UNIT – IV

SYNTAX

The study of the structure of phrases and sentences. sentences and phrases syntax is the study of sentence structure**.** Sentences are composed not directly out of words but of constituents which may consist of more than one word, called phrases. A phrase is an expression which is a constituent in a sentence and is the expansion of a head (i.e. key word). For instance, the constituent the king in (1), or the constituents my brother and an expensive car in (2) are Noun Phrases, abbreviated as NPs, because their key elements are the nouns (Ns) king, brother and car, respectively.1 It can happen that a phrase is realised by a single word, for example the NPs John, Mary and apples in (3) consist of the Ns John, Mary and apples, and nothing else. In (4) he is a special NP because its head is a pronoun rather than a noun. (1) The king laughed. (2) My brother bought an expensive car. (3) John gave Mary apples. (4) He went home. (1)-(4) are sentences.

The terms sentence and clause can be used synonymously. A sentence or clause is an expression which minimally contains a subject and a predicate, and which may also contain other types of elements, viz. complements and adjuncts. For instance, (1) consists of just a subject and a predicate. The NP the king is the subject, and the Verb Phrase (VP), composed of a single verb (V) laughed, is the predicate. A complement is a constituent whose presence is structurally "dictated" (required or licensed) by a particular word. The presence of the complement "follows" from the presence of the word which it is a complement of. For instance, in (2) above the NP my brother is the subject, the V bought is the predicate, and the NP an expensive car is a complement, more particularly a direct object, of the verb bought. An object is a particular kind of complement. In (3) above the subject is the NP John, the predicate is the V gave, and there are two complements, the NP Mary, functioning as an indirect object, and the NP apples functioning as a direct object. In (4) the complement of the V went is the Adverb Phrase (AdvP) home, consisting of the single adverb (Adv) home. The subject and the complement(s) together are said to be the arguments of the predicate. Arguments are the participants (entities) that are necessarily involved in the situation identified by the predicate. For example, in (2) the predicate bought has two arguments: the subject (somebody did the buying), and the object (something was bought). In English, subjects typically occur in the nominative case (I, he, etc.), whereas objects occur in the

accusative case (me, him, etc.), but observable case-marking is restricted to pronouns. Another difference between subjects and complements is that, in English, verbs agree with their subjects in person and number but do not agree with their complements. Also, subjects in English typically precede verbs, while complements follow them. In addition to the subject, verb and complement(s), the sentence or clause may also contain constituents which are not structurally required by the verb but add optional information about place, time, manner, purpose, etc. Such constituents are called adjuncts. Some of these function as adverbials, e.g. the Prepositional Phrase (PP) on Tuesday in (5) is a time adverbial, the Adverb Phrase (AdvP) very quickly in (6) is a manner adverbial. Some of the adjuncts function as attributes within noun phrases, e.g. the Adjective Phrase (AP), realised by a single Adjective (A) expensive in (5), is an attribute of car. (5) My brother bought an expensive car on Tuesday. (6) He went home very quickly. The terms subject, predicate, object (direct and indirect), adverbial, attribute; complement and adjunct refer to grammatical functions which constituents may perform in the sentence, whereas terms such as NP, VP, AP, AdvP, PP, N, V, A, Adv, P, etc. refer to syntactic categories, i.e. they name the grammatical category to which the constituent belongs. These two sets of terms are fairly independent of each other, e.g. an NP can function as subject, or as object, or as the complement of a preposition, or even as adverbial (e.g. the NP last year). Similarly, the function of adverbial can be performed by an AdvP (very quickly), a PP (on Tuesday), an NP (last year) or even by an embedded clause (e.g. when I was writing a letter).

