural networks (RNN; Elman, 1990) and long-term, short-term memory (LSTM; Hochreiter & Schmidhuber, 1997),[4] in part due to their attention mechanism (Adhikari et al., 2019; Hendrycks et al., 2020; Peters et al., 2019). Instead of encoding information about every word appearing before the target word, a transformer's attention mechanism focuses on the surrounding words that are most important. Consequently, transformers produce more effective contextual embeddings and process text more efficiently (Azunre, 2021; Pilehvar Camacho-Collados, 2020).

Due to their ability to produce contextual embeddings, NLP practitioners may use transformers for a wide variety of NLP tasks, such as text translation, keyword extraction, and text generation. In text classification, the BERT (Devlin et al., 2019) and models based on BERT (e.g., decoding-enhanced BERT with disentangled attention [DeBERTa]; He et al., 2021; robustly optimized BERT approach [RoBERTa]; Liu et al., 2019) are among the most popular. The BERT family of transformers adds a classification layer to the base transformer architecture that serves as the classification algorithm (as shown in Figure 1, a classification model requires both a text representation method and a classification algorithm). BERT and related transformers are considered state-of-the-art due to their effectiveness in representing language in addition to their capacity for *transfer learning* and *fine-tuning* (Sun et al., 2020; Wolf et al., 2020; Zhang et al., 2021). The following section expands upon transfer learning and fine-tuning in text classification. After, we elaborate on how these developments can serve as strengths when using BERT models to classify Big Five personality items.

Transformers: Pre-Training and Transfer Learning. As demonstrated in Figure 1, the traditional approach to text classification seeks to train a model to classify particular documents to particular classes (e.g., classifying personality items to one of the Big Five traits). So, researchers start this process by collecting text documents of a particular kind (e.g., Kobayashi et al., 2018a; Pandey & Pandey, 2019). While this approach to training a text classification model is popular today, it has limitations. Notably, the traditional approach to training limits those unable to collect enough text documents in domains where data is scarce; it produces models that struggle to process text that differs from the text seen during training (i.e., text with *out-of-vocabulary* words); also, this approach results in a model that researchers cannot use for other purposes (Azunre, 2021).

Considering these limitations, NLP researchers began to move away from the traditional approach to training (i.e., training classification models to solve a particular task). Instead, researchers have begun to embrace a "learning to learn" approach to training (Azunre, 2021, p. 16). This recent method is known as *transfer learning*—a process that instead *pre-trains* a model to gain a general representation of language (see Figure 2). During the pre-training phase, a model develops rich pre-trained embeddings (of words and symbols) by performing a variety of prediction task(s)—known as *source tasks*.[5] Thus, instead of training a model that starts without an understanding of word relationships, researchers can start with a model with a general representation of language in the form of pre-trained embeddings. After pre-training, researchers perform the *adaptation* step, where they adjust a pre-trained model to effectively perform their particular task—known as the target task (Pan & Yang, 2010). Next, we describe the ways to adapt a pre-trained transformer for performing text classification.

Adapting Transformers: Fine-Tuning and Feature Extraction. There are two general approaches to adapting a pre-trained transformer: (a) *feature extraction* and (b) *fine-tuning* (Peters et al., 2019). When using a pre-trained transformer for feature extraction, researchers use pre-trained embeddings as is. Illustrated in Figure 3A, this approach inputs unlabeled text documents to a pre-trained model and outputs "fixed" embeddings for each document. This results in a matrix of $N \times K$ where the number of rows (N) is equal to the number of documents, and the number of columns (K) is equal to the length of the embeddings (e.g., 512 or 768). With the addition of label information, this matrix can be input into an external machine-learning classifier to perform text classification. Moreover, specific transformer models used for feature extraction (e.g., Universal Sentence Encoder [USE] and SBERT) produce "fixed" embeddings. These specific models will always output the same embedding for a text document—regardless of additional factors. For example, a model like SBERT will always output the same embedding for the item "I am the life of the party" regardless of the additional items processed. One notable benefit of this approach is that researchers do not need to provide the model with labeled documents to produce embeddings. However, researchers must add label information later in the process (when training the classification algorithm).

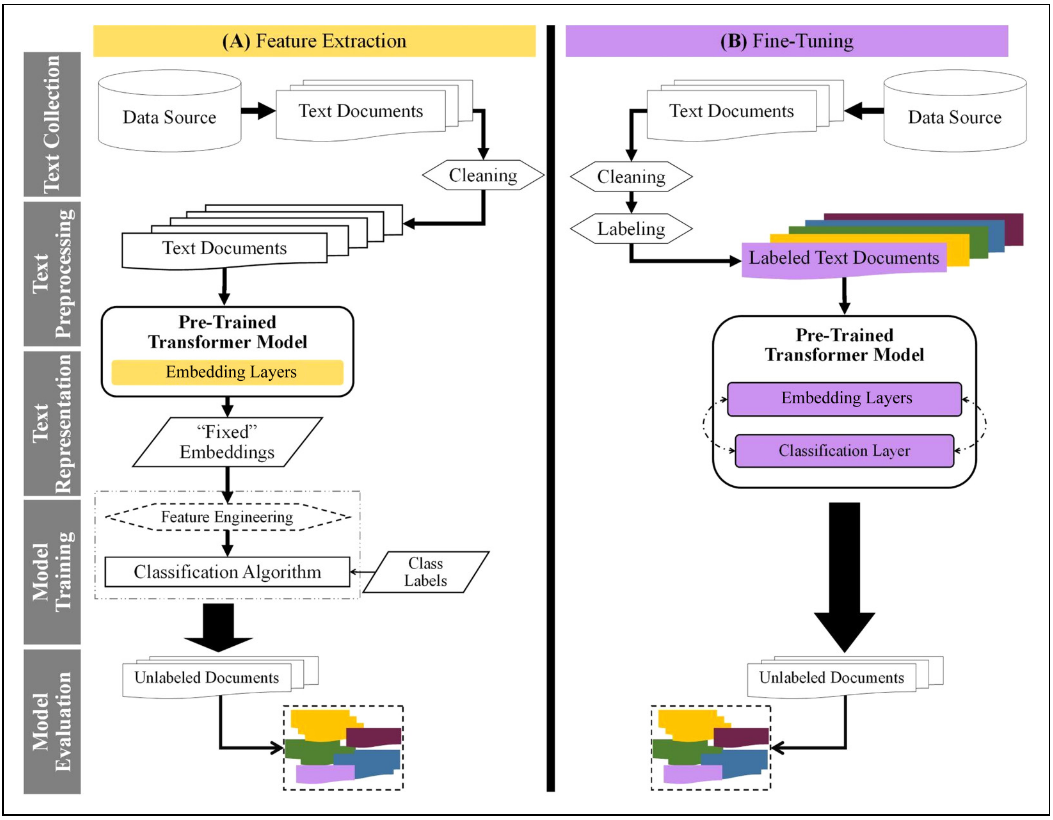

Figure 3. Using pre-trained transformers for text classification via feature extraction and fine-tuning.

The other approach to adapting a pre-trained transformer involves fine-tuning a pre-trained transformer model. The fine-tuning process (Figure 3B) adapts the pre-trained model to perform the specific classification task better; it does so by training the model using a sample (i.e., training set) of text documents with class labels (Sun et al., 2020). Unlike the feature extraction approach, fine-tuning updates pre-trained embeddings based on the training set and class labels. As such, embeddings are not "fixed." For example, the personality item "I am the life of the party" with the class label "extraversion" would receive a different embedding from a transformer model fine-tuned to classify items as "extraversion" or "introversion" versus a transformer model fine-tuned to classify items into one of the Big Five traits (i.e., "agreeableness," "conscientiousness," "extraversion," "neuroticism," and "openness").

While researchers consider the fine-tuning approach to be the state-of-the-art approach to text classification (He et al., 2021; Ruder, 2021; Sun et al., 2020; Zhang et al., 2021), the present research examines several types of transformer models using both approaches to adaptation (i.e., feature extraction and fine-tuning). Specifically, we evaluate two transformers developed for feature extraction: USE (Cer et al., 2018) and SBERT (Reimers & Gurevych, 2019). We also evaluate five fine-tuned transformer models: (i) ALBERT ("A Lite" version of BERT; Lan et al., 2020), (ii) BERT (Devlin et al., 2019), (iii) decoding-enhanced BERT with disentangled attention (DeBERTa; He et al., 2021), (iv) robustly optimized BERT approach (RoBERTa; Liu et al., 2019), and (v) XLNet (Generalized Autoregressive Transformer; Yang et al., 2020). Overall, these models are based on the transformer architecture and have been pre-trained on enormous amounts of data.

The justification for demonstrating several approaches was two-fold. First, although using a transformer for feature extraction is advantageous in some cases (e.g., document clustering tasks with unlabeled documents), it creates more steps in the overall text classification process. For example, after extracting the fixed-embedding matrix, researchers must append class labels, select a classification algorithm, format the fixed-embedding data, and determine which software package(s) to leverage when training the model. We wanted to illustrate this added complexity in text classification (see Figure 3). Second, since the value of a classification model is relative to its alternatives (Kobayashi et al., 2018a), and feature extraction approaches commonly used (Peters et al., 2019), we thought it would be important to examine transformer-based classification models that use feature extraction in addition to classification models that have been fine-tuned.

Advantages of Transformers in Text Classification

Using pre-trained transformers for text classification of personality items has several advantages. First, transformers better account for contextual factors compared to early text representation methods (Pilehvar & Camacho-Collados, 2020). This advantage is important for classifying personality items that often have very few distinct words and many words in common (e.g., Goldberg, 1993; Saucier, 1997). Second, the pre-training process exposes transformers to vast amounts of unclean text during pre-training, allowing them to simplify or eliminate the need for text cleaning and preprocessing (Devlin et al., 2019; Miyajiwala et al., 2022). Third, related to the former advantage, fine-tunable transformers combine text representation and classification into a single model, reducing the number of steps in the overall classification process and providing a more reproducible and efficient approach (Peters et al., 2019). Fourth, fine-tuning transformer models allows them to produce accurate results with less training data (e.g., Brown et al., 2020; Halder et al., 2020) and provides feedback to prioritize relevant features for each class, increasing the likelihood that the model is grounded in theory. Finally, while researchers have yet to apply transformers to classify personality items, pre-trained transformers have demonstrated impressive performance on classification tasks using documents of a similar nature (e.g., He et al., 2021; Lan et al., 2020; Yang et al., 2020). In addition to previously described advantages, Table 1 includes a description, relevant challenges, and recommendations for transformer-based approaches to text classification that aligns with each step in the text classification process (see Kobayashi et al., 2018a for an overview).

Demonstration: Training Transformers to Classify Personality Items

In this section, we apply several pre-trained transformers to classify Big Five items to demonstrate a novel and effective way to automate content analysis. First, we introduce each step by expanding on specific goals and challenges. Then, we describe our procedures while providing recommendations; a summarized version of this information is in Table 1. In addition to the demonstration described here, we have created a GitHub code repository—https://github.com/Shea-Fyffe/transforming-personality-scales. The repository provides several software tutorials, data, and other tools for training transformer-based text classification models. To reproduce the steps described below, see the files in the ~/tutorials directory of the code repository (we refer to these files as the "tutorials" throughout this manuscript).

Step 1: Collecting and Preprocessing Personality Items

To train an effective classification model for content analysis, researchers must collect scale items (i.e., text documents) and their corresponding dimensions (i.e., represented by labels). One challenge is identifying accessible data sources that provide structured data. While commonly used scales, such as those measuring the Big Five, are widely available, researchers may need to use other existing

Table 1. Advantages, Challenges, and Recommendations of Applying Transformer-Based Text Classification to Content Analyze Scale Items.

