CYKO
The Self Improving Computational kOmmunicator
Project for 6.836

Ben Yoder
Peter Gorniak

May 17, 2001

Abstract

How do words get created? Part of the answer to that question must deal with the acoustic properties of strings of phonemes. We investigate the distinctiveness of such phoneme strings by considering a population of individuals that must successfully acoustically communicate to prove their fitness. Each individual possesses its own dictionary of phoneme strings. An evolutionary algorithm evaluates the individual's fitness by making it speak each dictionary entry and try to identify the word spoken. The evaluation takes place in the real world through a speaker and microphone, and is thus subject to realistic acoustic conditions and noise levels. We show that individuals evolve over time that communicate extremely well, and discuss the nature of the changes that occur in the dictionary as well as the effects of noise on the evolutionary process.

1 Introduction

Human beings have evolved to use language as a means of communication. This form of communication is a highly sophisticated process that involves a shared vocabulary with attached semantic interpretations, a grammar used to arrange vocabulary items appropriately, a way to figure out what is appropriate given the current situation, and finally a means of converting what is to be said to an acoustic signal that can be decoded by the receiver. Every single facet of language communication raises many questions: How does it work? How did it come about? How does it fit in with the rest of the system? Given that human language use seems to stand at the current end of a long evolutionary chain, it is tempting to apply genetic algorithms to answer at least some of these questions. One can, for example, study the way a language as a symbolic system emerges from communication in a population (see for example Luc Steels: *Synthesising the origins of language and meaning using co-evolution, self-organisation and level formation*, 1996, In: Hurford, J., C. Knight and M. Studdert-Kennedy (ed.) (1997) Evolution of Human Language. Edinburgh Univ. Press. Edinburgh), or how we came to use our vocal tracts to speak (see for example Bart de Boer: *Emergence of vowel systems through self-organisation*, 2000, AI Communications 13 (2000) pp. 27-39)

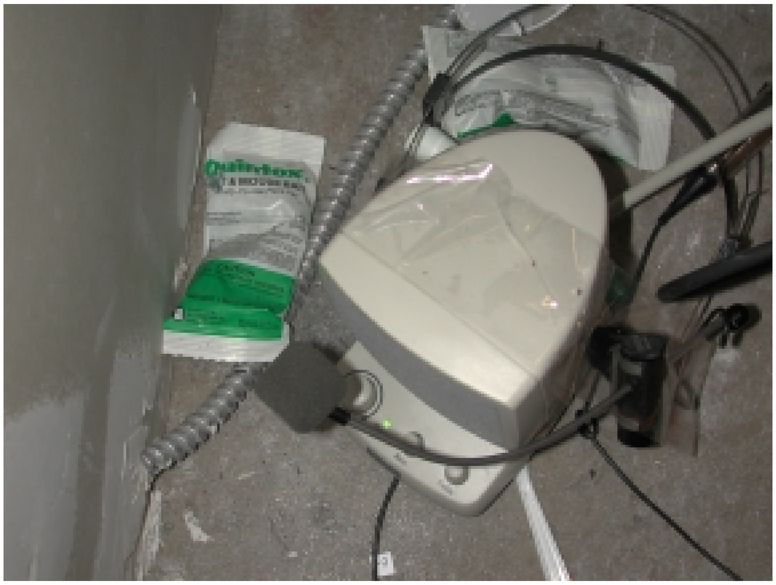

Figure 1: Speaker and Microphone under Floor

Here, we present our investigation into how human beings came to create the words they speak. We use an evolutionary algorithm as a backbone for this investigation, not only because of the noted evolved nature of language, but also because this problem constitutes an example of being able to carry out the evaluation of a fitness function in the real world. There is no need to simulate spoken words, rather they can actually be spoken and heard. This is a somewhat unique opportunity, as it is often hard or impossible to computationally evolve creatures outside of simulation.

We focus on the problem of the discernability of spoken words. We assume an existing set of phonemes that these words are constructed from (in our case 41 synthesized English phonemes). The question thus becomes how to arrange these phonemes into words such that the words can be easily distinguished when spoken aloud. With the system able to evolve discernible words in a relatively noiseless environment, we assess the properties of the resulting words and compare them to real spoken languages. Finally, we change the environment by adding a noise source and compare performance of the system to the noiseless case. Figure 1 shows our experimental setup with the speaker and microphone used for evaluation under the floorboards of our office.

