Speech Decoding as Machine Translation

Joseph G. Makin, David A. Moses, and Edward F. Chang

Abstract We aimed to improve the state of the art in decoding speech from neural activity, with the ultimate goal of developing a useful brain-machine interface (BMI) for individuals who have lost the ability to speak—from ALS, a stroke, or other traumatic brain injury. In our recent study (Makin et al. in Nat Neurosci 23:575–582, 2020), each of four participants undergoing clinical monitoring for epilepsy read aloud, making repeated passes through a set of some 30–50 sentences, while her electrocorticogram was simultaneously recorded. Our algorithm, which was inspired by recent ideas in machine translation, brought word error rates down from the previous state of the art, about 60, to 3%. In this chapter, we discuss those results, their limitations, and their implications for the general problem of speech decoding.

Keywords Brain-machine interface · ECoG · Speech decoding · Encoder-decoder networks

1 Introduction

The field of speech decoding began in 2009 with the successful synthesis of vowel formants from the firing rates of a small number of neurons, recorded with a microelectrode implanted into speech-motor cortex of a locked-in patient [3]. Isolated phonemes and monosyllables have subsequently been classified, with moderate accu-

J. G. Makin is now with the School of Electrical and Computer Engineering at Purdue University. For questions about the algorithm/code, contact him at `jgmakin@purdue.edu`. For questions about experiment/data, contact EFC at `edward.chang@ucsf.edu`.

J. G. Makin (✉) · D. A. Moses · E. F. Chang
Center for Integrative Neuroscience, UCSF, San Francisco, CA, USA
e-mail: jgmakin@purdue.edu

E. F. Chang
e-mail: edward.chang@ucsf.edu

Department of Neurological Surgery, UCSF, San Francisco, CA, USA

C. Guger et al. (eds.), *Brain-Computer Interface Research*,
SpringerBriefs in Electrical and Computer Engineering,
https://doi.org/10.1007/978-3-030-79287-9_3

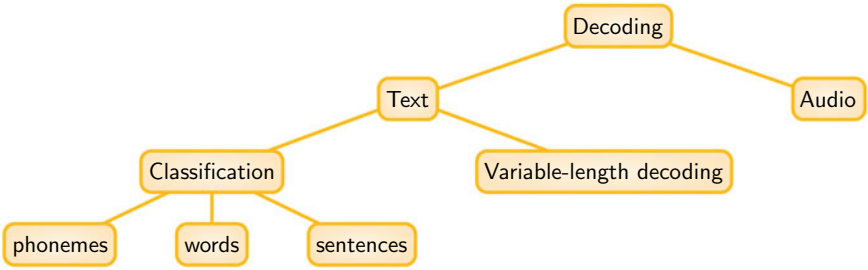

Fig. 1 Strategies for speech decoding

racies, from recordings made by penetrating electrodes [18] or from the electrocorticogram (ECoG) [3, 4, 14, 16]. Even setting aside their modest accuracies, these results are of interest mostly as proof of concept, because the phonemes produced in continuous speech are highly influenced by their neighbors ("coarticulation"); and (taking the other horn of the dilemma) a speech BMI that required its users to produce phonemes in isolation would forego the principal merits of decoding speech rather than handwriting or typing: speed and naturalness.

Several studies have attempted to decode continuously spoken speech [1, 2, 9, 11–13]. These can be divided on the basis of what modality they attempt to decode: audio ("speech synthesis") or text (see Fig. 1). Neural speech synthesis has markedly improved since the foundational work of Brumberg and colleagues [3], producing nearly intelligible output. In arguably the most successful of these studies [2], volunteers were subsequently recruited to transcribe the speech that had been synthesized from patients' ECoG data, in order to quantify the results. When limited to a vocabulary of just 50 words, transcribers achieved word error rates[1] (WERs) of about 50%. (On the other hand, the primary advantage of speech synthesis is that it is not, in principle, limited to a fixed vocabulary.)

When the aim is, alternatively, to output text, there remains the question of granularity. At one extreme, phonemes can be classified, and subsequently assembled into words and sentences with a language model. Operating with ECoG and a vocabulary of 100 words, such an approach has yielded word error rates of about 60% [9]. At the opposite extreme, Moses and colleagues classified entire sentences from their corresponding ECoG signatures [13], trading coverage of English for distinguishability of the tokens to be decoded. This model achieved WERs of 33% on a set of 50 sentences [11].

