The GRU and Transformer explaining Human Behavioural Data represented by N400.

Bachelor Thesis

International Business Communication

Radboud University

Thesis mentor: Danny Merkx

Ellen van den Boogaart

s4787455

Radboud University

Ellen van den Boogaart
S4787455
Google’s Transformer versus human language process
Radboud University

Abstract

Recurrent Neural Networks (RNN) are a popular type of neural network which are effective at processing language. The Gated Recurrent Unit (GRU) is a well known network that often outperforms other RNNs. Recently, a new neural network architecture has been introduced; the Transformer. In this investigation, the GRU and the Transformer are compared in their ability in predicting human sentence processing. The human language processing data is provided by Electroencephalography (EEG) measuring brain activity. This study investigates whether the GRU and Transformer differ in predicting human language processing measured by EEG. The language models of both types were trained to increase their language model accuracy. The language models compute surprisal values on a corpus of English sentences. This gives us surprisal values of different levels of how accurate the model is in capturing linguistic patterns. The surprisal values are compared to the human data given by the EEG experiment on the same corpus. A mixed linear model and a Generalized Additive Model (GAM) are used to compute the goodness-of-fit between the models and the human data, and its confidence interval for each human data set with the surprisal values. The findings show that the GRU and Transformer differ significantly in predicting human language processing data; the Transformer shows higher goodness-of-fit scores for the vast majority of the training. This implies that the Transformer outperforms the GRU as cognitive model.

keywords: neural networks, Google Transformer, n400, surprisal, GRU.

1. Introduction

Language is a complex and interesting form of data. Processing language is a complicated yet useful skill. A neural network is an example of a computational model able to perform natural language processing tasks. To efficiently perform these tasks on sentences, a network should know some rules of syntax, semantics and pragmatics (Kidd, 2018). Hansen and Salamon (1990) describe a neural network as a pattern recognition device, trained under supervision by demonstrating input and output pairs. The Recurrent Neural Network (RNNs), which is a common used neural network structure, has its strength in processing sequential data (Graves, Mohamed, & Hinton, 2013). Since language is a sequential signal, RNNs are used for language related tasks like machine translation, natural language processing tasks, speech recognition, handwriting recognition and time series analysis (Salvaris, Dean, & Tok, 2018). Although these neural networks seem to be far away from our day to day life, they are common in many applications. Some examples are the acoustic modeling of the speech recognition from your navigation system, the machine translation for translating a word, sentence or text (Zaremba, Sutskever, & Vinyals, 2015; Mikolov, Karafiát, Burget, Cernocký, & Khudanpur, 2010), as well as traffic flow prediction (Fu, Zhang, & Li 2016). These are just some of the many modern communication tools we use on a daily basis. Since computational systems are able to perform tasks we as humans do as well, they might serve as a model of human cognition. If language models are able to perform these tasks in a similar way as humans, it could also give us insight into our own language processing. Collobert and Weston (2008), for example, mention six aspects of language processing that can be done by neural networks; Part-of-Speech Tagging, Chunking, Named Entity Recognition, Semantic Role Labeling, Language Models and Semantically Related Words. These tasks are elements of human sentence processing. Overall, we are able to communicate using syntax, semantics and pragmatics. Ferrer i Cancho and Solé (2001, p. 2261) state that:

> *"Human language allows the construction of a virtually infinite range of combinations from a limited set of basic units. The process of sentence generation is astonishingly rapid and robust, and indicates that we are able to rapidly gather words to form sentences in a highly reliable fashion."*

According to Ferrer i Cancho and Solé (2001) we communicate through sentences of interacting words. In creating these sentences, we pick words from the mental lexicon that are most frequent and highly probable under the current context. Estimating the probability of a word in context is called next-word prediction. According to Crystal and House (1990), next word prediction plays a key role in human language processing. It gives a probability based on the previous words of the sentence. From the probability of a word given by next-word prediction, it is possible to compute a measure called surprisal. Surprisal is the extent to which a word could be expected given its prior context. Simply put, surprisal is the extent to how (un)expected a word is in a particular sentence. A word with a high probability is highly expected and would be assigned a low surprisal and vise versa (Aurnhammer, 2018). We can compare the surprisal of the neural network with human behavioural data, like brain activity.

Previous studies already compared several types of neural networks on their fit with human language processing. Aurnhammer and Frank (2018) investigated whether there is a difference between three different types of RNNs, namely; Simple Recurrent Networks (SRNs), Gated Recurrent Unit (GRU) and Long Short-Term Memory (LSTM). The difference between these networks is that the GRU and LSTM have an implemented gating mechanism that controls the flow of information and thus allows the cells to memorise and/or forget information over time. The LSTM and GRU have already proven itself in outperforming simple recurrent networks (SRNs) in several natural language processing tasks; on number agreement (Linzen et al, 2016) and conversational speech recognition (Xiong et al., 2017). Even though the LSTM and GRU reach a higher language model accuracy than the SRN in Aurnhammer and Frank's investigation (2018), the results show that there was no significant difference between the RNN types in terms of how well they explained the behavioural data. This means that neither the LSTM nor GRU performed significantly better as cognitive models of sentence reading than the SRN (Aurnhammer & Frank, 2018). Additionally, there was no significant difference between the LSTM and GRU.

Recently, a new network architecture was introduced by Vaswani, Shazeer, Parmar, Uszkoreit, Jones, Gomez, Keizer, & Polosukhin (2018) and is called the Transformer . The architecture is based on attention mechanisms and is thus distinctly different than the RNN architectures, which makes it interesting to investigate this network. Vaswani et al. delineate the Transformer as; "a model architecture eschewing recurrence and instead relying entirely on an attention mechanism to draw global dependencies between input and output." (Vaswani et al, 2018, p.2). When conducting this study, to the best of my knowledge, this new network has not been tested yet regarding surprisal ratings in correlation with human processing data.

