

Bilingual lexical representation
Extending the Bilingual Single Network

Erik Lormans

February 26, 2010

BA thesis Artificial Intelligence
Author: Erik Lormans
Student number: 0513474
Supervisors: Ton Dijkstra & Ida Sprinkhuizen-Kuyper
Radboud University Nijmegen
the Netherlands
eriklormans@student.ru.nl
February 26, 2010

Bilingual lexical representation

Extending the Bilingual Single Network

Abstract

Many studies have been done in the domain of bilingual word recognition. One important issue is whether bilinguals who read consider words from both their lexicons at the same time, or only one of them. This issue has been considered for example in the Bilingual Single Network model by Thomas (1997b). This researcher looked at how distributed models of monolingual word processing could be extended to the bilingual domain. With this computer model, he was able to illustrate the language-independence of lexical representations and interference effects.

In the present project, I have extended this model to include natural language as input without using language coding units. The results show a facilitation effect for cognates and an inhibition effect for false friends and translation equivalents. These results are completely in line with the dominant theoretical view, i.e., that bilinguals possess a non-selective lexical access procedure to their mental lexicon.

1 Introduction

Ever since the 50's, research has investigated if bilingual readers possess two separate lexical stores, one for each language, or one big "bilingual" lexical store. The broader goals of this research has been to find out what the underlying mechanisms are that allow access to and selection from the lexical databases of the bilingual. The experimental, neuropsychological, and computational techniques in this area of research have developed over the years to the point at which they allow researchers to answer some of these questions. A number of computational models, localist- or distributed- connectionist of nature, have already achieved considerable success in answering some of these questions about bilingual processing and memory mechanisms.

1.1 Literature review

A well-known localist model is the Bilingual Interactive Activation (BIA) model (Dijkstra and Van Heuven, 1998; Van Heuven et al., 1998), which has shown to be quite successful in explaining and modelling the process of lexical selection. It yields accurate simulations of interlingual orthographic neighborhood density

effects, cross-language masked orthographic priming effects and interlingual homograph recognition experiments. A debated aspect of the BIA model lies in its use of language tags, representations for language membership. For instance, according to this model the English word WORK is linked to an English language tag.

Another successful localist connectionist model is the SOPHIA model (SOPHIA is an acronym of Semantic, Orthographic, and Phonological Interactive Activation model) (Van Heuven and Dijkstra). The SOPHIA model is an extended implementation of the BIA+ model (Dijkstra and Van Heuven, 2002) with a few extra types of representations (among other things semantics and phonology). SOPHIA is able to simulate a number of effects in monolingual visual word recognition, among which are priming effects, the effects of consistency between phonological and orthographic codes, pseudo-homophone effects and the role of neighborhoods. SOPHIA is especially good at mimicking the facilitatory effects of body neighbors, neighbors that share their orthographic rime with the target word.

In comparison with the BIA model, the localist Bilingual Model of Lexical Access (BIMOLA) for auditory word recognition (Léwy and Grosjean), claims that different theoretical assumptions are necessary for the simulation of empirical effects in visual and auditory modalities. In particular, it considers between-word activation as more important than between-word inhibition. In this paper, we will not consider this model any further.

Thomas (1997a,b, 1998) considered how distributed models of the monolingual language system could be extended to the bilingual domain. Thomas decided to explore the Single Network hypothesis, which assumes that the interference effects are the results of saving the two languages in one single representational resource. In his Bilingual Single Network (BSN) model he was able to illustrate the language-independence of lexical representations and interference effects. Concerning the language-independence effects, the presentation of interlingual homographs to the model led to within-language frequency effects and there was an absence of long-term priming effects for translation equivalents between the languages. Interlingual homographs are words from different languages that are spelled identically but are different in their meaning or pronunciation. For example, “room” is an interlingual homograph between English and Dutch. Cognates are words that, usually as a consequence of a common origin, have identical or almost identical spelling and meaning, e.g. the English “milk” and the Dutch “melk”. Concerning the interference effects, the model demonstrated slower processing effects for interlingual homographs in comparison to cognate homographs. Note that both types of items are translation equivalents, i.e., word pairs that have

the same meaning. However, for the purpose of this thesis, we will define Translation equivalents as words that do not share the same orthography, e.g., bike in English and fiets in Dutch.

Besides the language-independence and interference effects, the model also demonstrated a facilitatory effect for cognates in L2 relative to L1 in the unbalanced network. In addition, the model was able account for cross-language semantic priming effects through the use of a common semantic output layer. It should be noted though, that because the model received two artificial languages as input, it would require language tags to know which was which. Because of these language tags the model was able to discriminate and identify L1 and L2, without these language tags this might not have been the case.

With the distributed Bilingual Simple Recurrent Network (BSRN) model (French, 1998; French and Jacquet, 2004) researchers tried to determine whether word order information only would be sufficient to discriminate between two languages. They assumed so, because, in sentences, people are usually capable of predicting the language membership of the next word, based on the previous words of the sentence. The result of these simulations was that as long as the language-switch occurred with a sufficiently low probability (0,1%), word order (a combination of multiple preceding words of one language) alone provided sufficient information to lead to distinct representations of the words of the two languages. The BSRN also was in line with the hypothesis that bilingual memory is organised in one single distributed lexicon.

Li and Farkas (Li and Farkas) tried to account for bilingual word production as well as bilingual word comprehension. To do so, they used the distributed Self-Organizing Connectionist Model of Bilingual Processing (SOMBIP). Simulations with this model showed that it could develop meaningful lexical-semantic categories via self organizing processes. The model was able to account for a variety of priming- and interference effects on the basis of associative pathways between phonology and semantics in the lexicon. It was also able to simulate the lexical representations of bilinguals with different levels of skillfulness of L2 and working memory capacity. On top of all this, the model did not use language tags.