An alternative to classifying (and subsequently assembling into larger units) phonemes, words, or sentences is to decode variable-length sequences of words. That is, the decoder consumes a long sequence of neural data, contemporaneous

[1] Errors are computed as the *minimum* number of word insertions, deletions, and substitutions required to transform the predicted into the true word sequence. Dividing by the number of words in the true sequence yields a word error *rate*. Intuitively, any sensible decoder should achieve error rates between 0 and 1.0, since the WER for a "decoder" that just predicts an empty sequence for every "input" is precisely 1.0. But in practice poor decoders can make errors at rates greater than 1.

with (for example) a single spoken sentence, and then begins emitting words, one at a time, until it decides to stop. The potential advantage of such an approach is that it does not impose assumptions about which parts of the ECoG signal correspond to which words or word parts. This allows for: phoneme classification and assembly into words to be solved jointly rather than sequentially; automatic handling of coarticulation; the assimilation of temporally dispersed (e.g., semantic) information; dispensing with a phoneme transcription, which would in any case be difficult to obtain from non-speaking persons; and the production of any sentence composed from the fixed vocabulary of words. The entire pipeline can be implemented as an artificial neural network, and trained end-to-end, from neural data to sentences. And indeed, such "encoder-decoder" networks have in the last five years become the standard for machine translation, where the input is a variable-length sequence, not of neural data, but of words in another language [7, 8, 19, 21].

Using ECoG as input to an encoder-decoder neural network, we achieved WERs as low as 3%, operating with a vocabulary of about 250 words and a set of 50 unique sentences [11]. Below we reprise those results and discuss their limitations.

2 Methods

We briefly describe the fundamental aspects of the study's methods. More details can be found in the original publication [11].

Participants. Drug-resistant epilepsy can sometimes be treated with brain surgery, in which case seizures are first localized with a neurological recording device. One common procedure is to perform a craniotomy and then place a grid of electrodes on the surface of the brain and monitor the electrocorticogram over the course of (typically) one or two weeks. During this period, patients are not anaesthetized and are able (*inter alia*) to read aloud without difficulty. The participants in the study reviewed here were epilepsy patients at the UCSF Medical Center. Prior to surgery, all participants (four female; all right-handed and left-hemisphere language-dominant; aged 47 [participant **a**], 31 [participant **b**], 29 [participant **c**], and 49 [participant **d**] years) gave written consent to take part in the study, which was carried out according to protocol approved by the UCSF Committee on Human Research.

Data. The electrocorticogram was recorded with high-density (4-mm pitch) arrays from the peri-Sylvian cortices of participants while they read aloud from one of two sets of sentences (see below). The ECoG on all channels and the microphone signal were then pre-processed offline according to the pipelines in Fig. 2a, b, respectively. Finally, the spoken sentences were transcribed. Participants occasionally misread or otherwise misproduced the prompts, so the transcriptions did not always precisely match them. However, the rare (less than one percent of the total) productions that did not correspond to any word in the relevant sentence set (see below) were all transcribed as a single out-of-vocabulary token.

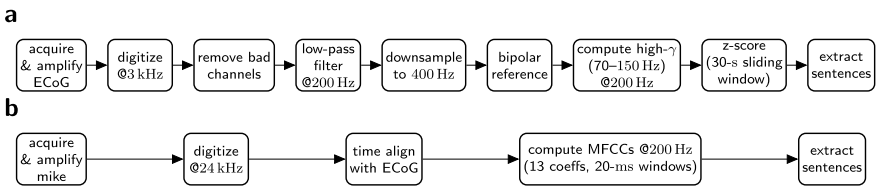

Fig. 2 Data preprocessing

Altogether, these provided a set of data "triplets," each of which consisted of

1. a matrix of ECoG data, with size (number of channels × number of samples);
2. a matrix of audio data (13 MFCCs × number of samples); and
3. the sequence of words in the corresponding sentence.