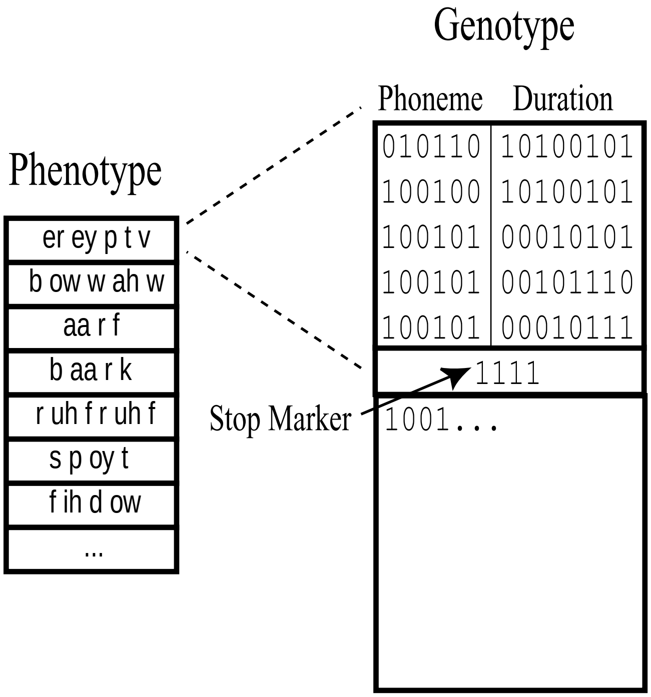

Figure 2: Genetic encoding of a CYKO's vocabulary

2 Cyko Genetics

The individuals in our evolving population are called CYKOs (Self Improving Computational kOmmunicators.) Each CYKO consists of a mapping from integers to phoneme strings, its current vocabulary. We consider this the CYKO's phenotype. This phenotype is produced from an underlying binary genotype. The correspondence between phenotype and genotype is shown in figure 2. Each phoneme is encoded using 14 bits. The first 6 bits specify the phoneme number out of the 41 English phonemes known to the speech synthesizer, including silence, using standard binary encoding. The upper 23 values of this field do not correspond to phonemes and are ignored in converting the genotype to the phenotype. In addition, the phoneme corresponding to silence is also ignored in this conversion. The remaining 8 bits specify the duration of the phoneme in milliseconds as a value between 0 and 255. We use Gray Code encoding to specify this number, which is an encoding in which two adjacent numbers' encoding differs by only one bit. While still subject to large changes, this encoding allows changes to a similar value for all number with a single bit flip, a property that is lacking in the standard encoding. The end of a word is marked by a 4 bit long stop marker that has all bits set to 1.

To start the evolutionary process, a population of CYKOs is randomly generated. Each CYKO starts with the same size vocabulary with word lengths varying randomly around an average by a few phonemes. Most of our runs start with word lengths between 8 and 12 phonemes. Each phoneme initially last 100 milliseconds. Phonemes for each word are picked randomly from all possible synthesizer phonemes excluding the silence phoneme. During each generation of the evolutionary process, each CYKO is evaluated as described in section 3. At the end of evaluation, a percentage of the best scoring CYKOs is transferred unchanged to the next generation. The population for the next generation is then grown to the fixed population size by randomly mating a possibly different percentage of the top scoring CYKOs. During our runs, we usually kept both the fraction of CYKOs to retain and the fraction of CYKOs to use in mating at 0.2.

In mating, a new genotype is created from two parent genotypes. Each word has a 50% chance of being produced by either parent. The parent selected then constitutes the main source for the genes for each word, but at each phoneme boundary there is a chance of crossing over and using the other parent as a source for that phoneme. Each bit read has a low chance of being flipped to its opposite value before being inserted into the child's genotype. At the start of each phoneme read, the first four bits are read to see whether they constitute a stop marker. If not, a chunk of 14 bits is read and transfered to the child under the conditions just described. If a stop marker is encountered in the parent that is the main source for the current word, or if the end of the genome of that parent is reached, a stop marker is written to the child's genome and reading continues if there is genetic material left to be read in at least one parent. While not being transfered to the child unless crossover occurs, genes are read simultaneously in the parent that is not the main source for the current word. If a stop marker is encountered in this genome, reading halts for it until the other parent reaches a stop marker or ends. Crossover cannot occur anymore for the current word in that case. If either parents' genome ends, the other parent's genome is used exclusively (no crossover and no selection for words occurs) until it also ends. Note that a stop marker is subject to mutation just like the rest of the genome. Thus, markers can be deleted or created and in the process lengthen or shorten words and change dictionary size. As the genome is read in 14 bit chunks (except for the markers themselves) and the highest values of the phoneme encoding do not map to actual phonemes, stop markers that consist solely of 1s do not correspond to any phoneme encoding.

Finally, each genotype is converted to a phenotype (as stated above, silence and non-existent phonemes are ignored) and the next generation stands ready to babble away.

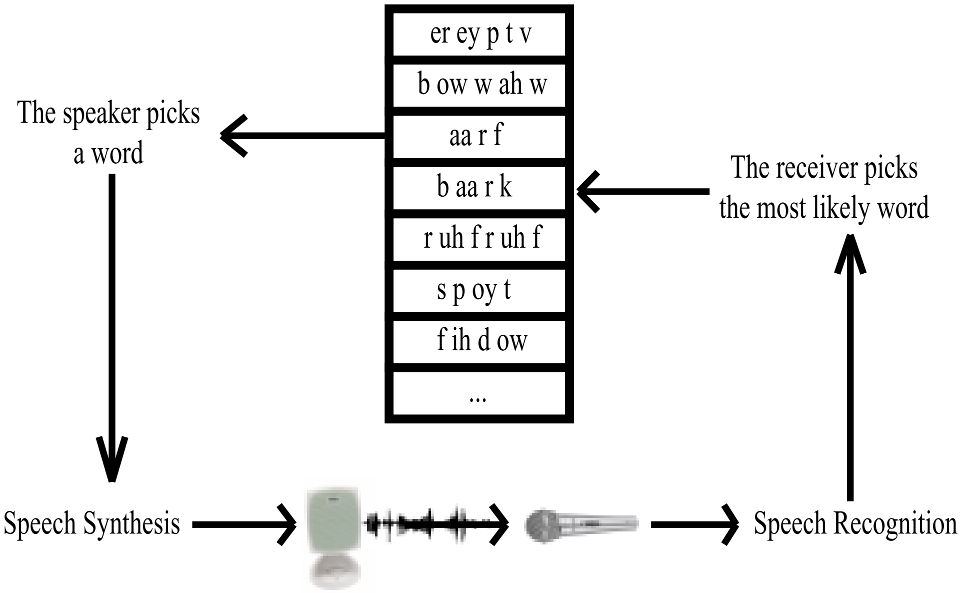

Figure 3: The evaluation of a single word.

