Exploring Phonotactics with Simple Recurrent Networks

Ivelin Stoianov and John Nerbonne

University of Groningen, Faculty of Arts, Email: {stoianov, nerbonne}@let.rug.nl

Abstract

Stoianov, Nerbonne and Bouma (1998) trained Simple Recurrent Networks (SRNs) on graphotactics of Dutch monosyllabic words, overcoming shortcomings of previous implementations. The current report is a continuation of our earlier research, but using phonetic data representations instead of orthographic, that is, learning phonotactics. In addition, we conducted further analysis of neural network performance with regard to some variables such as word frequency, length, neighborhood density and error location. The results are compared with reported psycholinguistics analyses. This informal comparison of SRNs and human performance suggests that SRNs can be used for modeling natural language processing.

1 Introduction – studying lexical constraints with SRNs.

The present paper reports on a project investigating how well natural language phonotactics may be learned using neural networks (NN) (Stoianov, Nerbonne, and Bouma 1998, hereafter SNB98), which is interesting from different perspectives. Firstly, it challenges connectionism to tackle symbolic problems – Tjong Kim Sang (1995; 1998) reported that symbolic and stochastic methods performed well on this problem, but connectionist techniques were not that successful. Next, phonotactics has important applications in problems such as speech segmentation and recognition (Shillcock et al 1997; Cairns et al 1997). Also, since phonotactics involves sequential processing, it constitutes a step toward syntax. And lastly, by modeling lexical problems with NNs that parallel human mental architecture, we contribute to the explanation about how people might process natural language.

In our preceding study on lexical grammar modeling, we learned graphotactics of 4500 Dutch monosyllables with Simple Recurrent Networks (SRN) (Elman 1990). For this purpose, SRNs were trained to predict the graphemes that can follow left contexts. Tjong Kim Sang found it difficult to learn similar data with SRNs and therefore we extended the standard SRN prediction task with a special algorithm that finds a threshold best distinguishing words from non-words (SNB98). Further work on this problem was presented in (Stoianov 1998), where other SRN evaluation techniques were developed, assessing learning from another perspective. They were based on matching the context-dependent character distribution in the training data with the context dependent character prediction produced by SRN trained on the same data. Analysis of those algorithms revealed that each of them should be used according to the specific problem to which SRNs are applied. For example, training an SRN on lexical decision task should be evaluated with the method proposed in SNB98, while training an SRN on phonotactics rules is best evaluated with the algorithms suggested in Stoianov (1998). In this paper, both methods are used for the sake of comparison.

The current study explores phonetic data representations by training a Neural Network (NN) on lexical phonotactics instead of graphotactics. Compared to the graphemic representations, the phonetic representations are shorter and contain about 70% more distinct elements (44 phonemes). Therefore, we predicted in SNB98 that phonotactics would be equal or less difficult to learn given equal network size. This expectation is further strengthened by the fact that the phonological constraints are stronger than the graphemic ones, as measured by mean lexical entropy. The results reported in the present study confirm this anticipation with better performance for phonotactics learning.

Connectionism was originally inspired by attempts to model and explain the human neural system as well as higher order cognitive functions, including language processing. As artificial connectionist models prove successful with the problems they have been designed for, it is challenging to seek better approximation of human cognitive functions. For example, Plaut et al. (1996) claim to model effects observed in human lexical processing. In this comprehensive study, performance analyses of Multilayered Perceptron and Attractor Neural Networks were presented. Both models were trained on mapping from orthography to phonology and employed static lexical representations. However, since language spans time, lexical processing should be sequential and dynamic. Therefore, we criticize connectionist models with static lexical representations. By providing an account of some of the temporal effects they were not able to model, we claim that dynamic processing is superior for lexical modeling, and we propose SRNs as a proper architecture for this purpose.

In the remainder of this paper, we provide, first some background on phonotactics in section two and connectionist modeling in section three – those might be skipped by readers acquainted with the problem. In section four we outline the experimental set-up, describe the evaluation procedures and provide results. Further, in section five we offer an analysis of the error profile. In section six we conclude with a discussion on connectionist phonotactic modeling and its applications.

2 Phonotactics and related issues

Just as syntax studies how words combine to form phrases and sentences, so phonotactics studies how phonemes combine to form syllables and words (Laver 1994). In this study, we focus on the phonotactics of Dutch monosyllables, which are represented with a list of about 6100 monosyllabic words extracted from CELEX lexical database.

Phonotactics can be used for different purposes. Besides studying phonotactic constraints from theoretical point of view, another task based on phonotactics is *lexical decision*. In a lexical decision task, a model M_L trained on language L tests whether a given string belongs to the training language. Models trained on phonotactics can be used both for phonotactic analysis and lexical decision. In section four we will discuss this problem in detail. Another process that might benefit from phonotactics is word *segmentation*. Speech is continuous, but we divide it into psychologically significant units such as words and syllables. There are a number of cues that we can use to distinguish these elements – prosodic markers, context, but also phonotactics. Infrequent sequences could signal word boundaries. See McQueen (1998) for psycholinguistic insights on this problem and Cairns et al. (1997) for connectionist modeling (next section for more details).