It was this set of triplets that was used to train and test the encoder-decoder neural network (see below). Note that the number of samples varied across triplets, being determined by how long it took the participant to speak the sentence. The number of channels varied across participants, depending on the number of electrodes implanted (256 [participants **a, b, d**] or 128 [**c**]) or too noisy to be used.

Two sets of sentences were used:

- MOCHA-TIMIT [22]: 460 sentences, ∼ 1800 unique words, participants **a**, **b**, **d**;
- picture descriptions: 30 sentences, ∼ 125 unique words, participants **c**, **d**.

For both sets, each sentence was presented briefly on a computer screen for recital, followed by a few seconds of rest (blank display). However, to avoid fatiguing participants, no more than 50 sentences were presented in a single session or "block." Thus, MOCHA-TIMIT could not be administered in a single block. To achieve consistency across participants, then, it was first divided into nine subsets, MOCHA-1, MOCHA-2, etc., of 50 sentences apiece (and 60 in MOCHA-9), each of which could be completed within one block, and within which sentence presentations were randomized. This resulted in better coverage of

- MOCHA-1: 50 sentences, ∼ 250 unique words, participants **a**, **b**, **d**

than the other subsets, and consequently we focus on decoding from it and the 30 picture descriptions in the main results below.

The encoder-decoder network. The architecture was inspired by recent artificial neural networks for machine translation [19], albeit with significant modifications. Abstractly, the encoder module first "consumes" an entire sequence of ECoG data (corresponding to one sentence), which it summarizes in a high-dimensional vector of fixed length, i.e. independent of the number of samples in the input sequence. The encoder then passes this summary to the decoder module, which unpacks it one word at a time.

We now describe our architecture in more detail, following Fig. 3 throughout:

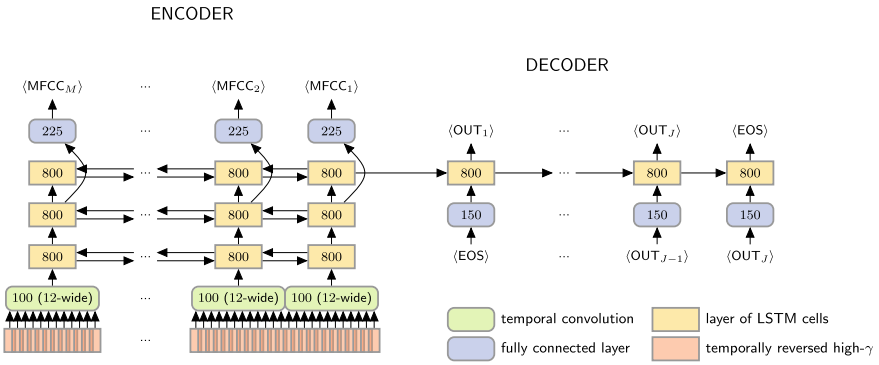

Fig. 3 Network architecture. The network consists of a pair of coupled RNNs, shown here "unrolled" in sequence steps. The encoder consumes ECoG and predicts audio (MFCCs); the decoder is initialized at the final encoder state and predicts words until it emits an end-of-sequence token (⟨EOS⟩)

1. *Temporal convolution*: The ECoG "signature" of a particular phoneme or word will not depend on the absolute time at which it was produced: it is time-invariant. Neural networks can efficiently exploit this invariance if they are required to apply the *same* filters at regular temporal intervals ("strides") along the entire length of the input sequence ("temporal convolution"). Furthermore, our filters "strode" by 12 samples, thereby downsampling their inputs from 200 to about 16 Hz. This helps because recurrent neural networks struggle with long sequences [6]. We used 100 filters (each spanning all of the input channels), so the output of the temporal convolutions is a sequence of length-100 vectors.
2. *Encoder recurrent neural network* (RNN): The vectors in this sequence are consumed one at a time (a) in forward order and (b) in backward order by a pair of RNNs, each with 400 units of long short-term memory (LSTM). Then, at every time point, the hidden states of these RNNs are concatenated together, creating a sequence of length-800 vectors. These are the input to a *second* pair of LSTM-based RNNs, which in turn produces inputs for a third pair.
   The hidden states of the second-layer RNNs are also used to predict the sequence of MFCCs, i.e. the speech audio. The hidden state of the deepest (third) RNNs *at the time step of the final sequence element* is interpreted to be a high-dimensional, length-independent summary of the entire input sequence, and is passed to the decoder RNN.
3. *Decoder RNN*: A single-layer, unidirectional, 800-unit, LSTM RNN is initialized at this high-dimensional summary. At each time step it *emits* a probability distribution over all the words in the vocabulary; and *consumes* either the previous word in the sequence (during training) or the previous *most probable word* (during testing). Notice that words, which are encoded as one-hot vectors, are first "embedded" into a dense, 150-dimensional space before entering the decoder RNN.