1.2 Research goals

For my bachelor-project we focussed on the BSN-model of Thomas (1997b). As mentioned before in the literature review, this model had some flaws. The BSN-model received items from two artificial languages as input, which necessitated the

use of language tags to separate the languages from one another. We believe the use of language tags can have too big of an influence on bilingual word recognition. Especially in the context of the BSN-model, we think language tags have played a role of too much importance. A second problem is the generalizability of the model. Although Thomas booked some nice results with his model, generalizability is limited because of the use of artificial languages.

As a consequence, we think it is important to reproduce the results of Thomas' experiment using words from natural languages as input and without the use of language tags. To study these points, we will recreate the BSN-model of Thomas. Instead of using 3-letter non-words, our model will receive 4-letter words from a natural language as input. We will use English and Dutch as natural languages for this purpose. Therefore, the model will be built according to the principles of the BSN-model, but in a structural view it will be somewhat different.

We predict differences in recognition time for cognates, interlingual homographs and translation equivalents as compared to control words. This would imply that these words are processed in a different way than the control words, a finding that would be in line with the hypothesis of non-selective access to the bilingual lexicon and empirical studies.

In the rest of this chapter, we will first discuss the BSN-model of Thomas in more detail. This is followed by a discussion of our own model, the extended BSN-model, leading to a description of the different simulations we performed, the results of these simulations and some general conclusions.

2 The BSN-model

Thomas seeks evidence of between language similarity effects in this network. He focusses on differences that arise between word forms that exist in both languages compared to those that only exist in one language. Thus, he focusses on the behaviour of a single network trained to generate the meanings of the words in two languages.

The first step Thomas took was to find a training set of stimuli to address the problem of bilingual lexical representation. For simplicity, he created two artificial languages, which were called A and B, to be used as input for the model.

Each language consisted of three-letter words using a ten-letter alphabet. This was done in order to scale down the size of the training set. The consonants that

were used are: b, f, g, s and t. The vowels that were used are: a, e, i, o and u. There were 96 words in each language, divided into 40 words that are legal in both languages, 40 words that are only legal in that language, 8 cognate homographs and 8 non-cognate homographs.

A binary vector of 120 units was used to generate the meanings for the words (Semantic Feature Units). Furthermore, a level of 10% sparseness was used to make sure that each feature had a 10% probability of being active in a given meaning. Thomas also ascertained that cognate homographs were assigned the same meaning for the occurrence of the word form in each language and that non-cognate homographs were assigned a different meaning for the word form in each language. The words that only existed in one language were referred to as Singles. Translation equivalents were created by randomly pairing these Singles together, provided that they were both Singles and of the same frequency. These words were then assigned the same meaning.

Bilinguals can selectively access lexical information from either of their two languages. Thomas argued that if two languages are to be stored in a single network, information specifying language membership must therefore be associated with each item. Eight units were used to code the language to which the word belonged. The first four units were active if the word belonged to language A, the last four units were active if the word belonged to language B. Language coding was added to both the input and the output vectors. The reason for adding language coding to the output vectors was to emphasize the representation salience of language, and it gave the model the potential to classify ambiguous inputs.

A three layer feedforward network (Figure 1) was used to learn the mappings between the orthographic and semantic codes. The network consisted of an input layer of 38 units; 30 units for the Orthographic Input (10 units for each letter) and 8 units for the Language Coding Unit (LCU). A hidden layer of 60 units and an output layer of 128 units; 120 units to represent the meaning of the word (Semantic Feature Units) and 8 units for the LCU.

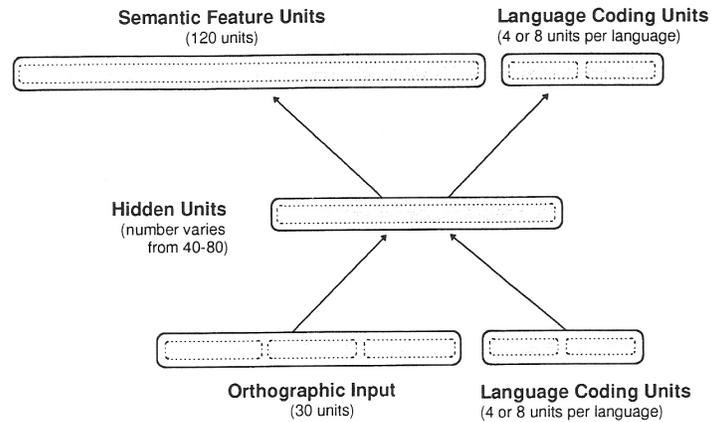


Figure 1: The BSN-model of Thomas

3 The extended BSN

As mentioned before, our goal for this bachelor-project was to try to reproduce the results of Thomas' experiment but using natural language as input and without using language tags. In order to achieve this goal, we built our own network based on the BSN model of Thomas, with some slight structural deviations. We call this model the extended BSN model (eBSN) (Figure 2).

We wanted to extend the BSN model to natural language. This could simply be done by using words from natural languages as input data for the model instead of words from artificial languages. The natural languages we used are Dutch and English. We chose to use four-letter words for the eBSN model instead of three-letter words. Instead of having an Orthographic Input Unit of 30 units to represent the letters of the word, the input layer of the eBSN model consists of an Orthographic Input Unit of 40 units, 10 units for each letter, and a LCU.

We also made some adjustments to the LCU. Instead of using 8 units, we only used 1. This unit can have a value of "0", meaning that the word belongs to the Dutch language, or a value of "1", meaning that the word belongs to the English language. Just like Thomas did in the BSN model, we also separated the LCU from the Orthographic Input Unit in the eBSN model. This way we were able to create a lesion between the LCU and the hidden layer, allowing me to turn the LCU on and off. This enabled me to compare results of simulations with the use of language tags to those of simulations without the use of language tags.

Furthermore, the eBSN model had 80 units in the hidden layer instead of the 60 Thomas used. The output layer of the eBSN model consisted of the Semantic Feature Unit with the same structure as the Semantic Feature Unit in the BSN model of Thomas. It also included a LCU. While in this case there were no ambiguous inputs, we choose to keep the LCU in the output layer so that for cognates we could tell whether the Dutch or the English word was being processed.

