Abstract

This is work-in-progress report on a year-long senior project. We examine the value of predictive dependencies in explaining human’s grammar-learning capabilities as measured by Saffran’s research. While predictive dependencies are useful for detecting ungrammatical sentences, they do not seem to have any value in assigning words to categories or learning phrase structure. An information theory approach based on the word of De Marcken looks promising for learning the rules of the grammar but analysis of its effectiveness is not complete. Future work will examine the problem of assigning words to categories and online algorithms for grammar induction.

1 Saffran’s Work

The impetus for my senior project is the work done by Jenny Saffran concerning the use of predictive dependencies as a mechanism for learning artificial grammars. Saffran’s original research in this domain involved creating sentences in an artificial grammar and playing back recordings of these sentences to test subjects[1]. In this experiment, the recording consisted of 50 sentences recorded in uniformly descending prosody to avoid giving any prosodic cues to phrase structure. Subjects were exposed to this test material four times during a 30-min test session on each of two consecutive days. Two different forced choice tests were administered, one to test recognition of ungrammatical sentences (the “grammar test”) and one to test knowledge of phrase boundaries (the “phrase test”). In both tests, subjects were presented with two word sequences from the artificial grammar and asked to select the sequence that was more acceptable. In the grammar test each question presented one grammatical and one ungrammatical sentence, and in the phrase test each question presented one sequence that did not violate phrase boundaries (in English, the dog) and one that did (bit the). The grammar adapted from Morgan and Newport had the following rules:
These rules generate 18 unique sentence types. Although this grammar can generate a sentence of up to seven words long, only sentences of five words or fewer were presented to subjects. Combining this grammar with a dictionary of 16 three and four letter words yielded 1624 possible sentences. A typical sentence presented to subjects would be labeled like so:

The results of the tests administered to subjects show that they are able to reliably select the grammatical sentence in the grammar test, and select the sequence that does not cross phrase boundaries in the boundary test.

2 The Saffran Learner

2.1 Motivation

Saffran suggests that the mechanism that allows subjects to learn the grammar so rapidly is “predictive dependencies.” Saffran describes the predictive dependencies present in her grammar:

This language contains the type of predictive structure found in natural languages. In A phrases, A words can occur without D words, but occurrences of D words perfectly predict the presence of A words; the same relationship obtains between C words and G words. Similarly, C phrases can occur without F words (as optional units at the ends of sentences), but if an F word is present, a C phrase must precede it.[1]

2.2 Learner Operation

2.2.1 Training Algorithm

The Saffran learner first reads a set of grammatical training sentences:

1. A list of all words present in the training data is created.

2. The probability of each word is calculated. For a word $a$:
\[ p(a) = \frac{\text{Occur}(a)}{n} \]

where \( \text{Occur}(a) \) is the number of times \( a \) appears in the training data and \( n \) is the total number of occurrences of all words in the corpus.

3. Predictive probabilities of co-occurrence in a sentence are calculated. If \( a \) and \( b \) are words, then the predictive probability from \( a \) to \( b \) is:

\[ pp(a, b) = \frac{\text{CoOccur}(a, b)}{\text{Occur}(a)} \]

where \( \text{CoOccur}(a, b) \) is the number of instances in which \( a \) and \( b \) co-occur in the same sentence. If \( a \) and \( b \) are the same word, the word must appear twice in the sentence to count as an instance of co-occurrence. Thus \( pp(a, b) \) is 1 if every time \( a \) is present in a sentence \( b \) is also present and 0 if every time \( a \) is present in a sentence \( b \) is not present. The predictive probability is also calculated in or-pairs, that is \( pp(a, b|c) \) is 1 if for every occurrence of \( a \) in a sentence either \( b \) or \( c \) is present in the sentence.

4. Predictive rules are deduced from the predictive probabilities. There are two types of rules:

(a) **Requirement**: \( a \) requires \( b \) if \( pp(a, b) = 1 \). Likewise, \( a \) requires \( b \) or \( c \) if \( pp(a, b|c) = 1 \).

(b) **Exclusion**: \( a \) excludes \( b \) if \( pp(a, b) = 0 \).

### 2.2.2 Evaluation Algorithm

The learner reads and evaluates its testing set:

1. Raw probabilities for each sentence are calculated. The probability of a sentence is the product of the predictive probabilities of each word with every other word. If sentence \( s \) is “abc,” then we have:

\[ p(s) = pp(a, b) \times pp(a, c) \times pp(b, a) \times pp(b, c) \times pp(c, a) \times pp(c, b) \]

2. The probabilities are weighted to alleviate the bias toward shorter sentences. The weighted probability of a sentence \( wp(s) = p(s) \times |s|^2 \) where \( |s| \) is the number of words in \( s \). Although this is a very crude compensation, we have not come up with a better one to date.

