Three Models for the Description of Language

Alaa Kharbouch and Zahi Karam

I. Introduction

The grammar of a language is a device that describes the structure of that language. The grammar is comprised of a set of rules whose goal is twofold: first these rules can be used to create sentences of the associated language and only these, and second they can be used to classify whether a given sentence is an element of the language or not. The goal of a linguist is to discover grammars that are simple and yet are able to fully span the language. In [1] Chomsky describes three possible options of increasing complexity for English grammars: Finite-state, Phrase Structure and Transformational. This paper briefly present these three grammars and summarizes Chomsky's analysis and results which state that finite-state grammars are inadequate because they fail to span all possible sentences of the English language, and phrase structure grammar is overly complex.

A. *Definition of Basic Structures*

Language, as defined in [1] is a finite or infinite set of sentences each of finite length constructed from a finite alphabet, A, of symbols. A string in A is defined as a concatenation of the symbols of A. Therefore, a grammar of a language is some sort of device that produces all the strings that are sentences in that language and only these.

II. Finite State Markov Processes

The first grammar that Chomsky examines is the finite-state grammar, which is defined as one that consists of a finite number of states (S_i) with transition symbols, a_{ij}, and a set C=$\{(S_i, S_j)\}$ of connected states. As this grammar evolves from state to state it produces strings of concatenated symbols a_{ij} that form all the sentences of a finite-state language L_G.

The symbols a_{ij} may be chosen to be phonemes, morphemes or words. Morphemes are defined as the smallest grammatically functioning elements of a language, e.g. ”boy”, ”s” in ”books”. Typically morphemes or words are used as they simplify the grammar, and then each word and morpheme is replaced by its phonemic spelling.

To analyze the types of languages that that finite-state grammars can handle, we must first define a dependency and a dependency set:

Consider the following sentence of the language L formed by the concatenation of the symbols a_i of its alphabet (Note that $\frown$ represents a string concatenation)

$$S = a_1 \frown ... \frown a_n.$$

S is said to have an (i,j)-dependency with respect to L if and only if replacing the symbol a_i with a symbol b_i ($a_i \neq b_i$) in S requires that the symbol a_j ($i < j$) be replaced with b_j so that the sentence S is still an element of the language L.

A dependency set of S in L,

$$D = \{(\alpha_1, \beta_1), ..., (\alpha_m, \beta_m)\},$$

contains dependencies that are distinct in both terms and each first element (α_i) in S precedes all second elements (β_i).

A sentence S with an m-term dependency requires a 2m state finite-state grammar. For a language L to be be a finite-state language we therefore require that there is a finite m such that all valid sentences in L have dependency sets of at most size m. This requirement allows us to decide whether a given language can be described by a finite state grammar or not. The languages L_1, L_2, and L_3 described below are three examples of nonfinite-state languages:

- L_1 consists of sentences formed from the concatenation of m occurrences of a followed by m occurrences of b.

$$L_1: \quad a \frown ..._{mtimes} \frown a \frown b..._{mtimes} \frown b$$

- L_2 consists of sentences where the second half is just the mirror image of the first half.

$$L_2: \quad a \frown b \frown b \frown a \frown a \frown b \frown b \frown a$$

- L_3 consists of sentences whose second half is just a copy of the first half.

$$L_3: \quad a \frown b \frown b \frown a \frown a \frown b \frown b \frown a$$

These are examples of nonfinite-state languages because for any fixed m we can create a valid sentence of that language of length $2m + 2$ that will have $m + 1$ dependencies.

Now that we have a criterion to determine if a given language can be fully represented by a finite-state grammar, we examine whether English satisfies that criterion. Sentences of the English language can contain an infinite number of embedded declarative sentences such as those of the form:

“if ..., then ...” or “either ..., or ...”,

and can have infinitely many subordinate clauses of the form:

“... since ...” or “... which ...”.

Each of the declarative sentences and the subordinate clauses describe a dependency, therefore we cannot bound the number of dependencies in sentences of the English language.