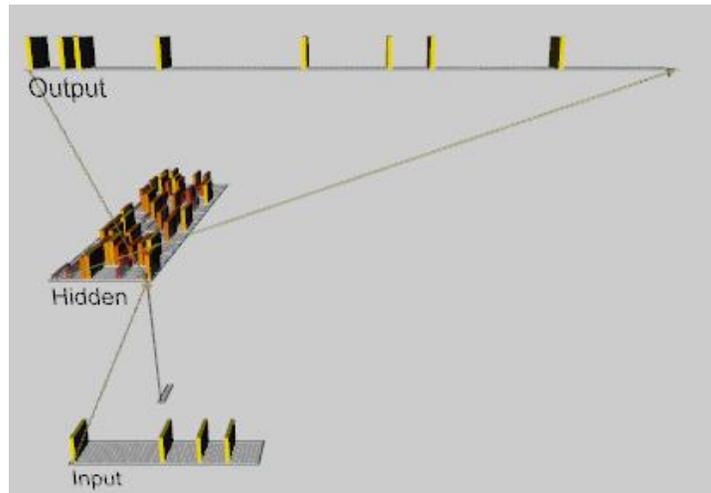


Figure 2: The eBSN-model, the bottom two layers are the input layer consisting of the orthographic input (40 units) and the LCU (1 unit). The middle layer is the hidden layer (80 units). And the top layer is the output layer consisting of the Semantic Feature Unit (120 units) and the LCU (1 unit). The coloured bars on top of each layer represent the amount of activation a unit is receiving.

3.1 Emergent

We created the eBSN model in a software environment called Emergent. Emergent is completely open source, which means it is freely available on the internet. It can be found at: http://grey.colorado.edu/emergent/index.php/Main_Page. Emergent includes a full GUI environment for constructing networks and the input/output patterns for the networks to process, as well as many different analysis tools for understanding what the networks are doing. Emergent is written in C++ and can be used on different operating systems (Windows, MacX, Linux/Unix). It is visually very strong. One can look over the network one created in 3D mode, and one can create visual graphs, which makes it easier to understand what is going on in the network. Emergent supports the multiple different learning and processing algorithms for neural networks, from backpropagation

to more biologically-based algorithms. The learning algorithm we used for our simulations is Leabra Hebbian Learning. Leabra stands for Local, Error-driven and Associative, Biologically Realistic Algorithm, and it implements a balance between Hebbian and error-driven learning on top of a biologically-based point-neuron activation function with inhibitory competition dynamics. For a more in-depth explanation of Leabra Hebbian Learning, we advise you to visit <http://grey.colorado.edu/emergent/index.php/Leabra>.

Looking at the screenshot of Emergent in Figure 3, one will see that on the left there is a browser panel with a “tree” of objects in the simulation (including the network and the input/output data et cetera). In the middle is the edit panel. It can display different objects depending on the selected tabs at the top and what is currently selected in the left browser panel. In this edit panel the build-in documentation can be shown, and depending on what is selected in the left browser panel, the value of the variables can be edited. On the right there is the view panel, which shows 3D displays of various simulation objects, including the network, input/output patterns, graphs of results et cetera.

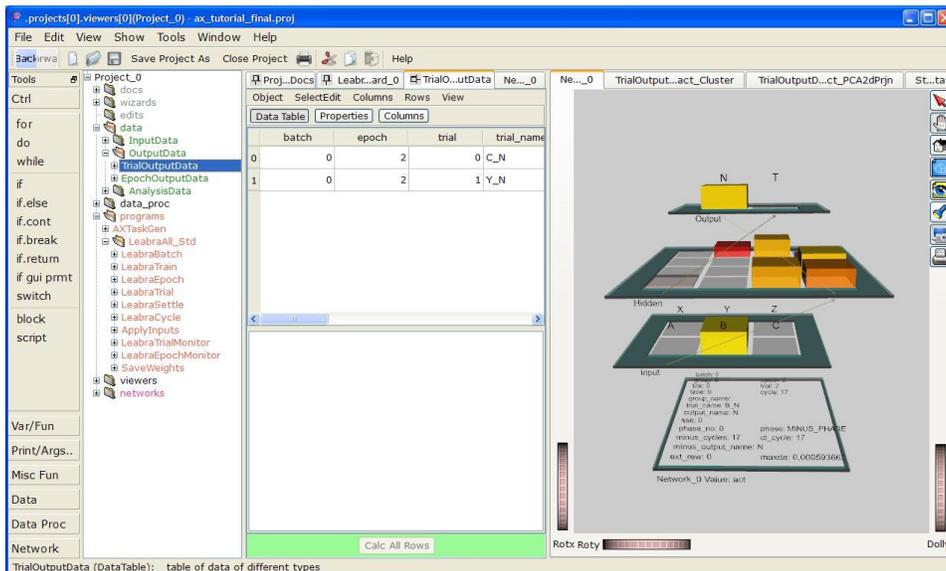


Figure 3: Screenshot of Emergent

3.2 Creating the input and output patterns

As mentioned before, Emergent provides the opportunity to create the input/output patterns in the program itself. However, this can also be done outside of the Emergent program as long as one creates a file that can be read by Emergent, containing the right input/output patterns. We decided to create the input/output patterns outside of Emergent and used Emergent only for the simulation aspect. The program files we wrote for this are all written in C++.

The first program we wrote receives the file containing the most frequent words along with their frequency for the Dutch or the English language as input. It then returns a file containing, for that language, only the words that are made up from the letters that are part of the ten-letter alphabet Thomas used. It essentially filters out the words containing letters that are not in this ten-letter alphabet.

Next, we wrote a program that creates the matching input vectors for all these words. This was based on the position of the letter in the ten-letter alphabet “bfgstaeiou” and used ten units to encode one letter of the word. It reads the words from the input file, which is the file containing the 10-letter alphabet words, and returns a file containing the words along with their matching input vector. E.g. it reads the Dutch word “auto” and returns “auto” and the input vector “0000010000000000000100001000000000000100”. In this input vector the first 40 units represent the encoding for the word, the last unit represents the value for the LCU.