Classification Step[a]	Applied Description	Advantages	Challenges	Recommendations
Text collection	Collecting and labeling scale items to train a classification model either by collecting labeled items or writing new items and manually labeling such items.	Transformers demand fewer labeled items to train an effective classification model.	When the number of labeled items per class label is small (e.g., fewer than 40), models may underperform. If no labeled scale items are available (i.e., scale constructs are completely novel and scale items are brand new) some manual labor may be required.	A text classification task can be reformulated into a slightly different type of natural language processing task, such as *language modeling* or *natural language inference* (e.g., Schick & Schütze, 2021) to better align with a transformer's pre-training task. Researchers can also use a semisupervised approach where subject-matter experts give construct labels to new items using the classification predictions from a transformer model trained on a small set of labeled items (e.g., Chen et al., 2020). Or, researchers may extract fixed embeddings using a model like SBERT (Reimers & Gurevych, 2019), then perform document clustering setting the number of clusters equal to the number of intended scale dimensions (see Kobayashi et al., 2018b for document clustering overview). Researchers can use *conditioned text generation* (see Keskar et al., 2019) or data augmentation methods, such as synonym replacement (see Bayer et al., 2022), to generate "synthetic" labeled scale items. If no labeled items are available, we recommend manually labeling 5 to 10 items per dimension. The selected items should be broadly representative of each dimension, so limit selecting items with similar wording. When manually labeling items, researchers should ensure SMEs reach a level of agreement that aligns with research recommendations (see Krippendorff, 2018; Short et al., 2010).
Text preprocessing	Cleaning and standardizing scale item text.	Transformers drastically reduce the amount of text preprocessing that researchers must perform.	One must decide how much (if any) text preprocessing to perform, e.g., whether to prepend implied text (e.g., "I am …") or split long items.	Researchers should document preprocessing that occurs *prior* to input into the transformer model—we recommend creating a preprocessing software script or function (see `~/R` folder in our repository). In most cases, researchers should perform little-to-no text preprocessing with two notable exceptions: when critical "implied" parts of an item's text are missing (i.e., items omitting the subject "I"), and when items are longer than 300 words. Although rare, researchers can split long items into sentences then aggregate sentence-level classification results back to the item level.
Text representation	Applying an effective text representation method to convert items into numeric data that can be used by a classification algorithm.	Fine-tuning allows pre-trained embeddings to be adapted to a specific task. Transformers do not require text representations to be further manipulated (i.e., feature engineering).	Researchers choosing to use a transformer-based approach must choose between two specific applications—feature extraction or fine-tuning (see Figure 3).	Researchers should report the type of model selected and, when using feature extraction, describe any transformations made to the fixed-embedding data. When enough labeled items are available for fine-tuning (see section "Question 1: How Many Items are Needed for Fine-Tuning?"), we suggest fine-tuning a model using at least 70% of the available training data. After model evaluation, researchers may further train the model using the testing data.
Model selection and training	Choosing a specific transformer model architecture (e.g., BERT, DeBERTa, XLNet) that most effectively classifies scale	Since transformer architectures often have a classification layer in addition to embedding	Researchers must select among the numerous types of transformers. [b] Those unfamiliar with	Researchers should document the hyper-parameters used for training. We recommend reporting training procedures as a software script on a cloud-based platform (e.g., Colab or Azure). When picking a transformer model, researchers should consider literature on

(continued)

Table 1. (continued)

Classification Step[a]	Applied Description	Advantages	Challenges	Recommendations
	items. Also, specifying the type of classification problem and model training parameters (i.e., hyper-parameters) that results in optimal model performance.	layers, researchers can perform text classification using a single model (as opposed to requiring a text representation method and classification algorithm separately). NLP experts consider fine-tuned transformers the current state-of-the-art approach to text classification, oftentimes achieving human-level performance (Nangia & Bowman, 2019).	neural networks may need to learn novel terminology as well as become accustomed to potentially new training procedures. Researchers must overcome the hardware and computational requirements for model training/fine-tuning.	current state-of-the-art models, source task alignment, pre-training data alignment, and computational resources (see section "Model Selection" of "Step 3: Model Selection and Training" for more information). When selecting the type of text classification problem to conduct (e.g., multilabel or multiclass), researchers must theoretically ground whether their class labels represent orthogonal or non-orthogonal factors—where the former implies a multiclass problem and the latter and multilabel problem. For hyper-parameters, we recommend at least 10 epochs if there are less than 100 items per label. Also, researchers should set a conservative learning rate (between .000002 and .0001). Researchers may need more epochs to make up for lower learning rates. Batch size should be at least two times the number of classes but less than 32 (e.g., Liang et al., 2022; Sun et al., 2020). If able, use an online cloud-based environment like Google Colab (see our code repository). Commonly, issues arise related to the GPU. During training, GPUs may run out of space or "memory" further process text. If this occurs, we recommend trying the following steps (in order by priority): (1) truncating text, (2) decreasing batch size, and lastly (3) selecting a model with a smaller architecture.
Model evaluation	Determine if the transformer-based classification model is valid and useful for conducting content analysis.	After training, a transformer-based classification model is easy to save and apply to new (unlabeled) items. Model evaluation for transformers aligns with model evaluation of methods researchers may be more familiar with.	Researchers must thoroughly evaluate model validity compared to alternative approaches to content analysis. Researchers must estimate the utility of a model when implemented in their scale development pipeline.	Like Kobayashi et al. (2018a), we suggest weighing the pros and cons of a model's performance when compared to human raters. Additionally, researchers should account for factors such as time, cost, and the implications of type I and type II errors. To evaluate practical utility, we recommend that researchers and practitioners conduct their own empirical investigations, like those outlined in the latter part of the section "Important Questions When Using Transformers for Classifying Personality Items." Researchers should often re-evaluate trained models based on the influx of newly developed items. After newly developed items have been given a label that is considered valid, the items can be used for additional fine-tuning.

[a] When compared to the steps described by Kobayashi et al. (2018a), the dimensionality reduction step is not common when applying transformers to text classification (e.g., Liang et al., 2022; Mirończuk & Protasiewicz, 2018; Sun et al., 2020; Zhang et al., 2021); so, we omit it as a step here but provide a relevant advantage in the Text representation step. Also, we include an additional step (i.e., "Text collection") that is often considered a substep in "Text preprocessing" (e.g., Kobayashi et al., 2018a). We split "Text collection" into an additional step to provide information that is both comprehensive yet structured in a way that is easier to navigate.

[b] See https://huggingface.co/models for the complete repository of usable transformer models.

BERT= Bidirectional Encoder Representations from Transformers; DeBERTa= decoding-enhanced BERT with disentangled attention; GPU= graphics processing unit; SBERT= Sentence-BERT; SME=subject matter expert.

scales to train models for other constructs. However, the biggest challenge is determining how many labeled scale items are needed for the classification model to be accurate. Given its criticality, we elaborate on this challenge in the latter part of this article. Specifically, we provide suggestions on how to proceed in the section "Question 1: How Many Items are Needed for Fine-Tuning?" and Table 1.

Additionally, researchers should evaluate the integrity of the *labeling* process (i.e., the process that determines which items belong to which labels (e.g., Mirończuk & Protasiewicz, 2018). If mislabeled, scale items are likely to hurt model performance (e.g., Chen et al., 2022; Phang et al., 2019; Saarikoski et al., 2015; Schick & Schütze, 2021). In the same vein, researchers may be motivated to include items that are indirectly related to the dimension labels of interest to obtain a larger number of items for training (e.g., collecting popular scales used in clinical psychology and labeling them as "neuroticism" items or collecting "extraversion" items from leadership scales). Instead, we recommend researchers focus on collecting scale items directly related to the dimensions under investigation; striving for items with a wide variety of wordings (e.g., negatively worded items, long items, short items, contextual personality items, and items with non-first-person subjects) since variation in training data has been shown to improve model performance (e.g., Chronopoulou et al., 2019; Wang et al., 2021; Yin et al., 2020).

Step 1: Our Demonstration. To collect personality items, we leveraged the open-source repositories: *International Personality Item Pool* (IPIP; Goldberg et al., 2006) and the *Synthetic Aperture Personality Assessment* project (SAPA; Condon, 2018). We focused on items that explicitly measured Big Five personality traits. We filtered out duplicate items by converting items to lowercase and then removing non-letter characters (e.g., whitespace, commas, apostrophes, and hyphens). We mapped Big Five items to the labels: (1) agreeableness, (2) conscientiousness, (3) extraversion, (4) neuroticism, and (5) openness. Table 2 presents scale-level information for the unique 852 items.

Since we utilized pre-trained models, we performed minimal text preprocessing on the 852 items. We considered the recommendations by Hickman et al. (2020) to preprocess the items. Specifically, the steps taken were: (1) add an explicit first-person subject (e.g., *I*) to each personality item; (2) expand contractions (e.g., *I'm, don't, that's*) to represent their element words (e.g., *I am, do not, that is*); (3) convert number symbols to word representations (e.g., "4" was transformed into

Table 2. Total Items Collect by Scale and Source.

Source/Scale	Items[a]
International Personality Item Pool (IPIP)	
Abridged Big Five Circumplex (AB5C)	441
Big Five Aspects Scales (BFAS)	26
Big Five Inventory (BFI)	60
HEXACO Personality Inventory (HEXACO)	88
Interpersonal Circumplex (IPIP-IPC)	2
NEO Personality Inventory Revised (NEO-PI-R)	111
Seven Factor Scales (7FACTOR)	50
Six Factor Personality Questionnaire (6FPQ)	16
Synthetic Aperture Personality Assessment (SAPA)	
SAPA Personality Inventory (SPI)	58

Note. Raw data from the international item pool can be found by visiting the link: https://ipip.ori.org/ItemAssignmentTable.htm. Raw data can be accessed by visiting the link: https://doi.org/10.7910/DVN/T1NQ4V (Condon, 2019).

[a]Item counts represent unique items after removing duplicates.

Table 3. Training and Testing Set Item Counts by Class Label.

Class Label	Training	Testing[a]
Agreeableness	152	25
Conscientiousness	153	25
Extraversion	158	23
Neuroticism	130	21
Openness	140	25
Total	733	119

Note. $N = 852$.
[a]The testing sets totaled 119 items (14.0% of the overall items). This number was generated through stratified sampling based on 15% of the total sample, while considering rater fatigue.

"four"); (4) convert words in items to lowercase, then capitalize the first letter of each item along with any occurrences of the word "I"; finally (5) append a period to each item. Since negative wording impacts the interpretation of personality items (DiStefano & Motl, 2009), we did not perform negation handling (see Hickman et al., 2020); we were concerned this may give a machine-learning model an advantage during text classification. We split pre-processed items into a training and testing set using a "Train/Test Split" approach (Vabalas et al., 2019), which resulted in 733 and 119 items for training and testing, respectively (see Table 3).

Step 2: Text Representation of Personality Items

Recalling from Figure 3, there are two ways to produce text representations when using transformers. There are several reasons a researcher might take a feature extraction approach to text classification. These reasons include a motivation to use a more interpretable model that requires less computational resources, wanting to perform additional steps related to feature engineering, or the desire to use a more complex classification algorithm. On the other hand, researchers may fine-tune a pre-trained model given that they have collected enough labeled scale items and want to achieve the best possible performance. Although fine-tuning often involves selecting a transformer architecture that is slightly more complex, transformers simplify the overall classification process by combing text representation and classification into one.

To process text, transformers require raw documents to undergo *tokenization*. This splits each item's text into its component tokens.[6] Then, the model converts each token to a numeric index corresponding to that token's line in the model's pre-trained vocabulary file. For instance, the BERT tokenizer will tokenize "I rarely feel depressed" as: [1045, 6524, 2514, 14777]; so, "I" is the 1045th token in the BERT vocabulary file. The model assigns each token a pre-trained contextual embedding that will be updated or "fine-tuned" during training. By converting tokens to a numeric index, the model can process and update embeddings more efficiently. Depending on the NLP task, tokenization may result in adding several special tokens to each text document. In a text classification task, for instance, tokenizers prepend a "[CLS]" token with the id [101]. This token serves as the composite for the token-level contextual embeddings and is used as the overall (document-level) representation during text classification (Devlin et al., 2019).

Step 2: Our Demonstration. To compare fine-tuned classification models to classification models trained with embeddings derived from feature extraction, we trained 11 separate models—six models using feature extraction and five fine-tuned models. Four out of the six feature extraction models used "fixed" embeddings produced by transformers—the USE (Cer et al., 2018) and SBERT (Reimers & Gurevych, 2019). Researchers developed USE and SBERT to produce fixed-length contextual embeddings that researchers can use for various NLP tasks. Additionally, we chose to include mean-aggregated word embedding to serve as a baseline because of their effectiveness in text classification tasks (e.g., De Boom et al., 2016; Rudkowsky et al., 2018) as well as usage in organizational and psychological research (e.g., Speer, 2021). The "*Classification with Fixed Embeddings*" tutorial (see GitHub repository) describes the software procedures used to extract fixed embeddings for the 852 items.

To demonstrate the fine-tuning process, we selected five different pre-trained transformer models (i) ALBERT (Lan et al., 2020), (ii) BERT (Devlin et al., 2019), (iii) DeBERTa (He et al., 2021), (iv) RoBERTa (Liu et al., 2019), and (v) XLNet (Generalized Autoregressive Transformer; Yang et al., 2020). These models were fine-tuned in the "*Fine-Tuning Transformer for Text Classification of Big Five Items*" tutorial (see project's GitHub repository) using the popular Python library *transformers* (Wolf et al., 2020).

Step 3: Model Selection and Training

Model Selection. Typically, prediction accuracy determines the validity of a classification model (Kobayashi et al., 2018a). Thus, researchers should aim for the model with the highest accuracy when selecting from the possibilities. Nonetheless, this information is likely unavailable at the time of model selection. Alternatively, researchers should examine the literature to determine the current state-of-the-art model for similar text classification tasks. Currently, DeBERTa (He et al., 2021) and RoBERTa (Liu et al., 2019) are among the current state-of-the-art text classification models (Wang et al., 2019). Also, researchers should recognize that different types of transformers may differ in terms of the pre-training source tasks, the size and depth of the neural network, and slight variations in tokenization and embedding (see Kalyan et al., 2021). Nonetheless, the factors determining model selection are nuanced and practical, such as the computational resources required and how easily a particular model can be integrated into the text classification pipeline. When choosing a model, researchers should consider how closely the model's source task aligns with the classification task, especially in cases where there are few labeled examples (Peng et al., 2020). They should also consider the computational resources needed for training. Those with access to more time and computing resources may benefit from choosing a larger model, such as DeBERTa or RoBERTa. Those with less access to computing resources should consider smaller "distilled" models or models like ALBERT (Lan et al., 2020) or DistilBERT (Sanh et al., 2020). Additionally, open-source repositories contain existing fine-tuned models for particular text classification tasks, such as sentiment analysis or emotion detection. These models may be particularly effective in situations where scale developers want to classify items measuring attitudes or emotions.