As human processing data I used data collected by Frank et al (2015). With Electroencephalography (EEG), brain activity was measured by picking up potential differences on the scalp surface. This is done with the use of electrodes attached to the scalp of a person (See image 1). On the basis of EEG, an event related potential response (ERP), like N400, can be recorded.

> *"We emphasize the effectiveness of the N400 as a dependent variable for examining almost every aspect of language processing and highlight its expanding use to probe semantic memory and to determine how the neurocognitive system dynamically and flexibly uses bottom-up and top-down information to make sense of the world.*"
>
> (Kutas & Federmeier, 2011, p. 621).

The EEG, ERP and N400 will be explained in further detail in the theoretical framework (see section 2.2). For now, it is important that the N400 is related to the semantic integration of stimuli (Kolb & Whishaw, 2000), and will thus show a high amplitude for an unexpected word. By comparing the N400 amplitudes with the surprisal ratings from the neural networks, we can learn more about how humans process language.

In this bachelor thesis I compare the Transformer architecture and the GRU on their ability to predict brain activity, to answer the following research question: Do the Transformer and the GRU network differ in predicting human language processing measured by EEG?

Image 1. EEG (Peters & Chaves, n.d.)

2. Theoretical framework

2.1 Recurrent Neural Networks

As explained in the introduction, neural networks are computational models that are used in natural language processing tasks. The Linear Neural Network is a basic network, while the Recurrent Neural Network (RNN) is more advanced. The difference between these two network types can be seen in Figure 1; which shows (a) the architecture of a Linear Neural Network and (b) a visual illustration with the unfolded structure of the model of the RNN. As displayed in the Figure 1b, the RNN takes information from the previous input into account, while the linear network computes the output estimation (y) by only passing the information of the input (x) to the next layer. It is important to know that the RNN computes the hidden state sequence ($\mathbf{S}_{t-1}$, …, S_{t+1}) and the output vector sequence ($o_{t-1}, \ldots, o_{t+1}$) by following equations (1 and 2) at time step 't' (Graves et al, 2013):

$$s_t = \sigma(\mathrm{Ws}_{t-1} + \mathrm{Ux}_t) \qquad (1)$$

Let's dig deeper into the RNN structure (Figure 1b). To start, we represent the input ($\mathbf{x}_t$) as vectors. Each vector is multiplied with a weight matrix (**U**), added to the hidden weight matrix (**W**) with the hidden state ($\mathbf{S}_t$) from the previous input and passed through a sigmoid activation function (σ) (equation 1). With the hidden state output ($\boldsymbol{o}_t$), the softmax function of the hidden state output over time (Vs_t) (equation 2), we are able to calculate a probability distribution of the lexicon by using a feed-forward layer.

$$o_t = softmax(\mathrm{Vs}_t) \qquad (2)$$

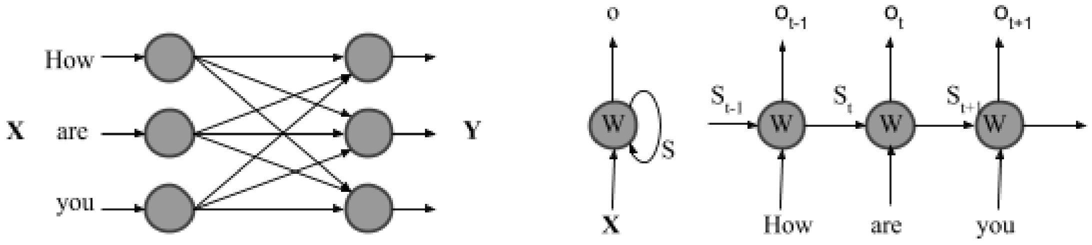


Figure 1. a. Linear Neural Network b. Recurrent Neural Network

To calculate the surprisal value, the next-word probability is used. Surprisal is inversely related to the probability of the particular word and is thus computed as the negative log-probability of the next word given the previous words (Levy, 2008; Aurnhammer & Frank, 2018):

$$\textit{Surprisal}\ (w_t) = -\ \log P(w_t \mid w_1,\ \ldots,\ w_{t-1}) \tag{3}$$

"[Surprisal] defines the word-based measure of cognitive effort in terms of the prefix-based one." (Hale, 2001, p. 4). Easily put, the surprisal rate is high for a particular word (w_t) when it is unexpected (low probability) given its previous context and vise versa. In the following sentences, an example is given of (1) an expected word, with a low surprisal rate, and (2) an unexpected word, with a high surprisal rate.

1. *Every evening, my mom cooks <u>dinner</u>*
2. *Every evening, my mom cooks <u>grass</u>*

2.1.1 Vanishing Gradient Problem

Aurnhammer (2018) states that the RNN's advantage is the ability to take the entire sequence of the prior words into account to predict the next word. A popular recurrent neural architecture is the Simple Recurrent Network (SRN). However, the SRN has difficulties with decay of information in the learning method due to 'vanishing gradient problem' (Hochreiter, 1998; Aurnhammer & Frank, 2018). The basic idea of this problem is that the language model has difficulties learning which information from past input has to be stored to create the desired output (Hochreiter, 1998).

2.1.2 The GRU and LSTM

The vanishing gradient problem is addressed by the Gated Recurrent Unit (GRU) and the Long Short-Term Memory (LSTM), which contain gates with trained weights. These gating mechanisms are able to control the flow of information to make sure that the cells can memorise or forget adequately certain information over time (Aurnhammer & Frank, 2018). Due to this, the network becomes more accurate in encoding long distance dependencies (Bahdanau, Cho, & Bengio, 2015). Since the study of Aurnhammer and Frank (2018) showed no significant difference between the SRN, LSTM and GRU in performing as cognitive model of sentence processing, I will only use the GRU. Previous studies shown that the GRU is comparable to the LSTM and even out performs it in some tasks (Dey & Salemt, 2017); like polyphonic music modeling and speech signal modeling (Chung, Gulcehre, Cho, & Bengio, 2014). "The GRU can be regarded as a more lightweight variation on the LSTM, making use of only two gates and a single hidden state, whereas the LSTM architecture provides three gates and introduces an additional memory state." (Aurnhammer & Frank, 2018, p. 1). The GRU network has also shown to be successful in performing sequential task over long distance (Dey & Salemt, 2017).

