Large Language Models for Psychological Assessment: A Comprehensive Overview

Jocelyn Brickman, M.A., Mehak Gupta, Ph.D., and Joshua R. Oltmanns, Ph.D.

Author's note:

Jocelyn Brickman, School of Psychology, Xavier University; Mehak Gupta, School of Engineering, SMU; Joshua R. Oltmanns, Department of Psychological & Brain Sciences, Washington University in St. Louis.

Correspondence should be addressed to Joshua R. Oltmanns, Department of Psychological & Brain Sciences, Washington University in St. Louis, St. Louis, MO. Email: j.oltmanns@wustl.edu.

Abstract

Large language models (LLMs) are extraordinary tools demonstrating potential to improve our understanding of psychological characteristics. They provide an unprecedented opportunity to supplement self-report in psychology research and practice with scalable behavioral assessment. However, they also pose unique risks and challenges. This article serves as an overview and guide for psychological scientists to evaluate LLMs for psychological assessment. In Section I, we briefly review the development of transformer-based LLMs and discuss their advances in natural language processing. In Section II, we describe the experimental design process including techniques for language data collection, audio processing and transcription, text preprocessing, and model selection, as well as analytic matters such as model output, model evaluation, hyperparameter tuning, model visualization, and topic modeling. At each stage, we describe options, important decisions, and resources for further in-depth learning, while providing examples from different areas of psychology. In Section III, we discuss important broader ethical and implementation issues and future directions for researchers using this methodology. The reader will develop an understanding of essential ideas and an ability to navigate the process of using LLMs for psychological assessment.

Large Language Models for Psychological Assessment: A Comprehensive Overview

Sound measurement is at the foundation of psychological science. In the past century, development of the construct validation process has propelled the field to meaningful theory testing (Clark & Watson, 2019; Strauss & Smith, 2009). Yet most of what we know about psychological constructs relies on self-report measurement, which has weaknesses such as socially desirable responding, over- and under-reporting, cultural and retrospective biases, and limitations in self-insight (e.g., Paulhus & Vazire, 2007). Psychologists strive to incorporate multimethod assessment into research and practice (APA Task Force on Psychological Assessment and Evaluation Guidelines, 2020) because it increases the validity of assessment (Hopwood & Bornstein, 2014; Meyer et al., 2001). However, actual use of multimethod assessment is rare, in large part because it can be burdensome and time consuming. Advances in artificial intelligence (AI), in particular Large Language Models (LLMs), provide opportunities to incorporate, as well as improve and facilitate, multimethod psychological assessment.

Language as an assessment tool has several strengths. It is *behavioral*, providing a more objective approach to assessment, and it can be *natural,* providing ecological validity to assessment, thereby avoiding some of the inherent limitations of self-report questionnaires. It is also *rich*, allowing individuals to express themselves in ways that break free of traditional rating scales (Kjell et al., 2024). Using language to study psychological constructs has already greatly expanded our understanding of them (Pennebaker et al., 2003). With extraordinary recent advances in technology, language will likely continue to expand our knowledge of psychological characteristics at a rapid pace.

There are also more practical advantages to using language as an assessment tool. Language as an assessment tool is *scalable*. Validated LLM tools could be more easily implemented into routine research and clinical activities that involve speech, supplementing self-report assessments and saving time and resources for both participants/patients and researchers/clinicians. LLM psychological assessment tools could also greatly enhance assessment *coverage*. For example, well-developed LLM-based tools may assess a wide variety of psychological constructs from a single language sample in a setting where a similar amount of assessment may take many hours of questionnaire completion. In emergencies or particularly low-resource situations, validated LLMs might provide assessments from language when no other assessment would be available.

The goal of this overview is to provide an accessible guide for psychologists to use LLMs to assess psychological constructs through language. We first present the history, significance, and development of the transformer-based LLM (Section I), explore the experimental design process (Section II), and consider important issues related to LLM ethics, implementation, and future directions (Section III). We also present helpful techniques, tools, and code. Included on the accompanying GitHub page are a coding-based tutorial on using LLMs for psychological assessment and files containing specific code examples for applying the techniques we describe in Section II. Although we strive for an introductory level of description, we use many machine learning terms that are essential for understanding and working with LLMs. For that reason, we also include a glossary Table 1. This table includes definitions as well as useful software packages where certain procedures can be performed.

Section I: Development of Transformer-Based LLMs

Language is central to human identity. Psychologists have long been interested in the relevance of human language expression for understanding a person (Sanford, 1942). The ability to use language to assess psychological constructs was significantly bolstered by the development of word-counting programs (Pennebaker & King, 1999). "Dictionaries," the backbone of this technique, use scoring rules derived from expert-ratings of words to score psychological constructs from text. This method is also known as a "bag of words" approach. The Linguistic Inquiry and Word Count (LIWC) software (Pennebaker et al., 2003) uses a bag of words approach to count word use in text documents and score psychological constructs. It was developed as a simple text analysis program that has continued to be refined since its inception, with versions continuing to be released (Boyd et al., 2022). LIWC provides scoring of various emotion and cognitive process categories in addition to grammatical and language use categories from text. It has become the most influential text analysis program in psychology, demonstrating the ability to shine light on attention, emotion, social, thinking, and personality processes from language (Tausczik & Pennebaker, 2010).

However, early statistical language processing efforts struggle with language tasks because human language can be ambiguous, with rule exceptions and meaning changes across contexts (Johri et al., 2021; Khurana et al., 2023; Rosenfield, 2000). Initial models had a finite set of rules, inflexible decision-making algorithms, and were unable to understand linguistic nuances. Further, it was impossible to write rules and meanings for every scenario.

"Word embeddings" became an important solution (Almeida & Xexéo, 2019). Word embeddings are lists of numeric values (i.e., word vectors) that represent the meaning of words across multiple dimensions, capturing semantic and syntactic connections between words. Early

models used two main strategies to generate word embeddings: 1) Prediction-based models (e.g., Word2vec; Mikolov et al., 2013) generate word embeddings by predicting a target word from context words (i.e., words immediately surrounding it), or by predicting context words from a target word. 2) Count-based models (e.g., GloVe; Pennington et al., 2014) generate word embeddings through counting global word co-occurrence in a text body. These early embedding models drastically improved the ability of computer programs to understand language, but the embeddings were static—that is, each word had only one embedding (Almeida & Xexéo, 2019). This was a problem because words with changing, context-dependent meanings would have the same word embeddings, regardless of how the word was used in a particular instance.

Recurrent neural networks (RNNs) and long short-term memory networks (LSTMs), are deep learning neural network frameworks for NLP. While they process static word embeddings, they update with word context, appropriately mapping words to different possible meanings based on surrounding words (Khurana et al., 2023; Johri et al., 2021). RNNs and LSTMs improved model performance because they are better at maintaining accuracy across changing contexts. These models no longer followed pre-determined rules and instead developed dynamic algorithms for decision making that could update with greater exposure to language samples (Johri et al., 2021). While these updated models outperformed previous methods, they required exposure to large amounts of data to learn words in different contexts. RNN and LSTM models are also limited in efficiency because they process language one word at a time, leading to long training times. These models require significant computational resources and still struggle to maintain understanding of word context over text that is longer than one sentence (Min et al., 2023; Vaswani et al., 2017).

The transformer model architecture, which was the foundation for the development of LLMs, can provide a context-specific, quantitative representation of language (Vaswani et al., 2017). The transformer was a significant advance largely due to its unique "self-attention" mechanism. Self-attention allows the model to process all words in relation to all other words in a text sample simultaneously, as opposed to older methods that used sequential attention (Figure 1). Sequential attention could lead to information build up and forgetting of information that came earlier in a text sample. In the transformer, since all words communicate with each other directly, relations between words can be more accurately captured and retained across longer lengths of text.

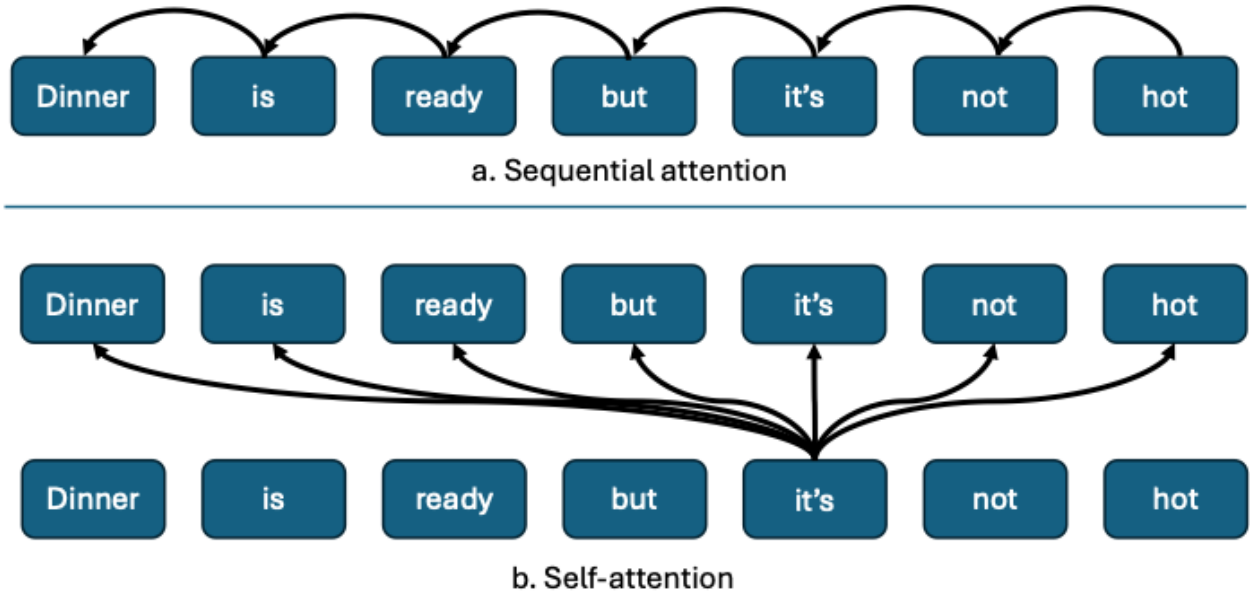


Figure 1. a) Method used by previous NLP models to process text. Each word processed individually. Model would initially perceive pronouns refer to "dinner" before processing is complete. b) Method used by transformers' self-attention mechanism to process text. Pronoun references are clearly understood.

A transformer model is a deep learning model that generally consists of "encoders," and "decoders" (Vaswani et al., 2017). But transformer models can vary in their composition of

encoders or decoders. Both encoders and decoders consist of self-attention “layers” which help transformers generate contextualized representations of input text (i.e., how the tokens in the text relate to one another). Encoders consist of a self-attention layer and a feed-forward neural network. Input first goes through the self-attention layer where relationships between each token and every other token in the sentence are learned. Multiple layers of encoders with similar architecture can be “stacked,” meaning input is processed through multiple encoder layers sequentially, which allows the model to capture more complex patterns. A decoder processes the output from the encoder and also has attention and feed forward neural network layers. The decoder’s attention layer is referred to as an “encoder-decoder attention” layer and it helps the decoder focus on relevant parts of the input sequence from the encoder. Since the decoder is used to generate text, its self-attention layer uses masking, where the tokens on the right side of the sequence are masked so that decoder cannot see future words of the sentence it is learning to generate. This prevents the model from knowing future tokens and constrains it to focusing only on preceding tokens to generate new text. Similar to encoders, decoders can also be stacked.

The advances in contextual understanding and speed provided by the transformer architecture enabled the creation of LLMs. The transformer allows the processing of massive amounts of data for training, often from online repositories. Initial transformer models were developed with a variety of large datasets for the time (Hadi et al., 2024). For example, Google’s Bidirectional Encoder Representations from Transformers (BERT; Devlin et al., 2019) was pre-trained on English Wikipedia and BooksCorpus (11,038 free books from the web). This training

process allowed the development of a model with millions of parameters for identifying words and thousands of embeddings, which give the model a general understanding of language.

In subsequent years, advances in computing resources have enabled the size of language models to grow (Hadi et al., 2024). Where initial transformer models were trained with millions of parameters (totaling less than 200GB of storage), models are now being trained on hundreds of billions of parameters (requiring over 7TB of storage), resulting in more powerful and versatile language models. While the term "LLM" is formally used to describe these newer, larger models, in this paper we use LLM to include the initial transformer models as well.

Broadly, transformer-based models can be divided into three types: encoder-only, decoder-only, and encoder-decoder. Tasks where input text needs to be understood to generate output text requires encoder-decoder architecture—For example, language translation (translating text from one language to another), summarization (distilling texts to only the main points, reformatting language [e.g., speech-to-text]), and question-answering. Here, an encoder changes the input text into a numerical representation while considering context of the text and a decoder uses the input numerical representation to generate the output text one token at a time (this is also called autoregressive text generation). LLMs like T5 (Text-to-Text Transfer Transformer) and BART (Bidirectional and Auto-regressive Transformer) are encoder-decoder models.

Encoder-only models are used in the scenarios focused on understanding input text to perform tasks like text classification (sorting language into categories), named entity recognition, sentiment analysis or retrieval tasks. Models such as BERT (Bidirectional Encoder Representation from Transformers), RoBERTa (Liu et al., 2019), and DeBERTa (He et al., 2020) fall

under this category. Decoder-only architectures are very popular and used for generative tasks where responses are predicted one token at a time. This architecture is used for large-scale generative models like GPT (Generative Pre-trained Transformer). Decoder-only models pre-trained on large text can perform generative tasks like summarization, question-answering and sentence completion. To that end, most transformer models can be used for more than one language task.

