Using machine learning to automate the collection, transcription, and analysis of verbal-report data

Tehilla Ostrovsky[1], Paul Ungermann[2], and Chris Donkin[1]

[1]School of Psychology, Ludwig Maximilian University, Munich, Germany

[2]Technical University of Munich, Munich, Germany

Author Note

Correspondence concerning this article should be addressed to Tehilla Ostrovsky, Ludwig Maximilian University, Munich, email: tehila.mechera-ostrovsky@psy.lmu.de. phone number: +41 76 6735 421

Abstract

What people think and say during experiments is important for our understanding of the human mind. However, the collection and analysis of verbal-report data in experiments is relatively costly, and so is grossly underutilized. Here, we aim to reduce such costs by providing software that will collect, transcribe, and analyse verbal-report data. Verbal data is collected using jsPsych De Leeuw (2015), making it suitable for online and lab-based experiments. The transcription and analyses rely on machine-learning methods (e.g., large-language models), making them substantially more efficient than current methods using human coders. We demonstrate how to use the software we provide in a case study via a simple memory experiment. This collection of software was made to be modular, so that the various components can be updated and replaced with superior models and new methods easily added. It is our sincere hope that this approach popularizes the collection of verbal-report data in psychology experiments.

Keywords: natural language processing models, self-report, verbal description, decision-making, cognitive model validity

Using machine learning to automate the collection, transcription, and analysis of verbal-report data

Introduction

Among Experimental Psychologists, self-report has a relatively bad rap. Compared to measures like choice proportions, accuracy, and reaction times, explicit verbal reports are often seen as subjective and noisy, both in terms of what participants produce and also in how they are analyzed. Two key properties of verbal reports give rise to this reputation. The first is that people do not have perfect insight into the reasons they do things, which was a major impetus for the cognitive revolution. Second, people can express themselves in such a variety of ways that verbal responses have typically needed to be analysed by other human coders, in a subjective and inefficient manner.

While there is no doubting that introspection can be problematic, it is becoming increasingly clear that we have over corrected, and that largely ignoring what people say they are doing is holding us back. For example, Frey, Pedroni, Mata, Rieskamp, and Hertwig (2017) compared behavioral measures from controlled experimental psychology tasks with self-reported questionnaires on risk-taking. They showed that self-report tasks frequently outperform behavioral tasks in assessing and predicting risk preferences. Responses to questions, such as 'Are you generally a risk-taking person or do you try to avoid risks?' and 'How likely would you be to go white-water rafting at high water in the spring?', not only better reveal stated preferences than choice proportions in experimental tasks, but they also correlate with specific real-world risky activities. Such a result likely reflects a general trend, given the slew of recent reports of poor test-retest reliability in measures taken from experimental tasks (Enkavi et al., 2019; Hedge, Powell, & Sumner, 2018; Rouder & Haaf, 2019).

Many cognitive theories also clearly imply what people *should* be able to (and not able to) say while performing a task. Ostrovsky and Newell (2024) argue that explicit verbal reports can be used to test such theoretical claims. Such a development is welcome, since implications about what should be in the minds of participants are rarely tested, and so are a neglected source of flexibility in our cognitive theories

(Szollosi & Donkin, 2019). Indeed, we suspect that for the vast majority of phenomena in psychology it is so obvious that we need to understand what people can and cannot say while performing a task, that the major reason that most researchers ignore verbal reports is pragmatic. That is, because collecting and analysing verbal data is simply much too challenging.

There are many ways of eliciting people's thoughts or reasoning during a task. Free-text or verbal reports can refer to the articulation of a person's thoughts, feelings, attitudes, or cognitive processes in response to specific stimuli or tasks. The primary aim of verbal reports is to gain a more granulated insight into the cognitive processes underlying behavior, decision-making, problem-solving, and other psychological phenomena. These reports may be spoken aloud or provided in written form. In short, collecting verbal report data requires participants to verbalize their thoughts, either in real-time while engaged in a specific task (e.g., using a 'think aloud' method), or retrospectively after the task is complete.

Many authors have written extensively about best practice and what should be considered when collecting verbal data (e.g. Ericsson & Simon, 1980; Fox, Ericsson, & Best, 2011; Ranyard & Svenson, 2011; Svenson, 1979, 1989). Many such concerns are around what Ericsson and Simon (1980) called the suitability of the task: determining whether and how to elicit verbal reports during the task. We leave it to the Discussion section to address this problem, since the details will inevitably be specific to the research question at hand. Rather, our goal in this manuscript is to deal with the second major issue that Ericsson and Simon address, the challenge of scalability.

Verbal report data can be large, messy, subjective, sometimes irrelevant, overwhelming, and thus, costly. The usual method for analysing verbal report data is via a human coder/rater. A researcher is typically trained to condense the text, for example, by creating shorter summaries, or by classifying responses into certain types. This approach is expensive, usually in terms of both time and money, especially since multiple raters will often be used to check whether the coding is reliable.

The cost of verbal-report data is, we suspect, mostly responsible for its lack of use.

For example, participant numbers in experiments must be kept relatively small, since the cost grows larger with increased sample sizes. As such, concerns about the variability in verbal-report data are reasonable – simply collecting more data to increase the signal relative to the noise is rarely an attractive option. Furthermore, the cost makes it daunting for an experimental psychologist to incorporate verbal report data into their own experimental paradigms, since it not always clear what participants might say, and coders must be given specific instructions when analysing text (e.g., what types of response to anticipate classifying).

Using large-language models to overcome the challenges with verbal reports

To address these challenges with the scalability of verbal reports, we follow the proposal in Ostrovsky and Newell (2024), to use large language models (LLMs). At the time of writing, LLMs are typically of the transformer architecture, such as GPT (Generative Pre-trained Transformer) models, BERT (Bidirectional Encoder Representations from Transformers) models and BART (Bidirectional and Auto-Regressive Transformers) models. Such models bring several advantages when analyzing verbal data. Their architecture and training allows them to represent words and sentences in a way that includes semantics and context, meaning that they can deal with the nuanced expressions found in "think-aloud" data. As such, LLMs are able to perform relatively well at many of the kinds of tasks that are usually given to human coders, such as summarising and classifying text.

The way by which LLMs analyze verbal data is primarily achieved through their embeddings – high-dimensional numerical representations for words or sentences that encode their semantic meaning in a given context. These embeddings are similar to the data created by other tracing methods used in psychological research, such as eye fixation patterns or EEG signals. In the same way, we can analyse these embeddings to reveal structure and systematicity in the numerical representations, and then look at what this implies for the text that produced them. That is, the LLMs translate between text and numbers, allowing us to use the methods developed for finding patterns in

numerical data to find patterns in text. This approach seems to work. Indeed, by now anyone who has used an LLM knows that it is capable of performing many of the tasks that we would give to a human rater of verbal report data.

Since LLMs seem to be able to approximate human raters, their most attractive property is that they can process large amounts of text data swiftly and relatively cheaply. It is hard to overstate the potential that such a benefit over traditional qualitative methods could offer. The relatively small cost of using LLMs to analyse verbal data should solve many of the aforementioned barriers to including verbal reports in experimental psychology. For example, researchers need not anticipate what kinds of things participants will say during an experiment, but can rather run and rerun various analyses to discover the kinds of things people said. The scalability of LLM analyses also means that the amount of verbal report data that researchers can collect and analyse can also now grow to sizes like we are used to with other dependent measures. Not only does this help increase the reliability of the data we collect, but it also gives us another way to deal with the noisiness of verbal data.

The aim of this article is to provide software that makes it easy for researchers to collect spoken verbal reports, to automatically translate those reports into text, to transform that text into numerical embeddings using an LLM, and to then apply a suite of analyses of the resultant quantitative data. Given the tremendous pace with which this field is moving, the software is designed to be easily changed to incorporate newer methods and models. As such, we refer to it throughout as a 'framework' rather than a particular piece of software.

Framework

The core role of the framework we propose here is to record participants' spoken reports during a study, automatically translate them into text, and analyse that text using various machine-learning based analyses. The framework is modular and subject to updates as desired. For a schematic description of the framework see Figure 1.

To give an overview, the recording aspect of the framework is implemented in

jsPsych (De Leeuw, 2015), a popular JavaScript-based framework to conduct studies in a web browser, which should allow it to be easily integrated into the kinds of studies already run using jsPsych. While the recording occurs continuously throughout the experiment, what is said is segmented into trials as per usual within jsPsych (Boxes 1-3, Figure 1). Both the recordings and the corresponding behavioral data (the default jsPsych output) are transferred to a server for storage.

