Real-time classification of auditory sentences using evoked cortical activity in humans

David A Moses[1,2,3], Matthew K Leonard[1,2], and Edward F Chang[1,2,3]‡

[1] Department of Neurological Surgery, UC San Francisco, CA, USA
[2] Center for Integrative Neuroscience, UC San Francisco, CA, USA
[3] Graduate Program in Bioengineering, UC Berkeley-UC San Francisco, CA, USA

E-mail: David.Moses@ucsf.edu, Matthew.Leonard@ucsf.edu, ChangEd@neurosurg.ucsf.edu

Abstract. *Objective.* Recent research has characterized the anatomical and functional basis of speech perception in the human auditory cortex. These advances have made it possible to decode speech information from activity in brain regions like the superior temporal gyrus, but no published work has demonstrated this ability in real-time, which is necessary for neuroprosthetic brain-computer interfaces. *Approach.* Here, we introduce a real-time Neural Speech Recognition (rtNSR) software package, which was used to classify spoken input from high-resolution electrocorticography signals in real-time. We tested the system with two human subjects implanted with electrode arrays over the lateral brain surface. Subjects listened to multiple repetitions of ten sentences, and rtNSR classified what was heard in real-time from neural activity patterns using direct sentence-level and HMM-based phoneme-level classification schemes. *Main results.* We observed single-trial sentence classification accuracies of 90% or higher for each subject with less than 7 minutes of training data, demonstrating the ability of rtNSR to use cortical recordings to perform accurate real-time speech decoding in a limited vocabulary setting. *Significance.* Further development and testing of the package with different speech paradigms could influence the design of future speech neuroprosthetic applications.

PACS numbers: 87.19.L, 43.71.An, 43.71Es, 43.71.Qr, 43.71.Sy, 43.72.Ne, 87.85.D-, 87.85.E-, 87.85.Wc

Submitted to: *J. Neural Eng.*
Keywords: neural speech recognition, real-time speech classification, speech perception, electrocorticography, high gamma, human auditory cortex

1. Introduction

Recent work has characterized the specific functional roles of the human superior temporal gyrus (STG) and neighboring brain areas in speech perception and language

‡ Author to whom any correspondence should be addressed.

understanding [1–6]. While subjects are listening to spoken speech, neural activity in this region can be used to decode and reconstruct speech information, including spectrotemporal acoustic properties [7–9] and phoneme sequences [10]. Previous work has implemented real-time systems capable of mapping sensorimotor activations using spectral decomposition of neural signals [11], using transcribed stimuli to generate neural encoding models (as opposed to decoding models) of segmental speech (e.g. phonemes) [12], decoding isolated phonemes from brain activity [13], and detecting speech production onsets and offsets from cortical responses [14]. However, to the best of our knowledge no published work has demonstrated real-time classification of phoneme sequences or entire sentences from neural signals, which would have practical applications in speech neuroprostheses.

In this work, we developed and tested a real-time Neural Speech Recognition (rtNSR) software package. As defined in our previous work, we use the term neural speech recognition to refer to performing speech decoding using neural responses as features [10]. The rtNSR package contains real-time code capable of presenting visual and acoustic stimuli, processing acquired neural signals, training probabilistic models, performing classification and decoding, and storing data and metadata. Our primary goal in this work was to perform an initial assessment of the capabilities of rtNSR using a relatively simple sentence prediction task. In this task, subjects listened to multiple presentations of ten pre-recorded spoken sentences. During these stimulus presentations, cortical activity is obtained in real-time via electrocorticography (ECoG) arrays and used in one of two classification schemes to predict the identity of the stimulus that the subject just heard. The results of this study indicate that rtNSR is capable of accurately decoding single-trial speech events in real-time, demonstrating its viability as a platform for an assistive speech application.

2. Methods

2.1. Subjects

The two subjects (A and B) who participated in this study were human epilepsy patients undergoing treatment at the UCSF Medical Center. To aid clinicians in localizing seizure foci, two 128-channel ECoG arrays with 4 mm center-to-center electrode spacing (PMT corp.) were surgically implanted on the cortical surface of each subject. Both subjects had unilateral coverage over the right hemisphere that included the STG. MRI Brain reconstructions with electrode locations were generated for each subject using the open source *img_pipe* package (see figure S1) [15].

Both patients gave their informed consent to be a subject for this research prior to surgery. The research protocol was approved by the UCSF Committee on Human Research.

Table 1: Information about each stimulus.

TIMIT label	Sentence transcription	Duration (s)
fcaj0_si1479	Have you got enough blankets?	1.108
fcaj0_si1804	It had gone like clockwork.	1.540
fdfb0_si1948	He moistened his lips uneasily.	1.527
fdxw0_si2141	It was nobody's fault.	1.161
fisb0_si2209	"A bullet," she answered.	1.508
mbbr0_si2315	Junior, what on earth's the matter with you?	1.679
mdlc2_si2244	Nobody likes snakes.	1.301
mdls0_si998	Yet they thrived on it.	1.000
mjdh0_si1984	And what eyes they were.	1.048
mjmm0_si625	A tiny handful never did make the concert.	2.106

Table 2: The phonemic labels used in this work and their respective categorizations.

Category	Phoneme
Silence	sp
Stop	b d g p t k
Affricate	jh
Fricative	f v s z sh th dh hh
Nasal	m n ng
Approximant	w y l r
Monophthong	iy aa ae eh ah uw ao ih uh er
Diphthong	ey ay ow oy

2.2. Speech stimuli

In each experimental task, the subject listened to multiple repetitions of ten phonetically transcribed speech stimuli from the Texas Instruments/Massachusetts Institute of Technology (TIMIT) dataset [16]. In each stimulus, a single speaker produces a single sentence. We trimmed silence from each end of each stimulus sound file prior to running the tasks. The TIMIT label, sentence transcription, and duration of each stimulus are provided in table 1.

We converted each speech sound label specified in the phonetic transcriptions to one of the 37 phonemic labels used in this work. This set of phonemic labels, which is provided in table 2, is comprised of 36 phonemes from the Arpabet and /sp/, a silence phoneme used to label non-speech data points.

