Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Distributed Computing

Brain2Word: Decoding Brain Activity for Language Generation

Master's Thesis

Nicolas Affolter

nicolaff@ethz.ch

Distributed Computing Group
Computer Engineering and Networks Laboratory
ETH Zürich

Supervisors:
Beni Egressy, Damian Pascual
Prof. Dr. Roger Wattenhofer

September 30, 2020

Acknowledgements

I thank Beni Egressy and Damian Pascual for supporting and guiding me with my master's thesis. Their inputs were really useful and sometimes challenging, which let me learn and improve my skills a lot.

Abstract

Brain decoding, understood as the process of mapping brain activities to the stimuli that generated them, has been an active research area in the last years. In the case of language stimuli, recent studies have shown that it is possible to decode fMRI scans into an embedding of the word a subject is reading. However, such word embeddings are designed for natural language processing tasks rather than for brain decoding. Therefore, they limit our ability to recover the precise stimulus. In this work, we propose to directly classify an fMRI scan, mapping it to the corresponding word within a fixed vocabulary. Unlike existing work, we evaluate on scans from previously unseen subjects. We argue that this is a more realistic setup and we present a model that can decode fMRI data from unseen subjects. Our model achieves 5.22% Top-1 and 13.59% Top-5 accuracy in this challenging task, significantly outperforming all the considered competitive baselines. Furthermore, we use the decoded words to guide language generation with the GPT-2 model. This way, we advance the quest for a system that translates brain activities into coherent text.

Contents

Acknowledgements i

Abstract ii

1 Introduction 1

2 Related Work 3

2.1 Brain Decoding . . . 3

2.2 Language models and the brain . . . 3

3 Background 5

3.1 Dataset . . . 5

3.2 Data Alignment . . . 6

3.3 Evaluation Tasks . . . 8

3.3.1 Pairwise classification . . . 8

3.3.2 Direct classification . . . 8

3.4 Evaluation Setup . . . 9

4 Brain Decoding Model 10

4.1 Regions of Interest . . . 12

4.2 Autoencoder . . . 12

4.3 Mean regularization . . . 13

4.4 Unsupervised pretraining . . . 13

4.5 Convolutional Approach . . . 13

4.6 Regularization . . . 14

5 Results 16

5.1 Ablation Study . . . 16

5.2 Pairwise Classification . . . 17

5.3 Direct Classification 18
5.4 Data Alignment 19
5.5 Convolutional Approach 20

6 Bridging fMRI to Language Generation 22
6.1 Results 23
6.2 GPT-2 Finetuning 25
6.3 Discussion 25

7 Conclusion 27

Chapter 1

Introduction

Recent advances in brain imaging suggest that it may be possible to infer what a person is perceiving from their brain scans. The ability of decoding brain signals has important applications in medicine, e.g., assisting handicapped people who cannot move or talk, as well as in the consumer industry, e.g., producing content that adapts to what a person is seeing, feeling or thinking. In this context, language is of particular interest, since it is the vehicle we use to express our thoughts. A body of research has focused on decoding functional Magnetic Resonance Imaging (fMRI) scans into a representation of the word a person is reading while being scanned. By measuring similarity between the decoded scan and actual word representations, they show that the decoded representation is closer to its corresponding word representation than to another word with a chance significantly higher than random.

Although an important first step in showing that inferring such information from brain scans is at all possible, this task is rather simple and has limited potential applications. The inference models used to solve it are equally simple, normally based on ridge regression or simple Multi-Layer Perceptrons (MLP), while they rely on complex subject-specific pre-processing and feature selection (Pereira et al., 2018; Sun et al., 2019). In this work, we argue that a more demanding setup needs to be considered in order to understand the extent to which we can currently map brain activities to words. In particular, we propose direct classification, i.e. to directly classify a brain scan as one of the v words within the considered vocabulary, as opposed to pairwise classification. Furthermore, we address brain decoding on unseen subjects, i.e., the training data does not contain any data from the test subject. This is known to be a remarkably hard problem, since fMRI scans are very different across subjects and even across recording sessions, among other reasons due to variable numbers of voxels and lack of alignment between scans. Thus, the challenge with this setup is twofold, the evaluation task is more demanding and strong generalization is required since subject-specific pre-processing is not possible.

On the bright side, in this setup we can exploit a larger training set consisting of the scans from $n - 1$ subjects in order to train more complex models, where

n is the number of subjects in the dataset. We propose a neural autoencoder model that takes as input a complete fMRI scan and outputs the stimulus word. We use minimal external knowledge, specifically the Regions of Interest (ROIs) of the brain scan, and let the model learn features that generalize to all subjects. We validate our model on the classical pairwise classification task and then, we demonstrate its performance in direct classification.

Then, we take a novel research direction and consider a practical application of brain decoding. We envision a system that decodes concepts from the brain, rather than complete sentences, and uses these concepts to guide the generation of coherent text. Such a system could help individuals with speech impairments to communicate. To this end, we leverage GPT-2, a recently proposed model for language generation which can produce outstandingly realistic text. We condition GPT-2 with the decoded brain scans and show, as a proof-of-concept, that brain activities can guide language generation. Although a long path still needs to be covered before having a fully functional system, our work sets a first stone towards translating brain activities into coherent text.

All in all, our contributions are:

- We propose a new and more challenging evaluation setup for brain decoding, i.e., to decode the brain activation from a subject unseen during training directly into a specific word in a bounded vocabulary.
- We present a neural network-based model that improves fMRI-to-word decoding by a significant margin in the existing evaluation framework as well as in our more challenging and realistic setup.
- We bridge fMRI decoding to a real-world application: language generation conditioned on brain activities.

Chapter 2

Related Work

2.1 Brain Decoding

Since the publication of the seminal work (Mitchell et al., 2008), decoding brain activity into words has attracted a lot of attention from the research community. In recent years, a large number of studies has tackled this problem from different angles. (Palatucci et al., 2009) proposed a model to learn new classes unseen during training, (Just et al., 2010; Huth et al., 2016; Handjaras et al., 2016) built brain decoders that helped them draw conclusions about the way the brain processes language. (Pereira et al., 2018) presented a model that decodes brain activity into word embeddings. (Wehbe et al., 2014) decoded text passages rather than single words and, similarly, (Sun et al., 2019) decoded sentences using distributed representations. These works represent just a part of a large body of research (Wang et al., 2020; Schwartz, Toneva, and Wehbe, 2019; Kivisaari et al., 2019) that has strongly contributed to the progress of decoding and understanding brain activities.

In most existing literature the scans used for training come from the same subject that is evaluated. Due to the misalignment of brain scans between subjects, evaluating on a different subject is a very challenging problem. Recent work (Van Uden et al., 2018; Nastase et al., 2020) has studied this problem in controlled settings and approached it from an algorithmic perspective. Here, we take a data-driven approach to successfully generalize brain decoding to unseen subjects.

