Predicting the N400 Component in Manipulated and Unchanged Texts with a Semantic Probability Model

Johannes Bjerva

Institutionen för lingvistik
Examensarbete 15 hp
Kandidatprogram i datorlingvistik (180 hp)
Vårterminen 2012
Handledare: Mats Wirén, Robert Östling, Petter Kallioinen

Stockholms universitet

Predicting the N400 Component in Manipulated and Unchanged Texts with a Semantic Probability Model

Abstract

Within the field of computational linguistics, recent research has made successful advances in integrating word space models with *n*-gram models. This is of particular interest when a model that encapsulates both semantic and syntactic information is desirable. A potential application for this can be found in the field of psycholinguistics, where the neural response N400 has been found to occur in contexts with semantic incongruities. Previous research has found correlations between *cloze probabilities* and N400, while more recent research has found correlations between cloze probabilities and language models.

This essay attempts to uncover whether or not a more direct connection between integrated models and N400 can be found, hypothesizing that low probabilities elicit strong N400 responses and vice versa. In an EEG experiment, participants read a text manipulated using a language model, and a text left unchanged. Analysis of the results shows that the manipulations to some extent yielded results supporting the hypothesis. Further results are found when analysing responses to the unchanged text. However, no significant correlations between N400 and the computational model are found. Future research should improve the experimental paradigm, so that a larger scale EEG recording can be used to construct a large EEG corpus.

Sammendrag

Innom datalingvistikken har tidligere forskning gjort framsteg når det gjelder å kombinere ordromsmodeller og *n*-grammodeller. Dette er av spesiell interesse når det er ønskelig å ha en modell som fanger både semantisk og syntaktisk informasjon. Et potensielt bruksområde for en slik modell finnes innom psykolingvistikk, der en neural respons som kalles N400 vist seg å oppstå i kontekster med semantisk inkongruens. Tidligere forskning har oppdaget en sterk korrelasjon mellom *cloze probabilities* og N400, og nylig forskning har funnet korrelasjoner mellom cloze probabilities og sannsynlighetsmodeller fra datalingvistikk.

Denne oppgaven har som mål å undersøke hvorvidt en mer direkte kobling mellom slike kombinerte modeller og N400 finnes, med hypotesen at lave sannsynligheter leder til store N400-responser og omvendt. Et antall forsøkspersoner leste en tekst manipulert ved hjelp av en slik modell, og en naturlig tekst, i et EEG-eksperiment. Resultatsanalysen viser at manipuleringene til en viss grad gav resultat som støtter hypotesen. Tilsvarende resultat ble funnet under resultatanalysen av responsene til den naturlige teksten. Ingen signifikante korrelasjoner ble oppdaget mellom N400 og den kombinerte modellen. Forbedringer for videre forskning involverer å blant annet forbedre eksperimentparadigmet slik at en storstilt EEG-inspilling kan gjennomføres for å konstruere en EEG-korpus.

Sammanfattning

Inom datalingvistiken har tidigare forskning visat lovande resultat vid kombinering av ordrumsmodeller och *n*-gramsmodeller. Detta är av speciellt intresse när det är önskvärt att ha en modell som fångar både semantisk och syntaktisk information. Ett potensielt användningsområde för en sådan modell finns inom psykolingvistiken, där en neural respons kallad N400 visat sig uppstå i situationer med semantisk inkongruens. Tidigare forskning har upptäckt en stark korrelation mellan *cloze probabilities* och N400, medan en nyare studie har upptäckt en korrelation mellan cloze probabilities och sannolikhetsmodeller från datalingvistiken.

Denna uppsats har som mål att undersöka huruvida en mer direkt koppling mellan sådana kombinerade modeller och N400 finns, med hypotesen att låga sannolikheter leder till stora N400-responser och vice versa. Ett antal försökspersoner läste en text manipulerad med hjälp av en probabilistisk modell, och en naturlig text, i ett EEG-experiment. Resultatsanalysen visar att manipuleringen till viss grad gav resultat som stödjer hypotesen. Motsvarande resultat hittades under resultatanalysen av responserna till den naturliga texten. Inga signifikanta korrelationer blev upptäckta mellan N400 och den kombinerade modellen. Förbättringar för vidare forskning involverar bland annat att förbättra experimentparadigmet så att en storskalig EEG-inspelning kan genomföras för att konstruera en EEG-korpus.

Keywords

Computational semantics, EEG corpus, Model integration, N400
Datorlingvistisk semantik, EEG-korpus, Modellintegrering, N400

Table of contents

1 Introduction

In computational linguistics, models attempting to reflect the semantic content of words (e.g. word space models) are often evaluated using tests of semantic coherence. Such tests often rely on investigating the proximity between two words in the given model, comparing this proximity to a given baseline. Often, such tests of semantic coherence include comparing antonyms, synonyms, heteronyms or words with other semantic meaning relations. Methods that are successful in judging words as, for instance, being synonymous, are thus considered to reflect the semantic content of words on a semantic level that encapsulates this feature.

Such tests can certainly be considered successful in showing whether or not a model reflects meaning on the relevant semantic level. However, whether a model can reflect neural representations of meaning, is not answered by such tests. In order to discover whether or not a computational model accurately reflects the way meaning is represented in the human mind, these models must be tested for potential correlations with data representing how meaning is mapped in the human brain.

This essay suggests a route towards investigating this, through evaluating various language models by comparing their outputs to that of the human brain, as measured by electrical voltage fields measured on the scalp using EEG. The focus of this essay will be a neural response called N400, which has been shown to be elicited by semantic incongruities. The language models focused on in this essay will be an *n*-gram model implemented with an advanced smoothing algorithm, interpolated with probabilities derived from a Word Space model.

Furthermore, the construction of a unique resource for further research is discussed, in which neural activity is used as an extra layer of annotation for an already richly-annotated corpus, resulting in a multimodal corpus.

2 Background

In this section there will be an attempt to cover the most relevant background literature from the fields of psycholinguistics and computational linguistics, and from the overlapping area that directly concerns this essay.

2.1 Neural representations of meaning

Within the field of psycholinguistics, much research has been done to investigate how meaning is represented in the human brain (see e.g. Hart and Kraut (2007) for a review). Such research has contributed to the discovery of Event Related Potentials (ERPs). These ERPs can be used to analyze how different stimuli in various conditions are interpreted by the human brain.

N400 is an ERP component that is strongly connected to semantics; various experiments have shown that this ERP is very susceptible to manipulations in the form of altering stimuli to be more or less congruent with their contexts (see section 2.6 for a more detailed account). Prior to manipulating such stimuli, however, it is necessary to gather data that reflects this semantic congruity. For this, cloze probabilities are often used. Obtaining cloze probabilities essentially means that a gap-fill exercise is carried out by a large number of participants. In these exercises, a word is removed from a sentence (often the final word), and the participants are given the task of filling the gap (Kutas and Federmeier, 2011). The cloze probability of a word occurring in a gap can then be calculated as follows:

$$P_{cloze}(w_i) = \frac{o(w_i)}{\sum_{j=1}^{j=n} o(w_j)} \tag{1}$$

where $P(w_i)$ is the probability of a given word, $o(w_i)$ is the occurrences of this word, and n the amount of words in the lexicon, all given from the gap-fill exercises.

2.2 Computational representations of meaning

Measuring cloze probabilities of large datasets requires a vast amount of both time and resources. The fact that certain contexts are particularly ambiguous (i.e. have high local entropy[1]) does not exactly alleviate the problem, as it would be practically impossible to cover all the possible words which could fill a gap (e.g. 'I saw a ___', 'My name is ___'). Obtaining reliable and exhaustive data that accurately describes the distribution of word frequencies for such sentences with a manual approach such as cloze probabilities is virtually impossible. Because of this, a computationally driven method might prove more successful.

Seeing as the words provided in a cloze probability completion task are words containing an appropriate semantic content depending on the preceding context, a criterion of high importance for such a method is that it reflects semantic content on some relevant level. Computational linguistics provides several potential methods for solving this issue. Examples of such methods are semantic networks such as WordNet (see e.g. Pedersen et al. (2004)) and word-space models such as Latent Semantic Analysis (Landauer et al., 2007; Papadimitriou et al., 1997) and Random Indexing (Kaski, 1998; Sahlgren, 2006).

Semantic networks are, however, mainly suited for creating taxonomic hierarchies (Sussna, 1993), depicting such relations as meronomy (A is a part of B), holonymy (B is a part of A) and hyponymy (A is a kind of B), although they can also represent synonymy (A denotes the same as B) and antonymy (A denotes the opposite of B) (see Figure 1).

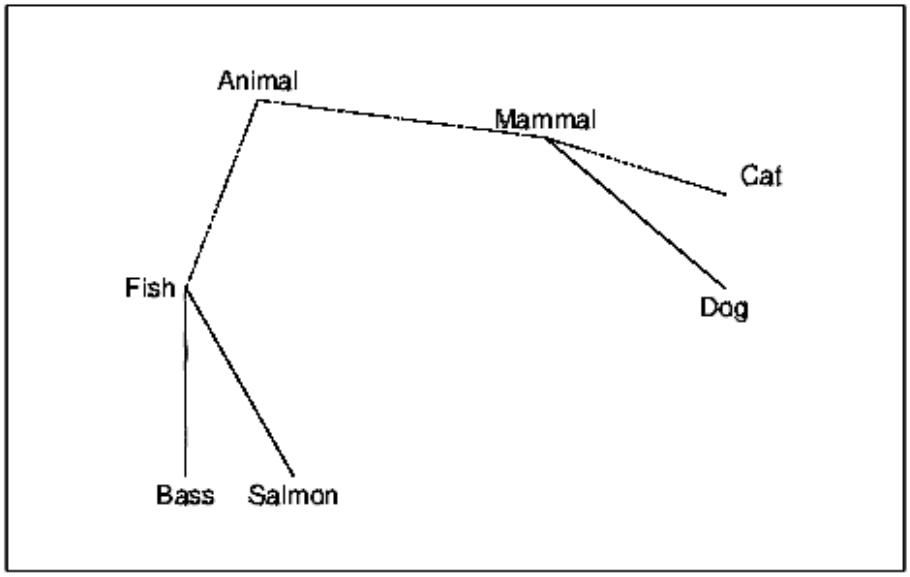


Figure 1: Semantic network

Furthermore, these networks are frequently constructed manually, rather than automatically. Automatic generation has, however, been attempted (see e.g. Navas et al. (2005)). Although successful, such machine-powered approaches tend to focus on generating clusters of words in *semantic fields*, rather than individual words represented by nodes in a graph. Seeing as manual construction of such semantic hierarchies is a time-consuming task, this puts them at a disadvantage compared to word space models.

As word space models seem to fit the bill for automatically creating computational representations of several types of semantic meaning relations (synonymy, antonymy etc.), they will remain the focus of this essay.