Section II: Experimental Design Process

LLMs show immense promise for psychological research and measurement, yet using these models remains complex and is often made more difficult by a lack of documentation for specific uses. This section outlines the experimental design process from start to finish and identifies relevant considerations at each step. This overview emphasizes details specific to NLP and the use of LLMs, but an understanding of general machine learning (ML) is also required for carrying out such analyses. While brief definitions of relevant terms are included in Table 1, we will also recommend helpful articles to explore these topics in more detail. Figure 2 presents a road map of the process. In each section, we discuss relevant concepts and decision making considerations, as well as provide examples from different areas of psychology and a continuous working example from research assessing Big Five/Five-Factor Model (FFM) personality traits from interview language (Oltmanns et al., under review).

In our working example, a representative community sample of N = 1,409 older adults was recruited from the St. Louis, Missouri area. The mean age of the sample was 59.5 years old, 54.5% identified as female, 65% identified as White/Caucasian, 32.7% Black/African-American, 2.3% other, and 1.7% reported Hispanic/Latino descent. Participants completed life narrative

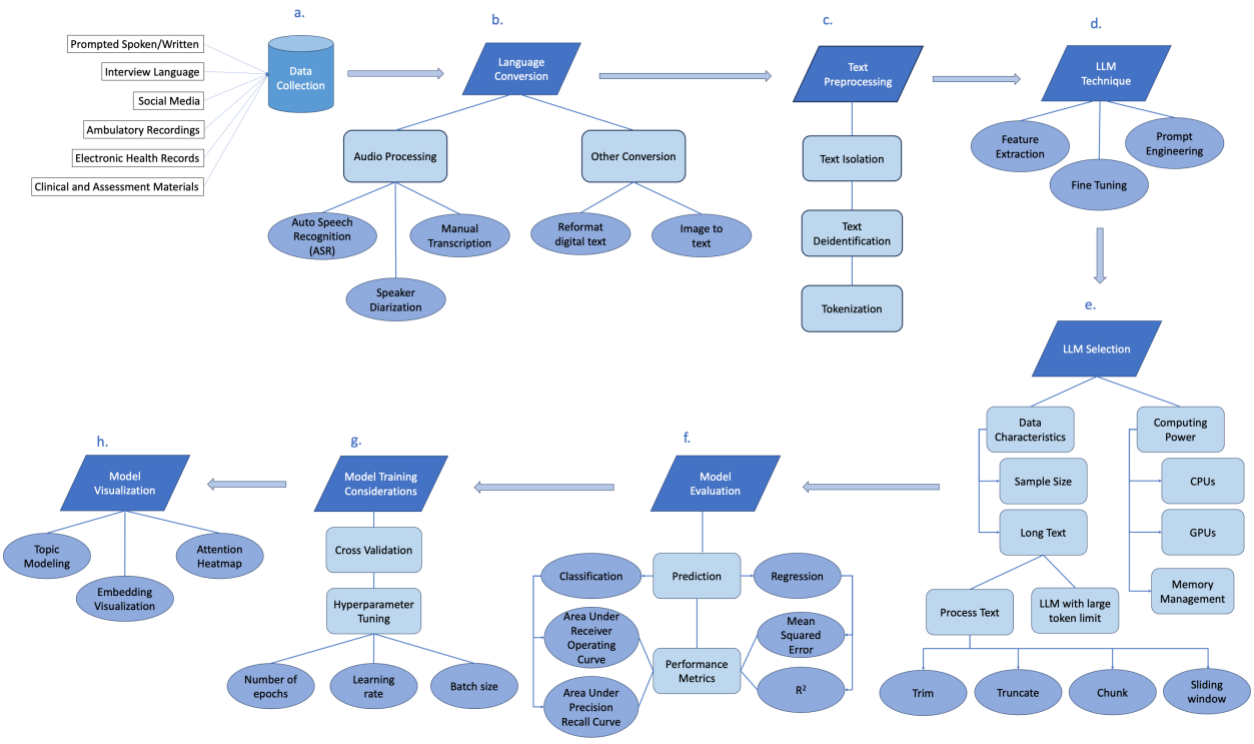
interviews in which they were asked to divide their adult life into three or four chapters, title their chapters, and then briefly describe those chapters. Next, participants were asked about high and low points, best and worst characters, and a turning point in their life story. Interviews lasted about 20 minutes, on average. Participants then completed the self-report NEO-Personality Inventory-Revised (NEO-PI-R; Costa & McCrae, 1992), from which five broad personality trait domains were scored (neuroticism, extraversion, openness, agreeableness, and conscientiousness). The NEO-PI-R scores were used to train language models of personality from the life narrative interviews.

Throughout Section II, we include "pseudocode" to show how python code may be used to complete certain steps. Each pseudocode is numbered with sequential steps for a given task. The names of snippets containing actual python code that go along with the pseudocode are included in parentheses next to the pseudocode titles and are located in the accompanying GitHub repository (link: https://github.com/mehak25/Intro-to-LLM). This repository also includes a hands-on coding tutorial on using LLMs.

The first decision in the process is what the researcher would like to ultimately predict or classify. This should influence data collection (Figure 2a). In our working example, we collected life narrative interviews to examine whether individuals' patterns of language use in storytelling could reliably predict personality traits. At each stage of Figure 2, there are important decisions to be made.

Figure 2

Overview of Experimental Design Process



Data Collection

We focus on several forms of natural language that show promise for psychological assessment with NLP (Figure 2a). Each type of language data has its own unique strengths and limitations in the data analysis pipeline. Ideally, multiple forms may be used in tandem to provide a more robust estimate and understanding of a psychological construct. Model prediction accuracy is heavily influenced by the quality and quantity of the data, making data collection and data preprocessing one of the most important considerations before the analysis process (Demszky et al., 2023). Importantly, models trained on one language sample type may not apply well to other language sample types and this will be a critical area of investigation in the future (c.f., Chekroud et al., 2024).

First, language may be collected through prompts. For example, recording verbal responses to prompted questions or written tasks. The process for collecting prompted

language can be self-administered, allowing participants to complete tasks without researchers and potentially in more comfortable locations. Collecting a sufficient amount of language from prompts can be difficult. Prompts to encourage speaking engagement such as asking carefully planned, open-ended questions, multi-part questions, or including explicit prompts or timers can be helpful to encourage continued speech.

Second, interviews capture an authentic and targeted language exchange between individuals. This may include job interviews, clinical interviews, group interviews, or life narrative interviews. It can be helpful to consider what kind of language would be most useful for future modeling purposes and how to encourage it in the interview. One potentially important downstream consideration for analysis is separating the interviewer and interviewee in audio files. It can be beneficial to record with multiple microphones, which makes it easier to separate the speakers downstream.

Third, social media posts are the most common form of natural written language being used for NLP (Chancellor & De Choudhury, 2020). Social media data often results in large sample sizes with short text length and would contain multiple status updates from individual users. Both Facebook and Twitter/X provide open-source Application to Programming Interfaces (APIs) to download large amounts of text[1]. APIs are tools that provide access to complex software programs or systems. Social media status language is unique—there may be topics an individual is more or less likely to post publicly about and there may be many participants and patients who do not use social media, which will impact the validity of the assessment.

[1] Step-by-step instructions to download text data from Facebook and Twitter/X are included in the supplemental material.

Fourth, ambulatory methods are used to collect more naturalistic and ecologically valid language data in everyday life (Mehl, 2017; Trull & Ebner-Priemer, 2013). Ambulatory recordings have several advantages: They can 1) have high ecological validity, 2) be collected multiple times over the course of a day, and 3) capture emotions and behaviors in real-time (Lazarevic et al., 2020). Ambulatory recordings are often implemented using smartphones, smartwatches, or other wearable recording devices. The Electronically Activated Recorder (Mehl, 2017) is available as a smartphone application that passively records speech in a naturalistic environment. If ambulatory data collection is active, it can be burdensome and uniquely difficult to collect.

Fifth, electronic health records (EHRs) are secure digital copies of patient charts including clinical notes from different settings, test results, and diagnoses. Extracting language data from EHRs can provide information on clinical treatment, professional opinion, and testing history that may reveal a significant amount about psychological functioning. For example, Yizhi Liu et al. (2023) used language models to find stigmatizing language in clinical notes to understand physician bias in patient assessment. LLMs can classify social determinants of health and behavioral health data from clinical notes in EHRs (Englhardt et al., 2024; Milligan et al., 2024).

Sixth, LLMs can assess language that was written by psychologists for professional purposes. For example, questionnaire items, vignette text, language from formal psychological testing measures, clinical diagnostic criteria and symptoms descriptions, intervention scripts, and research articles. This language can enlighten the test development process and examine coherence between clinical and assessment materials and human responses to these materials.

While clinical and assessment language is not natural language, use of LLMs to improve these materials and our understanding of them is promising.

Language Conversion

This section discusses several important considerations in processing audio or image files into text files for downstream NLP (Figure 2b).

Audio Processing and Transcription

After data collection, raw language samples need to be converted into formats better suited for analysis. This commonly includes transcribing speech samples from audio files to text but could also be reformatting digital language or transferring handwritten language to digital formats (Subramani et al., 2020). Conversion can be completed manually or through automated processes. Automatic Speech Recognition (ASR) requires much less time and fewer financial resources but is more likely to contain errors. Options include APIs such as OpenAI's Whisper, Google's Speech-to-Text, and Microsoft's Azure. The accuracy of automatic transcription tools has improved dramatically in the past few years (Spiller et al., 2023). These tools can be used on premise (i.e., implemented on a secure server at the researcher's institution), which is essential if the sample contains confidential information.

Speaker Diarization

When speakers on the same audio track need to be separated, this is called speaker diarization. Open-source diarization tools include SpeechBrain (Ravanelli et al., 2021), pyannote (Bredin et al., 2020), and WhisperX (Bain et al., 2023). These tools extract speech features from the audio signal, then uses deep learning models to differentiate between speakers based on

unique voice characteristics (e.g., variations in pitch, volume, vocal cord vibration). Diarization is still a difficult task to automate and often has errors, so manual review may be necessary.

Text Preprocessing

Successful transcription produces text files. However, further text preprocessing may be needed, for example to isolate language of interest or match the expected text formatting of an LLM (Figure 2c).

Text Isolation

Researchers might be interested in isolating language from one person. In interview transcriptions, speaker labels (e.g., “interviewer:”, “interviewee:”) are helpful. An example of speaker isolation is included in the GitHub repository. Other creative strategies can be helpful: For example, if the interviewer and interviewee need to be separated, the speaker who asks (or answers) more questions may be used as a proxy to identify them. Further, the speaker of interest may be identified by their use of specific words, phrases, or topics that they may be more likely to use.

Deidentification

Language data may contain confidential information that should either be deidentified or analyzed on a secure local sever (Hoory et al., 2021). Named Entity Recognition (NER) is an NLP technique that can de-identify text samples. NER locates predetermined words or phrases within text. For example, the open-source package spaCy (Honnibal & Montani, 2017) recognizes entities across eighteen categories such as names, organizations, geographic locations, and dates. Once identified, these words can be replaced with generic names. In our

working example, spaCY was used to de-identify transcripts and remove names of people, organizations, and locations.

Stop words

Stop words are commonly used words (e.g., “a,” “the,” “is,” “in”) that have traditionally been removed from language samples during pre-processing because their widespread use provided little unique information. LLMs capture contextual information from language, so they tend to work best when stop words are preserved (Shekhar et al., 2024) including contractions and all words and tenses.

Tokenization

Tokenization is the process of breaking down raw text into smaller units called “tokens,” which serve as input into the LLM (Zhao et al., 2023). Tokenizers split text into words and meaningful subword units. For example, the word “wind” remains [wind] after tokenization. However, “windsurf” becomes [wind] and [##surf], and "windsurfer” becomes [wind], [##surf], [##er]. Tokenization strategy varies by LLM, but most NLP packages make it easy to do. A brief code example of tokenization of a text using the Hugging Face transformers library is shared on the GitHub (called “tokenizer.py”).

Pseudocode (GitHub file: tokenizer.py):

1. ***Initialize pre-trained tokenizer.***
2. ***Loop over each word in your text to encode it into a token.***
3. ***Add special tokens like [CLS] to mark the beginning of the text and [SEP] to mark the separation between sentences.***

LLM Techniques for Psychological Assessment

Feature extraction, fine-tuning and prompt engineering are three primary ways to use LLMs for psychological assessment (Figure 2d). Each is explained in detail below, along with example applications in different areas of psychology.

Feature Extraction

One straightforward application of LLMs is to obtain contextualized embeddings from an input text. Unlike static embeddings, contextualized embeddings vary depending on how words appear in a sentence, thus capturing nuanced meaning specific to a given context. These embeddings can then be used in downstream analyses (Hussain et al., 2023).