This program was followed by a program that creates a random output vector for each word. For each word it receives as input, it creates a binary vector of 120 units. In this vector, every unit has 10% chance getting the value “1” and 90% chance of getting the value “0”. In this way, the feature units have the same level of sparseness as in the BSN-model of Thomas. The last unit in this binary vector is hard-coded, because this is the LCU. We included the LCU in the output layer to classify ambiguous input.

After we had run these programs, we had to verify that cognate homographs were assigned the same meaning for the occurrence of the word-form in each language. Because the binary output vectors were randomly created, this was not yet the case. We manually replaced the binary output vector of the cognate homographs in one language by the vectors of the cognate homographs in the other language, keeping in mind that we gave the correct value to the LCU. The same goes for the translation equivalents, we had to make sure that these words were given the same meaning. For the false friends we had to make sure that they had

a different meaning. However, because of the random assignment of meanings to words we did not encounter a situation in which two false friends were given the same meaning.

All this information then was combined into one file that could be read by Emergent. We wrote a program that takes as input this input data file, the file with the input vectors, and the file with the output vectors, and returns a data file as output that can be read by Emergent. All this program does, is copy the lines that can remain unchanged, and replace the information in the input and output fields with the input and output vectors we created.

These programs can be run for one language at a time. We first ran them for the Dutch language with the Dutch input files, setting the hard-coded value for the LCU to “0”. Then we ran them again with the English input files, setting the hard-coded LCU value to “1”. We also created a program that creates a data file which can be read by Emergent, in which the the words from both languages are mixed together in a 70-30 ratio. In this file every Dutch word appears 70 times, while every English word appears 30 times. We will explain in the simulations section why we used this 70-30 ratio.

4 The simulations

A total of four simulations were performed. In each simulation the performance of the eBSN-model on control words, cognates, false friends and translation equivalents was tested. For an overview of the words used per category see appendix A.

The simulations differed in their input they received. They received either only Dutch words, only English words, words from both languages with LCU or words from both languages without LCU. In each simulation the network had to learn a mapping from the orthographic input to the semantic output. In every simulation tenfold cross validation was performed, which means that every simulation was performed ten times and then averaged over the results. By comparing the results of the simulations with bilingual input to those of the simulations with monolingual input, we were able to draw conclusions with regard to lexical representation in bilinguals.

4.1 The Dutch simulation

This is a simulation to acquire results which can be used as a baseline. The goal of this simulation was to find out if the model is able to learn. Would we obtain a reasonable learning curve? Because only words from the Dutch language are being offered to the model here, the graphs of these results should look very similar. Figure 4 will help explain why.

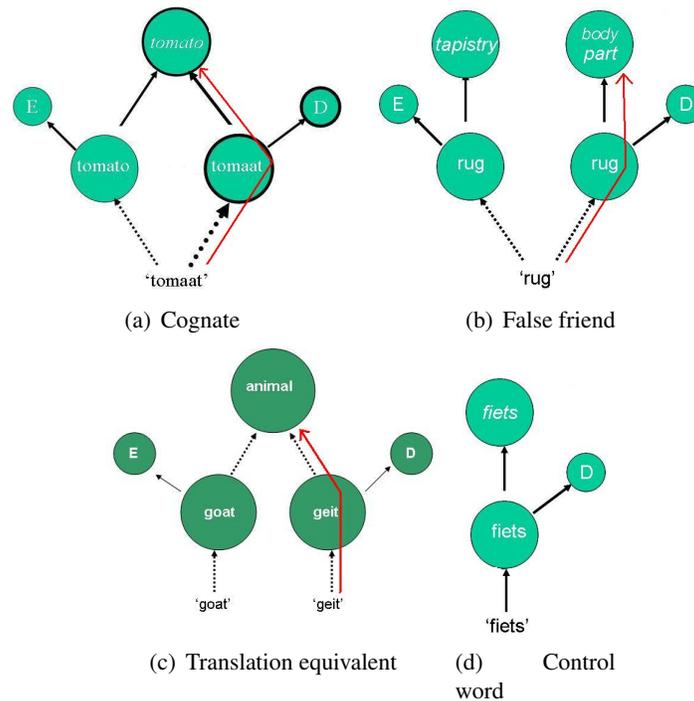


Figure 4: Routes travelled from orthography to semantics in the Dutch simulation

The eBSN-model learns a mapping from orthography to semantics. In the cognate situation (Fig. 4a) the orthographic form of a word is either the same, except for the value of the LCU (identical cognate), or the orthographic form is only slightly different (non-identical cognate). Therefore, in a standard situation, when words from both languages are presented, a facilitation effect can be expected. Because, in this simulation, only words from the Dutch language are presented, we expected the performance of cognates to be quite similar to those of the control words.

In the false friend situation (Fig. 4b) there is no inhibition as a result from competition with English words since we only offer Dutch words to the model. If one looks at Fig. 4b, one will notice that in this situation, the route that will be followed from orthography to semantics is similar to the route in the control word situation (Fig. 4d). Therefore, we expected the performance of false friends to also be quite similar to the performance of control words in this simulation.

For translation equivalents (Fig. 4c) there also should be no inhibition as a result from competition with English words. Fig. 4c shows that in this situation, again due to lack of competition, the travelled route from orthography to semantics is similar to the route in the control word situation (Fig. 4d). Because of this, we expected the performance of translation equivalents to be quite similar to the performance of control words in this simulation.

4.2 The English simulation

This simulation had the same set-up as the Dutch simulation, except that the network only receives words from the English language as input this time (see Fig. 5). Although the input was not the same as in the previous simulation, the input still only contained words from one language. Therefore, we expected to find the same kind of performance as in the Dutch simulations, namely that cognates, false friends and translation equivalents all perform similar to control words.