[1] Entropy can be described as a sort of *chaos value*, describing the uncertainty of a random variable. In our case, a high entropy means the probability distribution of words is flat (i.e. many words are equally improbable, while few to no words have high probabilities)

2.3 Word space models

> *'Semantics is partly a function of the statistical distribution of words.'*
> The distributional hypothesis (Harris, 1985)

The above quote can be taken to mean that the semantic content of any given word is closely linked to any word that tends to occur in similar contexts. Furthermore, Harris (1985) claims that this link is gradual in that if two words *A* and *B* that are more similar in meaning than *A* and *C*, then *A* and *B* will have more similar contextual distributions than *A* and *C*.

Using this assumption, the semantic content and neighbourhood of words (e.g. synonyms, antonyms, hyper-/hyponyms) can be mapped with a simplistic vector model. In such models, every word is described by a vector, and the distances between these vectors describe how closely the words are tied to each other on some (arbitrary) semantic level (Figure 2).

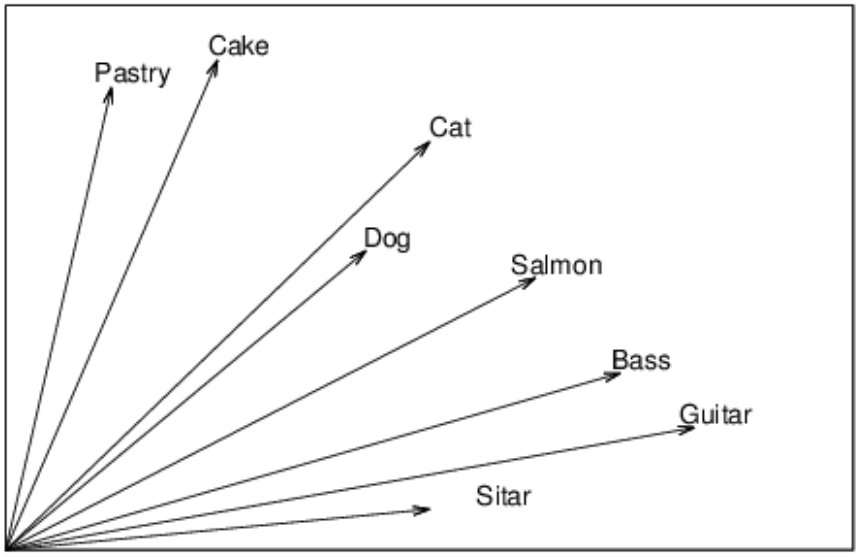


Figure 2: 2 Dimensional Word Space

In the simplest of implementations, every word w might be assigned a descriptive vector and a feature dimension in an n-dimensional space, where n is the total amount of words. These vectors could then be used to describe the semantic content of words, based on co-occurrences. That is to say, each word might be described as follows:

$$\vec{w} = \sum_{w \in \mathbb{C}} e_{f(w)} \tag{2}$$

where $\vec{w}$ is any given word's vector, w_f a word's feature dimension, $\mathbb{C}$ a set containing all the words in the context, e a word's elemental vector, and $f(w)$ a function that returns a given word's dimension. For example, given a document d_1 (containing the sentence: *dogs eat cats*), the vectors and the distance between them would change using this method as detailed in Table 1, using a context window of only the adjacent words.

Table 1: Vector generation with a simplistic method

	$\vec{V}_{cats}$	$\vec{V}_{eat}$	$\vec{V}_{dogs}$	$\cos(\vec{V}_{cats}, \vec{V}_{dogs})$
Feature dimension	1	2	3	0.0
Descriptive vector	0,1,0	1,0,1	0,1,0	1.0

Where $\vec{V}_{cats}$ is *cats'* descriptive vector, $\vec{V}_{dogs}$ is *dogs'* descriptive vector and $\cos(\vec{V}_{cats}, \vec{V}_{dogs})$ the cosine similarity between them. As is illustrated in the table, the original vectors with only the feature dimensions in place are orthogonal, with 0 cosine similarity. However, once the descriptive vectors are in place (adding feature dimensions from the adjacent words in the document), this similarity is improved, showing a connection between *cats* and *dogs*.

Although simple, assigning a feature dimension to each unique word quickly makes the task of generating a word space computationally infeasible. The problem with this solution becomes apparent as the amount of words in the lexicon increases. Since each word is assigned a new dimension, the dimensionality quickly increases to the point where computational constraints (e.g. memory usage) become too high. Because of this, a way of reducing the dimensionality is necessary. A commonly used method of dimensionality reduction is Latent Semantic Analysis (see e.g. Landauer and Dumais (1997); Landauer et al. (1998, 2007)).

In LSA, a term-document matrix which describes which words occur in which documents, is generated for the relevant corpus. This input matrix is then decomposed into vectors of lower dimensionality, using Singular Value Decomposition. However, LSA is limited by that it is not particularly scalable, due to the heavy computational costs of both SVD and the calculation of a complete co-occurrence matrix. LSA does have some advantages, such as a resulting word space that is less sparse and less noisy than a word space constructed using a simplistic method such as outlined in Equation 2 (Landauer et al., 2007). However, the high computational costs do encourage the use of other methods.

2.3.1 Dimensionality reduction through Random Indexing

A solution provided by Kanerva et al. (2000), Random Indexing (RI), involves using randomised mapping of data vectors for dimensionality reduction of a dataset. Using this method, Kaski (1998) shows that nearly an equal accuracy (i.e. the distances between vectors in the word space) compared to the word space representations prior to dimensionality reduction is achieved, provided the final dimensionality is not too small (reduction from 6000 to 100 dimensions in Kaski (1998)). RI has been applied successfully to generate word spaces in various NLP applications (see e.g. Sahlgren (2006); Wandmacher and Antoine (2007)).

In order to generate an RI word space, two phases are necessary. First, elemental vectors must be allocated to each word in the lexicon. In this phase, each word is assumed to already have an n-dimensional zero vector, before being allocated k randomised +/-1 values. Vectors generated in this manner are often referred to as *index vectors* or *basic vectors*. However, this paper will use the term *elemental vectors* as introduced by Cohen et al. (2010).

A training phase follows the assignment of elemental vectors (see Algorithm 1).

Algorithm 1 Random Indexing: Assigning document vectors (Training part 1)

for $d \in \mathbb{D}$ **do** ▷ $\mathbb{D}$ is a collection of documents d
  Initialize $\vec{d}$ ▷ $\vec{d}$ is initialised as an empty vector for the document d
  for $w \in d$ **do** ▷ w is a word in the document d
    $\vec{d} \leftarrow \vec{w}_{elem}$ ▷ w's elemental vector $\vec{w}_{elem}$ is added to d's document vector $\vec{d}$
  end for
end for

Here, every document is assigned an n-dimensional zero vector. The elemental vectors of all the tokens in the document are then added to this *document vector*. Following this, the final *term vectors* are trained (see Algorithm 2).

Algorithm 2 Random Indexing: Assigning term vectors (Training part 2)

```
for d ∈ 𝔻 do                          ▷ 𝔻 is a collection of documents d
    for w ∈ d do                      ▷ w is a word in the document d
        w⃗_term ← d⃗                   ▷ d's document vector d⃗ is added to w's term vector w⃗_term
    end for
end for
```

In essence, every word is assigned a *term vector* $\vec{w}_{term}$. Each occurrence the word has in a document leads to this document's vector $\vec{d}$ being added to the word's term vector.

2.3.2 Discovering implicit connections through Reflective Random Indexing

A novel variation of RI which is argued to be suitable for discovering implicit and indirect connections is proposed in Cohen et al. (2010). This method is dubbed Reflective Random Indexing (RRI), as one or more extra iterations are involved in the training of vectors. First, term vectors are generated as in standard RI (Steps 1 through 3). These are then used as input for Algorithm 1 (Step 2). That is to say, rather than generating *document vectors* using words' *elemental vectors*, the words' *term vectors* are used. Following this, the resulting *document vectors* are used as input for Algorithm 2 (Step 3), generating a new set of *term vectors* (see Figure 3).

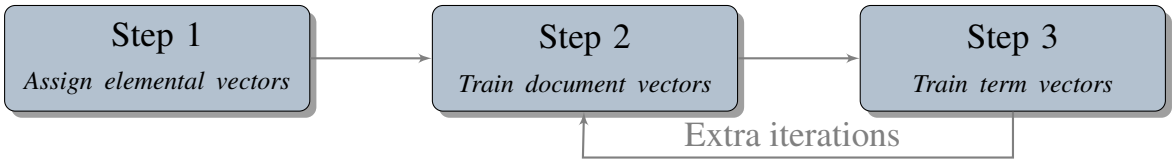


Figure 3: Extra iterations in Reflective Random Indexing

RRI has been applied successfully to find indirect inferences in a medical corpus, replicating historical findings and significantly outperforming standard RI (Cohen et al., 2010). Such direct inferences might indeed prove useful when attempting to uncover semantic relations. For example, in cases where two words have a high degree of attributional symmetry (i.e. share several attributes with one another), RRI might outperform RI. For example, given two documents d_1 (containing the words: *Elephant, Africa*) and d_2 (containing the words: *Elephant, Asia*), the vectors and the distance between them would change using RRI as detailed in Table 2.

Table 2: Differing cosine similarities over several iterations of RRI

	$\vec{V}_{elephant}$	$\vec{V}_{africa}$	$\vec{V}_{asia}$	$\cos(\vec{V}_{asia}, \vec{V}_{africa})$
Elemental vectors	1,0,0	0,1,0	0,0,1	0.0
1_{st} *iteration*	2,1,1	1,2,0	1,0,2	0.5
2_{nd} *iteration*	6,3,3	3,1,2	3,2,1	0.93

Where $\vec{V}_{elephant}$ is the context vector for elephants, $\vec{V}_{africa}$ the context vector for Africa and $\vec{V}_{asia}$ the context vector for Asia. As the iterations progress, the cosine similarity between $\vec{V}_{africa}$ and $\vec{V}_{asia}$ increases.

2.3.3 Semantic representations in word spaces

Sahlgren (2006) attempts to answer the question of exactly what type of semantic information word space models represent, and presents a categorisation paradigm of word-spaces where they can be di-

vided into two main categories: paradigmatic and syntagmatic. A *paradigmatic* relation is described as a relation of substitution. That is to say, any word x that can replace a given word y could be said to hold a paradigmatic relation with that word, as in the sentence *'This book is [interesting / boring]'*. On the other hand, a *syntagmatic* relation deals with co-occurrences of words. Collocations (e.g. 'cosmetic surgery', 'coffee break') are prime examples of such syntagmatic relations.

When constructing word spaces that are either syntagmatic or paradigmatic, certain key features can be derived from Sahlgren (2006) (see Table 3).

Table 3: Key features of syntagmatic and paradigmatic word spaces

Parameter	**Syntagmatic**	**Paradigmatic**
Context window	Large	Small
Context balance	-	Symmetric

The strengths and weaknesses of these two types of word spaces are presented through a various experiments in Sahlgren (2006).