2.1.3 The GRU and the Transformer

While the GRU and the LSTM are rather similar, there is a major difference between the GRU and the Transformer. One important difference is that the GRU receives the information of the input in a indirect way as a set of hidden states passed on through the sequence of the input. While the Transformer is able to pay attention to every previous single aspect of the input in a direct way. As exemplified in Figure 2b, with the Transformer, the word 'dogs' has direct access to information about 'my', 'mom' and 'loves'. With the GRU, the word 'dogs' only receives indirect information, passed on through 'loves', about 'my' and 'mom'.

The architecture of the recurrent cell of the GRU is delineated in Figure 3 with the architecture of recurrent cell of the Transformer next to it in Figure 4. Although the Transformer seems like a sequential system, the input is converted and processed as a unit. The GRU on the other hand, processes every word-embedding one by one in the recurrent layer cell combined with the hidden state of the previous word. In a sentence with 35 words, for example, the last word receives only indirect information about the first word and is thus not able to 'look' at this word directly. This is where the GRU and Transformer network differ in processing their input.

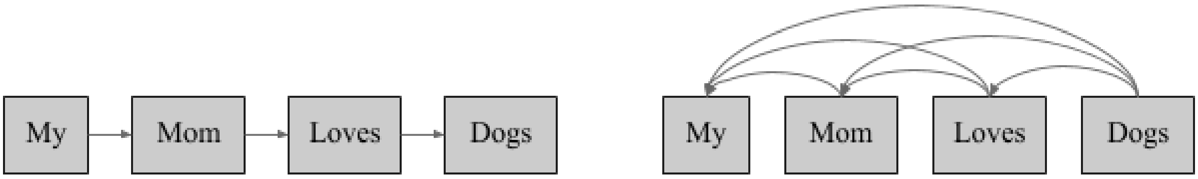


Figure 2. Flow of information of (a) GRU (b) Transformer

The hidden state of an input in the GRU (see figure 4) is computed as:

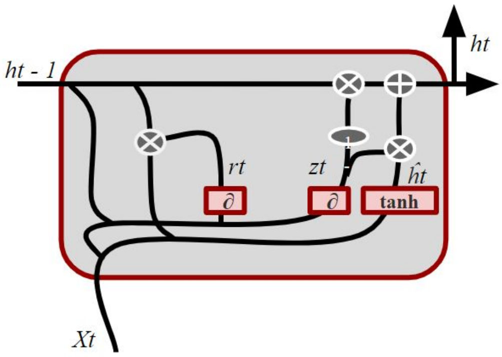
$$Z_t = \sigma\ (\ \mathrm{W_z}\ \ [\mathrm{h_{t\text{-}1}},\ \mathrm{X_t}]) \tag{4}$$

$$R_t = \sigma\ (\ \mathrm{W_R}\ [\mathrm{h_{t\text{-}1}},\ \mathrm{X_t}]) \tag{5}$$

$$\check{h}_t = \mathrm{tahn}(\ \mathrm{W}.\ [\ \mathrm{R_t}\ \times\ \mathrm{h_{t\text{-}1}},\ \mathrm{X_t}]) \tag{6}$$

$$h_t = (\ 1 - \mathrm{Z_t}\ )\ \times\ \mathrm{h_{t\text{-}1}} + \mathrm{Z_t}\ \times\ \check{\mathrm{h}}_\mathrm{t} \tag{7}$$

Where $\mathrm{Z_t}$ and $\mathrm{R_t}$ are the sigmoid activations (σ) of the previous hidden state ($\mathrm{h_{t\text{-}1}}$) and input ($\mathrm{X_t}$) multiplied by the weight matrix ($\mathrm{W_z}$ and $\mathrm{W_r}$). Then, $\check{h}_t$ is the tahn activation of the product of $\mathrm{R_t}$, previous hidden state($\mathrm{h_{t\text{-}1}}$), and input ($\mathrm{X_t}$) multiplied by the weight matrix (W.) h_t is the next hidden state computed as 1-$\mathrm{Z_t}$ multiplied by the previous hidden state + $\mathrm{Z_t}$ + $\check{h}_t$. The hidden state can be seen as the model's memory. In the model, it gets passed on to the processing of the next word.



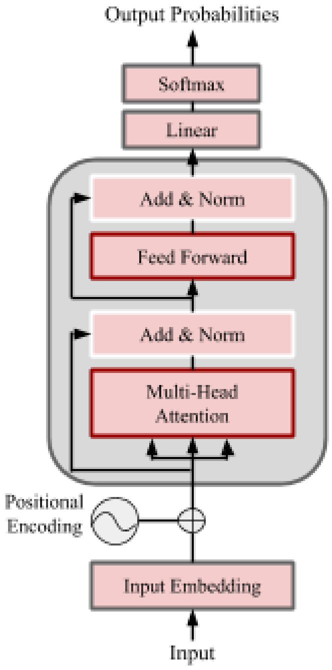


Figure 3. GRU architecture

Figure 4. Transformer architecture

To predict the next word in a sentence, the network looks at the previous words. However, some previous words are more important than others. The importance of the words is not directly related to their position. The penultimate word, for example, is not always most relevant. However, in a GRU, the information is passed throughout the network indirectly in the form of a weighted hidden state. With the use of attention (see equation 8), the Transformer network is able to pay less attention to less important words and more to the important ones in a direct way.

$$\text{Attention (Q, K, V)} = \text{softmax}\left(\frac{QK^{T}}{\sqrt{d_{k}}}\right)\text{V} \tag{8}$$

Where the product of the query (**Q**) is computed with all keys of the transposed key matrix (**K**$^{\mathbf{T}}$) and divided by the square root of the key dimension (d_k). The output gets then multiplicated by the V matrix and converted by a softmax function. Vaswani et al (2018) describe the attention function as "a weighted sum of the values, where the weight assigned to each value is computed by a compatibility function of the query with the corresponding key." (p. 4).