2.2 Language models and the brain

The field of Natural Language Processing (NLP) has undergone outstanding progress in the last few years thanks to a family of deep learning models called Transformers (Vaswani et al., 2017; Liu et al., 2019; Raffel et al., 2019). These models are currently state-of-the-art in most NLP tasks and remarkably, in language generation, e.g., the GPT-2 model (Radford et al., 2019). Recent work

has tried to establish a link between the brain and these models. (Gauthier and Levy, 2019) decodes fMRI to improve latent representations inside a transformer for NLP tasks. (Toneva and Wehbe, 2019) use fMRI scans to interpret and improve BERT (Devlin et al., 2018), a well-known transformer. Relatedly, (Muttenthaler, Hollenstein, and Barrett, 2020) use EEG features to modify attention weights in an LSTM based model.

Different from prior work, we devise a direct application of brain decoding: to use brain activities in order to guide language generation with GPT-2. (Nishimoto et al., 2011) demonstrated that it is possible to dynamically decode brain activity in the form of fMRI scans. Based on this result, we advance towards a brain-computer interface capable of translating brain activity into coherent text.

Chapter 3

Background

We call *brain decoder* or simply *decoder* a function capable of mapping brain activities to the stimulus that generated them. In this work, we map brain activities in the form of fMRI scans to the text presented to subjects during scanning. We consider two types of decoders, first, classical regression-based decoders (Bulat, Clark, and Shutova, 2017) which learn a parametric mapping from the fMRI scan to a vector representation of the text; and second, we propose classification-based decoders, which learn to map brain activities to a word within a bounded vocabulary.

3.1 Dataset

We use the dataset from (Pereira et al., 2018). This dataset contains fMRI scans from 15 subjects. Each subject was recorded reading 180 different words, one at a time. Each word, was shown to the subject following three different paradigms that ensure that all subjects focus on the same meaning, i.e., supporting the word with a word cloud, with sentences and with images. Additionally, 8 of the subjects were scanned while reading sentences from a dataset that consists of 384 sentences from 96 different passages. Finally, 6 of the subjects (with overlap with the 8 previous subjects) were also scanned reading another dataset of 243 sentences from 72 passages. 6 subjects were not recorded reading sentences.[1]

In this work, we are interested in decoding individual words and so, our dataset consists of 15 subjects with 540 scans each (180 words, three paradigms). However, we also explore pretraining our model with the sentence recordings.

Combining this dataset with other datasets was considered thoroughly, since this would further increase the overall dataset for training our models. One such datasets is the Harry Potter fMRI dataset from (Wehbe et al., 2014).[2] It contains fMRI scans of eight subjects, which were taken while each subject was

[1] For more details on the dataset refer to https://osf.io/crwz7

[2] For more details on the Harry Potter dataset refer to http://www.cs.cmu.edu/~fmri/plosone/

reading the first Harry Potter book. The main difference though between our main dataset and this one is, that the scans were not taken for every individual word but for four words at the time. This means that we can not relate each scan to one word specifically during training like we can do with the dataset from (Pereira et al., 2018). Since this is not possible, we considered using the Harry Potter dataset for pretraining, like we do with the sentence dataset. To do this the fMRIs from both datasets need to be aligned. In our simplest approach of just selecting the voxels, placing them in a 1D vector and use zero padding to match them to the same size, an issue arose based on the dimensionality of the two datasets. Our dataset contains about ten times as many voxels as the Harry Potter fMRI dataset, which means that in this simple matching procedure, most of the values in the vector would be set to zero for a sample from the Harry Potter fMRI dataset. This is why the addition of such a dataset would rather hinder the learning of our model instead of improving it. Since most other datasets we could find have similar issues, when comparing the dimensionality or other attributes, based on how the fMRIs were extracted, we decided to no longer consider the combination of different fMRI datasets in our approach.

3.2 Data Alignment

Since combining fMRI data from different subjects is known to be a difficult problem, a lot of time was invested in finding a good way of aligning the fMRI scans to provide a better preprocessed training dataset for our models. To understand why this is a demanding task, a brief introduction is needed, on the way the data was generated.

For every fMRI scan a 3D array with dimensions 88x128x85 is returned. Each value in this array represents a voxel of 3mmx3mmx3mm. Overall this means we have 957,440 voxels in our array which cover the whole brain. After preprocessing only about 20 percent of the voxels are still considered to have valid values and all the others are set to be zero.[3] Since preprocessing is different for all subjects, not every subject has non-zero values for the same voxels.

At this point we need to introduce an additional concept for handling fMRI scans. The fMRI scans can be partitioned into Regions Of Interest (ROIs) following the atlas from (Gordon et al., 2016), provided in the dataset. Each ROI is associated with one or more brain functions. Now once we have these voxels from the first preprocessing step, we can use the knowledge about ROI regions and apply ROI region voxel selection. This way we only retain voxels from brain regions relevant to our task. The voxels, which correspond to a certain ROI region, vary across subjects. This means that subjects can not only have very different ROI region locations in the brain, but also have a completely different number of

[3]For more details on preprocessing and voxel selection refer to https://www.nature.com/articles/s41467-018-03068-4.pdf

voxels for each ROI region. Since each brain is unique these observations should not be surprising, but they bring up important issues in regard to aligning the fMRIs across subjects in a useful manner.

After having some insights on why it is a demanding task to align the datasets of the subjects, we can look at some of our attempts on how it could be done. For all our approaches only non-zero voxels from selected ROI regions were considered. Our first approach and also most simple approach is to just select the subject with the highest number of remaining voxels (which was 51,494) and pad up all the other subjects to match this number. This is simple but losses all the spatial information we still had left after all the selection methods and hence does not align voxels from similar locations in the brain. Therefore, it was the focus of our more sophisticated method to conserve the spatial relationships between subjects as far as possible.

The best way to do this would be to align the voxels based on their location in the 3D grid, since this way the spatial location would be perfectly preserved. However, as mentioned, the locations of the ROI regions are not consistent among the subjects. We could consider the relevant ROI regions and take a union of all the corresponding voxel coordinates over all subjects and then use all these voxels for each subject, padding with zeros when necessary. At first this sounds very promising, since all spatial information can be preserved this way. On a second look one realizes, that this increases our fMRI vector size to 155,131 values, which is more than 2.5 times the input size we had for our first method. Additionally, we also add a lot more zero values to our model input, which can hinder the model to learn useful information. Therefore, this is not an acceptable solution, since our input size to our model was already in our previous approach big in comparison to the available amount of training data.

The best course of action seems to be a combination of the two. Instead of aligning all the voxels perfectly based on their spatial location, we will align the individual ROI regions. Since each ROI region has a different number of voxels across subjects, we need to select the highest number of voxels for every ROI region. This only increases our total number of voxels from our simple approach to 61,656. For every subject's ROI region which does not have the same number of voxels as the subject with the highest number of voxels for a given ROI region, we pad up the missing values. This way the ROI regions are better aligned.

For our final approach we wanted to include more of the spatial location of the voxels again without increasing the input vector size too much from our previous approach (New input size: 65,730). The idea was to match the voxels within a certain ROI region better. Since the voxels within the same ROI region across subjects are not from the same spatial location and also do not have the same number of voxels, this becomes a demanding task. In a first part the structure of each ROI region of each subject was inspected and compared. The best way of aligning them seemed to be along the z-axis in the 3D grid. This means that every

ROI region was split into slices based on the z-axis location of the voxels. Next, the slices were compared for each ROI region and across all subjects. Since neither the number of slices nor the number of voxels per slice matched, the alignment had to be completed with padding. The slices were aligned from lowest to highest slice (based on the z-axis). If a subject ran out of slices for a given ROI region, the remaining voxels would be padded with zeros. Within the slices the same approach was followed. The size of the slices and also the maximum number of slices was based on the highest number of slices and also voxels per slices across all subjects for a given ROI region respectively. This last approach for aligning the data will also be used for all our models presented in the following sections.