GRAMMAR

Before considering how grammatical structure can be represented, analyzed and used, we should ask what basis we might have for considering a particular grammar “correct”, or a particular sentence “grammatical,” in the first place. Of course, these are primarily questions for linguistics proper, but the answers we give certainly have consequences for computational linguistics. Traditionally, formal grammars have been designed to capture linguists' intuitions about well-formedness as concisely as possible, in a way that also allows generalizations about a particular language (e.g., subject-auxiliary inversion in English questions) and across languages (e.g., a consistent ordering of nominal subject, verb, and nominal object for declarative, pragmatically neutral main clauses). Concerning linguists' specific well-formedness judgments, it is worth noting that these are largely in agreement not only with

each other, but also with judgments of non-linguists—at least for “clearly grammatical” and “clearly ungrammatical” sentences (Pinker 2007). Also the discovery that conventional phrase structure supports elegant compositional theories of meaning lends credence to the traditional theoretical methodology.

However, traditional formal grammars have generally not covered any one language comprehensively, and have drawn sharp boundaries between well-formedness and ill-formedness, when in fact people's (including linguists') grammaticality judgments for many sentences are uncertain or equivocal. Moreover, when we seek to process sentences “in the wild”, we would like to accommodate regional, genre-specific, and register-dependent variations in language, dialects, and erroneous and sloppy language (e.g., misspellings, unpunctuated run-on sentences, hesitations and repairs in speech, faulty constituent orderings produced by non-native speakers, and fossilized errors by native speakers, such as “for you and I”—possibly a product of schoolteachers inveighing against “you and me” in subject position). Consequently linguists' idealized grammars need to be made variation-tolerant in most practical applications.

The way this need has typically been met is by admitting a far greater number of phrase structure rules than linguistic parsimony would sanction—say, 10,000 or more rules instead of a few hundred. These rules are not directly supplied by linguists (computational or otherwise), but rather can be “read off” corpora of written or spoken language that have been decorated by trained annotators (such as linguistics graduate students) with their basic phrasal tree structure. Unsupervised grammar acquisition (often starting with POS-tagged training corpora) is another avenue (see section 9), but results are apt to be less satisfactory. In conjunction with statistical training and parsing techniques, this loosening of grammar leads to a rather different conception of what comprises a grammatically flawed sentence: It is not necessarily one rejected by the grammar, but one whose analysis requires some rarely used rules.

As mentioned in section 1.2, the representations of grammars used in computational linguistics have varied from procedural ones to ones developed in formal linguistics, and systematic, tractably parsable variants developed by computationally oriented linguists. Winograd's shrdlu program, for example, contained code in his programmar language

expressing, To parse a sentence, try parsing a noun phrase (NP); if this fails, return NIL, otherwise try parsing a verb phrase (VP) next and if this fails, or succeeds with words remaining, return NIL, otherwise return success.

Similarly Woods' grammar for lunar was based on a certain kind of procedurally interpreted transition graph (an augmented transition network, or ATN), where the sentence subgraph might contain an edge labeled NP (analyze an NP using the NP subgraph) followed by an edge labeled VP (analogously interpreted). In both cases, local feature values (e.g., the number and person of a NP and VP) are registered, and checked for agreement as a condition for success. A closely related formalism is that of definite clause grammars (e.g., Pereira & Warren 1982), which employ Prolog to assert "facts" such as that if the input word sequence contains an NP reaching from index I1 to index I2 and a VP reaching from index I2 to index I3, then the input contains a sentence reaching from index I1 to index I3. (Again, feature agreement constraints can be incorporated into such assertions as well.) Given the goal of proving the presence of a sentence, the goal-chaining mechanism of Prolog then provides a procedural interpretation of these assertions.