For example, Wulff and Mata (2023) used an LLM to extract contextualized embedding features from the item language of multiple personality questionnaires. Results indicated that feature extraction can be useful for examining construct validity: Some questionnaires may claim to measure the *same* construct while the embedding features show *discrepancies* (e.g., "jingle fallacy"), while other questionnaires may claim to measure *different* constructs while the embedding representations are *similar* (e.g., "jangle fallacy"). Additionally, feature extraction has been used to support the validity of personality structure: LLM word embeddings related to personality show similar factor structure to that from previous research with human ratings (Cutler & Condon, 2023). Correlations were even stronger for the LLM embeddings than the previous ratings data, indicating LLMs may be an effective way to explore personality. Abdurahman et al. (2024) used contextualized embeddings from a pretrained LLM to represent the semantic meaning of items from self-report personality questionnaires. They then used these embeddings to predict individual's scores on previously unseen personality items based on linguistic similarity.

The following pseudocode demonstrates how to obtain and save embeddings after feeding text data through a pre-trained LLM. Once generated, the embeddings can be used as conventional predictor variables in traditional regression models (e.g., linear regression):

Pseudocode (GitHub file: save_embeddings.py):

1. ***For each text window, tokenize the text and then pass it through the model.***
2. ***Retrieve the output of the last layer of the model. The output of the language model is three-dimensional (number of samples, number of tokens, embedding size).***
3. ***Obtain document-level embeddings by either taking the mean of embeddings across all tokens or extracting the embedding associated with the first token (typically the [CLS] token).***

Fine-Tuning

The process of further training pretrained LLMs with more specific data is called fine-tuning. During fine-tuning, model weights are updated to reflect domain-specific language (e.g., language of interest to the psychologist) and adapt model decisions to best fit a specific task (e.g., predicting or scoring a personality trait, as in our working example). During fine-tuning, model weights can be updated either for the whole model or partial model depending on how much computational power is available for training ("training cost"). Fine-tuning is helpful for creating specialized models without the burden of needing very large, labeled datasets (Chae & Davidson, 2023; Demszky et al., 2023). To reduce the training cost of fine-tuning, a few samples can be used to update model weights which can be called few-shot fine-tuning. Few-shot fine-tuning is powerful because performance can be improved with much fewer data than would be

required to train a model from scratch. There are some challenges with fine-tuning: It is all the more important to have high-quality labeled data when fine-tuning. That is, the construct validity of the label measurements will be critical because the model will only be as good as the labels (that is, the quality of the assessment of the dependent variables). Also, fine-tuning is computationally expensive, requiring LLMs to be hosted on large servers to run the training cycle.

Fine-tuning is particularly well suited for assessing psychological constructs from language data. During fine-tuning, a pretrained language model's parameters are updated to reflect how language relates to the construct of interest, including subtle or nuanced patterns that might be imperceptible to human raters (Luxton, 2014). The resulting fine-tuned model can then accurately assess the construct from new, previously unseen language samples. Simchon et al. (2023) fine-tuned a model predicting personality traits from social media posts. The model was able to identify language patterns indicative of FFM personality traits and use that information to predict the personalities of new users. Our working example also uses fine-tuning to predict personality traits from interview language that does not explicitly ask about personality.

Fine-tuning is also being studied in clinical and social psychology. Ohse et al. (2024) used fine-tuning for depression assessment. The researchers fine-tuned BERT and GPT 3.5 using language responses to a depression interview with twelve labeled examples (i.e., interview transcripts with the corresponding depression scores). They evaluated the models using the F1 score, which is the harmonic mean of model precision and recall. Fine-tuned GPT 3.5 outperformed fine-tuned BERT in the prediction of depression from language in interviews (F1

score = .82 versus .62). In social psychology, fine-tuning was used to assess political beliefs from social media posts (Gül et al., 2024). GPT 3.5, Llama 2, and Mistral LLMs were fine-tuned to predict user alignment with political figures and stances (e.g., climate change, feminism). Fine-tuned GPT performed the best, with F1 scores all over .80, and at times exceeding .90. In cognitive psychology, LLMs have been used to explore the connection between cognitive abilities and behavior (Hardy et al., 2023). Fine-tuned LLMs may eventually be useful tools for study of cognitive processes such as memory, attention, perception, reasoning, and learning.

As model size continues to grow, the cost of traditional fine-tuning continues to increase beyond available resources. This has been addressed through parameter-efficient fine-tuning (PEFT; Lialin et al., 2024). PEFT is a broad term referring to any strategy that updates only a small set of model weights (e.g., a subset of existing model weights). PEFT strategies continue to be developed, with many demonstrating success compared to traditional fine-tuning. For example, Lin et al. (2024) trained a model using PEFT to generate positive alternatives to cognitive distortions, and their PEFT model outperformed other models.

In our working example—fine-tuning a model to predict FFM personality ratings from interview language—embeddings are updated to reflect nuances of the language used by the interviewee, and associations between the language and levels of personality ratings are learned. After training, the model can be used to predict personality ratings from unseen interview language (future research will have to assess the generalizability of the language sample that the model can be used with). See example code relevant for fine-tuning in the GitHub and the upcoming Model Training Considerations subsection.

Prompt Engineering

Prompt engineering involves carefully designing input text (“prompts”) to guide the output of LLMs, enabling improved performance without updating or retraining the model weights. It can be used to perform tasks or generate new text. Text prompts can be manually provided by the researchers to pretrained LLMs to either classify input text or perform a specific task. These prompts are considered “hard prompts,” as they are specific directives given to the model. Hard prompts are used when output text needs to strictly adhere to certain criteria (e.g., provide a specific assessment score, summarize text, or provide another response). Instructional tokens can also be used in hard prompts by adding them at the beginning of the input sequence. For example, if you want the LLM to provide a factual answer to your question, you can prepend your question with the instructional token “[ANSWER-QUESTION]” (e.g., “[ANSWER_QUESTION] What is the capital of United States?”). Interacting with LLMs through hard prompting can be simple, intuitive, and less computationally intensive.

Prompting has different strategies based on the amount of available labeled data. Zero-shot prompting prompts a model with instructions for a task but will not provide example data or example answers for the model to learn from. One-shot prompting prompts the model with instructions and one labeled example before the model completes the task, and few-shot prompting includes multiple labeled examples in the prompt before the model provides its response. The key difference from fine-tuning is that the labeled examples included in prompts do not modify any model parameters. When model weights are not updated, there is no computational training cost.

Another strategy is "soft prompting,” which is most used in a supervised learning context. Soft prompting requires model training and therefore may be referred to as prompt

"tuning." The soft prompt is a set of trainable embeddings that are added to the input text. The embeddings from the soft prompt are trained with labeled examples. These embeddings then act like a filter, cuing the model as to what language is associated with the task. Soft prompting is less computationally intensive than fine-tuning because only the added prompt embeddings need to be updated. Peng et al. (2024) compared hard prompting and soft prompting when identifying adverse events and social determinants of health from clinical narratives. Soft prompting performed better than hard prompting, indicating that LLMs can learn better from trainable soft prompt embeddings than human-generated hard prompts. Soft prompting reduced computing costs by 97% compared to fine-tuning. However, large models with several billion parameters were required for soft prompt models to show these benefits.

In prompt engineering studies, the prompt can vary for each case in the dataset to improve results or better study individual differences. For example, K. Yang et al. (2024) used LLMs to assess social attitudes and the propensity to be influenced by social contexts based on demographics (e.g., age, race, location, income, education level). The model performed poorly in zero-shot prompting. However, few-shot prompting that included labeled examples customized to match certain profile features for each individual improved performance.

Overall, prompt engineering allows for model customizations without the same data and resource requirements as fine-tuning, making it quicker (Chae & Davidson, 2023). In contrast to fine-tuning, model parameters are not updated, which is the most significant concern about prompt engineering. This is because psychology uses require generalizable, nuanced knowledge about a topic (Demszky et al, 2023). However, as the barriers to fine-tuning continue to grow for

the newer, more advanced LLMs (e.g., model size, closed-source), prompt engineering has become an exceedingly popular and effective strategy (Hua et al., 2024).

Prompt engineering has been applied across a variety of psychology domains. In cognitive psychology, GPT-4 predictions were compared to human memory performance (Huff & Ulakçı, 2024). GPT-4 was prompted to rate the relatedness of pairs of (1) context and (2) garden path sentences and the memorability of the garden path sentences. GPT-4 ratings of memorability significantly corresponded with human memory performance. This indicates LLMs may have utility as cognitive assessment tools in the future. In personality psychology, zero-shot prompting was employed to assess personality traits from social media posts (Peters & Matz, 2024). The LLM was hard prompted to attend to how personalities were reflected in language from online posts and to provide a numerical rating for each of the FFM personality traits. Results demonstrated moderate effect sizes for predicting personality.

Zero-shot prompting of GPT-3.5 has been used to assess attitudes in social psychology (Simons et al., 2024). Hard prompts were used to obtain GPT ratings on individuals' attitude certainty, importance, and moral conviction from social media posts. It was found that the GPT ratings replicated prior factor analytic structure and internal consistency reliability of human attitude ratings. This study was notable for its adherence to a psychometric construct validation approach for evaluating LLM-generated ratings based on language.

In clinical psychology, Tu et al. (2024) used zero-shot and few-shot prompting for PTSD assessment from language in clinical interviews. GPT-4 performed best with few-shot prompting and zero-shot prompting performed best with Llama-2. Predicting several different variable types from several different interview types, GPT-4 was on average 10% more accurate than

Llama-2, reaching an accuracy of 68%. GPT-4 showed close similarity to human ratings for PTSD-related scale variables, and more conservative predictions, while Llama-2 consistently over-predicted. Jeon et al. (2024) used a two-step prompting strategy to identify suicide risk from social media posts. In the first step, MentaLlama (Llama, fine-tuned on social media data related to mental health) was assigned an expert identity, provided a dictionary with suicide related terms, and asked to extract key phrases from the posts. They found that few-shot prompting in step 1 performed better than zero-shot, so a few labeled examples were added to the prompt. In the second step, a more generic LLM was prompted to summarize key phrases, and multiple summaries were evaluated for consistency. Recall of suicide-related posts was consistently high. Different expert identity assignments were found to influence the extracted phrases, indicating prompting LLMs to have different roles may produce different results.

Some research uses both fine-tuning and prompt engineering for psychological assessment. Galatzer-Levy et al. (2023) conducted zero-shot prompting with an LLM that had previously been fine-tuned on sources of medical language. The fine-tuned model was prompted to assess psychiatric functioning from clinical interviews and performed particularly well for depression detection but displayed difficulties with co-occurring diagnoses. Lin et al. (2024) combined and compared tuning and engineering strategies for two tasks in a Mandarin Chinese dataset: (1) detecting cognitive distortions (i.e., problem thinking styles related to depression) and (2) generating positively framed alternatives. Comparison of fine-tuning a pretrained language model versus transfer learning found fine-tuning was more accurate in detecting cognitive distortions. The researchers then compared fine-tuning, prompt tuning (P-tuning v2), and prompt engineering for generating positive alternatives to cognitive distortions.

The prompt-tuned model (ChatGLM-6B with soft embeddings) outperformed both the fine-tuned model and prompt engineering at generating positively reframed sentences. These findings suggest that prompt tuning a smaller model can be more efficient than fine-tuning or prompt engineering for generating psychologically meaningful text.

Hard prompts can be provided to the LLM with or without examples (e.g., text and variable score pairs). Below is an example of a hard prompt, which can be enhanced with additional instructions such as specifying a perspective or task:

> Language: [include the text here]
>
> Based on the text, please rate the level of [construct of interest here] by providing a numerical score [insert scale here].