3.3 Evaluation Tasks

3.3.1 Pairwise classification

In this task, a regression-based decoder is trained to produce a vector representation from a brain image (fMRI). Then, for each possible pair of words the correlation between the decoded vectors and the actual embedding vectors of both words is computed, i.e., four values. If the decoded vectors are more similar to their corresponding word embeddings than to the alternatives, the evaluation is considered correct. As such, the random baseline for this task is 0.5. The final result is the mean across test instances.

This task presents certain limitations arising from the representation the brain image is decoded into. In (Pereira et al., 2018), this representation is a GloVE embedding (Pennington, Socher, and Manning, 2014) and therefore, it contains information beyond semantics, such as word frequency in the data used to train the embedding. (Gauthier and Ivanova, 2018) show that decoding brain images into representations derived from models optimized to solve very different tasks, e.g., image captioning or machine translation, produce similar results as the baseline decoder from (Pereira et al., 2018). This suggests that training the decoder to produce a certain representation vector is fundamentally limited by the type of representation used. Therefore, as an additional evaluation task, we propose direct classification.

3.3.2 Direct classification

In this task, a classification-based decoder receives as input a brain scan and produces as output a vector of size v, where v is the size of the vocabulary. In our case $v = 180$. This vector contains the predicted probability for each word in the vocabulary. This way, the decoder is effectively a classifier that infers which word is seen by the subject when the scan is taken. This task is significantly more challenging than the pairwise classification task, with the random baseline

being $1/v$ for the Top-1 score (0.06% in our case). On the other hand, it does not suffer from limitations associated with the chosen vector representation. In this task, we report Top-1 and Top-5 scores, i.e., the classification is correct if the word is within the Top-X predictions.

3.4 Evaluation Setup

Most previous work on brain decoding (Pereira et al., 2018; Sun et al., 2019) has considered the scenario where the model is trained with data from the same subject that is being evaluated. We argue that this scenario is not suitable for real life applications and that it in fact limits our ability to decode brain activities.

In said scenario, for each new subject a training set needs to be recorded in order to train a personalised decoder model. Recording fMRI scans is a costly and slow process, e.g., for one subject in our target dataset it takes approximately 4 hours just to obtain the 180 brain scans, which still have to be processed.[4] Therefore, it is desirable to have models that are able to decode the brain activities of new subjects without the need for subject-specific training data. Furthermore, the amount of data that can be recorded for one subject is limited, which restricts the complexity of the decoders and forces the model to rely on subject-specific pre-processing. By using the data from all recorded subjects, we can build larger neural network-based decoders that learn general features across subjects, dispensing with the need for subject-specific processing steps. This however is a difficult problem since there is almost no alignment between the fMRI scans of different subjects.

In this work, we consider this challenging setup and follow a leave-one-out strategy in our evaluation. This way, we train our model with the data from $n - 1$ subjects and test it on the remaining subject; we repeat this process for each subject. This simulates a real-world scenario where an existing model is applied to a new, as yet unseen, subject.

[4] Five repetitions per paradigm (three paradigms), where each repetition needs two runs (90 words per run) and each run takes 8 minutes (Pereira et al., 2018).

Chapter 4

Brain Decoding Model

We propose a new model that leverages recent advances in deep learning in order to decode brain activities in the form of fMRI scans into text. Our model can be implemented as either a regression-based or a classification-based decoder by simply changing the last layer and the related term of the loss function. The regression-based decoder has a final linear layer that outputs a vector of the size of the target representation; following (Pereira et al., 2018) we use GloVE embeddings of size 300×1. The regression loss is calculated on this output and has the form:

$$\mathcal{L}_{reg} = \sum_{i}^{v} \left(\cos\left(y_{pr,i}, y_{true,i}\right) - \sum_{j \neq i}^{v} \cos\left(y_{pr,i}, y_{true,j}\right) \right)$$

Where $y_{pr,i}$ is the predicted word embedding for word i, $y_{true,j}$ is the real word embedding for word j and $\cos(x, y)$ is the cosine distance between vectors x and y. Note that to ease notation this formulation corresponds only to one paradigm of one subject, the total loss is calculated by summing over all paradigms for each subject in the training set. The same applies to the formulation of the other loss terms presented below. This loss is inspired by the triplet loss (Schroff, Kalenichenko, and Philbin, 2015) and aims at guiding the model's output as close as possible to the true embedding while keeping it as far as possible from the embeddings of the other words in the vocabulary. We observed that this loss function helped to prevent the model from collapsing towards a mean representation of the word embeddings.

In the classification based decoder, the regression layer of size 300×1 is turned into a non-linear layer and an additional softmax layer is added on top of it. This way the model outputs a one dimensional vector of probabilities of the size of the vocabulary v, 180×1 in our case. The classification loss is given by the categorical cross-entropy between the vector of predicted probabilities y_{pr} and the one hot representation of the target word y_{true}:

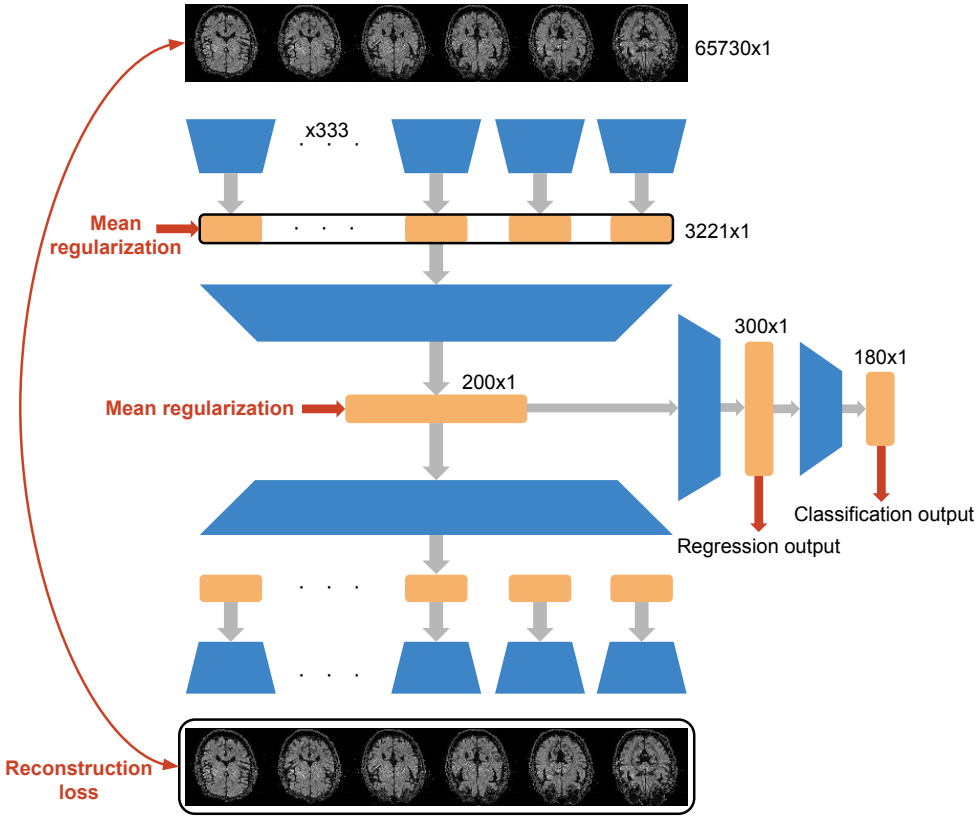

Figure 4.1: Architecture of the improved decoder. The blue trapezoids represent dense layers, the orange rectangles feature maps and the solid black lines concatenation. The shape of the feature maps is specified, as well as the points where the different regularization terms are applied.

$$\mathcal{L}_{class} = -\sum_{i}^{v} y_{true,i} \cdot log(y_{pr,i})$$

Where i is the index for a given word in the vocabulary.

