**Grammar based language models**

Due to the smoothing  techniques, bigram  and trigram  language models are robust   and have been successfully used more widely in speech recognition  than conventional grammars like context free or even context sensitive grammars.   Although these grammars are expected to better capture the inherent structures of the language, they have a couple of problems:

* *robustness*:   the grammar must be able to handle a vocabulary  of 10000 or more words, and ultimately a non-zero probability must be assigned to each possible word sequence.
* *ambiguity*: while *m*-gram language models   avoid any ambiguity in parsing,   context free grammars  are typically ambiguous and thus produce more than a single parse tree .

When using context free grammars , the terminals or symbols that can be observed are the words of the vocabulary  [[Lafferty et al. (1992)](https://wwwhomes.uni-bielefeld.de/gibbon/Handbooks/gibbon_handbook_1997/node444.html#Lafferty92), [Wright et al. (1993)](https://wwwhomes.uni-bielefeld.de/gibbon/Handbooks/gibbon_handbook_1997/node444.html#Wright93), [Della Pietra et al. (1994)](https://wwwhomes.uni-bielefeld.de/gibbon/Handbooks/gibbon_handbook_1997/node444.html#DellaPietra94), [Yamron (1994)](https://wwwhomes.uni-bielefeld.de/gibbon/Handbooks/gibbon_handbook_1997/node444.html" \l "Yamron94)]. We use the notation:

* *A*,*B*,*C*,..: non-terminals or abstract syntactic categories;
* *u*,*v*,*w*,..: terminals, i.e. the spoken words tex2html_wrap_inline45625;
* tex2html_wrap_inline45995: an arbitrary nonempty string of terminals and non-terminals.

We typically attach probabilities to the context free rules. For a production rule like  
eqnarray9727  
we have a conditional probability tex2html_wrap_inline45997 that a given non-terminal *A* will be rewritten as tex2html_wrap_inline45995. As usual, these probabilities must be normalised:  
eqnarray9729  
As usual in the framework of context free grammars , one often uses so-called normal forms like Chomsky and Greibach normal form for which standard introductions to formal languages in compiler construction theory can be consulted. Typically, these normal forms simplify the analysis of the problem under consideration, and there are automatic techniques for converting arbitrary context free grammars  into these normal forms.

For each parse tree  of a given string, we can compute its probability as the product over all its production rules. The probability of the given word string is then obtained by summing these probabilities over all possible parse trees.  The fundamental definitions for context free grammars   are well covered in textbooks [[Fu (1982)](https://wwwhomes.uni-bielefeld.de/gibbon/Handbooks/gibbon_handbook_1997/node444.html#Fu82), [Gonzalez & Thomason (1978)](https://wwwhomes.uni-bielefeld.de/gibbon/Handbooks/gibbon_handbook_1997/node444.html#Gonzalez78)].

For the application of stochastic context free grammars,       stochastic grammar we consider four questions or problems:

* *parsing problem*:     How to find the best parse tree along with its probability? How to efficiently calculate the total probability of all parse trees for a given sentence? The solution is given by the stochastic extension of the CYK algorithm [[Fu (1982)](https://wwwhomes.uni-bielefeld.de/gibbon/Handbooks/gibbon_handbook_1997/node444.html#Fu82), [Gonzalez & Thomason (1978)](https://wwwhomes.uni-bielefeld.de/gibbon/Handbooks/gibbon_handbook_1997/node444.html#Gonzalez78)], which is sometimes referred to as *inside algorithm* [[Jelinek et al. (1992)](https://wwwhomes.uni-bielefeld.de/gibbon/Handbooks/gibbon_handbook_1997/node444.html" \l "Jelinek92)].
* *predicting the next word*: How can we use a grammar to predict the next word, i.e. given the word sequence tex2html_wrap_inline46003, how can we compute the conditional probability for the word tex2html_wrap_inline45555 in position *n*:  
  eqnarray9751  
  The solution is provided by the so-called *left corner algorithm* by Jelinek and Lafferty [[Jelinek et al. (1992)](https://wwwhomes.uni-bielefeld.de/gibbon/Handbooks/gibbon_handbook_1997/node444.html" \l "Jelinek92)].
* *learning the rule probabilities*: Given the rules of a grammar, how can we estimate the rule probabilities from training sentences ? The solution is provided by the corresponding version of the EM algorithm , which is called the *inside-outside algorithm* [[Jelinek et al. (1992)](https://wwwhomes.uni-bielefeld.de/gibbon/Handbooks/gibbon_handbook_1997/node444.html" \l "Jelinek92)].
* *grammatical inference*: How can we learn the grammar as a whole from a set of training data  [[Fu (1982)](https://wwwhomes.uni-bielefeld.de/gibbon/Handbooks/gibbon_handbook_1997/node444.html#Fu82), [Gonzalez & Thomason (1978)](https://wwwhomes.uni-bielefeld.de/gibbon/Handbooks/gibbon_handbook_1997/node444.html#Gonzalez78)]? It is this problem for which no good solutions exist yet.

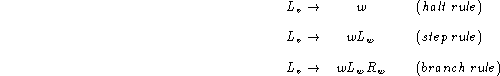
The main stumbling block in using grammar based language models so far seems to be that the grammar rules as such are not available, either because handwritten grammars are simply not good enough or because the problem of grammatical inference is too big. To mitigate these problems, there have been attempts to use special types of context free grammars  which are referred to as lexicalised or link     grammars. Their non-terminals are assumed to depend on the words of the lexicon. These grammars include the bigram  and trigram  language models as special cases and thus provide a smooth transition from the *m*-gram language models   to context free grammars  [[Lafferty et al. (1992)](https://wwwhomes.uni-bielefeld.de/gibbon/Handbooks/gibbon_handbook_1997/node444.html#Lafferty92), [Della Pietra et al. (1994)](https://wwwhomes.uni-bielefeld.de/gibbon/Handbooks/gibbon_handbook_1997/node444.html#DellaPietra94), [Yamron (1994)](https://wwwhomes.uni-bielefeld.de/gibbon/Handbooks/gibbon_handbook_1997/node444.html" \l "Yamron94)]. Although so far there are only preliminary results, we will consider these approaches in more detail because they offer promising extensions from *m*-gram models   and they provide a good framework to show the relationship between *m*-gram language models,   finite state grammars  and context free grammars.