2.3.4 Measuring word space similarities

In word space models, all words being described by the model are distributed in an n-dimensional space depending on their statistical distribution. From this word space, some sort of relationship between words can be derived depending on their distance from each other. In order to investigate this, however, some means of measuring this distance is needed. One such means is to quite simply measure the Euclidian distance between two vectors. This can be described as the length of a line connecting the end points two vectors. The Euclidian distance between two vectors u and v in an n-dimensional space is calculated as follows (adapted from Deza and Deza (2009, p. 92)):

$$d(u,v) = \sqrt{(u_1 - v_1)^2 + (u_2 - v_2)^2 + \ldots + (u_n - v_n)^2} \tag{3}$$

Where $d(u,v)$ is the distance between the vectors u and v. This distance measure has a disadvantage, however. As it can be viewed as the distance *as the crow flies*, the frequencies of words strongly impact the distances between them. This means that two words that are synonymous but differ in word frequency might have a very similar statistical distribution, but still be considered as very different using this measure. In NLP, this is often disadvantageous. Due to this, a distance measure that is often used is the cosine similarity measure (adapted from Deza and Deza (2009, p. 310)):

$$cos(\theta) = \frac{u \cdot v}{\|u\|\|v\|} \tag{4}$$

Where θ is the angle between u and v. In this measure, word frequencies do not directly contribute to the distance measure. Only co-occurrence frequencies will matter. The advantages of a cosine similarity measure compared to Euclidian distance are illustrated in Figure 4. The figure shows that the euclidean distance between vectors *salmon* and *guitar* is much smaller than the distance between vectors *sitar* and *guitar*. However, using cosine similarities, *sitar* and *guitar* are the closest. This is primarily due to the low word frequency of *sitar* compared to the other words.

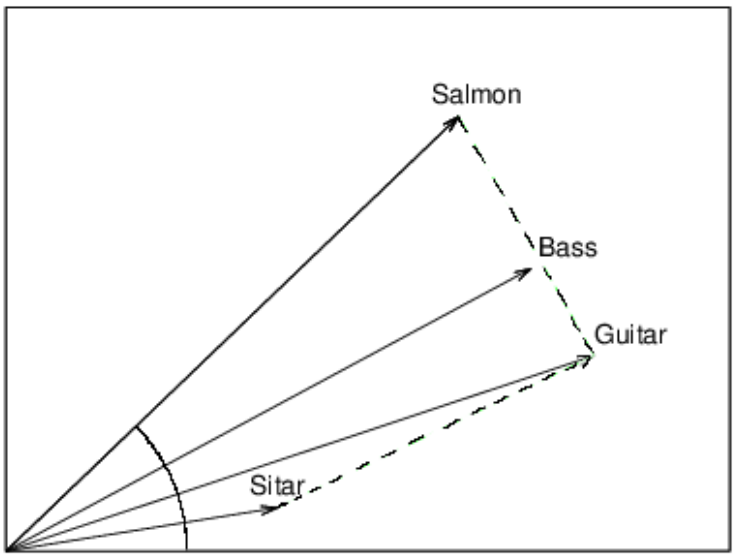


Figure 4: Differences between Cosine (solid bow) and Euclidian (dashed lines) distances

It is apparent that the raw word frequencies impact the euclidian distances to such an extent that it is not as suitable a measure as cosine similarity in some applications. However, this problem with euclidian distances disappears when vectors are normalised to the same length. Cosine similarity is nonetheless one of the most used distance measurements used in NLP (see e.g. Coccaro and Jurafsky (1998); Jurafsky and Martin (2009); Mitchell and Lapata (2009); Wandmacher and Antoine (2007)).

Word space models can be applied to a vast amount of tasks where it is useful to know which words are or are not closely related to each other. However, their insensitivity to a given word's preceding context highly limits their potential applications. Although word-space models could be seen as incorporating semantic information successfully on some level, it is hard to argue that they also represent syntactic information. Due to this, a model that captures such syntactic features in given contexts can be seen as necessary.

2.4 N-gram models

In order to capture possible sequences of words and their probabilities, a language model is needed. A language model can be defined as a statistical model which predicts a word n based on its preceding context c. One of the most frequently used ways of doing this involves using n-grams to calculate probabilities of word sequences, thus modeling a given language. A simple way of estimating such probabilities is to look at how many times a particular word has occurred after a context, as follows.

$$P(w_i|w_{i-n\ldots i-1}) = \frac{o(w_{i-n\ldots i})}{o(w_{i-n\ldots i-1})} \tag{5}$$

Where $o(x)$ denotes the amount of times an n-gram has been observed in the corpus. The simplicity of this solution comes at a high cost, however, as unobserved n-grams will have an estimated probability of zero.

Imagine that an n-gram model is trained (as in Equation 5) on a corpus consisting of the two sentences *'I like coffee'* and *'I like tea'*. Using n=3, these two sentences each represent an individual trigram. When asking the model what probabilities these trigrams have, they will both have $p = 0.5$. However, when asking the model what probability the trigram *'I like water'* has, it will receive $p = 0.0$. Clearly, this is

not optimal. This problem is alleviated by smoothing techniques, which essentially reassign some of the probability mass to unobserved *n*-grams, so as to provide a more realistic probability distribution.

2.4.1 Smoothing techniques in n-gram models

In order to escape the consequences of zero-probabilities, the collective probabilities of all *n*-grams are redistributed so as to include unobserved *n*-grams, through a process known as *smoothing* or *discounting*. Smoothing algorithms primarily come in two different flavours: interpolated models and back-off models (Chen and Goodman, 1998). In interpolated models, the distribution of *n*-grams of lower orders are always incorporated into the probability calculations (i.e. $P(w_3|w_2)$ as well as $P(w_3|w_{1,2})$). In back-off models, however, these probabilities are only considered when the highest order *n*-grams have zero observations in the corpus.

Absolute Discounting is an example of an interpolated model, in which the frequencies of both higher and lower order *n*-grams are incorporated into the probability calculations (Chen and Goodman, 1998). Kneser-Ney smoothing is an extension of Absolute Discounting, in which the lower-order *n*-gram distributions are combined with the higher-order *n*-gram distributions in a novel manner. Essentially, Absolute Discounting incorporates lower-order *n*-gram even when it might not be advantageous. Incorporating lower-order *n*-grams should only be of importance when the higher-order *n*-grams are low in frequency. Due to this, the interpolation should be optimised to match these cases (Chen and Goodman, 1998).

A concrete example is when a common word only occurs after a single word. Here, the unigram frequency will be high, meaning that it will be assigned as a potential candidate after many novel unigrams. However, since the word only occurs with the other word, the bigram probability models the situation well, while the unigram probability does not. According to Chen and Goodman (1998), the weight of unigram probabilities should not be proportional to a word's occurrences, but rather to the amount of different words that it follows.

The performance of different smoothing types varies greatly. Chen and Goodman (1998) thoroughly reviewed a vast array of different smoothing algorithms for use with language models. In this comparative study, they found that Kneser-Ney smoothing (Kneser and Ney, 1995) consistently outperformed other types of smoothing, while a novel variation of Kneser-Ney smoothing further increased performance.

2.5 Model integration

To recap, *n*-gram models can be said to represent syntactic information, whereas word-space models can be said to represent semantic information. As neither type of model can fully account for both of these key areas, a combination of the two might prove even more successful than they do in isolation. Such a combination has been suggested by Coccaro and Jurafsky (1998) and further developed by both Deng and Khudanpur (2003) and Wandmacher and Antoine (2007), as well as Mitchell and Lapata (2008, 2009). In all cases, the combination of these two methods yielded lower perplexities[1] in various situations.

2.5.1 Calculating probabilities from a word space model

Before any further steps can be taken, probabilities must first be derived from our word-space model. When using an LSA word space, Coccaro and Jurafsky (1998) propose that this can be done by calculating probabilities by using normalised distances between a given word and its preceding context. In this application, the probability was essentially the distance between the given word and the centroid vector of the x words that preceded it. This, however, led to an overall decrease in performance which can be explained by that the overall variance of distance in a vector space is much smaller than that of an *n*-gram model (Coccaro and Jurafsky, 1998). In order to alleviate this problem, an arbitrary power factor (λ) was introduced. Using $\lambda = 7.0$, a perplexity decrease of 11% was obtained when testing the model

[1] Perplexity can in this case be described as how probable a text is in a given language model. A low perplexity indicates that the model is less 'surprised' by a given text, which is interpreted as that the model performs well.

on a corpus. The resulting equation for calculating probabilities from a word-space can be defined as follows (adapted from Wandmacher and Antoine (2007)):

$$P_{wordspace}(w_i|c) = \frac{(cos(\vec{w}_i,\vec{c}) - cos_{min}(\vec{c}))^{\lambda}}{\sum^{j}(cos(\vec{w}_j,\vec{c}) - cos_{min}(\vec{c}))^{\lambda}} \tag{6}$$

Where w_i is the current word, c its preceding context, and $\vec{w}_i$ and $\vec{c}$ their respective vectors in any given word-space. The value of $cos_{min}(\vec{c})$ is calculated by finding the minimum distance between the context and any word in the lexicon. The probabilities are then normalised, so as to ensure that the sum of all probabilities is 1. Although a specific type of word space (LSA) was used in both Wandmacher and Antoine (2007) and Coccaro and Jurafsky (1998), there is no reason to assume that calculating probabilities would not be successful in other types of word spaces (such as RI).

The issue encountered at this stage is the integration of this semantic information with the information provided by an *n*-gram model. Several methods have been proposed for combining word space models with *n*-gram probabilities (e.g. Bellegarda (2004); Coccaro and Jurafsky (1998); Khudanpur and Wu (1999); Wandmacher and Antoine (2007)). Different types of *interpolation* have proven to be particularly successful in combining information from similar sources such as two subsets of probability distributions, as in the case presented here. As both models are now of a similar format, namely as a probability distribution, the two models can be interpolated.

2.5.2 Linear and geometric interpolation of two probability distributions

Linear interpolation provides a simplistic way of interpolating two models (Wandmacher and Antoine, 2007). It can be described as follows:

$$P'(w_i) = \lambda \cdot P_n(w_i) + (1-\lambda) \cdot P_{ws}(w_i) \tag{7}$$

Where λ is a weighting coefficient between 0 and 1, $P_n(w_i)$ a given word's probability in the *n*-gram model, and $P_{ws}(w_i)$ the same word's probability in the word space model.

A more sophisticated way of interpolating two models is presented, in which high end-probabilities are only achieved if the two models *agree* on the likelihood of a given word (Coccaro and Jurafsky, 1998). Geometric interpolation can be described as follows:

$$P'(w_i) = \frac{P_n(w_i)^{\lambda_1} \cdot P_{ws}(w_i)^{(1-\lambda_1)}}{\sum^{j} P_n(w_j)^{\lambda_1} \cdot P_{ws}(w_j)^{(1-\lambda_1)}} \tag{8}$$

Where λ is a weighting coefficient between 0 and 1, $P_n(w_i)$ a given word's probability in the *n*-gram model, and $P_{ws}(w_i)$ the same word's probability in the word space model.