We first consider a simple model which takes as input a one dimensional fMRI scan of size $65,730 \times 1$ voxels and generates a latent vector of size 200×1; the input size is such that all the scans in the dataset fit. The exact way on how we fit the scans can be found in the data alignment section. The latent vector is used to produce either the regression or the classification target. Apart from the regression and classification layers, the model consists of two non-linear fully connected layers that produce feature maps of size 2000×1 and 200×1

respectively. Each non-linear layer has 0.4 dropout, batch normalization and Leaky ReLU activation ($\alpha = 0.3$). We take this simple model as a base model and improve it with the extensions detailed below. We do an ablation study of the extensions in the results section to measure their impact on the performance of the model. The complete model is depicted in Figure 4.1.

4.1 Regions of Interest

As mentioned in the data alignment section, fMRI scans can be partitioned into Regions Of Interest (ROIs), for which we follow the atlas from (Gordon et al., 2016). To exploit this knowledge and reduce the size of the model, we process each region separately in the first layer of our model. This way, we use one dense layer for each of the 333 ROIs from the atlas and concatenate their outputs. Note that the ROIs vary in size, and thus, so do the individual dense layers. We set each of these to produce an output vector of size $max(\frac{ROISize}{20}, 1)$, where the factor 20 is a hyperparameter chosen to regulate the size of the hidden layer. On our hyperparameter search we found this value to be adequate.

4.2 Autoencoder

We turn the model into an autoencoder (decoder-encoder) by adding an encoder that mirrors the base model, i.e., the decoder. This encoder reconstructs the input brain activities (fMRI) from the latent vector and to this end, we add a reconstruction term to the loss function. The reconstruction loss is given by:

$$\mathcal{L}_{rec} = \sum_{i}^{v} \cos\left(x_{out,i}, x_{in,i}\right)$$

Where x_{in} is the input fMRI scan and x_{out} is the reconstructed fMRI, i.e., the output of the encoder. The rationale behind using an autoencoder is that the reconstruction loss should help learning by increasing the training signal and by acting as a regularizer.

Since we do not want the number of trainable weights to increase further when adding the encoder and also to make the regularization stronger, the autoencoder was built with transposed dense layers instead of regular dense layers. This way the weights of the encoder are linked to the decoder, which limits overfitting by the autoencoder.

4.3 Mean regularization

Since the model should produce the same output for scans from different subjects when exposed to the same word, the latent representations inside the model should converge. In other words, we expect the model to progressively discard subject-specific physiological information in order to extract the word the subject is reading. To this end, we regularize the output of each layer of the decoder to be similar to the mean representation for a given word across subjects at that layer and dissimilar to the mean representation of the other words. More formally, we add a term to the loss function with the same structure as the regression loss:

$$\mathcal{L}_{mean} = \sum_{i}^{v} \left(\cos\left(h_i^{(l)}, h_i^{(l)}\right) - \sum_{j \neq i}^{v} \cos\left(h_i^{(l)}, h_j^{(l)}\right) \right)$$

Where $h_i^{(l)}$ is the predicted hidden representation of word i in layer l and $h_i^{(l)}$ is the mean of the predicted hidden representations for word i at layer l across all subjects; these mean representations are updated after every epoch. At the beginning of the training the model produces meaningless representations and, for this reason, we first train without mean regularization and only when learning saturates, the mean regularization is activated and the model resumes training until early stopping occurs.

4.4 Unsupervised pretraining

As detailed above, the dataset from (Pereira et al., 2018) contains two additional sets of fMRI scans, amounting to a total of $4,530$ scans. These scans were recorded while subjects read sentences, instead of the words that conform our learning target. Therefore, we can only use these additional scans in an unsupervised manner. The autoencoder structure of our improved model allows us to do this: we pretrain our model on the sentence scans using exclusively the reconstruction loss $\mathcal{L}_{rec}$ for 30 epochs. Afterwards, we start the supervised training phase on the word scans. With the pretraining phase we aim to exploit general language-related fMRI features in order to place our model at a better starting point. This can help the model to eventually reach a better minimum on our training objective.

4.5 Convolutional Approach

Usually in machine learning if we are dealing with large data arrays as input, we tend to use convolutional networks, since they use less weights and can also

take advantage of local spatial coherence, which is good in our case, where voxels next to each other in the input vector are also spatially close to each other in the brain scan. Since our input is given as 1D vectors, we decided to use 1D convolutional layers. We use this approach in two different set ups. In a first approach we replace all the dense layers from our complete model, except for our dense layer for predicting the GloVe embedding and the classification layer, with 1D convolutional layers. In addition to that we replace in a second approach the ROI region small layers with one bigger convolutional layer. Each convolutional layer was completed with a max-pooling layer for the decoder and a up-sampling layer for the encoder. For the first approach the small convolutional layers used 4 filters and a kernel size of 4 as well. The max-pooling layer size was set to 10 with a stride of 10 as well. In the second approach these small layers were replaced with one bigger convolutional layer with 4 filters as well but a kernel size of 20. The max-pooling layer was kept the same. The remaining layers were the same for both approaches. The follow up convolutional layer used 4 filters and had a kernel size of 4 as well. The max pooling layer size was set to 2 with a stride of 2. The loss functions were kept the same for both convolutional models as we used them in our main model.

4.6 Regularization

Models with such large input size and only having very limited data available for training are prone to overfitting. Since a reduction of the input size, increase of available training data or better alignment of the individual subject's data is not possible without hindering at least one of the other two, a form of regularization has to be applied to reduce overfitting on the training subjects and make the model focus on information shared between the subjects. As mentioned in previous sections dropout for all the layers and also turning the model into an autoencoder were first methods of regularization for our model. Another straight forward one is to reduce the individual layer sizes of the intermediate layers of the model as well as the small ROI dense layers.

During the development of our main model many other regularization methods were looked at and we want to briefly touch on two of them. We tried to use LASSO and L2 layer regularization applied to individual layers. This way the weights of each layer get punished for becoming too big and therefore adapting too much on the training dataset. After some experiments with this idea it became evident that the regularization was too strong for our model, since it increased the training loss and also did not improve the validation loss either. Next we looked into a better suited regularization loss function for our layers: The Group LASSO. This approach is less restrictive than our previous one and shouldn't affect the training loss as strongly. Nevertheless, it turned out the addition of the LASSO Group or of any other layerwise regularization only in-

creases the training as well as the validation loss and is therefore not a useful regularization method in our case.