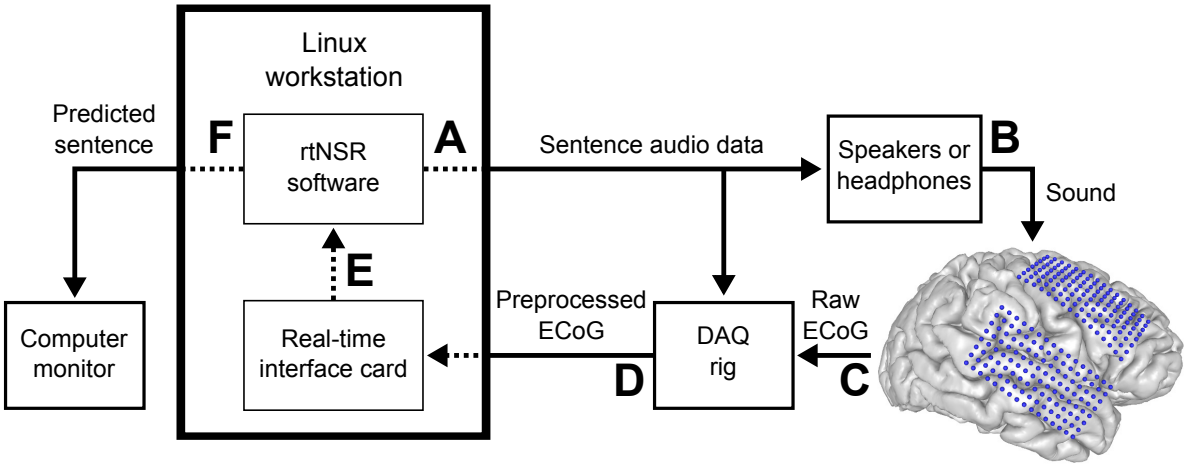

Figure 1: A schematic depiction of the real-time stimulus prediction system with the letters A-F denoting the flow of information throughout the system. A Linux workstation implementing rtNSR plays the stimuli to the subject during ECoG data collection (A-B). The raw ECoG signals are amplified, preprocessed, and synchronized with the audio data in the data acquisition (DAQ) rig (C). The preprocessed ECoG signals are streamed to the workstation through a real-time interface card (D). The rtNSR software acquires the signals from the card, processes them, and uses them to perform sentence classification (E). Sentence predictions are displayed on a computer monitor (F). The MRI brain reconstruction for subject A is shown here with electrode locations depicted as blue dots. Electrode coverage was similar for subject B (see figure S1).

2.3. Real-time processing setup

An overview of the real-time stimulus prediction system is depicted in figure 1. The capital letter labels in this figure correspond to the data flow steps during each stimulus presentation (each trial) in each task block. At the start of each trial, a Linux workstation (64-bit Ubuntu 14.04, Intel Core i7-4790K processor, 32 GB of RAM) implementing rtNSR plays one of the stimuli to the subject (A-B). Simultaneously, the implanted ECoG arrays record cortical local field potentials at 256 cortical sites, which are processed in the data acquisition (DAQ) rig (C). Within the DAQ rig, the ECoG signals are amplified and quantized at 3051.76 Hz using a pre-amplifier (PZ2, Tucker-Davis Technologies) and preprocessed using a digital signal processor (RZ2, Tucker-Davis Technologies). Before the ECoG signals are preprocessed, they are stored on the rig along with the time-aligned audio waveform. During preprocessing, the signals are notch filtered at 60, 120, and 180 Hz to reduce line noise. Next, each channel is band-passed at 70–150 Hz, squared, and smoothed using a low-pass filter at 10 Hz to extract power in the high gamma band. High gamma power was used because previous research has shown that activity in this band strongly correlates with multi-unit activity [17] and is associated with important speech features [6, 7, 10]. These high gamma signals are then decimated to 98.44 Hz and streamed to the Linux workstation using a real-time interface card (PO8e, Tucker-Davis Technologies) where they are processed in rtNSR and saved to disk for offline analyses

(D-E). Further discussion of preprocessing considerations and feature extraction are available in section 4.

Within rtNSR, the signals acquired from the real-time card are normalized by z-scoring the data for each channel using a 30-second sliding window. These z-score values are clipped to lie within the range of $[-2, 2]$ to mitigate signal artifacts caused by epileptic activity, channel noise, or other factors. If a trained model is available, then, immediately after the stimulus presentation, signals from relevant channels are used as features in this model to predict which stimulus was just presented to the subject (detailed descriptions of the channel selection and modeling procedures are given in section 2.6). The stimulus prediction and updated running classification accuracy measures are displayed on a monitor (F).

2.4. rtNSR design

Our rtNSR system is implemented in the Python programming language [18] and is designed for flexible and efficient real-time neural signal modeling and speech decoding. Based on the software pipelining implementation technique [19], rtNSR uses multiple data processing elements that run in parallel as individual processes. Typically, each of these processes obtains inputs from one or more separate processes (via software pipes or shared memory buffers), performs a specific task with or manipulation on the inputs, and sends outputs to one or more other processes. Each process is defined as a sub-class of a parent real-time process class implementing general methods for real-time processing (including data sharing and process setup methods). rtNSR contains many of these single-purpose process classes, such as a process that reads streaming data from the real-time interface card and a process that performs sliding window normalization. This highly modularized software architecture allows for individual steps in the real-time processing workflow to be interchanged and rearranged with relative ease while leveraging the computational efficiency associated with pipelining and parallelization. For example, during real-time simulations performed offline for debugging and system evaluation, we simply replaced the real-time card reader process in the data processing workflow with a process that loads and streams out pre-recorded neural data. A block diagram depicting the rtNSR components and data flow used during the real-time experiments is provided in figure 2.

2.5. Experimental task blocks

For subject A, we collected a total of 300 stimulus presentations (30 for each stimulus) across a total of 4 task blocks. For subject B, we collected a total of 250 stimulus presentations (25 for each stimulus) across a total of 3 task blocks. At the start of each block, a 1-second beep is played to signal the start of the task. This sound triggers an audio onset detector in the preprocessor to inject a start token into an arbitrarily chosen recording channel. The sentences are then presented with a constant onset-to-onset time interval. As a result, rtNSR can easily keep track of which neural data points

are associated with each stimulus presentation (see section 4 for further discussion on stimulus timing). Within each block, we randomized the stimulus presentations while ensuring that each stimulus was presented an equal number of times.

In each task block, the onset-to-onset interval was approximately 2.57 seconds, the stimuli were presented aurally via loudspeakers, and the subject was not able to see the real-time stimulus classifications. However, in the final block for subject A, the onset-to-onset interval was approximately 5.14 seconds, the stimuli were presented using headphones, and the subject was instructed to view the real-time sentence classifications and respond with either a "thumbs up" or a "thumbs down" to indicate if the prediction matched what was heard through the headphones (see supplementary video 1). The extra time in the onset-to-onset interval for this block was not used during modeling and was only included to allow the subject to respond before the onset of a new sentence.

2.6. Stimulus classification schemes

Stimulus classification models were trained for each subject using data collected during experimentation. Each time a model was trained, the collected data were first analyzed to identify which channels should be considered relevant to speech perception processing [10]. A simple bad channel detector was used to exclude any channels for which 75% or more of the acquired data points had a z-score of 0.25 or less. Afterwards, two data subsets were created: one subset comprised of neural data sampled during sentence perception of each stimulus presentation (30 time points per stimulus presentation) and another similarly constructed subset containing data points sampled during the silence after each sentence. Two-tailed Welch's t-tests were then performed for each channel between the two data subsets. Channels that exhibited a p-value less than 0.001 were considered relevant (significantly modulated by the presence of auditory speech stimuli) and the remaining channels were excluded during modeling. Applying these procedures to the data acquired before the final testing block resulted in 79 and 122 relevant electrode channels for subjects A and B, respectively (see figure S1).