Model Training and Hyper-Parameters. Transformers (and deep learning models more broadly) introduce a set of novel hyper-parameters that may be unfamiliar to organizational researchers. Researchers and practitioners set hyper-parameters to control the learning process. Thus, we begin by providing a brief overview of essential hyper-parameters. Then, we move on to describe our training procedures. During training, a transformer updates the weights in its network in a series of *steps.* Formally the total number of steps (T) the model takes during training is a product of:

$$T = E \times \frac{N}{k} \tag{1}$$

In equation (1) E is the number of epochs, N is the total number of training examples, and k the batch size. The number of epochs (E) indicates the number of times a transformer model will process the whole training set. Within an epoch, the batch size (k) determines the number of training examples to process before taking a step. For example, for an epoch with 100 training examples (*N*), a batch size (*k*) of 20 would take the model 5 steps (i.e., 100 divided by 20) to complete the epoch. At each step, the model updates layer parameters by a learning rate, which determines the extent to which the model can adjust its parameters each time it processes a batch of training data—this affects how quickly the model learns. For example, a higher learning rate can make the model learn patterns in the training data faster, but it can also make the model less accurate if set too high. The latter situation may lead to *catastrophic forgetting*, whereby a transformer model overwrites knowledge gained from pre-training with knowledge gained from fine-tuning (see Goodfellow et al., 2015). Hence, this increases the likelihood that the model will underperform when classifying novel items or text that is unlike the data seen during training (Sun et al., 2020).

Oftentimes, researchers emphasize selecting hyper-parameters, such as the number of epochs, batch size, and learning rate (e.g., Adhikari et al., 2019; Zhang et al., 2021), on a case-by-case basis. Still, several general guidelines appear in the literature that help better inform how to set hyper-parameters—especially in cases where one has a small amount of training data (less than 100 examples per label). First, while the original BERT authors (i.e., Devlin et al., 2019) suggest using no more than four epochs, researchers should consider using more than 10 epochs when the training data are small (e.g., Adhikari et al., 2019; Zhang et al., 2021). Also, to avoid catastrophic forgetting and overfitting, researchers should use a learning rate between .000002 and .0001 in addition to a batch size smaller than 32, keeping in mind that setting a lower learning rate, such as .000002, may require increasing the number of epochs to produce optimal results (e.g., Liang et al., 2022; Sun et al., 2020).

Step 3: Our Demonstration. We trained two machine-learning classifiers, (a) a boosted decision-tree algorithm (XGB; *XGBoost*) and (b) and linear support vector machines (SVM),[7] for each of the three fixed-embedding methods (i.e., USE, SBERT, and word embeddings). We selected these classifiers given their success in text classification tasks (Kobayashi et al., 2018a) and their availability in open-source software. This resulted in six fixed-embedding classification models (i.e., 3 text representations × 2 classifiers) in addition to the five fine-tuned transformer models. The fixed-embedding models were trained using the *caret* package (Kuhn, 2021) in R, and the fine-tuned models using the *transformers* (Wolf et al., 2020) library in Python.

We sought to control confounding factors, such as hyper-parameters, bearing in mind that the goal of this illustration was to show how various models using fixed embeddings compared to fine-tuned transformers. For the fixed-embedding models, we initialized hyper-parameters at their respective defaults. When training the five fine-tuned models, we set a conservative base learning rate of .00002. Given that we trained the fine-tuned transformer models using a batch size of 16 for 10 epochs (e.g., Liang et al., 2022; Sun et al., 2020; Zhang et al., 2021), we felt it would be inappropriate to train the fixed-embedding models using all the training data. Thus, for the fixed-embedding models, we performed k-fold cross-validation (see Kobayashi et al., 2018a) by breaking the training data into seven smaller "folds." This technique allowed the selected classification algorithms to train in a way that better aligns with our fine-tuned models. For example, in both training scenarios, we divided the training data into multiple subsets ("folds" for the fixed-embedding models and "batches" for the fine-tuned models). This allows for a more robust estimate of performance that is less susceptible to the effects of randomness or noise in the data. For the fixed-embedding models, we used the version of the model with the highest accuracy out of the seven folds; for the fine-tuned transformers, we selected the version that was most accurate across the 10 epochs—these models were then saved and used for the final model evaluation on the scale items in the testing set.

Table 4. Ranked Model Performance When Classifying Big Five Personality Items by Trait Label.

Model	Accuracy [$\mathrm{CI}_{\mathrm{LL}}$, $\mathrm{CI}_{\mathrm{UL}}$]	Precision[a]	Recall[b]	F1-score[c]
Human raters[d]	.706 [.677, .735]	.71	.71	.71
Fine-tuned transformers				
ALBERT	.621 [.532, .704]	.63	.62	.62
BERT	.807 [.727, .868]	.81	.81	.81
DeBERTa	**.824** [.745, .882]	.83	.82	.82
RoBERTa	**.824** [.745, .882]	.82	.82	.82
XLNet	**.824** [.745, .882]	.83	.82	.82
Fixed-embedding (feature extraction) models				
Aggregate word embeddings-SVM	.548 [.467, .628]	.55	.55	.54
Aggregate word embeddings-XGB	.618 [.537, .693]	.64	.62	.63
Sentence-BERT embeddings-XGB	.680 [.600, .751]	.68	.68	.68
Sentence-USE embeddings-SVM	.688 [.608, .758]	.69	.69	.68
Sentence-BERT embeddings-SVM	.701 [.622, .770]	.70	.70	.70
Sentence-USE embeddings-XGB	.715 [.637, .783]	.72	.72	.71

Note. Accuracy estimates are rounded to the nearest hundredth for greater detail. Models presented in **bold** are beyond the upper limit of human baseline accuracy based on bootstrapped estimates, $p < .05$.

[abc] Macroweighted precision was calculated as $\sum_{c=1}^{C} \frac{f_c}{N} \cdot \frac{tp_c}{tp_c + fp_c}$ where tp_c is the number of true positives, fp_c the number of false positives, and f_c the total number of items for class c in the test set size N. Macroweighted recall (i.e., sensitivity) was calculated as $\sum_{c=1}^{C} \frac{f_c}{N} \cdot \frac{tp_c}{tp_c + fn_c}$ where fn_c is the number of false negatives. Macroweighted F1-score was calculated as $\sum_{c=1}^{C} \frac{f_c}{N} \cdot \frac{2 \times tp_c}{2 \times tp_c + fp_c + fn_c}$.

[d] Rater accuracy was calculated as the total number of correct ratings divided by the total number of ratings. Additionally, rater accuracy was re-calculated using direct consensus, where the most common label was used as the prediction for each item. This resulted in a slightly higher accuracy estimate (0.764).

ALBERT= "A Lite" version of BERT; BERT= Bidirectional Encoder Representations from Transformers; CI = accuracy confidence interval calculated based on Wilson (1927) interval; DeBERTa= decoding-enhanced BERT with disentangled attention; LL = lower limit; RoBERTa= robustly optimized BERT approach; SVM= support vector machine; UL = upper limit; USE= Universal Sentence Encoder; XGB=*XGBoost*.

Step 4: Model Evaluation

Researchers should select evaluation measures based on the type of classification problem (e.g., binary, multiclass, multilabel), nature of the data, and priorities of the researcher (Kobayashi et al., 2018a). Since we performed a multiclass classification problem (i.e., a classification task where the model classifies each item to belong to just one of the Big Five traits), we calculated metrics as a weighted average across classes (see Table 4). We focus on four metrics: accuracy, recall, precision, and F1-score. *Accuracy* is the proportion of correct predictions out of all predictions made. *Recall* (also referred to as *sensitivity*) is the proportion of correctly predicted known positives —this is a proxy for the "power" of the model. *Precision* is the proportion of correctly predicted positives over the total number of positive predictions. In other words, one minus the precision score is the model's likelihood of committing a type I error. *F1-score* is calculated as the harmonic mean of precision and recall, which considers the fact that precision and recall are on a different scale (i.e., both use true positives as their numerator but precision divides by *predicted* positives whereas recall divides by *actual* positives). Readers can think of the F1-score as the "effectiveness" of a model (Kowsari et al., 2019, p. 45). However, in isolation, these evaluation metrics give little information regarding a model's practical value (Kobayashi et al., 2018a). In what follows, we evaluate the performance of the models described here versus human raters when classifying Big Five personality items by their content.

Examining Transformer Model Performance

Comparing Model Performance with Human Raters

To conduct the manual content validation task, we recruited eight psychology graduate students to serve as SMEs. Six students specialized in Industrial and Organizational Psychology; the remaining students specialized in Human Factors or Developmental Psychology, respectively. Raters participated in a 3 to 4 hours training that began with an overview of the study, followed by an in-depth video lecture covering the Big Five personality traits. After the video lecture, raters practiced by classifying 30 randomly selected items from the training set. We required raters to repeat the training until they achieved an accuracy of 80%.

After completing training, the training instructed raters to independently (and honestly) classify the test items using an online survey. Rating procedures were fully crossed, meaning all raters rated all items in the testing set. This design is ideal for evaluations involving human raters (Putka et al., 2008). We used Fleiss' kappa to evaluate agreement among the eight raters on the testing-set items. Kappa values for the 119 items ($\kappa = 0.59$) were >0.40 showing acceptable interrater agreement (Fleiss, 1981).

Results of Model Performance

Table 4 shows the overall results for outcomes including accuracy, precision, recall, and F1-score. Rater accuracy ranged from 69% to 78% with raters being 71% accurate on average. In terms of overall accuracy, human raters outperformed most of the classification models trained using feature extraction and fixed document embeddings. However, four of the five fine-tuned transformer models produced higher accuracy than the human raters. As demonstrated in Table 4, three models—DeBERTa, RoBERTa, and XLNet—outperformed the average human rater beyond estimated confidence intervals. These findings align with a number of studies demonstrating that transformer models are comparable to (or better than) human raters on an assortment of NLP tasks (e.g., Alberti et al., 2019; Conneau & Kiela, 2018; Nangia & Bowman, 2019; Zellers et al., 2018). In addition, four of the five fine-tuned transformer models outperformed rater predictions when using a direct-consensus approach[8]—though not by a significant margin.

Important Questions When Using Transformers for Classifying Personality Items

While we provided several recommendations throughout our demonstration above (see Table 1), there are still several concerns that have not been fully addressed. We address them here in the form of questions researchers may have when applying transformer models to classify personality items. In many cases, we provide an empirical example to further elaborate on implications as well as recommendations for how to address such concerns.

Question 1: How Many Items are Needed for Fine-Tuning?

While researchers emphasize that transformer models should be fine-tuned to be most effective when classifying documents (e.g., Devlin et al., 2019; Sun et al., 2020), fine-tuning a model with a small number of training examples can lead to inconsistent or inferior performance (e.g., Phang et al., 2019; Zhang et al., 2021). The number of examples required to reach a point of "sufficiency" varies based on factors such as how well the classification task aligns with tasks performed during pre-training, model size and complexity, and the quality of the training data (e.g., Chronopoulou et al., 2019; Wang et al., 2021; Yin et al., 2020).

Accordingly, scholars in computer science and computational linguistics advocate for an ad hoc approach as opposed to providing clear guidelines (Ruder, 2017). Although researchers hesitate to define the minimum number of training examples recommended to fine-tune a model, the status quo seems to suggest that fewer than 32 examples per label (i.e., 32 items per dimension in our case) severely compromise model performance (e.g., Bansal et al., 2020; Halder et al., 2020; Schick & Schütze, 2021). In the literature, text classification with only a few labeled training examples is commonly referred to as "few-shot" learning, or in cases where no labeled data are provided, "zero-shot" learning (Ruder, 2017).

Researchers with an insufficient number of labeled examples have several different options. For example, freezing various layers of the transformer architecture (e.g., Chronopoulou et al., 2019), training with smaller learning rates (e.g., Howard & Ruder, 2018), and training the model multiple times with several different random seeds are options (e.g., Phang et al., 2019). However, better aligning the classification task with a source task performed by a transformer model during pre-training (see Figure 1) produces compelling results (e.g., Brown et al., 2020; Liu et al., 2019).

Recommendation. Instead of using the standard BERT family of transformer models for text classification, researchers can reframe a text classification task as a *language modeling* task—since many transformers are pre-trained using language modeling (e.g., GPT-3, BERT, and RoBERTa). In a language modeling task, given some *prompt* of existing words, a language model predicts the next word or words (i.e., the *completion*) in the sequence. By fine-tuning a transformer using

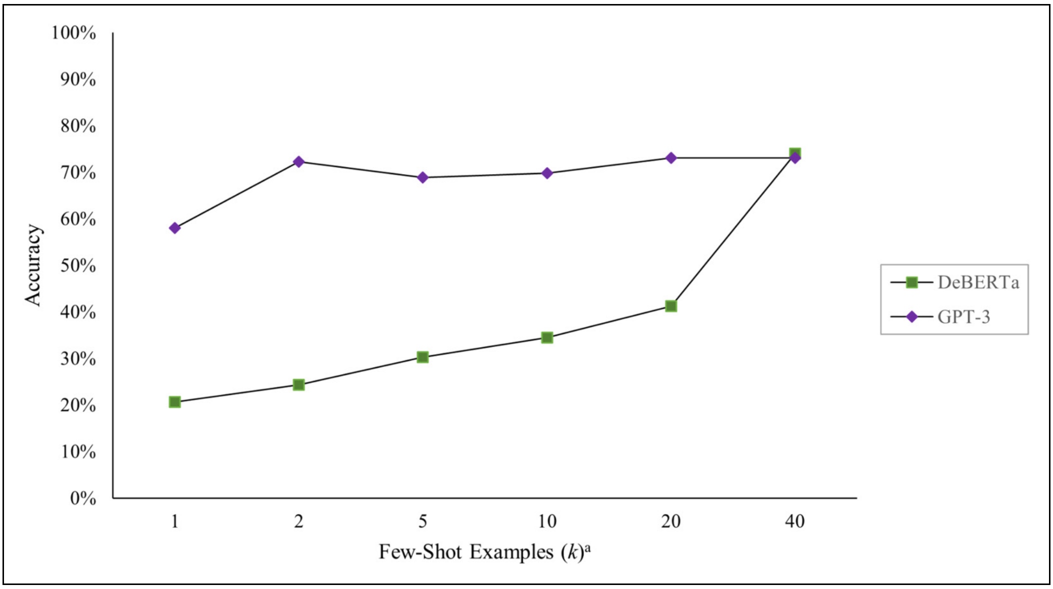

Figure 4. Few-shot classification accuracy by number of training examples.
Note: A comparison of classification accuracy on a testing set (N = 119) between a standard fine-tuned model (i.e., DeBERTa) and an autoregressive decoder transformer (i.e., GPT-3). When reframing a text classification task to better align with a transformer's source task (i.e., a task accomplished during pre-training), researchers can drastically increase model performance with fewer examples. [a] Values represent the number of randomly selected examples per label. Since our classification included five labels, when k = 5, for example, the total number of items used to train the model was equal to 25; when k = 2, there were 10 items used to train the models. DeBERTa= decoding-enhanced Bidirectional Encoder Representations from Transformers with disentangled attention.

personality items as prompts and their corresponding trait label as completions, a model learns to predict trait labels when given a novel item prompt. We demonstrate this in the tutorial: "*Few-Shot Learning with Transformers*" (see the ~/tutorials directory of project's code repository).