An issue closely related to language learning is the type of negative data used in testing the recognition capabilities of a model M_L. Because different strategies for random string generation bias the estimated model performance, we examined recognition on different sets of strings. In order to illustrate this problem, let's assume that there are two models, M_1 and M_2 trained on language L and that the first model performs better than the second does. If we test these models on genuine words with frequently used spellings, both models would probably accept these words. Further, both models would reject words that are entirely different from any word in L. But probed on unclear cases – random sequences similar to words from L and rare words from L – the model M_1 would accept more words than M_2 and would reject more random strings than M_2. This boundary set of unclear cases is where we can evaluate best the performance of language models. For that purpose, we used the following negative data sets: random strings generated with monosyllabic structure $([[[C]C]C][V]VC[C[C]])$ and the same strings, but categorised with regard to their phonemic Levenshtein distance to any Dutch monosyllabic word (1, 2 and 3) (Nerbonne et al. 1996).

3 The human language processor and connectionist modeling

Language learning is not an easy task (Seidenberg 1997 for review), especially if we want to avoid the convenient symbolic representations and methods especially designed for describing structured linguistic objects. Symbolic methods are designed for representing and processing high-level systems, e.g., artificial or natural languages. More problems come with low-level implementations, especially with basic processors like ours – the brain. The brain is not a computer with a single CPU, common addressable memory and a Prolog interpreter. It is a complex of highly interconnected neurons, each of which, although representing a complex chemical factory, is subject to approximation with a simple mathematical expression. Structures built of neuron-like units are connectionist NNs. In connectionist language learning we avoid high-level descriptions and encode both language objects and processing methods directly into a set of weights. In NNs, data is represented distributively and the model's action is based on a *function* – mapping from a given input domain to another output domain. If we consider the more specific phonotactics problem, the input domain is a distributed representation of phonemes. The network maps the input and the contextual information embodied in the structure of the NN to the output domain, which is a distributed representation of phonemes (possible successors).

There is a number of NN models; most of them are designed for static data processing, and many connectionist language implementations employ such static models, e.g., Plaut et al. (1996). Language takes place in time, however. We produce and hear sequences of sounds, which we may represent statically, but at least spoken language is always expressed sequentially. Therefore, an NN model for language should have an internal dynamics allowing reaction that depends on the past input events. Such a model with a long history in connectionism is the Simple Recurrent Network (SRN) developed by Jeffrey Elman (1990). SRNs have contextual memory, and their output depends on current and past input objects (Fig.1). The capacity of this network for sequential processing might be thought of as analogous to the capacity of the Multilayer Perceptron, which in turn is a universal approximator with unlimited precision (assuming only enough hidden neurons). Indeed, there are a number of applications based on SRNs, e.g., orthography to phonology conversion (Stoianov, Stowe and Nerbonne, 1999) and syllable learning (Gasser 1992), among others.

Now, let's consider some connectionist language learning systems related to our problem. Perhaps the first big successful project on connectionist language learning was by Sejnowski and Rosenberg (1987), the famous NETtalk model, trained on orthography to phonology conversion for English words. The authors in this early connectionist model implemented sequential processing on a static MLP using a text window – the surrounding context was represented as a window surrounding the grapheme to be pronounced, which shifts in time. The idea was very successful, in spite of the shortcoming that the fixed-size window is never long enough to encode all long-term dependencies. SRNs don't have this problem because the context potentially encodes the entire past input.

Invented by Elman (1990), SRNs were initially used to encode simple artificial grammars; similar experiments were conducted by Cleeremans et al. (1989). Further, Elman conducted investigations on how context evolves in time. The analysis showed graded encoding of the input sequence: similar items presented to the input were located at close, but different, shifting positions. This is notable, because the rules for context were not encoded, but evolved in learning. The capacity of SRNs to learn simple artificial languages was further explored in a number of studies, e.g., Gasser (1992) who modeled recognition and production of 135 syllables generated by artificial grammars.

As mentioned in the previous section, an essential application of phonotactics is speech segmentation. Shillcock et al. (1997) and Cairns et al. (1997) trained SRN models on English phonotactics, using 2 million segments. The NN was presented a single phoneme at a time and was trained to produce the previous, the current and the next phonemes. The output corresponding to the predicted phoneme was matched against the following phoneme, measuring cross-entropy; this produced a varying error signal with occasional peaks corresponding to word boundaries. The reported "performance is quite modest, at around one-fifth of word boundaries, and is coincident with syllable boundaries"(p.137); it was significantly improved by adding other cues such as prosodic information. This means that phonotactics might be used alone for syllables detection, but polysillabic word detection needs extra cues. Although not very successful, this research is a significant attempt to employ connectionist models in natural language application.

Parallel Distributed Processing (PDP) approach to language modeling was also exploited by Dell et al. (1993). They showed that words could be described not only by the symbolic approach using word structure and content, but also by a connectionist approach. They trained SRNs to predict the phoneme that follows the current input phoneme, given context information. The data sets contained 100 - 500 English words. An important issue concerned in this paper was an analysis and modeling of a number of speech-error phenomena, which was taken as a strong support for PDP models, in particular SRNs. Some of these phenomena were: phonological movement error (*reading list* - l*eading list*), manner errors (*department* - j*epartment*), phonotactic regularity violation (*dorm* - *dlorm*), consonant-vowel category slip and initial consonant slipping (initial consonants drop more often than non-initial ones).

A recent connectionist study on phonological regularities was presented in Rodd (1997), where SRNs were trained on 602 Turkish words; the network was trained to predict the following phonemes. Analyzing the hidden layer representations developed during the training, the author found that hidden units came to correspond to graded detectors for natural phonological classes such as vowels, consonants, voiced stops and front and back vowels. This is further evidence that NN models can capture important properties of the data they have been trained on without any prior knowledge, based only on statistical co-occurrences.