Once the recordings are saved, a machine-learning pipeline is used to convert the audio recordings into written text for further analysis (Box 4, Figure 1). For this, we provide a platform-independent open-source solution that is ready to use, along with an in-depth tutorial on how to integrate it into a new study [1]. Finally, we provide a set of possible analyses that can be adapted to be run on written text. Most of these analyses rely on embeddings created by an open-source LLM (Box 5, Figure 1). A series of subsequent analyses are provided (Boxes 6-8, Figure 1), though this is where we expect the exact contents to be readily modified and expanded by the users. We now provide a detailed description of each of these functionalities in turn. Then, in the Case Study section, we demonstrate and evaluate the performance of each of these functionalities.

Recording

The initial step within the framework involves the recording of participants. This functionality is activated at the beginning of each trial. As illustrated in the code snippet 1, the jsPsych function `onTrialStartRecording` is invoked to initiate the recording process. The recording process concludes at the end of each trial with the function `onTrialFinishRecording`. These functions activate the recordings locally on the person's computer (see top 3 left rectangles Figure 1). At the end of each trial, the recording is sent back to the server and is saved as a `.wav` file (see the top right rectangle in Figure 1).

Upon completion of the experiment, to ensure participants have the option to defer their participation, they are asked whether they still consent to the transmission and

[1] https://github.com/tehillamo/AutoV-LLM/blob/main/beginners_guide.pdf

analysis of their data. If they decline, their data is immediately deleted from the server. Similarly, if they leave the experiment early, their data is deleted.

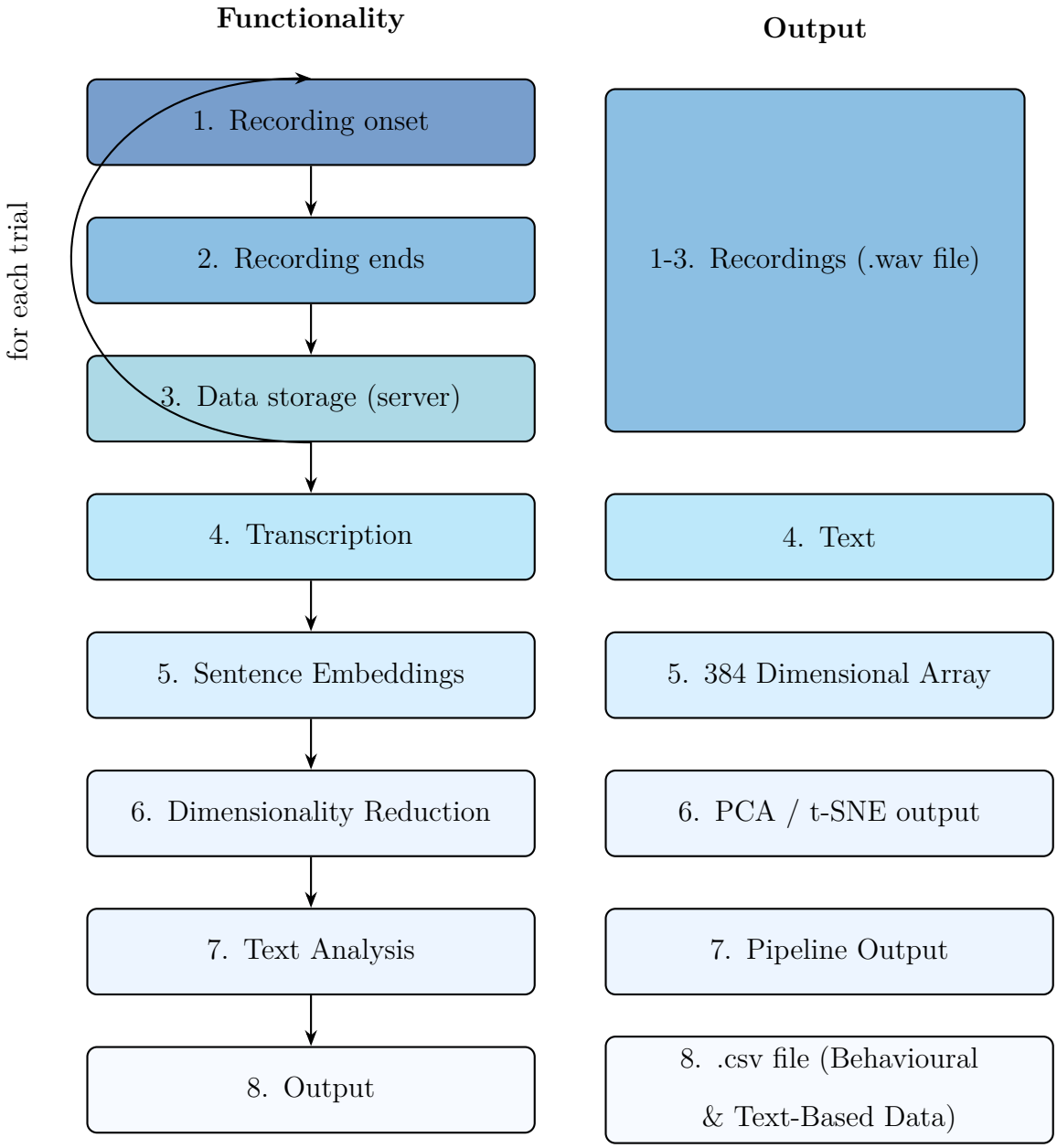
```
var jsPsych = initJsPsych({
    on_trial_start: function (trial) {
        onTrialStartRecording(trial);
    },
    on_trial_finish: function (trial) {
        onTrialFinishRecording(trial);
    },
    on_finish: function () {   sendData(jsPsych.data.allData.
       trials);
    },
});
```

Listing 1: JavaScript code snippet for jsPsych Initialization of Trial & of Recording

Transcription

Once the full recordings are stored on the server, experimenters can proceed to send these recordings for transcription, transforming the audio recordings into text. The output of this step is the transcribed text for each trial. Examples of such transcribed recordings are shown in Table 1 under the column *transcribed text*. In the current version of the framework, this step is done using OpenAI's Whisper machine-learning model, chosen for its (at the time) state-of-the-art performance in speech-to-text tasks and its open-source availability (see box 4 in Figure 1). At the time of writing the *large* version of the model had 1.54B parameters (more information about this model can be found at `huggingface.co/openai/whisper-large`). As these models are constantly updated, the specific model version is a parameter in our framework and can be adjusted in the script under `transcription model: "large"` within the configuration file (which can be found at `github.com/tehillamo/AutoV-LLM/blob/main/data_pipeline/config.json`).

Figure 1. Flowchart describing the timeline of the functionalities and the outputs of the framework



Boxes 1-3 represent the recording of each trial. Box 4 shows the transcription of the recorded trials into text. Box 5 represents the conversion of the text data into embeddings. Box 6 illustrates the step of the dimensionality reduction performed on the embeddings from Box 5. Box 7 represents the pipelines applied to the transcribed text from Box 4's output, and Box 8 shows the final output after all processes from Boxes 1-7 have been applied.

Embeddings

The next step in this framework involves generating text embeddings for each trial, based on the transcribed text (see box 5 in Figure 1). Embeddings are numerical representations of text, produced by transforming words or phrases into vectors of real numbers. These high-dimensional embedding vectors encode the meaning of a given piece of text in its given context. Sentence embeddings are specifically designed to capture semantic similarity, ensuring that semantically similar sentences are positioned closer together in the vector space. As such, the embeddings facilitate a range of analyses and visualization techniques that can help a researcher to summarise and understand what was spoken.

We compute these embeddings using the Sentence Transformer 'all-MiniLM-L6', which is an open-source model, is relatively small, is fast to use, while still offering good performance relative to larger models (more information about this model can be found at https://huggingface.co/sentence-transformers/all-MiniLM-L6-v1). This version of the model has 22.7 million parameters and outputs a 384 dimensional embedding vector. As these models are constantly updated, this model can be changed in the script under `model = SentenceTransformer('all-MiniLM-L6-v2')` within the `embeddings.py` file (which can be found at github.com/tehillamo/AutoV-LLM/blob/main/data_pipeline/embeddings.py).