We used two types of real-time stimulus classification schemes in our tasks: a "Direct" classification scheme during testing with subject A and an "HMM-based" classification scheme during testing with subject B. As described in section 2.4, we were able to slightly modify the experimentation setup to simulate stored neural data as if it were being obtained in real-time without altering the classification scheme functionality. This enabled us to compute results for each subject using the classification scheme that was not used during real-time testing for that subject. After data collection and these offline simulations, results using both schemes were available for each subject. We used the scikit-learn Python package to implement the models employed in both schemes [20].

2.6.1. Direct classification scheme In the direct classification scheme, each stimulus (sentence) was treated as one of ten classes. The feature vectors used during classification were each constructed by concatenating the z-scored high gamma power values for each

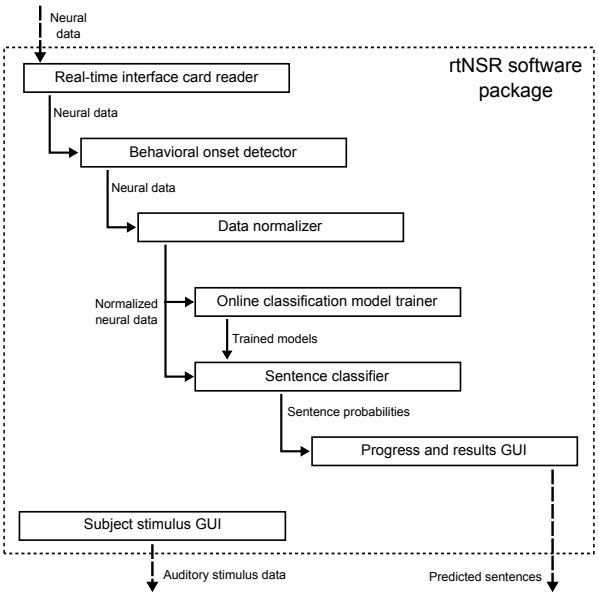

Figure 2: A schematic depiction of the rtNSR implementation used during experimentation. The solid rectangles represent real-time process classes and the arrows represent data that is passed between processes. The *Real-time interface card reader* process reads neural data streamed from the real-time interface card. This data is passed to the *Behavioral onset detector* process, which detects a one-time injected onset token that signifies the start of the task (see section 2.5). The neural data is then passed to the *Data normalizer* process, which performs sliding window normalization and magnitude clipping. The normalized neural data is passed to the *Sentence classifier* process where the data is used to perform sentence classification. This process outputs sentence probabilities to the *Progress and results GUI* process, which extracts the most likely sentence from each of these sentence probability vectors and displays each predicted sentence on a monitor. When using the direct classification scheme, the *Online classification model trainer* process also obtains the normalized neural data, performs model training and relevant electrode selection in real-time, and passes trained models (with relevant electrode numbers) to the sentence classification process (see section 2.6.1). Throughout the task, the *Subject stimulus GUI* process controls auditory presentation of the sentence stimuli to the subject.

relevant channel at each time point during a stimulus presentation. Because the stimuli varied in duration, some of the neural data obtained during the silence periods after a stimulus presentation were included in the feature vector associated with that stimulus presentation. The feature vector for each stimulus presentation contained the neural data at each of the $T = 253$ time points associated with that presentation (which spans the 2.57 second time window allotted for each presentation, as described in section 2.5). For example, a stimulus presentation that began at time index t would be associated with a feature vector containing the neural data points for each relevant channel at time indices $\{t, t+1, \ldots, t+T-1\}$ (with a length of T times the number of relevant channels) and with a target label equal to the identity of that stimulus.

During model training, we use principal component analysis (PCA) to reduce the

dimensionality of the feature vectors to the minimum number of features required to explain at least 99% of the variance. The resulting feature vectors have lengths that are typically around 100 elements (less than 1% of the lengths of the original vectors). These new feature vectors are used to train a linear discriminant analysis (LDA) model implementing the least-squares solution with automatic shrinkage using the Ledoit-Wolf lemma [21]. Once trained, we used these combination PCA-LDA models to classify previously-unseen neural responses into one of the ten stimulus labels in real-time. Model training, which typically took 2–5 seconds, was first performed in real-time using all available data for a subject when at least 2 repetitions of each stimulus were presented and subsequently performed prior to starting a new task block and in real-time whenever 10 stimulus presentations had occurred since the most recent training.

2.6.2. HMM-based classification scheme In the HMM-based classification scheme, each stimulus is represented as a hidden Markov model (HMM), where each hidden state q_t is the phoneme that occurs at time index t for that stimulus and each observed state y_t is the neural feature vector associated with time index t. This classification scheme was inspired by the phoneme decoding results described in [10].

For HMMs, the joint probability is normally

$$p\left(q,y\right)=p\left(q_0\right)\prod_{t=0}^{T-2}p\left(q_{t+1}|q_t\right)\prod_{t=0}^{T-1}p\left(y_t|q_t\right), \tag{1}$$

where $q=\{q_0,\ldots,q_{T-1}\}$, $y=\{y_0,\ldots,y_{T-1}\}$, and $T=253$ (as defined in section 2.6.1). However, because each presentation of a stimulus uses the exact same audio waveform, the values of q are already known for each stimulus from the phonetic transcriptions of the stimuli. This simplifies the HMM for each stimulus because the values of the hidden states are known. In this scenario, Bayes' theorem can be used to express the conditional probability associated with each simplified HMM as

$$p\left(y|q\right)=\frac{p\left(q,y\right)}{p\left(q\right)}=\prod_{t=0}^{T-1}p\left(y_t|q_t\right). \tag{2}$$

For each stimulus presentation, our HMM-based classification scheme uses (2) to estimate $p\left(y|q\right)$ for each of the ten competing simplified HMMs (one per stimulus) and predicts the stimulus that yielded the largest $p\left(y|q\right)$ value. This can be formally expressed as

$$\hat{s}=\underset{s\in S}{\operatorname{argmax}}\prod_{t=0}^{T-1}p\left(y_t|q_{t,s}\right)=\underset{s\in S}{\operatorname{argmax}}\sum_{t=0}^{T-1}\log p\left(y_t|q_{t,s}\right), \tag{3}$$

where $\hat{s}$ is the predicted stimulus, S is the set of possible stimuli, and $q_{t,s}$ is the phoneme at time index t for stimulus s. We use log probabilities as expressed in the latter part of (3) for computational efficiency and numerical stability.

Each feature vector y_t contains the z-scored high gamma power values for each relevant channel at the following time indices: $t+\{0,2,\ldots,38,40\}$. This parameterization of the feature vectors resembles the high gamma windows described in previous research [10]. We used PCA-LDA modeling (as described in section 2.6.1) to obtain the $p\left(y_t|q_t\right)$

values at each time point. Model training, which typically took 10–20 seconds, was performed using all available data for a subject prior to starting each new task block (classifications were not performed in the first block).

2.7. Evaluation methods

We primarily used classification accuracy (the percent of classification attempts that resulted in correct classifications) to evaluate rtNSR. We computed classification accuracies for each task block and in a sliding window fashion across the blocks to measure how the accuracy changed over time.