3 CYKO Evaluation

3.1 Overview

A CYKO contains a number of different words which are made up of strings of phonemes. To determine the fitness of the CYKO, each word is evaluated in turn, receiving a score of one if the word is correctly recognized, and zero if it is incorrectly recognized. Thus, for a vocabulary size of 20, the best score possible is a 20 while the worst score is zero.

The process depicted in figure 3 evaluates a single word:

1. The string of phonemes and their durations for the word are used

as input to a speech synthesizer which then plays the sounds created through a speaker.

2. A speech recognizer listens through a microphone and matches what it hears to the most likely word.

3. The score is incremented by one if the word recognized by the recognizer is the same as the one said by the synthesizer.

Thus, the two major components necessary for evaluation are speech synthesis and speech recognition.

3.2 Speech Synthesis

The speech synthesis program we used is called MBROLA (see http://www.festvox.org/mbrola/ for details) which is based on the concatenation of diphones. It takes as input a list of phonemes, durations for the phonemes, and a piece-wise linear designation of the pitch over each phoneme. In order to cut down on the number of free variables, only the duration was modified. This speech synthesis program used a set of 41 phonemes including silence, but the words were constrained to include only non-silence phonemes. As the source code for this program was not present, a system command was used to run the synthesizer as from the command line, and it created a ".wav" file upon completion. This ".wav" file was then played through a speaker. In order to synchronize the speech recognition system with the synthesis system, a signal was sent to the recognition system upon the start and the end of each word. This allowed the decoding to only take place near the time the word was spoken.

3.3 Speech Recognition

The speech recognition system used here is a simple Hidden Markov Model (HMM) based system using Mel Frequency Cepstral Coefficients (MFCCs) (see *Fundamentals of Speech Recognition* by Radiner and Juang, 1993 for details). Acoustic observations were obtained every 10 ms. by windowing the sound with a 20 ms. Hamming window, and from this computing the MFCCs. These MFCCs were then used to train uniphone HMMs. The HMMs have continuous density output probabilities which take the form of a Gaussian Mixture Model (GMM). The uniphone HMMs were first initialized using a segmental k-means algorithm, and then using the Baum-Welch algorithm for Expectation Maximization (EM). Then, a state based bottom up clustering method was used to create triphones as well as left and right biphones by grouping those contexts which were acoustically similar. These triphones were then retrained using the clusters from the trained uniphone through several more rounds of Baum-Welch. To evaluate its accuracy, it was tested for phoneme level accuracy on the TIMIT data set and scored reasonably well, as seen in table 1.

Name	Percent Correct	Percent Error
SPHINX	73.8	33.9
HTK	76.7	27.7
our system	74.0	31.0

Table 1: Phoneme recognition performance on the TIMIT database compared to several other recognizers

The speech recognition system used to evaluate the CYKOs was running in a real time mode where it received MFCCs as they arrived from a front end which was constantly reading sound from the microphone. These MFCCs would be ignored, unless it was told by the evaluator that an utterance was in progress, in which case it would evaluate them using a Viterbi search with a relatively large beam width to find the most likely word. It would then send this word back to the evaluator which would award the current CYKO a point if the word was the correct one or nothing if the word was not.

The main thing constraining the scalability of our system to large vocabulary sizes is the amount of time necessary to evaluate each CYKO. As the utterance must pass through the real world (out through the speaker and in through the microphone), This means that for a size 30 vocabulary, with a word taking about 2 seconds to say, the evaluation of each CYKO takes about a minute.

4 Results

The CYKOs were evaluated at varying levels of noise and with varying population sizes. We evaluated the CYKOs with vocabulary sizes of 10, 20, and 30 words, with population sizes of 10, 20, and 30 respectively. We also experimented with varying sources of noise to determine several things. First of all, to make sure that the CYKOs were completely dependent on the sound coming from the microphone, and secondly to determine whether the CYKOs could evolve "away" from the current acoustic situation (the CYKOs may use phonemes which are least distorted by the noise). All runs were done with a mutation rate of .2%, a crossover prob. of 30%, and a retain and reproduction fraction of .2.

The results for a size 10 vocabulary and a population of size 10 in the presence of no noise are shown in figure

As can be seen in figure 4, the CYKOs quickly reach a maximum of 10, and stay near there for most of the generations. It is more interesting to run this with a larger vocabulary size so that the increase is more gradual.

The results for a size 20 vocabulary with a population of size 20 are shown in figure 5.

As can be seen in figure 5, the CYKOs take much longer to achieve the maximum of 20 points. The CYKOs are also extremely sensitive to noise.