To describe this type of grammar, we will use a special version of the Greibach normal form. Each context free grammar  can be converted into a grammar whose rules are of the following three types:  
  
The third rule type describes a branching process, which is required for the full power of context free grammars.   Without it, we are restricted to regular grammars which can be conveniently represented by finite state networks.   Each network state is identified with a non-terminal of the grammar.

To obtain a bigram  language model in this framework, we use a separate non-terminal tex2html_wrap_inline46021 for each word *w*. The bigram  model itself is expressed by the following rule type for the word pair (*vw*):  
eqnarray9784  
The bigram  rules can be represented as a finite state network  as illustrated in Figure [7.5](https://wwwhomes.uni-bielefeld.de/gibbon/Handbooks/gibbon_handbook_1997/node228.html#BigramNet) for a given word sequence. Unlike the conventional representation, the nodes  stand for the observed words, i.e. the terminals, whereas the links represent the non-terminals. For a complete description of the grammar, we have to include the rules for sentence begin and end. For a trigram  model, we have to use a separate non-terminal for each word pair (*vw*), and the rules depend on the word triples (*uvw*):  
eqnarray9791  
The trigram  rules are illustrated in Figure [7.6](https://wwwhomes.uni-bielefeld.de/gibbon/Handbooks/gibbon_handbook_1997/node228.html#TrigramNet).

 figure9797  
**Figure 7.5:** Bigram as a finite state grammar

 figure9805  
**Figure 7.6:** Trigram as a finite state grammar

To go beyond the power of *m*-gram  and finite state language models , we need more than one non-terminal on the right-hand side. To simplify the presentation, we restrict ourselves to the special case of pairwise word dependencies, which can be viewed as an extended bigram  model. In addition to the standard bigram  non-terminals now denoted by tex2html_wrap_inline46035, we have additional non-terminals for the branching process, which are denoted by tex2html_wrap_inline46037 for word *w*:  
  
The production rule for non-terminal tex2html_wrap_inline46037 is different from the rules for tex2html_wrap_inline46035 and, for example, could have this form:  
eqnarray9823  
This type of rule is capable of handling long range dependencies[gif](https://wwwhomes.uni-bielefeld.de/gibbon/Handbooks/gibbon_handbook_1997/footnode.html#10705) sich as those in the following English examples [[Della Pietra et al. (1994)](https://wwwhomes.uni-bielefeld.de/gibbon/Handbooks/gibbon_handbook_1997/node444.html#DellaPietra94)]:

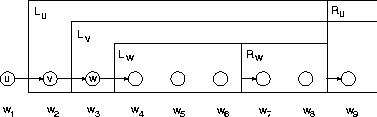
*between ...and ...*

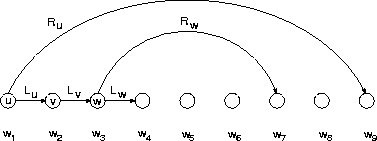
*neither ...nor ...*

*to describe ...as ...*

*to prevent ...from ...*

A conventional bigram  or trigram  language model cannot capture these dependencies. In general, for a given word string, context free grammars  define a hierarchical structure by the parsing  process. Figure [7.7](https://wwwhomes.uni-bielefeld.de/gibbon/Handbooks/gibbon_handbook_1997/node228.html#ParseTree) shows the parsing structure in the conventional form. There is another equivalent representation shown in Figure [7.8](https://wwwhomes.uni-bielefeld.de/gibbon/Handbooks/gibbon_handbook_1997/node228.html#ParseLink), where the long range dependencies are expressed by special *links* in addition to the short range (i.e. next neighbour) bigram  links. In this link representation, the three different types of rule can be seen as follows. The *step* rule stands for the standard bigram  link to the next right word. The *branch* rule has two links: in addition to the bigram  link, there is a long range link to another word, e.g. the link from word tex2html_wrap_inline46045 to tex2html_wrap_inline46047 in Figure [7.8](https://wwwhomes.uni-bielefeld.de/gibbon/Handbooks/gibbon_handbook_1997/node228.html#ParseLink). Since there is the long range link for word tex2html_wrap_inline46047, there is no bigram  link for tex2html_wrap_inline46047, and therefore the *halt* rule must be used for word tex2html_wrap_inline46053, the predecessor of tex2html_wrap_inline46047. For this type of grammar based language model, there are only some preliminary experimental results [[Della Pietra et al. (1994)](https://wwwhomes.uni-bielefeld.de/gibbon/Handbooks/gibbon_handbook_1997/node444.html#DellaPietra94), [Yamron (1994)](https://wwwhomes.uni-bielefeld.de/gibbon/Handbooks/gibbon_handbook_1997/node444.html" \l "Yamron94)]. A still existing problem is the question of how to select the relevant word links in view of the large number of possible spurious links. In [[Della Pietra et al. (1994)](https://wwwhomes.uni-bielefeld.de/gibbon/Handbooks/gibbon_handbook_1997/node444.html#DellaPietra94)], a method has been suggested for inferring the links automatically from a text corpus.

   
**Figure 7.7:** Parsing in the conventional form

   
**Figure 7.8:** Parsing in link grammar form