Dimensionality Reduction

The first analysis of embeddings we provide (Box 6 in Figure 1) attempts to reduce the high-dimensionality of the embeddings to determine if the text varies on any systematic, meaningful psychological concepts (such as "confidence", "context", or "riskiness"). Such reduction can be useful because the embeddings are of such high-dimensionality, making the problem similar to those in other psychological research fields, where things like Principal Component Analysis (PCA) or factor analysis are used to simplify data. The idea underlying dimensionality reduction methods is that the text embeddings may vary in such a way that, for a particular experiment, there is a small

number of frequently occurring features of the text, such as statements about confidence or risk, that account for most of the variability in what was said by participants.

We have implemented two techniques for dimensionality reduction: PCA and the t-Distributed Stochastic Neighbor Embedding (t-SNE) (Van der Maaten & Hinton, 2008). We offer PCA as it is a historically popular method of dimensionality reduction in a number of fields of Psychology. We also offer an alternative, t-SNE, which is similar but reduces the dimensionality of the data while preserving the data's neighborhood structure. This technique is, at the time of writing, popular for use with text embeddings.

Users can modify the method for dimensionality reduction to either `"PCA"`, `"t-SNE"`, or `"both"` in the `config.json` script under: `"reduction_algorithm": ["PCA", "TSNE", "both"]`. Both methods require the user to specify the number of dimensions to which the raw embeddings should be reduced, and while the default setting is 2, this can be configured by changing the line: `"dimension": 2` in the same configuration file. Finally, the analyses provided by default applies the reduction to all participants and trials simultaneously, and not on a participant-by-participant basis, but this can also be changed in the `config.json` file by modifying the `"reduction\_per\_participant"` option from false (the default) to true.

The output of this functionality is, for both methods of reduction, the position of each trials' text in the new, reduced space. For instance, if the experimenter selects the default option in our framework, which compresses the raw, high-dimensional embeddings array to 2 dimensions, the output for each trial will be a 2-dimensional vector representing a compressed numeric representation of the sentence.

Text Analysis

Box 7 of Figure 1 is a catch-all term for the variety of analyses one might use to extract meaningful insights from transcribed text. By default, we have provided a text classification algorithm that allows users to categorize each verbal report into predefined

labels, as well as a keyword extraction algorithm (for more details see Grootendorst (2020a)), and an auto-summarization method. These tools are just some of the techniques that researchers can use to distill substantial information from text. Below, we provide details on each of these default pipelines.

It should be noted that this aspect of the framework is intentionally flexible, because the appropriateness of the different pipelines is context-dependent and must be carefully chosen by the researcher. For example, summaries may not be practical for brief transcriptions, and a lot of work may have to go into creating adequate labels for labeling techniques. As a default, these methodologies are included in the output, however, the reader should not take this as a sign of endorsement for their use. Rather, we expect users to select and modify these tools as needed. Indeed, it bears repeating that due to the rapidly evolving machine-learning landscape, the entire framework is modular so as to facilitate easy substitutions, such as switching between the choice of particular models (both for transcription and for the embeddings) and between text analyses methods (for more details, see our documentation at https://github.com/tehillamo/AutoV-LLM/).

Text Labelling. One way to understand verbal report data is to find an appropriate label for each utterance. We use a zero-shot classification algorithm to identify the most likely text label for a given input. The text labeling employs a "zero shot" learning pipeline to assign labels to participant-recorded sentences (Sanh et al., 2022; Yin, Hay, & Roth, 2019). This method has the benefit of not requiring training with specific labels; instead, it uses the set of labels provided by the user to classify the text according to how it was trained, generically speaking. A score is given to each label, indicating how well it matches the input text (in the space of high-dimensional embeddings). The label with the highest score is considered the most appropriate label.

In this pipeline, users can modify the set of labels to their needs by adjusting the `text_classes` variable (`text_classes = ["label_1", "label_2"]`) in the configuration file and change the classifier with `pipe = pipeline(model="facebook/bart-large-mnli")` in the

`keyword_similarity_zero_shot` function, found in the `text_classification.py` script within this GitHub repository: https://github.com/tehillamo/AutoV-LLM/blob/main/data_pipeline/text_classification.py. By default, the output is the most appropriate label, but the code can be altered to also yield the score for all of the candidate labels. The line of code responsible for selecting and saving the label with the highest similarity in the output file is located in the `text_classification.py` script. This line of code is:

```
df.loc[indices_with_text, 'highest_similarity'] = df.loc[
indices_with_text, 'highest_similarity'].apply(lambda x: (x['labels'][0],
 x['scores'][0])).
```

Users who wish to save the entire list of labels, along with their associated scores, can remove or comment out this line. If users only want labels that surpass some level of appropriateness (or confidence, per the nomenclature in the field), then they can set a threshold in the configuration file `config.json` under `text_classification_threshold: your_chosen_threshold`. Smaller threshold settings (e.g., .1) will lead to more labels being considered appropriate, while higher thresholds (e.g., .3 or above) return fewer labels, which can improve accuracy but may lead to some appropriate labels being missed.

Keyword Extraction. Keyword extraction is the automated process of identifying the most relevant words and phrases from an input text. We use the *BERT* model, a bi-directional transformer that converts phrases and documents into vectors that capture their meaning (Grootendorst, 2020b). KeyBERT is a simple and effective keyword extraction technique that utilizes BERT embeddings to generate keywords and keyphrases most closely related to the document (for details about this model see https://github.com/MaartenGr/KeyBERT). As with other pipeline scripts, the model used can be modified in the `keywords.py` script by adjusting the variable `kw_model = KeyBERT()` found in the repository at https://github.com/tehillamo/AutoV-LLM/blob/main/data_pipeline/keywords.py. The output of this algorithm is currently set to a default of 5 keywords, as defined by the argument `top_n_keywords` in

the function configuration file. Researchers can adjust this value as needed.

Auto-summarization. For text summarization we use the BART pipeline, which is a sequence-to-sequence model, meaning that it is a type of deep learning model specifically designed to handle tasks where the input and output are both sequences of data. At the time of writing the model has 400 million parameters (for more details, refer to https://github.com/facebookresearch/fairseq/tree/main/examples/bart), which it uses to shorten a given text input down to a specified length, while preserving the most important features of the input text.

As with any other text analysis pipeline, this model can be replaced with another in the `summarization.py` script by modifying the line `summarizer = pipeline("summarization", model="facebook/bart-large-cnn")`, available in our repository at https://github.com/tehillamo/AutoV-LLM/blob/main/data_pipeline/summarization.py. The output from the summarization analysis consists of summaries for each verbal report, provided they are sufficiently lengthy to warrant summarization. The current settings limit the summary length to between 30 and 130 words. These parameters can be adjusted by modifying the `max_length_summary` and `min_length_summary` arguments in the config file.

Output

The final step in the framework is to merge all of the outputs computed in each of the above sections, including the transcribed text, embeddings, and the results of the included analyses of the text, with the behavioral data obtained from the actual study. This process results in a .csv file with each feature, such as the transcribed text, presented its own separate column (see Box 8 in Figure 1). For a truncated example see Table 1. The script that merges all data together can be found and altered in https://github.com/tehillamo/AutoV-LLM/blob/main/data_pipeline/merge_behavioral_data.py.

Table 1

Sample data from our machine learning pipeline

trial number	transcribed text	rt	text embedding
100	West and geese I don't	10532	[5.8e-02, -2.5e-02, -1.0e-02, 2.0-02, <...>]
104	Pretty sure it's spout.	10113	[-1.8e-02, -4.3e-02, -3.8-02, 2.2e-02 <...>]
520	I think Canada was there	3092	[9.6e-02, 9.6e-02, -3.2e-02, -9.2e-03 <...>]
516	China or geese	18038	[1.5e-02, 5.2e-02, -1.7e-02, 6.5e-02 <...>]
496	Float. Pretty sure. Float.	5669	[1.9e-02, 1.7e-02, -8.5e-02, 3.8e-02 <...>]
292	I don't remember what I said.	14267	[-7.6e-02, 3.0e-03 4.4e-02, 1.2e-01 <...>]
484	Sweden I do have	2836	[6.7e-02, 4.3e-02, -2.5e-02, -4.0e-02 <...>]
448	I do remember seeing Taiwan.	5540	[9.6e-02, 4.1e-02, -1.2e-02, 4.5e-02 <...>]

*Note: The **trial number** and **rt** columns show the trial number and reaction time recorded by default in the jsPsych framework. The **transcribed text** column contains the transcription of verbal reports, and the **text embedding** column represents the embeddings of those transcriptions for each trial, both generated by our framework. The transcribed text is truncated due to space limitations, and we selected only a few rows for better visualization (refer to the trial number column).*