Overall it became evident that the best regularization method is the reduction of the intermediate layer sizes. This way we do not only reduce the size of the weight matrices, which gives the model less options to overfit on the training data, but also reduces the amount of information the model can pass on to further layers and it should focus on the features, which can be found in the data from all the subjects. Hence a lot of time was invested to find the right combination of layers as well as the right layer sizes, since only the input as well as the output size were fixed.

Chapter 5

Results

As explained above, we follow in our experiments the leave-one-out approach, i.e., we use all subjects except one for training and we evaluate on this left-out subject. This process is repeated for all subjects. We use subject $M15$ for validation and the rest for testing. We perform the ablation study on the validation subject. Likewise, the hyperparameters used in our model and described so far are also found by grid search evaluated on the validation subject. The evaluation on the remaining 14 test subjects is used for the final results on the pairwise and classification tasks.

As already mentioned, for the calculation of the pairwise accuracy, we follow (Pereira et al., 2018) in all cases and use GloVE embeddings as the decoding target.

5.1 Ablation Study

In this study, we first consider the base model and then progressively add the extensions in the order they were presented. We evaluate both versions of the model: the regression-based and classification-based decoder. In Table 5.1, we report the pairwise classification accuracy for the regression-based decoder, and the Top-1 and Top-5 scores for the classification-based decoder.

Model	Pairwise	Top-1	Top-5
Base	0.8268	4.07%	11.66%
+ ROI	0.8336	4.25%	11.85%
+ Reconstruction	0.8411	4.81%	12.96%
+ Mean reg.	0.8464	5.55%	13.14%
+ Pretraining	**0.8637**	**6.29%**	**15.00%**

Table 5.1: Ablation study. The extensions are progressively added to the model.

We see that all four extensions monotonically improve the performance of the

model for the three metrics. This study validates our design choices and thus, in the remaining experiments we use the complete model.

Finally, we want to further investigate the hypothesis that mapping brain activities to word embeddings is limiting the model by introducing unwanted information in the decoding target (e.g., word frequency) (Gauthier and Ivanova, 2018). To this end, we add to our best classification model the regression loss as an additional optimization objective. If the word embedding was a good representation of concepts, this additional term would help (by increasing the training signal), or at least not harm the classification performance. However, adding this loss term to the complete classification model degrades the Top-1 and Top-5 scores down to 5.89% and 13.55% respectively. This supports the hypothesis that the GloVE embedding is a noisy representation of the concept, which further underscores the need for a better evaluation task, such as our proposed direct classification.

5.2 Pairwise Classification

To put our model into context with respect to existing work, we evaluate it on the pairwise classification task and compare it with four competitive baselines. First, we take the model from (Pereira et al., 2018) which uses ridge regression, in the following we refer to this model as Universal Decoder. Second, we take a simple MLP consisting of a non-linear layer that maps the input to a feature vector of size 2000×1 followed by a linear layer that outputs the regression target, i.e., a GloVe embedding (300×1). Third, we take a big MLP with one non-linear dense layer per ROI, as in our complete model, followed by a linear layer that outputs the GloVe embedding. Last, we evaluate the VQ-VAE model from (Van Den Oord, Vinyals et al., 2017) adapted to regression-based decoding of fMRI. This model discretizes the latent space, thus, we hypothesize that it may naturally separate the scans according to the word that they encode.

Given the reduced capacity of the Universal Decoder, training it on subjects different than the test subject produced close to random performance. Therefore, we train and evaluate it in the same manner as in the original work (Pereira et al., 2018), i.e., for each paradigm 170 words of a given subject are used for training and the remaining 10 for testing. This is repeated 18 times over to cover all the words and the results are averaged. This is different from the other models where we follow the leave-on-out approach, i.e., the target subject is never seen during training.

We report the results on all the 14 test subjects in Figure 5.1. First, we see that the VQ-VAE has the worst performance, which rejects our hypothesis about the discrete latent space. All the other models outperform the Universal Decoder even with the disadvantageous training setup (unseen test subject). It

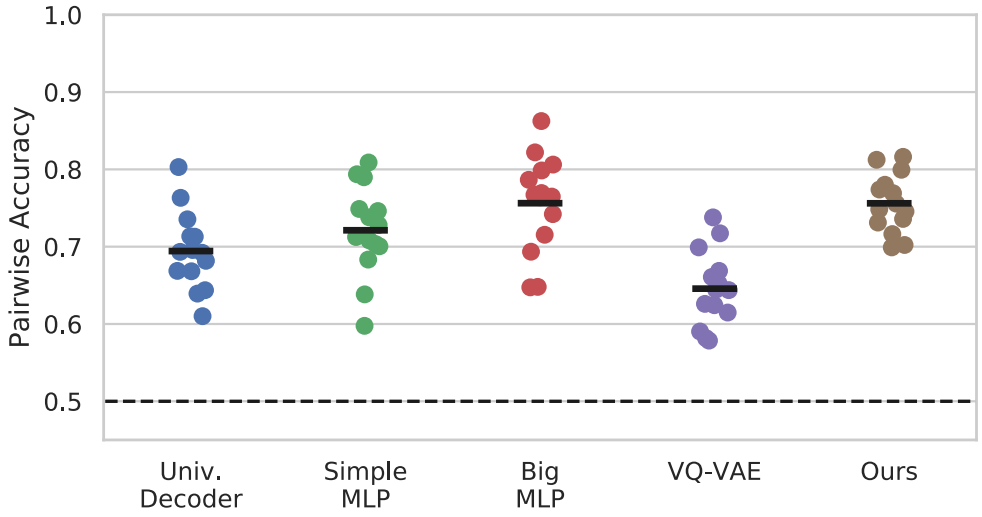

Figure 5.1: Pairwise accuracy of the different models. Each point represents a subject, the solid lines are the mean across subjects and the dashed line the random baseline.

is also noteworthy that the “Big MLP" performs on par with our model in this task, albeit with higher variance across subjects. These results show that neural network-based decoders successfully generalize to unseen subjects and even clearly outperform classical models trained on the target subject.

5.3 Direct Classification

Next, we evaluate our model in the direct classification task. We compare our model against five competitive baselines, the same four as above adapted to the classification task, and additionally, against Principal Component Analysis decomposition (PCA), for dimensionality reduction, followed by XGBoost (Chen and Guestrin, 2016), a tree-based classification algorithm. To perform classification using the Universal Decoder we take the output of the model and do nearest neighbour search on the GloVE embeddings of the 180 words of our vocabulary. For the other models, we turn the last layer into a classification layer by adding softmax and changing the output size to 180, i.e., the number of classes.