We make a few technical notes:

- **Time reversal**. Presumably, the first elements of the ECoG data are most related to the first word or words of the sentence. To reduce the number of computation steps separating these, the sequences of ECoG data were temporally reversed before entering the network, following Sutskever and colleagues [19].
- **MFCC sequences**. The sequences of MFCCs therefore also need to be temporally reversed. But they also need to be downsampled, to match the downsampling effect of the strided temporal convolution (see above). We simply decimated the sequences, selecting every twelfth vector (without bothering first to low-pass filter). The purpose of targetting speech audio is simply to guide training onto the right track [5, 20]; during testing, the predicted MFCCs are not used.
- **Training and testing**. The entire network was trained to map ECoG to audio (MFCCs) and text (word sequences) with stochastic gradient descent via back-propagation (with AdaM optimization [10]). Dropout [17] was applied to all layers except the recurrent connections. The remaining details of the training and testing procedure, including hyperparameter optimization, cross-validation, and transfer learning can be found in the original report [11].

3 Results

Decoder performance. Encoder-decoder performance on data from all four participants is shown in the first "violin" of each of the violin plots in Fig. 4. Subfigure labels correspond to participant IDs. Note that participants **a** and **b** read the 50 sentences from MOCHA-1, whereas participants **c** and **d** read the 30 picture descriptions (see **Methods**). The most impressive results are for participant **b**, for whom the encoder-decoder usually achieved WERs close to 0—perfect decoding. Only for participant **a**, who provided only two repeats of each sentence, were WERs outside the acceptable range of speech transcription (25% [15]). For two participants WERs were close to or below 5%, the performance of professional transcribers for spoken speech [23]—albeit with much larger vocabularies.

To understand better the high performance of the encoder-decoder, we trained new sets of networks with certain critical aspects of the architecture or the data removed (Fig. 4):

- **Grid density** ("low density"): Data from a lower-density grid can be simulated by dropping every other electrode from the data. This pseudo-grid will have 8-mm (rather than 4-mm) inter-electrode spacing and one quarter the number of electrodes. Moving to such a grid typically increases (median) WER by about 20 percentage points.
- **Speech audio** ("no MFCCs"): It will be impossible to acquire speech audio from non-speaking subjects—the ultimate target population for a speech prothesis. Training a network without requiring the encoder to predict MFCCs typically increases WER by 15–30 percentage points.