We encourage readers to access the tutorial for more information on the steps taken to produce the results displayed in Figure 4. When inspecting Figure 4, one can see the limits of a fine-tuning approach with fewer than 40-labeled examples per class. However, if the classification task is reformatted into a language modeling task, we drastically improve classification accuracy with even a handful of examples (e.g., Scao & Rush, 2021). While the few-shot learning model underperforms the fine-tuned models presented in Table 4, the model starts to perform on par with our human raters after just two-labeled examples per class ($N = 10$).

In the current research, we illustrate a "Train/Test Split" approach (Vabalas et al., 2019) to fine-tuning. This approach is quite popular in transformer-based text classification as well as text classification more broadly. Nonetheless, we primarily used the "test set" (which can be more accurately considered a "developmental set" because items were labeled) to evaluate model performance. Like others (e.g., Kobayashi et al., 2018a), we recommend rigorously evaluating a classification model prior to implementing it in practice. Still, researchers and practitioners should consider further fine-tuning a trained model using the testing set (if labels are present) after model evaluation.

Question 2: How to Classify Items Measuring Multiple Traits?

When predicting the class label of a new text document, classification models produce predictions in the form of logits. Depending on the type of classification task, sigmoid or softmax functions convert logit predictions to probabilities ranging between .00 and 1.00. The simplest type of text classification task is *binary classification* (Padhy et al., 2020). In binary classification, a model selects the most likely class among two possibilities. For instance, one could train a binary classification model to predict whether an item is a Big Five item or a non-Big Five item, or a binary classification model could be used to predict whether items measure cognitive or noncognitive constructs. Binary classification converts logit-valued predictions to probabilities via the sigmoid function.

In *multiclass classification*, a model selects the most likely class given three or more classes or categories. Here, a softmax function outputs probability values for each class. As such, classes are mutually exclusive, and probabilities sum to 1 (Kowsari et al., 2019). By conducting a multiclass classification task, for the purpose of content analysis, one implies that items can only belong to one class (analogous to the idea of simple structure in factor analysis).

In cases where evidence suggests that items may belong to multiple classes, one may perform a *multilabel classification*. One can consider this type of classification task to be k binary one-vs-all classifications, where k is the number of classes (Padhy et al., 2020). In this case, the models trained in this research would perform five iterations each time, treating items from one Big Five dimension as the positive class and combining items from all other dimensions into a negative class (Padhy et al., 2020). The sigmoid function converts predictions to probabilities where a value close to zero would indicate that the input is likely to belong to the "negative" class, and a value close to one would indicate that the input is likely to belong to the "positive" class. In multilabel classification, classification probabilities are not required to sum to one. As a result, items may belong to multiple traits or factors.

Before describing recommendations, we provide an applied example to clarify the distinction between multiclass and multilabel classification. After training a multilabel text classification model to predict the class of Big Five personality items, the predicted probabilities of the class "neuroticism," for example, would be interpreted as "the probability that the classification model thinks an item is related to neuroticism after taking the other Big Five Factors into account." In contrast, the predicted probabilities of the class "neuroticism" produced by a multiclass classification would be interpreted as, “the probability that the classification model thinks an item belongs to neuroticism when compared to the other Big Five factors.” While the distinction between multiclass and multilabel classification appears subtle, these differences have several practical and conceptual implications. In the following sections, we provide recommendations for how to navigate such implications.

Recommendation. A multiclass text classification assumes that each Big Five item corresponds to a single trait; an assumption often made by researchers (e.g., Marsh et al., 2010). However, we acknowledge that personality items can measure multiple traits, as researchers have frequently discussed these “blended items” (e.g., Goldberg & Velicer, 2006; Hofstee et al., 1992; Schwaba et al., 2020). For example, the item “I feel comfortable with myself” tends to negatively load on neuroticism and positively load on extraversion (DeGeest & Schmidt, 2015). Thus, we provide an option to perform multilabel classification in the “*Fine-tuning Transformer Models for Text Classification of Big Five Items*” tutorial. As a reminder, multilabel classification performs a binary one-vs-all classification for each class—iteratively treating the one class as the positive class and combining all other classes into a negative class (Padhy et al., 2020); multilabel classification expects the negative class to adequately compliment the positive class, or sample everything that is *not* the positive class. Accordingly, if the negative class is not exhaustive, researchers should be careful when communicating results. For example, if one were to train a multilabel classification model using the training data presented in this research; then, produce a “neuroticism” class probability for an item, that probability should be interpreted as “the model’s confidence that the item relates to neuroticism after taking the other Big Five traits into account,” and not “the model’s confidence that the item relates to neuroticism or something else.” This more conservative interpretation results from the negative class not being exhaustive. In the next question, we elaborate on how this problem relates to measuring items outside of the Big Five.

Question 3: How to Account for “Other” (Non-Big Five) Items?

A prominent issue is the consideration and treatment of negative classes outside of, or adjacent to, the focal one(s). Notably, researchers may have items representing these “Other” classes (i.e., known “Others”) or lack examples of “Other” (i.e., unknown “Others”). A common approach to identifying “Other” examples (i.e., items) is to use multilabel classification (Padhy et al., 2020; Roady et al., 2020). However, this approach assumes items representing the “Other” class are exhaustive and accurately represented in the training data (Geng et al., 2021; Hendrycks et al., 2020). Alternatively, this approach could be simplified, and researchers could perform a binary classification task which combines items representing focal classes (i.e., Big Five dimensions) into one “Big Five” class and items representing nonfocal classes (i.e., non-Big Five dimensions) into a single “Other” class.

In all cases, the central obstacle becomes how to best determine the items that represent “Other” especially when considering the countless variations of items that are possible. For instance, “Other” items could be items that measure a personality trait beyond the Big Five, items that measure nonpersonality constructs (e.g., values, health, mental ability), or even items that have a different grammatical structure than a standard personality item (e.g., other-report items, questions). Moreover, researchers must determine how large the “Other” class should be relative to the “Big Five” class. Below, we discuss two situations researchers may encounter when determining what content should make up the “Other” class. In one situation, researchers know the content making up the “Other” (e.g., items tapping known traits outside the Big Five). Alternatively, researchers may be in a situation where they are not aware of what other constructs (and thus items) will constitute the “Other” class(es).

When "Other" Traits are Known. When translated to a text classification task, items measuring undesired constructs belong to a class we generically refer to as "Other." Accounting for "Other" items would help control the type I errors (see Meehl & Rosen, 1955) as it would prevent the model from assuming all input documents have content belonging to a focal class. Personality items outside the Big Five would constitute "Other." Still, there remains considerable debate about the generality of the model, and whether statistically adjacent traits could be encompassed by (Costa & McCrae, 1995; John et al., 2008; Saucier& Goldberg, 1998) or lie outside the Big Five (e.g., Ashton et al., 2004; Block, 2010; Paunonen & Jackson, 2000). Nonetheless, determining a baseline estimate for the proportion of items outside of the Big Five would be valuable for several reasons. First, this estimate ensures that the relative proportions of labeled items (when training the preliminary model described in the following section) are representative of the overall population. Second, a baseline estimate may help researchers better establish a threshold or cutoff for classification probabilities. For example, when compared to a baseline estimate of 15%, a probability threshold resulting in 50% of items being flagged as "Other" should signal to researchers that their "Other" class is poorly constructed or that their threshold should be increased.

To estimate the proportion of scales "outside" of the Big Five, we drew from several empirical and review studies (i.e., Bainbridge et al., 2022; Paunonen & Jackson, 2000; Saucier& Goldberg, 1998; Schwaba et al., 2020). These studies provided estimates of the proportion of items outside of the Big Five using various metrics. Table 5 provides a summary of baseline estimates. According to these results, past research suggests that a vast portion of personality items will fall within the Big Five framework. Nonetheless, these studies are only useful in cases where the items presented to the model are, in fact, personality items and not items measuring nonpersonality constructs. Below, we provide recommendations for addressing both cases, or cases in which a model will only process personality items and cases in which a model may process personality and nonpersonality items.

Recommendation. First, we recommend that researchers review past content analyses (if available), poorly performing (e.g., reworked or retired) scale items, and the literature to identify items to include in the "Other" class. The models we present—for example—assume that input documents will be Big Five personality items. If implemented in practice, these models overlook the possibility for item writers to generate items that measure non-Big Five traits. Thus, we would review the literature to identify items measuring "Other" dimensions (i.e., constructs outside the Big Five). Researchers may also want to ensure that the "Other" class contains items of inferior quality (e.g., misspelled or grammatically incorrect). We provide examples in Table 6.

If enough known "Other" examples exist, we suggest a multistage approach whereby researchers use a binary classification transformer model first to identify if items are related to the Big Five (or their focal classes of interest). This model would help distinguish desirable (Big Five) items from

Table 5. Percentage of Personality Scales Measuring Non-Big Five Traits.

Study	Method of Estimation	Estimate [CI_{LL}, CI_{UL}]
Paunonen & Jackson (2000)	Scale Variance Account For	.038 [.011, .130] [a]
Saucier & Goldberg (1998)	Scale Variance Account For	.000 [.000, .069] [a]
Schwaba et al. (2020)	Network Analysis	.012
Bainbridge et al. (2022)	Facet Ranking Regression	.154 [.080, .275] [a]

Note. $N = 852$.

[a] Confidence intervals taken from Bainbridge et al. (2022; see Supplemental Material A).

CI = Accuracy confidence interval calculated based on Wilson (1927) Interval; LL = lower limit; UL = upper limit.

Table 6. Example Items Representing the "Other" Class.

Item	Item Source Scale [a]
I feel thankful for what I have received in life.	Values in Action Inventory of Strengths
I like to exaggerate my troubles.	Values in Action Inventory of Strengths
I misuse power.	Temperament and Character Inventory
I take in stray animals.	Temperament and Character Inventory
I need things to be arranged in a particular order.	Obsessive-Compulsive Inventory
I believe that unfortunate events occur because of bad luck.	Powerful Others and Chance
Not inventive ideas original telling not have is.	Denatured item
I do forget restless frequently things.	Denatured item

Note. Items like those above could be used as the "Other" class when training a binary classification model to predict a positive class ("Big Five" item) versus a negative class ("Other" items). This model could be applied to flag a variety of issues in scale development (e.g., poorly written items and items measuring cognitive constructs) given representative examples are provided in the "Other" class.

[a]Examples taken from: Powerful Others and Chance scale (Levenson, 1981), Obsessive-Compulsive Inventory (Foa et al., 2002), Temperament and Character Inventory (Cloninger et al., 1993), and Values in Action Inventory of Strengths (Peterson & Seligman, 2004). We generated "denatured items" by randomly shuffling the words in a Big Five item, then swapping out a word for a random word among all words used across items.

undesirable ("Other") items. Next, a multiclass model—like those described in the demonstration—would be applied to evaluate items that have not been flagged by the first model. We recommend this two-step approach as opposed to training a single multiclass model with an additional "Other" class. It is important to remember that researchers should use class labels to organize and categorize text documents in a way that is meaningful and relevant to a particular task or objective. In a multiclass problem, the "Other" class would not represent a significant collection of related documents. Researchers should stray away from creating classes that are arbitrary and misalign with the various classes present during classification. In treating the Big Five dimensions as labels, for example, we signify that each label represents a personality trait; "Other" is not a personality trait, so it misaligns with a multiclass problem. After using an initial binary classification model to determine if an item is likely related to the Big Five or not related to the Big Five (i.e., "Other"), those items that are "related" to the Big Five (based on a classification probability threshold determined by the researcher) can be used as input for the model described in our demonstration.[9]

When "Other" Traits are Unknown. In some cases, the content of "Other" may be *unknown* either completely or partially. While researchers in machine-learning call this the "open-set problem" (see Roady et al., 2020), in areas related to psychological measurement it relates to the second component of content validity—content representativeness—or the extent to which a measure captures the construct(s) of interest (Colquitt et al., 2019). Commonly, SMEs estimate content representativeness (Haynes et al., 1995); still, it is unlikely that SME judgment is fully sufficient, given the limitations of human cognition (Shrestha et al., 2021). In other words, it is difficult, if not impossible, for SMEs to consider every item that could exist outside of (and within) the scope of the Big Five. Hence, Allport (1937) famously described the scope of personality content as "a semantic nightmare that could keep psychologists at work for a lifetime" (pp. 353–354). Even so, transformers could play a role in making this problem more tractable. We elaborate below.