Because duration was highly variable across sentences and could have been used by the classification schemes for improved sentence discrimination, we also assessed how varying T, the number of time points used during modeling of each stimulus presentation (described in section 2.6), affected classification accuracy. We performed offline testing with 21 different values for T that were (roughly) equally-spaced within the range of $[1, 253]$. For both classification schemes, 10-fold stratified cross-validation was used on all the available data for each subject.

To assess the speed of our real-time classification schemes, we measured the amount of time it took each classification scheme to perform classifications during offline simulations. For the direct classification scheme, we measured the amount of time required to make each sentence prediction from a concatenated neural feature vector, which was performed every $T = 253$ time points. For the HMM-based classification scheme, we measured the amount of time required to compute the phoneme likelihood values $p\left(y_t|q_t\right)$ at each time point and the amount of time required to perform a sentence classification using the associated phoneme likelihoods every $T = 253$ time points.

3. Results

For subject A, we achieved stimulus prediction accuracies of 90% with the direct classification scheme in real-time and 98% with the HMM-based classification scheme during offline simulation after training on 250 stimulus presentations (approximately 11 minutes of training data). For subject B, we achieved accuracies of 90% with the direct classification scheme during offline simulation and 91% with the HMM-based classification scheme in real-time after training on 150 stimulus presentations (approximately 6.5 minutes of training data). Confusion matrices for these results are provided in figure S2. All observed classification accuracies are depicted in figure 3. The real-time classification performance during the final task block for subject A with the direct classification scheme is demonstrated in supplementary video 1.

Figure 4 depicts the effect that varying the number of time points used during classification had on accuracy. When only the first 89 time points (approximately 0.9 seconds) for each trial were used, which is less than the number of time points associated with the shortest sentence, the classification accuracies plateaued at 90% or higher. These

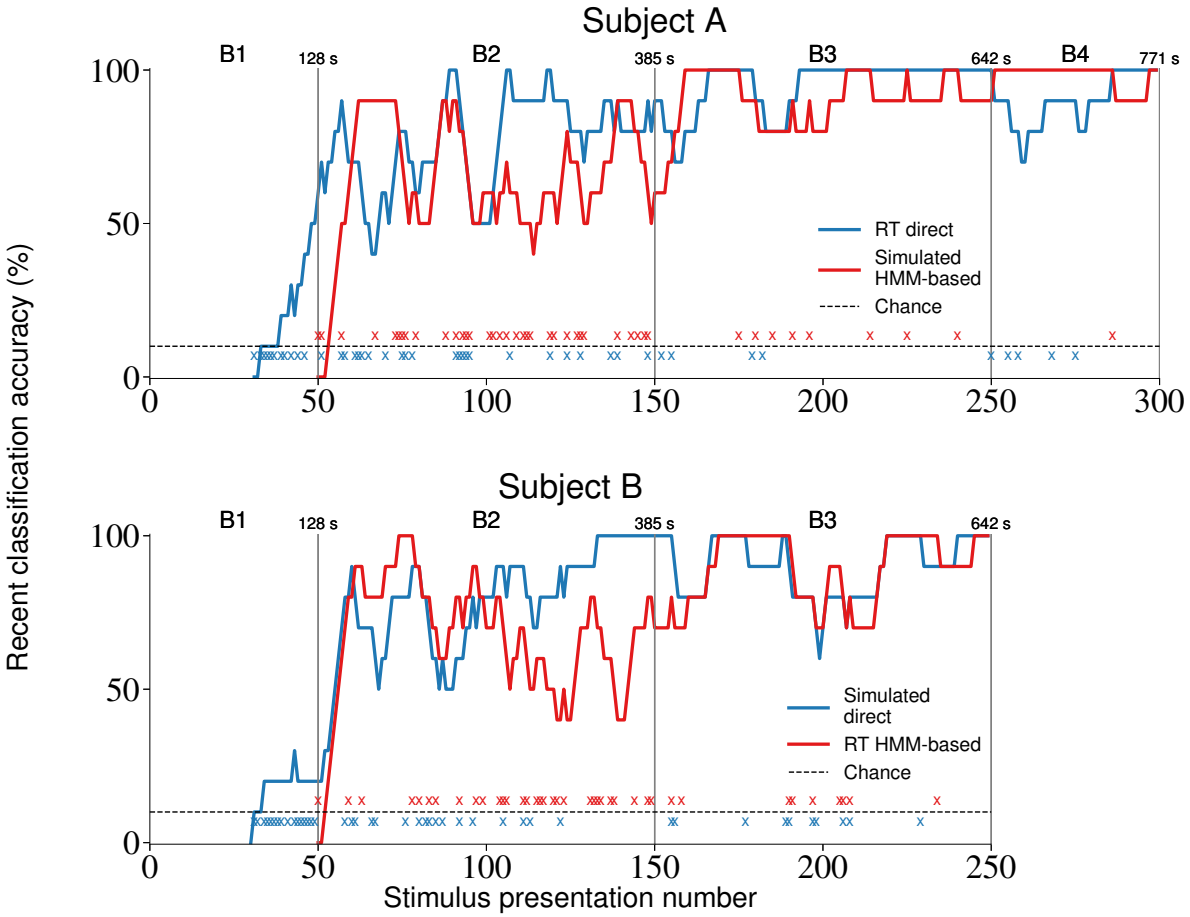

Figure 3: Stimulus classification accuracies for each subject, task block, and classification scheme. The colored curves depict, for each stimulus presentation, the percentage of the 10 most recent classification attempts (including the current attempt) that were correct. The blue and red curves represent testing with the direct and HMM-based classification schemes, respectively. Results obtained from real-time testing contain "RT" in the label and those obtained from offline simulations contain "Simulated" in the label. A colored x marker indicates a trial that was incorrectly classified with the associated classification scheme. The task blocks are labeled (with "B" followed by the block number) and separated by vertical lines. The total duration of recorded data at the end of each block is given above these vertical lines (rounded to the nearest second). Chance accuracy (10%) is depicted as a horizontal dashed line. These plots exhibit that rtNSR is able to achieve high real-time classification accuracies after short training intervals.

results indicate that the classification schemes are relying on more than just sentence length when performing classifications and that highly accurate classification can be performed using neural responses collected during perception of the first $2-3$ words of the sentences.