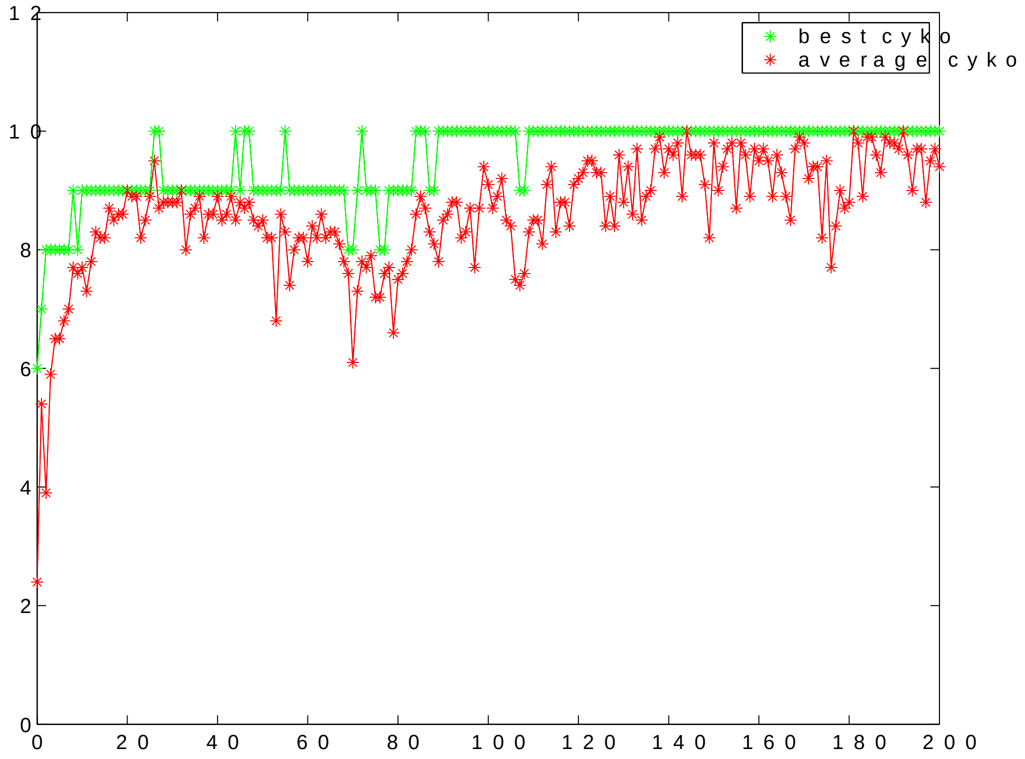

Figure 4: Size 10 vocabulary with no noise with population of size 10

This run was done overnight, as evaluation time for 20 CYKOs with a vocabulary size of 20 is about 15 min. The dip at the end of this run we believe is due to the extra noise introduced when the lab began to get more noisy during the morning. Even with its placement under the floorboards, it was easy to verify by observation that the CYKOs were strongly affected; making noise at appropriate times would cause mistakes in the word recognition.

In order to verify that the performance of the CYKOs was directly de-

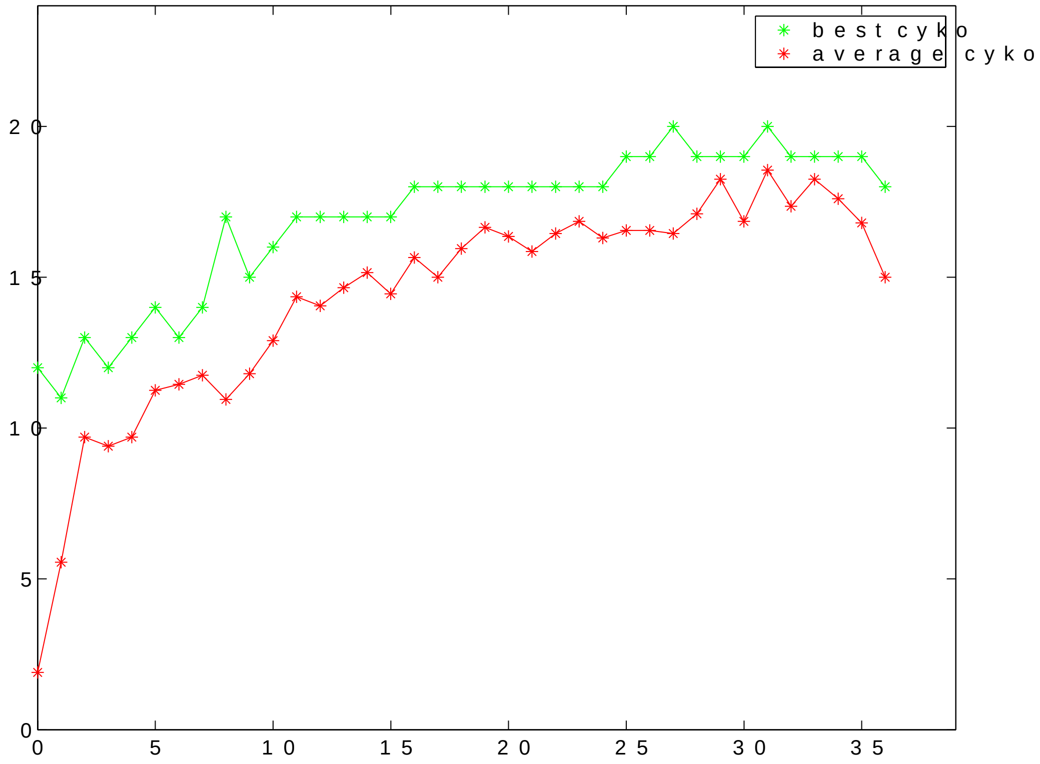

Figure 5: Size 20 vocabulary with no noise with population of size 20

pendent on the sounds it was receiving from the microphone, we placed a noise source (a speaker continually saying “She had your dark suit in greasy wash water all year.”) near the microphone, and turned the volume up fairly high. As shown in figure 6, the performance deteriorated completely in the presence of high noise.