Recommendation. Researchers posit that language is the source of personality content (John et al., 1988; Saucier & Goldberg, 1998). Accordingly, novel *generative* transformers (e.g., *GPT-3*; Brown et al., 2020) may help better address this problem. Generative decoder-based transformers, like GPT-3, differ from encoder-based transformers (e.g., BERT and DeBERTa) in how they are pre-trained. When pre-training, decoder-based transformers attempt to accurately predict the next word in a sentence or phrase by utilizing only the words that appear before the target word (Radford et al., 2018). In other words, generative transformers attempt to predict the next word in a sequence of text while only seeing words that occur before that word; however, models like BERT and DeBERTa can see both words before and after the target word. As a result, generative decoder-based transformers excel at text-generation tasks. When applied to scale development, researchers could generate *adversarial* data to improve the precision of classification models (see de Rosa & Papa, 2021 for an overview). Adversarial examples are "fake" examples (produced by a generative model) that help a classification model better discriminate between ideal and non-ideal cases (e.g., Croce et al., 2020). Researchers could leverage this strategy to train classification models to be robust to items that cross-load, measure "Other" things, or that lack the appropriate grammatical structure.

Question 4: Does Text Classification Relate to Established Content Validation Approaches?

Researchers have already established approaches to content validation that can help scale developers identify potentially problematic scale items (e.g., Anderson & Gerbing, 1991; Hinkin & Tracey, 1999). These approaches recruit non-expert raters to verify which of the intended constructs (if any) each item belongs (i.e., a manual text classification task). Anderson and Gerbing's (1991) approach, for example, produces two metrics, *substantive agreement* (p_{sa}) and *substantive validity* (c_{sv}) for each item rated (see Colquitt et al., 2019, p. 1244, for p_{sa} and c_{sv} formulas). Conceptually like a model's classification probability, the substantive-agreement coefficient represents the proportion of raters who indicate the item measures the intended construct. The substantive-validity coefficient represents the degree to which raters indicate that an item measures "its [intended] construct more than any other construct" (Anderson & Gerbing, 1991, p. 734). Both metrics relate to important scale characteristics, such as reliability (Colquitt et al., 2019); however, there is insufficient research on the relationship between substantive agreement, substantive validity, and classification probabilities. Here, we present an empirical illustration in hopes of how transformer-based text classification methods may relate to well-established, manual approaches to content validation.

For the 119 items in our testing set, we calculated substantive agreement (p_{sa}) and substantive validity (c_{sv}) using classifications from our eight human raters. Overall, the average p_{sa} equaled .707 ($SE = 0.031$) and c_{sv} equaled .627 ($SE = 0.034$). Although the testing set consisted of randomly selected items from different Big Five scales (see Table 2), these averages were adequate when compared to item sets taken from single scales (Colquitt et al., 2019); this suggests that our test set has content that is highly relevant to the Big Five personality traits. Then, we compared the classification probabilities produced by the fine-tuned DeBERTa model to p_{sa} and c_{sv} values. There was a strong relationship between the classification probability of an item (produced by the DeBERTa model trained in the tutorial) and substantive agreement and substantive validity values derived from human raters. The correlation between substantive agreement and classification probability [$r(117) = .575, p < .001$], was just slightly higher than the correlation between substantive validity and classification probability [$r(117) = .565, p < .001$].

In addition, we separated items into ordered categories based on the numeric p_{sa} and c_{sv} values (see Table 5 in Colquitt et al., 2019), where higher categories indicate stronger levels of content validity. We present these results in Figure 5 below. As shown by (A) in Figure 5, items in the lowest substantive agreement category had an average classification probability of .480 ($SE = .091$); whereas the DeBERTa model was highly confident when predicting items in the highest or "very strong" substantive agreement category ($M = .992$, $SE = .005$). For items with a substantive validity score lower than .04, the DeBERTa model was unsure of its classifications ($M = .402$, $SE = .136$);

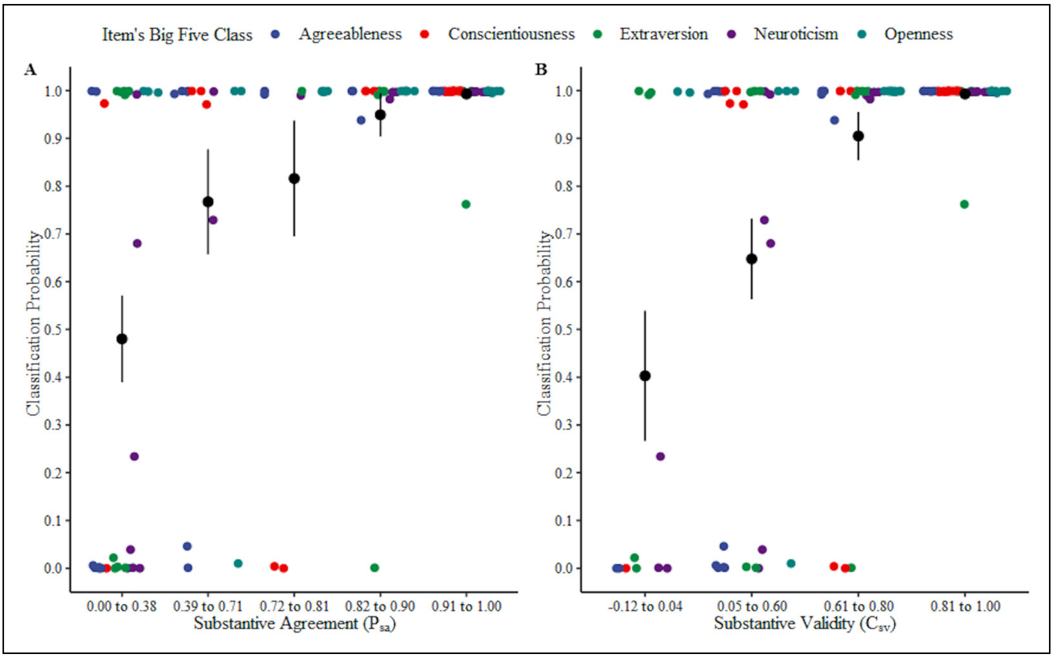

Figure 5. Classification probability by rater's levels of substantive agreement and substantive validity.

however, the DeBERTa classification model was highly confident regarding the predicted class for items with a $c_{sv} > .80$ ($M = .992$, $SE = .005$).

There are several notable trends in Figure 5. When DeBERTa was not confident about the predicted class of an item (i.e., classification probability < .50), that item was likely to have a poor substantive agreement and validity. Moreover, if we removed items with a classification probability of below .80 from the scale, we would lose only one item with "very strong" p_{sa} and c_{sv}. However, there were a number of type II errors based on the p_{sa} and c_{sv}. In other words, using classification probability thresholds does not guarantee that the remaining scale items will have strong content validity. This finding is likely a result of DeBERTa's highly skewed classification probabilities (which were .97 on average for the test items)—keeping in mind that this model was more accurate than human raters—and classification probabilities were based on a multiclass classification problem. This finding may also relate to our sample of raters, as there is discussion about the level of subject matter expertise and number of raters to use when conducting content analysis (see Colquitt et al., 2019). Depending on the collection of items, number of raters, and level of expertise, p_{sa}, and c_{sv} estimates could have varied (e.g., average p_{sa} equaled .707, $SE = 0.031$, and average c_{sv} equaled .627, $SE = 0.034$). Our raters rated 23 items as "poor," yet the model gave the items a classification probability of above .80. While these items confused our raters, the model selected the correct label (as intended by the original scale developer) 100% of the time.

Recommendation. Our results suggest a strong relationship between substantive agreement, substantive validity, and an item's classification probability. In practice, implementing a transformer model (like the ones illustrated in this research) would help identify items with poor content validity indices. However, based on our results, such a model would also be confident about the content of an item that may confuse humans even if the model indicated a label that aligns with the scale developer's expectation. Our major recommendation is for future research to continue exploring the relationships between NLP-based indices and traditional psychometric indices. We find it interesting, for example, that *all* of the 23 items that our raters deemed as having weak content validity were predicted correctly by the model. While this may be due to our sample of raters, this could suggest content areas that are particularly prone to misinterpretation by raters.

Question 5: Do the Proposed Techniques Have the Potential to Improve Factor Analysis?

Several researchers have attempted to relate well-established content analysis techniques to factor analysis (see Colquitt et al., 2019), but few have examined the relationship between NLP methods and factor analysis (e.g., Cutler & Condon, 2022). Still, NLP methods such as text classification have the potential to augment factor analysis. Researchers have highlighted how outcomes of factor analysis depend upon the initial set of items selected for response data collection (Ashton & Lee, 2005; Hopwood & Donnellan, 2010; Russell, 2002). Though, unlike factor analysis, text classification models do not require the collection of response data. As such, they give researchers the opportunity to make data-driven decisions when selecting an initial set of items for scale administration. These decisions could help prevent the administration of items that will perform poorly during factor analysis. Hence, artificial intelligence (AI) models have been leveraged to improve decision making in other domains (e.g., Vodrahalli et al., 2022).

To examine this potential, we investigated the relationship between the factor loadings and label classification probabilities. To calculate factor loadings, we leveraged the Eugene-Springfield Community Sample dataset (see Goldberg & Saucier, 2016). First, we subset the items overlapping between the response data and the testing set used in the content analysis experiment. This resulted in 100 items. We removed missing cases from the Eugene-Springfield Community Sample dataset, leaving 461 responses for the analysis. We conducted an exploratory factor analysis (EFA) in R using polychoric item correlations in the *psych* package (Revelle, 2021). Then, we flagged poorly performing items using data from the EFA analysis.[10] Lastly, we correlated classification probabilities with absolute factor loadings from the exploratory factor analysis—Table 7 describes these results.

After ranking items by classification probabilities generated by DeBERTa, we found that only 16% of the top quartiles were flagged as poorly performing items from EFA in contrast with 60%

Table 7. Item Factor Loadings and Transformer Model Predicted Class Logit Scores.

Variable	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)
1. Prob_{A}	–	–.33	–.20	–.05	–.32	**.32**	–.30	.17	.16	–.29
2. Prob_{C}		–	–.37	–.17	–.28	.08	**.76**	–.44	–.20	–.19
3. Prob_{E}			–	–.11	–.21	.02	–.28	**.54**	–.19	–.14
4. Prob_{N}				–	–.19	–.02	–.24	–.19	**.77**	–.34
5. Prob_{O}					–	–.06	–.16	–.18	–.29	**.78**
6. Fac_{A}						–	–.24	–.13	–.11	–.10
7. Fac_{C}							–	–.29	–.26	–.08
8. Fac_{E}								–	–.08	–.17
9. Fac_{N}									–	–.35
10. Fac_{O}										–

Note. Factor analyzed responses based on 461 respondents and 100 items. Factor loadings (*Fac*) calculated using item polychoric correlations. Cell values represent correlation between the *absolute value* of factor loadings and classification Logit Scores (*Logit*) from the fine-tuned *DeBERTa* model. Subscripts represent Big Five factors agreeableness (A), conscientiousness (C), extraversion (E), neuroticism (N), and openness (O). Maximum correlations with each factor are in **bold.** DeBERTa= decoding-enhanced Bidirectional Encoder Representations from Transformers with disentangled attention; Prob= probability.

of items in the bottom quartile. Results also indicate that 90% of the items with less than a .95 classification probability had issues arising from EFA. This finding suggests a transformer's classification probabilities may be useful when flagging items that will perform poorly during factor analysis. To elaborate on this relationship, we examined the correlations between predicted label probabilities and factor loadings; Table 7 illustrates these results. Interestingly, all the EFA factors correlated uniquely to just one of the label probability variables—with four out of the five factors being highly correlated ($r = .54–.78$). While DeBERTa's classification probabilities were not identical to factors loadings, there seems to be a strong enough relationship to warrant further research.

Recommendation. Results suggest that, by applying well-estimated cutoffs to classification probabilities, transformer models could help flag potentially problematic items before data collection. These results could help researchers and practitioners navigate issues that may arise during scale development (e.g., Block, 1995; Hopwood & Donnellan, 2010; Preacher & MacCallum, 2003). Researchers could estimate these thresholds for themselves. The process—for those developing scales where archival factor analytic data exists—would begin by using finalized scale items (i.e., items that performed well during factor analysis) to train a classification model. Next, researchers could input poorly performing items, removed items, and cross-loading items (identified by archival EFA data or past research) into the model for prediction. The predicted classification probabilities of the poorly performing items would help researchers to better estimate a lower bound for the prediction threshold. Hypothetically, text classification has the potential to replace factor analysis; however, for now, we advise against this. Underscoring this suggestion is the fact that the class labels used to train the proposed classification models are derivatives of factor analysis, given the Big Five model is based in factor analysis (Costa & McCrae, 1995). This implies that scale developers, especially those of novel scales, should use factor analysis to establish their class labels.

Discussion

The overarching goal of our research was to propose a state-of-the-art text classification technique for content analyzing personality items. To support this goal, we first introduced transformer models, elaborated on their advantages, and described considerations researchers should have when applying transformers for text classification. Then, we evaluated the proposed approach against human raters and alternative text classification techniques. We continued by addressing several lingering questions that researchers and practitioners may have when performing the methods described. Now, in the subsequent sections, we expand on the contributions to scale development, the limitations of this study, and avenues for future research.