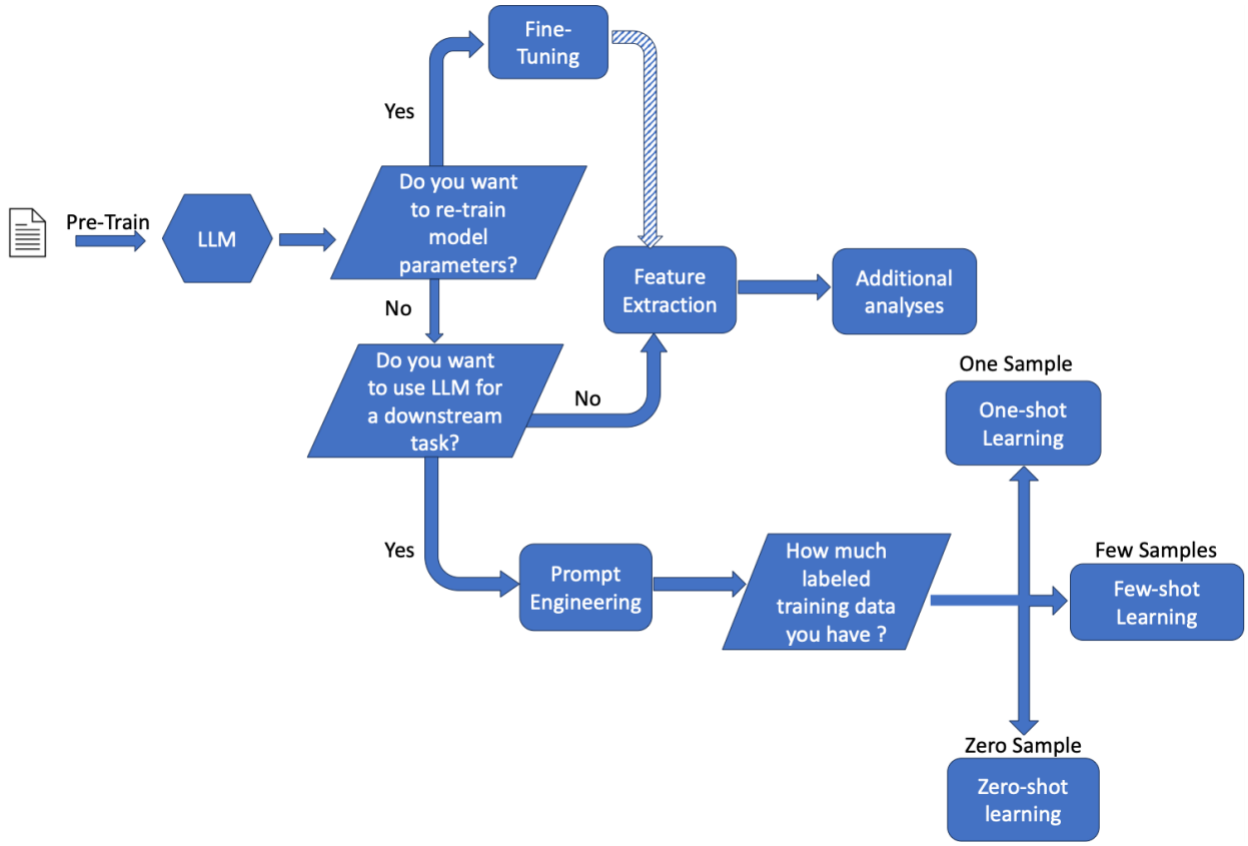
Soft prompts, in contrast, involve prepending trainable embeddings to the model input. The following pseudocode demonstrates how to prepend soft prompts to language input embeddings:

Pseudocode (GitHub file: soft_prompt.py):

1. ***Create soft prompt by giving prompt length and model embedding size to nn.Parameter package in PyTorch.***
2. ***Prepend the soft prompt to the input of the model.***
3. ***Train the model using updated input.***

Figure 3

Flow Chart of Techniques for Using LLMs for Psychological Assessment



Note. Striped line = optional.

Processing Labels

There are several important considerations for processing psychological variable labels (i.e., dependent variables) that may be predicted using LLMs for psychological assessment. Detail on these considerations including merging text data with psychological variable data, scaling of variables, splitting the dataset for training and testing, and avoiding data leakage are included in the online supplemental material.

LLM Selection

Key LLM selection decision points include their training data, text limits, size (measured in number of parameters and memory required to store the model), usage limits, and model transparency (Fields et al., 2024) (Figure 2e). It is becoming more common for models to have

"model cards," that provide this information in an organized fashion (Mitchell et al., 2019). Other important considerations in LLM selection include characteristics of the assessment data, task specifics (e.g., what you want the model to do), and computing resources. Table 2 describes common LLMs including their training data, model size, and text limits. Parameters are the building blocks of LLMs and include weights, biases, word embeddings, neural network layers, self-attention mechanisms, and feed forward neural networks. LLMs are classified as small if they contain less than one billion parameters, medium with 1-10 billion parameters, large with 10-100 billion parameters, and very large with over 100 billion parameters (Minaee et al. 2024).

Google's BERT (Devlin et al., 2019) is one of the earliest and most frequently used LLMs. BERT is a small, encoder-only model best suited for tasks requiring understanding of full text sequences such as text classification or NER. Additional BERT-based models continue to be developed such as RoBERTa (an optimized version of BERT using more training data and a longer training time among other training improvements; Yinhan Liu et al., 2019), DistilBERT (a slimmer, faster version of BERT; Sanh et al., 2019), and XLNet (which incorporates non-English languages; Z. Yang et al., 2019). While the term LLM generally does not include the initial transformer models mentioned previously, they remain a great option due to modest computing requirements and optimization for text classification.

Generative Pretrained Transformers (GPT; Achiam et al., 2023) are a family of decoder-only models by OpenAI that marked the transition to formal LLMs. These are very large models, containing more than 175 billion parameters, that are behind ChatGPT. Although prior GPT models have been publicly released, the most advanced models may be unavailable to the public. However, some can be fine-tuned through APIs. Another family of LLMs is the Llama

family by Meta (Touvron et al., 2023). Llama models range in size from medium to large and are open-source, meaning the model weights are available to the research community (Minaee et al., 2024). For more information about the structure of specific models, performance comparisons, and training considerations, see Minaee et al. (2024), Naveed et al. (2024), and W.X. Zhao et al. (2023).

LLMs are becoming increasingly accessible. Hugging Face is an open-source community that provides tool access (Hussain et al., 2023). Hugging Face has two main components: First, an online repository that stores trained language models, information regarding model performance, publicly available datasets, and detailed tutorials. Second, a series of python libraries that have simplified code to access transformer models, tokenizers, and optimization tools. Additionally, Hugging Face stores domain and task specific models previously created by others that are open to the public, for example BERT-based classification models trained on social media posts to predict sentences discussing anxiety or depression, Llama-based chatbots trained to provide empathic support and resources about mental health treatment, and RoBERTa-based models fine-tuned on PubMed papers.

Maximum Sequence Length

LLMs have varying maximum sequence lengths—also called context windows—which limit the number of tokens that can be input into the model at one time. If the token limit is exceeded, the text input will be truncated at the token limit, potentially cutting off important information. Some earlier models such as BERT, have relatively short limits (e.g., 512 tokens, which is around 400 words), while models such as GPT and Llama support context windows of several thousand tokens. Recently, some models have pushed these limits upwards of 200,000

tokens (e.g., Claude; Anthropic, 2024). While larger context windows may improve performance on long texts, they also significantly increase computational cost and memory requirements, leading to less common use in applied research to date (Y. Ding et al., 2024).

Currently, there are multiple other strategies to process longer texts (see Figure 4): 1) Truncate the text (i.e., discard all text that is beyond the token limit). This is the default strategy, so if long texts are not managed in other strategic ways, models will automatically truncate texts. 2) Trim the text (i.e., select portions of the original text to stay under the token limit). Research has shown that performance is better when tokens are selected from throughout the document rather than simply truncating (Tuteja & Juclà, 2023). 3) Chunk the text. “Chunking” is when the text is chunked into blocks of text that are within the token limit. For example, if the token limit is 512 and there are 1,536 tokens total, chunking the text would split the original long text into sthree chunks. The chunks can then be input to the model separately and the results are averaged across them. 4) Use a "sliding window” approach. In a sliding window approach, the original text is split into blocks that are below the token limit, but the blocks contain overlapping text that is referred to as the “stride.” This overlap helps preserve the context across chunks but will increase training time.

Other techniques may involve using one batch per document or hierarchical modeling. LLMs process data in batches, updating model parameters after each batch. Creating one batch for each long document enables the model to process one full document at a time. Hierarchical modeling techniques may also organize long texts into manageable chunks and ensure adequate aggregation of units into participant-level representations (Dai et al., 2022; M. Ding et

al., 2020; Wu et al., 2021). This may address the concern of text-participant attribution and can help with equal weighting of text samples when some participant texts are longer than others.

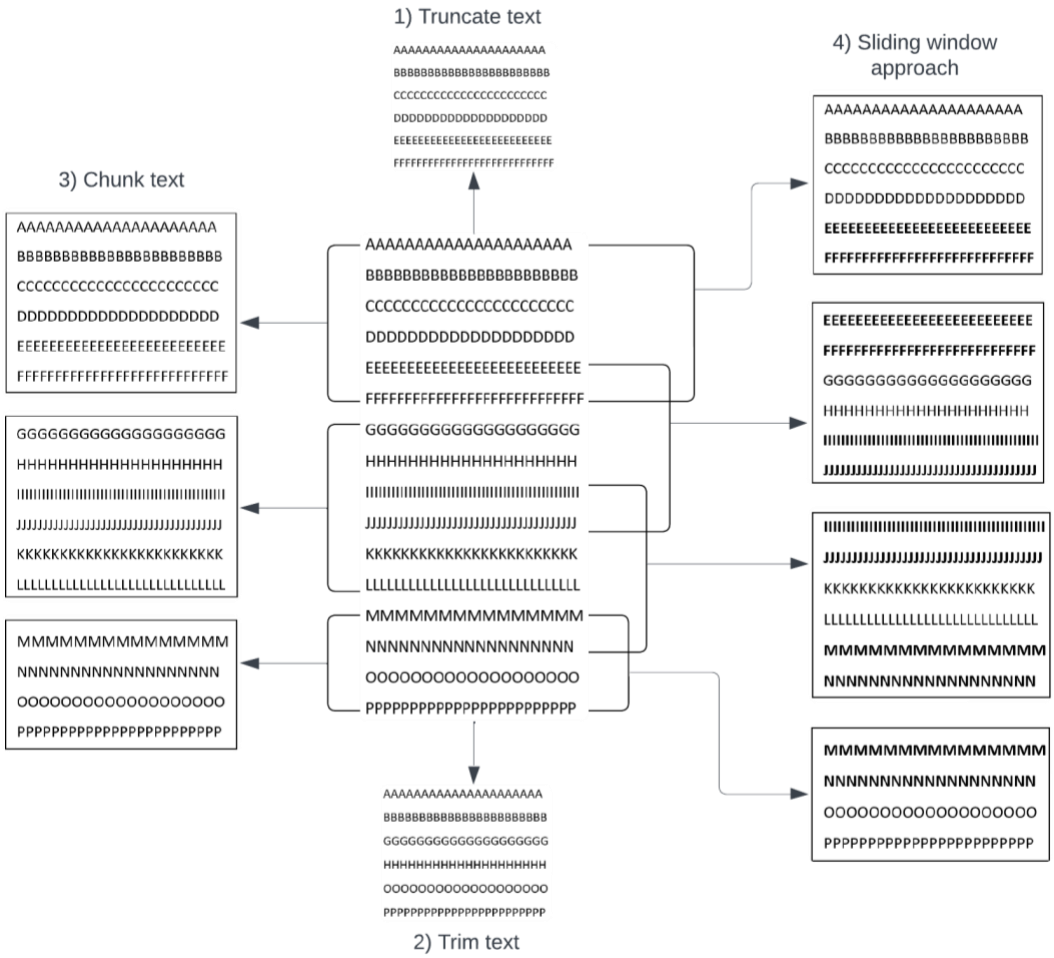
The following pseudocode example demonstrates how to implement the sliding window approach:

Pseudocode (GitHub file: sliding_window.py):

1. ***Loop over the text and divide text into sub-texts of length of window.***
2. ***Use overlap variable to decide how much overlap to keep between sub-texts.***
3. ***Tokenize each sub-text using a new or pre-trained tokenizer available on hugging face or simpletransformers.***

Figure 4

Strategies for handling long text



Note. 1) Truncate text that is longer than the token limit. 2) Trim selected text so it is shorter than the token limit. 3) Chunk the long text into segments the same length as the token limit. 4) Split long text into segments shorter than the token limit, with each segment overlapping.

Required Computing Resources

Computing resources are of the utmost importance (Kaddour et al., 2023). Small LLMs can be run using the central processing unit (CPU) of any computer, but many require graphics processing units (GPUs). GPUs are computer processors that were originally designed for video gaming that perform parallel computations and process large data quickly, making them well suited for machine learning and working with LLMs. Baseline GPU memory requirements for fine-tuning LLMs can reach upwards of 80GB (also the size of the largest commercially available

GPUs; Tuggener et al., 2024). To estimate how much memory is required, a rule of thumb is 8 * X, where X is the number of billion parameters of the LLM being used, stored in a 16-bit format (Mittal, 2024). There are public platforms and cloud servers that allow researcher access to GPUs (e.g., Google Colab offers free access to GPUs with 16GB of memory and paid access to GPUs with 40GB of memory). Microsoft Azure Machine Learning offers paid GPU access to a wider range of power and memory configurations than Colab. Institutions may also have shared access to more powerful GPU computing resources.

In our working example, we used cloud servers and university-based computing resources. On-demand access to cloud servers was helpful, while university-based computing was more cost effective and helpful for batch job processing. Even with a small language model, running the fine-tuning analyses required over 30GB of GPU RAM.

Managing memory usage is also critical for working with LLMs. We explored strategies to reduce both static and dynamic memory requirements, including precision reduction, data streaming, gradient checkpointing, and mini-batch optimization. A detailed discussion of these strategies along with implementation examples is provided in the online supplemental materials.

Model Evaluation

Models must be configured during training to produce the desired output (Figure 2f). In NLP tasks, language-based predictions generally fall into two categories: classification and regression, each with its own evaluation metrics (Berggren et al., 2019). Language can be used to predict a binary classification (e.g., does someone have a specific attribute, yes or no?), multiclass classifications (e.g., a set of possible labels), or continuous values (e.g., a ratio score).

Multiclass labels can be nominal (e.g., predicting one of five political affiliations) or ordinal (e.g., predicting one of four increasing difficulty levels). Models can also be trained as multi-label classifiers, where multiple labels can be selected for each language sample. Lastly, a regression task trains models to predict continuous values.

Classification and regression tasks are evaluated using different metrics. Classification evaluation metrics are focused on prediction accuracy. Regression-based metrics are focused on reducing prediction error. Descriptions of evaluation across different metrics can be further studied in tutorials by Vickers et al. (2024) and Pargent et al. (2023).