Due to the MultiHead attention (see equation 9 and 10) of the query-, values-, and keys matrices, the importance of the relationship between the words is weighted. To make sure the sequence of the sentence stays intact, they implemented positional encodings in the input. The Multi-Head Attention allows the model to take information from different representation subspaces due to matrix multiplications (Vaswani et al, 2018):

$$\textit{MultiHead (Q, K, V)} = \text{Concat}(\text{head}_1, \ldots, \text{head}_h)W^O \tag{9}$$

$$\text{where head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V) \tag{10}$$

By using multiple heads, the features of the weight matrix are divided over *n* matrices. Each head computes its own attention output, which are all concatenated (Concat) as the final attention output. Both the MultiHead Attention layer and Feed Forward layer are followed by a residual connection (Add & Norm). Here, the input is added (Add) to the attention output without going through the attention layer. Because of this, the network is able to look directly at the 'original' input. Additionally, it normalises (Norm) the data, of the now residual output, have a 0 (zero) mean and a unit (1) variance. Finally the residual output goes through the Feed Forward layer and another Add & Norm layer, and computes the final Transformer output. The Transformer is originally meant as encoder/decoder structure (see for complete architecture Vaswani et al., 2018). However, since it is not necessary to decode in this study, I only use the encoder. The designers of this architecture explain the attention function as "mapping a query and a set of key-value pairs to an output, where the query, keys, values and output are all vectors." (Vaswani et al, 2018). The compatibility function of the query with the corresponding key computes the weight assigned to each value. The sum of these values is computed as the output. Since these matrix computation can be done in parallel, this architecture is fast. Furthermore, there are less steps in which information can be lost. If the Transformer has a better fit with the human language processing data than the GRU, it could be that our brain is more likely to pay direct attention to single aspects at any previous position in a sentence. This gives us more information about how our brain processes language and how we could construct networks for applications that need to work as similar as possible with our brain.

2.2. Electroencephalography (EEG)

The human language processing data in this study is brain activity measured by Electroencephalography (EEG) by Frank et al (2015). EEG is used to measure brain activity by detecting potential differences on the scalp surface. In their book, Kolb and Whishaw (2000) explain that by attaching electrodes on the scalp, one is able to produce a record (nowadays on the computer) showing electrical activity waves of the brain (see image 2). EEG is mainly used in research and for medical reasoning regarding brain functioning, sleep stages, coma, head injuries and other brain (ab)normalities. A change in the brain activity due to a certain stimulus is called Event related potential response (ERP). ERP is the average of a multiple EEG records in a specific time frame responding to a discrete sensory stimulus. This stimulus can be written words, sound, pictures, etc. By using more than just one response record (see image 3), one is able to average recordings together and represent a distinctive wave pattern as a clear line with a number of positive and negative waves in a timelapse of a few hundred milliseconds (Kolb & Whishaw, 2000). This distinctive wave pattern (see image 4), distincts from the baseline activity. The baseline is the brain activity measured by the EEG without manipulating it by using a stimuli.

One of these waves of the ERP is the N400 ('N' stands for negative), which is used in this study. Lieberman (2015) explains the ERP effects including the N400 as follows; The brains first reaction, on sound for example, is a combination of the N100 and P200 that occur in the first 200 milliseconds after the stimulus. If the sounds over a time are similar, think of the same first letter or syllable, the effect will be low. The ERP effect will be high if the sound is rather new or different. Between 300ms and 500ms the N400 shows, this effect reflects the cost of semantic integration. In simple words, how well you think the word fits in the sentence. Finally, you have the P600 curve that will react on the syntax and linguistic correctness of the sentence. In this study we use the N400 effect compared to the baseline of the EEG, since a correlation has been found between the N400 amplitude and word surprisal and next word prediction (Frank, Otten, Galli, & Vigliocco, 2015; Frank & Willems, 2017). According to Kutas and Federmeier (2011), is N400 a highly effective dependent variable in investigating aspects regarding language processing, meaning processing, and many more. In their article they give an overview of all the study areas on which N400 was used as measure tool, since it was invented almost 40 years ago (Kutas & Federmeier, 2011).

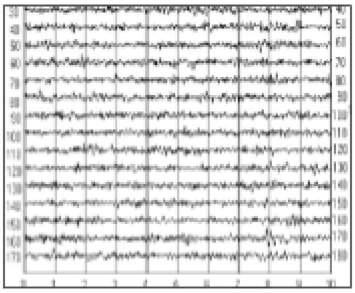


Image 2. Electronic brain activity waves (Poulos et al, 2003)

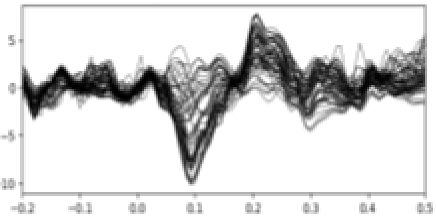


Image 3. Several response records (EEG processing [..], n.d.)

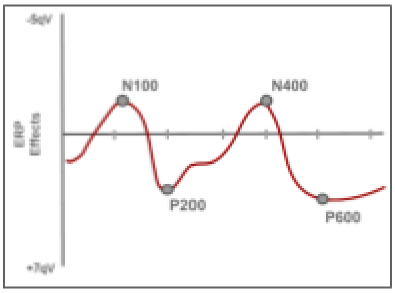


Image 4. ERP

3. Methodology

3.1. Neural Networks

As explained before, the GRU and Transformer were compared in the fit of their surprisal output with the human processing data. Since the neural networks start with random weights, which could influence the outcome, we train several models of each network type. Six different neural network models of each network type were trained. This gives us a total of twelve networks, that were trained and compared at the same stages in the training to see if they differ at any point in the training. I took 9 points; after 1K, 3K, 10K, 30K, 100K, 300K, 1M, 3M, and 6,47M sentences. Due to this, it is possible to compare the models at iqual language model quality levels.