Contributions to Personality Scale Development

Although researchers have demonstrated NLP applications in the organizational context (e.g., Pandey & Pandey, 2019; Short et al., 2010; Speer, 2021), the present study provides several novel contributions. First, we apply text classification for a novel purpose, to help automate and enhance the content validation of traditional psychological scales. Importantly, our proposed method would improve the efficiency of scale development in the technologically fast-changing test environment. Overall, researchers should be optimistic about the potential of transformers to automate, if not drastically augment, the content analysis process.

Second, while performing text classification, we introduced researchers and practitioners to several emerging NLP models with significant potential. Specifically, we illustrate text classification using several transformer models, which researchers have yet to widely adopt in the organizational and psychological sciences (Boyd & Schwartz, 2021; Eichstaedt et al., 2020; Kennedy et al., 2021)—we find just one example of transformers applied for text classification (i.e., Min et al., 2021). Still, we demonstrated how transformers could save considerable time and effort by simplifying the text classification process without compromising performance. We hope the methods described here help excite researchers—both familiar and unfamiliar—about transformers, so much so that researchers and practitioners may look to apply these models to help solve their particular problems. Third, we provide several step-by-step tutorials for training transformer models. Given that NLP tutorials often cater to those outside the organizational and psychological sciences (Kobayashi et al., 2018a), we hope our materials help ease non-experts into the world of transformer models. Fourth, we end this research by addressing several important questions when applying transformers to automate the content validation process. Our recommendations for overcoming the questions presented produce several significant implications. For example, transformers may give scale developers insight into a scale's factor structure *before* administering the scale, offer immediate and empirical feedback to item writers, and help to map the broad spectrum of personality systematically.

Lastly, we hope our research illustrates the general appeal of text classification using scale items as a research methodology. For those looking to publish organizational and psychological research, there is an immense number of research opportunities for leveraging archival or existing data. For example, one could compare text classification indices or text similarity indices (e.g., cosine distance) of scale items to published convergence and divergence effect sizes (e.g., Bainbridge et al., 2022; Schwaba et al., 2020). Also, future research could extend some of the empirical demonstrations we present by, for instance, examining the ability of classification probabilities to predict poorly performing items. Researchers could perform these studies without collecting scale responses from participants.

Limitations and Future Work

Here, we point out areas for improvement. First, although the methods presented generalize to additional psychological constructs, our illustration focuses on the Big Five model of personality in which open-source data are widely available. A few-shot or zero-shot model could be used for developing scales measuring psychological constructs where labeled data is scarce (Rahman et al., 2018).[11] Additionally, since we classified Likert statements, researchers classifying longer documents (e.g., essays, job descriptions, and cover letters) must consider document length when preprocessing. We recommend that researchers split (i.e., chunk) documents into sentences and then perform the classification task. Researchers can aggregate sentence-level predictions up to the document level when interpreting results. Second, there are several limitations related to our research procedures. Notably, our raters did not have the ability to classify items as "Other" or "not applicable." Additionally, our primary demonstration did not train the text classification models to classify a sixth "Other" label in addition to the Big Five labels. Still, we provide a tutorial (see "*Fine-Tuning Transformer for Big Five Inclusion"* in our GitHub repository) and discussion on how to navigate this issue (see the "Question 3: How to Account for "Other" [Non-Big Five] Items?" section). Future research could compare classification accuracy under this rating scenario (i.e., with the addition of an "Other" category). Third, there are limitations regarding technology. Although we provide our tutorials in a virtual environment, we acknowledge there are still significant hardware requirements for those wanting to use transformers on their local computers. As a final limitation related to technology, since we used GPT-3 for few-shot classification, those researchers wanting to replicate our few-shot tutorial should apply for an access key.[12]

There is a broad spectrum of opportunity for future research like the research presented here. First, there are NLP applications beyond text classification that may add value to scale development and assessment. Researchers could apply "masked language modeling," for example, to examine advanced lexical patterns in items and their impact on psychometric indices of scale items (e.g., Cutler & Condon, 2022). Second, future research could also explore the convergence between

NLP-based indices (e.g., classification probabilities and cosine similarity) and common psychometric indices such as item factor loadings. Third, future research could extend transformers to address several of the substantive questions facing the broader area of personality. For instance, by using a transformer trained to classify Big Five and non-Big Five items or a generative model (e.g., GPT-3) trained to produce "blended" or cross-loading items, researchers could thoroughly examine problems related to the representativeness and relevance of current Big Five inventories. Fourth, we hope researchers continue to put effort in constructing openly available data sources, such as the IPIP (Goldberg et al., 2006) and SAPA project (Condon, 2018). These resources are critically important to research of this nature. Fifth, there is potential for researchers and practitioners to extend the proposed approaches to the development of cognitive and noncognitive scales beyond those measuring Big Five personality. Finally, future research should continue to compare transformer models to human raters in various contexts (e.g., nominal groups, timed rating sessions, or different classification tasks).

Conclusion

This article demonstrates a novel NLP-based approach to content analysis using state-of-the-art pretrained transformer models. By applying transformers for text classification, we illustrate an automated approach to the content validation of Big Five personality scales. When compared to traditional approaches to text classification, our proposed method can drastically reduce the effort involved without compromising performance—performing as well as (if not better than) human raters when classifying personality items by their trait label. We hope this research provides a springboard into the world of transformers for scale developers and the field more broadly.

Acknowledgments

We gratefully acknowledge Dr. Lee Anna Clark at the University of Notre Dame, Dr. David Condon at the University of Oregon, and Dr. Lewis Goldberg of the Oregon Research Institute—for graciously providing guidance and support. Additionally, we are thankful for Associate Editor Louis Tay and the anonymous reviewers at Organizational Research Methods—their constructive feedback on earlier revisions helped make this article what it is.

Author Note

Data, code, and additional online materials are provided in the Github repository: https://github.com/Shea-Fyffe/transforming-personality-scales.

Declaration of Conflicting Interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) received no financial support for the research, authorship, and/or publication of this article.

ORCID iDs

Shea Fyffe https://orcid.org/0000-0003-0312-7915
Philseok Lee https://orcid.org/0000-0002-6965-0808

Supplemental Material

Supplemental material for this article is available online.

Notes

1. To clarify, NLP is a domain of computer-assisted methods for analyzing text data generated by humans (Liddy, 2001)—a subset of AI-based techniques relevant to text and speech data.
2. For simplicity, we use the terms "class(es)" and "label(s)" interchangeably.
3. Despite using the term "content validity" in earlier parts of this article, readers should note that this study more accurately focuses on *content relevance*—one of the two subcomponents of content validity (Haynes et al., 1995). The additional component—content representativeness—is not central to our investigation, though we elaborate on this issue in the latter parts of this article.
4. RNN and LSTM models must account for all words appearing before a target word when updating the contextual embedding of that word. This method is highly inefficient, computationally expensive, and encodes words one at a time (Azunre, 2021).
5. For example, a source task might be predicting a word given surrounding words or predicting the next sentence given the sentence prior. Typically, pre-trained models perform source tasks millions of times (e.g., Devlin et al., 2019) using a broad and vast spectrum of text documents (e.g., Wikipedia articles, digitized books, social media posts, web pages).
6. Researchers should keep in mind that "tokens" are not synonymous with "words," since some words can be composed of two to three tokens. For example, the word "other's" results in the tokens [other, ', s] (see Song et al., 2021).
7. Following a reviewer's suggestion, we conducted additional experiments with a nonlinear support-vector machine (i.e., SVM with radial-basis kernel). Our results showed that using a nonlinear support vector machine showed no significant improvement. This may be a result of several factors. First, we did not tune model hyper-parameters (as we wanted to make our analysis as straightforward as possible). Second, in cases like ours where the number of features is larger than the number of test cases, a linear kernel may perform well (Hsu et al., 2010). Given there was no meaningful difference in performance and linear SVMs are more efficient in terms of training time (Kowsari et al., 2019), we used a linear SVM (in addition to *XGBoost*) as our selected classifiers.
8. Using the direct-consensus method, we treated the most common label as the predicted label. In the case of a tie, we randomly sampled one of the viable options. This approach resulted in a rater classification accuracy of 76.4%.
9. While we do not describe training a binary classification transformer like the one suggested in detail, we provide a dataset (see `~/raw-data/supplemental-item-data.csv`) and the *"Fine-Tuning Transformer for Big Five Inclusion"* tutorial (see project's GitHub repository) to perform such a task.
10. Specifically, we flagged items with a communality below .20, an absolute factor loading below .40, a secondary factor loading that was >75% of the primary loading, and items that loaded to factors that did not align with their actual label.
11. Those interested in applying this approach to assessments with fewer items should see the *Few-Shot Learning with Transformers* in this project's code repository or the *Zero-Shot Classification* pipeline in Python's *Transformers* library (Wolf et al., 2020).
12. See https://beta.openai.com/ to register for a GPT-3 access key.

References

Adhikari, A., Ram, A., Tang, R., & Lin, J. (2019). DocBERT: Bert for document classification. *ArXiv:1904.08398 [Cs]*. http://arxiv.org/abs/1904.08398

Alberti, C., Lee, K., & Collins, M. (2019). A BERT baseline for the natural questions. *ArXiv:1901.08634 [Cs]*. http://arxiv.org/abs/1901.08634

Allport, G. W. (1937). *Personality: A psychological interpretation* (pp. xiv, 588). Holt.

Anderson, J. C., & Gerbing, D. W. (1991). Predicting the performance of measures in a confirmatory factor analysis with a pretest assessment of their substantive validities. *Journal of Applied Psychology*, *76*(5), 732-740. https://doi.org/10.1037/0021-9010.76.5.732

Ashton, M. C., & Lee, K. (2005). A defence of the lexical approach to the study of personality structure. *European Journal of Personality*, *19*(1), 5-24. https://doi.org/10.1002/per.541

Ashton, M. C., Lee, K., Perugini, M., Szarota, P., de Vries, R. E., Di Blas, L., Boies, K., & De Raad, B. (2004). A six-factor structure of personality-descriptive adjectives: Solutions from psycholexical studies in seven languages. *Journal of Personality and Social Psychology*, *86*(2), 356-366. https://doi.org/10.1037/0022-3514.86.2.356

Azunre, P. (2021). *Transfer learning for natural language processing*. Manning Publications Co.

Bainbridge, T. F., Ludeke, S. G., & Smillie, L. D. (2022). Evaluating the Big Five as an organizing framework for commonly used psychological trait scales. *Journal of Personality and Social Psychology*, *122*(4), 749-777. https://doi.org/10.1037/pspp0000395

Bansal, T., Jha, R., Munkhdalai, T., & McCallum, A. (2020). *Self-supervised meta-learning for few-shot natural language classification tasks. Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 522-534. https://doi.org/10.18653/v1/2020.emnlp-main.38

Bayer, M., Kaufhold, M.-A., & Reuter, C. (2022). A survey on data augmentation for text classification. *ACM Computing Surveys*, *55*(7), 1-39. https://doi.org/10.1145/3544558

Block, J. (1995). A contrarian view of the five-factor approach to personality description. *Psychological Bulletin*, *117*(2), 187-215. https://doi.org/10.1037/0033-2909.117.2.187

Block, J. (2010). The five-factor framing of personality and beyond: Some ruminations. *Psychological Inquiry*, *21*(1), 2-25. https://doi.org/10.1080/10478401003596626

Boyd, R. L., & Schwartz, H. A. (2021). Natural language analysis and the psychology of verbal behavior: The past, present, and future states of the field. *Journal of Language and Social Psychology*, *40*(1), 21-41. https://doi.org/10.1177/0261927X20967028

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., … D. Amodei (2020). Language models are few-shot learners. *ArXiv:2005.14165 [Cs]*. http://arxiv.org/abs/2005.14165

Campion, M. C., Campion, M. A., Campion, E. D., & Reider, M. H. (2016). Initial investigation into computer scoring of candidate essays for personnel selection. *Journal of Applied Psychology*, *101*(7), 958-975. https://doi.org/10.1037/apl0000108

Cer, D., Yang, Y., Kong, S., Hua, N., Limtiaco, N., St. John, R., Constant, N., Guajardo-Cespedes, M., Yuan, S., Tar, C., Strope, B., & Kurzweil, R. (2018). *Universal Sentence Encoder for English. Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 169-174. https://doi.org/10.18653/v1/D18-2029

Chen, Y., Zhong, R., Zha, S., Karypis, G., & He, H. (2022). *Meta-learning via language model in-context tuning* (arXiv:2110.07814). arXiv. https://doi.org/10.48550/arXiv.2110.07814

Chronopoulou, A., Baziotis, C., & Potamianos, A. (2019). *An embarrassingly simple approach for transfer learning from pretrained language models. Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 2089-2095. https://doi.org/10.18653/v1/N19-1213

Clark, L. A., & Watson, D. (2016). *Constructing validity: Basic issues in objective scale development* (p. 203). American Psychological Association. https://doi.org/10.1037/14805-012

Clark, L. A., & Watson, D. (2019). Constructing validity: New developments in creating objective measuring instruments. *Psychological Assessment*, *31*(12), 1412-1427. https://doi.org/10.1037/pas0000626

Cloninger, C. R., Svrakic, D. M., & Przybeck, T. R. (1993). A psychobiological model of temperament and character. *Archives of General Psychiatry*, *50*(12), 975-990. https://doi.org/10.1001/archpsyc.1993.01820240059008

Colquitt, J. A., Sabey, T. B., Rodell, J. B., & Hill, E. T. (2019). Content validation guidelines: Evaluation criteria for definitional correspondence and definitional distinctiveness. *Journal of Applied Psychology*, *104*(10), 1243-1265. https://doi.org/10.1037/apl0000406