Through experiments with linear and geometric interpolation, as well as other types of model integration (e.g. recency promotion, semantic caching), Wandmacher and Antoine (2007) found that the highest perplexity reductions were found with geometric interpolation.

The graphs in Figure 5[1] illustrate the differences in probability distributions derived from geometric interpolation and linear interpolation.

[1] Figure 5 is copied from Coccaro and Jurafsky (1998) with minor modifications.

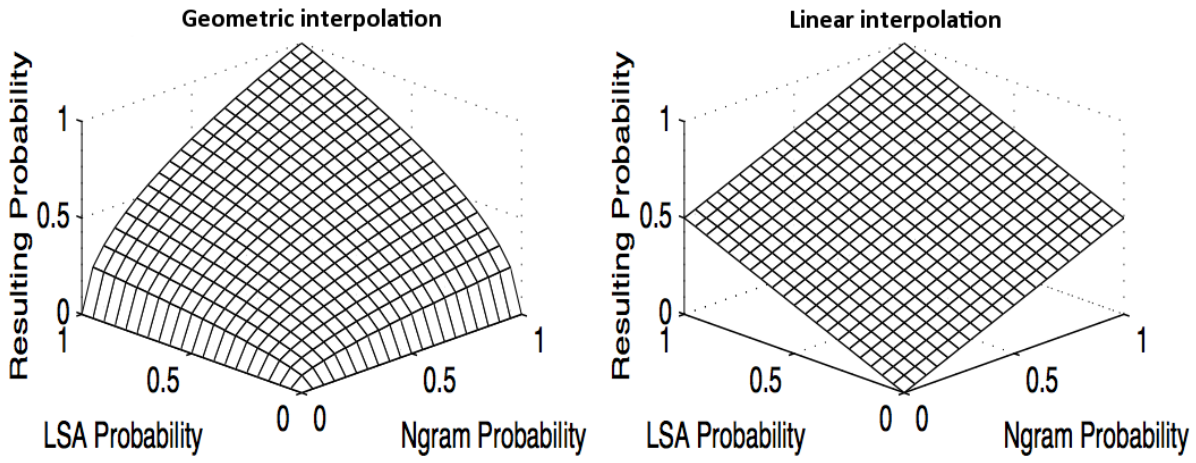


Figure 5: Differences between Geometric and Linear interpolation

As the figure illustrates, a high resulting probability can only be obtained from geometric interpolation if both models agree upon the high probability. In linear interpolation, however, a word with a high probability in one model and a low-to-zero probability in the other can still give a high resulting probability.

The differences between how the models and the different types of interpolation can differ in predicting a word in a given context are illustrated in Table 4.

Context: 'Professors and Ph.D. students at universities often do ___'

Table 4: Word probabilities in N-gram model, Word Space and Interpolations

Word	**N-gram prob.**	**Word space prob.**	**Linear Int.**	**Geometric Int.**
research	Medium	High	High	High
articles	Low	High	Medium	Low
not	High	Low	Medium	Low

As the table illustrates, only the cases with medium/high probabilities in both probability distributions are assigned high probabilities with the geometric interpolation. In linear interpolation, however, all cases with medium/high probabilities in either model are given relatively high probabilities with linear interpolation.

2.6 N400

Event related potentials have remained a research area with much focus since their discovery. An early ERP component to be discovered was P3b (see Kutas and Federmeier (2011) for a review). This component has been found to be inversely correlated with a stimulus's probability of occurrence (Kutas and Hillyard, 1983). An example of a paradigm which elicits large P3bs is the so-called 'oddball paradigm'. Here, an array of identical stimuli is presented with intermitting distinct stimuli appearing occasionally (e.g. presentation of speech sounds in the following order: /b b b b b f b b b b b b f b b b/). The focus of this study, N400, was discovered by Kutas and Hillyard (1980) when this oddball-paradigm was altered. Participants read a collection of sentences, some of which ended with either an an anomalous or an unlikely word. This led to the discovery of an unexpected negativity which peaked at approximately 400 ms after stimulus onset. It is worth to note, however, that the negativity is not in absolute terms, but relative to a 100 ms prestimulus baseline. This, in addition to the fact that the poststimulus onset window of the component lies in the temporal region of 200-600 ms, led to it being assigned the heuristic label N400 (see e.g. Kutas and Federmeier (2011)). Within the field of psycholinguistics, N400 remains one of the most studied ERP components (Kutas and Federmeier, 2011).

This study and several others have shown that the amplitude of the N400 is highly manipulative, due to its sensitivity to semantic anomalies and lower word probabilities (see e.g. Kutas and Federmeier (2009)). An early study examining how the magnitude of N400 is influenced by an item's expectancy was carried out by Kutas and Hillyard (1984). Every word's expectancy was estimated using a set of cloze norms developed by Bloom and Fischler (1980). In Kutas and Hillyard (1984), it is concluded that a strong correlation holds place between N400 responses and cloze probabilities. A recent study collaborates this and further shows that the level of sentential constraint (i.e. the amount of possible words to fill a gap, as selected by a sample of individuals) does not influence N400 in situations with semantic anomalies (Federmeier et al., 2007). Where semantic anomalies are absent, however, low sentential constraint elicits a slight N400 response.

2.7 An intersection between psycholinguistics and computational linguistics

A study by Hahn (2011) attempted to investigate whether any correlations could be found between probabilities calculated by an *n*-gram model and cloze probabilities. This study was successful in that such correlations indeed were found. However, this only shows a rather indirect connection between the computational model and the neural response in question. A further step that should be taken is to directly investigate the connection between the probabilities calculated by such a model and neural responses. Ideally, a causal relationship should be established between the two, through experimental manipulation of probabilities, hypothesizing that lower probabilities will yield stronger N400 responses. Seeing as N400 is sensitive to semantics, using a computational model that encapsulates semantic content might yield further results.

2.8 A novel language resource

Corpora are well suited for doing quantitative language research, as they contain large amounts of data which is a necessity for most, if not all, computational applications based on statistics (see e.g. Jurafsky and Martin (2009) for a review of various applications). In order to do such quantitative research when it comes to psycholinguistics, corpora may provide a convenient format. A large corpus containing words annotated with neural activity on different levels would undoubtedly prove to be a highly valuable resource. Adding neural annotation to an already rich resource such as STB-SUC[1], further increases the amount of potential studies which can be performed. After a substantial amount of neural annotation has been incorporated to the corpus, neural activity can be investigated as a function of any of the other annotations applied, or measures derived from them, such as word probabilities.

2.9 Goals

The goals of this essay can be defined by four key points:

1. How are semantically and syntactically oriented language models best integrated?
2. Can a language model's probability estimates of replaced words predict the amplitude of N400 responses (hypothesising that low word probabilities elicit larger N400 responses)?
3. How does a language model's probability estimates of words in an unchanged text correlate with the N400 component?
4. What practical considerations are important when constructing a multimodal corpus with EEG data? Is this a worthwhile project for psycholinguists?

[1] Swedish Tree Bank - Stockholm Umeå Corpus (http://spraakbanken.gu.se/eng/resources/suc)

3 Method

This study consists of two sub-studies: the first aiming to provide answers to the first question and the second to the remaining three of the aforementioned questions.

3.1 Study I: Interpolation of two language models

This experiment attempts to provide an answer to the first research question. That is to say, how successfully two language models reflecting two different aspects of language can be integrated into one, more accurate, language model.

3.1.1 Model implementation

Two types of computational models have been implemented for this study. The first is a probabilistic *n*-gram model, implemented using *n*-grams up to the magnitude of *n*=4 (e.g. *This is a dog*). This *n*-gram model is smoothed with Kneser-Ney smoothing, as detailed in section 2.4.1. The *n*-gram model was trained on data obtained from Google *n*-grams[1], which is a large corpus based on word and *n*-gram occurrences in a large amount of Swedish websites. The second model is a word-space model, implemented using Reflective Random Indexing (see section 2.3.2) for dimensionality reductions and to optimise the word-space for discovering implicit connections. This model was trained on a different corpus, consisting of approximately 1 billion words gathered from Swedish blogs[2]. When training the model, the 200 most frequent words were not included, as they are primarily function words with little to no semantic content.

Seeing as word space models allow for manipulation of several parameters (see e.g. Sahlgren (2006) for a review), a range of different values were used to investigate their influence on the results. These different settings were then evaluated using a test of semantic coherence. In this test, the word space model is provided three words, two of which have a semantic meaning relation, and one random word. The model then determines which two words are related to each other. As the word space model aims to reflect semantic content, this test should show which of the parameter settings is optimal.

The best-performing setting (see Section 4.1) was used for the word space model throughout the remainder of this essay. Varying the amount of iterations of RRI were also tested Following this, probabilities were calculated from the word space model as detailed in Section 2.5.1, again altering parameters in order to find the optimal settings. The two probabilistic models were then integrated using both Linear Interpolation and Geometric Interpolation (see Section 2.5.2) and perplexities calculated in order to find the optimal parameter settings

3.1.2 Model evaluation

A commonly used way of assessing the quality of a language model, is using it to calculate the perplexity of an evaluation text distinct from the data used to train the model (see e.g. Chen and Goodman (1998)). In order to verify the quality of the *n*-gram model and the interpolation with a word-space model used in this study, such perplexity calculations were performed.

The target corpus for the perplexity calculations was SUC (the same corpus as used in the neural experiments), which is completely distinct from the models' training materials.

[1] Google Catalog No. LDC2009T25: http://www.ldc.upenn.edu/Catalog/catalogEntry.jsp?catalogId=LDC2009T25
[2] Blogs gathered by Twingly (www.twingly.com)

3.2 Study II: EEG reading study

Two texts from the test section of Stockholm Umeå Corpus[1] were chosen as the reading material for this study. These texts are annotated morphosyntactically, which provides the necessary information for text manipulation using a fully automated procedure. A total of 20 participants performed the experiment (mean age = 21, SD = 2.21, 10 male and 10 female, all right-handed). Three participants were excluded from the results due to the amount of data rejected in pre-processing in the case of one participant, and the other two participants' data contained too high variance in comparison with other participants.

For the first portion of this study, approximately 25% of the content words (nouns, verbs and adjectives) were replaced with a novel word. The replacement words were chosen so that words with both high and low probability were included. The probabilities were calculated using the *n*-gram model smoothed with Kneser-Ney smoothing, as detailed in section 2.4.1. All content words were chosen as potential targets for manipulation in order to have a large breadth in regards to what words were replaced. Although larger N400 effects may be simpler to elicit by only replacing nouns, manipulation of adjectives and verbs might also yield interesting results.