We represent the results in the same manner as in the previous section, Figure 5.2 shows the Top-1 scores and Figure 5.3 the Top-5. In this more complicated task (the random baseline is 0.6% for Top-1 and 2.8% for Top-5) the Universal Decoder mean accuracy is 0.94% for the Top-1 score and 4.5% for Top-5, slightly above random. Again, the VQ-VAE performs the worst among the neural models with scores similar to those of the Universal Decoder, and PCA plus XGBoost

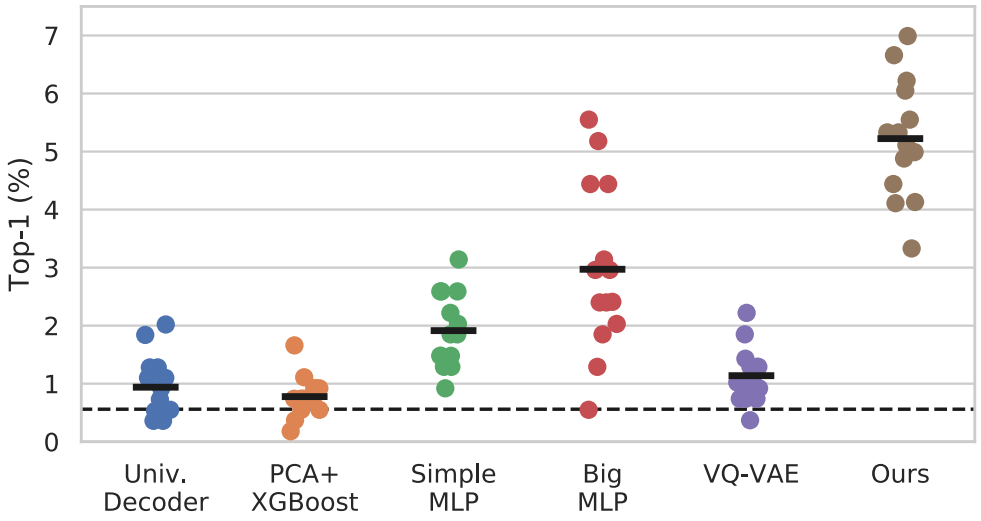

Figure 5.2: Top-1 Score across models.

performs very close to random. On the other hand, the Top-1 mean accuracy for the simple MLP, big MLP and our model is 1.91%, 2.97% and 5.22% respectively. We see that in this challenging setup, our complete model is clearly the best for both Top-1 and Top-5 scores. In particular, its Top-5 mean accuracy is above 13.59%, almost 5 times the random baseline. This result is outstanding given the difficulty of the task, i.e., decoding the exact word corresponding to the fMRI scan of an unseen subject.

The good performance of our decoder on this realistic scenario shows the potential of using brain decoding in real life applications. In the following chapter we show a proof-of-concept of how language generation can be guided by decoded fMRI scans.

5.4 Data Alignment

The results for our main model combined with each of our four data alignment methods can be found in Table 5.2. The first thing that becomes evident is that our second method with the exact voxel matching, which increases the input size to 155,131, performs the worst of all four methods for all evaluation scores. Further it can be seen that the third (ROI regions matching) and fourth (ROI region matching with additional z-axis alignment) approach have slightly better results than our first and most simple method and this in spite of the increase of the input vector size. The third and fourth approach have nearly the same score for the main model despite that the fourth approach has a much more complex

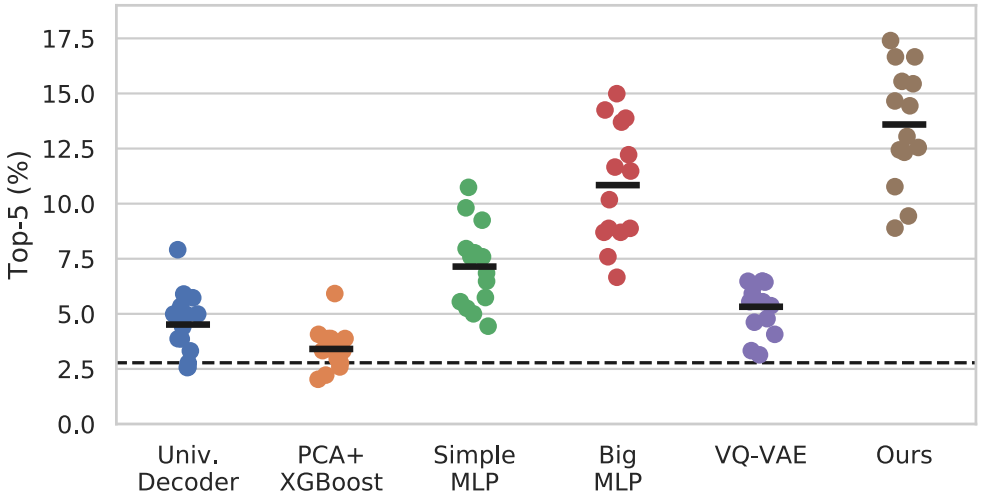

Figure 5.3: Top-5 Score across models.

data alignment method and also increases the input size. Overall it has to be said that the closeness in performance for all four of our methods shows that better inter subject data alignment can only make a limited contribution in improving the data prepossessing and therefore increasing the data value.

Model	Input Size	Pairwise	Top-1	Top-5
+ Simple	**51,494**	0.8588	5.77%	14.05%
+ Exact matching	155,131	0.8465	4.93%	13.21%
+ ROI	61,656	0.8622	6.03%	14.66%
+ ROI and z-axis	65,730	**0.8637**	**6.29%**	**15.00%**

Table 5.2: Data alignment. Scores for our main model for our validation patient with different data alignment methods.

5.5 Convolutional Approach

In Table 5.3 we report the results for our two convolutional models also in relation to our main model. Our first approach with the one by one exchange of dense layers for convolutional layers performs worse for the pairwise accuracy as well as for the direct classification scores in comparison to our main model. The second approach, where we replaced all the small ROI convolutional layers with one larger convolutional layer, scores better for both the pairwise accuracy and the direct classification scores than the first approach but still worse than our main

model. This does not support the hypothesis that convolutional layers are better suited for larger inputs and could help to extract local spatial features from the input vectors. Additionally, it becomes evident that applying the convolutional layers to the smaller ROI region inputs instead of the combined input hinders the model rather than improving it. This stands in direct contrast to the dense layer approach, where it was the opposite way round.

Model	Pairwise	Top-1	Top-5
Main model	**0.8637**	**6.29%**	**15.00%**
+ Convolution with ROI layers	0.8355	4.34%	12.17%
+ Convolution no ROI layers	0.8423	4.81%	12.75%

Table 5.3: Convolutional approach. Scores for our validation subject for our two convolutional models in comparison to our main model.

Chapter 6

Bridging fMRI to Language Generation

Next, we present an approach for combining information from the fMRI scans with a powerful language model. In particular, we generate text conditioned on the fMRI data using GPT-2 (Radford et al., 2019).

On the one hand, we use our fMRI classification decoder to transform a brain scan into a probability vector over our vocabulary of 180 words. We select the top 5 predictions, $w_1, w_2, ..., w_5$, and calculate their GloVE embeddings. We will use these embeddings as anchor points for the language generation model. On the other hand, the GPT-2 model autoregressively generates text, i.e., from previously generated context. To generate a new word, it processes the past context and produces a vector of probabilities over the whole vocabulary, then it samples the next word from the top-k words with the highest probabilities. We denote this vector of probabilities by $\vec{p} = \{p_1, p_2, ..., p_m\}$, where p_i is the probability corresponding to token t_i from the GPT-2 vocabulary and $m = 50257$ is the size of the GPT-2 vocabulary.