During offline simulation of the HMM-based classification scheme with subject A, computing the phoneme likelihoods at each time point took on average 2.64 ms ($\sigma = 0.61$ ms, $N = 12650$) and each sentence classification (using the pre-computed phoneme

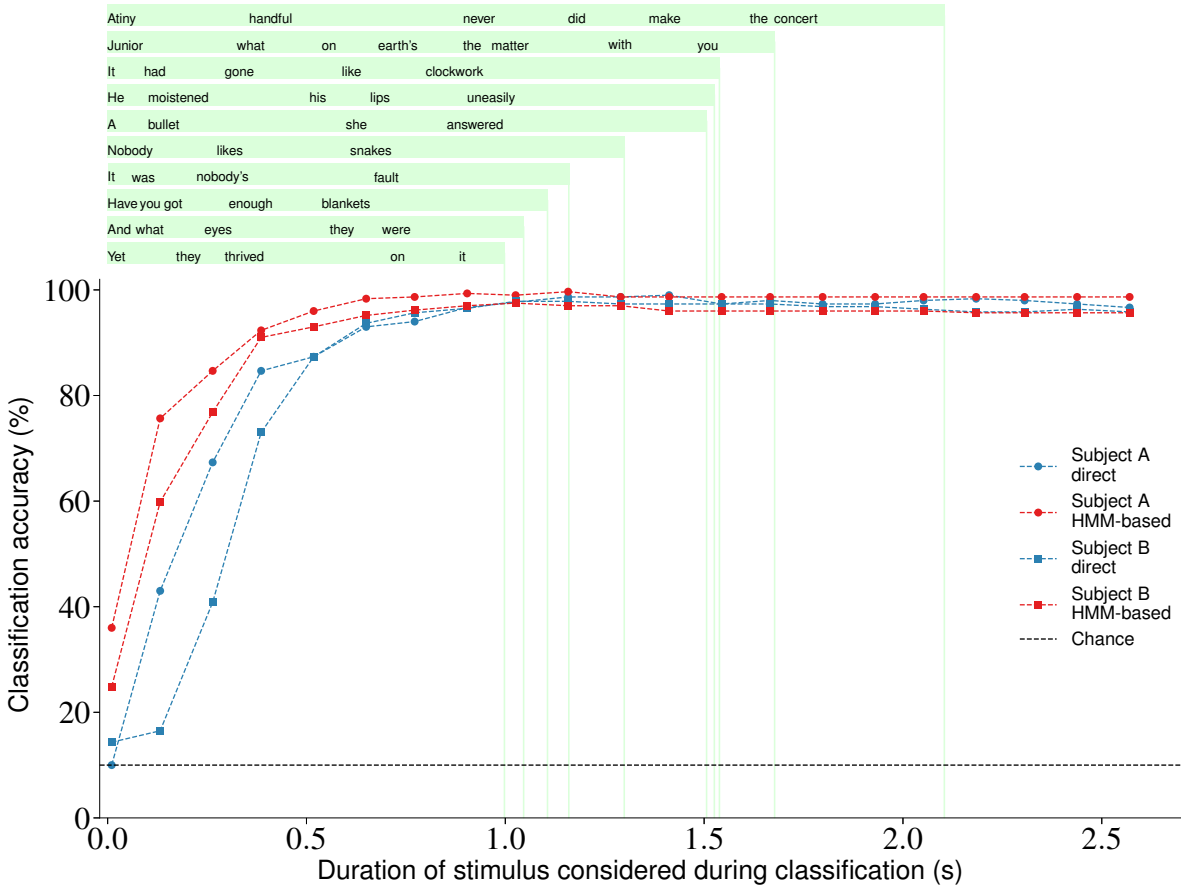

Figure 4: The effects of varying the duration of each stimulus presentation used during classification. For each subject and classification scheme, the corresponding colored dots or squares and dashed line depict the classification accuracies associated with the considered stimulus durations. The transcriptions for each sentence are shown above the plot. The left boundary of each word is aligned to the time at which that word begins within the sentence audio files. The light green rectangles and vertical lines indicate the full time span and offset time, respectively, for each sentence. Chance accuracy (10%) is depicted as a horizontal dashed line. These accuracy curves indicate that both classification schemes are able to leverage information in the neural signals during perception of the initial 0.75 seconds of each sentence to accurately discriminate between sentences.

likelihoods) took on average 0.07 ms ($\sigma = 0.01$ ms, $N = 50$). During offline simulation of the direct classification scheme with subject B, each sentence classification took on average 10.23 ms ($\sigma = 3.99$ ms, $N = 100$).

4. Discussion

In this work, we have introduced a real-time Neural Speech Recognition (rtNSR) software package and demonstrated its ability to perform real-time, single-trial stimulus classification using cortical responses evoked during speech perception. We achieved high

classification accuracies after short training intervals using both direct (sentence-level) and HMM-based (phoneme-level) classification schemes. The HMM-based classification scheme exhibited the highest observed accuracy in a single block (98% accuracy with subject A).

We showed that neural activity collected during perception of naturally spoken sentences could be used directly for classification without including acoustic, phonetic, or any other stimulus information (other than sentence identity) during modeling (with the direct classification scheme). We also showed that similar performance could be achieved with a more sophisticated classification approach that involved modeling the neural representations of phonemes (with the HMM-based classification scheme). Additionally, we demonstrated that the performance of our system did not rely on sentence length, a trivial stimulus feature, since peak classification accuracies were obtained using only a subset of time points associated with each trial that was smaller than the duration of the shortest sentence in the task. Finally, we showed that rtNSR was able to perform real-time classifications quickly; on average, the direct classification scheme only required 10 ms every 2.57 seconds (the stimulus time window duration) to perform a classification and the HMM-based classification scheme only required less than 3 ms every 10.16 ms (the sampling interval) to compute phoneme likelihoods at each time point and a negligible amount of time to make a sentence prediction from the phoneme likelihoods.

Our results serve as a proof-of-concept that rtNSR is capable of performing speech classification from neural signals in real-time. We built the rtNSR system to have a modular architecture in which individual components can be improved or replaced with task-specific and optimized implementations for future applications. For example, the high gamma power estimation algorithm implemented on the DAQ rig can be replaced with digital filters in rtNSR that directly approximate the high gamma analytic amplitude, a representation of high gamma activity that has been used in previous speech-related research [3, 7, 10]. Also, the sentence classification process can be replaced by a process implementing a more sophisticated classification model, such as a recurrent neural network classifier. In addition, the software's robust task design and execution capabilities make it amenable to a variety of task paradigms, including isolated word or continuous speech production or perception tasks, visual stimulus presentation tasks, and covert speech tasks. Through augmentation of the system's data acquisition and feature extraction functionality, it can also be deployed in applications involving alternative types of neural signal acquisition, such as via electroencephalography or microelectrode arrays.

For an initial evaluation of our system, we used a relatively simple sentence classification task with only 10 unique stimuli. Although the observed classification accuracies were very high in this example task, demonstrating our ability to learn the relationship between auditory speech stimulus features and neural activity recorded with ECoG in real-time, further testing is needed to determine how well the classification schemes scale as the number of stimuli increases. We expect the HMM-based classification scheme to scale more favorably than the direct classification scheme because it can take

advantage of shared phonemic content across the stimuli and can predict stimuli that were not presented during training. However, it is also possible that an increase in the variety of coarticulation contexts and other sources of variability in the stimuli will negatively affect accuracy if they are not explicitly considered during modeling.