3.2.1 Stimuli preparation

In order to prevent any human bias when replacing words in the text, a fully automated approach was used. An automated script replaced common nouns, verbs and adjectives with words sharing the same morphosyntactic tagging. This was done so as to ensure the replacements would be grammatically correct (i.e. a word tagged as a singular, definite noun, such as 'bilen' (*'the car'*), would be replaced with a word with the same tag, such as 'katten' (*'the cat'*)). Approximately 15% of words belonging to the aforementioned classes were replaced.

The replacements were made taking word probabilities, as calculated by the Kneser-Ney smoothed *n*-gram model, into consideration. For each word, probabilities for words sharing the same morphosyntactic tags were calculated. Replacement words were chosen from this subset of words. The replacements were picked so as to ensure a wide spread in terms of probabilities (Table 5). They were divided into three groups depending on these probabilities: high, medium and low. The high probability group contained the 10 most likely words, while the low probability group contained words in the least likely quartile of all the words. The medium probability group contained all words in between these groups.

Table 5: Example of word replacements in different probability conditions, given the context:
Men halvön som det ligger på var obebyggd och värdefull, ett stycke orörd __
But the peninsula it was situated on was uninhabited and valuable – an untouched piece of __

Condition	Replacement	Translation
Original word	Sörmlandsnatur	Sörmland nature
High probability	Skog	Forest
Medium probability	Urskog	Primeval forest
Low probability	Gåslever	Goose liver

3.2.2 Experimental setup

Participants were seated in an experimental room adjacent to an experimental control room. The experimental room was enclosed by a Faraday cage, so as to prevent external electric fields from disturbing the EEG recordings. Words from the text were presented centrally on a screen placed approximately 1 m from test participants. The size of the stimuli on the screen was sized so that words would not extend further than 4° in the visual field of the participants while reading, as visual acuity is decreased outside of this range (as discussed in Hörberg et al. (2012)). Words were presented one word at a time at varying presentation times (see Table 6).

[1] SUC IDs: kk14 (fiction novel) and kl07 (crime novel)

Table 6: Presentation times of words in the study. Actual presentation times were randomised uniformly in the intervals stated. All values are in milliseconds.

	Min. presentation time	Max. presentation time	Poststimulus pause
Content word	500 ms	700 ms	200 ms
Function word	100 ms	200 ms	200 ms

Replacement words were presented in the same way as other content words. However, after the poststimulus pause three exclamation marks ('!!!') were be shown to indicate that a replacement had occurred. Following this, text presentation continued as normally (see Figure 6). This was done in order to restore the reading experience to a more natural state without losing any potentially important contextual information. That is to say, if the word *house* was replaced by *car*, unwanted neural responses might occur later in the text when the house is mentioned as an entity known to the reader.

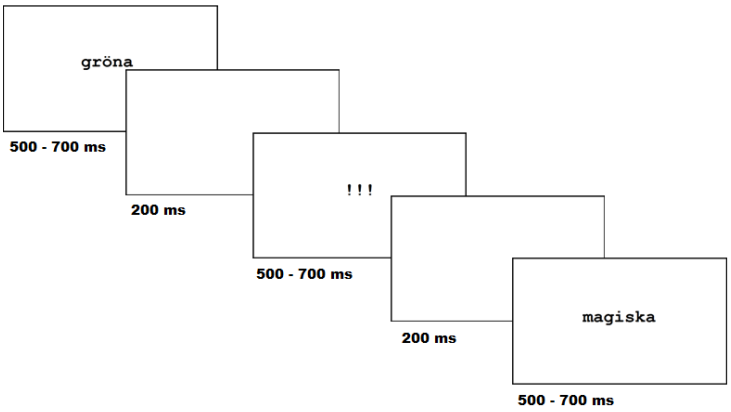


Figure 6: Stimuli presentation

Prior to the experiments, participants were given basic information about the task at hand. They were informed that some words in the text would be replaced, and these replacements would be revealed by following exclamation marks and then corrected by the original word. In the word replacement experiment, the participants read approximately 1200 words each.

During the second portion of the study the experimental setup was identical to that of the first portion, with the exception of no words being replaced. The text read by the participants was preserved in its original state, so as to allow for analysis of neural activity in an unchanged text. In this condition, the participants read approximately 800 words each.

All participants read the same texts. However, in the word replacement experiment all the manipulations were unique to each participant.

3.2.3 Processing and analysis of EEG data

During the reading study, data was recorded with an *EGI* electroencephalography system consisting of a 128 channel hydrocel electrode net and a *Net Amp 300*. The central Cz channel was used as reference during the recording process. The signal was sampled at to 20 kHz and then resampled to 250 Hz during recording. The impedance was kept below 50 kΩ on most electrodes.

Recordings were filtered through a bandpass filter in *Net Station* (1-40 Hz). Blink artifacts were removed with an automatic procedure[1]. Further pre-processing, including artifact detection and statistical rejection of data was also carried out with this toolkit. All epochs (periods during which the data is analysed) were re-referenced to linked mastoids and baseline corrected relative to the 200 ms preceding stimulus onset.

[1] J. Dien's EP Toolkit (http://sourceforge.net/projects/erppcatoolkit/)

4 Results

4.1 Results of Study I: Interpolation of two language models

In order to find optimal parameter settings for the dimensionality reduction through Random Indexing, some key parameters were manipulated in a total of four conditions(see Table 7).

Table 7: Parameter manipulation in the Random Indexing word space

Condition	Dimensions	Elemental vector alterations	Context window
1	200	4	200 words
2	2000	4	200 words
3	2000	8	200 words
4	2000	8	500 words

In the table, *Dimensions* indicates the dimensionality of the context vectors used. *Elemental vector alterations* is the amount of dimensions in the elemental vectors that were assigned a value of +/- 1. *Context window* is the amount of words, preceding any given word, whose elemental vectors were added to any given word's context vector.

Using these different parameters, a test of semantic coherence was carried out (see Figure 7).

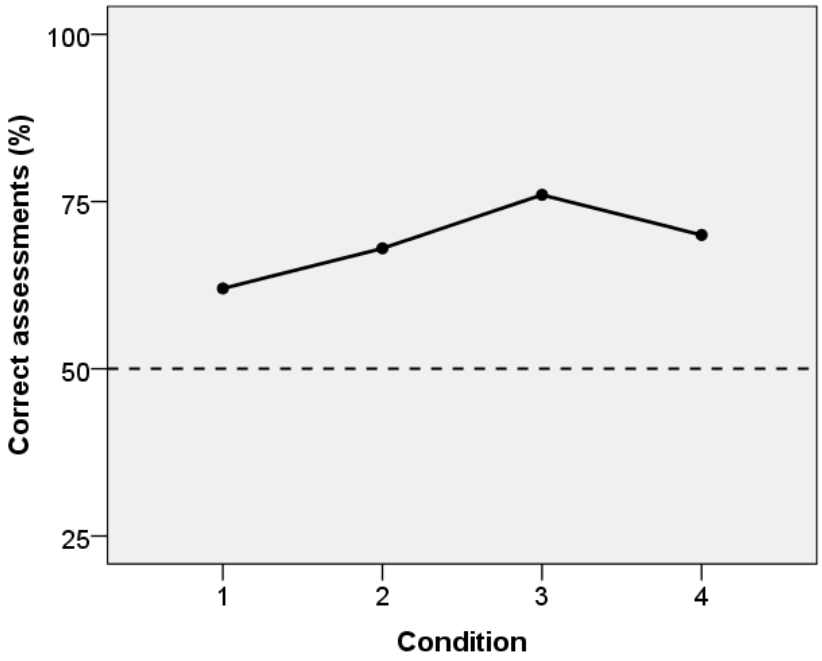


Figure 7: Percentage correct responses with varying RI parameters

The dotted line indicates the baseline of 50%, obtained through randomly guessing in the test of semantic coherence. The optimal results are found in condition 3 with approximately 76% correct assessments. The effect of having more iterations with Reflective Random Indexing was investigated, using the most optimal condition from the RI tests (see Figure 8).

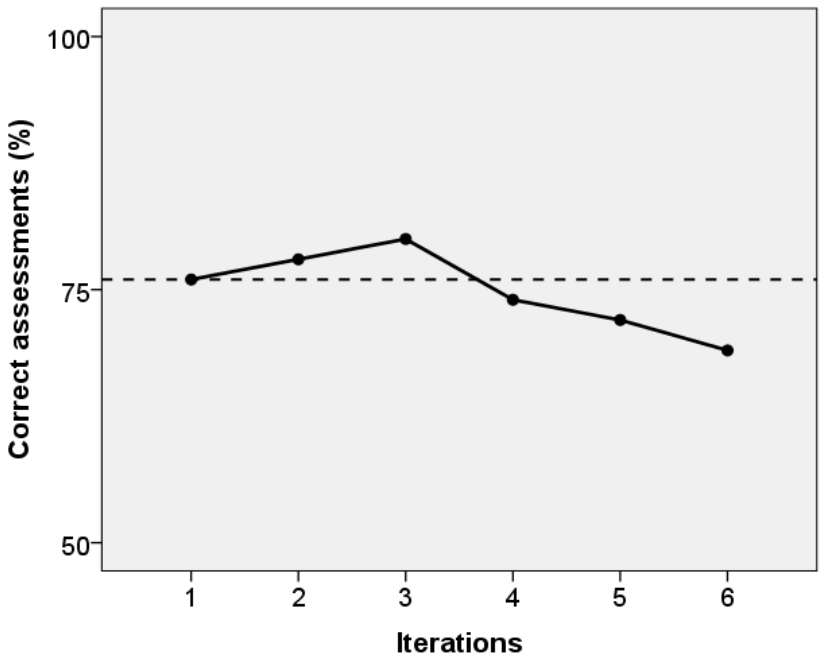


Figure 8: Percentage correct responses with varying iterations of RRI

The dotted line indicates the baseline of 76%, which was obtained by the best performing Random Indexing condition above. The optimal results are here achieved using 3 iterations of RRI.

Converting the best performing vector model (i.e. the 3_{rd} condition in the RI test and 3 iterations of RRI) to a probability distribution with different parameter settings yielded the following perplexities (Table 8).

Table 8: Parameters and perplexities in word space probability conversions

Preceding context (words)	λ	**Average perplexity**
50	300	6.977
50	100	7.059
50	10	7.451
5	100	6.996
20	100	7.048
50	100	7.059
100	100	7.062

In the table, the preceding context is the amount of words preceding any given word, whose vectors were included in the probability calculations of the given word, and λ is an arbitrary power factor (see Section 2.5.1 for details), and the average perplexities are log_2.

Linear and Geometric interpolation was then performed between this probability mass and the Kneser-Ney smoothed *n*-gram model, with different parameter settings (see Table 9).