Another finite-state grammar that could be used for the English language is the n^{th} order approximation. This grammar decides the probability of a given word occurring conditioned on the $n-1$ previous words that were observed. This method is attractive since as n increases the sentences created by this grammar start to resemble proper English sentences. However, this n^{th} order approximation to English fails to meet the requirement that the grammar be able to distinguish between grammatically correct and incorrect sentences, this can best be seen by the following example presented by Chomsky:

colorless green ideas sleep furiously
furiously sleep ideas green colorless

In both sentences the likelihood that any of the words follows the other is almost nil. Therefore, this grammar would treat both sentences in the same manner declaring both as ungrammatical, and yet the first sentence is grammatically correct while the second is not. This section leads to the conclusion that English is not a finite-state language.

III. Phrase Structure

In “immediate constituent analysis”, the words in a sentence are initially collected as a part of phrases, and the components of each phrase may again be grouped into its constituent phrases, and this is repeated until the smallest constituents, such as words or morphemes, are reached. This kind of description offers us the advantage of simplification, as each phrase or constituent can be given a label such as “noun phrase (NP)” or “verb phrase (VP)”, and can encapsulate a complex class of expressions. An example of this kind of analysis on the sentence: ”the man took the book” is shown in the diagram in figure 1. We can ask what kind of specification for a grammar would correspond to this view of language or analysis, and this brings us to Phrase-Structure grammars.

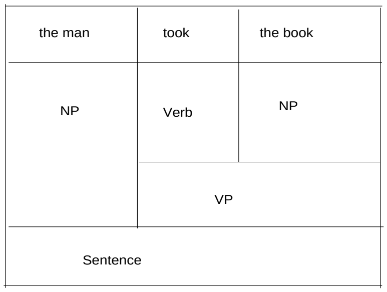

Fig. 1. Constituent analysis example

A phrase-structure grammar consists of a vocabulary or alphabet V_p, a finite set of initial strings $\sum$, and a set of rules F. These rules specify an instruction to replace X_i in V_p with another string Y_i also in the vocabulary, by changing a single symbol of X_i. We denote such a rule here as follows: $X_i \longrightarrow Y_i$. To produce a sentence using this system we begin with an initial string in $\sum$, and repeatedly apply the rules in F where possible. If a certain string is obtained from another by applying a rule in F, then the former is said to *follow from* the latter. A *derivation* is a sequence of strings $D = (S_1, ..., S_t)$ such that each string S_{i+1} follows from the preceding string in the sequence S_i.

A particular phrase-structure grammar is denoted as $[\sum, F]$. We now provide an example of such a system that can produce the sentence in figure 1. Note that $\sharp$ is a marker for the beginning or end of a sentence or string.

$$\begin{array}{rc} \sum: & \sharp \frown \text{Sentence} \frown \sharp \\ F: & \text{Sentence} \rightarrow \text{NP} \frown \text{VP} \\ & \text{VP} \rightarrow \text{Verb} \frown \text{NP} \\ & \text{NP} \rightarrow \text{the} \frown \text{man}, \quad \text{the} \frown \text{book} \\ & \text{Verb} \rightarrow \text{took} \end{array}$$

where the commas separates alternatives for the output of a rule.

We can propose two derivations from this system, using $S_i \Rightarrow S_{i+1}$ to denote that string S_{i+1} follows from S_i.

Derivation D_1:

$$\sharp \frown \text{Sentence} \frown \sharp$$
$$\Rightarrow \sharp \frown \text{NP} \frown \text{VP} \frown \sharp$$
$$\Rightarrow \sharp \frown \text{NP} \frown \text{Verb} \frown \text{NP} \frown \sharp$$
$$\Rightarrow \sharp \frown \text{the} \frown \text{man} \frown \text{Verb} \frown \text{NP} \frown \sharp$$

$$\Rightarrow \sharp \frown \text{the} \frown \text{man} \frown \text{Verb} \frown \text{the} \frown \text{book} \frown \sharp$$
$$\Rightarrow \sharp \frown \text{the} \frown \text{man} \frown \text{took} \frown \text{the} \frown \text{book} \frown \sharp$$