Ideally, CYKOs should also be able to evolve away from noise sources. In the next experiment, we set up a CYKO away from noise, and then had

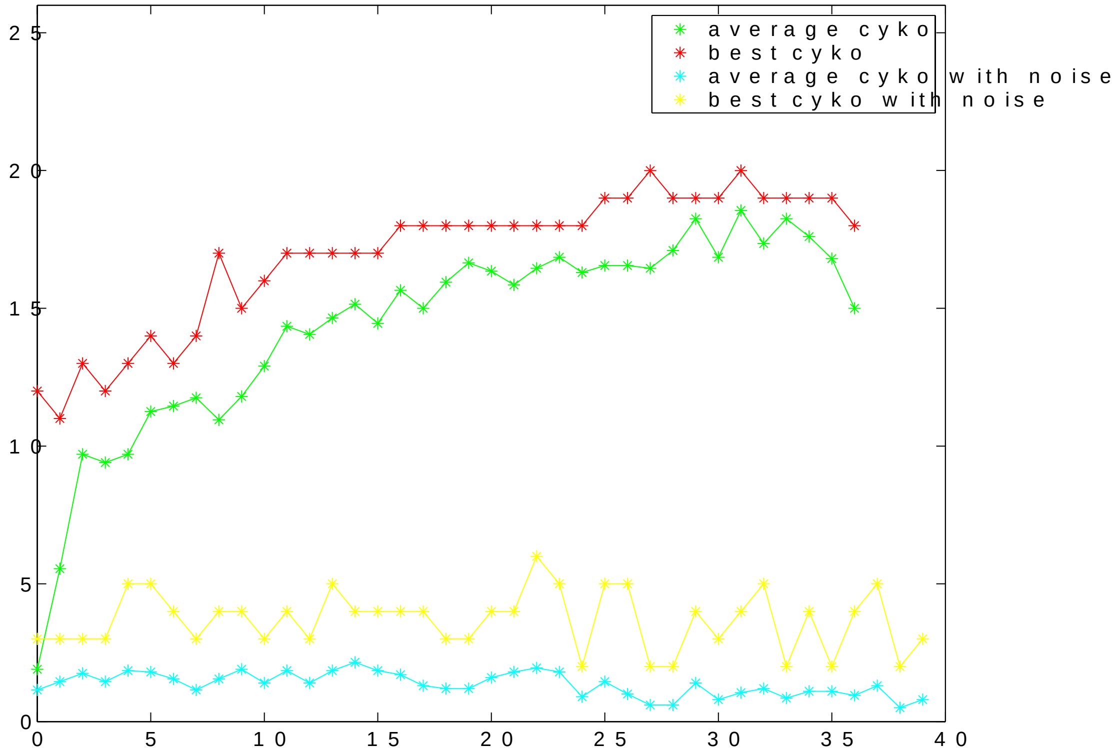

Figure 6: Size 20 vocabulary with no noise with population of size 20 in the presence of strong noise

it in the presence of people talking for several hours. Figure 7 shows that in the presence of a small amount of noise, the system was able to at least partially adapt and recover in the presence of a small amount of noise. It is difficult to tell exactly how much adaptation was occurring, as the amount of talking nearby had a tendency to come and go, and the amount of data collected is not statistically significant, but the graph is included as at the very least, it shows some of the systems reaction to noise.