We established that one trivial stimulus feature (duration) did not drive classification performance, but there are other potential features that may have impacted accuracy. In this task, nine speakers produced the ten stimuli, resulting in a large degree of variability in the speaker-dependent acoustic properties of the stimuli that may have been leveraged by the classification schemes. When analyzing the sentence confusions observed during classification (see figure S2), we did not find evidence that speaker identity was driving our classifiers in this task. However, it is possible that in experiments involving a larger set of sentences from relatively few speakers the direct classification scheme would be more susceptible to relying on speaker identity than the HMM-based classification scheme, since the latter uses phoneme models that do not incorporate stimulus identity information while being trained to discriminate between phonemes. Future work using a wider variety of stimuli with multiple speech samples produced by each speaker could address the effects of this type of information on classification performance.

In future work, we plan to expand the HMM-based classification scheme into a real-time continuous speech decoder that uses language modeling and Viterbi decoding (similar to a real-time version of the system described in [10]). The performance achieved in this work using phoneme modeling with naturally spoken sentences (as opposed to isolated words or syllables) is a promising proof-of-concept for potential continuous decoding applications. Unlike our task, a real-time continuous decoding application should not rely on explicit stimulus timing, although precise transcriptions of the stimuli would still be required for model training. The methods described in this work could also be applied to real-time experimental paradigms in overt and covert speech production tasks guided by existing offline speech decoding research efforts [22–28].

After further development of rtNSR, our goal is to deploy the system as part of a speech prosthesis that restores communicative capabilities to individuals diagnosed with locked-in syndrome or other impairments. Locked-in patients typically have little to no voluntary muscle control but retain cognition and awareness [29–33]. Although methods exist to provide basic communicative capabilities to locked-in patients [33–36] and are associated with increases in patient-reported quality of life [31, 32], these approaches often involve tedious and difficult to learn procedures such as selecting characters one at a time at rates less than 10 characters per minute (typing rates are typically more than 175 characters per minute in healthy individuals). Development of a device capable of directly interpreting intended speech from neural activity could result in significant improvements to the speed and naturalness of assistive speech technology and, as a result, the quality of life for impaired patients. Existing brain-computer interface (BCI) research has shown that ECoG signals can be successfully used in real-time motor control applications [37,38], and the classification accuracies observed in this task using ECoG are similar to or higher than those exhibited in these approaches (although direct performance comparisons may

not be possible due to fundamental differences in task designs and constraints). Our system's modular real-time framework allows for incorporation of feedback and subject adaptation, important components in closed-loop BCIs that will most likely be beneficial in future speech prostheses. Given the performance exhibited by rtNSR in this work and its capacity for expansion, we are confident in its ability to serve as a platform for the design and implementation of the proposed speech prosthetic device.

Acknowledgments

All authors thank the various members of EFC's lab for help during data recording and the patients who volunteered to be subjects in this work.

This work was supported by the National Institutes of Health National Research Service Award F32-DC013486 and Grants R00-NS065120, DP2-OD00862, and R01-DC012379, the Ester A. and Joseph Klingenstein Foundation, and the National Science Foundation Grant No. 1144247. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health.

Author contributions

DAM developed and tested the rtNSR system, performed all data collection and analyses, and wrote the manuscript. MKL provided project guidance. EFC led the research project. All authors edited the manuscript.

Conflict of interest

The authors declare no conflicts of interest.

References

[1] Dana F. Boatman, Charles B. Hall, Moise H. Goldstein, Ronald P. Lesser, and Barry J. Gordon. Neuroperceptual differences in consonant and vowel discrimination: as revealed by direct cortical electrical interference. *Cortex*, 33:83–98, mar 1997.

[2] Jeffrey R Binder, Julie Anne Frost Bellgowan, Thomas A Hammeke, Patrick Bellgowan, Jane Springer, and Jacqueline N Kaufman. Human temporal lobe activation by speech and nonspeech sounds. *Cerebral Cortex*, 10(5):512–528, 2000.

[3] Ryan T Canolty, Maryam Soltani, Sarang S Dalal, Erik Edwards, Nina F Dronkers, Srikantan S Nagarajan, Heidi E Kirsch, Nicholas M Barbaro, and Robert T Knight. Spatiotemporal dynamics of word processing in the human brain. *Frontiers in Neuroscience*, 1(1):185–196, 2007.

[4] Gregory Hickok and David Poeppel. The cortical organization of speech processing. *Nature Reviews Neuroscience*, 8(May):393–402, 2007.

[5] Josef P Rauschecker and Sophie K Scott. Maps and streams in the auditory cortex: nonhuman primates illuminate human speech processing. *Nature neuroscience*, 12(6):718–724, jun 2009.

[6] Nima Mesgarani, Connie Cheung, Keith Johnson, and EF Edward F Chang. Phonetic feature encoding in human superior temporal gyrus. *Science*, 343(6174):1006–1010, feb 2014.

[7] Brian N Pasley, Stephen V David, Nima Mesgarani, Adeen Flinker, Shihab A Shamma, Nathan E Crone, Robert T Knight, and Edward F Chang. Reconstructing speech from human auditory cortex. *PLoS biology*, 10(1):e1001251, jan 2012.

[8] Minda Yang, Sameer A Sheth, Catherine A Schevon, Guy M Mckhann Ii, Nima Mesgarani, Guy M Mckhann Ii, and Nima Mesgarani. Speech reconstruction from human auditory cortex with deep neural networks. In *Sixteenth Annual Conference of the International Speech Communication Association*, pages 1121–1125, 2015.

[9] Matthew K. Leonard, Maxime O. Baud, Matthias J. Sjerps, and Edward F. Chang. Perceptual restoration of masked speech in human cortex. *Nature Communications*, 7:13619, 2016.

[10] David A Moses, Nima Mesgarani, Matthew K Leonard, and Edward F Chang. Neural speech recognition: continuous phoneme decoding using spatiotemporal representations of human cortical activity. *Journal of Neural Engineering*, 13(5):056004, 2016.

[11] Connie Cheung and Edward F Chang. Real-time, time-frequency mapping of event-related cortical activation. *Journal of neural engineering*, 9(4):046018, aug 2012.

[12] Bahar Khalighinejad, Tasha Nagamine, Ashesh Mehta, and Nima Mesgarani. NAPLib: An open source toolbox for real-time and offline Neural Acoustic Processing. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, pages 846–850, 2017.

[13] Eric C Leuthardt, Charles Gaona, Mohit Sharma, Nicholas Szrama, Jarod Roland, Zac Freudenberg, Jamie Solis, Jonathan Breshears, and Gerwin Schalk. Using the electrocorticographic speech network to control a braincomputer interface in humans. *Journal of Neural Engineering*, 8(3):036004, 2011.

[14] Vasileios G. Kanas, Iosif Mporas, Heather L. Benz, Kyriakos N. Sgarbas, Anastasios Bezerianos, and Nathan E. Crone. Real-time voice activity detection for ECoG-based speech brain machine interfaces. *International Conference on Digital Signal Processing, DSP*, 2014-January(August):862–865, 2014.

[15] Liberty S. Hamilton, David L. Chang, Morgan B. Lee, and Edward F. Chang. Semi-automated Anatomical Labeling and Inter-subject Warping of High-Density Intracranial Recording Electrodes in Electrocorticography. *Frontiers in Neuroinformatics*, 11(October):62, 2017.