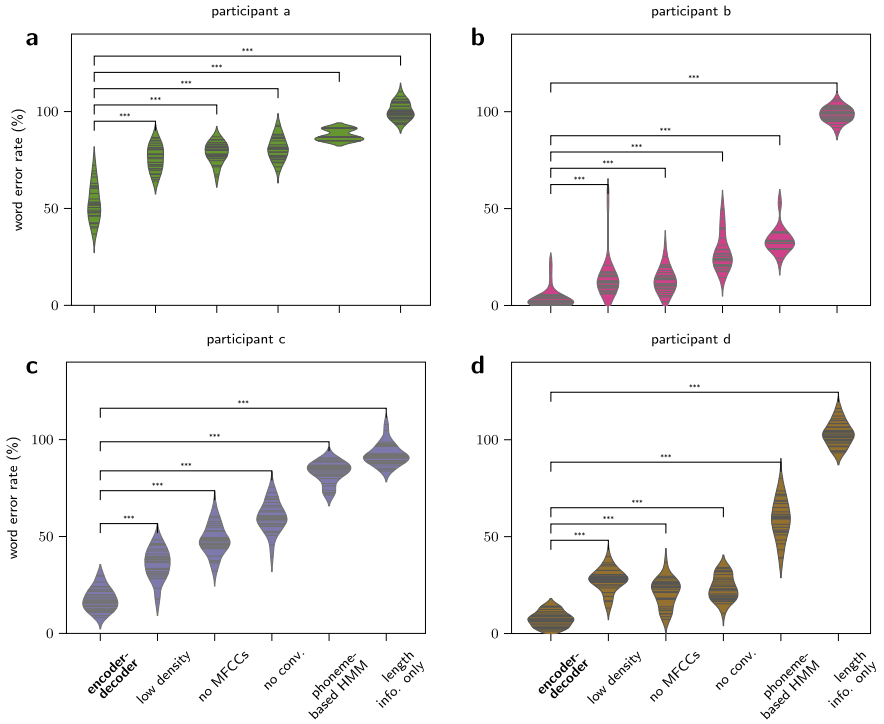

Fig. 4 Decoder performance. For all four participants (**a**–**d**) and for encoder-decoders as well as various "competitors," text output was evaluated in terms of word error rate (WER), with the sentences actually spoken serving as ground truth. For any single participant and decoder type, the distribution of WERs is across 30 instances of that decoder trained de novo and evaluated on randomly selected held-out blocks. For every participant, the distribution of WERs under the encoder-decoder is significantly better than any competitor ($p < 0.0005$ under a one-sided Wilcoxon signed-rank test, Holm-Bonferroni corrected for five comparisons). In addition to its standard implementation, the encoder-decoder was evaluated under a simulated lower-density grid ("low density"), without audio data during training ("no MFCCs"), without temporal convolution ("no conv."), and with input sequences of pure noise but of the correct length ("length info. only")

- **Temporal convolution** ("no conv."): Using a fully connected input layer amounts to dropping (a) the assumption of time-invariance of the ECoG data, as well as (b) the downsampling. It increased WERs by 20–40 percentage points.

Is the encoder-decoder really just a sentence classifier? For word-based decoding to stand any chance of succeeding, it is necessary to guarantee that the words read during a block used for testing have also been read at least once across the blocks used for training. Given our time constraints, we therefore decided to use, for each participant, a single set of sentences across all training and testing blocks. But this raises the possibility that the encoder RNN is merely *classifying* its inputs—say, with a label from the integers 1–50, which it then hands off to the decoder RNN. The latter

could in turn learn how to unpack the 50 labels into their corresponding sentences. This would render the results in Fig. 4 much less general and (therefore) interesting.

The short answer is that the encoder-decoder is *not* merely acting as a sentence classifier. We return to this point in the **Discussion**. Here we show that it performs better than other sentence classifiers, and that *performance improves when it is trained on sentences outside the test set*.

- **Sequence-length information** ("length-info only"): Recall that the ECoG sequences were manually extracted at the sentence boundaries (Fig. 2a). Therefore, if time of production varied more across than within sentence types, it would theoretically be possible for the network to *classify* input sentences based only on their length. We tested this by replacing each ECoG sequence with a sequence of pure noise—but still of the true length—and then re-training and testing networks. This resulted in WERs near 100% for all participant (Fig. 4).
- **Sentence classification** ("phoneme-based HMM"): We compared against a state-of-the-art, HMM-based sentence classifier for neural data [13]. It attempts to decode phonemes from ECoG data and then checks which of the sentences in a closed set (in our case, either MOCHA-1 or the picture descriptions) is most consistent with this phoneme sequence. Using this decoder increased WERs by 30–70 percentage points.
- **Training on non-test-sentences**. For some participants, we were able to collect blocks with sentences outside the test set, in particular MOCHA-2–MOCHA-9. Adding these blocks to a training set originally consisting of two blocks of MOCHA-1 ("+task TL") improved decoding performance on MOCHA-1 by as