At present the most commonly employed declarative representations of grammatical structure are context-free grammars (CFGs) as defined by Noam Chomsky (1956, 1957), because of their simplicity and efficient parsability. Chomsky had argued that only deep linguistic representations are context-free, while surface form is generated by transformations (for example, in English passivization and in question formation) that result in a non-context-free language. However, it was later shown that on the one hand, unrestricted Chomskian transformational grammars allowed for computationally intractable and even undecidable languages, and on the other, that the phenomena regarded by Chomsky as calling for a transformational analysis could be handled within a context-free framework by use of suitable features in the specification of syntactic categories. Notably, unbounded movement, such as the apparent movement of the final verb object to the front of the sentence in "Which car did Jack urge you to buy?", was shown to be analyzable in terms of a gap (or slash) feature of type /NP[wh] that is carried by each of the two embedded VPs, providing a pathway for matching the category of the fronted object to the category of the vacated object position. Within non-transformational grammar frameworks, one therefore speaks of unbounded (or long-distance) dependencies instead of unbounded movement. At

the same time it should be noted that at least some natural languages have been shown to be mildly context-sensitive (e.g., Dutch and Swiss German exhibit cross-serial dependencies where a series of nominals "NP1 NP2 NP3 …" need to be matched, in the same order, with a subsequent series of verbs, "V1 V2 V3 …"). Grammatical frameworks that seem to allow for approximately the right degree of mild context sensitivity include Head Grammar, Tree-Adjoining Grammar (TAG), Combinatory Categorial Grammar (CCG), and Linear Indexed Grammar (LIG). Head grammars allow insertion of a complement between the head of a phrase (e.g., the initial verb of a VP, the final noun of a NP, or the VP of a sentence) and an already present complement; they were a historical predecessor of Head-Driven Phrase Structure Grammar (HPSG), a type of unification grammar (see below) that has received much attention in computational linguistics.

PARSING

Natural language analysis in the early days of AI tended to rely on template matching, for example, matching templates such as (X has Y) or (how many Y are there on X) to the input to be analyzed. This of course depended on having a very restricted discourse and task domain. By the late 1960s and early 70s, quite sophisticated recursive parsing techniques were being employed. For example, Woods' lunar system used a top-down recursive parsing strategy interpreting an ATN in the manner roughly indicated in section 2.2 (though ATNs in principle allow other parsing styles). It also saved recognized constituents in a table, much like the class of parsers we are about to describe. Later parsers were influenced by the efficient and conceptually elegant CFG parsers described by Jay Earley (1970) and (separately) by John Cocke, Tadao Kasami, and Daniel Younger (e.g., Younger 1967). The latter algorithm, termed the CYK or CKY algorithm for the three separate authors, was particularly simple, using a bottom-up dynamic programming approach to first identify and tabulate the possible types (nonterminal labels) of sentence segments of length 1 (i.e., words), then the possible types of sentence segments of length 2, and so on, always building on the previously discovered segment types to recognize longer phrases. This process runs in cubic time in the length of the sentence, and a parse tree can be constructed from the tabulated constituents in quadratic time. The CYK algorithm assumes a Chomsky Normal Form (CNF) grammar, allowing only productions of form Np → Nq Nr, or Np → w, i.e., generation of

two nonterminals or a word from any given nonterminal. This is only a superficial limitation, because arbitrary CF grammars are easily converted to CNF.

The method most frequently employed nowadays in fully analyzing sentential structure is *chart parsing*. This is a conceptually simple and efficient dynamic programming method closely related to the algorithms just mentioned; i.e., it begins by assigning possible analyses to the smallest constituents and then inferring larger constituents based on these, until an instance of the top-level category (usually S) is found that spans the given text or text segment. There are many variants, depending on whether only complete constituents are posited or incomplete ones as well (to be progressively extended), and whether we proceed left-to-right through the word stream or in some other order (e.g., some seemingly best-first order). A common variant is a *left-corner* chart parser, in which partial constituents are posited whenever their "left corner"—i.e., leftmost constituent on the right-hand side of a rule—is already in place. Newly completed constituents are placed on an *agenda*, and items are successively taken off the agenda and used if possible as left corners of new, higher-level constituents, and to extend partially completed constituents. At the same time, completed constituents (or rather, categories) are placed in a chart, which can be thought of as a triangular table of width *n* and height *n* (the number of words processed), where the cell at indices (*i, j*), with j > i, contains the categories of all complete constituents so far verified reaching from position *i* to position *j* in the input. The chart is used both to avoid duplication of constituents already built, and ultimately to reconstruct one or more global structural analyses. (If all possible chart entries are built, the final chart will allow reconstruction of all possible parses.) Chart-parsing methods carry over to PCFGs essentially without change, still running within a cubic time bound in terms of sentence length. An extra task is maintaining probabilities of completed chart entries (and perhaps bounds on probabilities of incomplete entries, for pruning purposes).