Derivation D_2:

$$\sharp \frown \text{Sentence} \frown \sharp$$
$$\Rightarrow \sharp \frown \text{NP} \frown \text{VP} \frown \sharp$$
$$\Rightarrow \sharp \frown \text{the} \frown \text{man} \frown \text{VP} \frown \sharp$$
$$\Rightarrow \sharp \frown \text{the} \frown \text{man} \frown \text{Verb} \frown \text{NP} \frown \sharp$$
$$\Rightarrow \sharp \frown \text{the} \frown \text{man} \frown \text{took} \frown \text{NP} \frown \sharp$$
$$\Rightarrow \sharp \frown \text{the} \frown \text{man} \frown \text{took} \frown \text{the} \frown \text{book} \frown \sharp$$

We can draw diagrams that corresponds to these derivations, by building a tree that begins with the initial string and appropriately adding child branches to a node when a rule is applied. Both D_1 and D_2 result in the diagram shown in figure 2, which gives the phrase-structure of the sentence arrived at with both derivations. D_1 and D_2 are thus said to be *equivalent*, and they differ only in the order in which the rules are applied to reach the sentence.

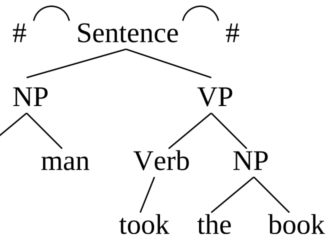

Fig. 2. Phrase structure diagram for derivations D_1 and D_2

However, one can also generate examples where the application of the rules in a different order results in a different tree diagram or phrase structure. Consider the system:

$$\begin{aligned} \sum: &\quad \sharp \frown \text{Sentence} \frown \sharp \\ F: &\quad \text{Sentence} \rightarrow \text{NP} \frown \text{VP} \\ &\quad \text{VP} \rightarrow \text{Verb} \frown \text{NP} \\ &\quad \text{Verb} \rightarrow \text{are} \frown \text{flying} \\ &\quad \text{Verb} \rightarrow \text{are} \\ &\quad \text{NP} \rightarrow \text{they} \\ &\quad \text{NP} \rightarrow \text{planes} \\ &\quad \text{NP} \rightarrow \text{flying} \frown \text{planes} \end{aligned}$$

We can find two derivations D_3 and D_4 that differ in the sequence of rules of F that are applied to but lead to the same sentence: “they are flying planes”. The diagrams that correspond to D_3 and D_4 are shown in figure 3 (on the left and right, respectively), and so this sentence can have two different phrase structures out of this grammar. Such a case, where different phrase structures are obtained for the same sentence derived using the grammar, is referred to as a *constructional homonymity*. However, the sentence itself has an inherent ambiguity. The word ”they”, for example, can be taken to mean ”those specks on the horizon” and the sentence could be taken to mean ”those specks on the horizon - are - flying planes”. Alternatively, “they” could refer to “those pilots” and one can understand the sentence as “those pilots - are flying - planes”. This kind of ambiguity to a native speaker of English in a sentence such as this is called a semantic ambiguity. Chomsky suggests that one way of assessing the quality or appropriateness of a grammar is by observing whether cases of constructional homonymity are results of semantic ambiguities.

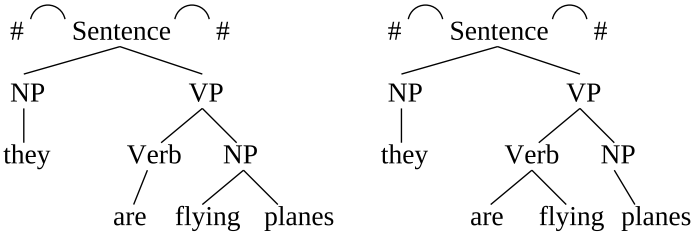

Fig. 3. Phrase structure diagram for derivations D_3 and D_4

The examples of phrase-structure grammars mentioned so far are cases of Context-free grammars(CFG). In a context-free grammar, the rules in F are constrained to have only a single symbol on the left hand side. More specifically, a context-free grammar cannot have a rule of the form:

$$Z \frown X \frown W \rightarrow Z \frown Y \frown W$$

where X,Y,Z and W are symbols in the vocabulary. This rule specifies that X can be replaced with Y only if it is preceded by Z and followed by W, so in this case Z and W act as the *context*.