3.1.1. Architectures

For the GRU network, we used the parameters used in the similar experiment from Aurnhammer and Frank (2018). The architecture of the GRU consisted of a 400-unit word embedding layer, a single 500-unit recurrent layer, a 400-unit feed-forward layer with tanh activation function and a output layer mapping to the lexicon with log-softmax activation function. The most important part for us is the 500-unit recurrent layer. This is where the model differs in being an GRU, LSTM or Transformer, depending on which cells you use. For the Transformer, we used a single Transformer cell with a 400 unit word embedding layer, 8 attention heads and a 1024 feedforward layer. The size of the Transformer was chosen to approximate the amount of weights in the GRU as closely as possible. For the Transformer and GRU, this model consists of the same word embedding layer, feed-forward layer with tanh activation function and output layer with log-softmax activation function. The softmax activation function converts the output of the networks into a probability distribution over the output lexicon (0 to 1). As mentioned by Aurnhammer and Frank (2018), pre-trained word embeddings are not used. The weights, which are random in the beginning, are adjusted and learned during the training task.

3.1.2. Network training

Since the weights of the neural networks are random in the beginning, the estimation will not be accurate at all in the beginning. To train the network, it needs training data. In this case it was provided a corpora of english sentences. To train the GRU, Aurnhammer and Frank (2018) used section 13 of the Corpora from the Web (COW, 2014 version; Schäfer, 2015) which consists of randomly ordered sentences from web pages. From this corpus, they took the 10.000 most frequent word types as the model’s vocabulary. To this vocabulary, they added 103 words that were not present in the 10.000 most frequent word types but were in the experimental stimuli (see 3.2.2). At this point, they only kept the sentences that contained the word types from their vocabulary of 10.103. Ultimately, since the maximum sentence length in the experimental stimuli was 39 words, sentences with more words were removed from the training corpora (punctuation was not counted as word). We use the final training corpora which consists of 6.470.000 sentences with 94.422.754 tokens in total (Aurnhammer & Frank, 2018). When comparing the actual next word to the estimated probability probability of the next word and computing the cross entropy loss between those, one can see how accurate the network is. According to the loss for all words, the network applies slight changes in the network’s weights. To compare the development of networks’ fit with the human data during the trainings, the surprisal ratings were taken at 9 points over time in every training repetition (after 1K, 3K, 10K, 30K, 100K, 300K, 1M, 3M, 6,47M sentences). This gives us a total of 9 (points during training) × 6 (initials per network type) × 2 (GRU and Transformer) = 108 sets of surprisal ratings .

3.2. Human Processing Data

The human processing data in this experiment contains the ERP amplitudes of the N400 (see section 2.2). This data is taken from the experiment by Frank et al (2015), where the ERP response was measured indicating the amount of information conveyed by words in a sentence.

Table 1. Data of participants and processing corpus

Participants	**Sentences**	**Range sent. length**	**Mean sent. length**	**Tokens**	**Total data points**
24	205	5-15	9.4	1931	46,344*

* 24 (participants) × 1931 (tokens) = 46,344 (Total data points)
*After excluding a set of data (see 3.3), 24,618 data points remained

3.2.1. Participants

The 24 participants were all native English speakers from the UCL Psychology subject pool, of which 10 female and 14 male participants with a mean age of 28 years (Frank et al, 2015).

3.2.2. Processing corpus

To measure the human processing data, the participants were asked to read British-English sentences. The corpus consisted of 205 sentences, containing 1931 word tokens, taken from the UCL corpus of reading times (Frank, Monsalve, Thompson, & Vigliocco, 2013). For this study, three British-English short novels were taken from a website for aspiring writers to publish their, otherwise unpublished, work (www.free-online-novels.com). With a list of most frequent word, a selection of 361 sentences was made. Finally, for an eye-tracking experiment, 205 sentences that fitted the display were left.

3.2.3 The EEG

While the participants read the sentences, the electrodes attached to their scalp measured the brain activity. The sentences were presented word by word on the participant’s monitor. The duration of the every word being shown equalled *190 + 20m* ms (*m* is number of characters of the word) (Frank et al, 2015). For 166 sentences, a comprehension question (yes/no) had to be answered. All participants answered at least 80% of these questions correctly. "The EEG signal was recorded continuously at a rate of 500 Hz from 32 scalp sites and the two mastoids relative to a mid-frontal site using silver/silver-chloride electrodes with impedances below 5 kΩ." (Frank et al, 2015, p. 3). As the N400 is an indicator for lexical, semantic and conceptual processing (Aurnhammer & Frank, 2018; Frank et al, 2015), the signals were filtered between 0.01 and 35 Hz (offline between 0.05 and 25 Hz) and allocated into trials of 100 ms before each word and 924 ms after each word (Frank et al, 2015).

3.3. Goodness-of-fit

The neural network predicting surprisal values that fits the human processing data more accurate, tend to be better language models and gives us information on how humans process language. To define the goodness-of-fit between the estimated surprisal and N400 response, I used a mixed linear models approach. As explained (in 2.2.) the N400 size is the difference of the ERP effect and the baseline of the brain activity between 300ms and 500ms. With this baseline, the effects of the most important variables, other than surprisal, influencing N400 (e.g. word length) got factored out (Aurnhammer & Frank, 2018). A small set of data got excluded, like the data from words with an initial position, a final position, attached to a coma, and clitics. Further, any peaks in the N400 data over 100 μ V got excluded as well. To measure the predictive power of the surprisal ratings, a mixed linear model by Statsmodels (Seabold & Perktold, 2010) is used. The goodness-of-fit of the N400 data equals the log-likelihood ratio between the baseline and a regression model, with surprisal as fixed effect and a by-subject random intercept.