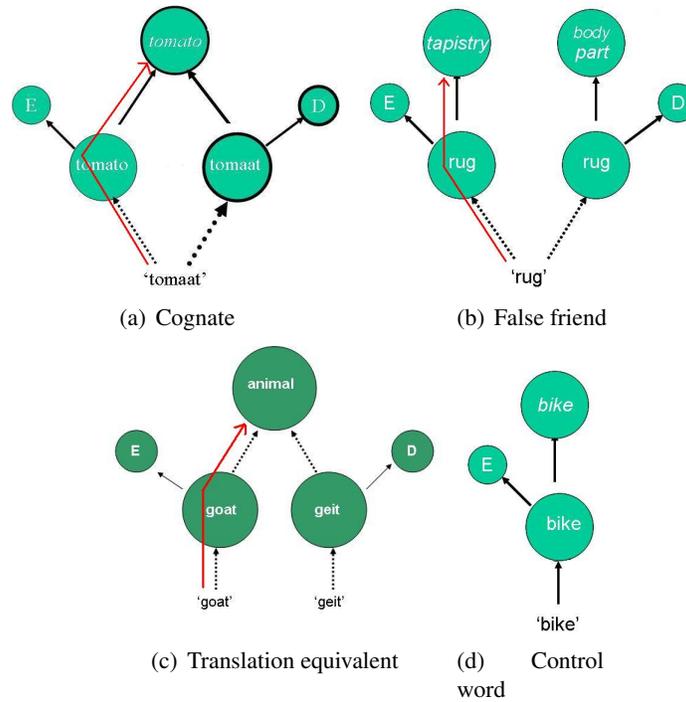


Figure 5: Routes travelled from orthography to semantics in the English simulation

4.3 Mixed simulation with language coding

As mentioned before, the first two simulations served to acquire results which can be used as a baseline. In this simulation I wanted to acquire the same type of results, the same graphs, so that the results could be compared against the baseline and against the results of the mixed simulation without language coding.

The input that was given to the network in this simulation consisted of words from both languages randomly mixed together in a 70-30 ratio. Every Dutch word appeared 70 times and every English word appeared 30 times. The reason for this 70-30 ratio is that we wanted to see what would happen when one language was learned first and then another language was learned later. However, gathering results for the performance on the different categories of words turned out to be pretty time consuming. This meant that we were already at the limits of time that is supposed to be spent on a bachelor-project. Therefore the potential outcome of this effect is not included in the results.

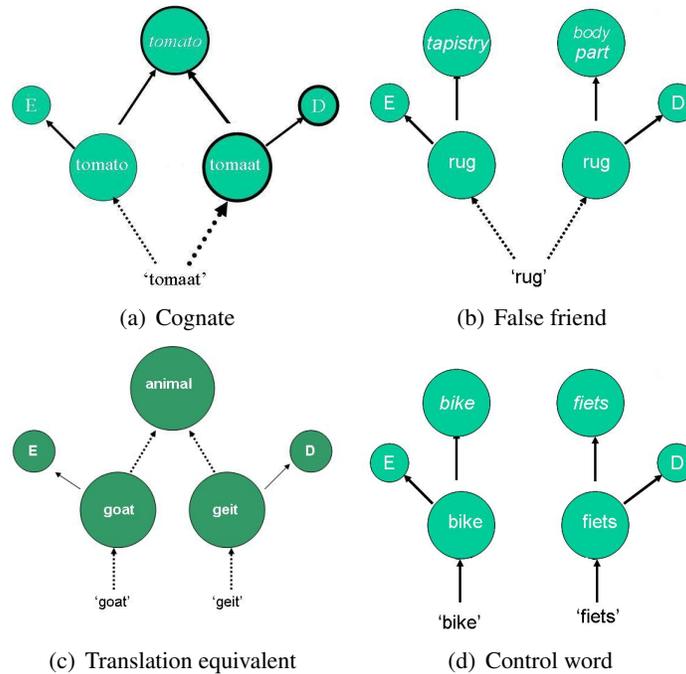


Figure 6: Routes travelled from orthography to semantics in the mixed simulation with language coding

In the cognate situation (Fig. 6a), at one stage the Dutch word “gift” might be presented and the network tries to learn a mapping from orthography to semantic. At another stage, the English word “gift” might be presented and the network will again try to learn a mapping from the same orthography to the same semantic meaning. Therefore, in one epoch, the network will be better able to learn a mapping for cognates than for control words, since in one epoch the route from an orthographic vector to a semantic vector will be travelled more often for cognates than for control words. Therefore, we expected to find a facilitatory effect of cognates in respect to control words.

Since in this simulation language coding is added to the input, there is still no competition for false friends between the Dutch meaning and the English meaning. Because of the LCU the network knows which route it has to travel (Fig. 6b) and that route is similar to the route travelled in the control word situation (Fig. 6d). Therefore, we expected the performance of false friends to be similar to the performance of control words in this simulation.

The same story held for translation equivalents. Because of the presence of a LCU there is no competition between the two routes that can be travelled (Fig. 6c). Again, the network knows which route has to be travelled, and that route is similar to the route travelled in the control word situation (Fig. 6d). So we expected the performance of translation equivalents to be similar to the performance of control words in this simulation.

4.4 Mixed simulation without language coding

In this simulation we again aimed at acquiring the same type of results and graphs, so that the results can be compared against the baseline and against the results of the mixed simulation with language coding.

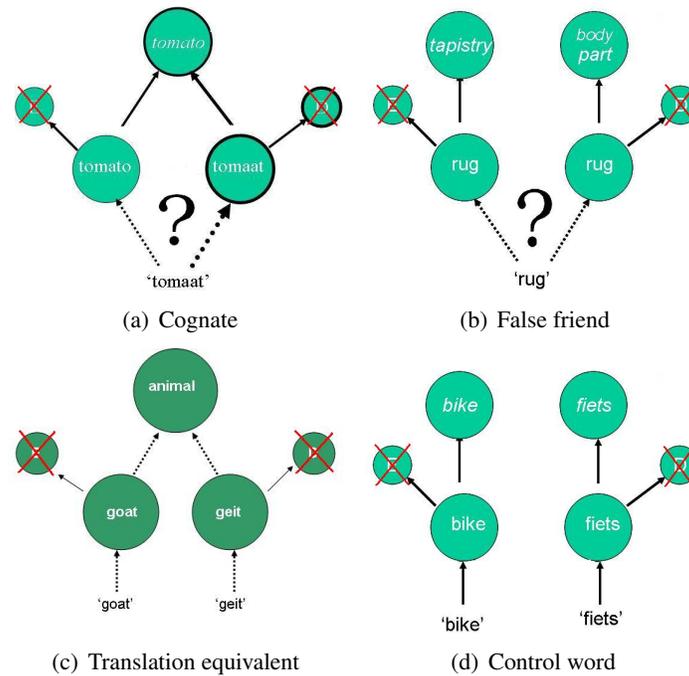


Figure 7: Routes travelled from orthography to semantics in the mixed simulation without language coding