A history of connectionist natural language modeling was sketched by Seidenberg, Plaut and colleagues. In a number of papers: Seidenberg and McClelland (1989), Plaut et al. (1996), Plaut (1997), and Plaut and Kello (1998), they exploited MLP and Attractor NNs. The main importance of their contributions was that they propose a complete language processing model, including orthography, phonology and semantics, plus related interconnections. In an extensive series of experiments, they demonstrated that the models above perform similarly to humans with regard to a number of parameters, such as word frequency, consistency of orthography-to-phonology mapping, dyslexia, etc. Still, there are other effects that were not and could not be exhibited, such as effects of word length and error positioning. This is because the above models are static, which restricts their capacities and does not allow the expression of dynamic properties. As we mentioned earlier, words should be processed dynamically, and we find SRNs appropriate for this purpose. In the following sections we will present experiments on phonotactics learning with Simple Recurrent Networks. In addition, we will investigate how network performance correlates with human lexical processing.

4 Phonotactics Learning with SRNs

A Simple Recurrent Network was trained to predict the phonemes that follow phonemes presented sequentially to the input layer, that is, context-dependent character prediction. The working set of sequences comprised all 6100 monosyllabic Dutch words extracted from CELEX lexical database. The same corpus was used in our previous studies on graphotactics (SNB98; Stoianov 1998). The data set contained phonetic word representations and the frequencies of word oc-

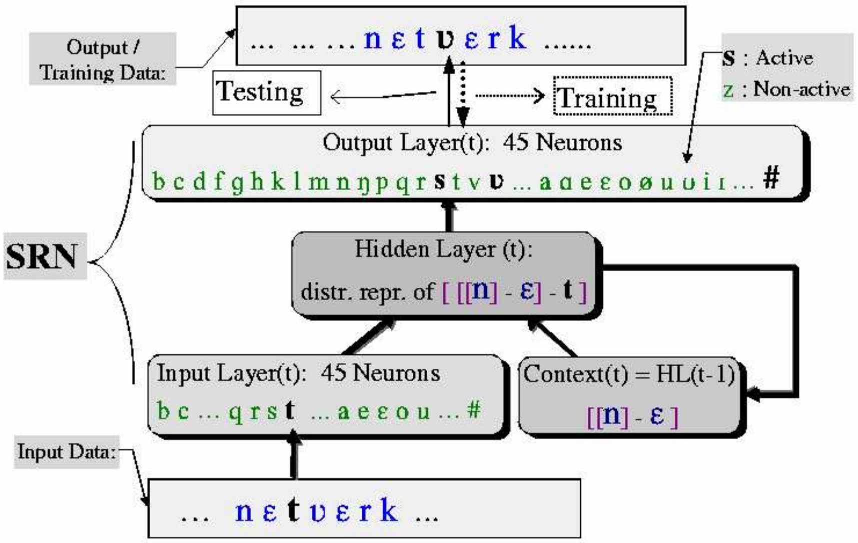
Output / Training Data:
... ... n ɛ t ʋ ɛ r k ......
Testing
Training
s : Active
z : Non-active
Output Layer(t): 45 Neurons
b c d f g h k l m n ŋ p q r s t v ʋ ... a ɑ e ɛ o ø u ʊ i ɪ ... #
SRN
Hidden Layer (t):
distr. repr. of [[[n] - ɛ] - t]
Input Layer(t): 45 Neurons
b c ... q r s t ... a e ɛ o u ... #
Context(t) = HL(t-1)
[[n] - ɛ]
Input Data:
... n ɛ t ʋ ɛ r k ...

Figure 1: SRN and phonotactics learning. If the training data set contains the words *net#, nets#, netverk#* then after the network has processed a left context *ne*, the reaction to an input *t* will be active neurons that correspond to phonemes *#, s, v*.

currence in the Dutch language. This corpus was split into a training set (5100 words) and a test set (1000 words).

The phonetic word representations have mean length of $3.92(\sigma = 0.94, min = 2, max = 8)$ respectively and they are built of 44 phonemes, plus one extra symbol representing space ('#') used as a filler specifing *end-of-word*. The phonemes were encoded orthogonally, that is, for each phoneme, there was one input or output neuron respectively. This representation is almost equivalent to a distributed, feature-based representation, because one can always add two layers of neurons to encode and decode feature-based representations to orthogonal classes. In turn, these mappings are trivial to learn. However, the orthogonal encoding has the advantage of allowing phonemes to be activated independently, while the feature-based encoding allows activating groups of phonemes, which is an architectural constraint.

During training and testing, the phonetic word representations were given to the network sequentially, one symbol at a time. The network decision at each moment is based on the current input and full left context, encoded distributively at the context layer. The rules that govern context encoding emerge in learning (Elman 1990). At each training step, the network prediction was corrected using the phoneme that follows the current left context. In the whole training data set, it is possible for more than one phoneme to follow a given left context. Therefore, SRN training results in *likelihoods* that a given phoneme should follow a left context, sequentially presented to the input layer.

Experiments were conducted with different numbers of hidden neurons, ranging from 10 to 100. Better performance and faster error convergence was found with larger networks. In this report, we present results for a network with 100 hidden neurons, resulting in approximately 19,000 weights. The network architecture is presented in Fig.1, where an example prediction for the phonetic representation of the word *netverk* is shown as well, at the moment when the sub-sequence *net* has been presented to the input.

An example of the network reaction is given in (1). The word *frest* (*vreest*) is presented to the network one phoneme at a time and the network reactions for each time steps are given. The phonemes corresponding to the most active neurons and their activations are listed. The presented activations range from 0 to 100. Note that after the phoneme *f*, consonants and vowels are activated (1a), while after presenting *fr*, only vowels remain active (1b). After the vowel *e*, the network activates only consonants (1c); the left context *fres* may be followed by the phoneme *t* and the special end-of-word symbol (1d), which in turn is the only active symbol after *t* is presented (1e).