Case Study

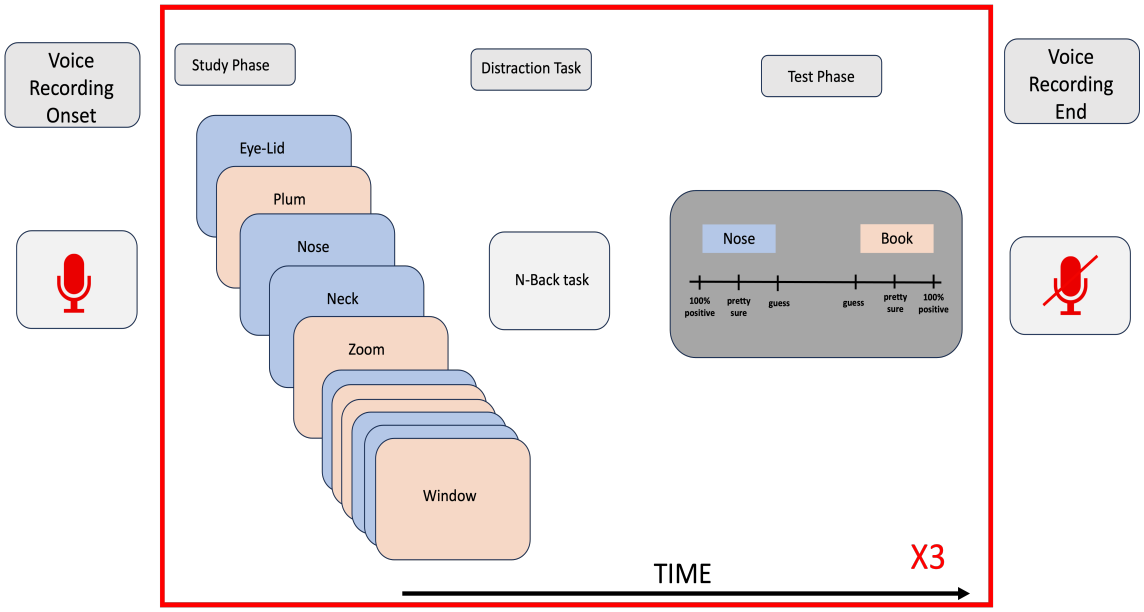
To demonstrate the use of our framework, we conducted a relatively standard recognition-memory study. Participants were asked to remember lists of words presented one at a time during a study phase, and then were tested on their memory for those words, and their confidence in that memory, in a test phase. The study also included a manipulation of whether the study and test words came from a related category (body parts, cities, or countries), but this factor turned out to be mostly irrelevant for our purpose, and will be discussed in just one of the coming sections. Since the role of this experiment is as a case study in the use of our framework, we will only provide the experimental design features pertinent to this aim, and leave a more detailed description of the experiment to the Supplementary Materials.

Methods

Thirty one participants were first asked to activate their microphones and grant permission for recording. After giving consent, they were then given general instructions explaining the task they were about to undertake. The instructions also included a statement that encouraged participants to use the think-aloud method. In short, they were asked to talk about how they made their decisions. They were asked to speak clearly and naturally, and were reassured not to worry about how they phrased their responses (for the exact wording, see the Instructions section in the Supplementary Materials). We will talk about the importance of such instructions in the Discussion section.

The main structure of the experiment was a Study phase, Distraction task, and Test phase (see Figure 2). In the Study phase, participants were shown a set of 32 items that they were instructed to remember for a later test. During the Distraction phase, to ensure that no studied items were still in working memory, participants performed a working-memory task for about 3 minutes. Finally, participants completed a test phase, where they were shown two items and were asked to indicate which of the two items were presented in the preceding study phase, and to indicate their confidence in that judgement using a slider that was marked to go from "guess" to "pretty sure" to "100% positive". Crucially, some test trials contained two words that had not been studied earlier, while the vast majority of trials were such that one of the words was previously studied. This study-test phase cycle was repeated three times for each participant before the experiment was finished. Finally, after they completed the experiment, the recording stopped and participants were asked whether they still consented to sending their recordings to our server.

Figure 2. An example of a trial's timeline



Note. The experiment began with asking for permission from the participant to activate their microphone. After obtaining consent, the instructions were presented, followed by the three experimental blocks of the memory recall task. In every of the experimental blocks (**X3** marked in red), in each trial, participants studied a list of context-related and context-unrelated words, completed a working memory distraction task, and then were asked to indicate on a Likert scale how certain they were about their memory of a pair of words. Upon completing all trials (i.e., all three experimental blocks), the recording was stopped, and the participant was asked again if they were willing to send their recorded voices to the server (crossed out microphone symbol).

Results

Descriptive Statistics

Behavioural Data. Participants' performance in the memory task is not of particular interest here, and so it suffices to say that the experiment yielded pretty 'typical' results. Namely, people were able to correctly indicate the items that were presented during the study phase with reasonable accuracy: they correctly identified "old" (previously studied) items on an average of .76 trials (SD = .075). Participants

also gave a confidence rating on a scale of -1 to 1, where 0 is indifference between the two test items and -1 and 1 are complete certainty about either the incorrect or correct item. The confidence ratings were calibrated with the experimental task, in the sense that they were less confident (M = .06, SD = .12) when neither of the test items were presented in the study phase (i.e., both test items were unstudied), than when one of the two items was presented earlier (M = .52, SD = .13). Finally, participants' confidence ratings were calibrated with their own memories, such that the magnitude of their confidence ratings for incorrect responses was lower (M = .43, SD = .19) than for correct responses (M = .78, SD = .11).

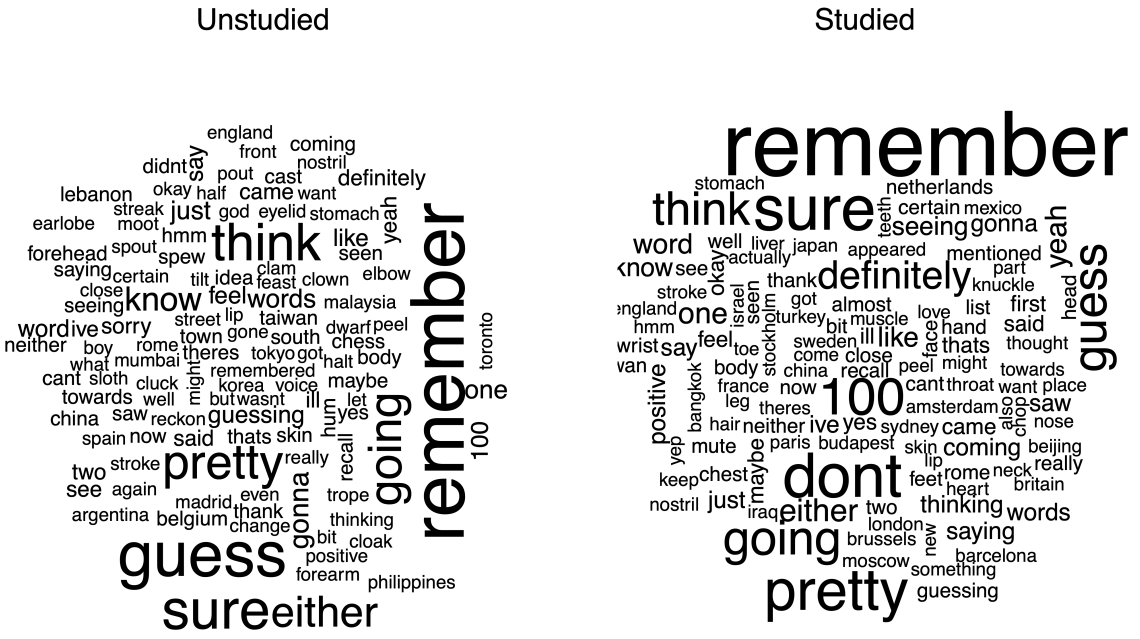
Verbal Reports. Before we evaluate the various components of our framework, we want to give a quick insight into the types of things that participants talked about during the experiment. To do this, we present simple word clouds for all verbal report data from test phase trials. We made word clouds separately for test trials that included a word that was studied earlier (i.e., Studied Trials) and when both test words were not studied earlier (i.e., Unstudied Trials). Figure 3, left cloud) shows that participants frequently used words indicating greater reliance on memory (e.g., "remember", "know") and higher confidence (e.g., "definitely", "pretty sure", and "sure") on Studied trials, but spoke as if they were uncertain (e.g., "guess", "think") on Unstudied trials (see Figure 3, right cloud). The word clouds also contain a lot of various nouns, and in almost all cases, these are the study items. This result makes sense, since looking at the raw verbal report data reveals that it was very common for participants to speak aloud the two test items presented on each trial.