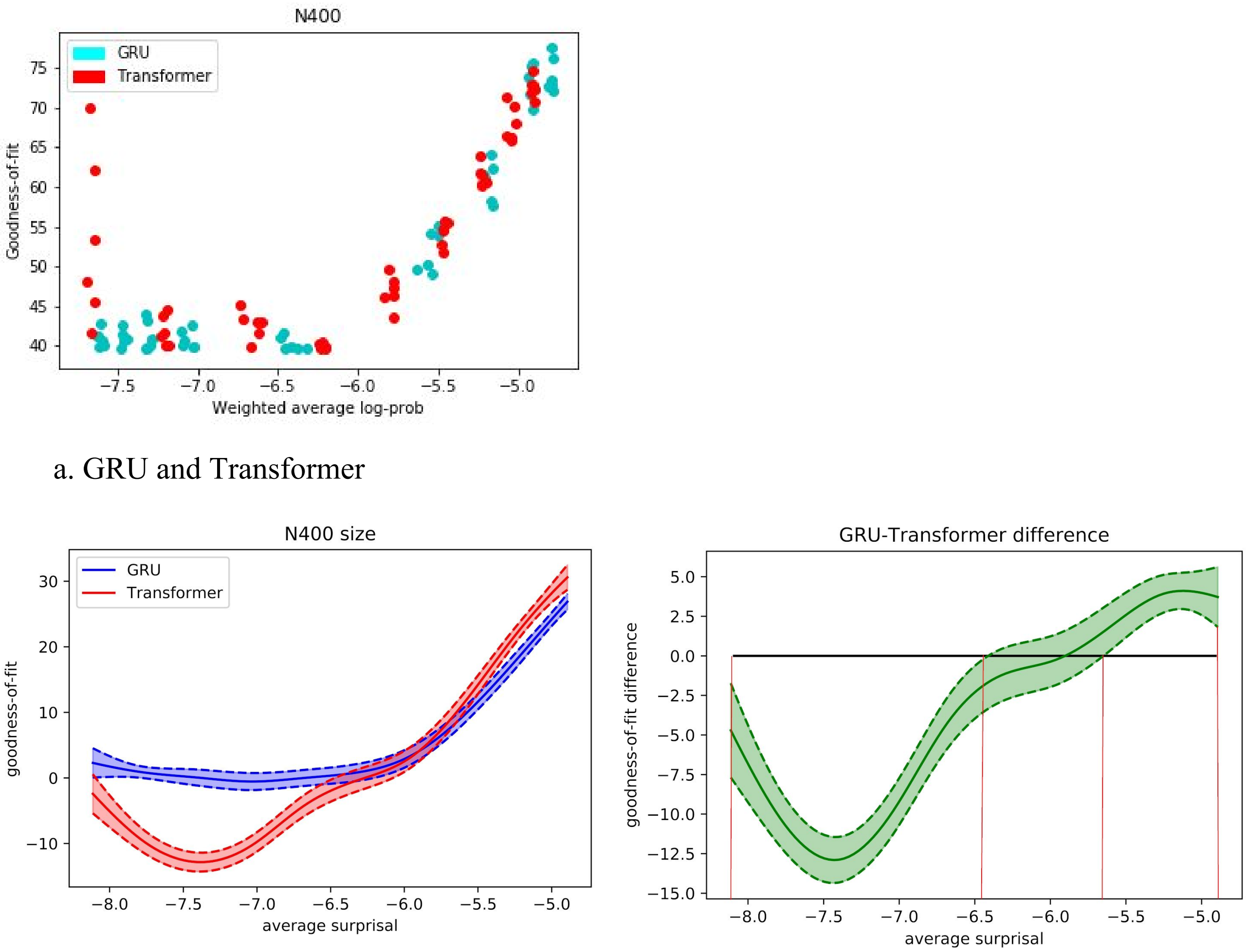
3.4 Mixed Linear Model

The mixed linear model is used to model the relationship between the surprisal ratings of our neural network models and the N400 data. Compared to a linear regression model, the mixed linear model has some advantages in this case. With the simple linear regression model, fixed effects and random slopes and intercepts are not taken into consideration. The mixed linear model makes sure that the results show the effect of the variable you are interested in, while accounting for known effects like word frequency. For this study, a mixed linear model of the results was fitted to the network's surprisal estimates and the N400 data including all fixed effects and a by-subject random intercepts. In this case, the fixed effects are the word length, the word frequency, the word position and baseline activity. To take possible differences between the participants into account, the by-subject random intercept and by-subject random slopes were added for all fixed effects. Additionally, the same test was done excluding the surprisal data, which we call the baseline model. It is important to note, this is another baseline than the one mentioned in the explanation of the EEG and N400 data. By subtracting the baseline model from the model, including the surprisal, we get the log likelihood ratio (i.e. goodness-of-fit).

I apply the Generalized Additive Model to the goodness-of-fit scores. This is a useful technique for measuring non-linear analysis over time (Dominici, McDermott, Zeger, & Samet, 2002). With the use of GAM, it is possible to transform a nonlinear function of non parametric functions into a generalized linear model. The strength of this method is the ability to process non-linear and non-monotonic relationships between the variables and the response and to compute a smooth model from this data (Guisan, Edwards, & Hastie, 2002). Because of its flexibility, compared to parametric techniques like the GLM, the GAM is popular in many research fields like weather trends, seasonality, genetics, epidemiology, medicine research, air pollution and many more.

4. Results

Figure 5a show the goodness-of-fit of each set human data with the surprisal values of all versions of the neural models over the training procedure. The average surprisal values of the models are given in the model as 'weighted average log-prob' and indicate the language model accuracy. The data in the figure equals the log likelihood ratio between the surprisal and the baseline model. This ratio displays the goodness-of-fit of the surprisal values to the human data set as a function of language model accuracy. The plot clearly shows that the goodness-of-fit improves as the neural network models proceed in the training procedure and thus acquire a higher level of language model accuracy.



a. GRU and Transformer

b. GAM fitted curves GRU and Transformer

c. Difference GRU and Transformer

Figure 5a shows the goodness-of-fit of each set of human data (N400) with the average surprisal values (higher is better) as the independent function.
Figure 5b displays the curves fitted of the GRU and Transformer by the Generalized Additive Model (GAM) using pyGAM.
Figure 5c displays the difference between the GRU and Transformer. The shaded areas, in b. and c., indicate the 95% confidence interval of this model.