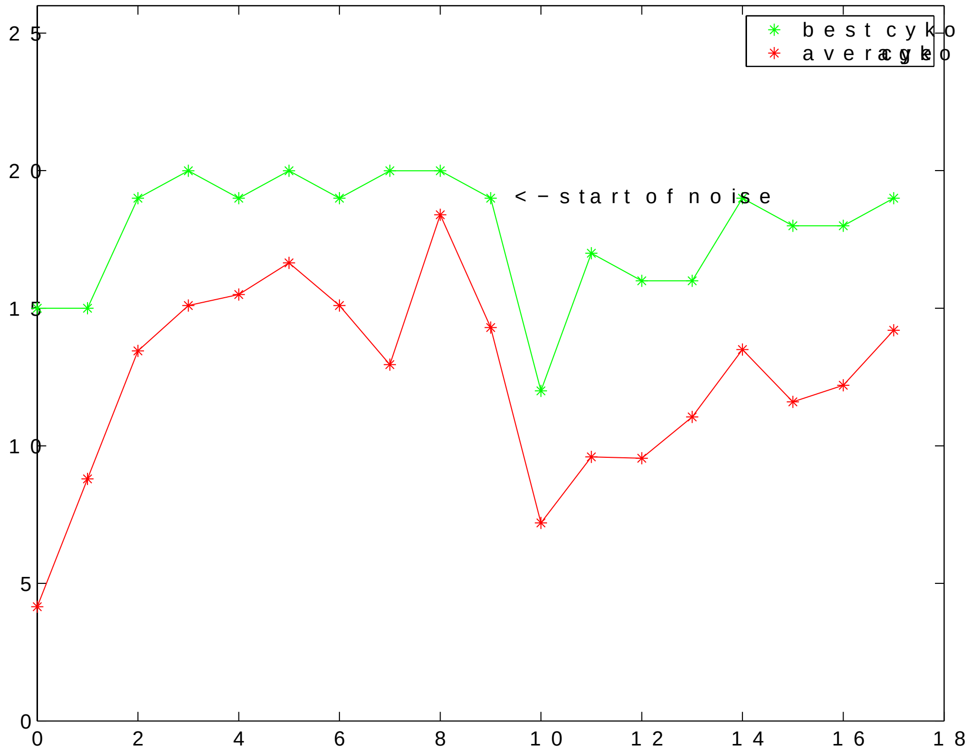

Figure 7: Size 20 vocabulary with noise starting at generation 9 and continuing throughout

The largest vocabulary used was a vocabulary of size 30 which was run with a population of size of 30. Figure 8 shows the results of this system. This system was run for 50 iterations, requiring 900 evaluations of a word per generations, and taking about 2 seconds to evaluate each word. Thus, to run it for 50 generations took about 25 hours. As shown, the system still eventually levels off near the maximum value.

The difficulty in this experiment is in determining why the CYKOs im-

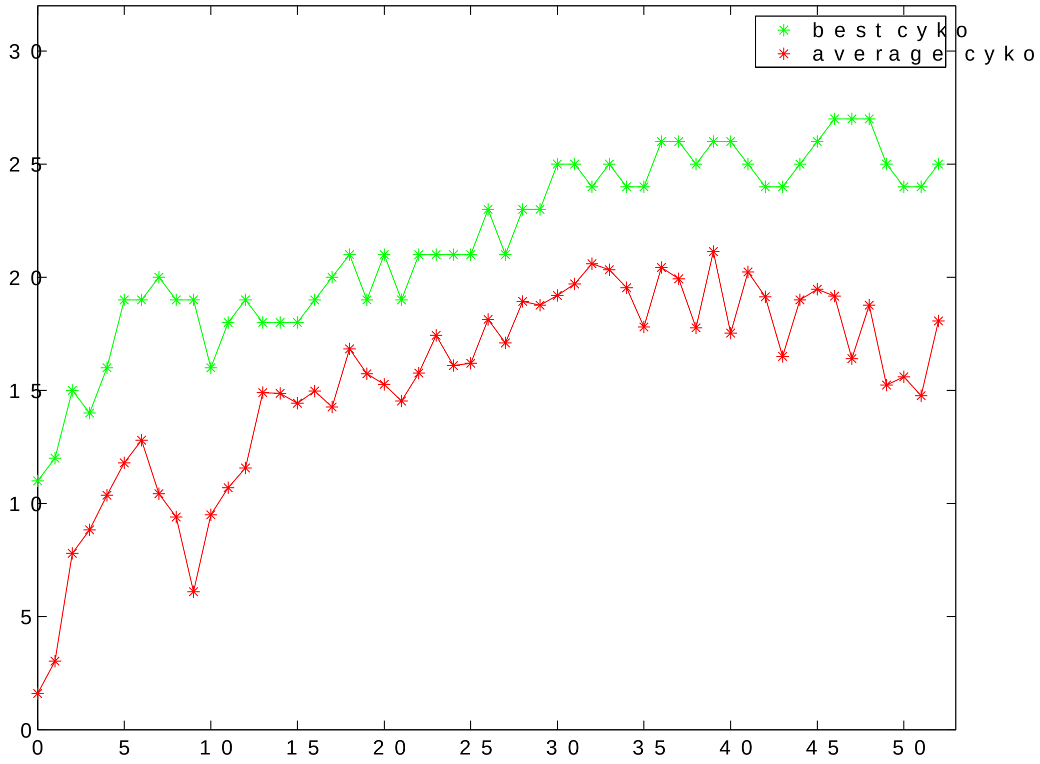

Figure 8: Size 30 vocabulary with a size 30 population and no noise

prove. There appear to be two levels of improvement, one is the sudden sharp improvement at the beginning, and the other is a slow gradual improvement. We had difficulty in determining the driving factor for either of these improvements from looking at phoneme probabilities or other factors.

The sudden sharp improvement occurs because the decoder often continually picks the same word. The decoder is given a list of possible words, as well as a word consisting only of silence, and returns the most likely word string. As the decoder starts listening a few fractions of a second before the synthesizer speaks, and continues slightly longer than it takes the synthesizer to "speak" the word, the idea is that it will match the beginning and end to silence, with the correct word sandwiched in between. The difficulty is that often the decoder picks a single word to account for the silence instead of the silence word, and hence will continually pick the same word for all the utterances of the CYKO.

We believe this is due to odd effects of the speech synthesizer. As is standard practice in speech recognition, the speakers voice is analyzed to compute average MFCCs which are then subtracted from the MFCCs generated in the utterance. As the recognition system was going to be analyzing words from the speech synthesizer, we calibrated the system by computing the average MFCCs for the synthesis program. Due to the synthetic nature of the synthesizer's voice, certain phonemes in specific contexts apparently now behaved more like silence after the average MFCC was subtracted. Thus words with those contexts were thrown out early on, and accounted for the sudden sharp improvement.