Because for cognates (Fig. 7a), one tries to learn a mapping from words which have either the same orthographic form in both languages or a very similar orthographic form in both languages, to the same semantic form, the absence of a LCU

should not be of much influence. Therefore, we expected the performance of cognates in this simulation to be similar to the performance of cognates in the previous simulation, meaning that they will have facilitatory effect with respect to control words.

In the false friend situation (Fig. 7b), without the LCU the network now does not know which of the two meanings to map to. It does not know which of the two routes should be travelled. Competition now occurs which means that we expected to find an inhibitory effect of false friends with respect to control words (Fig. 7d) in this simulation.

For the translation equivalents situation (Fig. 7c), the network has two different orthographies which map to the same semantic output vector. The goal of the network is to learn a mapping from orthographic input vectors to semantic output vectors. Thus, when two different orthographic input vectors map to the same semantic output vector, this can be considered as a form of competition. Therefore, we also expected an inhibitory effect of translation equivalents in respect to control words in this simulation.

5 The results

In this section we will describe the results and discuss which expectations came true. All the statistical analyses for these results were done via Paired-Samples T Test using the SPSS 15.0 software environment.

5.1 The Dutch simulation

As mentioned before, this simulation was meant to acquire results which can be used as a baseline. To demonstrate the ability of the model to learn, the results should show learning curves for all types of words used in the simulation. As one can see in Figure 8, this is indeed the case. The graphs for the control words, the cognates, the false friends and the translation equivalents all show a nice learning curve.

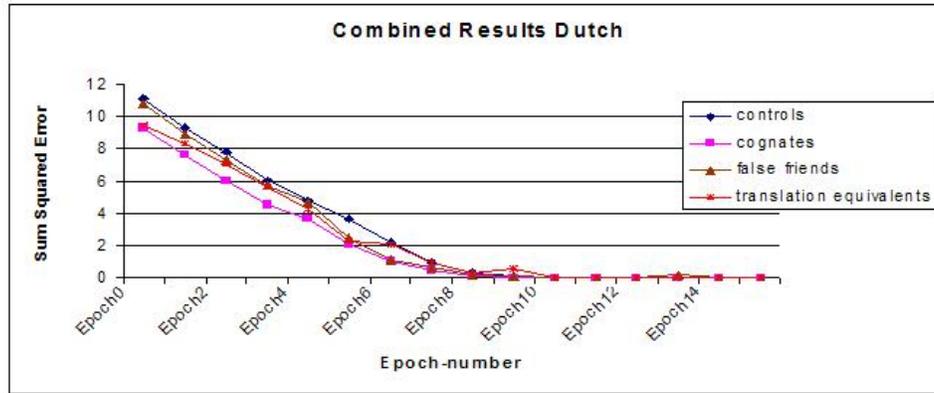


Figure 8: Graph depicting the results of the Dutch simulation

Given that this is a monolingual simulation, we also expected the performance of cognates to be quite similar to the performance of control words. In Figure 8 one can see that the graph of the cognates is a good fit to the graph of the control words. Also, even though the performance of the cognates is slightly better than the performance of the control words, the statistical analysis showed that this was not a significant difference ($p = 0.56$). Figure 8 and the statistical results clearly show that no facilitation effect has occurred, which is a good result.

One can also see that the graph of the false friends fits the graph of the control words. Which is exactly what we expected to find because there is no competition between words from different languages in this simulation. The statistical analysis also showed that the difference between the control words and the false friends was not significant ($p = 0.85$).

The graph of the translation equivalents also fits the graph of the control words, which again is what we expected to happen, because there is no competition present. The statistical analysis showed that the difference between the translation equivalents and the control words was not significant ($p = 0.79$).

Because the graphs fit very well together and they all show learning curves, these result can be used as a baseline for the mixed simulations.

5.2 The English simulation

Although the input in this simulation is not the same as in the previous simulation, it still only contains monolingual input. Therefore the results for this simulation

should be the same as the results for the Dutch simulation. As one can see from Figure 9, the results show that this is indeed the case.

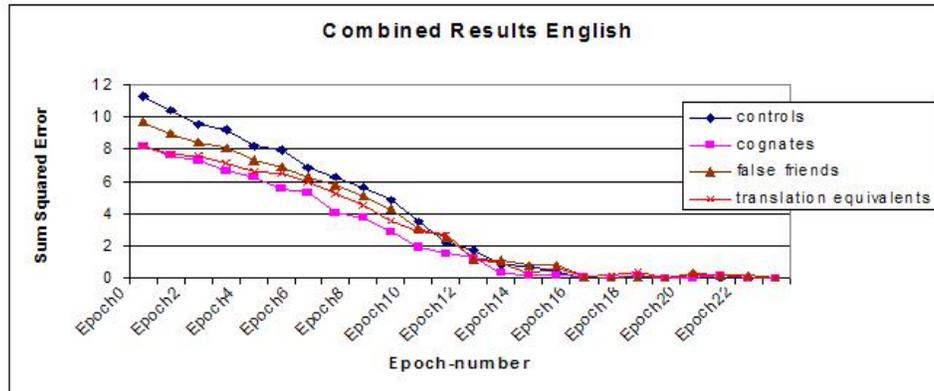


Figure 9: Graph depicting the results of the English simulation

Again, the individual graphs of the control words, the cognates, the false friends and the translation equivalents all depict the occurrence of a learning curve. One can also see that the graph of the cognates fits the graph of the control words. And statistical analysis showed that just like in the Dutch simulation the difference between the cognates and the control words was not significant ($p = 0.29$).

