The Verb Phrase
&
Subcategorization

Presentation by:
B. Senthil Kumar
Dept. of CSE

Overview

- The Verb phrase [*VP*] and Subcategorization
- Auxiliaries
- Grammar Equivalence
- Chomsky Normal Form [CNF]
- Finite-State and Context-Free Grammars

The Verb Phrase

- The VP consists of the verb and a number of other constituents.

VP → *Verb*	disappear
VP → *Verb NP*	prefer *a morning flight*
VP → *Verb NP PP*	leave *Boston* in the morning
VP → *Verb PP*	leaving *on Thursday*

- An entire embedded sentence, called **sentential complement**, can follow the verb.

You [$_{VP}$ [$_{V}$ said [$_{S}$ there were two flights that were the cheapest]]]

You [$_{VP}$ [$_{V}$ said [$_{S}$ you had a two hundred sixty six dollar fare]]]

[$_{VP}$ [$_{V}$ Tell] [$_{NP}$ me] [$_{S}$ how to get from the airport in Philadelphia to downtown]]

I [$_{VP}$ [$_{V}$ think [$_{S}$ I would like to take the nine thirty flight]]

  - *VP* → *Verb S*

The Verb Phrase

- Another potential constituent of the VP is another VP
  - Often the case for verbs like *want, would like, try, intent, need*

- I *want* [$_{VP}$ to fly from Chennai to Singapore]
  Hi, I *want* [$_{VP}$ to arrange three flights]

Hello, I'm *trying* [$_{VP}$ to find a flight that goes from Pittsburgh to Denver after two p.m.]

  - Verbs can also be followed by *particles*, word that resemble a preposition but that combine with the verb to form a *phrasal verb*, like *take off*.
    - These particles are integral part of the verb in a way that other post-verbal elements are not;
    - Phrasal verbs are treated as *individual verbs* composed of two words.

Subcategorization

- A VP can have many possible kinds of constituents, not every verb is compatible with every VP.
  - *I want a flight … [a verb with an NP complement]*
  - *I want to fly to … [a verb with an infinitive VP complement]*
  - **I found to fly to Dallas.*
- The idea that verbs are compatible with different kinds of complements
  - Traditional grammar **subcategorize** verbs into two categories (transitive and intransitive).
  - Transitive verbs take a direct object *NP (I found a flight)* while intransitive verbs do not (**I disappeared a flight*)
  - Modern grammars distinguish as many as 100 subcategories

Subcategorization

- The idea that verbs are compatible with different kinds of complements:
  - A verb like *find* subcategorizes for an *NP*, verb like *want* subcategorizes for either an *NP* or a *infinitive VP*
  - These constituents are called the *complements* of the verb
  - These possible set of complements are called the *subcategorization frame*

The Verb Phrase & Subcategorization

- Some subcategorization frames and example verbs:

Frame	Verb	Example
ϕ	eat, sleep	I want to eat
NP	prefer, **find**,leave	Find [$_{NP}$ the flight from Pittsburgh to Boston]
NP NP	show, give, find	Show [$_{NP}$ me] [$_{NP}$ airlines with flights from Denver]
PP$_{from}$ PP$_{to}$	fly, travel	I would like to fly [$_{PP}$ from Boston] [$_{PP}$ to New York]
NP PP$_{with}$	help, load	Can you help [$_{NP}$ me] [$_{PP}$ with a flight]
VPto	prefer,want, need	I would prefer [$_{VPto}$ to go by United airlines]
VPbrst	can, would, might	I can [$_{VPbrst}$ go from Boston]
S	mean	Does this mean [$_{S}$ AA has a hub in Boston?]

The Verb Phrase & Subcategorization

- To relate verbs and their complements use agreement features:
  - Make separate subtypes of the Verb class

Verb-with-NP-complement ⟶ *find* | *leave* | *repeat* | …

Verb-with-S-complement ⟶ *think* | *believe* | *say* | …

Verb-with-Inf-VP-complement ⟶ *want* | *try* | *need* | …

VP Rules with subtype:

VP ⟶ *Verb-with-no-complement*	disappear
VP ⟶ *Verb-with-NP-complement NP*	prefer a morning flight
VP ⟶ *Verb-with-S-complement S*	said there were two flights

  - Problem: a vast explosion in the number of rules
  - Solution: ***feature structure***

Auxiliaries

- ***Auxiliaries*** *or* ***helping verbs***
  - A subclass of verbs
  - Having particular syntactic constraints which can be viewed as a kind of subcategorization
  - Including the **modal** verbs *can, could, may, might, must, ought to, will, would, shall,* and *should*
  - The **perfect** auxiliary – *have*
  - The **progressive** auxiliary – *be (am, is, are)*
  - The **passive** auxiliary – *be (was, were, being, been)*

Auxiliaries

- Modal verbs subcategorize for a VP whose head verb is a bare stem.
  - *can go in the morning, will try to find a flight*
  - The perfect verb *have* subcategorizes for a *VP* whose head verb is the past participle form:
    - *have booked 3 flights*
  - The progressive verb *be* subcategorizes for a *VP* whose head verb is the gerundive participle:
    - *am going from Atlanta*
  - The passive verb *be* subcategorizes for a *VP* whose head verb is the past participle:
    - *was delayed by inclement weather*

Auxiliaries

- A sentence may have multiple auxiliary verbs, but they must occur in a particular order.
    - modal < perfect < progressive < passive

modal perfect	could have been a contender
modal passive	will be married
perfect progressive	have been feasting
modal perfect passive	might have been prevented

Grammar Equivalence

- Two grammars are equivalent if they generate the same set of strings.
- Two kinds of equivalence
  - *Strong equivalence*
    - If two grammars generate the same set of strings *and* if they assign the same phrase structure to each sentence
  - *Weak equivalence*
    - Two grammars generate the same set of strings but *do not* assign the same phrase structure to each sentence.

Chomsky Normal Form [CNF]

- It is useful to have a **normal form** for grammars.
  - A CFG is in **Chomsky normal form** (CNF) if it is ε-free and each production is either of the form $A \rightarrow B\ C$ or $A \rightarrow a$
- Any grammar can be converted into a weakly-equivalent CNF grammar.
  - For ex: $A \rightarrow B\ C\ D$ can be converted into the following two CNF rules:

    $A \rightarrow B\ X$

    $X \rightarrow C\ D$

FSA and CFG

- Recursion problem with finite-state grammars
  - Recursion cannot be handled in finite automata
  - Recursion is quite common in a complete model of NP

Nominal → Nominal PP -– recursion
Noun Phrase [NP]
(Det)(Card)(Ord)(Quant)(AP)Nominal
*(Det)(Card)(Ord)(Quant)(AP)Nominal (PP)**
*(Det)(Card)(Ord)(Quant)(AP)Nominal (P NP)**
*(Det)(Card)(Ord)(Quant)(AP) Nominal (RelClause|GerundVP|PP)**

- Chomsky (1959) proved that a context-free language *L* can generated by a finite automaton if and only if:
  - there is a CFG that generates *L* that does not have any center-embedded recursions (of the form $A \rightarrow \alpha A \beta$)
- An augmented version of the FSA: the *recursive transition network* [**RTN]**

References

- Speech and Language Processing, Daniel Jurafsky and James H. Martin [Chapter 9. Context-Free Grammars for English]

<Thank You />