In the beginning, with the neural networks only trained on a relatively small set of sentences, the language model accuracy of both the GRU as the Transformer is fairly low. With this level of language model accuracy, the goodness-of-fit is low as well. As can be seen in the figures displaying data from the Transformer, some network versions give a high goodness-of-fit score while having limited training (weighted average log-prob < -7.5). This could be due the fact that each network version holds random weights at the beginning of the training. Since these weights are not pre-trained or adjusted at the start, there is a possibility that the network overfits on the few examples it has seen and that these training examples generalize well to the training data. This possibility is kept in mind without excluding other potential factors. However, with the progression of the language models becoming well-trained and thus more accurate in capturing linguistic patterns, the goodness-of-fit clearly improves. Figure 5b displays the fitted curves by the GAM (using pyGAM) indicating the goodness-of-fit as a linear function model of both the GRU and Transformer. The 95% confidence interval is displayed as the shaded areas surrounding the fitted curves. With this confidence interval, we can see if the models differ in a significant way from each other at any given point of this data. In Figure 5b, at some parts of the curves, the goodness-of-fit of the GRU lies apart from the shaded confidence interval of the Transformer and vise versa. This implies that the language models differ significantly regarding the goodness-of-fit. In Figure 5c. the difference between the curves of the models is displayed. The zero line (black) is the assumption that there is no difference between the two curves. when the curve and its confidence interval (the shaded area) lie above or beneath the zero line, the difference is significant. Where the curve is situated beneath zero, the GRU has significantly higher goodness-of-fit scores than the Transformer. Additionally, where the curve is situated above zero, the Transformer scores higher regarding the goodness-of-fit. In this figure, we see that the GRU has significantly higher goodness-of-fit scores than the Transformer for the first sets of training sentences (average surprisal < - 6.4). This difference takes a turn when the Transformer starts to give significantly higher goodness-of-fit scores (average surprisal > - 5.7). The part where Figure 5b and 5c displays a significant difference in the advantage of the GRU might look like more than half of the training process. However, the GRU only reaches significantly higher goodness-of-fit scores until the fourth point of the training, which is till the 30K sentences and thus only covers 0.46% of the entire training corpora of 6.47 million sentences. On the other hand, the Transformer shows significantly higher scores after 300K sentences (point six of the training) which covers 95.4% of the training sentences.

5. Discussion and Conclusion

The comparison of the GRU and the Transformer revealed significant differences in terms of the ability in predicting human data. With these results, it is possible to answer the research question of this study; Do the Transformer and the GRU network differ in predicting human language processing measured by EEG? Since the GAM curves diverge from each others confidence interval, in the majority of the figure, the GRU and Transformer differ significantly. In the first part of the training process (till 30K sentences; 0.46%) the GRU showed higher goodness-of-fit scores, while the Transformer scored higher in terms of the goodness-of-fit for the obvious majority of the training process (from 300K sentences; 95.4%). These findings imply that the Transformer has a better fit with the human sentence processing data than the GRU. This might suggests that human sentence processing involves a mechanism more akin to the direct access to context of the Transformers rather than toe gating mechanism of the GRU.

In a similar study by Aurnhammer and Frank (2018), the GRU, LSTM and SRN were compared in the goodness-of-fit with human processing data. The difference between gated and non-gated recurrent network types was not significant, unlike the difference between the GRU and the Transformer. The fundamental difference between recurrent and attention based networks might explain why our results differ from the findings by Aurnhammer and Frank (2018). Although significant differences provide us with interesting insight on how we process language, getting insignificant results could be as relevant. Since the Transformer is a relatively new network architecture (2018), any research including this network might be relevant. By comparing the Transformer with other neural network architectures, we gather potential information to determine the position of the Transformer in the neural network field. This is useful for further research as well as for projects considering the use of a Transformer model.

In this case, the findings imply that our brain processes a sentence with directly paying attention to the previous words of the sentence. This might be a big game changer in several fields like; language acquisition, second language learning, language development of children, and other language related areas. It provides compelling opportunities for further research in these fields. In terms of the EEG part of this research, there are a lot of potential areas to investigate regarding the fields of communication, language and culture. Some examples could be; the differences between age or gender, differences between first language and second language, different stages in the language development of children. Furthermore, similar studies could be conducted with taking into account different languages and cultures. Arabic and Chinese, for example, have an entirely distinct alphabetic structure than Germanic languages.

For further research, in terms of the technical analysis, it should be considered that with using Statsmodels (Seabold & Perktold, 2010) it is not possible to include more than one random factor. In this study, I included the by-subject random intercepts and slopes as random variable. If a different module for conducting the statistical analysis would be used in further research, the by-item random intercept could be included. Furthermore, instead of using the data from Frank et al (2015), it might be valuable to collect EEG data in an experiment where subjects are presented complete sentences instead of a word by word paradigm to investigate a more natural reading scenario.

In summary, in this thesis I have shown that there are significant differences between the GRU and the Transformer with regard to predicting human behavioral data represented by N400. The Transformer showed a significantly higher goodness-of-fit score with the N400 size for the vast majority of the training. This might suggests that human sentence processing is more likely to be similar to the 'direct access to context' mechanism of the Transformer than to the gating mechanism of the GRU.

References

Aurnhammer, C., & Frank, S. L. (2018). Comparing gated and simple recurrent neural network architectures as models of human sentence processing. doi:10.31234/osf.io/wec74

Bahdanau, D., Cho, K. & Bengio, Y. (2015). Neural machine translation by jointly learning to align and translate. *Proceedings of the Third International Conference on Learning Representations.* San Diego, CA: International Conference on Learning Representations.

Bezanson, J., Edelman, A., Karpinski, S., & Shah, V. B. (2017). A fresh approach to numerical computing. SIAM Review, *59*, 65–98.

Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling, arXiv preprint arXiv:1412.3555.

Collobert, R., & Weston, J. (2008). A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, 160-167. ACM.

Crystal, T. H. & House, A. S. (1990). Articulation rate and the duration of syllables and stress groups in connected speech. *The Journal of the Acoustical Society of America*, *88*(1), 101–112

Dey, R., & Salemt, F. M. (2017). Gate-variants of Gated Recurrent Unit (GRU) neural networks. *IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS)*. doi:10.1109/mwscas.2017.8053243

Dominici, F., McDermott, A., Zeger, S. L., Samet, J. M. (2002). On the Use of Generalized Additive Models in Time-Series Studies of Air Pollution and Health, *American Journal of Epidemiology*, *156*(3), 193–203. https://doi.org/10.1093/aje/kwf062

EEG processing and Event Related Potentials (ERPs). [Image 3] (n.d.). Retrieved from https://martinos.org/mne/dev/auto_tutorials/plot_eeg_erp.html

Ferrer i Cacho, R., Solé, R. V. (2001). The small world of human language. *Proceedings of the Royal Society of London. Series B: Biological Sciences.* doi:https://doi.org/10.1098/rspb.2001.1800

Frank, S.L., Monsalve I. F., Thompson, R.L., Vigliocco. G. (2013). Reading time data for evaluating broad-coverage models of English sentence processing. *Behavior Research Methods*, *45*, 1182-1190

Frank, S. L., Otten, L. J., Gally, G. & Vigliocco, G. (2015). The ERP response to the amount of information conveyed by words in sentences. *Brain and Language, 140*, 1-11.

Frank, S. L. & Willems, R. M. (2017). Word predictability and semantic similarity show distinct patterns of brain activity during language comprehension. *Language, Cognition and Neuroscience, 32*(9), 1192–1203.

Friedman, J. H., & Stuetzle, W. (1981). Projection pursuit regression. *Journal of the American statistical Association*, *76*(376), 817-823.

Fu, R., Zhang, Z., & Li, L. (2016). Using LSTM and GRU neural network methods for traffic flow prediction. *2016 31st Youth Academic Annual Conference of Chinese Association of Automation (YAC)*. doi:10.1109/yac.2016.7804912

Graves, A., Mohamed, A., & Hinton, G. (2013). Speech recognition with deep recurrent neural networks. *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. doi:10.1109/icassp.2013.6638947

Guisan, A., Edwards Jr, T. C., & Hastie, T. (2002). Generalized linear and generalized additive models in studies of species distributions: setting the scene. *Ecological modelling*, *157*(2-3), 89-100.

Hale, J. (2001). A probabilistic Earley parser as a psycholinguistic model. In *Proceedings of the second meeting of the North American Chapter of the Association for Computational Linguistics on Language Technologies*, 1-8.

Hansen, L. K., & Salamon, P. (1990). Neural network ensembles. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (10), 993-1001.

Hastie, T., & Tibshirani, R. (1987). Generalized additive models: some applications. *Journal of the American Statistical Association*, *82*(398), 371-386.

Hochreiter, S. (1998). The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, *6*(02): 107-116.

Kidd, C. (2018). NLU vs NLP: What's the Difference? Retrieved from https://www.bmc.com/blogs/nlu-vs-nlp-natural-language-understanding-processing/

Kolb, B. & Whishaw, I.Q. (2000). An Introduction to Brain and Behavior

Kutas, M., & Federmeier, K. D. (2011). Thirty years and counting: Finding meaning in the N400 component of the event related brain potential (ERP). *Annual Review of Psychology, 62*, 621–647.

Levy, R. (2008). Expectation-based syntactic comprehension. Cognition, 106(3):1126–1177. Lieberman, M. [The Ling Space]. (2015). *Neurolinguistic Processing.* [Video File]. Retrieved from https://www.youtube.com/watch?v=wi5mQs7c56c

Linzen, T., Dupoux, E., & Goldberg, Y. (2016). Assessing the ability of LSTM to learn syntax-sensitive dependencies. *Transactions of the Association for Computational Linguistics, 4,* 521–535.

Mikolov, T., Karafiát, M., Burget, L., Cernocký, J., & Khudanpur, S. (2010). Recurrent Neural Network Based Language Model. *11th Anual Coonference of the International Speech Communication Association.* Interspeech-2010, 1045-1048.

Peters, B., & Chaves, C. [Image 1] (n.d.). What It's Like to Have an Electroencephalogram (EEG). Retrieved from https://www.verywellhealth.com/what-is-an-eeg-test-and-what-is-it-used-for-3014879

Poulos, M., Alexandris, N., Belessioti, V., & Magkos, E. [Image 2] (2003). Comparison between computational geometry and coherence methods applied to the EEG for medical diagnostic purposes. *Recent Advances in intelligent Systems and Signal Processing, ICAISC.*

Salvaris, M., Dean, D., & Tok, W. (2018). *Deep learning with Azure: Building and deploying artificial intelligence solutions on the Microsoft AI platform*. New York: Apress.

Schafer, R. (2015). Processing and querying large web corpora with the COW 14 architecture. *Proceedings of the 3rd Workshop on the Challenges in the Management of Large Corpora,* 28-34. Mannheim, Germany: Institut für Deutsche Sprache.

Seabold, S., & Perktold, J. (2010). Statsmodels: Econometric and statistical modeling with python. In *Proceedings of the 9th Python in Science Conference*, *57*, 61. Scipy.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Llion, J., Gomez, A. N., Kaiser, L. & Polosukhin, I. (2018). Attention Is All You Need.

Xiong, W., Droppo, J., Huang, X., Seide, F., Seltzer, M., Stolcke, A., Yu, D., & Zweig, G. (2017). Toward human parity in conversational speech recognition. *IEEE/ACM Transactions on Audio, Speech and Language Processing,* 2410–2423.

Zaremba, W., Sutskever, I. & Vinyals, O. (2015). Recurrent Neural Network Regularization. *Neural and Evolutionary Computing.*