The slow gradual improvement after this we attribute to a better choice of phonemes and contexts although we were unable to determine what the specific choices were. To compare phoneme choices from the evolved CYKOs to those chosen in English text, we grouped the phonemes into two rough categories, consonants and vowels, and then computed some bigrams such as P(C|V) (the probability of a consonant given a vowel was the preceding phoneme), P(V|V) (the probability of a vowel given a vowel was the preceding phoneme), and so on. Figure 9 shows the results of these computations in a CYKO and in given English text computed from the Wall Street Journal speech corpus. As can be seen in figure 9, the probabilities for the CYKO and those in English text are substantially different. Thus its improvement

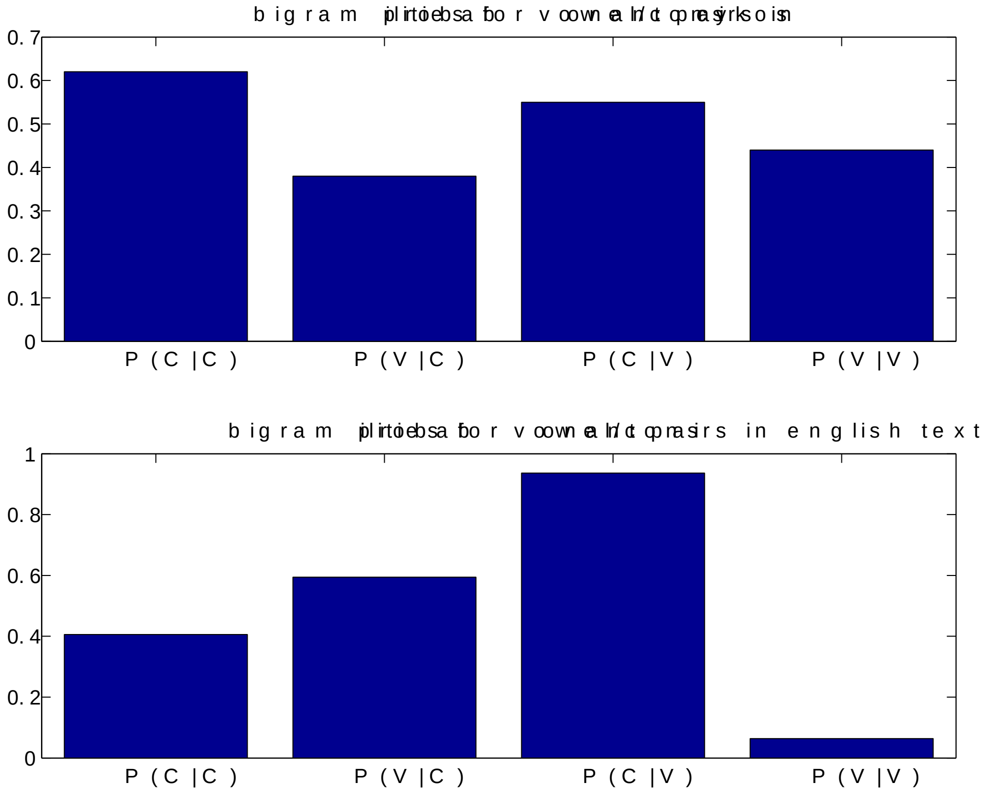

Figure 9: comparison of bigram probabilities for consonant/vowel phonemes

was not due to grouping phonemes in the same way that English is grouped.

We also looked at phoneme frequencies by generation to see if certain phonemes were preferred. Figures 10 and 11 show the phoneme probabilities for each phoneme over several generations.