To guide language generation, we modify this vector of probabilities. We adjust the next token prediction scores $\vec{p}$ based on the cosine distance in the GloVE space between each word in the GPT-2 vocabulary and the anchor points. We use the GloVE embeddings to give a common space where the words decoded from the fMRI scan and the GPT-2 token predictions can be compared. It is important to notice that 3382 tokens of 50,257 do not have a matching GloVE embedding. Those tokens correspond to uncommon words or word snippets and their probabilities were kept unchanged. For all the other tokens the additional cosine distance term guides the next token generation towards the anchor points. The adjusted scores $\vec{p'}$ are calculated as

$$p'_i = p_i + k \sum_{j=1}^{5} \cos\left(\gamma(t_i), \gamma(w_j)\right) \tag{6.1}$$

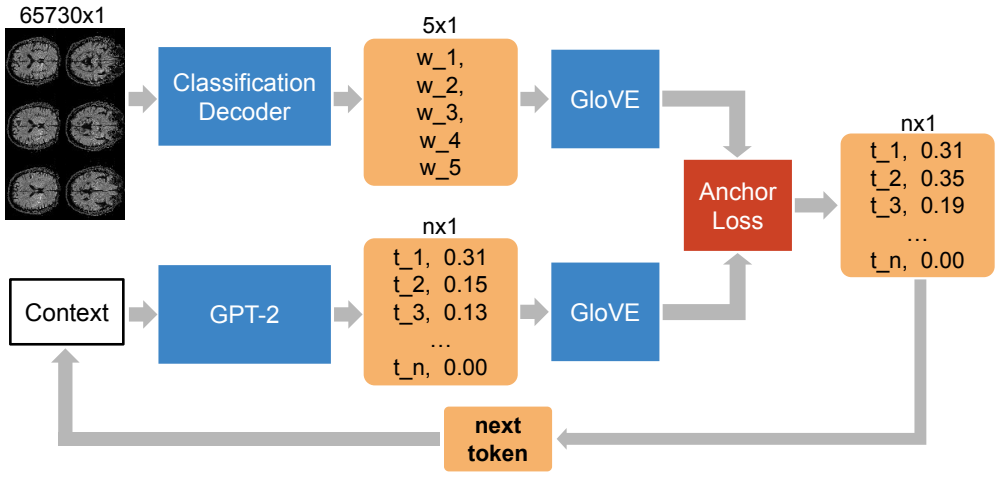

Figure 6.1: Architecture of the conditioned language generation model. Our classification decoder finds anchor points for the language generation model (GPT-2), which generates the next token given some initial context.

Where t_i is the i$^{\text{th}}$ token of the vocabulary, γ denotes the GloVE embedding and k is a hyperparameter controlling how heavily the scores are influenced by the anchors w. Finally, GPT-2 samples the next token from the top-k tokens with the highest updated and renormalized probability score. The process repeats with the new token added to the context. Our complete model is illustrated in Figure 6.1.

We emphasize that this is a general approach for conditioning language generation models on external input. Anchor points from any upstream model can be used to steer language generation. Moreover the generative model can also be replaced. Our model uses GloVE embeddings to connect the two parts together, but again any other word embedding scheme could be adopted.

6.1 Results

Now, we evaluate the end-to-end application of our fMRI language generation model. We take fMRI images as the input to the brain decoder. In this proof-of-concept we restrict our experiments to the fMRI scans where the classification decoder has a correct Top-5 prediction, that is the correct word appears in the top 5 words. We use a dataset of 40 brain scans for test and 10 for validation, which we use to tune the value of the anchor term k; we settle on $k = 7.0$. For the language generation we use the GPT-2 *small* model inputting snippets from the Harry Potter books (Davis, 2018) as initial context. The snippets are made of 2 consecutive sentences randomly selected and to avoid topic-specific content

we filter out snippets with proper nouns.

First, we present an objective evaluation of the fMRI-conditioned language generation model. We use perplexity to quantify the fluency of the text and relevant word count as a measure of the success of the conditioning. The perplexity is calculated based on the direct output of the GPT-2 model, before the anchoring term is added. This ensures that our evaluation of perplexity is not biased due to the increase of the token probabilities through the anchor term. The word count represents how well the generation is guided towards the semantic content of the fMRI scan. We count both, the number of occurrences of the correct word corresponding to the fMRI scan, as well as the number of occurrences of the 10 nearest neighbours in the GloVE space. We perform 10 runs per brain scan with the same random 10 initial contexts for each fMRI scan. In each run we generate 30 tokens, which is approximately 1 or 2 sentences. As a comparison we take the vanilla GPT-2 predictions with no conditioning on the fMRI data and evaluate the same metrics.

Model	Perplexity	Word Count	Rel. Words
GPT-2	89.24	0.00	0.04
+ anchoring	**50.64**	**0.52**	**1.02**

Table 6.1: Comparison of language generation with and without conditioning on fMRI brain scans.

The results can be seen in Table 6.1. As expected, the vanilla GPT-2 predictions have a 0.00 average word count and an average related words count 0.04: with a vocabulary size of $m = 50,257$ it is highly unlikely that the generated text contains the desired word and so, this serves to set the random baseline. With anchoring the average word count increases to 0.52 and the related words to 1.02, which is significant since we only generate 30 words, and demonstrates the success of our guided generation approach. Moreover the fluency of the generated texts does not appear to suffer, in fact the average perplexity score improves compared with the benchmark. We hypothesize that anchoring helps the generation to revolve around a reduced set of topics, reducing the chances of generating low probability words.

For a qualitative analysis, we present an example of text generated by our model in Table 6.2. To study how the anchors affect the generation, we compare it to text generated without anchoring.We see that the model produces coherent text and that the target word does appear. However, no punctuation tokens are generated and although this is not a big issue given the length of the text, it shows a direction to improve our conditioning strategy. Also, note that bigger generation models would improve the quality of the generated text.

Context:
“Everyone stand by a broomstick. Come on, hurry up."
Anchor:
level, picture, sign, mechanism, device
With anchoring:
*If the team is not up to the same **level** as the other team then they will have a hard time even if they work up a number of good*
Without anchoring:
"That’s a DMT / If you can do that, your masters can do that too." "TURNING MAIN

Table 6.2: Comparison of language generation with and without conditioning on fMRI brain scans. The context is a snippet from Harry Potter and the anchor words are the Top-5 predictions from our fMRI decoder. The correct word corresponding to the fMRI brain scan is emphasized in boldface.

6.2 GPT-2 Finetuning

In our approach we opted to go for text snippets from the Harry Potter books for our context of generating text. This is why we decided to also go one step further and apply finetuning to the GPT-2 model with the Harry Potter books. This way the text generation should be focused more on the Harry Potter topic, from where the text snippets for the context originate. This worked fine when just generating regular text but came with major drawbacks when trying to include our anchoring procedure. If the anchoring words did not have a direct relation to the Harry Potter theme, they would not show up without increasing the strength of the anchoring (parameter k in Equation 6.1) to unseen heights. Values that high enforce the anchoring to such an extent that for most cases only one of the anchoring words will show up repeatedly during text generation. Overall, we can therefore see that finetuning can help our approach for topic related words but hurts our anchoring procedure in most cases and is therefore not a feasible method to follow up on.