Most documentation about language modeling uses the term “language classification” to describe the broad category of tasks mentioned above (including regression). Most available information refers to classification tasks and not regression tasks. For some tools, such as simple transformers (Rajapakse, 2019), the default information will address classification tasks, but steps to convert the code to regression are included in the documentation. In some cases, information about classification will still apply to regression, as a regression task can be conceptualized as a classification task with one label. In general, classification tasks tend to achieve better overall performance, but regression tasks offer more precise predictions and are often more relevant to psychological constructs, which are often measured continuously.

In our working example, the personality scores were continuous, and the model was trained to complete a regression task. A model can be configured for regression using the simple transformers library by setting the regression parameter to True in ClassificationArgs, as shown in regression.py on Line 55. Line 80-81 shows how to extract the predictions of personality ratings during testing.

Model Training Considerations

Analyzing text data with LLMs relies heavily on general machine learning procedures (Figure 2g). Pargent et al. (2023), Choi et al. (2020), Badillo et al. (2020), Jiang et al. (2020), and Pandey et al. (2020) are helpful overview articles and tutorials. Coursera (https://www.coursera.org/) and Towards Data Science (https://towardsdatascience.com/) are also practical resources for examples, tutorials, and discussions.

Cross Validation

Cross validation is a technique used to estimate model reliability and accommodate limited amounts of data (Yates et al., 2023). The data are divided into equal portions or "folds". The number of folds may vary (referred to as "k"), with five or ten being the most common. K minus one folds are used to train the model and the remaining fold is used to test the model. This process is repeated until each fold has been the testing fold. The overall estimate of model performance is the average of all combinations (Wong, 2015). For smaller datasets, leave-one-out cross validation is recommended (see Table 1). Cross validation is important because it provides a more reliable estimate of model performance, reducing bias from randomness in the data. Variability in performance across iterations can indicate inconsistencies in the data, increased data complexity, or difficulties with the model's ability to learn (Shulga, 2018). In our working example, we utilized 5-fold cross validation to help estimate model performance.

Hyperparameter Tuning

Hyperparameters are settings that affect how a model learns and are they are adjusted to optimize model performance. Customizing these settings is known as hyperparameter tuning. There are many hyperparameters. Three are emphasized as having the greatest impact: learning

rate, batch size, and number of epochs (Devlin et al., 2019). The learning rate determines how much the model's parameters are adjusted in response to training examples. Higher learning rates may speed up the training process but may not result in optimal model performance because they may not be sensitive enough. Lower learning rates remedy this problem but will slow down the training process. Specifically for LLMs, learning rates tend to be much smaller than with other machine learning models, as LLMs operate best with subtle adjustments. Learning rate warm-up strategies are also useful when training LLMs because they gradually increase the learning rate at the onset of training, facilitating stability. Batch size is the number of data samples that are seen by the model before calculating errors and updating the parameters. Batch size is dependent on available computing resources because all data for a given batch need to be held in memory before the model's weights are updated. Epochs are the number of times the model passes back and forth through the entire dataset. Training for too few epochs can result in underfitting, where the model does not learn enough about the data. Training for too many epochs can result in overfitting, where a model learns too much about the data and then does not perform well on other, unseen data.

When determining values for hyperparameters, it is recommended to begin with the same values used to train the base LLM (Devlin et al., 2019). These values are likely published, and some models (e.g., BERT) even recommend possible ranges for hyperparameter values for future fine-tuning. It is then important to experiment with different settings to determine what works best for a particular dataset. There are multiple strategies for finding optimal hyperparameter values, with grid search, automatic optimization, or random search being the most common (Bischl et al., 2023). A grid search will systematically train multiple iterations of

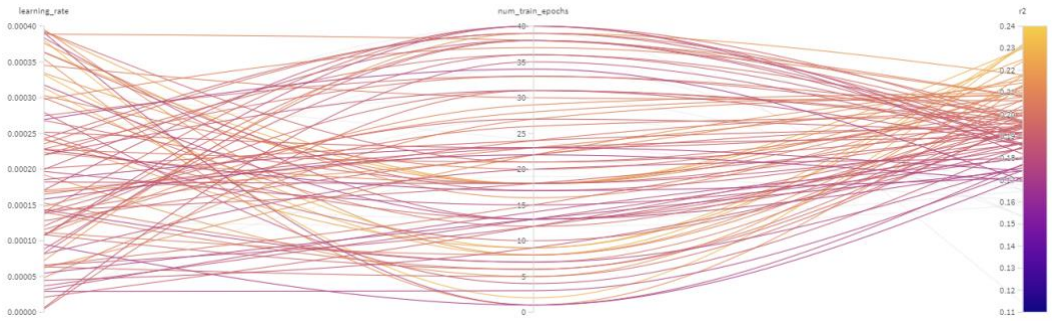
models, trying every combination of values within the given ranges. Automatic optimization strategies will dynamically adjust the values of specific hyperparameters each iteration, testing values that are uniquely promising, and using algorithms to predict what those values would be. Random search tests a wide variety of values within in a specified range, with no meaningful decisions about which values to try. It is important to note that the optimal settings for a given model may not fall in the recommended values range. This is where automatic optimization strategies can be helpful, as they are able to efficiently expand hyperparameter values away from the recommended ranges based on context specific information.

The tuning process can be time consuming and requires significant computing resources, as each combination of parameters is used to train the entire language model. Overfitting is a concern (X. Liu & Wang, 2021). Several strategies are recommended to avoid overfitting in hyperparameter tuning: 1) Early stopping prevents models from overfitting by determining the optimal number of epochs and ending the training process once the model's performance is no longer improving after a specified number of epochs, typically five to ten. (Dodge et al., 2020). 2) The optimal values are those that resulted in the greatest average performance across all validation folds—not the best values of any individual run. 3) Dropout and weight decay can reduce overfitting. Dropout randomly removes connections between elements of the model during training, and weight decay adds penalties to highly influential paths to encourage the model to examine patterns more generally (Srivastana et al., 2014). After the optimal hyperparameters are determined, the full model should be retrained using these values. See the GitHub page for important training arguments for hyperparameters and early stopping.

Weights and Biases is a helpful tool for hyperparameter tuning (Biewald, 2020). This software is free for students, educators and academic researchers and facilitates hyperparameter sweeps. Weights and Biases can be integrated with other libraries (e.g., Simple Transformers, Hugging Face transformers) to automatically log training and evaluation data in real time and visualize model performance. We ran hyperparameter sweeps in our working example with Weights and Biases. Figure 5 shows an example of a hyperparameter tuning log. The results of each combination of learning rate and epochs are plotted, indicating model performance with respect to different combinations of these hyperparameters. Note the range in performance across different combinations, providing useful information about optimal hyperparameter settings.

Figure 5

Example of Weights and Biases hyperparameter tuning log



Model Visualization

Deep learning uses nonlinear relations across multiple layers, which make it difficult to understand precisely how LLMs make decisions (this is known as the “black box” problem). Techniques are being developed to increase model decision explainability (H. Zhao et al., 2024). However, simple model visualizations can be helpful (Figure 2h). One simple method is to

correlate token usage from the text input with the target variable. Examining tokens that appear in more than 10% of the sample and selecting those with the highest positive and highest negative correlations is a straightforward approach to identify potentially important features.

Topic modeling is useful for providing insights into the content of language data and decreasing the manual labor required by exploring themes qualitatively. BERTopic (Grootendorst, 2022) is a python package for topic modeling that also harnesses the power of transformer models. It generates contextualized embeddings from the input text using Sentence-BERT (sBERT; Reimers & Gurevych, 2019), then reduces the dimensionality of those embeddings and clusters semantically similar documents together. BERTopic then identifies words that contribute most to the topics. Code for initiating BERTopic modeling and viewing topics is provided in the source materials (Grootendorst, 2022).

If working with longer language samples, it is helpful to split samples into sentence-level data when performing topic modeling to adequately capture the variation of topics discussed by one person. Because the narratives were so long in our working example, we reformatted the dataset to have utterances from participants in unique rows. Using BERTopic probability scores across participants, topics were correlated with personality traits. Many topic-personality correlations were face valid (e.g., extraversion with a friends topic, agreeableness with a community topic, and neuroticism with a mental health problems topic). Topics can be trimmed, for example removing topics that may reflect methodological artifacts of the text sample, but it is important to keep in mind how trimming the topics will affect future utility and/or replicability of the topic model.

It is also helpful to visualize embeddings in 2D space. t-Distributed Stochastic Neighbor Embedding (t-SNE; Van der Maaten & Hinton, 2008) is a dimensionality reduction technique used to visualize high-dimensional data, such as LLM embeddings, in 2-dimensional space. The relative positioning of data points in the visualization provides insight into semantic meaning similarity. CLS embeddings, in particular, are useful for visual inspection because they represent the embedding for the full text sample. The following pseudocode demonstrates visualizing embeddings in 2d space using t-SNE:

Pseudocode (GitHub file: embedding_visualize.py):

1. ***Extract CLS embeddings from pre-trained or fine-tuned model for each text in the dataset.***
2. ***Use t-SNE to transform embeddings into 2D space.***
3. ***Plot the scatter plot for all samples.***

Attention weights for each token in the input text can also be visualized (Vig, 2019). This provides information about the importance of each language feature for prediction of the outcome. Create a 2D matrix to visualize CLS tokens:

Pseudocode (GitHub file: attention_visualize.py):

1. ***Extract attention layers from the model output.***
2. ***Select layer and head for which to view attention weights (most commonly used 0^{th} layer and 0^{th} attention head).***
3. ***This will provide a square matrix as 2D array.***

A heatmap can illustrate how much attention (or weight) is given to each token in the input text to perform the output task. Create a heatmap of the above attention matrix which will show how each token is semantically connected to each other token in the input text:

Pseudocode (GitHub file: attention_visualize.py):

1. ***Extract attention layers form model output.***
2. ***Select attention layer and head to visualize.***
3. ***Visualize the attention matrix using heatmap.***

Section III: Important Issues for Consideration and Future Directions

In Section III, we discuss issues, implementation, and future directions that will be important for using LLMs for psychological assessment.

Ethical Considerations

LLMs contain *biases* that are prevalent in society and that researchers and the field at large should be aware of and prepared to continuously address in a transparent manner (Bender et al., 2021). Working with LLMs may involve sensitive data that need to be handled securely for the *privacy* and respect for research participants and patients. LLMs require significant energy resources that have a detrimental *environmental impact.*

Bias

LLMs can be conceptualized as “stochastic parrots” that lack human understanding of meaning. With some randomness, they confidently repeat back what they were trained on, which will include stereotypes and harmful biases that are prevalent in online training data (Bender et al., 2021). Training data from vast online samples reflect society at large. As a result, they will have negative biases against minority groups that can perpetuate harm. Research has

demonstrated bias in LLMs across gender, race, culture, and other demographics (Raza et al., 2024). For example, showing a preference for male pronouns for certain professions (de Vassimon Manela et al., 2021), indicating some religious groups are more violent than others (Abid et al., 2021), favoring majority groups (Zhang et al, 2020) and propagating differential treatment recommendations based on race (Omiye et al., 2023). These biases emerge when LLMs are trained on data that provide an imbalanced or inaccurate representation of a group or do not represent them at all. While LLMs contain bias, the level varies (Nadeem et al., 2020; Raza et al., 2024). Researchers may select LLMs based on fairness evidence. In the future, it may be beneficial to concentrate on specific representative training samples rather than simply collecting as much training data as possible (Bender et al., 2021).

It is unclear whether bias in LLMs can be eliminated. Without careful evaluation of bias in psychological research, they can be perpetuated and amplified. For example, LLMs trained on biased data may perpetuate job and financial inequality, amplify harmful content online, misdiagnose and influence clinician decision making in healthcare, and otherwise prioritize majority backgrounds (Ferrara, 2023). Techniques are being developed that may reduce model biases including data augmentation, bias correction algorithms, and fairness metrics (Cai et al., 2024; Liang et al., 2021; Raza et al., 2024; Sun et al., 2019). However, these techniques are not able to fully remove bias. Psychological researchers using LLMs for psychological assessment must be 1) aware of bias, especially that is directly relevant to their area of research, 2) active in ensuring fairness in model development (e.g., comparing model results and predictions across various groups), 3) transparent about the existence of biases in the models that they use (e.g., describing the biases and their potential influence on the results in Discussion sections), 4)

constantly updating their LLM use in accordance with the latest techniques to reduce harm, and 5) supporting or collaborating with researchers from minority groups, especially those that might be a focus of the research. Psychology research conferences should have regular panels with experts on LLMs to help spread awareness of bias and best use practices to manage bias in research. Together these strategies will help the field understand and mitigate bias, reduce the possibility of harm, and improve useful models.

Privacy

Text data is often more sensitive than questionnaire data and it is imperative to take measured steps to protect it. Research participants and patients should complete transparent consent forms with the potential risks and benefits and plans for data usage in accordance with APA ethical principles (APA, 2017). Data should be deidentified when possible. There should be a secure, password-protected and encrypted server where the data can be stored and only accessible to authorized personnel (who all have training in data security). Additionally, the server and its network can include a firewall to protect the data. Regular audits of the security system can be conducted to prevent data breaches.