We will now go through and describe how we used each of the components of our framework. Where possible, we will also provide a brief attempt at evaluating the effectiveness of the corresponding component.

Recording

In our study, the framework generated a total of 533 .wav files per participant. Across all test trials, where participants were asked to choose between two words, the

Figure 3. Word clouds for verbal reports Studied and Unstudied Trials



Note: Size of the words in the cloud represents the frequency of the words appearing in the verbal reports; the larger and bolder it is displayed, the higher the words' frequency.

average length of the .wav files was 6.08 seconds (SD = 5.02) and the average file size was 1.18 MB. During the study trials, where participants were shown the words with a fixation-cross in between, the timing of the trials was fixed at 2.5 seconds and as expected the recordings duration were on average 2.45 seconds long (SD = .034) and the mean data size per trial was .46 MB (SD = .12 MB).

Effectiveness Test. The effectiveness of the recording depends largely on the quality of participants' microphones, the length of their utterances, and how much they enunciate. We, the authors, listened to a decent sample of the recordings, and we shared a vague impression that the recordings were mostly good enough to be usable, but there were certainly participants whose audio data could reasonably be discarded. However, we do note that we did not exclude any such participants from what follows,

as we wanted the quantities we report to reflect the properties of the raw data.

Transcription

The second functionality of the software involved transcribing participants' recordings (see Figure 1, Output Box 4). We used the *large* version of the OpenAI Whisper model that has 1550 million parameters, was trained on 680,000 hours of labelled speech data annotated using large-scale weak supervision. We used the large version because the smaller versions produced noticeably, and often laughably, poor transcription performance (its inadequacy was evident from simply looking at the transcribed text in the .csv output files).

Effectiveness Test. We compared the software's transcriptions with those by human-raters to assess the accuracy and reliability of the software, providing an evaluation of the model's transcription capabilities (Radford et al., 2022). To do so, we selected a 5% sample from test trials and 5% from study trials across 3 blocks for each of the 31 participants, resulting in 404 test trials and 295 study trials. These sampled trials were then evaluated by the authors of the manuscript, who acted as the human raters for this test.

The evaluators listened to each of their sampled trials and created their own transcription into text. The rater then provided a subjective evaluation of whether the alignment between their transcription and that of the *Whisper* model was 'sufficiently aligned' (good enough) (e.g., Model transcription: "Kiev, tricking Kiev", human rater's transcription: "Kiev, chicken Kiev"). The rater could also indicate whether the recording was 'cutoff', meaning that the recording was not complete, and so it was only possible to transcribe the truncated part of what was spoken.

A summary of how often the match between the two transcriptions was perfect (i.e., exactly the same), 'good enough', and 'cutoff' is given in Table 2. It should become immediately clear why we included a subjective judgement of 'good enough' in our analysis, as a perfect match was rather rare. Differences between human and LLM transcriptions were considerably more likely for test trials, where the utterances were

typically much longer than for study trials, and so allowed for more possible errors. However, two things were clear from listening to the audio recordings and comparing the two transcriptions. First of all, the errors were typically unimportant from the perspective of what was meant by the speaker, which is most clearly seen by the fact that 'good enough' ratings, if anything, are higher for test trials than study trials. Second, doing the transcriptions from the audio recordings was very obviously an error-prone process, even for human raters. Often times when there the two transcriptions were inconsistent, it was also not particularly clear what was truly spoken, and hence whether either of the transcriptions was correct.

Finally, acknowledging the subjective nature of our 'good enough' measure, we also use the *word-error rate* as a more quantitative measure of the difference between the two transcriptions. This metric is widely used to evaluate the performance of speech recognition and machine translation systems. For instance, if the reference sequence is "I need to buy apple" and the system transcribes it as "I need buy apples", the error rate is calculated as .2 or 20%, reflecting one error (omission of "to") in a sequence of five words. The results in Table 2 suggest that this transcription model yields roughly 3 errors for every ten words it transcribes.

Embeddings

The third functionality of the software involves embedding sentences into 384-dimensional arrays (Figure 1 Output Box 5) using the *Sentence Transformer* model 'all-MiniLM-L6-v2'.

Effectiveness Test. The performance of LLMs is something that LLM developers are intensely monitoring, and the relative performance of all models is continuously changing. At the time of writing (September, 2024), the model we used was ranked 107th out of 477 on the 'Hugging Face' leaderboard (for a complete list of models, see https://huggingface.co/spaces/mteb/leaderboard), which provides a comprehensive comparison of models, highlighting their versatility, effectiveness, and strengths or weaknesses in various contexts. The model we used was higher on the

Table 2

Effectiveness Test for Voice Transcription

Label	Consistency Rate	
	All Trials	Test Trials
"Perfect match"	.36	.19
"Good enough"	.69	.74
"Cutoff"	.19	.19
Word Error Rate (WER)	.3	.3

Note. ***All Trials*** *include trials from both the Study Phase and the Test Phase (refer to Methods for details).* ***Test Trials*** *recordings consist of trials where participants rated their confidence on the familiarity of one of two words. Since this phase was not limited by time, compared to the Study phase, the recordings were typically longer.* ***Word Error Rate (WER)*** *represent the metric used to evaluate the consistency between humans and model's transcription.*

leaderboard when it was chosen for inclusion in the Framework, and by the time this manuscript is read by others, we expect it will have slid even further down the rankings.

Dimensionality Reduction

In the fourth step, the software applies dimensionality reduction to the sentence embeddings, originally in a 384-dimensional format (see Box 6 in Figure 1). While the framework offers two options, PCA (linear) and t-SNE, we show the output of just the latter. We reduced the embeddings using this method down to two dimensions. The distribution of utterances on each trial, broken down according to whether there were only unstudied items or a studied item present, is plotted in Panel A of Figure 4.

Effectiveness Test. Looking at the reduced embeddings in Figure 4, it appears that there is no systematic difference depending on whether the words were studied or not. However, this does not mean that the reduced embeddings did not have any systematic structure. To gain some understanding of what these new dimensions