Note that with a context-free phrase structure grammar we can capture all the sentences in the set of language L_1 (The language containing all sentences composed of n a’s followed by the same number of b’s, for all possible integer $n \geq 0$). This was not possible with a finite-state markov process grammar. We denote this set of strings here as (a^n, b^n). A system that can generate L_1 sentences is:

$$\begin{aligned} \sum: &\quad \sharp \frown \text{S} \frown \sharp \\ F: &\quad \text{S} \rightarrow \text{a} \frown \text{S} \frown \text{b} \\ &\quad \text{S} \rightarrow \text{a} \frown \text{b} \end{aligned}$$

To generate a string with n a's and b's, the first rule is applied $n-1$ times, followed by the second rule carried out once.

However, we cannot find a CFG to correspond to a language (a^n, b^n, c^n), i.e. the language consisting of all sentences composed of n a's followed by the same number of b's and then n c's. A phrase structure grammar that can generate sentences for this language is:

$$\begin{aligned}\sum : &\quad \sharp \frown \mathrm{S} \frown \sharp \\ F : &\quad \mathrm{S} \rightarrow \mathrm{a} \frown \mathrm{S} \frown \mathrm{B} \frown \mathrm{c}, \quad \mathrm{a} \frown \mathrm{b} \frown \mathrm{c} \\ &\quad \mathrm{c} \frown \mathrm{B} \rightarrow \mathrm{B} \frown \mathrm{c} \\ &\quad \mathrm{B} \frown \mathrm{b} \rightarrow \mathrm{b} \frown \mathrm{b}\end{aligned}$$

The context-sensitive rules in this case are required for this language. For example, if we begin by applying the rule
$\mathrm{S} \rightarrow \mathrm{a} \frown \mathrm{S} \frown \mathrm{B} \frown \mathrm{c}$ then
$\mathrm{S} \rightarrow \mathrm{a} \frown \mathrm{b} \frown \mathrm{c}$ followed by a single application of the second and then the third rule, we obtain the string "aabbcc" or the $n = 2$ case.

Chomsky argues that the phrase-structure framework is more powerful (in addition to being simpler) than finite-state Markov descriptions. It can be shown that any finite state grammar can be written as a phrase-structure grammar. This can be accomplished by simply including rules corresponding to each connection or transition between two states in the finite-state model, where for the most part a symbol representing the previous state is replaced with a concatenation of a transition string from the finite-state model with the symbol representing the new state. In addition, phrase structure grammars are generally more abstract, as they include symbols that are not part of the sentences of the language of interest as part of the description of the grammar. An example of this was the use of "NP" and "VP" which are not part of the english vocabulary but played a significant role in the analysis or description. The paper concluded that English is beyond the reach of finite-state markov grammars because of the dependency issue mentioned earlier and the "mirror-image" properties shared with the languages L_1 and L_2. However, both L_1 and L_2 can be described by phrase-structure grammar, so this consideration cannot be used to eliminate phrase-structure grammar as far as English is concerned. Chomsky does not go as far as to claim with certainty that English is a language that can result from phrase-structure grammar, but phrase-structure at this point escapes any disqualifying criteria proposed so far.