Table 9: Parameter manipulations for Perplexity Calculations

Condition	λ	**Interpolation**
N-gram	*n/a*	*n/a*
Linear	0.2	Linear
Geo 1	7	Geometric
Geo 2	20	Geometric
Geo 3	50	Geometric

Where λ is a weighting parameter between the two probability masses (see Section 2.5.2 for details). Perplexity calculations were carried out on each of these conditions (see Figure 9).

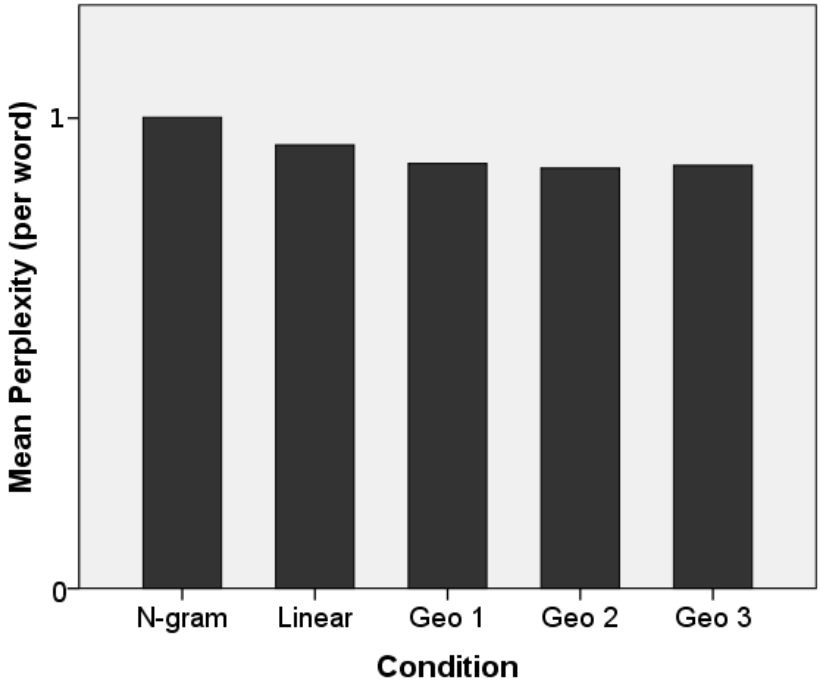


Figure 9: Perplexities with different types of interpolation, normalised to *n*-gram perplexity

When applying Linear Interpolation, the perplexities drop somewhat. When Geometric Interpolation is applied, the perplexities drop further. However, there only minor differences are caused by the varying parameter settings in Geometric Interpolation.

4.2 Results of Study II: EEG reading study

In this section the results from both conditions in the EEG reading study are presented.

4.2.1 Manipulated text

In the word replacement experiment, the replacements were divided into three arbitrary groups depending on the words' probabilities (high, medium, low). Within each of these groups, data points from the participants in the 400 ms post stimulus window were gathered. Each group contained approximately 30 data points per participant.

An ANOVA test of Between-Subjects Effects shows no significant differences between any of these groups ($F(2,53)=0.268$, $p=0.766$, see Figure 10).

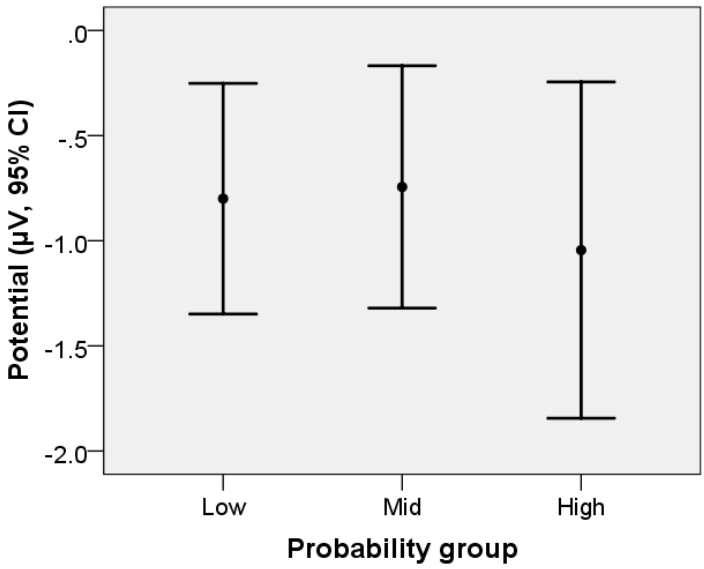
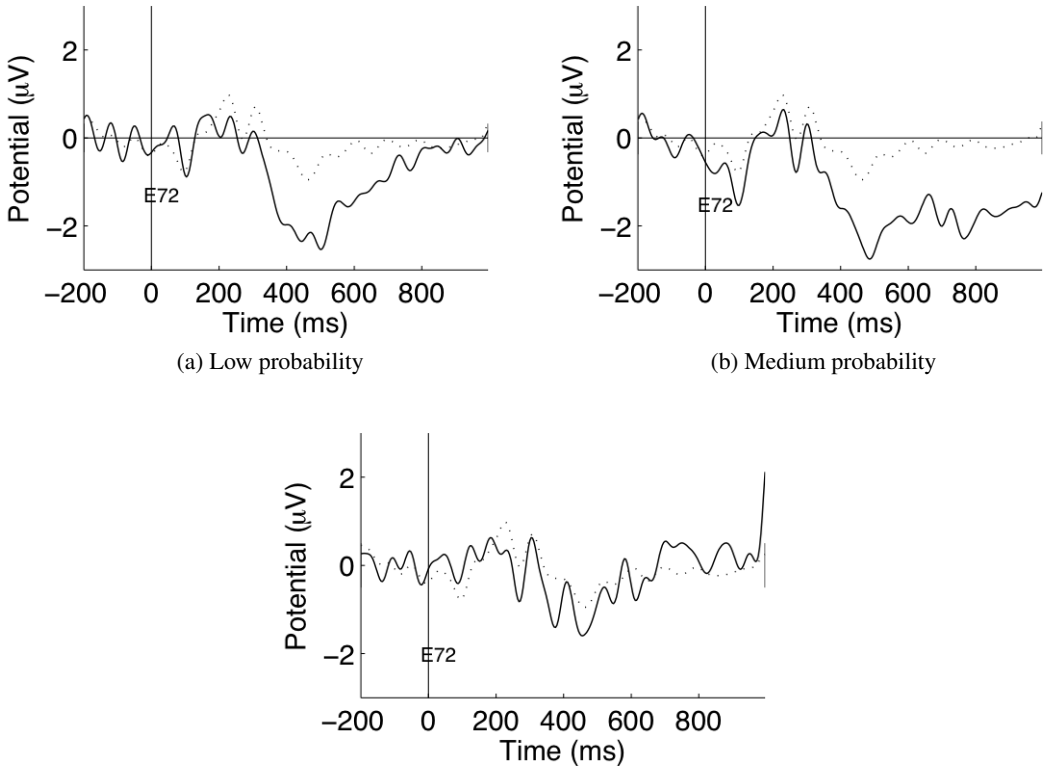


Figure 10: Word replacement probabilities

For the main analysis the electrode with maximum difference between content and function words where used (55), yielding insignificant differences. A later inspection of the waveforms for the manipulation shows larger N400 differences just one and two electrodes posterior from this electrode (see Figure 11). However, an ANOVA test of Between-Subjects Effects shows no significant differences between any of these groups (p=0.681).



(a) Low probability

(b) Medium probability

(c) High probability

Figure 11: Waveforms from low (a), medium (b) and high (c) probabilities (all represented by solid lines), compared to a baseline from all content words (dashed line).

4.2.2 Unchanged text

Regression analyses were carried out attempting to find correlations between the neural data within the 400 ms post-stimulus onset window and various measurements from the language models (see Table 10). Further categorisation of the data was carried out by dividing them according to parts of speech. Only the analysis of Nouns and Verbs will be included here, as analysing other categories yielded no further results.

Table 10: Regression analyses between EEG data and Entropy, KN probs., V probs. and Interp. probs.

POS	Measure	Linear R^2	Linear P	Quadratic R^2	Quadratic P	Cubic R^2	Cubic P
All	**Entropy**	0.000	0.739	0.000	0.385	0.001	0.112
All	**KN p**	0.000	0.772	0.000	0.958	0.001	0.842
All	**Vector p**	0.000	0.859	0.000	0.935	0.000	0.892
All	**Interp. p**	0.000	0.792	0.000	0.945	0.001	0.871
NN	**Entropy**	0.000	0.792	0.000	0.481	0.001	0.252
NN	**KN p**	0.000	0.563	0.000	0.814	0.000	0.873
NN	**Vector p**	0.000	0.732	0.000	0.973	0.000	0.732
NN	**Interp. p**	0.000	0.815	0.000	0.897	0.000	0.827
VB	**Entropy**	0.000	0.856	0.000	0.734	0.000	0.247
VB	**KN p**	0.000	0.648	0.000	0.924	0.001	0.646
VB	**Vector p**	0.000	0.354	0.000	0.746	0.000	0.758
VB	**Interp. p**	0.000	0.548	0.000	0.859	0.000	0.702

In the table, *KNp* is probabilities calculated by the Kneser-Ney smoothed *n*-gram model, *Vectorp* is probabilities calculated from the word space model, *Interp.p* is probabilities calculated through the best performing interpolation of these two models, and *Entropy* is the entropy, as calculated by the *n*-gram model. As the table shows, no significant correlations were found between the N400 response and either of the probability measures or entropy. Plotting the EEG data against entropy visualises this apparent absence of any patterns (see Figure 12).

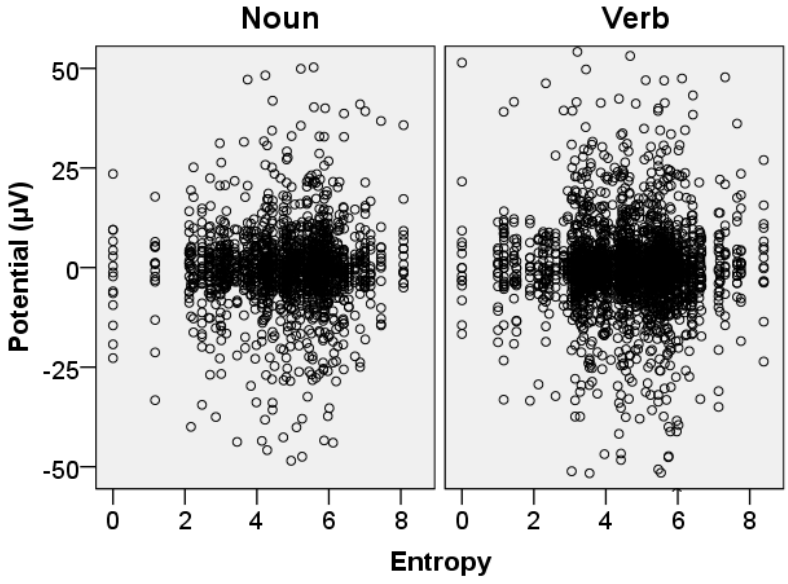


Figure 12: Entropy plotted against neural responses

The entropy calculations used here are calculated by the Kneser-Ney smoothed *n*-gram model. A distinct yet equally unclear image can be seen when plotting the EEG data against probabilities (see Figure 13).