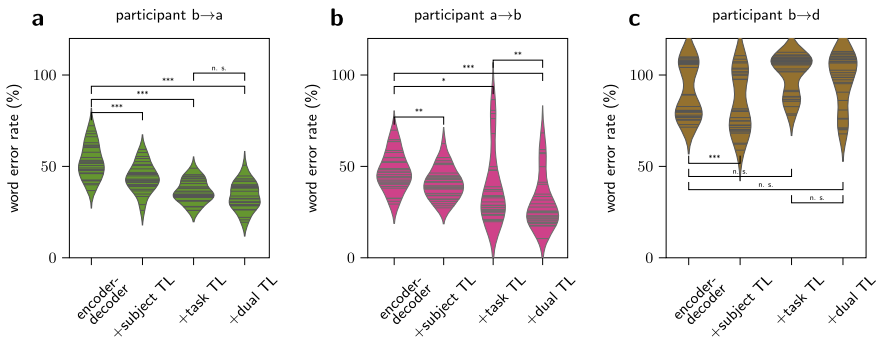

Fig. 5 Performance of the encoder-decoder under *transfer learning*. For the three participants (color code as in Fig. 4) with sufficient coverage of MOCHA-1, 30 encoder-decoders were tested on a randomly held-out block of MOCHA-1. They were trained on 2 blocks of MOCHA-1 ("encoder-decoder"), or on those two blocks *plus*: one block of each of MOCHA-2–MOCHA-9 ("+task TL"); another participant's MOCHA-1 blocks ("+subject TL"); or two participants' MOCHA-1–MOCHA–9 blocks. Significance, indicated by stars (*: $p < 0.05$, **: $p < 0.005$, ***: $p < 0.0005$, n.s.: not significant), was computed with a one-sided Wilcoxon signed-rank test, and Holm-Bonferroni corrected for 14 comparisons: the 12 shown here plus cross-subject transfer learning on the picture descriptions, which did not yield significant improvements

much as 15 percentage points (Fig. 5). This would be impossible if the encoder-decoder were merely classifying sentences of MOCHA-1.

Cross-participant transfer learning. In addition to demonstrating that the encoder-decoder is not acting as a sentence classifier, "cross-task" transfer learning shows how to augment our training data, which is of interest given how limited data-collection time is with epilepsy patients. Still, the total is upper bounded by the amount of a time a participant spends in the hospital. This bound could be breached, however, if we could exploit data from *other participants*. It turns out that we can, by pre-training the encoder-decoder on one participant before training it on another, target participant. On the MOCHA-1 sentences, this cross-participant transfer learning shows significant, albeit modest, reduction in WER for all participants (Fig. 5, "+subject TL"). The two forms of transfer learning can also be combined to yield further improvements (Fig. 5, "+dual TL").

4 Discussion

We decoded speech from ECoG data with error rates near zero, but only in the context of some 50 sentences comprising 250 unique words. The rigidity of the trained network can be seen in some of its errors, when it substitutes a whole sentence of MOCHA-1 for another. On the other hand, we showed that the encoder-decoder is not merely classifying input sentences, since training on non-MOCHA-1 sentences improves performance.

It also turns out that if the encoder is trained on ECoG sequences corresponding to single words (rather than single sentences), the correct word can be identified from its final hidden state with accuracies of up to 80% (for participant **b**; unpublished data), at least on the 250 words of MOCHA-1. The result is not fully general because of coarticulation: a word that appears in MOCHA-1 only after some other word may be produced differently in other contexts. But it very strongly suggests that the architecture can generalize to arbitrary sentences composed from a vocabulary of at least 250 words—given the appropriate training corpus.

The larger remaining questions have to do with clinical translation into patients who have lost the ability to speak. There are at least two problems:

1. It will no longer be possible to train the encoder-decoder to predict speech audio, which will hurt performance (see **Results**).
2. Cortical plasticity post-injury (for example) may obscure or eliminate the relevant neural signals.

In fact, the MFCCs can be replaced with phoneme sequences, sometimes with no drop in performance, and it may be possible (although not easy) to estimate these—or, for that matter, the MFCCs—by controlling the timing of the task. And it may be possible to learn much of the decoder from healthy patients via transfer learning (see **Results**), although it remains to be seen how effectively models transfer to non-speaking patients.

Acknowledgements The project was funded by a research contract under Facebook's Sponsored Academic Research Agreement. Data were collected and pre-processed by members of the Chang lab, some (MOCHA-TIMIT) under NIH grant U01 NS098971. Some neural networks were trained using GPUs generously donated by the Nvidia Corporation.