[16] John Garofolo, Lori Lamel, William Fisher, Jonathan Fiscus, David Pallett, Nancy Dahlgren, and Victor Zue. TIMIT Acoustic-Phonetic Continuous Speech Corpus LDC93S1. *Linguistic Data Consortium*, page 1, 1993.

[17] Nathan E Crone, Diana L Miglioretti, Barry Gordon, and Ronald P Lesser. Functional mapping of human sensorimotor cortex with electrocorticographic spectral analysis. II. Event-related synchronization in the gamma band. *Brain*, 121:2301–2315, 1998.

[18] Python Software Foundation. Python Language Reference, 2010.

[19] M. Lam. Software pipelining: An effective scheduling technique for VLIW machines. *ACM Sigplan Notices*, 23(7):318–328, 1988.

[20] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2012.

[21] Olivier Ledoit and Michael Wolf. Honey, I Shrunk the Sample Covariance Matrix. *The Journal of Portfolio Management*, 30(4):110–119, 2004.

[22] Spencer Kellis, Kai Miller, Kyle Thomson, Richard Brown, Paul House, and Bradley Greger. Decoding spoken words using local field potentials recorded from the cortical surface. *Journal of neural engineering*, 7(5):056007, 2010.

[23] Xiaomei Pei, Dennis L Barbour, Eric C Leuthardt, and Gerwin Schalk. Decoding vowels and consonants in spoken and imagined words using electrocorticographic signals in humans. *Journal of Neural Engineering*, 8(4):046028, 2011.

[24] B. Denby, T. Schultz, K. Honda, T. Hueber, J. M. Gilbert, and J. S. Brumberg. Silent speech

interfaces. *Speech Communication*, 52(4):270–287, 2010.

[25] Stéphanie Martin, Peter Brunner, Chris Holdgraf, Hans-Jochen Heinze, Nathan E Crone, Jochem Rieger, Gerwin Schalk, Robert T Knight, and Brian N Pasley. Decoding spectrotemporal features of overt and covert speech from the human cortex. *Frontiers in neuroengineering*, 7(May):14, 2014.

[26] Emily M Mugler, James L Patton, Robert D Flint, Zachary a Wright, Stephan U Schuele, Joshua Rosenow, Jerry J Shih, Dean J Krusienski, and Marc W Slutzky. Direct classification of all American English phonemes using signals from functional speech motor cortex. *Journal of neural engineering*, 11(3):035015, 2014.

[27] Christian Herff, Dominic Heger, Adriana de Pesters, Dominic Telaar, Peter Brunner, Gerwin Schalk, and Tanja Schultz. Brain-to-text: decoding spoken phrases from phone representations in the brain. *Frontiers in Neuroscience*, 9(June):1–11, 2015.

[28] Stéphanie Martin, Peter Brunner, Iñaki Iturrate, José del R. Millán, Gerwin Schalk, Robert T. Knight, and Brian N. Pasley. Word pair classification during imagined speech using direct brain recordings. *Scientific Reports*, 6(1):25803, 2016.

[29] American Congress of Rehabilitation Medicine. Recommendations for use of uniform nomenclature pertinent to patients with severe alterations in consciousness. *Archives of Physical Medicine and Rehabilitation*, 76(2):205–209, 1995.

[30] Steven Laureys, Frédéric Pellas, Philippe Van Eeckhout, Sofiane Ghorbel, Caroline Schnakers, Fabien Perrin, Jacques Berré, Marie Elisabeth Faymonville, Karl Heinz Pantke, Francois Damas, Maurice Lamy, Gustave Moonen, and Serge Goldman. The locked-in syndrome: What is it like to be conscious but paralyzed and voiceless? *Progress in Brain Research*, 150(5):495–511, jan 2005.

[31] Marie-Aurélie Bruno, Jan L Bernheim, Didier Ledoux, Frédéric Pellas, Athena Demertzi, and Steven Laureys. A survey on self-assessed well-being in a cohort of chronic locked-in syndrome patients: happy majority, miserable minority. *BMJ open*, 1(1):e000039, 2011.

[32] Marie-Christine Rousseau, Karine Baumstarck, Marine Alessandrini, Véronique Blandin, Thierry Billette de Villemeur, and Pascal Auquier. Quality of life in patients with locked-in syndrome: Evolution over a 6-year period. *Orphanet journal of rare diseases*, 10:88, 2015.

[33] Mariska J. Vansteensel, Elmar G.M. Pels, Martin G. Bleichner, Mariana P. Branco, Timothy Denison, Zachary V. Freudenburg, Peter Gosselaar, Sacha Leinders, Thomas H. Ottens, Max A. Van Den Boom, Peter C. Van Rijen, Erik J. Aarnoutse, and Nick F. Ramsey. Fully Implanted BrainComputer Interface in a Locked-In Patient with ALS. *New England Journal of Medicine*, 375(21):NEJMoa1608085, 2016.

[34] Martin Spüler, Wolfgang Rosenstiel, and Martin Bogdan. Online Adaptation of a c-VEP Brain-Computer Interface(BCI) Based on Error-Related Potentials and Unsupervised Learning. *PLoS ONE*, 7(12), 2012.

[35] Eric W Sellers, David B Ryan, and Christopher K Hauser. Noninvasive brain-computer interface enables communication after brainstem stroke. *Science translational medicine*, 6(257):257re7, oct 2014.

[36] B O Mainsah, L M Collins, K a Colwell, E W Sellers, D B Ryan, K Caves, and C S Throckmorton. Increasing BCI communication rates with dynamic stopping towards more practical use: An ALS study. *Journal of Neural Engineering*, 12(1):016013, 2015.

[37] Eric C. Leuthardt, Kai J. Miller, Gerwin Schalk, Rajesh P.N. Rao, and Jeffrey G. Ojemann. Electrocorticography-based brain computer interface - The seattle experience. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 14(2):194–198, 2006.

[38] Guy Hotson, David P McMullen, Matthew S Fifer, Matthew S Johannes, Kapil D Katyal, Matthew P Para, Robert Armiger, William S Anderson, Nitish V Thakor, Brock A Wester, and Nathan E Crone. Individual finger control of a modular prosthetic limb using high-density electrocorticography in a human subject. *Journal of Neural Engineering*, 13(2):026017, 2016.