Because of their greater expressiveness, TAGs and CCGs are harder to parse in the worst case (O(n6)) than CFGs and projective DGs (O(n3)), at least with current algorithms (see Vijay-Shankar & Weir 1994 for parsing algorithms for TAG, CCG, and LIG based on bottom-up dynamic programming). However, it does not follow that TAG parsing or CCG parsing is impractical for real grammars and real language, and in fact parsers exist for both that are competitive with more common CFG-based parsers.

Finally we mention *connectionist* models of parsing, which perform syntactic analysis using layered (artificial) neural nets (ANNs, NNs) (see Palmer-Brown et al. 2002; Mayberry and Miikkulainen 2008; and Bengio 2008 for surveys). There is typically a layer of input units (nodes), one or more layers of hidden units, and an output layer, where each layer has (excitatory and inhibitory) connections forward to the next layer, typically conveying evidence for higher-level constituents to that layer. There may also be connections within a hidden layer, implementing cooperation or competition among alternatives. A linguistic entity such as a phoneme, word, or phrase of a particular type may be represented within a layer either by a pattern of activation of units in that layer (a *distributed* representation) or by a single activated unit (a *localist* representation).

One of the problems that connectionist models need to confront is that inputs are temporally sequenced, so that in order to combine constituent parts, the network must retain information about recently processed parts. Two possible approaches are the use of *simple recurrent networks* (*SRNs*) and, in localist networks, sustained activation. SRNs use one-to-one feedback connections from the hidden layer to special *context units* aligned with the previous layer (normally the input layer or perhaps a secondary hidden layer), in effect storing their current outputs in those context units. Thus at the next cycle, the hidden units can use their own previous outputs, along with the new inputs from the input layer, to determine their next outputs. In localist models it is common to assume that once a unit (standing for a particular concept) becomes active, it stays active for some length of time, so that multiple concepts corresponding to multiple parts of the same sentence, and their properties, can be simultaneously active.

A problem that arises is how the properties of an entity that are active at a given point in time can be properly tied to that entity, and not to other activated entities. (This is the *variable binding* problem, which has spawned a variety of approaches—see Browne and Sun 1999). One solution is to assume that unit activation consists of pulses emitted at a globally fixed frequency, and pulse trains that are in phase with one another correspond to the same entity (e.g., see Henderson 1994). Much current connectionist research borrows from symbolic processing perspectives, by assuming that parsing assigns linguistic phrase structures to sentences, and treating the choice of a structure as simultaneous satisfaction of

symbolic linguistic constraints (or biases). Also, more radical forms of hybridization and modularization are being explored, such as interfacing a NN parser to a symbolic stack, or using a neural net to learn the probabilities needed in a statistical parser, or interconnecting the parser network with separate prediction networks and learning networks. For an overview of connectionist sentence processing and some hybrid methods (see Crocker 2010).

IC ANALYSIS

Immediate constituent analysis, also called **IC Analysis**, in linguistics, a system of grammatical analysis that divides sentences into successive layers, or constituents, until, in the final layer, each constituent consists of only a word or meaningful part of a word. (A constituent is any word or construction that enters into some larger construction.) In the sentence “The old man ran away,” the first division into immediate constituents would be between “the old man” and “ran away.” The immediate constituents of “the old man” are “the” and “old man.” At the next level “old man” is divided into “old” and “man.” The term was introduced by the United States linguist Leonard Bloomfield in 1933, though the underlying principle is common both to the traditional practice of parsing and to many modern systems of grammatical analysis.