References

1. Angrick M, Herff C, Mugler E, Tate MC, Slutzky MW, Krusienski DJ, Schultz T (2019) Speech synthesis from ECoG using densely connected 3D convolutional neural networks. J Neural Eng
2. Anumanchipalli GK, Chartier J, Chang EF (2019) Speech synthesis from neural decoding of spoken sentences. Nature 568(7753):493–498
3. Brumberg JS, Kennedy PR, Guenther FH (2009) Artificial speech synthesizer control by brain-computer interface. In: Interspeech, pp 636–639
4. Brumberg JS, Wright EJ, Andreasen DS, Guenther FH, Kennedy PR (2011) Classification of intended phoneme production from chronic intracortical microelectrode recordings in speech-motor cortex. Front Neuroeng 5:1–12
5. Caruana R (1997) Multi-task learning. Multitask Learn 28:41–75
6. Cho K, Van Merrienboer B, Bahdanau D, Bengio Y (2014) On the properties of neural machine translation: encoder–decoder approaches. In: Proceedings of SSST-8, eighth workshop on syntax, semantics and structure in statistical translation, pp 103–111
7. Cho K, Van Merrienboer B, Gulcehre C, Bahdanau D, Bougares F, Schwenk H, Bengio Y (2014) Learning phrase representations using RNN encoder-decoder for statistical machine translation. In: 2014 conference on empirical methods in natural language processing (EMNLP), pp 1724–1734
8. Gehring J, Auli M, Grangier D, Yarats D, Dauphin YN (2017) Convolutional sequence to sequence learning. In: 34th international conference on machine learning, ICML 2017, vol 3, pp 2029–2042
9. Herff C, Heger D, De Pesters A, Telaar D, Brunner P, Schalk G, Schultz T (2015) Brain-to-text: decoding spoken phrases from phone representations in the brain. Front Neurosci 9:1–11
10. Kingma DP, Ba J (2014) Adam: a method for stochastic optimization
11. Makin JG, Moses DA, Chang EF (2020) Machine translation of cortical activity to text with an encoder-decoder framework. Nat Neurosci 23:575–582
12. Martin S, Brunner P, Holdgraf C, Heinze HJ, Crone NE, Rieger J, Schalk G, Knight RT, Pasley BN (2014) Decoding spectrotemporal features of overt and covert speech from the human cortex. Front Neuroeng 7:1–15
13. Moses DA, Leonard MK, Makin JG, Chang EF (2019) Real-time decoding of question-and-answer speech dialogue using human cortical activity. Nat Commun 10(1)
14. Mugler EM, Tate MC, Livescu K, Templer JW, Goldrick MA, Slutzky MW (2018) Differential representation of articulatory gestures and phonemes in precentral and inferior frontal gyri. J Neurosci 4653(46):1206–1218
15. Munteanu C, Penn G, Baecker R, Toms E, James D (2006) Measuring the acceptable word error rate of machine-generated webcast transcripts. In: Interspeech, pp 157–160
16. Pei X, Barbour DL, Leuthardt EC (2011) Decoding vowels and consonants in spoken and imagined words using electrocorticographic signals in humans. J Neural Eng 8(4):1–11
17. Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R (2014) Dropout: a simple way to prevent neural networks from overfitting. J Mach Learn Res 15:1929–1958
18. Stavisky SD, Rezaii P, Willett FR, Hochberg LR, Shenoy KV, Henderson JM (2018) Decoding speech from intracortical multielectrode arrays in dorsal "arm/hand areas" of human motor cortex. In: Proceedings of the annual international conference of the IEEE Engineering in Medicine and Biology Society, EMBS, pp 93–97

19. Sutskever I, Vinyals O, Le QV (2014) Sequence to sequence learning with neural networks. In: Advances in neural information processing systems 27: proceedings of the 2014 conference, pp 1–9
20. Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A (2015) Going deeper with convolutions. In: 2015 IEEE conference on computer vision and pattern recognition (CVPR). IEEE, pp 1–9
21. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser L, Polosukhin I (2017) Attention is all you need. In: Advances in neural information processing systems, pp 5998–6008
22. Wrench A (2019) MOCHA-TIMIT. Online database
23. Xiong W, Droppo J, Huang X, Seide F, Seltzer ML, Stolcke A, Yu D, Zweig G (2017) Toward human parity in conversational speech recognition. IEEE/ACM Trans Audio Speech Lang Process 25(12):2410–2423