(1a) f → r(15) Net[t3,n2,l15,r15,i3,e4,a3,o2,u2, ɪ4,ɛ3,ɑ4, ɔ4,ʉ2,ɛi3]
(1b) r → e(6) Net[i3,y2,e6,a6,o5,u5,ɪ7,ɛ6,ɑ8,ɔ6, ʉ4,ɛi6,oey3]
(1c) e → s(6) Net[p6,t9,k9,m4,n6,l3,r5,f7,s6,X4,ʋ4,#2]
(1d) s → t(24) Net[t24,#36]
(1e) t → #(58) Net[#58]

4.1 Task-dependent evaluation of phonotactics learning

As we mentioned earlier, we can use phonotactics in *lexical decision* problems or to identify the *phonological constraints* in the training language. The training procedure is the same in both cases, but the evaluation procedures differ.

In the *lexical decision* task, the model aims at detecting sequences $w = [c_1, c_2, ...c_n]$ that are valid in a language L. The acceptance decision might depend on local decisions for every predicted phoneme or on global analysis. In the first, *localist* case, an algorithm judges whether each predicted phoneme c_i is allowed by the phonological constraints in the language L. In case it results in a negative answer, the whole sequence w would be classified as a non-word. The second approach is *holistic*. It asks to what extent the whole sequence $(c_1, c_2, ...c_n)$ belongs to the language L. One way to evaluate the network's confidence in string w is to multiply the network responses n_i (corresponding to phonemes c_i) or similarly, to sum $log(n_i)$. In both local and global cases, we need to determine an acceptance threshold θ that will distinguish words from non-words. An *Optimal threshold* algorithm that searches for such a threshold was proposed in SNB98. This algorithm examines the training data set and a set of non-words and finds a threshold that minimizes the lexical decision error. In SNB98 graphotactics was studied on the same training set, and the local decision rule was used.

We might define the *phonological constraints* of language L as the conditional distributions $P_L(c_i/c_1c_2...c_{i-1})$. Consequently, evaluating NN performance on this problem would be matching $P_L(c_i/c_1c_2...c_{i-1})$ against the network predictions $N^{C_i}(c_1c_2...c_{i-1})$ for all phonemes c_i and available contexts $[c_1, c_2, ...c_{i-1}]$ in L. Such a matching algorithm was presented in (Stoianov 1998), where the SRN is evaluated on context-dependent phoneme prediction. The algorithm makes a tree traversal of a tree-based corpus representation and computes the mean L_2 distances and $cosine$ between $P_L(c_i/c_1c_2...c_{i-1})$ and $N^{C_i}(c_1c_2...c_{i-1})$.

Another algorithm that estimates the network performance on the training language L was presented in (Stoianov 1998); it is based on computing the mean error in *prediction* at every left context $[c_1, c_2, ...c_{i-1}]$ available in the language L. In that case, there is an erroneous prediction if the activation $N^{C_i}(c_1c_2...c_{i-1})$ is higher than a certain threshold θ when the phoneme c_i can not follow the context, $[c_1, c_2, ...c_{i-1}]$, or in the opposite case, if the network activation is lower than θ, but the phoneme actually can follow. The advantage of this evaluation is that it estimates very quickly the network decision for every available context by tree traversal. It uses negative data implicitly, which is represented as negation of the allowed successors at every tree node in language L.

In the above evaluation procedures, the local errors were weighted by the frequency of the word the phonemes belong to. This results in fair estimation of the network performance, accounting for the distribution of the words in language.

4.2 Training

The training process was organized in epochs, in the course of which the whole training data set (5100 words) was presented to the network in accordance with the word frequencies. In order to reduce learning time, the actual word frequencies were shrunk by applying a logarithmic function, resulting in about 12,200 training sequences per session. Such an approach was used by other authors as well (e.g., Plaut et. al 1996). Next, for each word, the sequence of phonemes was presented to the input, one at a time, followed by the end-of-word symbol ('#'). Each time step was completed by copying the hidden layer activations to the SRN context layer, which was used in the following step (Elman 1990). At the same time, after the network generated its expectations for the phonemes at the output layer, the representation of the actual following phoneme was used to compute an error for the current time step. This error was used by the Back-Propagation Through Time (BPTT) learning algorithm (see for details Haykin 1994; SNB98), which consists of a forward move where errors are collected and a backward move, during which global error is back-propagated through time until the beginning of the current training sequence (i.e., word). This process is followed by updating the network weights with values accumulated during the backward move. The state of the network (i.e., the context memory) is reset after processing each word.

The network was trained in 20 epochs, resulting in approximately 250,000 word presentations or 1,000,000 segments. The total number of individual word presentations ranged from 20 to 200, according to the individual word frequencies. The network started with a sharp error drop to about 9.5%, slowly decreasing down to 4.4% (see Table 1).

The parameters of the learning algorithm were as follows: learning coefficient

Epoch	1	2-4	5-10	11-15	16-20
Error (%)	9.5	6	5	4.9	4.4

Table 1: Dynamics of the SRN error during the training.

η started at 0.4 and decreased by 30% after each epoch, ended at a bottom limit of 0.001; momentum term $\alpha = 0.5$. The BPTT algorithm is well known for its tendency to fall into local minima in searching for the global minimum. Learning a task as complex as phonotactics must guard against this tendency. Therefore we applied a special algorithm that supervises the training process and minimizes this risk. The supervisor is an evolutionary algorithm that trains a pool of networks on the same problem and, after each training epoch, eliminates the network with the worst performance, keeping clones of the networks that performed better. This training method was developed in our previous studies on graphotactics (SNB98) and was found to perform better than the standard single-network training.