A string of words (which consists of minimally one word) is a constituent in a tree if there is a node which exclusively dominates it, i.e. dominates all and only the words in that string. For instance, in (7f) each word is a separate constituent because each one is exclusively dominated by a node (he by the node Pron, will by the node Aux, go by the node V, home by the node Adv, very by the node Deg, and quickly by the node Adv), but the strings go home and very quickly are also constituents because they are exclusively dominated by the lower VP and the AdvP, respectively, and the string go home very quickly is a constituent, too, because it is exclusively dominated by the higher VP. However, the words home very do not form a constituent in (7f) because there is no node in this tree which would dominate these two words and only these two words. When a node dominates lower nodes without the intervention of intermediate nodes, we speak about immediate domination. A string of words is called an immediate constituent (IC) in a tree when there is a node which immediately dominates all and only the words in that string. Thus, the immediate constituents of the sentence in (7f) are He, will, and go home very quickly, because these are the NP, Aux and VP which are immediately dominated by the sentence. The sentence is “mother” to its

immediate constituents, the immediate constituents are “daughters” to the sentence, and “sisters” to each other. The immediate constituents (i.e. daughters) of the VP go home very quickly are the lower VP go home and the AdvP very quickly. The immediate constituents of the lower VP go home are the V go and the AdvP home, and those of the AdvP very quickly are the DegP very and the Adv quickly. 6.3 Simple and complex sentences Until now, all the constituents (apart from the topmost ones) within our example sentences have been phrases and lexical items of various kinds: NPs and Ns, VPs and Vs, APs and As, AdvPs and Advs, PPs and Ps, DegPs and Degs, Auxes and Ds. None of the constituents was a sentence (S). Therefore we can say that all our examples so far have been simple sentences. A simple sentence is a sentence which contains no lower sentence (clause) embedded in it; to put it in another way, it is a sentence which has no S-node other than the topmost S-node in it. However, it can happen that a non-topmost constituent within a sentence is itself a sentence. This is the case in (9), where the complement (more precisely the object) of the verb believes is not an NP but an S. This lower sentence (S2) functions as a complement clause within the higher sentence (S1). (9) [S1 Peter believes [S2 that you will buy a car]]. The phenomenon in which a constituent contains constituents of the same category as itself is known as recursion. For instance, in our previous examples (7e) and (7f) we saw that a VP contained a lower VP. In (9), however, recursion applies to the category S, so here we can speak about sentential or clausal recursion. A sentence containing a lower sentence embedded in it is called a complex sentence. (9) is a complex sentence, because it contains two sentences: a higher one, called matrix clause: Peter believes (that) you will buy a car, and a lower one, called embedded clause or subordinate clause (or just subclause, for short): (that) you will buy a car. It can happen that a subclause has its own subclause and so the upper sublause is the matrix clause of the lower The topmost matrix clause minus the subclause it contains is also known as the main clause. So in both (9) and (10) the main clause is Peter believes … Let us now consider two different kinds of subclause in (12a) and (12b). (12)a. I didn’t know [George/he collects stamps]. b. I’ve never known [George/him collect stamps]. In (12a) the verb collects carries the inflectional suffix -s, which shows that the verb is inflected for agreement with its subject (third person singular) and simultaneously for present tense. We regard tense as an inflection on the first auxiliary or, if there is no auxiliary, on the verb in the sentence, consequently we distinguish only two tenses in English: present tense, e.g. collect-s and collect-0 or will-0, and past tense, e.g. collect-ed, or will-ed = would. 3 By contrast, in (12b) the verb collect does not agree with George and it is tenseless. The verb in (12b) is uninflected for agreement and tense. A further difference is that in (12a) the subject of the embedded clause, George, can be