Figure 9 also shows that the graph of false friends fits the graph of control words. According to the statistical analysis the difference between the false friends and the control words also was not significant ($p = 0.75$).

The graph of the translation equivalents also fits the graph of the control words. And again, the difference between the translation equivalents and the control words was not significant ($p = 0.50$).

These results are also suitable to be used as a baseline for the mixed simulations.

5.3 Mixed simulation with language coding

Figure 10 shows the results of the mixed simulation with language coding. If one looks at the graphs, a difference that immediately pops out when one compares these graphs to the graphs of the previous baseline simulations, is that the values on the y-axis are smaller. This is because of the 70-30 ratio we implemented.

Because of this ratio, in each epoch every word occurs multiple times, whereas it only occurred ones per epoch in the previous baseline simulations. That explains the lower sum squared error values.

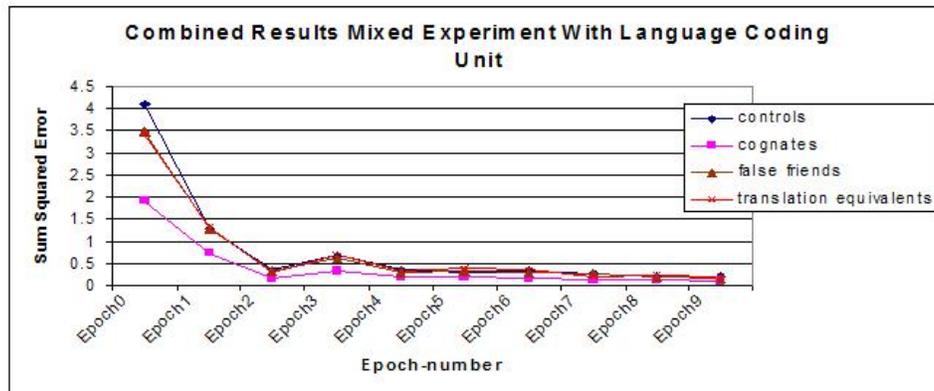


Figure 10: Graph depicting the results of the Mixed simulation with LCU

In this simulation, the orthographic vector for a Dutch word is almost the same as that of the English version of the word, when the word is a cognate. Which is why for a cognate the route from orthography to semantics will be travelled more often per epoch than that of a control word. That is the reason why we expected to find a facilitatory effect here. If one compares the graphs in Figure 10 to those of Figure 8 and 9, then one can notice that the graphs of the false friends and the translation equivalents still fit the graph of the control words. While the graph of the cognates obviously lies lower than the other graphs. This is an indication that a facilitation effect has occurred. However, although a facilitation effect did occur, according to the statistical analysis the difference between the cognates and the control words was not significant ($p = 0.34$). The presence of a facilitation effect suggests that cognates are processed differently than control words. This is an indication of non-selective lexical access.

Because of the presence of the LCU the network knows to which language a word belongs, even when we present words from both languages at the same time. The graph of the false friends is very similar to the graph of the control words. And the statistical analysis showed that the difference between the false friends and the control words was not significant ($p = 0.87$). These are both indications that indeed no competition occurred.

We expected to obtain the same results for the translation equivalents and the results again confirmed our expectations. One can see that the graph of the transla-

tion equivalents is very similar the graph of the control words. And the statistical analysis showed that the difference between the translation equivalents and the control words was not significant ($p = 0.90$). Both are indications that no competition has occurred.

5.4 Mixed simulation without language coding

Figure 11 shows the results of the mixed simulation without language coding. One can immediately notice some interesting results.

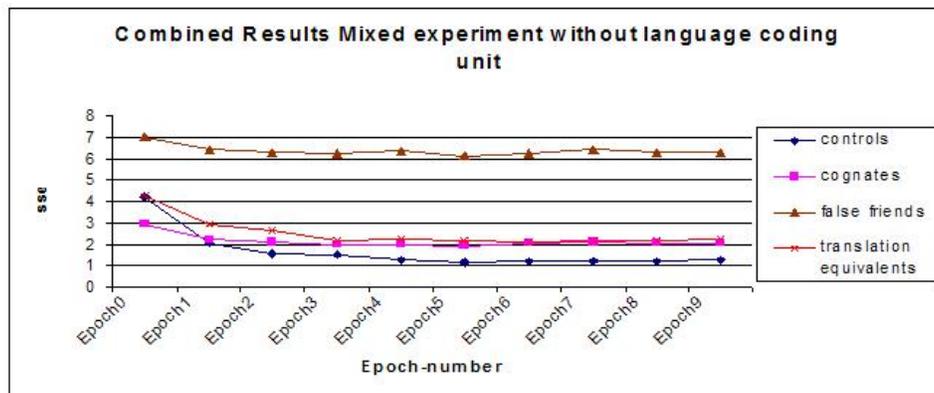


Figure 11: Graph depicting the results of the Mixed simulation without LCU

First of all, the graph of the cognates lies higher than the graph of the control words. We expected to find a facilitatory effect for cognates, because we are trying to map words with either the same or a very similar orthographic form to the same semantic form. Therefore the absence of a LCU should not be of much influence.

In order to see whether the Dutch or the English version of the cognate was being processed we kept language membership information in the output vector. By lesioning the LCU, there was no language membership information available in the input layer, but it was still available in the output layer. Of the 121 bits in the output vector of a cognate, 1 bit has a different value. This explains why the graph of the cognates lies higher than the graph of the control words when it should have been lying beneath it. Statistical analysis showed that the difference between the cognates and the control words was not significant ($p = 0.14$).