4.3 Performance

As we discussed earlier, the performance of SRNs trained on phonotactics should be evaluated with respect to the task the network is going to be applied to. One global measure of how well *phonological constraints* were learned was based on matching the real and predicted context-dependent phonemic distributions in the training language. Applied to the trained SRN, this procedure resulted in meanL_2 distance of $0.065(\sigma = 0.045)$ and mean $cosine$ distance of $0.75(\sigma = 0.23)$, with ideal values of zero and one, respectively. These distances tell us to what extent the network has learned the task, but they do not concern phoneme classification, which the other tree-based evaluation algorithm measure. This method resulted in 5.5% erroneous phoneme prediction at a threshold of 0.02. This means that if we want to classify phonemes with this SRN, they would be accepted as allowed successors if the activation of the correspondent neurons are higher than 0.02.

The *lexical decision* task shifts the focus from best phoneme prediction to best sequence classification. However, the *optimal threshold* algorithm that tracks this problem needs negative data in addition to the positive data, which biases the estimation with regard to the random string set. A class of random strings that is close to the training data – a set of monosyllables – contains strings with monosyllabic structure. The SRN was evaluated on the training data set and a set of such random strings. The performance measured the percentage of correctly predicted phonemes. We used the *localist* lexical decision rule, which resulted in about 5% error at a threshold of 0.016; the error varied $\pm 0.5\%$ depending on the generated random data set. As we can see, the optimal threshold is slightly lower than the optimal threshold from the previous algorithm. This means that the network would accept more phonemes, which, in turn, is compensated for by the fact that a string is accepted only if all phonemes are predicted. In this case, it is better to increase the phoneme acceptance rate. We continue with more precise determination of the random test set and divide it into three groups according to the Levenshtein

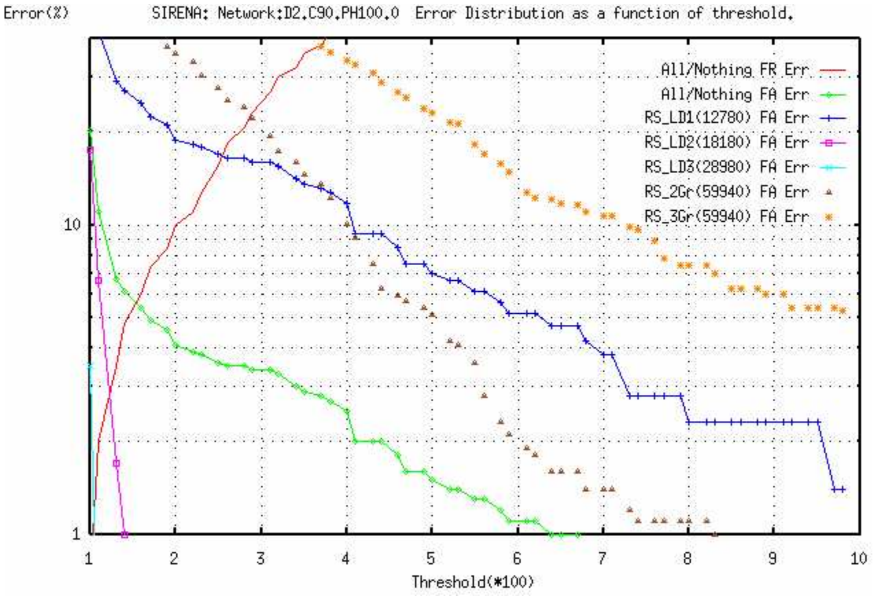

Figure 2: SRN error (in %, logarithmic scale) as a function of threshold. A. False rejectance (FR) of the positive training data: a line rising from the bottom left corner. B. False acceptance (FA) of negative data: (I) random monosyllables (a line with circles that meets the FR line at 4.6%) (II) random strings with LD 1,2 and 3+ (lines with various shapes; 3+ is barely visible) and (III) random negative strings constructed of bi- and tri-grams (dotted lines crossing the FR line at 23% and 35% respectively). The numbers in the legend stand for the number of tested tokens (phonemes).

distance (LD) between the random strings and any word from the training data set (one, two and three). The results are shown in Fig.2. As expected, larger distances resulted in smaller error (in Fig.2, the line showing random strings LD=3+ is almost invisible due to the very small error). In the opposite case, strings that were close to real words resulted in larger error. Further, the generalization capabilities of the network were tested on a test set, which contained the phonetic representations of 1000 Dutch words, unseen during training. The performance on this test set was about 6%. In the next section, we will see that error is very sensitive to phonological neighborhood, and we attribute the error increase to testing on unseen words with low-density neighborhoods.

Phonetic word representations are slightly shorter than orthographic ones. In addition, an orthogonal phonemic encoding involves more neurons because there are almost twice as many phonemes as graphemes. Therefore, phonotactics should be learned more perfectly than graphotactics for same hidden layer size, which we in fact observed (increasing the neuron layer size results in more memory and better task distribution among the neurons in the layer). In orthographics, there is about 7% error (SNB98), while phonotactics learning results in about 5% error.

Another possible explanation is that phonotactic constraints are stronger than those in graphotactics are.