Dimensionality Reduction & Word Clouds

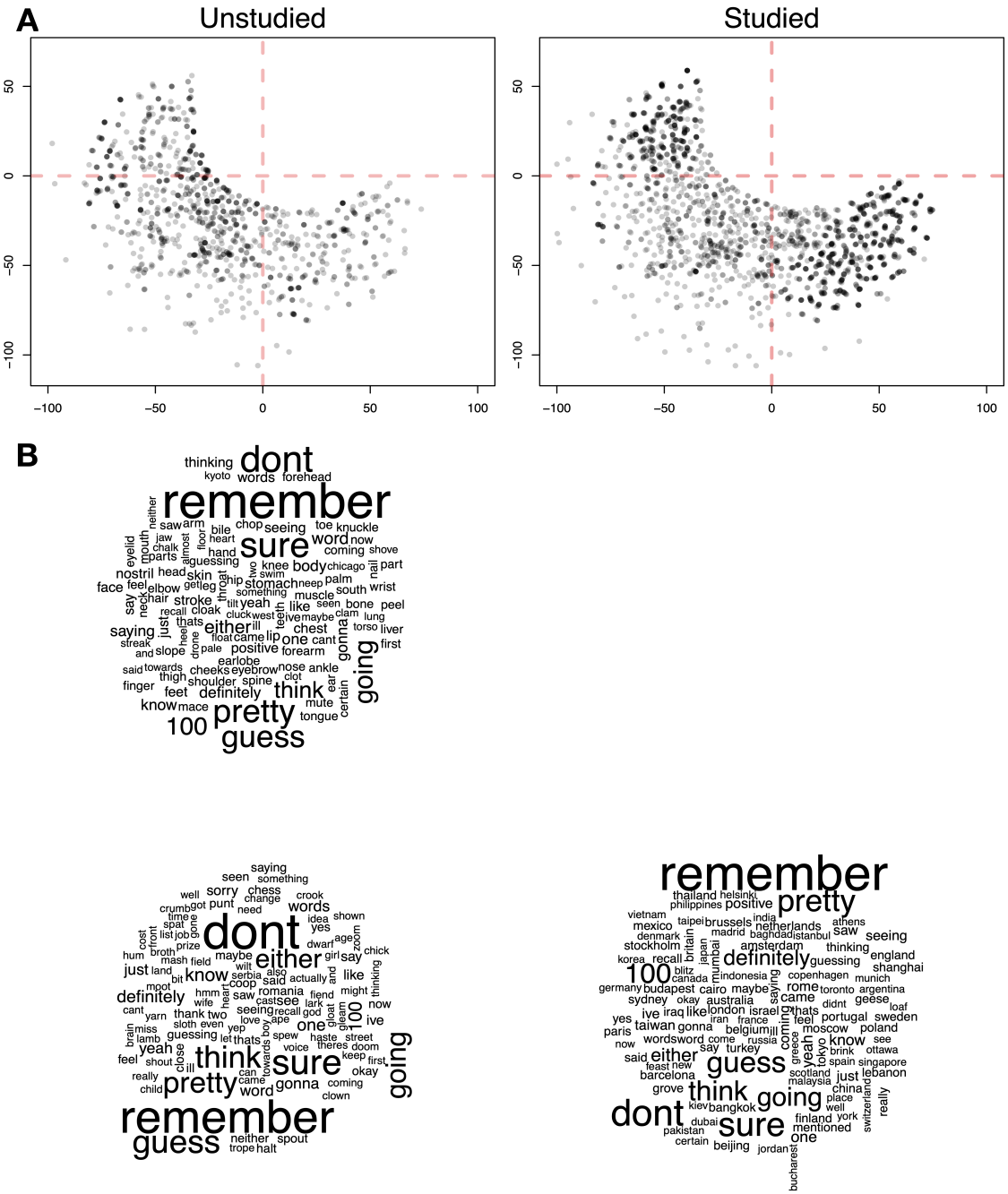


Figure 4. Dimensionality reduction applied to text data uttered by participants during experimental trials. Each point represents a unique utterance, mapped onto a two-dimensional space, with dashed red lines at the 0 mark to reveal patterns and clustering based on the content and context of the speech. The dashed red lines separate the data at the 0 mark to explore potential clusters. The left plot corresponds to the "Unstudied" trial type, where participants encountered novel words, while the right plot shows the "Studied" trial type, where participants were previously exposed to the words. The bottom subfigures represent word clouds, highlighting the most frequently uttered words present in each quarter of the plots where data was clustered

represent, we used word clouds to summarise the different sections of the reduced space and plotted the results in Panel B of Figure 4. Looking at the word clouds, we see that words about confidence or certainty occur often in all three of the quadrants in which there are data (the fourth, upper-right quadrant consistent of only 'missing' data, where the participant had said nothing during the recording). Furthermore, they seem to occur with roughly the same relative frequency, consistent with the fact that there are no systematic differences across unstudied and studied items.

The reduced embeddings seem to have captured differences in utterances not about certainty, but about the meaning of the nouns spoken. Namely, we see a lot of body parts in the top left quadrant, while the bottom right corner contains a lot of cities and countries. This result is unsurprising, given that half of the words in each study list was from one particular category: either body parts, countries, or cities. As such, this result reflects the fact that during the test phase, participants would typically refer to the words on screen while thinking aloud – e.g., "I definitely don't remember seeing Iraq". In other words, the dimensionality-reduction method has discovered the systematic structure that we built into the study and test items we used in our experiment.

Text Analysis

We decided that the most suitable text-analysis method for this experiment was the labeling pipeline, since the text was too brief for summarization or for key-word extraction. We supplied the pipeline with the transcribed text from participants along with a set of candidate labels, listed in Table 3. We now analyze the output of this pipeline, the most likely label for the text from each trial.

In Figure 5 we plot the frequency of label use, separated by whether the trial had a previously studied word present in the test trial (i.e., Studied) or not (Unstudied). As a visual aid, we color coded the 8 labels according to two rough categories: 'certain,' and 'uncertain,' using green and red colours, respectively. From the higher frequency of green bars in panel *b*, it appears that participants use more certain phrases on Studied trials than Unstudied trials.

Table 3

Mean Confidence Error - Effectiveness of Labeling

Label	Rank
absolutely uncertain	0
very uncertain	1
somewhat uncertain	2
a little uncertain	3
a little certain	4
somewhat certain	5
very certain	6
absolutely certain	7

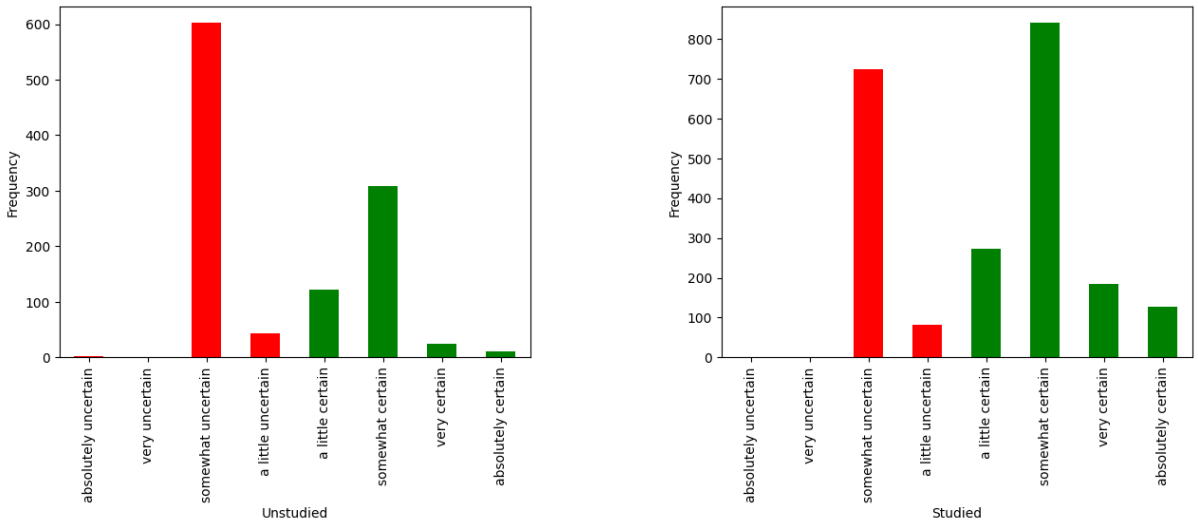
Note. Coding scheme for calculating mean confidence rate for labels

.

Effectiveness Test. To test the effectiveness of the labelling model's output, we compared the model's labels to those made by human raters, in a similar way we did for transcription evaluation earlier. A random sample of 10% of the "test" trials were given to human raters (again, these were the authors of the manuscript). The raters were presented with the transcribed text and were asked to choose the most appropriate label from the list of candidate labels that was given to the model.

A summary of the alignment between the model and human labels is presented in Table 4. We used three metrics to assess the functionality of labeling. Initially, we looked at how often the label assigned by the model and by human raters were identical. We call this measure the "Overall Accuracy", and was 36% for our sample, where chance performance would be 12.5% (since there were 8 labels). Like the transcription process, some of the distinctions among the labels were rather minor, and so we also looked at whether the labels were categorically similar. Following the color coding scheme in Figure 5, both human and model labels were dichotomized into "certain" and "uncertain", after which the agreement rate between the model's labels and those chosen by humans was 82%. As a final way of comparing the labels, we

Figure 5. Frequency of label use



(a) Unstudied Trial (b) Studied Trial

Note. The bars represent the frequency of certainty in participants' verbal reports transcribed from two experimental trial types: 'Unstudied' and 'Studied'. The x-axis represents the reports based on certainty levels ranging from 'absolutely uncertain' to 'absolutely certain'. Red bars represent a meta label of 'uncertainty', while green bars represent 'certainty'.

assigned the eight labels the values 1 to 8, reflecting the increase from absolute uncertainty to absolute certainty (see Table 3). Using these numerical ranking values, we calculated the mean absolute error (MAE) between the model's predicted labels and the human labels. The MAE was approximately 1, suggesting that the model's predictions are typically off by one class (see Table 4).