In Figure 11 the graph of the false friends lies much higher than the graph of the control words. There is a big difference between the relative positions of the

two graphs in this figure compared to the figures of the first two simulations. In contrast with those two simulations, in this simulation the network does not know which of the two meanings to map too because of the absence of the LCU. It now does not know which of the two routes should be travelled. The statistical analysis showed that the difference between the false friends and the control words was significant ($p = 0.00$). These results are indications that competition did occur during learning and that there is an inhibitory effect of false friends with respect to control words. The presence of an inhibitory effect suggests that false friends are processed differently than control words. This is an indication of non-selective lexical access.

The graph of the translation equivalents also lies higher than the graph of the control words, although not as high as the graph of the false friends. This is also a noticeable difference compared to the graphs for the first two simulations. The network now has two different orthographic input vectors that try to map to the same semantic output vector. The statistical analysis showed that the difference between the translation equivalents and the control words was significant ($p = 0.03$). These results are clear indications that competition did occur and that there is an inhibitory effect of translation equivalents with respect to control words. The presence of this inhibitory effect suggests that translation equivalents are also processed differently than control words, an indication of non-selective lexical access.

6 Conclusion

This bachelor-project was focussed on the BSN model of Thomas (1997b). The BSN model received items from two artificial languages as input, which necessitated the use of language tags to separate the languages from one another. We believed the use of language tags can have too big of an influence on bilingual word recognition. And the generalizability of the model is limited because of the use of artificial languages.

As a consequence, we thought it was important to reproduce the results of Thomas' experiment using words from natural languages as input and without the use of language tags. Therefore, we recreated the BSN model of Thomas. Instead of using 3-letter non-words, our model received 4-letter words from a natural language as input. We used English and Dutch as natural languages for this purpose.

We predicted differences in recognition time for cognates, interlingual homographs and translation equivalents as compared to control words. This would imply that these words are processed in a different way than the control words, a finding

that would be in line with the hypothesis of non-selective access to the bilingual lexicon and empirical studies.

In all, the Dutch and the English simulations were very successful. All predictions were born out. In both simulations one can see appropriate learning curves, meaning that these results can be used as a baseline to compare the results of the mixed simulations with.

In the mixed simulation, we were able to show the presence of a facilitation effect for cognates with regard to the control words, by comparing the results from this simulation to the baseline results. The presence of this effect is an indication of non-selective access in bilinguals. As for the false friends and translation equivalents, the results again confirmed our expectations.

By comparing the results from the mixed simulation without language coding to the baseline results, we were able to show an inhibitory effect for both the false friends and the translation equivalents with regard to the control words. These are also in line with empirical findings of non-selective lexical access in bilinguals. The results for the false friends and the translation equivalents were satisfactory, but the results for the cognates could have been better. In this simulation there is definitely room for improvement in the cognate situation.

For future research, one can try to improve the results in the cognate situation for the mixed simulation without language coding, by removing the LCU in the output vector. With the BSN-model Thomas was also able to demonstrate within-language frequency effects as well as a facilitatory effect for cognate homographs in L2 relative to L1 in the unbalanced network. The eBSN model can be further extended to include the factors word frequency and learning procedure (L1 and L2 presented at the same time, L2 presented later on) for cognates, interlingual homographs and translation equivalents to try and reproduce these results.

In all, our simulations were very successful. In the Dutch and the English simulations all our predictions were born out. And in the mixed simulations, we were able to show the presence of a facilitation effect for cognates with regard to the control words, as well as an inhibitory effect for both the false friends and the translation equivalents with regard to the control words. The presence of these effects are in line with empirical findings of non-selective lexical access in bilinguals.

References

Dijkstra, A. and Van Heuven, W. (1998). The BIA model and bilingual word

- recognition. In Grainger, J. and Jacobs, A., editors, *Localist Connectionist Approaches to Human Cognition*, pages 189–225. Mahwah, NJ, USA: Lawrence Erlbaum Associates.
- Dijkstra, A. and Van Heuven, W. (2002). The architecture of the bilingual word recognition system: From identification to decision. *Bilingualism: Language and Cognition*, 5:175–197.
- French, R. (1998). A simple recurrent network model of bilingual memory. In Gernsbacher, M. and Derry, S., editors, *Proceedings of the 20th annual Conference of the Cognitive Science Society*, pages 368–373. Mahwah, NJ: Erlbaum.
- French, R. and Jacquet, M. (2004). Understanding bilingual memory: models and data. *TRENDS in Cognitive Sciences*, 8(2).
- Léwy, N. and Grosjean, F. A computational model of bilingual lexical access. Manuscript in preparation.
- Li, P. and Farkas, I. A Self-Organizing Connectionist Model of Bilingual Processing. In Heredia, R. and Altarriba, J., editors, *Bilingual Sentence Processing*. North Holland: Elsevier Science Publisher. In press.
- Thomas, M. (1997a). *Connectionist networks and knowledge representation: The case of bilingual lexical processing*. PhD thesis, Oxford University.
- Thomas, M. (1997b). Distributed representations and the bilingual lexicon: One store or two? In Bullinaria, J., G. D. and Houghton, G., editors, *Proceedings of the 4th Annual Neural Computation and Psychology Workshop*. Springer.
- Thomas, M. (1998). Bilingualism and the Single route / Dual Route debate. *Proceedings of the 20th Annual Conference of the Cognitive Science Society*, pages 1061–1066.
- Van Heuven, W. and Dijkstra, A. The Semantic, Orthographic, and Phonological Interactive Activation Model. Working title.
- Van Heuven, W., Dijkstra, A., and Grainger, J. (1998). Orthographic neighborhood effects in bilingual word recognition. *Journal of Memory and Language*, 39:458–483.

Appendix A

Table 1: Dutch words per category

Control words	Cognates	False friends	Translation equivalents
bast	base	auto	baas
gast	gift	boot	beet
fase	safe	bout	biet
toga	sofa	gage	buis
tese	test	gist	geit
tuig	tube	gust	
		sage	

Table 2: English words per category

Control words	Cognates	False friends	Translation equivalents
babe	base	auto	boss
sift	gift	boot	bite
ease	safe	bout	beet
soft	sofa	gage	tube
teat	test	gist	goat
stub	tube	gust	
		sage	