Another interesting issue was how SRN performance compares to other known models, e.g. *n-grams*. The trained SRN definitely outperformed *bi-* and *tri-*grams, which we showed by testing SRNs on nonwords composed of *bi-* and *tri-*grams extracted from the training data set (see Fig.2, dotted lines). The SRN error in that case was $20 - 23\%$ and $30 - 35\%$ error respectively. This means that approximately every four of five and every two of three nonwords generated of *bi-* and *tri-*grams respectively, would be rejected by the SRN, while the correspondent n-gram models would accept them.

5 Effects of Frequency, Neighbourhood, Length and Position on Error

The distributed way of representing data and data processing rules makes it difficult to analyze the trained networks by direct observation of the hidden layer activations and neuron weights. Smaller models may be analyzed to some extent, as in Elman (1990) and SNB98, but larger networks need very extensive examination. Another approach to the analysis of connectionist models, which assumes that NNs are black boxes, is to examine the variation of network performance while varying some properties of the data (Plaut et al. 1996; Stoianov, Stowe, and Nerbonne 1999, among others). For example, one can vary word frequency, length, etc., and study network error. In the case of modeling human cognitive functions, this approach has the advantage of allowing comparison between cognitive systems and their artificial connectionist models. In this study of phonotactics, we model lexical decision and will therefore compare the neural network behavior to psycholinguistic studies on lexical decision.

Frequency is one of the most investigated word characteristics. Numerous previous studies have demonstrated that the ease and time with which spoken words are recognized are monotonically related to experienced frequency of words in the language (Luce, Pisoni, and Goldinger 1990; Plaut et al 1996).

Another important phenomenon that affects word perception is similarity to other words. *Similarity neighborhood* is defined as the collection of words that are phonetically similar to a given stimulus word. Thus, words might be in a dense neighborhood with many phonetically similar words or in a sparse neighborhood with few phonetically similar words. It is still controversial how this influences cognitive processes (Balota, Paul, and Spieler *in press*) and how it should be measured. Intuitively, it seems likely that words with larger neighborhoods are easier to access due to many similar items, but from another perspective these words might be difficult due to nearby competitors. However, in the more specific lexical decision task, the overall activity of many candidates is likely to increase familiarity thereby increasing the ease of lexical decisions.

As far as neighborhood measurement is concerned, a popular approach is the so-called *Coltheart-N* measure that counts the number of words that might be produced by replacing a single letter with some other letter of the alphabet. We adopt another definition, that is sensitive to similarity of subsyllabic elemens, and which regards words as similar when they share two of the sub-syllables – onset, nucleus and coda. Empty onsets or codas are counted as same. The word neighborhood is computed by counting the number of the similar words. This estimation is another approximation of the neighborhood size, but the complexity of the distance-measuring problem is high, so we reduced it by probing for sub-syllables rather than for units of variable size, starting from a single phoneme. This decreases computational time and uses phonological units that are accepted by most linguists. The neighborhood size of the corpus we used ranged from 0 to 77 and had mean value of $30, \sigma = 13$. For example, the phonetic neighborhood of the Dutch word $bruts$ is given in (2). Note that the neighborhood list only contains Dutch words.

(2) $brIts, brots, bruj, bruit, bruk, brur, brus, brut, buts, kuts, puts, tuts$

With regard to the property *word length*, longer words provide more evidence – more phonemes are available to the network. On the other hand, the network error accumulating in recursion increases the chance of errors. Hence, we expect U-shaped patterns of dependence when varying length. Such a pattern was observed in another study on modeling grapheme-to-phoneme conversion with SRNs (Stoianov, Stowe, and Nerbonne 1999). Static connectionist models have difficulties in simulating word length effect, because of their static word representations, while recurrent networks naturally capture these phenomena due to the dynamic processing.

5.1 Findings

In connectionist phonotactics modeling, we can compare network performance with scores in human lexical decision tasks. Lexical decision scores correspond to SRN score in word recognition, that is, word acceptance and non-word rejection. Similarly, we can find a SRN correlate to Reaction Times (RT) in the lexical decision task. One possible SRN counterpart of RT is related to the network confidence, or the amount of evidence that the test string is a word from the training language. The less confident the network, the slower the reaction, which can be implemented with a lateral inhibition (Haykin 1994; Plaut et al. 1996). The network confidence might be expressed as the product of the activations of the neurons corresponding to the phonemes of the test word. A similar measure, which we call *uncertainty* U is the negative sum of neuron activation logarithms, normalized with respect to word length $|w|$ (3). Note that U varies inversely with confidence. Less certain sequences get higher (positive) scores.

(3) $U = -\frac{1}{|w|} \sum_{i=1}^{|w|} ln(n_{c_i})$

The results of error projection regarding the above properties are given in Table 2. As expected, there was a strong frequency effect. Error dropped significantly as frequency slightly increased and continued dropping smoothly (Table 2a). Length showed the expected pattern as well, with higher error for short words that decreased with increasing length but slightly increased for very long words, which we explained earlier via accumulated error (Table 2c). With regard to neighborhood density, SRNs were more confident on words with high-density neighborhoods

(a)

Freq effect	Low	Mid	High
U	2.30	2.20	2.18
Error (%)	3	1.5	0.5

(b)

Neighb. effect	Low	Mid	High
U	2.62	2.30	2.21
Error (%)	10.6	4.5	0.8

(c)

Length effect	2	3	4-6	7	8
U	2.767	2.43	2.20	2.11	2.15
Error (%)	8.8	3.4	1.9	3.7	6.2

Table 2: Effect of (a) frequency, (b) neighborhood density and (c) length effect on word uncertainty U and error.

rather than on words with low-density neighborhood (Table 2b). This pattern confirmed the hypothesis of the lexical decision literature that larger neighborhoods make it easier for words to be recognized as such.