3. If the test item is a sentence, the predictive rules are applied to get the final probabilities. For a sentence \( s \), \( fp(s) = wp(s) \) if it does not violate any rules and \( fp(s) = 0 \) if it does. As phrases are not expected to conform to the rules of a sentence, for all test phrases \( s \) \( fp(s) = wp(s) \).

---

1. This is valuable feature because it allows for rules in which the presence of a category excludes another instance of itself in the same sentence. For example, in the Saffran grammar presented earlier, a sentence may only have one \( a \) word.

2. This is, however, similar to the bias of Markov models toward shorter sequences, so the problem is not unique to our algorithm. In MM however, the bias is proportional to the length of the sentence whereas our bias is proportional to the square of the length. Either bias is difficult if not impossible to accurately correct.
2.3 Results

2.3.1 Category Data

The learner was trained with the 18 unique sentence forms produced by Safran’s grammar. We did not cap the sentences at five words as Safran did in her experiments. The learner was tested on the training sentences as well as the category forms of the sentences in Safran’s test set and a set of sentences where each sentence was formed by a single category (i.e. sentences such as “a”). We predicted that the learner would correctly identify grammatical and ungrammatical sentences in Safran’s test set, identify all sentences from the training set grammatical, and identify the one-category sentences as ungrammatical. We had no reason to believe that it would successfully identify whether phrases were complete or crossed phrase boundaries, but if it did it would indicate so via higher phrase probability.

Sentence Tests: The results in the sentence tests were consistent with our expectations. The learner correctly identified the training data as grammatical and the one-category sentences as not, and it correctly responded to all but one of Safran’s sentence tests. The test our learner failed was also the only test in which the experimental group performed significantly worse than chance. This test is described by Safran thus:

Rule 3: If there is an E word, there cannot be a CP.

BIFF KLOR JUX (A - D - E)
BIFF KLOR LUM JUX (A - D - C - E) [1]

where the former sentence is grammatical and the latter is not. Safran’s description of the rule is, however, oversimplified. If the rule were of that form, our learner would have detected the rule. More accurately, this test is based on the restriction that sentences can expand as $s => A PBPCP$ but never $s => A PCPB$. Thus, “adee” is a valid sentence but “adce” is not. Since the learner does not record any positional relationships between words, if the “adee” is acceptable “adce” must be as well. This suggests that recording positional information as a part of the learner’s predictive rules may yield better performance, but given Safran’s experimental results this may actually be a less accurate model of human learning techniques.

Phrase Tests: The results of the phrase tests were also consistent with our expectations. We were not able to find any meaningful correlation between the weighted or unweighted probability of a phrase and whether it crossed phrase boundaries. This is intuitive since this would require knowledge of the actual expansion rules of the grammar. Our learner does not attempt to create these rules because with predictive dependencies alone it is not possible to learn the rules. For example, both of the following hypothetical grammars could produce

\[^{3}p < .01\]
the predictive dependency “e requires a” but in one grammar the phrase “ae” crosses phrase boundaries and in the other it does not:

\[
S \quad \text{S}
\]

\[
\begin{align*}
&\text{AP} & \text{BP} & \text{AP} & \text{BP} \\
&a \quad (e) & b & a \quad (e) & b
\end{align*}
\]

2.3.2 Word Data

We did not expect that the learner would be successful when tested on data consisting of sentences from the grammar. Since the learner is forming rules between particular words and not categories in the sentences, unless it is exposed to every possible sentence in the language it cannot accurately learn the rules of the grammar. This is not different from the learner’s requirements when trained on category data; the learner must always be presented with all possible combinations in order to learn the correct dependencies. Since the learner is not capable of learning which words belong to which categories, it seems that predictive dependencies is worthwhile model if bootstrapped to another mechanism which can detect the categories of words.

2.4 Analysis

The learner’s performance is on par with that of humans in Saffran’s sentence tests if we can assume another mechanism for determining the membership of words to categories. While predictive dependencies are able to serve a major role in detecting ungrammatical sentences, their role in learning phrase boundaries is questionable.

3 Future Work

We have implemented a second learner based on Carl DeMarcken’s work from an Information Theory perspective. The basic approach of this learner is to:

1. Read the training data and conjecture all possible binary, headed, rules that could have generated these sentences. Recursive rules are allowed.

2. Calculate the entropy of each rule based on the methods described by De Marcken[2]. Rules with the lowest entropy are most likely to be actual rules of the grammar.

While the De Marcken learner is now fully implemented, analysis will be done at the start of the next semester. In order to test the learner properly, we must test several definitions of the joint and conditional probabilities that the learner evaluates.

\[4\text{In this small grammar, that is still 4056 sentences!}\]
Future work will likely involve examining “bootsrapping” methods for determining category membership of words and also exploring online algorithms for grammar induction. The goal is to eventually train on corpuses of child-directed speech.

References