At times it may be necessary to work with third party service providers. This must be completed in a manner that research participants and patients have consented to and is compliant with the relevant regulation authorities (e.g., Institutional Review Boards (IRBs), Health Insurance Portability and Accountability Act [HIPAA], General Data Protection Regulation [GDPR]). The minimum number of third parties should be involved in the process. When using APIs, connections should be secure, authenticated, and encrypted. Vendors will have compliance standards that should be reviewed.

Environmental Impact

Deep learning is computationally expensive. It requires significant power, which leads to a growing carbon footprint (Patterson et al., 2021). As a result, researchers are devising ways to train models more efficiently and reduce negative consequences such as excessive water usage and CO2 emissions (Rillig et al., 2023). The estimated energy usage for an analysis can be directly calculated, which can be helpful for planning efficient analyses (Hershcovich et al., 2022; Strubell et al., 2020). Researchers should be aware of the energy use that potential analyses would require and take steps to reduce unnecessary analyses. This may include reporting training times, using efficient computational hardware and models, and being aware of power resources used—for example by data centers and cloud computing services (Strubell et al., 2020). Researchers should also consider any potential *positive* downstream environmental impacts of a model (Hershcovich et al. 2022). The rapid increase of power required and used for training LLMs poses serious ethical dilemmas for researchers that should be understood, prioritized, and addressed in transparent ways moving forward.

Other LLM Limitations

LLMs will only generalize to the population in which they were developed. Researchers should strive for approximately equal representation for every group that a model should generalize to (Ntoutsi et al., 2020). This means continuing to emphasize the inclusion and collection of language from diverse groups. Of course, most models will not include an accurate representation of everyone. This must be acknowledged in model description materials and research articles. This will help prevent the use of models in groups for which the model may

not work or even produce harmful results. Recently, some psychology journals have required discussion section “generalizability statements” that are on par with this recommendation.

LLMs may also only generalize to the situation in which they were trained (e.g., interview, cognitive task, social media). Research should cross-validate models across contexts and models should not be applied across context without validation in the new context. Models built from text gathered in controlled environments may not apply to models using real life settings (e.g., ambulatory recordings). However, these generalizability questions will be exciting future research directions.

Token limits are currently limitations in working with LLMs. Early models had relatively smaller token limits, for example, 512 tokens. We have outlined ways to work with longer texts, but this is a primary area of future development. Newer LLMs have much longer token limits that could greatly facilitate LLM comprehension of longer texts. However, models with long token limits should be tested to ensure that they are in fact remembering (or properly maintaining) context across long texts. At the current time, managing token limits can be challenging, but simpler methods are likely to emerge as models continue to advance.

Interpretability and Explainability

LLMs use deep learning techniques that can be thought of as a “black box” into which we do not understand. The massive nonlinear complexity of the algorithms and layers in these models can make their decisions indecipherable to humans. This can cause problems for researchers and clinicians, as we should be able to justify research conclusions and clinical decisions. As a result, researchers must do what is possible to understand how decisions are being made.

Luckily, techniques exist to inform how LLMs are making decisions and predictions. Attention visualization is used to identify how neural networks are focusing their attention on tokens available to them. The differential weights placed on input tokens by the LLM in the process of their predictions can be examined as a heat map or text highlighting, showing the user which parts of the text are most important to the prediction that was made (e.g., Jeon et al., 2024). SHapley Additive exPlanations (SHAP) are an explainable AI technique based in game theory that attribute differential importance to the input tokens (Lundberg & Lee, 2017). SHAP values are often visualized in waterfall plots, which help researchers interpret the key token predictors of an outcome of interest. However, because SHAP requires repeated evaluations of a model with different feature combinations—and LLM-based analyses often involve extremely high-dimensional inputs—it may only be feasible in smaller-scale LLM applications.

LLM outputs should also be understood through traditional psychometric validation techniques. After a model is fine-tuned, for example, it may produce a predicted score for an outcome of interest. In the future LLM output scores should be validated just as psychological variables have been in the past, with construct validation such as convergent, discriminant, and criterion validity tests (c.f., Strauss & Smith, 2009). Nomological networks of the model output should be examined (e.g., what other constructs does it predict and what does it not predict?) (c.f., Cronbach & Meehl, 1955), helping us place LLM-based scores in the broader research literature. Reliability should be understood through tests of internal consistency and test-retest reliability (c.f., Simons et al., 2024). Construct validation techniques will provide an understanding of what LLM-based predicted scores represent just as they facilitated understanding of psychological scores in the past.

Humans and LLM-based Psychological Assessments

LLMs hold promise for the automation and augmentation of assessment methods; however, results still vary based on the task. Each use case should be validated against human raters to evaluate model performance. For example, Schoenegger et al. (2024) compared the abilities of lay persons, psychology experts, pre-trained LLMs, and a specialized AI model trained on personality data, to predict correlations between personality items. Results indicated AI models made better predictions than 85% of individual humans. However, median predictions from the whole group of psychology experts rivaled the specialized AI and outperformed those of pre-trained models. This suggests that LLM performance might be superior to most individual evaluators, yet experts still hold advanced knowledge collectively.

Given the limitations outlined above, LLM-based psychological assessments should not be relied on as standalone assessments in clinical or applied situations without human oversight. Humans should always have oversight and final judgment over any consequential decisions that might be made by an LLM. Ideally, they will be administered as a tool within an assessment battery of multiple measures. They are currently best considered a potentially helpful tool for understanding psychological phenomena.

Collaboration among psychologists, computer scientists, and others is essential for LLM tools to be as useful for psychological assessment. Professionals from each area have unique insight, questions, and ways of thinking about assessment and developing research projects. Reliance on team science will also reduce the burden on any one scientist to have mastery of all cutting-edge methodologies. Interdisciplinary data science PhD programs will be important for producing scientists who may help bridge the gap between disciplines. Profitable collaborations

will occur when professionals from different areas come together with a mutual respect and put in the time needed to work together efficiently and effectively. While this can be a challenge, effective interdisciplinary collaboration will be necessary to develop LLM-based psychological assessment methods that will be as effective as science and medicine will need them to be.

Researchers and clinicians who administer LLMs for assessment should have proper training in their effective use. Currently, we are not aware of any official guidelines or standards. It may be fruitful to develop guidelines that LLM-based researchers should follow to receive the proper training that will provide a foundational knowledge and essential skills that are needed to effectively engage in this area. Further, it would be especially useful to ensure this training provides the tools necessary for researchers to continue to grow their knowledge and stay aware of the latest best practices in the field throughout their careers.

Guidelines for the development and administration of LLM-based psychological assessments may also be helpful. Organizations such as the American Psychological Association have provided resources and updates on policy for AI generally (American Psychological Association, 2023). Researchers have also published useful guidance about ethical use of LLMs in science (Parker et al., 2023; Watkins, 2023). These protocols may include standards on transparency, data collection, management, privacy and security, bias mitigation, generalizability, training, and deployment. Organizations that may help develop these standards include research organizations, institutions, professional associations, publishers, or advocacy groups. Guidelines may help promote responsible practices and reduce potential harms. In conducting meta-analytic reviews, psychologists follow the PRISMA guidelines (Page et al.,

2020). Beyond universal best practices, we emphasize the importance of flexible guidelines that account for the unique context of model development.

Future Directions

There is growing evidence that multimodal models, including more than just language, improve predictive utility (Morales et al., 2018). It will be fruitful to pair LLMs with standard psychological assessments and other technologies to examine unique and combined predictive power across features (e.g., Harari et al., 2017; Jacobson & Bhattacharya, 2022). The transformer model can also be used with non-language predictors (Wang & Sun, 2022). In the future, modeling features from video recordings in tandem with traditional psychological assessments will provide a more holistic assessment of a person.

LLMs will soon have longer context attention, better strategies to mitigate bias, better regulatory standards and guidelines, available training, and techniques for model explainability and interpretability, security, validation, and access. Alternative model architectures have already rivaled the transformer model in NLP, such as state space models (Gu & Dao, 2023). It is important for researchers to stay informed on these developments. We recommend following journals, new books, podcasts, online courses and webinars, attending conferences, and maintaining communication with interdisciplinary collaborators. Commitment to rigorous methodology such as the collection of high-quality data including well-validated assessments with useful and targeted language samples across diverse populations is also imperative. Consortiums with the purpose of bringing together researchers with similar interests in specific LLM applications may be useful to enhance data size and diversity.

Conclusions

LLMs offer important advantages compared to traditional psychometric approaches such as the self-report questionnaire. This includes their behavioral nature, scalability, and allowance for a broader range of response possibilities. Language assessments can be derived from routine tasks or in naturalistic environments using smartphones. Despite potential advances, there are significant risks and biases with this technology. Psychologists must be aware of the biases in LLMs and ways of mitigating them.

The purpose of this overview is to provide accessible guidance on a novel and complex methodology. Despite rapid advances, relatively little is known about using LLMs for psychological assessment. While a growing number of high-quality studies are emerging, many face limitations related to sample size, diversity, language data types, or psychological measurement. We encourage psychologists to strive for strong psychometric, methodological, and interdisciplinary contributions in the evolving area of using LLMs for psychological assessment, and hope this paper helps promote them.

References

Abdurahman, S., Vu, H., Zou, W., Ungar, L., & Bhatia, S. (2024). A deep learning approach to personality assessment: Generalizing across items and expanding the reach of survey-based research. *Journal of Personality and Social Psychology*, *126*(2), 312–331. https://doi-org.xavier.idm.oclc.org/10.1037/pspp0000480

Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., Almeida, D., Altenschmidt, J., Altman, S., & Anadkat, S. (2023). Gpt-4 technical report. *arXiv preprint arXiv:2303.08774.*

Almeida, F., & Xexéo, G. (2019). Word embeddings: A survey*. arXiv preprint arXiv:1901.09069.*

American Psychological Association (2017). *Ethical principles of psychologists and code of conduct.* https://www.apa.org/ethics/code

American Psychological Association. (2023, November). *Apa Journals Policy on generative AI: Additional guidance. American Psychological Association.* https://www.apa.org/pubs/journals/resources/publishing-tips/policy-generative-ai

Anthropic, A. I. (2024). The claude 3 model family: Opus, sonnet, haiku. Claude-3 Model Card, 1.

APA Task Force on Psychological Assessment and Evaluation Guidelines. (2020). *APA Guidelines for Psychological Assessment and Evaluation* (510142020–001). American Psychological Association. https://doi.org/10.1037/e51014202

Badillo, S., Banfai, B., Birzele, F., Davydov, I. I., Hutchinson, L., Kam-Thong, T., Siebourg-Polster, J., Steiert, B., & Zhang, J. D. (2020). An introduction to machine learning. *Clinical Pharmacology & Therapeutics*, *107*(4), 871-885.

Bain, M., Huh, J., Han, T., & Zisserman, A. (2023). Whisperx: Time-accurate speech transcription of long-form audio. *arXiv preprint arXiv:2303.00747.*

Bender, E. M., Gebru, T., McMillan-Major, A., & Shmitchell, S. (2021, March). On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency* (pp. 610-623).

Berggren, S. J., Rama, T., & Øvrelid, L. (2019, August). Regression or classification? automated essay scoring for Norwegian. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications* (pp. 92-102).

Biewald, L. (2020). Experiment tracking with weights and biases. Software available from wandb. com, 2(5).

Bischl, B., Binder, M., Lang, M., Pielok, T., Richter, J., Coors, S., Thomas, J., Ullmann, T., Becker, M., & Boulesteix, A. L. (2023). Hyperparameter optimization: Foundations, algorithms, best practices, and open challenges. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *13*(2), e1484.

Boyd, R. L., Ashokkumar, A., Seraj, S., & Pennebaker, J. W. (2022). The development and psychometric properties of LIWC-22. *Austin, TX: University of Texas at Austin, 10*.

Bredin, H., Yin, R., Coria, J. M., Gelly, G., Korshunov, P., Lavechin, M., ... & Gill, M. P. (2020, May). Pyannote. audio: neural building blocks for speaker diarization. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 7124-7128). IEEE.

Cai, Y., Cao, D., Guo, R., Wen, Y., Liu, G., & Chen, E. (2024). Locating and mitigating gender bias in large language models. *arXiv preprint arXiv:2403.14409.*

Chae, Y., & Davidson, T. (2023). Large language models for text classification: From zero-shot learning to fine-tuning. *Open Science Foundation*.

Chancellor, S., & De Choudhury, M. (2020). Methods in predictive techniques for mental health status on social media: a critical review. *NPJ Digital Medicine*, *43*, 1-11.

Chekroud, A. M., Hawrilenko, M., Loho, H., Bondar, J., Gueorguieva, R., Hasan, A., Kambeitz, J., Corlett, P. R., Koutsouleris, N., & Krumholz, H. M. (2024). Illusory generalizability of clinical prediction models. *Science*, *383*(6679), 164-167.

Choi, R. Y., Coyner, A. S., Kalpathy-Cramer, J., Chiang, M. F., & Campbell, J. P. (2020). Introduction to machine learning, neural networks, and deep learning. Translational vision science & technology, 9(2), 14-14.

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

Costa, P. T., & McCrae, R. R. (1992). *Revised NEO Personality Inventory (NEO PI-R) and NEO Five-Factor Inventory (NEO-FFI) professional manual*. Odessa, FL: Psychological Assessment Resources.

Cronbach, L. J., & Meehl, P. E. (1955). Construct validity in psychological tests. *Psychological Bulletin, 52*, 281-302.