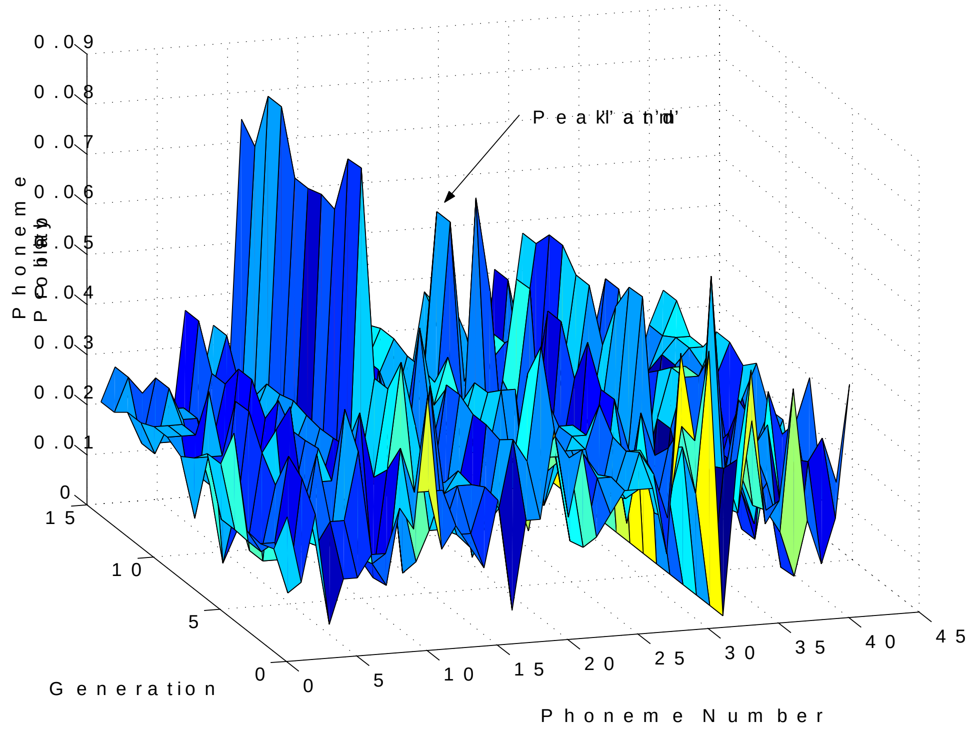

Figure 10: Comparison of phoneme probabilities over generations

These plots were created from a runs of 30 individuals with a vocabulary size of 30. As shown in the figures, one of the peaks in each graph corresponds to the 'l' and 'm' phonemes. It is difficult to tell whether these phonemes are being selected, or whether this is random chance. Note that

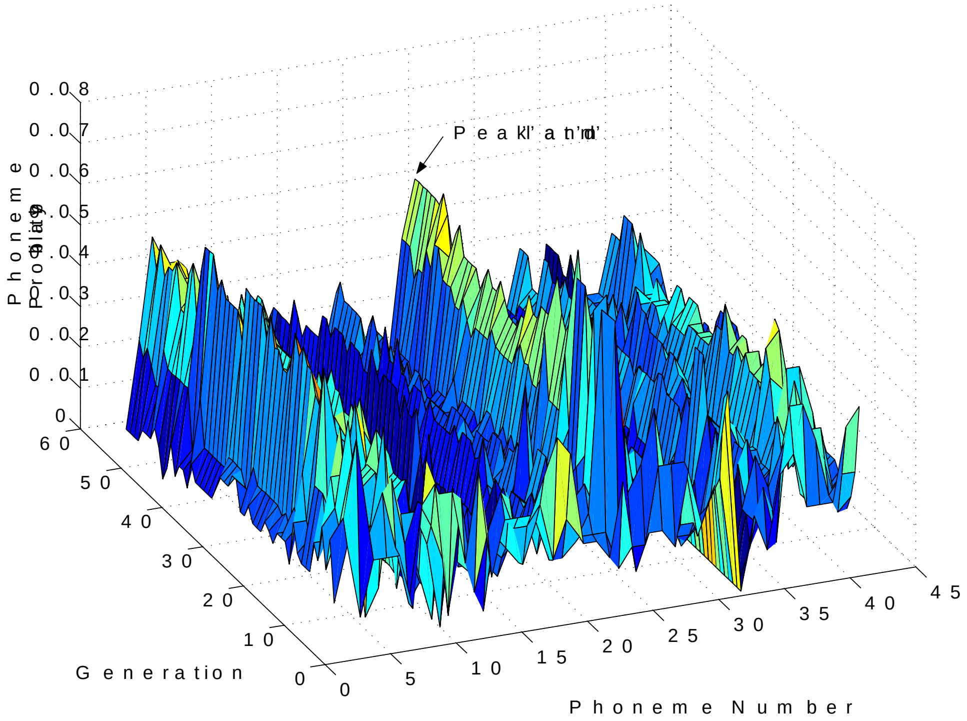

Figure 11: Comparison of phoneme probabilities over generations

the largest peak in figure 10 is due to the 'dh' phoneme and completely vanishes in figure 11. In order to tell what phonemes would be most recognizable, much more data would be necessary. It is clear that the CYKOs do improve, however, and while the exact method is unknown, we still believe that it is due to selecting recognizable phonemes in appropriate contexts.

5 Conclusion

We have shown here that acoustically distinguishable words based on a fixed set of phonemes can be evolved using a genetic algorithm. This is an example of a fitness function carried out in the real world, and thus subject to inherent environmental noise as well as slow execution times. In all reasonably noiseless cases our CYKOs reached high communication success levels. We also showed that noise levels do indeed affect performance, and have some preliminary indication that the system adapts over time to deal with noise.

There are two main problems with our approach.

1. Evaluation times are so long that we had to keep population and vocabulary sizes very low. We are likely missing the more significant trends in the evolutionary process due to too low diversity in the population and vocabularies that do not provide statistically significant numbers. Ideally, one would use a population size of several thousand individuals and perhaps some thousand words. This should give statistically significant results about which phonemes are preferred overall, about bigram statistics like the ones computed above, as well as about resulting durations of individual phonemes.

2. Both synthesizer and recognizer use English phonemes and are trained using English speakers. This obviously biases our system. A version without such dependencies would illuminate more general properties of spoken language as such.

Beyond dealing with these problems, there are several promising extensions to the system. Some simply involve further evaluation with more and different noise sources. Others involve adding additional parameters like pitch. One could investigate the lengths of the resulting words by incorporating a penalty related to communication length. Finally, there are more far reaching changes on the horizon. One could eliminate the dependency on a set of phonemes and directly model the human vocal tract, thus allowing for the development of a new set of phonemes. On the other end of the spectrum, one could increase the complexity and purpose of the language game, make the CYKOs refer to objects they see with cameras, and investigate the properties of the evolving languages as opposed to the properties of individual words dealt with here.