5.2 Syllabic structure

Phonotactic constraints might hint at how the stream of phonemes is encoded in the language processing system. The popular phoneme, syllable and word entities may not be the only units that we use for lexical access and production. There are suggestions that in addition, some sub-syllabic elements are involved in those processes, that is, the syllables might have not linear structure, but more complex representations (Kessler and Treiman, 1997). For that purpose, we will analyze how the error is located within words with respect to the following sub-syllabic elements – *onset, nucleus* and *coda*. The particular hypothesis we will test is whether the Dutch monosyllables follow structure (4) that was found in English as well (Kessler and Treiman 1997).

(4) (Onset – Rhyme (nucleus – coda))

The distribution of error within words (Table 3a) tells us that the network makes more mistakes at the beginning than at the end of words, where SRN becomes more confident in its decision. The more precise analysis of the error position in the onset, the nucleus and the coda reveals other interesting phenomena (Table 3b). First, error within the coda increases at the coda's end. We attribute this to error accumulated at the end of the words. The mean entropy in the coda $(1.32, \sigma = 0.87)$ is smaller than the mean entropy in the onset $(1.53, \sigma = 0.78)$, where we don't observe such effects, that is, looser constraints are not the reason for this error increase. Next, the error at the transition onset-nucleus is much higher than the error at the surrounding positions, which means that the break between onset and *rhyme* (the conjunction nucleus-coda) is a significant. This distribution is consistent with the statistical findings that the entropy is larger in the *body* (the conjunction onset-nucleus) $(3.45, \sigma = 0.39)$, than in the rhyme $(1.94, \sigma = 1.21)$. All this data supports the earlier hypothesis, that onset and rhyme play significant

(a)

Word Position	2	3	4	5	6	7	8
Error(%)	4.8	2.0	1.5	1.1	1.0	0.3	0.00

(b)

Sub-syllables	Onset		Nucleus	Coda			
Relative Position	2	3	1	1	2	3	4
Error(%)	2.3	0.0	5.4	1.1	2.2	5.0	4.5

Table 3: Distribution of error along words (a) and within sub-syllables (b). Word and Onset positions start from 2, because the prediction starts after the first phoneme. Nucleus and Coda positions are renumbered from 1 because they do not correspond to word position.

role in lexical access and that the syllabic structure (4) is valid for Dutch too.

6 Discussion

Phonotactics restricts the enormous variety of sequences of phonemes to those strings that are valid in a given language. Phonotactic rules also determine the phonological grammar of words; and there is a number of lexical tasks that might make use of this knowledge. These tasks might be divided into two categories – analytical and predictive. In the first group, we include *segmentation* and *lexical decision* tasks, where the phonotactic rules determine possible word boundaries in the stream of phonemes or judge the degree of well-formedness of a given sequence, respectively. The opposite, predictive task expresses the anticipations of which phonemes might follow a certain sequence, or left context. When extended recursively, these expectations generate hypotheses of words that begin with this left context. These lexical tasks inevitably participate in speech processing, which requires phonotactics, represented in one or another way.

In this paper we use Simple Recurrent Networks for encoding phonotactics. SRNs not only belong to the family of distributed connectionist models, but they are dynamical, too. This property distinguishes them from the static connectionist models and enables them to process long distance dependencies in sequential data. Static models can process sequences by employing two techniques. The first, *window*, technique was used in the NETtalk system (Sejnowski and Rosenberg 1987), but it restricts the scope of sequential dependencies. Another approach is to encode lexemes statically (Seidenberg and McClelland 1989; Plaut et al. 1996 among others), which ignores dynamism at the phonetic level. We prefer the dynamic approach and use SRNs.

The network learned the phonotactics of Dutch monosyllables without any background knowledge, but only by observing words which were sequentially presented to the network – one phoneme at a time to the input layer with one phoneme targeted at the output layer. If we interpret phonotactics as context-dependent prediction, the network learned this mapping with slight error, which was demonstrated with variety of estimation methods. These methods were chosen with regard to the specific task to be accomplished – lexical decision or phonemic prediction.

Further, SRNs were not only able to learn phonotactics, but they also exhibited behavior similar to the psycholinguistic data reported in other studies. We showed frequency, neighborhood, length and positional effects that are consistent with data reported for humans in similar tasks. Of course, SRNs are not the only connectionist model capable of dynamic processing, nor may they be characterized as the most biologically plausible neural network, but we demonstrated that connectionist models with relatively simple structure have the capacity to learn and model phonotactics. And NNs are structurally much closer to the human mental architecture than any other symbolic or stochastic methods. Even the popular genetic algorithms trained on phonotactics (Belz 1998) do not explain better how people process words, firstly, because we learn phonotactics in the course of our life, and secondly, because evolutionary approaches use completely different, stochastic learning techniques.

It is unclear whether a phonotactics module must be postulated in human language processing. Some tasks might access pronunciations as a whole and make little use of phonotactics. On the other hand, there are many indications of the psychological significance of phonotactics. Errors of speech and understanding tend to favor phonotactically plausible strings above others, and foreign accents tend to preserve the phonotactics of the native language. If it should turn out that these indications of the psychological reality of phonotactics are misleading, then SRNs and the work reported on here may still be of value, since the problem of sequential processing certainly exists at some levels of natural language.

It is difficult to exhaust all the issues related to language modeling. Simple Recurrent Networks seem to be a good model for lexical structure – phonotactics, graphotactics and orthography-to-phonology mapping have been modeled successfully with this type of NN, but there are other problems that are still to be solved. In subsequent studies, we plan to investigate network damage and its potential relation to similar processes in humans – various types of aphasia.