Discussion

In this paper, we introduce a newly developed framework that works within the jsPsych platform, designed to facilitate the collection and analysis of verbal reports in both offline and online studies. It automates numerous aspects of the collection of verbal report data, offers a range of possible analyses, and is readily expandable to incorporate new functionality. We used a case study as a proof-of-concept for the

Table 4

Effectiveness Test for Labeling

Test	Consistency Rate
Overall Accuracy	.36
Binary Accuracy	.82
Mean Confidence Difference (MAE)	1.03

Note. All methods represent the metric used to evaluate the consistency between humans and model's transcription. ***Overall Accuracy*** *represents a perfect match between the human and the models' assigned labels.* ***Binary Accuracy*** *refers to the match between the less nuanced labeling method, namely, the more rough classification of labels as either 'uncertain' or 'certain'.* ***Mean Confidence Difference*** *refers to the average deviation of the model (here, it is by around one class)*

.

framework, and, where possible, provided relatively crude evaluations of the efficacy of various components of the framework. We will break the following discussion into two parts – first focusing on issues that came up while we developed the framework, and second, on the issues that arose when we used the framework during the case study.

Developing the Framework

Modularity and Adaptability. The current framework was designed with modularity as its main objective. This means all components, including text-analysis pipelines, models, and their sizes, are adjustable. We viewed such flexibility as necessary, because the models and methods available will rapidly become more sophisticated and potentially more affordable, which will enhance performance across the various functionalities (e.g., embeddings, transcription, and labeling). Furthermore, while our goal was to package together a core functionality that would be useful for most researchers – recording, transcription, and embeddings – we expect that the requirements of any individual research project will be highly variable.

Arriving at a version of the framework that would accompany this manuscript, and

that we used in in our case study, involved a number of choices. In our analyses, we chose specific models — such as the *Whisper model* for transcription, the *SentenceBERT model* for embedding, and the *Zero-shot model* for labeling — as well as various text analysis tools and techniques. These choices were based on practical considerations like whether the models were open source, whether the models could be run locally or needed a server, whether using the models cost money, where data would be stored, how large the data files produced would be and how much it would cost to store them, as well as other privacy issues (which we will return to shortly). Such questions, and more, will go into the choices about whether certain components in our framework need to be changed or replaced. The default set of methods and models we provide here were chosen to keep costs down, and hence it is likely that more accurate or better performing methods are available.

As of release, some of the components of our models have limitations. For instance, for the transcription task, our choice of the *Whisper model* in its current version has a difficulty in accurately recognizing words from verbal reports that are brief, lack coherence, or contain grammatical irregularities. However, transcription models are expected to improve over time as larger models are developed and become more sensitive, making them better equipped to handle such complexities.

We also expect dimensionality reduction techniques to improve in the near future. One promising example is to use an autoencoder, which is a neural network that uses multiple hidden layers to learn to reproduce the high-dimensional data it is fed. The hidden layers of the autoencoders are of increasingly smaller dimensionality, and hence compress data into simpler representations. Since implementing this technique requires training a neural network model, we chose not to include it into our framework, but it is the kind of analysis that we might expect to become common in the near future.

Data Privacy. One overarching concern during development of the framework was considerations about data privacy. To that end, we have implemented a number of particular mechanisms. Firstly, participants are required to confirm that they are being recorded. After the study is completed, participants must again confirm the use of their

data. If a participant refuses, all recordings and data will be deleted from the server. Additionally, if a participant closes their browser window, their data will also be deleted as they were unable to confirm the use of their data. Each participant's data is anonymized using a Universally Unique Identifier (UUID) that is created specifically for them. Finally, everything can be processed locally, so that we can avoid using any third-party services like Google or OpenAI, where it is not always clear whether the data is being kept or used.

Timing. One of the most difficult decisions researchers must make when collecting verbal-report data is when it should be elicited. For us, that meant determining how best to record and segment the verbal recordings. We chose to record for the entirety of the experiment, because that seemed to allow for the vast majority of use cases we could imagine. For example, the researcher can instruct participants to speak whenever they want, or they could prompt participants to respond at particular times (e.g., after a behavioral response has been made, but before the next trial). In the latter case, a lot of what is recorded might be silence, but such irrelevant portions of the recording/transcriptions can be ignored during the data analysis phase.

The only complication in using continuous recording is that the user of the framework must consider and control how the recordings will be segmented in the output data files. For instance, if there is a reason to believe that participants might provide useful information while viewing a fixation cross, then the appropriate change to the jsPsych code must be made to make sure that any spoken text during the fixation cross is included as part of a jsPsych 'trial'. This way, both the stimulus presentation and the following fixation cross are treated as a single trial for data segmentation, recording, and transcription. Similarly, if the researcher wants to add some time for verbal report after a behavioral response, they need to add some time to the experiment in jsPsych, and perhaps have participants push a button to proceed (or advance after some predetermined time).

Evaluating the efficacy of the framework. One of the goals of our case study was to evaluate how well the framework would work as a user. We began this evaluation

entirely informally, in the sense that we listened to samples of audio, read transcriptions, looked at the outputs of analyses with different settings, etc., until we had a sense that things were working well. For the manuscript, we came up with slightly more regimented ways to test things, in which we took samples of text and audio, and generated our own transcriptions and labels. The reader will have already presumably surmised that we did not intend these to be rigorous benchmarks or tests. Rather, we report on the efficacy of the framework for two reasons. First, we simply wanted to provide a proof-of-concept demonstration that the methods work. Second, we thought that such methods could give the reader an idea of how they might provide their own internal, informal checks that things are working as intended.

The reason for not carrying out careful and rigorous tests of this release version of the framework is that, because of the speed with which the specific models being used are developed and thus outdated, it would largely be a waste of time and effort. We expect that by the time this manuscript is published, users will be able to use newer and better transcription models. Furthermore, any performance we were able to record would be specific to the nature of our experiment (i.e., timing of recordings, the nature of the spoken utterances, etc.), and would be unlikely to generalise to other use cases for the framework.

Our relatively informal approach to testing the efficacy of our framework is only possible because of an interesting property of verbal-report data, which is that it is relatively easy to check whether things are working properly. That is, unlike the vast majority of dependent variables, all researchers have a well-developed understanding of the spoken word, and how to know what it means. As such, anyone can simply look at the outputs of the framework, or what was done in a given analysis, and check whether it makes sense. The fact that verbal reports are more inherently 'transparent' is a peculiar advantage, which makes it easier to criticise than many other dependent variables, such as fMRI or EEG signals. For example, other researchers can easily understand and critique the choice of labels we used, and use their own intuitions about whether a language model has made the same kind of label assignment as they would

have. For other variables, it is much harder to validate analyses as sensible or valid. We hope that this property makes it easier than is typical to develop good methods of analysis, and draw sensible conclusions from resultant data.

Lessons from the Case Study

Noisy Data. One thing that is immediately obvious, is that verbal reports are noisy. As its name implies, the think-aloud method can yield anything. As much as we would like them to, participants need not clearly articulate how they solved the task at hand. However, we are used to this as researchers, as it is no different from the fact that behavioral responses need not be made for the reasons that we expect or hope. In our case study, we also had noise in our verbal data because some reports were too short, were truncated by trial segmentation, poor microphones, or the utterances simply lacked any coherent explicable meaning.

As researchers, we can take steps to reduce the noise in verbal reports, and two kinds of strategies seem possible. First, we can improve the analysis/models. To some extent, this will happen naturally, since over time, we have already seen models become increasingly good at finding the signal in noisy data, even in shorter utterances. We can also adapt our analyses. For example, we had many trials during the study phase of the memory task in which participants just read the to-be-remembered word aloud. If accurate transcription was important for this aspect of our study, then we could have fine-tuned a transcription model to be sensitive to the study items.

Second, we could also increase the accuracy of transcription and embeddings by increasing data volume – whether through longer verbal reports. Longer verbal reports provide more context, enabling current models to better correct unclear words and "regularize" responses by replacing unusual words with more common ones. Clear instructions throughout the experiment can encourage participants to give more detailed responses. Ensuring that the structure of the experimental design permits sufficient time for participants to fully express their thoughts before moving on can also help to reduce the number of truncated responses. For example, we might have reduced

the number of truncated responses by requiring participants to press a button to confirm that they were finished talking about the current trial. As mentioned earlier, the modularity of jsPsych and our framework supports these adaptations. Researchers can decide how to define jsPsych 'trials' (i.e., what should be included in the trials) and can place instructions and buttons as needed.

Lastly, it's also important to note the variability in the quality of our collected recordings. In some trials, it was difficult to discern exactly what was said due to the presence of background noise. This quality variation is expected, especially in online experiments. Using high-quality microphones and encouraging clear speech will improve the clarity of verbal report recording and transcriptions. For studies that require high transcription accuracy, conducting experiments offline in a lab setting with standardized equipment can ensure consistent audio quality.