Condon, D. (2019). *Database of individual differences survey tools*. Harvard Dataverse. https://doi.org/10.7910/DVN/T1NQ4V

Condon, D. M. (2018). The SAPA personality inventory: An empirically-derived, hierarchically-organized self-report personality assessment model. *PsyArXiv*. https://doi.org/10.31234/osf.io/sc4p9

Condon, D. M., Wood, D., Mõttus, R., Booth, T., Costantini, G., Greiff, S., Johnson, W., Lukaszewski, A., Murray, A., Revelle, W., Wright, A. G. C., Ziegler, M., & Zimmermann, J. (2020). Bottom-up construction of a personality taxonomy. *European Journal of Psychological Assessment*, *36*(6), 923-934. https://doi.org/10.1027/1015-5759/a000626

Conneau, A., & Kiela, D. (2018, May). *SentEval: An evaluation toolkit for universal sentence representations. Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. LREC 2018. https://www.aclweb.org/anthology/L18-1269

Costa, P. T., & McCrae, R. R. (1995). Domains and facets: Hierarchical personality assessment using the Revised NEO Personality Inventory. *Journal of Personality Assessment*, *64*(1), 21-50. https://doi.org/10.1207/s15327752jpa6401_2

Croce, D., Castellucci, G., & Basili, R. (2020). *GAN-BERT: Generative adversarial learning for robust text classification with a bunch of labeled examples. Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2114-2119. https://doi.org/10.18653/v1/2020.acl-main.191

Cutler, A., & Condon, D. M. (2022). *Deep lexical hypothesis: Identifying personality structure in natural language* (arXiv:2203.02092). arXiv. https://doi.org/10.48550/arXiv.2203.02092

De Boom, C., Van Canneyt, S., Demeester, T., & Dhoedt, B. (2016). Representation learning for very short texts using weighted word embedding aggregation. *Pattern Recognition Letters*, *80*, 150-156. https://doi.org/10.1016/j.patrec.2016.06.012

de Rosa, G. H., & Papa, J. P. (2021). A survey on text generation using generative adversarial networks. *Pattern Recognition*, *119*, 108098. https://doi.org/10.1016/j.patcog.2021.108098

DeGeest, D. S., & Schmidt, F. (2015). A rigorous test of the fit of the circumplex model to big five personality data: Theoretical and methodological issues and two large sample empirical tests. *Multivariate Behavioral Research*, *50*(3), 350-364. https://doi.org/10.1080/00273171.2015.1004568

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. *ArXiv:1810.04805 [Cs]*. http://arxiv.org/abs/1810.04805

DiStefano, C., & Motl, R. W. (2009). Personality correlates of method effects due to negatively worded items on the Rosenberg Self-Esteem Scale. *Personality and Individual Differences*, *46*(3), 309-313. https://doi.org/10.1016/j.paid.2008.10.020

Domingos, P. (2012). A few useful things to know about machine learning. *Communications of the ACM*, *55*(10), 78-87. https://doi.org/10.1145/2347736.2347755

Eichstaedt, J. C., Kern, M. L., Yaden, D. B., Schwartz, H. A., Giorgi, S., Park, G., Hagan, C., Tobolsky, V., Smith, L. K., Buffone, A., Iwry, J., Seligman, M., & Ungar, L. H. (2020). Closed and open vocabulary approaches to text analysis: A review, quantitative comparison, and recommendations. *PsyArXiv*. https://doi.org/10.31234/osf.io/t52c6

Elman, J. L. (1990). Finding structure in time. *Cognitive Science*, *14*(2), 179-211. https://doi.org/10.1207/s15516709cog1402_1

Fleiss, J. L. (1981). Balanced incomplete block designs for inter-rater reliability studies. *Applied Psychological Measurement*, *5*(1), 105-112. https://doi.org/10.1177/014662168100500115

Foa, E. B., Huppert, J. D., Leiberg, S., Langner, R., Kichic, R., Hajcak, G., & Salkovskis, P. M. (2002). The Obsessive-Compulsive Inventory: Development and validation of a short version. *Psychological Assessment*, *14*(4), 485-496. https://doi.org/10.1037/1040-3590.14.4.485

Geng, C., Huang, S., & Chen, S. (2021). Recent advances in open set recognition: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *43*(10), 3614-3631. https://doi.org/10.1109/TPAMI.2020.2981604

Goldberg, L. R. (1993). The structure of phenotypic personality traits. *American Psychologist*, *48*(1), 26-34. https://doi.org/10.1037/0003-066X.48.1.26

Goldberg, L. R., Johnson, J. A., Eber, H. W., Hogan, R., Ashton, M. C., Cloninger, C. R., & Gough, H. G. (2006). The international personality item pool and the future of public-domain personality measures. *Journal of Research in Personality*, *40*(1), 84-96. https://doi.org/10.1016/j.jrp.2005.08.007

Goldberg, L. R., & Saucier, G. (2016). *The Eugene-Springfield community sample: Information available from the research participants* (Tech. Rep. No. 56-1). Oregon Research Institute.

Goldberg, L. R., & Velicer, W. F. (2006). Principles of exploratory factor analysis. *Differentiating normal and abnormal personality* (2nd ed., pp. 209-237). Springer Publishing Company.

Goodfellow, I. J., Mirza, M., Xiao, D., Courville, A., & Bengio, Y. (2015). *An empirical investigation of catastrophic forgetting in gradient-based neural networks* (arXiv:1312.6211). arXiv. https://doi.org/10.48550/arXiv.1312.6211

Götz, F., Maertens, R., Linden, D. S., & van der. (2021). *Let the algorithm speak: How to use neural networks for automatic item generation in psychological scale development*. PsyArXiv. https://doi.org/10.31234/osf.io/m6s28

Halder, K., Akbik, A., Krapac, J., & Vollgraf, R. (2020). *Task-aware representation of sentences for generic text classification.* Proceedings of the 28th International Conference on Computational Linguistics, 3202-3213. https://doi.org/10.18653/v1/2020.coling-main.285

Harris, Z. S. (1954). Distributional structure. *WORD*, *10*(2–3), 146-162. https://doi.org/10.1080/00437956.1954.11659520

Hattie, J. (1985). Methodology review: Assessing unidimensionality of tests and items. *Applied Psychological Measurement*, *9*(2), 139-164. https://doi.org/10.1177/014662168500900204

Haynes, S. N., Richard, D. C. S., & Kubany, E. S. (1995). Content validity in psychological assessment: A functional approach to concepts and methods. *Psychological Assessment*, *7*(3), 238-247. https://doi.org/10.1037/1040-3590.7.3.238

He, P., Liu, X., Gao, J., & Chen, W. (2021). DeBERTa: Decoding-enhanced BERT with disentangled attention. *ArXiv:2006.03654 [Cs]*. http://arxiv.org/abs/2006.03654

Hendrycks, D., Liu, X., Wallace, E., Dziedzic, A., Krishnan, R., & Song, D. (2020). *Pretrained transformers improve out-of-distribution robustness. Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2744-2751. https://doi.org/10.18653/v1/2020.acl-main.244

Hickman, L., Thapa, S., Tay, L., Cao, M., & Srinivasan, P. (2020). Text preprocessing for text mining in organizational research: Review and recommendations. *Organizational Research Methods*, 114, 1-33. https://doi.org/10.1177/1094428120971683

Hinkin, T. R. (1998). A brief tutorial on the development of measures for use in survey questionnaires. *Organizational Research Methods*, *1*(1), 104-121. https://doi.org/10.1177/109442819800100106

Hinkin, T. R., & Tracey, J. B. (1999). An analysis of variance approach to content validation. *Organizational Research Methods*, *2*(2), 175-186. https://doi.org/10.1177/109442819922004

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

Hofstee, W. K., de Raad, B., & Goldberg, L. R. (1992). Integration of the Big Five and circumplex approaches to trait structure. *Journal of Personality and Social Psychology*, *63*(1), 146-163. https://doi.org/10.1037/0022-3514.63.1.146

Hommel, B. E., Wollang, F.-J. M., Kotova, V., Zacher, H., & Schmukle, S. C. (2022). Transformer-based deep neural language modeling for construct-specific automatic item generation. *Psychometrika*, *87*(2), 749-772. https://doi.org/10.1007/s11336-021-09823-9

Hopwood, C. J., & Donnellan, M. B. (2010). How should the internal structure of personality inventories be evaluated? *Personality and Social Psychology Review*, *14*(3), 332-346. https://doi.org/10.1177/1088868310361240

Howard, J., & Ruder, S. (2018). Universal language model fine-tuning for text classification. *ArXiv:1801.06146 [Cs, Stat]*. http://arxiv.org/abs/1801.06146

Hsu, C., Chang, C., & Lin, C. (2010). A practical guide to support vector classification [Technical Report]. National Taiwan University. https://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf

Ilmini, W. M. K. S., & Fernando, T. G. I. (2017). *Computational personality traits assessment: A review. 2017 IEEE International Conference on Industrial and Information Systems (ICIIS)*, 1-6. https://doi.org/10.1109/ICIINFS.2017.8300416

Jiao, H., & Lissitz, R. W. (2020). *Application of artificial intelligence to assessment*. IAP.

John, O. P., Angleitner, A., & Ostendorf, F. (1988). The lexical approach to personality: A historical review of trait taxonomic research. *European Journal of Personality*, *2*(3), 171-203. https://doi.org/10.1002/per.2410020302

John, O. P., Naumann, L. P., & Soto, C. J. (2008). Paradigm shift to the integrative Big Five trait taxonomy: History, measurement, and conceptual issues. In *Handbook of personality: Theory and research* (3rd ed, pp. 114-158). The Guilford Press.

Kalyan, K. S., Rajasekharan, A., & Sangeetha, S. (2021). AMMUS: A survey of transformer-based pretrained models in natural language processing. *ArXiv:2108.05542 [Cs]*. http://arxiv.org/abs/2108.05542

Kennedy, B., Ashokkumar, A., Boyd, R. L., & Dehghani, M. (2021). Text analysis for psychology: Methods, principles, and practices. *PsyArXiv*. https://doi.org/10.31234/osf.io/h2b8t

Keskar, N. S., McCann, B., Varshney, L. R., Xiong, C., & Socher, R. (2019). CTRL: A conditional transformer language model for controllable generation. ArXiv:1909.05858 [Cs]. http://arxiv.org/abs/1909.05858

Kobayashi, V. B., Mol, S. T., Berkers, H. A., Kismihók, G., & Den Hartog, D. N. (2018a). Text mining in organizational research. *Organizational Research Methods*, *21*(3), 733-765. https://doi.org/10.1177/1094428117722619

Kobayashi, V. B., Mol, S. T., Berkers, H. A., Kismihók, G., & Den Hartog, D. N. (2018b). Text classification for organizational researchers: A tutorial. *Organizational Research Methods*, *21*(3), 766-799. https://doi.org/10.1177/1094428117719322

Kowsari, K., Meimandi, K. J., Heidarysafa, M., Mendu, S., Barnes, L. E., & Brown, D. E. (2019). Text classification algorithms: A survey. *Information*, *10*(4), 150. https://doi.org/10.3390/info10040150

Krippendorff, K. (2018). *Content analysis: An introduction to its methodology*. Sage.

Kuhn, M. (2021). *caret: Classification and regression training* [Manual]. https://CRAN.R-project.org/package=caret

Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., & Soricut, R. (2020). ALBERT: A lite BERT for self-supervised learning of language representations. *ArXiv:1909.11942 [Cs]*. http://arxiv.org/abs/1909.11942

Lee, P., Fyffe, S., Son, M., Jia, Z., & Yao, Z. (2023). A paradigm shift from "human writing" to "machine generation" in personality test development: An application of state-of-the-art natural language processing. *Journal of Business and Psychology*, *38*(1), 163-190. https://doi.org/10.1007/s10869-022-09864-6

Levenson, H. (1981). Differentiating among internality, powerful others, and chance. In H. M. Lefcourt (Ed.), *Research with the locus of control construct* (pp. 15-63). Academic Press. https://doi.org/10.1016/B978-0-12-443201-7.50006-3

Liang, C., Jiang, H., Zuo, S., He, P., Liu, X., Gao, J., Chen, W., & Zhao, T. (2022). *No parameters left behind: Sensitivity guided adaptive learning rate for training large transformer models* (arXiv:2202.02664). arXiv. https://doi.org/10.48550/arXiv.2202.02664

Liddy, E. (2001). *Natural language processing. In Encyclopedia of library and information science* (2nd ed.). Marcel Decker, Inc. https://surface.syr.edu/istpub/63

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., & Stoyanov, V. (2019). RoBERTa: A robustly optimized BERT pretraining approach. *ArXiv:1907.11692 [Cs]*. http://arxiv.org/abs/1907.11692

Liu, Z., Lin, Y., & Sun, M. (2020). *Representation learning for natural language processing*. Springer Singapore. https://doi.org/10.1007/978-981-15-5573-2

Loevinger, J. (1957). Objective tests as instruments of psychological theory. *Psychological Reports*, *3*, 635-694. https://doi.org/10.2466/PR0.3.7.635-694

Marsh, H. W., Lüdtke, O., Muthén, B., Asparouhov, T., Morin, A. J. S., Trautwein, U., & Nagengast, B. (2010). A new look at the Big Five factor structure through exploratory structural equation modeling. *Psychological Assessment*, *22*(3), 471-491. https://doi.org/10.1037/a0019227