Cutler, A., & Condon, D. M. (2023). Deep lexical hypothesis: Identifying personality structure in natural language. *Journal of Personality and Social Psychology*, *125*, 173-197.

Dai, X., Chalkidis, I., Darkner, S., & Elliott, D. (2022). Revisiting transformer-based models for long document classification. *arXiv preprint arXiv:2204.06683*.

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

Ding, M., Zhou, C., Yang, H., & Tang, J. (2020). Cogltx: Applying bert to long texts. *Advances in Neural Information Processing Systems*, *33*, 12792-12804.

Ding, Y., Zhang, L. L., Zhang, C., Xu, Y., Shang, N., Xu, J., Yang, F., & Yang, M. (2024). Longrope: Extending llm context window beyond 2 million tokens. *arXiv preprint arXiv:2402.13753.*

Dodge, J., Ilharco, G., Schwartz, R., Farhadi, A., Hajishirzi, H., & Smith, N. (2020). Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping. *arXiv preprint arXiv:2002.06305*.

Englhardt, Z., Ma, C., Morris, M. E., Chang, C.-C., Xu, X. O., Qin, L., McDuff, D., Liu, X., Patel, S., & Iyer, V. (2024). From classification to clinical insights: Towards analyzing and reasoning about mobile and behavioral health data with large language models. a*rXiv preprint arXiv:2311.13063.*

Ferrara, E. (2023). Should chatgpt be biased? Challenges and risks of bias in large language models. *arXiv preprint arXiv:2304.03738.*

Fields, J., Chovanec, K., & Madiraju, P. (2024). A survey of text classification with transformers: How wide? How large? How long? How accurate? How expensive? How safe? *IEEE Access*, *12*, 6518-6531. https://doi.org/10.1109/access.2024.3349952

Galatzer-Levy, I. R., McDuff, D., Natarajan, V., & Karthikesalingam, A. (2023). The Capability of large language models to measure psychiatric functioning. https://doi.org/10.48550/arXiv.2308.01834

Grootendorst, M. (2022). BERTopic: Neural topic modeling with a class-based TF-IDF procedure. arXiv:2203.05794. Retrieved March 01, 2022, from https://ui.adsabs.harvard.edu/abs/2022arXiv220305794G

Gu, A., & Dao, T. (2023). Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752.*

Gül, İ., Lebret, R., & Aberer, K. (2024). Stance Detection on Social Media with Fine-Tuned Large Language Models. *arXiv preprint arXiv:2404.12171*.

Harari, G. M., Müller, S. R., Aung, M. S., & Rentfrow, P. J. (2017). Smartphone sensing methods for studying behavior in everyday life. *Current Opinion in Behavioral Sciences, 18,* 83-90.

Hardy, M., Sucholutsky, I., Thompson, B., & Griffiths, T. (2023). Large language models meet cognitive science: LLMs as tools, models, and participants. In *Proceedings of the annual meeting of the cognitive science society* (Vol. 45, No. 45).

He, P., Liu, X., Gao, J., & Chen, W. (2020). Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*.

Hershcovich, D., Webersinke, N., Kraus, M., Bingler, J. A., & Leippold, M. (2022). Towards climate awareness in NLP research. *arXiv preprint arXiv:2205.05071.*

Honnibal, M., & Montani, I. (2017). spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing.

Hoory, S., Feder, A., Tendler, A., Erell, S., Peled-Cohen, A., Laish, I., Nakhost, H., Stemmer, U., Benjamini, A., & Hassidim, A. (2021). Learning and evaluating a differentially private pretrained language model. *Findings of the Association for Computational Linguistics: EMNLP 2021* (pp. 1178-1189).Hoo

Hopwood, C. J., & Bornstein, R. F. (Eds.). (2014). Multimethod clinical assessment. Guilford Publications.

Hua, Y., Liu, F., Yang, K., Li, Z., Sheu, Y.-h., Zhou, P., Moran, L. V., Ananiadou, S., & Beam, A. (2024). Large language models in mental health care: a scoping review. a*rXiv preprint arXiv:2401.02984.*

Huff, M., & Ulakçı, E. (2024). Towards a Psychology of Machines: Large Language Models Predict Human Memory. *arXiv preprint arXiv:2403.05152.*

Hussain, Z., Binz, M., Mata, R., & Wulff, D. U. (2023, December 5). A tutorial on open-source large language models for behavioral science. https://doi.org/10.31234/osf.io/f7stn

Jacobson, N. C., & Bhattacharya, S. (2022). Digital biomarkers of anxiety disorder symptom changes: Personalized deep learning models using smartphone sensors accurately predict anxiety symptoms from ecological momentary assessments. *Behaviour Research and Therapy*, *149*, 104013.

Jeon, H., Yoo, D., Lee, D., Son, S., Kim, S., & Han, J. (2024). A dual-prompting for interpretable mental health language models. *arXiv preprint arXiv:2402.14854*.

Jiang, A. Q., Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, D. S., de las Casas, D., Bressand, F., Lengyel, G., Lample, G., Saulnier, L., Lavaud, L. R., Lachaux, M-A., Stock, P., Le Scao, T. L., Lavril, T., Wang, T., Lacroix, T., & El Sayed, W. (2023). Mistral 7B. *arXiv preprint arXiv:2310.06825*

Jiang, T., Gradus, J. L., & Rosellini, A. J. (2020). Supervised machine learning: a brief primer. *Behavior Therapy, 51*(5), 675-687.

Johri, P., Khatri, S. K., Al-Taani, A. T., Sabharwal, M., Suvanov, S., & Kumar, A. (2021). Natural language processing: History, evolution, application, and future work. In: Abraham, A., Castillo, O., Virmani, D. (eds) *Proceedings of 3rd International Conference on Computing Informatics and Networks.* Lecture Notes in Networks and Systems, vol 167. Springer, Singapore. https://doi.org/10.1007/978-981-15-9712-1_31

Khurana, D., Koli, A., Khatter, K., & Sukhdev, S. (2023). Natural language processing: state of the art, current trends and challenges. *Multimed Tools Appl, 82*, 3713–3744. https://doi.org/10.1007/s11042-022-13428-4

Kjell, O. N. E., Kjell, K., & Schwartz, H. A. (2024). Beyond rating scales: With targeted evaluation, large language models are poised for psychological assessment. *Psychiatry Research, 333,* 1–12. https://doi.org/10.1016/j.psychres.2023.115667

Lazarević, L. B., Bjekić, J., Živanović, M., & Knežević, G. (2020). Ambulatory assessment of language use: Evidence on the temporal stability of electronically activated recorder and stream of consciousness data. *Behavior Research Methods*, *52*, 1817-1835.

Liang, P. P., Wu, C., Morency, L. P., & Salakhutdinov, R. (2021, July). Towards understanding and mitigating social biases in language models. In International Conference on Machine Learning (pp. 6565-6576). PMLR.

Lialin, V., Deshpande, V., Yao, X., & Rumshisky, A. (2024). Scaling down to scale up: A guide to parameter-efficient fine-tuning. *arXiv preprint arXiv:2303.15647.*

Lin, S., Wang, Y., Dong, J., & Ni, S. (2024). Detection and positive reconstruction of cognitive distortion sentences: Mandarin dataset and evaluation. *arXiv preprint arXiv:2405.15334*.

Liu, X., & Wang, C. (2021). An empirical study on hyperparameter optimization for fine-tuning pretrained language models. *arXiv preprint arXiv:2106.09204*.

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 preprint arXiv:1907.11692*.

Liu, Y., Wang, W., Gao, G. G., & Agarwal, R. (2023). Echoes of biases: how stigmatizing language affects AI performance. *arXiv preprint arXiv:2305.10201.*

Lundberg, S. M., & Lee, S. I. (2017). A unified approach to interpreting model predictions. *Advances in Neural Information Processing Systems* (pp. 4765-4774).

Luxton, D. D. (2014). Artificial intelligence in psychological practice: Current and future applications and implications. *Professional Psychology: Research and Practice, 45*(5),332–339. https://doi.org/10.1037/a0034559

Mehl, M. R. (2017). The electronically activated recorder (EAR) a method for the naturalistic observation of daily social behavior. *Current Directions in Psychological Science*, *26*, 184-190.

Meyer, G. J., Finn, S. E., Eyde, L. D., Kay, G. G., Moreland, K. L., Dies, R. R., ... & Reed, G. M. (2001). Psychological testing and psychological assessment: A review of evidence and issues. *American Psychologist*, *56*, 128-165. https://doi.org/10.1037/0003-066X.56.2.128

Milligan, G., Bernard, A., Dowthwaite, L., Vallejos, E. P., Davis, J., Salhi, L., & Goulding, J. (2024). Developing a single-session outcome measure using natural language processing on digital mental health transcripts. *Counselling and Psychotherapy Research*. https://doi.org/10.1002/capr.12766

Minaee, S., Mikolov, T., Nikzad, N., Chenaghlu, M., Socher, R., Amatriain, X., & Gao, J. (2024). *Large Language Models: A Survey*. (arXiv:2402.06196). ArXiv. https://doi.org/10.48550/arXiv.2402.06196

Min, B., Ross, H., Sulem, E., Veyseh, A. P. B., Nguyen, T. H., Sainz, O., Agirre, E., Heintz, I., & Roth, D. (2023). Recent advances in natural language processing via large pretrained language models: A survey. *ACM Computing Surveys*, *56*(2), 1-40. https://doi.org/10.1145/3605943

Mitchell, M., Wu, S., Zaldivar, A., Barnes, P., Vasserman, L., Hutchinson, B., ... & Gebru, T. (2019, January). Model cards for model reporting. In *Proceedings of the conference on fairness, accountability, and transparency* (pp. 220-229).

Mittal, A. (2024, May 3). Optimizing memory for large language model inference and fine-tuning. Unite.AI. https://www.unite.ai/optimizing-memory-for-large-language-model-inference-and-fine-tuning/

Morales, M., Scherer, S., & Levitan, R. (2018). A linguistically-informed fusion approach for multimodal depression detection. In *Proceedings of the Fifth Workshop on Computational Linguistics and Clinical Psychology*: *From Keyboard to Clinic* (pp. 13-24).

Naveed, H., Khan, A. U., Qiu, S., Saqib, M., Anwar, S., Usman, M., Barnes, N., & Mian, A. (2024). A comprehensive overview of large language models. *arXiv preprint arXiv:2307.06435*.

Ntoutsi, E., Fafalios, P., Gadiraju, U., Iosifidis, V., Nejdl, W., Vidal, M. E., Ruggieri, S., Turini, F., Papadopoulos, S., & Krasanakis, E. (2020). Bias in data-driven artificial intelligence systems—An introductory survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *10*(3), e1356. https://doi.org/10.1002/widm.1356

Ohse, J., Hadžić, B., Mohammed, P., Peperkorn, N., Danner, M., Yorita, A., Kubota, N., Rätsch, M., & Shiban, Y. (2024). Zero-Shot Strike: Testing the generalisation capabilities of out-of-the-box LLM models for depression detection. *Computer Speech & Language*, *88*.

Oltmanns, J. R., Khandelwal, R., Ma, J., Brickman, J., Do, T., Hussain, R., & Gupta, M. (under review). *Language-based AI assessment for personality disorder science and practice.*

Omiye, J. A., Lester, J. C., Spichak, S., Rotemberg, V., & Daneshjou, R. (2023). Large language models propagate race-based medicine. *NPJ Digital Medicine, 6*.

Page, M. J., McKenzie, J. E., Bossuyt, P. M., Boutron, I., Hoffmann, T. C., Mulrow, C. D., ... & Moher, D. (2021). The PRISMA 2020 statement: an updated guideline for reporting systematic reviews. *BMJ*. https://doi.org/10.1136/bmj.n71

Pandey, Y. N., Rastogi, A., Kainkaryam, S., Bhattacharya, S., Saputelli, L., Pandey, Y. N., Rastogi, A., Kainkaryam, S., Bhattacharya, S., & Saputelli, L. (2020). Overview of machine learning and deep learning concepts. Machine Learning in the Oil and Gas Industry: Including Geosciences, Reservoir Engineering, and Production Engineering with Python, 75-152.

Pargent, F., Schoedel, R., & Stachl, C. (2023). Best practices in supervised machine learning: A tutorial for psychologists. *Advances in Methods and Practices in Psychological Science, 6*(3), 25152459231162559.

Parker, J. L., Richard, V. M., & Becker, K. (2023). Guidelines for the integration of large language models in developing and refining interview protocols. *The Qualitative Report, 28*, 3460-3474.

Patterson, D., Gonzalez, J., Le, Q., Liang, C., Munguia, L. M., Rothchild, D., ... & Dean, J. (2021). Carbon emissions and large neural network training. *arXiv preprint arXiv:2104.10350.*

Paulhus, D. L., & Vazire, S. (2007). The self-report method. Handbook of research methods in personality psychology, *1*, 224-239.

Peng, C., Yang, X., Smith, K. E., Yu, Z., Chen, A., Bian, J., & Wu, Y. (2024). Model tuning or prompt Tuning? a study of large language models for clinical concept and relation extraction. *Journal of Biomedical Informatics, 153*, 104630. https://doi.org/10.1016/j.jbi.2024.104630

Pennebaker, J. W., & King, L. A. (1999). Linguistic styles: language use as an individual difference. *Journal of Personality and Social Psychology*, *77*, 1296-1312.