Supplementary data

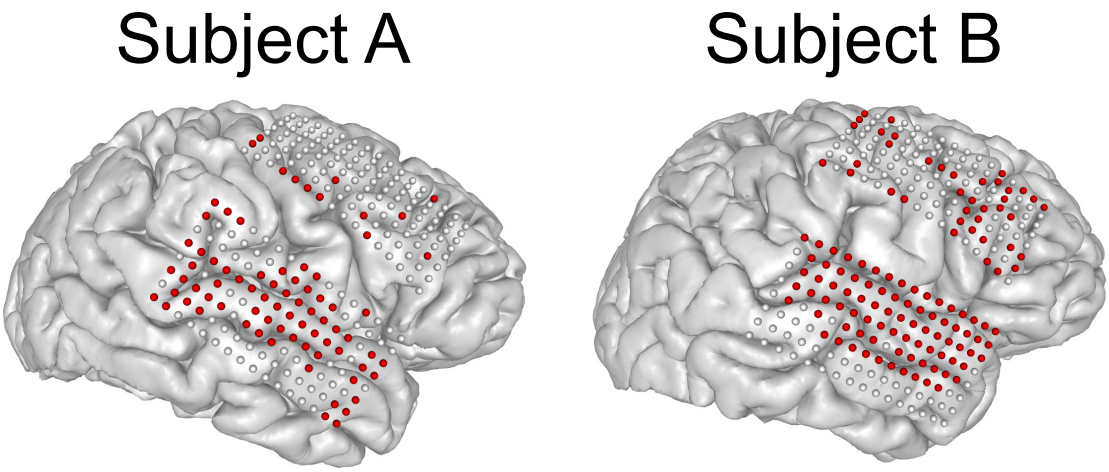

Figure S1: 3-D MRI brain reconstruction, electrode coverage (white and red dots), and relevant electrodes (red dots) for each subject. The depicted relevant electrodes were determined using the data acquired prior to the final testing block for each subject.

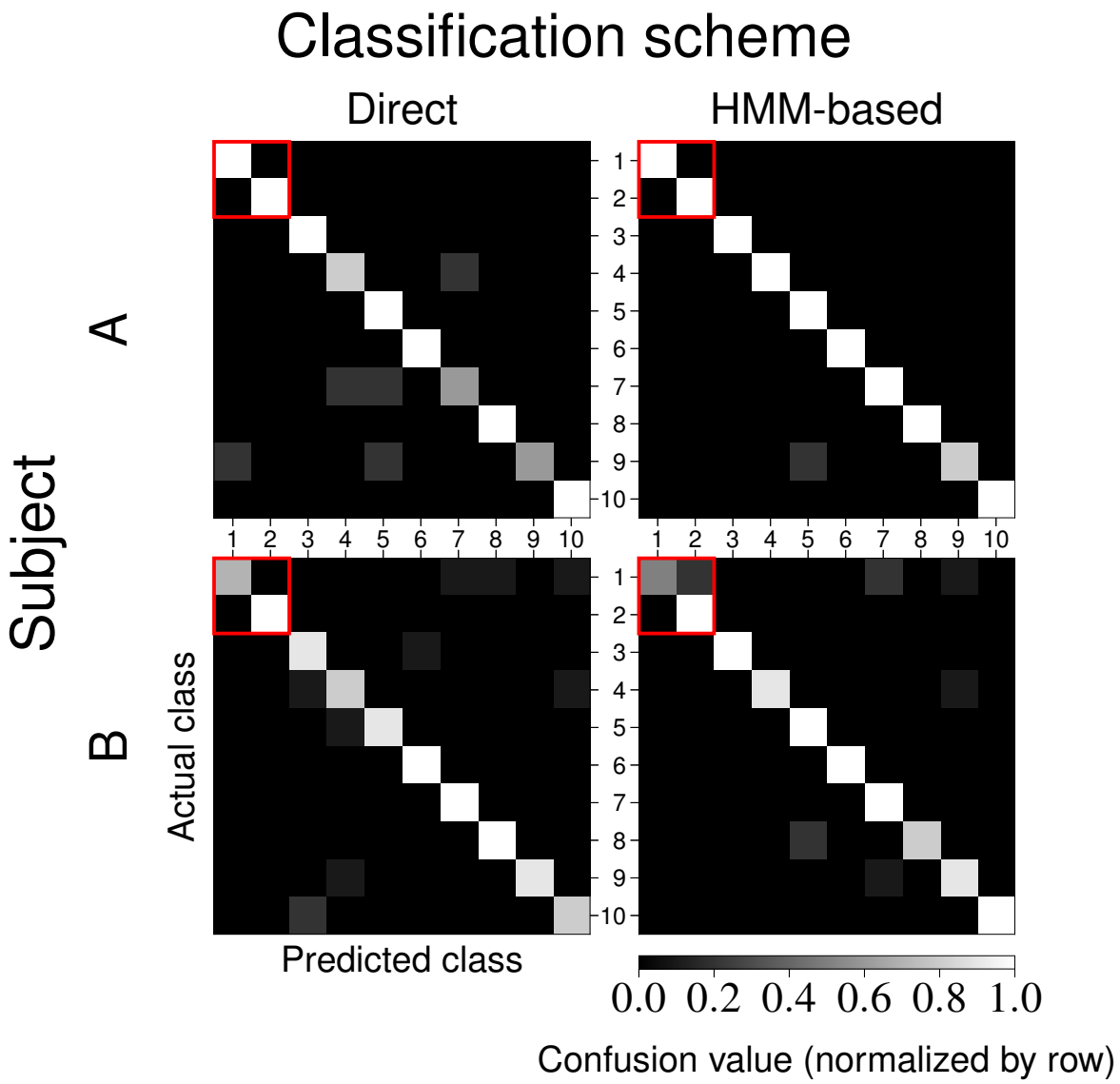

Figure S2: Confusion matrices computed using the final task block for each subject and classification scheme. Each row depicts results for a single subject (A or B) and each column depicts results for a single classification scheme (direct or HMM-based). The class numbers 1-10 correspond to the stimulus order given in table 1. The color-value mapping is identical across all confusion matrices and uses row-normalized confusion values. The red outline signifies the two stimuli (classes 1 and 2) that were produced by the same speaker (all other stimuli were produced by unique speakers). In general, these two stimuli were not confused with each other, suggesting that the classifiers were not relying on speaker identity to make predictions. During HMM-based classification with subject B, class 1 was confused with class 2 20% of the time, but it was also confused with classes 7 and 9 20% and 10% of the time, respectively).

Supplementary video 1

This video was recorded during the final task block with subject A and portrays the real-time capabilities of the rtNSR software package. This subject gave written consent to allow us to publish this video. The subject's face has been blurred for anonymity. In each trial, the subject listened to one of the ten sentences played through headphones. The direct classification model, which had been previously trained on 250 stimulus presentations (a little under 11 minutes of neural data) and was retrained in real-time as data was being collected, predicted which sentence the subject just heard using the neural signals collected in real-time from the implanted ECoG array. The predicted sentence is displayed on a monitor for a few seconds until the start of the next trial. Although the software monitored whether or not each sentence was correctly classified, for demonstrative purposes the subject was instructed to respond with a "thumbs up" if the classified sentence matches what was heard through the headphones and a "thumbs down" otherwise. Information about the performance of the system was displayed in a running accuracy plot and in text at the bottom of the monitor (to protect patient anonymity, the video was further blurred to obscure a reflection of the subject's face in the monitor). Classification accuracy associated with the trials in the video is plotted as the blue "RT direct" curve for subject A during the B4 task block in figure 3. The colors of the classified sentences were arbitrarily chosen and do not reflect any information about the classifications. We confirmed offline that the subject's "thumbs up" or "thumbs down" feedback was accurate (near the 2:24 time mark, the subject originally responds incorrectly with a "thumbs up" but quickly corrects the response to a "thumbs down").