IV. Transformational Grammar

In the examples shown so far only one option was included for replacing the element "Verb". However, even with the verb stem fixed (e.g. "take") there are many alternative forms to the verb that can be used in place of "took" in the sentence "the man took the book", including "takes", "has taken", "is taking", "will be taking", "has been taking", etc. A direct approach describing the elements of these strings would be complicated, due to the strong dependencies between them (examples are "has taken" but not "has taking", and "is being taken" but not "is being taking"). One approach to simplify the analysis of the "Verb" component in terms of its smaller elements is by considering so-called discontinuous elements. For example if a target sentence contains the phrase "has been taking", we isolate elements such "has..en" , "be..ing" and "take". Chomsky proceeds with a long and elaborate example, whose main points we will try to summarize here. The idea is to contruct the sentence "the man has been taking the book" by including rules that allow us to introduce and rearrange symbols so that "be" and "en" combine to form "been", and "take" and "ing" eventually combine to form "taking". Included also in the grammar is a symbol that indicates the tense, which later modifies the verb stem through a rule. Examples of these and others rules included in this exercise are:

$$\begin{aligned}\mathrm{have} \frown \mathrm{past} &\rightarrow \mathrm{had} \\ \mathrm{will} \frown \mathrm{past} &\rightarrow \mathrm{would} \\ \mathrm{take} \frown \mathrm{ing} &\rightarrow \mathrm{taking} \\ \mathrm{be} \frown \mathrm{en} &\rightarrow \mathrm{been}\end{aligned}$$

A re-ordering rule is proposed and used in the example to arrive at the sentence:

$$\mathrm{Af} \frown \mathrm{v} \rightarrow \mathrm{v} \frown \mathrm{Af}$$

Where Af is the class of verbs that contain the affixes such as "en" and "ing" and the tense symbols "past", "present" etc, and v is the verb stems such as "take", "eat", "will", "can", etc. However, this rule assumes that the classification of certain symbols, expressions, or strings is known. The derivation process is required to know, for example that "take" belongs to the v class or was originally a v symbol earlier in the derivation. There are no provisions for the use of such rules in the original definition of phrase-structure grammars, as they require the knowledge of not only the current string in the derivation, but the "history of derivation" as well. Chomsky suggests that such extensions to the framework or grammar definitions that allow for steps similar to the ones outlined in this example may be useful. The paper then points to complications with the use of the string "$be \frown en$" for passives. This element is associated with many conditions and restrictions in its use. For example, one can have "the man would have been eating the food" but not "the man is eaten the food" (here "is" comes from "be", and "eat" combines with "en". In the context "the food —- by the man" one is required to use "$be \frown en$". Passives may be generated differently, by simply transforming other sentences into the passive form with rules that look like:

$$\text{NP}_1 \frown \text{Auxiliary} \frown \text{V} \frown \text{NP}_2 \rightarrow$$
$$\text{NP}_2 \frown \text{Auxiliary} \frown \text{be} \frown \text{en} \frown \text{V} \frown \text{by} \frown \text{NP}_1$$

The ideas discussed in this section suggesting added mechanisms beyond the original phrase-structure specifications and this approximately summarized sentence transformation are incorporated in a larger framework of *grammatical transformations*, whose rigorous specifications will not be included in this report. Grammatical transformations use the concept of *phrase markers* which include all information about the phrase structure of a sentence. A phrase-marker, which is common to a set of equivalent derivations, essentially contains enough "derivation history" information to construct a tree diagram similar to those in figures 2 and 3

V. Conclusion

In the previous sections we presented and analyzed three different possible grammars for the English language. Of the three only the transformational phrase structure yielded a desirable grammar which was both simple and encompassed all the sentences of the English language and only these. Another criterion that is of interest when deciding whether a given grammar is adequate is whether it will provide nonequivalent derivations for ambiguous sentences, such as:

"they are flying planes" and "the raising of flowers"

Finite state grammar cannot provide alternate derivations for either of these example sentences. Phrase structure provides nonequivalent derivations for only the first sentence. Again, the transformational grammar properly represented the ambiguity in these two examples through nonequivalent derivations, and therefore satisfied this criterion as well.

The desired transformational phrase structure as presented by Chomsky has a three level structure: the first level corresponds to applying a sequence of phrase structure rules to an initial string to produce a terminal string. The second level corresponds to applying transformational rules that would result in the desired string of morphemes in the correct order. Finally the third level converts the morpheme strings into their equivalent phonemic representation.