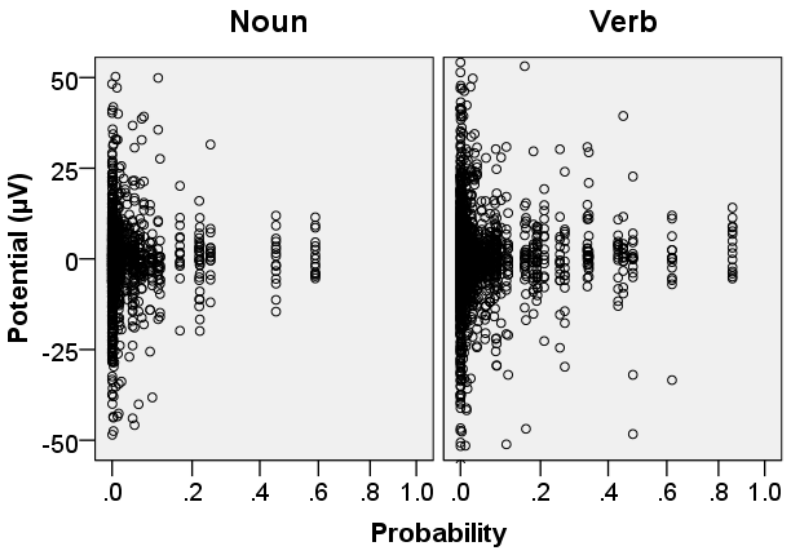


Figure 13: Probabilities plotted against neural responses

The probability calculations used here are calculated by the Kneser-Ney smoothed *n*-gram model.

A further analysis was carried out in which the data points from the natural reading study were divided into groups based on their probabilities according to the n-gram model. The grouping was done by grouping them according to quartiles (Q_n) when sorted by probabilities. Q_1 contains the first 25% of the data points – that is to say, the EEG responses from the 25% of the words with the lowest probabilities. Q_2 contains the data points from the point at which Q_1 ends, until the median, and so on until Q_4 (see Figure 14). Each group contained a total of approximately 3000 data points.

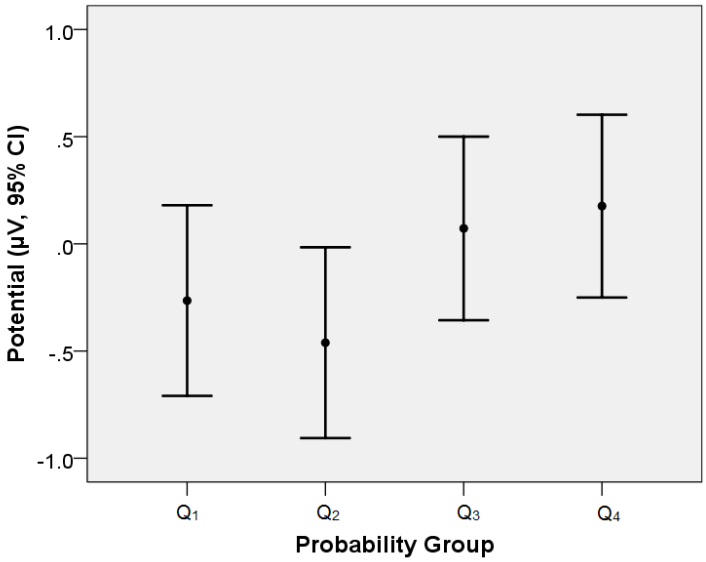


Figure 14: Neural responses divided by word probability groups

Using a two-tailed Post-Hoc ANOVA with the LSD method, significant differences were found between Q_2 and Q_4 *(p=0.042)*. A similar but weak tendency is found between Q_2 and Q_3 (p=0.092).

5 Discussion

5.1 Method discussion

In this section, the strengths and weaknesses of the methodology used in this essay are discussed.

5.1.1 The implementation of the language models

The language models that were implemented have some room for improvement. In particular these improvements pertain to the word space model, which was trained on a moderate amount of data gathered from blogs (approximately 1 billion words). A very simple step to take would be to increase the amount of training data used, and to also include different types of texts in the training data. However, this is not expected to improve the results more than marginally.

A more severe weakness of the word space model implementation of this essay can be found in the way the training data was used. Although the 200 most frequent words in the training data were not included due to them mainly introducing more noise to the word space, more optimal solutions can be found. If the training data were to be POS tagged, the word space model could easily be made to only include content words. This is beneficial when taking into consideration that function words have little to no semantic content.

Lastly, it is possible that using different types of language models would lead to better correlations with the N400 component. This is, however, also expected to be a factor that will not improve results more than marginally.

5.1.2 The experimental reading paradigm

Concerning the neural data gathered from the neural experiments appears, it does appear to be quite noisy. This can be attributed to the fact that the experimental setup might not have been optimal. Many participants reported difficulties in following the texts being presented. There can be many reasons for the participants to have experienced such difficulties.

The text without word replacements contained many instances of dialogues, making it particularly difficult to follow as there was no way of knowing which person in the text was speaking until the end of an utterance. Although such dialogues are commonplace in texts, it might be best to avoid them where possible in such experiments. However, if they must be included (such as in the case of recording EEG data for a whole corpus), an option might be to use colour-coding for the different characters in a text. This comes with other disadvantages, however, as certain neural responses might occur when the text changes colour.

The rate at which words were presented to the participants is another factor that might have led to the difficulties they experienced during the experiments. Although previous experiments have used similar presentation times (see e.g. Kutas and Federmeier (2011)), the presentation times used in this study (ranging from 100 ms to 200 ms for function words and from 500 ms to 700 ms for content words) might still be to short. This seems likely if it is taken into consideration that many reading experiments involve participants reading one sentence at a time, without any contextual context to keep track of (e.g. Kutas and Hillyard (1984)). The experiments in this essay, however, do include a relatively large contextual context. Adding to the fact that there were relatively short pauses between sentences and paragraphs, it seems that the presentation and pause times need to be tweaked and optimised before further research is undertaken.

Furthermore, the fact that words were presented one at a time might have led to difficulties when following a natural text. This can be attributed to that readers will often prefer to reread certain passages or words (words with low occurrence frequencies in particular), pause on certain words or some times skip certain passages (see Just and Carpenter (1980) for a more detailed account). However, allowing such (natural) behaviour in an EEG recording context is problematic for two main reasons. Firstly, it is necessary to know what word a participant is looking at at any given time in order to know which ERP was a response to what stimulus. Fortunately, this is a problem that can be solved by employing

techniques such as eye-tracking. Secondly, muscular activity occurring when a participant moves their eyes would lead to unwanted artifacts in the EEG data. Although such artifacts can be removed with automatic procedures, they end up adding more noise to the data.

Additionally, the fact that participants were only offered one break (between Experiments II and III) might have led to an increased overall difficulty. This might also be hard to remedy however, as taking longer pauses in the middle of a text might lead to the participants forgetting the content of what they read prior to the pause.

5.2 Results discussion

5.2.1 Discussion of Study I

The parameter manipulation for dimensionality reductions through Random Indexing show that, as expected, a higher dimensionality improves results. This, in combination with a higher amount of elemental vector alterations, yielded the best results in the test of semantic coherence. This is a highly plausible result, as higher dimensionalities should increase the accuracy of the word space. This is due to the simple fact that reducing the original word space (with approximately 50,000 unique words, each with its own feature dimension) to 2000 dimensions is a better compromise than reducing it to 200 dimensions. As for the context window, no major manipulations were attempted. However, a shorter context window of 200 words allowed the model to perform better than when using a larger context window of 500 words. This can be explained by increased amounts of noise in the vectors as the context window extends. As the context window extends, any given word is likely to be affected by the vectors of words that are decreasingly related. In this case, with a context window of 500 words, a word in one blog post would very possibly have its vector affected by words in other blog posts. It goes without saying that a word probably is not particularly closely related to a word occurring in a different text.

Increasing the iterations in the Reflective Random Indexing approach shows results comparable to those presented by Cohen et al. (2010). Optimal performance occurs when the algorithm has completed 3 iterations. Following this, the model's performance drops drastically.

Analysis of the results from the experiments dealing with transformation of word space distances into probabilities reveals that manipulating the parameters does not greatly affect the performance of the model's predictive powers. Certain patterns can be seen, however. Perplexities from the RRI model increase when the preceding context, from which the centroid vector is calculated, is increased. This is most likely a result of increased amounts of noise as more and more vectors are added to the context. The increased amounts of noise end up making all words more or less equally (im)probable. In turn, this provides very little useful information when it comes to predicting what the next word will be. If all possibilities are equally probable, it is almost just as well to simply make a random guess of what the next word is. Due to this, the optimal parameter setting might indeed be to only include a very restricted number of the preceding content words – perhaps only nouns – and certainly only content words. Including words with little or no semantic content, as previously mentioned, is not likely to provide much relevant information.

Interpolation of the two language models shows that Linear Interpolation outperforms the *n*-gram model, which in turn is outperformed by Geometric Interpolation. These results are further supported by the similar findings in Coccaro and Jurafsky (1998) and Wandmacher and Antoine (2007). Manipulation of the different parameters did not seem to affect results more than marginally.

5.2.2 Discussion of Study II

The statistical analysis of the experiments with word replacements shows that no significant differences could be found between the different probability groups, when analyzing electrode no. 55. Higher and lower probabilities did not significantly affect the N400 response elicited in the participants. A later inspection of the waveform diagrams from two electrodes posterior from this electrode (no. 72) did, however, show seemingly clear effect, where low and medium probabilities elicit a larger negative response than the high probability condition. However, this effect was also found to be statistically insignificant.

Although not statistically significant, it can be argued that a certain tendency is visible in the waveform diagrams (see Figure 11). The lack of significance can be explained by large individual variations between participants in the experiment. The noisiness of the EEG data is most likely also a contributing factor.

As mentioned, the EEG data gathered through the experiments in this essay appears to be too noisy for any correlations to be found. The null results could be attributed to the possibility that there, in fact, is no correlation between the models and the neural responses. However, as previous research (see e.g. Hahn (2011)) has shown that such correlations do exist (however indirectly between language models and cloze probabilities), the point of view more likely to be accurate is that the largest issue in this study has been the gathering of neural data through the reading experiments.