replaced by a pronoun in the nominative case: he, but in (12b) George can only be replaced by a pronoun in the accusative case: him. In (13a) the auxiliary will is inflected for tense (it is in the present tense: will-0, its past tense form would be: would, i.e. will-ed). This is in contrast with the untensed particle to of the infinitive in (13b). And just like in (12a) and (12b), the subject of the bracketed clause, George, can be replaced by the nominative case pronoun he in (13a) and the accusative case pronoun him in (13b). (13)a. I expect [George will win]. / I expect [he will win]. b. I expect [George to win]. / I expect [him to win]. From (12) and (13) we can conclude that sentences or clauses can be finite and nonfinite. A finite clause has a subject in the nominative case and contains a verb or an auxiliary inflected for tense / agreement. A non-finite clause does not have a nominative subject and does not contain a verb or auxiliary inflected for tense / agreement. The subclauses in (12a) and (13a) are finite, whereas those in (12b) and (13b) are non-finite. The subject of an English non-finite subclause can be an invisible pronoun called PRO (pronounced: 'big pro'), too, as in (14a) and (14b). (14)a. [PRO to swim here] is dangerous. b. We want [PRO to buy a new printer]. In (14a) the PRO has a general interpretation ('anyone'), whereas in (14b) it inherits the features of its antecedent, the main clause subject we, and so PRO, like we, is also first person plural.4 PRO satisfies the requirement that we have set up for the subjects of non-finite clauses: it is not a subject in the nominative case. Non-finite verb forms are the bare infinitive and to-infinitive forms (e.g. (to) write), the -ing form (e.g. writing), and the -en form (e.g. written) of verbs. (Though the latter two are inflected, they are not inflected for tense and agreement!) To sum up: the bracketed subclauses in (12b), (13b), (14a) and (14b) are non-finite, those in (9), (10), (12a) and (13a) are finite. All the main clauses are finite. Finally, consider the bracketed subclause in (15), which we saw last week. This is part of the NP the shoes which we saw last week. (15) [S1 I've bought [NP the shoes [S2 which we saw last week]]]. Here the NP itself is the object (complement) of the matrix verb bought. The subclause modifies (is an adjunct to) the noun shoes. Since which relates to (refers back to) shoes, it is called a Relative Pronoun and the subclause which contains it is called a Relative Clause. More precisely, we can say it is a Defining (or Restrictive) Relative Clause because it helps identify the referent of the word shoes, i.e. tells us which particular shoes the speaker is actually talking about.

TRANSFORMATIONAL-GENERATIVE GRAMMAR

The most significant development in linguistic theory and research in the 20th century was the rise of generative grammar, and, more especially, of transformational-generative grammar,

or transformational grammar, as it came to be known. Two versions of transformational grammar were put forward in the mid-1950s, the first by Zellig S. Harris and the second by Noam Chomsky, his pupil. It was Chomsky's system that attracted the most attention. As first presented by Chomsky in Syntactic Structures (1957), transformational grammar can be seen partly as a reaction against post-Bloomfieldian structuralism and partly as a continuation of it. What Chomsky reacted against most strongly was the post-Bloomfieldian concern with discovery procedures. In his opinion, linguistics should set itself the more modest and more realistic goal of formulating criteria for evaluating alternative descriptions of a language without regard to the question of how these descriptions had been arrived at. The statements made by linguists in describing a language should, however, be cast within the framework of a far more precise theory of grammar than had hitherto been the case, and this theory should be formalized in terms of modern mathematical notions. Within a few years, Chomsky had broken with the post-Bloomfieldians on a number of other points also. He had adopted what he called a "mentalistic" theory of language, by which term he implied that the linguist should be concerned with the speaker's creative linguistic competence and not his performance, the actual utterances produced. He had challenged the post-Bloomfieldian concept of the phoneme (see below), which many scholars regarded as the most solid and enduring result of the previous generation's work. And he had challenged the structuralists' insistence upon the uniqueness of every language, claiming instead that all languages were, to a considerable degree, cut to the same pattern—they shared a certain number of formal and substantive universals.