Dimension Reduction. Our analyses focused on how people talked about their confidence depending on whether they were trying to remember words they had studied and those they had not. We expected that the dimensionality reduction analysis would reveal systematic differences in expressions of certainty, since we did observe people talk a lot about how confident they were. However, our analysis indicated that other aspects of the verbal reports were more important. Indeed, we found no systematic differences between studied and unstudied words, but rather found that the reduced dimensions captured variation in the meanings of the nouns spoken, rather than certainty about word recognition or the concept of context. This is not to say that the analysis was flawed, but a reminder that dimensionality reduction methods are theoretically agnostic, and will find whatever structure is present. As such, the onus is on the researcher to design experiments that encourage people to speak more or less about certain things.

There are several effective strategies for managing the spoken content in verbal reports to increase the chances that the reduced embeddings will capture what a researcher intends. As previously discussed, researchers can direct participants to focus on specific topics of interest instead of allowing them to choose their topics freely. For instance, participants could be encouraged to talk about their confidence in their

choices or explain how the associations between nouns (i.e., context) helped them with memory recall. We might also have asked participants not to say aloud the words they had to remember, however, this might have affected the way that people performed in the experiment, and so care must be taken with such instructions. Another approach is to alter the analyses, for example, by excluding from the dimensionality reduction, the trials on which participants said the study words, or by replacing all such words with a single word. We also could have looked at dimensionality reduction solutions with more dimensions, as it might have been that the third dimension was about confidence.

Transcription & Text Analysis Pipelines. Our case study results showed a relatively poor alignment between human raters and machine/model transcriptions in terms of the precise labels that were chosen. Indeed, we saw similar discrepancies when we used an even greater variety of possible labels (in analyses not presented here). However, in many cases, the choice between specific candidate labels, as a human rater, felt largely arbitrary, and so we would expect similar discrepancies in human classification. More importantly, it was rare that the machine classifier would make a completely incongruous classification (i.e., where it appears to have lost all semantic meaning). For the sake of analysis that uses aggregation, perfect transcription is hardly crucial. Indeed, we accept some level of disagreement among human raters. One benefit here is that we can overcome such noise with substantially larger data sets, and so labeling can still preserve the semantic meaning, despite the inevitable discrepancies with human judgement.

While we can largely ignore problems of noisy labeling, we should be sensitive to biases in machine-based analyses. For example, in our labelling analyses we had the algorithm classify all utterances according to their degree of certainty. Now, a classifier will make such a determination, regardless of whether or not the participants actually talked about their confidence on a given trial. As such, it is possible for these analyses to produce misleading results, since certain words or phrases might be systematically assigned certain labels, regardless of whether it makes any sense. For example, the classifier in our analysis might have consistently labelled country or city names as 'high

certainty'. This would have produced a misleading bias in the analysis, which might have caused us to conclude that people were more confident when making decisions about proper nouns, say. There is no prescriptive method for ruling out such biases, they can only minimized by researchers who remain vigilant when checking that their conclusions are not due to trivial features of an automated analysis.

Conclusion

We have presented a practical framework tailored for gathering and evaluating verbal reports in computer-based psychology experiments. It provides researchers with the ability to deal with many of the biggest issues with verbal-report data. Most importantly, work that has typically required human judgement can now be done by machine-learning methods, thus making the transcription and analysis of verbal reports much more scalable. We showed that the framework works using a simple memory experiment. The code we provide can be used as is, meaning that psychology researchers should be able to start incorporating verbal reports into their experiments. However, the framework is modular, meaning it can be updated and expanded as methods are improved and discovered, and the full potential of quantifying verbal data is realised.

The data that support the findings of this study are available at https://osf.io/xa7rk/. Data are available upon submission.

Open Practices Statement

The data for the experiment is available at https://osf.io/xa7rk/. The software and analysis scripts, a friendly user guide and the materials are publicly available at https://github.com/tehillamo/AutoV-LLM.

Declarations

Funding. Not applicable.

Conflicts of interest/Competing interests. No potential conflicts of interest were reported by the authors.

Ethics approval. Our case study (see Section “Case study”) was approved by the Ethics Committee at Ludwig Maximilian University, Munich, the Faculty of Psychology and Pedagogic, with the name ”Verbal2Validate: Automating Think-Aloud’ Methods for Cognitive Model Validation“

Consent to participate. Informed consent was obtained from all individual participants included in the study (see more information about data privacy under “Data Privacy” section).

Consent for publication. All participants provided consent for publication of their data (note, participants were asked to give consent to publish their **transcribed voice** only).

Availability of data and materials. The dataset generated and analysed for the current study are available in the "AutoV-LLM" repository under https://osf.io/xa7rk/.

Code availability. The analysis scripts as well as the software’s scripts are publicly available at https://github.com/tehillamo/AutoV-LLM.

Authors’ contributions. Tehilla Ostrovsky and Chris Donkin designed the study and conducted the experiment. Tehilla Ostrovsky and Chris Donkin and Paul Ungermann wrote the manuscript. Paul Ungermann and Tehilla Ostrovsky contributed to software development and data analysis. Chris Donkin supervised the research.

References

De Leeuw, J. R. (2015). jspsych: A javascript library for creating behavioral experiments in a web browser. *Behavior research methods*, *47*, 1–12.

Enkavi, A. Z., Eisenberg, I. W., Bissett, P. G., Mazza, G. L., MacKinnon, D. P., Marsch, L. A., & Poldrack, R. A. (2019). Large-scale analysis of test–retest reliabilities of self-regulation measures. *Proceedings of the National Academy of Sciences*, *116*(12), 5472–5477.

Ericsson, K. A., & Simon, H. A. (1980). Verbal reports as data. *Psychological review*, *87*(3), 215.

Fox, M. C., Ericsson, K. A., & Best, R. (2011). Do procedures for verbal reporting of thinking have to be reactive? a meta-analysis and recommendations for best reporting methods. *Psychological Bulletin*, *137*(2), 316–344.

Frey, R., Pedroni, A., Mata, R., Rieskamp, J., & Hertwig, R. (2017). Risk preference shares the psychometric structure of major psychological traits. *Science advances*, *3*(10), e1701381.

Grootendorst, M. (2020a). *Keybert: Minimal keyword extraction with bert.* Zenodo. Retrieved from https://doi.org/10.5281/zenodo.4461265 doi: 10.5281/zenodo.4461265

Grootendorst, M. (2020b). *Keybert: Minimal keyword extraction with bert.* Zenodo. Retrieved from https://doi.org/10.5281/zenodo.4461265 doi: 10.5281/zenodo.4461265

Hedge, C., Powell, G., & Sumner, P. (2018). The reliability paradox: Why robust cognitive tasks do not produce reliable individual differences. *Behavior research methods*, *50*, 1166–1186.

Ostrovsky, T., & Newell, B. R. (2024). Verbal reports as data revisited: Using natural language models to validate cognitive models. *Decision*, *11*(4). doi: https://psycnet.apa.org/doi/10.1037/dec0000243

Radford, A., Kim, J. W., Xu, T., Brockman, G., McLeavey, C., & Sutskever, I. (2022). *Robust speech recognition via large-scale weak supervision.*

Ranyard, R., & Svenson, O. (2011). *Verbal data and decision process analysis.* Psychology Press Oxford, UK.

Rouder, J. N., & Haaf, J. M. (2019). A psychometrics of individual differences in experimental tasks. *Psychonomic bulletin & review*, *26*(2), 452–467.

Sanh, V., Webson, A., Raffel, C., Bach, S. H., Sutawika, L., Alyafeai, Z., ... Rush, A. M. (2022). *Multitask prompted training enables zero-shot task generalization.* doi: https://doi.org/10.48550/arXiv.2110.08207

Svenson, O. (1979). Process descriptions of decision making. *Organizational Behavior and Human Performance*, *23*(1), 86–112.

Svenson, O. (1989). Eliciting and analyzing verbal protocols in process studies of judgment and decision making. *Process and structure in human decision making*, 65–81.

Szollosi, A., & Donkin, C. (2019). Neglected sources of flexibility in psychological theories: From replicability to good explanations. *Computational Brain & Behavior*, *2*(3), 190–192.

Van der Maaten, L., & Hinton, G. (2008). Visualizing data using t-sne. *Journal of machine learning research*, *9*(11).

Yin, W., Hay, J., & Roth, D. (2019). Benchmarking zero-shot text classification: Datasets, evaluation and entailment approach. *arXiv preprint arXiv:1909.00161*.