Separating the EEG data into groups by probabilities did, however, show significant differences between probability groups. Specifically, significant differences were found between the Q_2 (the 2_{nd} to lowest probability group) and Q_4 (the highest probability group). The differences show significantly larger N400 responses as word probabilities (calculated from the interpolated language model used in this study) decrease. Or, inversely, as the word probabilities increase, the negativity in the 400 ms post-stimulus window decreases. These results support the initial hypothesis in that probabilities calculated from a language model can successfully predict neural responses. The significance levels are, however, quite low (p=0.042). This may be a result of the somewhat noisy data, or might simply indicate that the hypothesis itself is inaccurate. However, considering the results of previous research, the most likely source of error is the experimental paradigm employed in this essay (see e.g. Hahn (2011); Kutas and Federmeier (2011)).

Furthermore, no significant differences were found between Q_1 (the lowest probability group) and any of the other groups. This can be interpreted as contradicting the initial hypothesis, as this group does not differ from any of the other groups. However, an alternative explanation can also be found. When calculating probabilities on whole texts with material that might not be included in the training material, a large amount of words is likely to receive probabilities very close to zero (in this case $p \approx 0.1 \cdot 10^{-15}$). Although these probabilities might in some cases be accurate, they are more likely a result of zero-occurrences in the training data. Such zero-occurrences might often be due to compound nouns, which might be semantically appropriate, but absent in the training data used for the language model. Words that are simply uncommon might also be in this group. Such words might be perceived by a reader as non-words, which have been found to elicit other neural responses than N400, such as P3b as mentioned in Section 2.6 (see e.g. Kutas and Federmeier (2011)). As detailed in section 2.4.1, however, these zero-occurrences are compensated for by an advanced smoothing algorithm called Kneser-Ney smoothing (Kneser and Ney, 1995). Although this algorithm has been shown to outperform most other smoothing algorithms in normal NLP applications (Chen and Goodman, 1998), it is possible that this approach is not optimal for this particular application. A smoothing algorithm or language model that detects compound nouns and compensates probabilities accordingly might prove more successful in this case.

The goal of the experiments carried out in this essay was to investigate whether or not any significant correlations could be discovered between language models and neural responses. The neural response focussed on in this study, N400, was used as the dependent variable due to it being highly manipulative and affected by semantic incongruities. However, the results showed that no significant correlations were found between any of the measures derived from the language models used in this study. One might draw the conclusion that this indicates that correlations between the measures and the neural responses do not exist. Such a decision would most likely be premature. The large amount of imperfections in the experimental setup have most likely led to too high amounts of noise, making new EEG experiments necessary before any real conclusions regarding this can be drawn.

5.3 Future research

The production of a fully fledged EEG corpus is a suggested path for future research. When successful, an EEG corpus can be a highly useful tool for psycholinguists and others with similar research interests. This, however, is a large project to undertake. As EEG data often is noisy, analysis is often done on a group level. For this to be possible, a large amount of participants is needed. In addition, in order to allow for large scale analysis of for example various language structures, a large amount of data needs to be gathered per participant.

In order for this to be manageable and in order to make sure this resource can be used to its fullest potential, a novel tool for performing searches in this corpus is necessary. Such a tool could include functions for various searching criteria. For example, a researcher might want to search for all points in the corpus where a noun with high probability elicited a large N400 response. Such a large scale tool would indeed provide unique material for researchers. Furthermore, it could provide a good basis for further research. If, for example, a researcher discovers that at certain points in the corpus, any noun preceded by the context *X* elicits a large N400 response, a separate experiment investigating this factor and manipulating the context as an independent variable could be conducted.

6 Conclusions

At the beginning of this essay, four goals were presented.

1. *How are semantically and syntactically oriented language models best integrated?*
   The results indicate that the best way of integrating these models is through Geometric Interpolation. As for the parameters used in this interpolation, no major differences could be found when altering them.

2. *Can a language model's probability estimates of replaced content words predict the amplitude of N400 responses?*
   Although some differences in N400 amplitude between probability groups might be present, no significant differences were found. However, this is most likely due to a flawed experimental paradigm. No conclusion can be drawn.

3. *How does a language model's probability estimates of content words in an unchanged text correlate with the N400 component?*
   No significant correlations could be found between probabilities calculated by any model used in this study and N400 amplitude. However, when dividing words into groups according to their probabilities, weak significant differences were found. This indicates that, with clearer data and perhaps better language models, correlations might be found.

4. *What practical considerations are important when constructing a multimodal corpus with EEG data? Is this a worthwhile project for psycholinguists?*
   Some of the practical considerations uncovered here include optimising the experimental paradigm for longer reading sessions where unmanipulated texts are read. One suggestion is to use eye-tracking so that participants may read the text in a more natural way. This, however, has certain disadvantages as eye-movements would cause artifacts in the EEG data.

   The fact that the limited results that were retrievable do support the hypothesis that lower word probabilities elicit larger N400 responses, indicates that this is indeed a potentially worthwhile path to continue on for psycholinguists.

References

Bellegarda, Jerome R. Statistical language model adaption: review and perspectives. *Speech Communication*, 42:93–108, 2004.

Bloom, P.A. and Fischler, I. Completion norms for 329 sentence contexts. *Memory & Cognition*, 8(6): 631–642, 1980.

Chen, Stanley F. and Goodman, Joshua. An empirical study of smoothing techniques for language modeling. *Harvard University, Center for Research in Computing Technology*, TR-10-98:1–63, 1998.

Coccaro, Noah and Jurafsky, Daniel. Towards better integration of semantic predictors in statistical language modeling. In *Proceedings of the International Conference on Spoken Language Processing*, volume 6, pages 2403–2406, 1998.

Cohen, Trevor; Schvaneveldt, Roger, and Widdows, Dominic. Reflective random indexing and indirect inference: A scalable method for discovery of implicit connections. *Journal of Biomedical Informatics*, 43:240–256, 2010.

Deng, Yonggang and Khudanpur, Sanjeev. Latent semantic information in maximum entropy language models for conversational speech recognition. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, volume 1, pages 56–63, 2003.

Deza, Michel M. and Deza, Elena. *Encyclopedia of Distances*. Springer Berlin Heidelberg, 1st edition, 2009.

Federmeier, Kara D.; Wlotko, Edward W.; Ochoa-Dewald, Esmeralda De, and Kutas, Marta. Multiple effects of sentential constraint on word processing. *Brain Research*, 1146(4):75–84, 2007.

Hahn, Lance W. Measuring local context as context–word probabilities. In *Behavior Research Methods*, volume 1554-351X, pages 1–17, 2011.

Harris, Z.S. Distributional structure. In Katz, J.J., editor, *The Philosophy of Linguistics*, pages 26–47. Oxford University Press, 1985.

Hart, J. and Kraut, M.A. *Neural basis of semantic memory*. Cambridge Univ Pr, 2007.

Hörberg, Thomas; Koptjevskaja-Tamm, Maria, and Kallioinen, Petter. The neurophysiological correlate to grammatical function reanalysis in swedish. *Language and Cognitive Processes*, 2012.

Jurafsky, Daniel and Martin, James H. *Speech and Language Processing*. Pearson Education Inc., 2nd edition, 2009.

Just, Marcel A. and Carpenter, Patricia A. A theory of reading: From eye fixations to comprehension. *Psychological review*, 87:329–354, 1980.

Kanerva, P.; Kristofersson, J., and Holst, A. Random indexing of text samples for latent semantic analysis. In *Proceedings of the 22nd annual conference of the cognitive science society*, volume 1036. Citeseer, 2000.

Kaski, Samuel. Dimensionality reduction by random mapping: Fast similarity computation for clustering. In *Proceedings of International Joint Conference on Neural Networks*, volume 1, pages 413–418, 1998.

Khudanpur, Sanjeev and Wu, Jun. A maximum entropy language model integrating n-grams and topic dependencies for conversational speech recognition. In *Proceedings of Acoustics, Speech, and Signal Processing*, pages 553–556, 1999.

Kneser, Reinhard and Ney, Hermann. Improved backing-off for m-gram language modeling. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 1, pages 181–184, 1995.

Kutas, Marta and Federmeier, Kara D. N400. *Scholarpedia*, 4(10):7790, 2009.

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

Kutas, Marta and Hillyard, Steven A. Reading senseless sentences: Brain potentials reflect semantic incongruity. *Science*, 207(4427):203–205, 1980.

Kutas, Marta and Hillyard, Steven A. Event-related brain potentials to grammatical errors and semantic anomalies. *Memory & Cognition*, 11(5):539–550, 1983.

Kutas, Marta and Hillyard, Steven A. Brain potentials during reading reflect word expectancy and semantic association. *Nature*, 307(5947):161–163, 1984.

Landauer, Thomas K. and Dumais, Susan T. A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction and representation of knowledge. *Psychological Review*, 104(2): 211–240, 1997.

Landauer, Thomas K.; Foltz, Peter W., and Laham, Darrell. An introduction to latent semantic analysis. *Discourse Processes*, 25(2–3):259–284, 1998.

Landauer, Thomas K.; McNamara, Danielle S.; Dennis, Simon, and Kintsch, Walter. *Handbook of Latent Semantic Analysis*. Lawrence Erlbaum Associates, Inc., 1st edition, 2007.

Mitchell, Jeff and Lapata, Mirella. Vector-based models of semantic composition. In *Proceedings of Association for Computational Linguistics*, volume 1, pages 236–244, 2008.

Mitchell, Jeff and Lapata, Mirella. Language models based on semantic composition. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, volume 1, pages 430–439, 2009.

Navas, I.; Sanz., I.; Aldana, J.F., and Berlanga, R. Automatic generation of semantic fields for resource discovery in the semantic web. In Andersen, Kim; Debenham, John, and Wagner, Roland, editors, *Database and Expert Systems Applications*, volume 3588 of *Lecture Notes in Computer Science*, pages 706–715. Springer Berlin / Heidelberg, 2005.

Papadimitriou, Christos H.; Raghavan, Prabhakar; Tamaki, Hisao, and Vempala, Santosh. Latent semantic indexing: A probabilistic analysis. In *Proceedings of the 17th Association for Computing Machinery Symposium on the Principles of Database Systems*, volume 1, pages 169–168, 1997.

Pedersen, Ted; Patwardhan, Siddharth, and Michelizzi, Jason. Wordnet::similarity - measuring the relatedness of concepts. In *Proceedings of the Nineteenth National Conference on Artificial Intelligence*, pages 1024–1027, 2004.

Sahlgren, Magnus. *The Word-Space Model: Using distributional analysis to represent syntagmatic and paradigmatic relations between words in high-dimensional vector spaces*. PhD thesis, Department of Linguistics, Stockholm University, 2006.

Sussna, Michael. Word sense disambiguation for free-text indexing using a massive semantic network. In *Proceedings of the second international conference on Information and knowledge management*, pages 67–74. ACM, 1993.

Wandmacher, Tonio and Antoine, Jean-Yves. Methods to integrate a language model with semantic information for a word prediction component. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, volume 1, pages 506–513, 2007.

Appendices

List of Tables

List of Figures

List of Algorithms

Stockholms universitet/Stockholm University
SE-106 91 Stockholm
Telefon 08 - 16 20 00
www.su.se

Stockholms universitet