Another, even more challenging problem is syntax modeling. In the current study we demonstrate the SRN capacity to learn lexical grammar. Syntax is sentence grammar, but it is hard to learn because there are many more than 26 or 44 input elements (words), and there are even more sequences (sentences). One possible solution is to use word tags. This restricts the input elements to some 50 tags, significantly decreases the number of the sequences, and simplifies the learning task. Another solution to sentence modeling using words, is to employ a connectionist model Recurrent Autoassociative Networks (Stoianov 1999), which develop static distributed representations of words. However, this will require larger neural network and more computational resources.

References

Balota, D., Paul, S. and Spieler, D.(in press), Attentional control of lexical processing pathways during word recognition and reading.

Belz, A.(1998), Discovering phonotactic finite-state automata by genetic search, *36th Annual Meeting of the Association for Computational Linguistics and 17th Int. Conference on Compuational Linguistics*, Vol. 2, pp. 1472–1474.

Cairns, P., Shillcock, R., Chater, N. and Levy, J.(1977), Bootstrapping word boundaries: A bottom-up corpus-based approach to speech segmentation, *Cognitive Psychology* **33**, 111–153.

Cleeremans, A., D.Servan-Schreiber and J.L.McClelland(1989), Finite state automata and simple recurrent networks, *Neural Computation* pp. 372–381.

Dell, G., Juliano, C. and Govindjee, A.(1993), Structure and content in language production: A theory of frame constraints in phonological speech errors, *Cognitive Science* **17**, 145–195.

Elman, J. L.(1990), Finding structure in time, *Cognitive Science* **14**, 213–252.

Gasser, M.(1992), Learning distributed representations for syllables, *Proc. of 14th Annual Conference of Cognitive Science Society*, pp. 396–401.

Haykin, S.(1994), *Neural Networks*, Macmillian Publ, NJ.

Kessler, B. and Treiman, R.(1997), Syllable structure and the distribution of phonemes in english syllables, *Journal of Memory and Language* **37**, 295–311.

Laver, J.(1994), *Princes of Phonetics*, Cambridge Univ.Press, Cambridge, UK.

Luce, P. L., Pisoni, D. B. and Goldinger, S. D.(1990), Similarity neighborhoods of spoken words, *in* G. T. M. Altmann (ed.), *Cognitive Models of Speech Processing*, A Bradford Book, Cambridge, Massachusetts, USA.

McQueen, J.(1998), Segmentation of continuos speech using phonotactics, *Journal of Memory and Language* **39**, 21–46.

Nerbonne, J., Heeringa, W., van den Hout, E., van den Kooi, P., Otten, S. and van de Vis, W.(1996), Phonetic distance between dutch dialect, *in* W. G.Dueux and S.Gillis (eds), *Proc. of Computational Linguistics in the Netherlands*, pp. 185–202.

Plaut, D.(1997), Structure and function in the lexical system: Insights from distributed models of word reading and lexical decision, *Language and Cognitive Processes*.

Plaut, D. and Kello, C.(1998), The emergence of phonology from the interplay of speech comprehension and production: A distributed connectionist approach, *in* B. MacWhinney (ed.), *The emergence of Language*, Erlbaum, Mahweh, NJ.

Plaut, D., McClelland, J., Seidenberg, M. and Patterson, K.(1996), Understanding normal and impaired word reading: Computational principles in quasi-regular domains, *Psychological Review* **103**, 56–115.

Rodd, J.(1997), Recurrent neural-network learning of phonological regularities in turkish, *Proc. of Int. Conf. on Computational Natural Language Learning, Madrid*, pp. 97–106.

Sang, E. T. K.(1995), The limitations of modeling finite state grammars with simple recurrent networks, *Proc of 5-th CLIN*, pp. 133–143.

Sang, E. T. K.(1998), *Machine Learning of Phonotactics*, PhD thesis, University of Groningen.

Seidenberg, M.(1997), Language acquisition and use: Learning and applying probabilistic constraints, *Science* **275**, 1599–1603.

Seidenberg, M. and McClelland, J.(1989), A distributed, developmental model of word recognition and naming, *Psychological Review* **96**, 523–568.

Sejnowski, T. and Rosenberg, C.(1987), Parallel networks that learn to pronounce english text, *Complex Systems* **1**, 145–168.

Shillcock, R., Cairns, P., Chater, N. and Levy, J.(1997), Statistical and connectionist modelling of the development of speech segmentation, *in* Broeder and Murre (eds), *Models of Language Learning*, MIT Press.

Stoianov, I. P.(1998), Tree-based analysis of simple recurrent network learning, *36 Annual Meeting of the Association for Computational Linguistics and 17 Int. Conference on Compuational Linguistics*, Vol. 2, pp. 1502–1504.

Stoianov, I. P.(1999), Recurrent autoassociative networks and sequential processing, *International Joint Conference on Neural Networks*, Washington DC.

Stoianov, I. P., Nerbonne, J. and Bouma, H.(1998), Modelling the phonotactic structure of natural language words with simple recurrent networks, *in* v. H. Coppen and Teunissen (eds), *Computational Linguistics in the Netherlands, 1997*, Rodopi, Amsterdam, NL, pp. 77–96.

Stoianov, I. P., Stowe, L. and Nerbonne, J.(1999), Connectionist learning to read aloud and correlation to human data, *21 Annual Meeting of the Cognitive Science Society, Vancouver*, Vancouver.