Pennebaker, J. W., Mehl, M. R., & Niederhoffer, K. G. (2003). Psychological aspects of natural language use: Our words, our selves. *Annual Review of Psychology*, *54*, 547–577. https://doi.org/10.1146/annurev.psych.54.101601.145041

Peters, H., & Matz, S. (2023). Large language models can infer psychological dispositions of social media users (arXiv:2309.08631). arXiv. http://arxiv.org/abs/2309.08631

Rajapakse, T. C. (2019). *Simple Transformers*. GitHub. https://github.com/ThilinaRajapakse/simpletransformers

Rathje, S., Mirea, D. M., Sucholutsky, I., Marjieh, R., Robertson, C. E., & Van Bavel, J. J. (2024). GPT is an effective tool for multilingual psychological text analysis. Proceedings of the National Academy of Sciences, 121(34), e2308950121.

Raza, S., Bamgbose, O., Ghuge, S., Tavakoli, F., & Reji, D. J. (2024). Developing safe and responsible large language models--A comprehensive framework. *arXiv preprint arXiv:2404.01399.*

Reimers, N., & Gurevych, I. (2019). Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Rillig, M. C., Ågerstrand, M., Bi, M., Gould, K. A., & Sauerland, U. (2023). Risks and benefits of large language models for the environment. *Environmental Science & Technology, 57, 3464-3466.*

Sanford, F. H. (1942). Speech and personality. *Psychological Bulletin*, *39*, 811-845.

Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108.*

Schoenegger, P., Greenberg, S., Grishin, A., Lewis, J., & Caviola, L. (2024). Can AI understand human personality?--Comparing human experts and AI systems at predicting personality correlations. *arXiv preprint arXiv:2406.08170.*

Simchon, A., Sutton, A., Edwards, M., & Lewandowsky, S. (2023). Online reading habits can reveal personality traits: Towards detecting psychological microtargeting. *PNAS Nexus*, *2*(6), pgad191. https://doi.org/10.1093/pnasnexus/pgad191

Simons, J. J., Ze, W. L., Bhattacharya, P., Loh, B. S., & Gao, W. (2024). From traces to measures: Large language models as a tool for psychological measurement from text. *arXiv preprint arXiv:2405.07447*.

Shekhar, S., Dubey, T., Mukherjee, K., Saxena, A., Tyagi, A., & Kotla, N. (2024). Towards Optimizing the Costs of LLM Usage. *arXiv preprint arXiv:2402.01742*.

Shulga, D. (2018). 5 Reasons why you should use Cross-Validation in your Data Science Projects. Towards Data Science. https://towardsdatascience.com/5-reasons-why-you-should-use-cross-validation-in-your-data-science-project-8163311a1e79

Spiller, T. R., Rabe, F., Ben-Zion, Z., Korem, N., Burrer, A., Homan, P., Harpaz-Rotem, I., & Duek, O. (2023). Efficient and accurate transcription in mental health research-A tutorial on using whisper AI for audio file transcription.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, *15*, 1929-1958.

Strauss, M. E., & Smith, G. T. (2009). Construct validity: Advances in theory and methodology. *Annual Review of Clinical Psychology*, *5*, 1-25.

Strubell, E., Ganesh, A., & McCallum, A. (2020, April). Energy and policy considerations for modern deep learning research. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 34, No. 09, pp. 13693-13696).

Subramani, N., Matton, A., Greaves, M., & Lam, A. (2020). A survey of deep learning approaches for OCR and document understanding. *arXiv preprint arXiv:2011.13534*.

Sun, T., Gaut, A., Tang, S., Huang, Y., ElSherief, M., Zhao, J., ... & Wang, W. Y. (2019). Mitigating gender bias in natural language processing: Literature review. arXiv preprint arXiv:1906.08976.

Tausczik, Y. R., & Pennebaker, J. W. (2010). The psychological meaning of words: LIWC and computerized text analysis methods. *Journal of Language and Social Psychology, 29,* 24-54.

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., & Bhosale, S. (2023). Llama 2: Open foundation and fine-tuned chat models. a*rXiv preprint arXiv:2307.09288*.

Trull, T. J., & Ebner-Priemer, U. (2013). Ambulatory assessment. *Annual review of clinical psychology*, *9*, 151-176.

Tu, S., Powers, A., Merrill, N., Fani, N., Carter, S., Doogan, S., & Choi, J. D. (2024). Automating PTSD diagnostics in clinical interviews: Leveraging large language models for trauma assessments. *arXiv preprint arXiv:2405.11178*.

Tuggener, L., Sager, P., Taoudi-Benchekroun, Y., Grewe, B. F., & Stadelmann, T. (2024). So you want your private LLM at home? A survey and benchmark of methods for efficient GPTs. *11th IEEE Swiss Conference on Data Science (SDS), Zurich, Switzerland, 30-31 May 2024.* https://doi.org/10.21256/zhaw-30279

Tuteja, M., & Juclà, D. G. (2023, December). Long text classification using transformers with paragraph selection strategies. *In Proceedings of the Natural Legal Language Processing Workshop 2023* (pp. 17-24).

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

de Vassimon Manela, D., Errington, D., Fisher, T., van Breugel, B., & Minervini, P. (2021, April). Stereotype and skew: Quantifying gender bias in pretrained and fine-tuned language models. *In Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume* (pp. 2232-2242).

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.

Vickers, P., Barrault, L., Monti, E., & Aletras, N. (2024). We need to talk about classification evaluation metrics in NLP. *arXiv preprint arXiv:2401.03831.*

Vig, J. (2019). A multiscale visualization of attention in the transformer model. *arXiv preprint arXiv:1906.05714*.

Wang, Z., & Sun, J. (2022, August). Survtrace: Transformers for survival analysis with competing events. In *Proceedings of the 13th ACM international conference on bioinformatics, computational biology and health informatics* (pp. 1-9).

Watkins, R. (2023). Guidance for researchers and peer-reviewers on the ethical use of Large Language Models (LLMs) in scientific research workflows. *AI and Ethics*, 1-6.

Wong, T. T. (2015). Performance evaluation of classification algorithms by k-fold and leave-one-out cross validation. *Pattern recognition*, *48*(9), 2839-2846.

Wu, C., Wu, F., Qi, T., & Huang, Y. (2021). Hi-transformer: Hierarchical interactive transformer for efficient and effective long document modeling. *arXiv preprint arXiv:2106.01040*.

Wulff, D. U., & Mata, R. (2023, October 12). Using embeddings to automate jingle–jangle detection and tackle taxonomic incommensurability. https://doi.org/10.31234/osf.io/9h7aw

Yang, K., Li, H., Wen, H., Peng, T. Q., Tang, J., & Liu, H. (2024). Are large language models (LLMs) good social predictors? *arXiv preprint arXiv:2402.12620*.

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R. R., & Le, Q. V. (2019). Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems, 32*.

Yates, L. A., Aandahl, Z., Richards, S. A., & Brook, B. W. (2023). Cross validation for model selection: A review with examples from ecology. *Ecological Monographs*, *93*(1). https://doi.org/10.1002/ecm.1557

Zhang, H., Lu, A. X., Abdalla, M., McDermott, M., & Ghassemi, M. (2020, April). Hurtful words: quantifying biases in clinical contextual word embeddings. In *proceedings of the ACM Conference on Health, Inference, and Learning* (pp. 110-120).

Zhao, H., Chen, H., Yang, F., Liu, N., Deng, H., Cai, H., Wang, S., Yin, D., & Du, M. (2024). Explainability for large language models: A survey. *ACM Trans. Intell. Syst. Technol.*, *15*(2), Article 20. https://doi.org/10.1145/3639372

Zhao, W. X., Zhou, K., Li, J., Tang, T., Wang, X., Hou, Y., Min, Y., Zhang, B., Zhang, J., & Dong, Z. (2023). A survey of large language models. *arXiv preprint arXiv:2303.18223*.

Table 1

Glossary

Terminology	Description	Coding packages
Word Embeddings	Multi-dimensional vectors of numbers representing semantic relationships between words.	spaCy, transformers
Parameter	Trainable internal values that are optimized during training to reduce prediction error.	torch.nn.Parameter
Deep Learning	Advanced type of machine learning inspired by how humans process information. Learns patterns by processing data through multiple layers, allowing for increasingly more complex connections.	tensorflow, PyTorch, Keras
Model Layer	Step in a deep learning neural network model that receives information, performs computations, and passes updated information to next step.	torch.nn, tensorflow.keras.layers
Transformer models	Advanced neural network model that captures complex relationships in language using a "self-attention" mechanism, widely used in NLP tasks, and are the basis for LLMs.	transformers, simpletransformers, torch.nn
Feed Forward Layer (Neural Network)	A component in transformer models that comes after self-attention. It processes each word separately, taking the contextual information provided by self-attention about the word. It then transforms that information to make it more useful in the next step.	torch.nn, tensorflow.keras.layers.Dense
Audio Processing	Stage in which audio is converted into usable text and speaker segments. Typically includes automatic speech recognition (transcribing speech into text) and speaker diarization (identifying who spoke when).	speechbrain, whisper, pyannote-audio, azure
Text Pre-processing	Changing language data into a format useful for NLP. Includes de-identification, tokenization (splitting text data into smaller units), decisions related to context, and isolating language of interest.	spaCy, transformers.AutoTokenizer, scikit-learn
Label	A known outcome or target variable that a language model is trained to predict. Like a dependent variable in traditional statistics.	Specified in training arguments
Hard Prompt	Text instructions written by humans and given to an LLM to guide it in performing a specific task.	n/a

Soft Prompt	Trainable embeddings prepended to input text and updated during training. They can then guide the model's behavior or focus without using explicit language instructions.	peft, transformers
Parameter Efficient Fine-Tuning (PEFT)	Technique to avoid excessive training cost by updating only a small set of model weights during training rather than the full set of parameters.	peft, torch.nn.Parameter
Hyperparameter	One of several fixed settings defined before training a machine learning model. Hyperparameters can significantly impact model performance and are therefore a key focus during model development.	wandb, torch.optim, simpletransformers
Hyperparameter tuning	The process of adjusting and testing different hyperparameter values to optimize model performance. Typically done using a validation dataset to identify the best settings before evaluating on test data.	wandb, optuna, torch.optim, simpletransformers
Learning rate	A hyperparameter that controls how much the model's parameters are adjusted in response to each update during model training.	torch.optim, tensorflow.keras.optimizers
Batch size	A hyperparameter that determines how many training samples the model processes before calculating errors and updating the parameters.	Specified in training arguments
Epochs	A hyperparameter that determines how many full passes the model makes through an entire dataset during the training process.	Specified in training arguments
Overfitting	When a model capitalizes on specifics in the training data and learns to predict variance unique to the sample, rather than broader patterns that will generalize to new data.	n/a
Underfitting	When a model fails to capture meaningful patterns in the data and performs poorly on both training and test sets. This can result from an overly simple model, insufficient training time, or not enough data.	n/a
Early stopping	Technique used to prevent overfitting by stopping training when model performance on validation data stops improving. Requires an evaluation metric (e.g., R^2) and a patience interval to decide when to stop.	transformers.Trainer, simpletransformers, tensorflow.keras.callbacks.EarlyStopping
Patience interval	The number of consecutive epochs without improvement on a validation metric before training is stopped with early stopping. Typical values are 5 or 10 epochs. Setting this too low (e.g., 1) risks prematurely stopping training, as improvements may not be strictly linear.	Specified in training arguments

Table 2

Potentially Useful LLMs for Psychological Assessment

Model Family	Parameters	Developer	API	Open Source	License	Training Data	Max Token Limit
BERT	4M - 340M	Google (Devlin et al., 2019)	Hugging Face	Yes	Apache 2.0	BookCorpus and English Wikipedia	512
GPT	Estimated 8B - 1T+	OpenAI (Achiam et al., 2023)	OpenAI API	No	Proprietary	Publicly available and licensed data	128k
Llama	1B - 405B	Meta	Llama API, Hugging Face	Yes	Custom	Publicly available online data	128K
Claude	Estimated 70B - 2T+	Anthropic	Anthropic API	No	Proprietary	Public internet information, non-public third-party data, internally created data; fine-tuned with Constitutional AI	1M
Gemini	1.8B - 1T+	Google DeepMind	Gemini API, Google Vertex AI	No	Proprietary	Publicly available filtered web data, licensed 3rd party data	2M
Falcon	1B - 180B	Technology Innovation Institute	Hugging Face	Yes	Apache 2.0	RefinedWeb, filter of Common Crawl dataset	32K
Mistral	7B - 120B+	Mistral AI (Jiang et al., 2023)	Mistral AI, Hugging Face	Some	Apache 2.0	Open web data	128K
DeepSeek	7B - 671B	Hangzhou DeepSeek AI	DeepSeek API, Hugging Face	Some	Apache 2.0 (most)	Web data, code, math	128K
Qwen	0.5B - 110B	Alibaba Cloud	Alibaba Cloud, Hugging Face	Some	Apache 2.0	Public web documents, encyclopedia, books, math, code, synthetic data	1M
Gemma	1B - 27B	Google DeepMind	Google Vertex AI, Hugging Face	Yes	Apache 2.0	Web documents, code, and math	128K

Note. M = million, B = billion, T = trillion. API = application to programming interfaces, BERT = Bidirectional Encoder Representation from Transformers , GPT = Generative Pretrained Transformer.