Of the transformational rules that are applied in the second level some are obligatory and others are optional. Optional in the sense that if these transformations are not applied we would still end up with a grammatically acceptable sentence. Chomsky uses this distinction to define a kernel of basic sentences which are obtained from applying only obligatory transformations to the terminal string obtained from applying the phrase structure rules in the first level of the grammar. The kernel can be limited to a small set of simple, active and declarative sentences, for example:

"the man ate the food".

then through the optional transformations one can derive the desired phrase, for example if the desired was a question:

"was the food eaten by the man".

Therefore, even though the English language is not actually made up of kernel sentences, we can obtain any desired sentence by complex transformations on kernel sentences that are to a large extent meaning-preserving. One way to understand this is that the kernel sentences contain most of the basic information content of the sentences that are generated from them via a set of transformations.

A. *Links to Information Theory*

Shannon had estimated [2] that the redundancy in English text is about 50%. This means that it is possible to compress English text to approximately half its size with a very low probability of loosing information. The grammar he used to represent the English language was a finite-state n^{th}-order approximation. Although this finite-state model lends itself well to mathematical analysis using the conventional information theoretic tools, we had concluded on the basis of our review of Chomsky's work that it does not properly capture the English language: it is not able to properly distinguish between the different meanings of an ambiguous sentence, and it excludes a large number of grammatically correct sentences, that are comprised of highly improbable sequences of words while still including a vast number of ungrammatical sentences. An important question that seems to arise is whether our estimate of the redundancy would increase if we used the phrase structure transformational grammar that Chomsky claims best represents the English language.

We can obtain some insight into this by taking a view that the information present in a specific representation is linked to the amount of information it takes to convey or transmit a sentence. The derivations of sentences with phrase-structure grammars is reminiscent of a decompression or decoding algorithm, as rules are applied, sometimes expanding the string until an English sentence is reached. One can imagine a coding scheme that transmits the initial string followed by the sequence of rules that are applied to specify a derivation. The receiver only needs to know which rule from a finite set is used to obtain the next string in the derivation. Furthermore, some rules at certain points in the derivation can be eliminated, as both the sender and receiver know that a string on the left hand side of a rule does not appear in the current string and is therefore not relevant at that point. An index integer can be sent to specify a certain rule from the set of possible ones.

The goal of the English language is to allow communication between people. Therefore our goal in compression is not to convey the exact the person said but what information he was trying to convey. One can therefore imagine a lossy compression algorithm that transmits the sentence from the kernel of basic sentences which contains the desired information, perhaps by restricting the information source to a less complex grammar or a shorter vocabulary. A coding scheme of this type could also use AEP to encode only the set of most likely sentences from the kernel.

We have suggested an outline for a lossy and lossless compression schemes motivated by the phrase structure transformational grammars. However, one must remember that this encoded data needs to be transmitted over a lossy channel, either print or speech. Therefore if we were able to completely remove the redundancy inherent in an English sentence, our encoding would be very sensitive to the errors in the channel. Specifically in our suggested lossy encoding, if we are not able to correctly decode, at the receiver, as little as one morpheme we completely loose the meaning of the transmitted sentence.

Another insight involves the use of English with a limited phrase-structure grammar or another language that can be described by the models in this review as a universal coding scheme for information and media that is not necessarily conversational. For example, a blank or white image can be encoded or transmitted by simply listing all the identical pixel values, but it could instead be described by a word or string such as “white”. Another example is an image that consists of a circle and a line, where instead of transmitting all pixel values, we send information that specifies a recipe for constructing the image, using syntax similar to that of a programming language (which are not beyond the scope of the paper or the grammar models). For example:

```
\put{line}{0,5,20,12};
\put{circle}{4,5,3};
```

VI. Bibliography

1 Chomsky, N., Three Models for the Description of Language, IEEE Transactions on Information Theory, 1956.

2 Shannon, C., Prediction and Entropy of Printed English, Bell System Technical Journal, 1951.