6.3 Discussion

The model presented in this section serves as a proof-of-concept for the application of fMRI decoding to language generation. The reason behind our choice of conditioning on the Top-5 decoded words is the following: the model tends to generate text in unpredictable directions, therefore, the four "incorrect" words from the Top-5 can be assimilated to these random directions without overshad-

owing the effect of the correct anchor. This way, we cover a larger amount of cases while still conditioning towards the correct topic.

Our experiments show that the output of fMRI decoding can guide language generation without loss in fluency. However, to enable real-time fMRI-to-text decoding some improvements are necessary, apart from further improving fMRI-to-word decoding. For instance, in order to account for the delay in fMRI scans due to blood flow, it would be desirable to have a measure of certainty for the decoded word which triggers language generation when the decoder is certain and halts it otherwise. Also it would be necessary to record a new dataset tailored to this application that covers the most important and general concepts a person may want to express, such as "positive", "negative", "happiness", "nature", etc. We are well-aware that there is still a long path before we can reliably turn thoughts into words, for example for coma patients. Nevertheless, we believe that our work provides new tools and ideas to make this possible one day.

Chapter 7

Conclusion

In this work we have presented a model to decode fMRI scans into words that outperforms existing models by a big margin. Furthermore, we have shown that a more realistic task is necessary to understand the performance of decoding models and to this end we propose direct classification. We have run our experiments on the extremely demanding scenario where no data from the target subject is available at training time and demonstrated that our model successfully generalizes to unseen subjects. Based on the results obtained by our decoder, we introduce a strategy for conditioning language generation towards the semantic content of fMRI scans. This way, we contribute towards a real system for translating brain activities to coherent text.

Bibliography

Bulat, L.; Clark, S.; and Shutova, E. 2017. Speaking, seeing, understanding: Correlating semantic models with conceptual representation in the brain. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 1081–1091.

Chen, T.; and Guestrin, C. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 785–794.

Davis, S. W. 2018. Harry-Potter-Book-Text-Generator. https://github.com/HvyD/Harry-Potter-Book-Text-Generator/tree/master/data.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* .

Gauthier, J.; and Ivanova, A. 2018. Does the brain represent words? An evaluation of brain decoding studies of language understanding. *arXiv preprint arXiv:1806.00591* .

Gauthier, J.; and Levy, R. 2019. Linking artificial and human neural representations of language. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 529–539.

Gordon, E. M.; Laumann, T. O.; Adeyemo, B.; Huckins, J. F.; Kelley, W. M.; and Petersen, S. E. 2016. Generation and evaluation of a cortical area parcellation from resting-state correlations. *Cerebral cortex* 26(1): 288–303.

Handjaras, G.; Ricciardi, E.; Leo, A.; Lenci, A.; Cecchetti, L.; Cosottini, M.; Marotta, G.; and Pietrini, P. 2016. How concepts are encoded in the human brain: a modality independent, category-based cortical organization of semantic knowledge. *Neuroimage* 135: 232–242.

Huth, A. G.; De Heer, W. A.; Griffiths, T. L.; Theunissen, F. E.; and Gallant, J. L. 2016. Natural speech reveals the semantic maps that tile human cerebral cortex. *Nature* 532(7600): 453–458.

Just, M. A.; Cherkassky, V. L.; Aryal, S.; and Mitchell, T. M. 2010. A neurosemantic theory of concrete noun representation based on the underlying brain codes. *PloS one* 5(1): e8622.

Kivisaari, S. L.; van Vliet, M.; Hultén, A.; Lindh-Knuutila, T.; Faisal, A.; and Salmelin, R. 2019. Reconstructing meaning from bits of information. *Nature communications* 10(1): 1–11.

Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; and Stoyanov, V. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692* .

Mitchell, T. M.; Shinkareva, S. V.; Carlson, A.; Chang, K.-M.; Malave, V. L.; Mason, R. A.; and Just, M. A. 2008. Predicting human brain activity associated with the meanings of nouns. *science* 320(5880): 1191–1195.

Muttenthaler, L.; Hollenstein, N.; and Barrett, M. 2020. Human brain activity for machine attention. *arXiv preprint arXiv:2006.05113* .

Nastase, S. A.; Liu, Y.-F.; Hillman, H.; Norman, K. A.; and Hasson, U. 2020. Leveraging shared connectivity to aggregate heterogeneous datasets into a common response space. *NeuroImage* 116865.

Nishimoto, S.; Vu, A. T.; Naselaris, T.; Benjamini, Y.; Yu, B.; and Gallant, J. L. 2011. Reconstructing visual experiences from brain activity evoked by natural movies. *Current Biology* 21(19): 1641–1646.

Palatucci, M.; Pomerleau, D.; Hinton, G. E.; and Mitchell, T. M. 2009. Zero-shot learning with semantic output codes. In *Advances in neural information processing systems*, 1410–1418.

Pennington, J.; Socher, R.; and Manning, C. D. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 1532–1543.

Pereira, F.; Lou, B.; Pritchett, B.; Ritter, S.; Gershman, S. J.; Kanwisher, N.; Botvinick, M.; and Fedorenko, E. 2018. Toward a universal decoder of linguistic meaning from brain activation. *Nature communications* 9(1): 1–13.

Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; and Sutskever, I. 2019. Language models are unsupervised multitask learners. *OpenAI Blog* 1(8): 9.

Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; and Liu, P. J. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683* .

Schroff, F.; Kalenichenko, D.; and Philbin, J. 2015. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 815–823.

Schwartz, D.; Toneva, M.; and Wehbe, L. 2019. Inducing brain-relevant bias in natural language processing models. In *Advances in Neural Information Processing Systems*, 14123–14133.

Sun, J.; Wang, S.; Zhang, J.; and Zong, C. 2019. Towards sentence-level brain decoding with distributed representations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 7047–7054.

Toneva, M.; and Wehbe, L. 2019. Interpreting and improving natural-language processing (in machines) with natural language-processing (in the brain). In *Advances in Neural Information Processing Systems*, 14954–14964.

Van Den Oord, A.; Vinyals, O.; et al. 2017. Neural discrete representation learning. In *Advances in Neural Information Processing Systems*, 6306–6315.

Van Uden, C. E.; Nastase, S. A.; Connolly, A. C.; Feilong, M.; Hansen, I.; Gobbini, M. I.; and Haxby, J. V. 2018. Modeling semantic encoding in a common neural representational space. *Frontiers in neuroscience* 12: 437.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *Advances in neural information processing systems*, 5998–6008.

Wang, S.; Zhang, J.; Lin, N.; and Zong, C. 2020. Probing Brain Activation Patterns by Dissociating Semantics and Syntax in Sentences. In *AAAI*, 9201–9208.

Wehbe, L.; Murphy, B.; Talukdar, P.; Fyshe, A.; Ramdas, A.; and Mitchell, T. 2014. Simultaneously uncovering the patterns of brain regions involved in different story reading subprocesses. *PloS one* 9(11): e112575.