McCrae, R. R., & Costa, P. T. (1987). Validation of the five-factor model of personality across instruments and observers. *Journal of Personality and Social Psychology*, *52*(1), 81-90. https://doi.org/10.1037/0022-3514.52.1.81

Meehl, P. E., & Rosen, A. (1955). Antecedent probability and the efficiency of psychometric signs, patterns, or cutting scores. *Psychological Bulletin*, *52*(3), 194-216. https://doi.org/10.1037/h0048070

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *ArXiv:1301.3781 [Cs]*. http://arxiv.org/abs/1301.3781

Min, H., Peng, Y., Shoss, M., & Yang, B. (2021). Using machine learning to investigate the public's emotional responses to work from home during the COVID-19 pandemic. *Journal of Applied Psychology*, *106*, 214-229. https://doi.org/10.1037/apl0000886

Mirończuk, M. M., & Protasiewicz, J. (2018). A recent overview of the state-of-the-art elements of text classification. *Expert Systems with Applications*, *106*, 36-54. https://doi.org/10.1016/j.eswa.2018.03.058

Miyajiwala, A., Ladkat, A., Jagadale, S., & Joshi, R. (2022). On sensitivity of deep learning based text classification algorithms to practical input perturbations. In K. Arai (Ed.), *Intelligent computing* (pp. 613-626). Springer International Publishing. https://doi.org/10.1007/978-3-031-10464-0_42

Nangia, N., & Bowman, S. R. (2019). Human vs. muppet: A conservative estimate of human performance on the glue benchmark. *ArXiv:1905.10425 [Cs]*. http://arxiv.org/abs/1905.10425

Norman, W. T. (1963). Toward an adequate taxonomy of personality attributes: Replicated factor structure in peer nomination personality ratings. *The Journal of Abnormal and Social Psychology*, *66*(6), 574-583. https://doi.org/10.1037/h0040291

Padhy, S., Nado, Z., Ren, J., Liu, J., Snoek, J., & Lakshminarayanan, B. (2020). Revisiting one-vs-all classifiers for predictive uncertainty and out-of-distribution detection in neural networks. *ArXiv:2007.05134 [Cs, Stat]*. http://arxiv.org/abs/2007.05134

Pan, S. J., & Yang, Q. (2010). A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, *22*(10), 1345-1359. https://doi.org/10.1109/TKDE.2009.191

Pandey, S., & Pandey, S. K. (2019). Applying natural language processing capabilities in computerized textual analysis to measure organizational culture. *Organizational Research Methods*, *22*(3), 765-797. https://doi.org/10.1177/1094428117745648

Paunonen, S. V., & Jackson, D. N. (2000). What is beyond the big five? Plenty!. *Journal of Personality*, *68*(5), 821-835. https://doi.org/10.1111/1467-6494.00117

Peng, Y., Chen, Q., & Lu, Z. (2020). *An empirical study of multi-task learning on BERT for biomedical text mining*. Proceedings of the 19th SIGBioMed Workshop on Biomedical Language Processing, 205-214. https://doi.org/10.18653/v1/2020.bionlp-1.22

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

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018). Deep contextualized word representations. *ArXiv:1802.05365 [Cs]*. http://arxiv.org/abs/1802.05365

Peters, M. E., Ruder, S., & Smith, N. A. (2019). *To tune or not to tune? Adapting pretrained representations to diverse tasks* (arXiv:1903.05987). arXiv. http://arxiv.org/abs/1903.05987

Peterson, C., & Seligman, M. E. P. (2004). *Character strengths and virtues: A handbook and classification* (pp. xiv, 800). Oxford University Press.

Phang, J., Févry, T., & Bowman, S. R. (2019). Sentence encoders on STILTS: Supplementary training on intermediate labeled-data tasks. *ArXiv:1811.01088 [Cs]*. http://arxiv.org/abs/1811.01088

Pilehvar, M. T., & Camacho-Collados, J. (2020). Embeddings in natural language processing: Theory and advances in vector representations of meaning. *Synthesis Lectures on Human Language Technologies*, *13*(4), 1-175. https://doi.org/10.2200/S01057ED1V01Y202009HLT047

Preacher, K. J., & MacCallum, R. C. (2003). Repairing Tom Swift's electric factor analysis machine. *Understanding Statistics*, *2*(1), 13-43. https://doi.org/10.1207/S15328031US0201_02

Putka, D. J., Le, H., McCloy, R. A., & Diaz, T. (2008). Ill-structured measurement designs in organizational research: Implications for estimating interrater reliability. *The Journal of Applied Psychology*, *93*(5), 959-981. https://doi.org/10.1037/0021-9010.93.5.959

Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2018). *Improving language understanding by generative pre-training* (pp. 1-12) [Technical Report]. OpenAI. https://www.cs.ubc.ca/~amuham01/LING530/papers/radford2018improving.pdf

Rahman, S., Khan, S., & Porikli, F. (2018). A unified approach for conventional zero-shot, generalized zero-shot, and few-shot learning. *IEEE Transactions on Image Processing*, *27*(11), 5652-5667. https://doi.org/10.1109/TIP.2018.2861573

Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence embeddings using Siamese BERT-networks. *ArXiv:1908.10084 [Cs]*. http://arxiv.org/abs/1908.10084

Revelle, W. (2021). *psych: Procedures for psychological, psychometric, and personality research* [Manual]. https://CRAN.R-project.org/package=psych

Roady, R., Hayes, T. L., Kemker, R., Gonzales, A., & Kanan, C. (2020). Are open set classification methods effective on large-scale datasets? *PLoS ONE*, *15*(9), e0238302. https://doi.org/10.1371/journal.pone.0238302

Rosellini, A. J., & Brown, T. A. (2021). Developing and validating clinical questionnaires. *Annual Review of Clinical Psychology*, *17*, 55-81. https://doi.org/10.1146/annurev-clinpsy-081219-115343

Ruder, S. (2017). *Transfer learning—Machine learning's next frontier*. http://ruder.io/transfer-learning/

Ruder, S. (2021). *Recent advances in language model fine-tuning*. http://ruder.io/recent-advances-lm-fine-tuning

Rudkowsky, E., Haselmayer, M., Wastian, M., Jenny, M., Emrich, Š, & Sedlmair, M. (2018). More than bags of words: Sentiment analysis with word embeddings. *Communication Methods and Measures*, *12*(2–3), 140-157. https://doi.org/10.1080/19312458.2018.1455817

Russell, D. W. (2002). In search of underlying dimensions: The use (and abuse) of factor analysis in personality and social psychology bulletin. *Personality and Social Psychology Bulletin*, *28*(12), 1629-1646. https://doi.org/10.1177/014616702237645

Saarikoski, J., Joutsijoki, H., Jarvelin, K., Laurikkala, J., & Juhola, M. (2015). On the influence of training data quality on text document classification using machine learning methods. *International Journal of Knowledge Engineering and Data Mining*, *3*(2), 143-169. https://doi.org/10.1504/IJKEDM.2015.071284

Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2020). DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter. *ArXiv:1910.01108 [Cs]*. http://arxiv.org/abs/1910.01108

Saucier, G. (1997). Effects of variable selection on the factor structure of person descriptors. *Journal of Personality and Social Psychology*, *73*(6), 1296-1312. https://doi.org/10.1037/0022-3514.73.6.1296

Saucier, G., & Goldberg, L. R. (1998). What is beyond the big five? *Journal of Personality*, *66*, 495-524. https://doi.org/10.1111/1467-6494.00022

Scao, T. L., & Rush, A. M. (2021). How many data points is a prompt worth? *ArXiv:2103.08493 [Cs]*. http://arxiv.org/abs/2103.08493

Schick, T., & Schütze, H. (2021). It's not just size that matters: Small language models are also few-shot learners. *ArXiv:2009.07118 [Cs]*. http://arxiv.org/abs/2009.07118

Schwaba, T., Rhemtulla, M., Hopwood, C. J., & Bleidorn, W. (2020). A facet atlas: Visualizing networks that describe the blends, cores, and peripheries of personality structure. *PLoS ONE*, *15*(7), 0236893. https://doi.org/10.1371/journal.pone.0236893

Short, J. C., Broberg, J. C., Cogliser, C. C., & Brigham, K. H. (2010). Construct validation using computer-aided text analysis (CATA): An illustration using entrepreneurial orientation. *Organizational Research Methods*, *13*(2), 320-347. https://doi.org/10.1177/1094428109335949

Short, J. C., McKenny, A. F., & Reid, S. W. (2018). More than words? Computer-aided text analysis in organizational behavior and psychology research. *Annual Review of Organizational Psychology and Organizational Behavior*, *5*(1), 415-435. https://doi.org/10.1146/annurev-orgpsych-032117-104622

Shrestha, Y. R., He, V. F., Puranam, P., & von Krogh, G. (2021). Algorithm supported induction for building theory: How can we use prediction models to theorize? *Organization Science*, *32*(3), 856-880. https://doi.org/10.1287/orsc.2020.1382

Smith, R. W., Min, H., Ng, M. A., Haynes, N. J., & Clark, M. A. (2022). A content validation of work passion: Was the passion ever there? *Journal of Business and Psychology*, *38*(1), 191-213. https://doi.org/10.1007/s10869-022-09807-1

Song, X., Salcianu, A., Song, Y., Dopson, D., & Zhou, D. (2021). Fast wordpiece tokenization. *ArXiv:2012.15524 [Cs]*. http://arxiv.org/abs/2012.15524

Speer, A. B. (2021). Scoring dimension-level job performance from narrative comments: Validity and generalizability when using natural language processing. *Organizational Research Methods*, *24*(3), 572-594. https://doi.org/10.1177/1094428120930815

Sun, C., Qiu, X., Xu, Y., & Huang, X. (2020). How to fine-tune BERT for text classification? *ArXiv:1905.05583 [Cs]*. http://arxiv.org/abs/1905.05583

Tellegen, A., & Waller, N. G. (2008). Exploring personality through test construction: Development of the multidimensional personality questionnaire. In *The SAGE handbook of personality theory and assessment, vol 2: Personality measurement and testing* (pp. 261-292). Sage Publications, Inc. https://doi.org/10.4135/9781849200479.n13

Vabalas, A., Gowen, E., Poliakoff, E., & Casson, A. J. (2019). Machine learning algorithm validation with a limited sample size. *PLoS ONE*, *14*(11), 0224365. https://doi.org/10.1371/journal.pone.0224365

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention is all you need. *ArXiv:1706.03762 [Cs]*. http://arxiv.org/abs/1706.03762

Vodrahalli, K., Gerstenberg, T., & Zou, J. (2022). Uncalibrated models can improve human-AI collaboration. *ArXiv:2202.05983 [Cs]*. http://arxiv.org/abs/2202.05983

von Davier, M. (2018). Automated item generation with recurrent neural networks. *Psychometrika*, *83*(4), 847-857. https://doi.org/10.1007/s11336-018-9608-y

Wang, A., Pruksachatkun, Y., Nangia, N., Singh, A., Michael, J., Hill, F., Levy, O., & Bowman, S. R. (2019). SuperGLUE: A stickier benchmark for general-purpose language understanding systems. *ArXiv Preprint 1905.00537*.

Wang, S., Fang, H., Khabsa, M., Mao, H., & Ma, H. (2021). Entailment as Few-Shot Learner. *ArXiv:2104.14690 [Cs]*. http://arxiv.org/abs/2104.14690

Wilson, E. B. (1927). Probable inference, the law of succession, and statistical inference. *Journal of the American Statistical Association*, *22*(158), 209-212. https://doi.org/10.2307/2276774

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Scao, T. L., Gugger, S., … (2020). HuggingFace's transformers: State-of-the-art natural language processing. *ArXiv:1910.03771 [Cs]*. http://arxiv.org/abs/1910.03771

Worthington, R. L., & Whittaker, T. A. (2006). Scale development research: A content analysis and recommendations for best practices. *The Counseling Psychologist*, *34*(6), 806-838. https://doi.org/10.1177/0011000006288127

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R., & Le, Q. V. (2020). XLNET: Generalized autoregressive pretraining for language understanding. *ArXiv:1906.08237 [Cs]*. http://arxiv.org/abs/1906.08237

Yin, W., Rajani, N. F., Radev, D., Socher, R., & Xiong, C. (2020). Universal natural language processing with limited annotations: Try few-shot textual entailment as a start. *ArXiv:2010.02584 [Cs]*. http://arxiv.org/abs/2010.02584

Zellers, R., Bisk, Y., Schwartz, R., & Choi, Y. (2018). SWAG: A large-scale adversarial dataset for grounded commonsense inference. *ArXiv:1808.05326 [Cs]*. http://arxiv.org/abs/1808.05326

Zhang, T., Wu, F., Katiyar, A., Weinberger, K. Q., & Artzi, Y. (2021). Revisiting few-sample BERT fine-tuning. *ArXiv:2006.05987 [Cs]*. http://arxiv.org/abs/2006.05987

Author Biographies

Shea Fyffe is a PhD candidate in industrial-organizational psychology at George Mason University. His research broadly focuses on the assessment & measurement of personality in the workplace. Specifically, he is interested in "alternative" forms of personality assessment—for example—text-based and game-based personality assessments, as well as the types of methods used to implement such assessments (e.g., natural language processing and machine learning more broadly).

Philseok Lee is an assistant professor of psychology at George Mason University. His research focuses on the developments and applications of modern psychometric modeling, the application of machine learning and natural language processing techniques to work settings, faking issues in personnel selection, and the development of noncognitive personnel assessments.

Seth Kaplan is a professor of psychology at George Mason University. His research focuses on workplace well-being and employee emotions, the meaning of work, and team performance in organizational contexts.