NATURAL LANGUAGE PROCESSING

BY
ASSIST. PROF. DR. ISRAA ABDULAMEER

NLP SYLLABUS

1- TEXT PROCESSING

- Introduction to NLP:
- Understanding
- Extracting proper noun Algorithm
- The Dictionary & the Morphology ,
- Syntax analysis
- Rules of English Grammar, Example of PROLOG program of English Grammar solved in: 1- Append Mechanism. Syntax Analysis, Formal Method, Append Mechanism with Singular & Plural consideration
- 1- Append Mechanism. Syntax Analysis, Formal Method, Append Mechanism with Singular & Plural Consideration.
- 2- Difference Pair Idea, Semantic Analysis (Formal Method), Extracting meaning from keywords, Example of PROLOG program (DOCSYS) for a manual of a company.
- Machine Translation (MT)

2- SPEECH RECOGNITION

- Spoken language Processing
- SR System model
- Hidden Markov Model (HMM
- speech synthesis
- The relationship between NL & SR, Compares between Written text processing & Speech processing, Natural Language Generation: Example and Program.

REFERENCES:

- 1. William A. Stubblefield & Luger E.George,"Artificial
- 2. Intelligence and the Design of Expert Systems", 1998.
- 3. Daniel H. Marcellus " Artificial Intelligence and the design of expert systems" 1998. ch8 & ch9
- 4. Daniel Jurafsky and James H. Martin "Speech and language processing : Introduction to natural language processing , computational linguistics and speech recognition" second edition 2006. ch8 & ch24.

NATURAL LANGUAGE PROCESSING

NATURAL LANGUAGE PROCESSING

- A natural language is human spoken language, such as English, Arabic , Chinese, Japanese and etc..

- NLP is a subfield of Artificial Intelligence and linguistics. It studies the problems of automated generation and understanding of natural human languages where the natural language generation systems convert information from computer databases into normal-sounding human language, and natural language understanding systems convert samples of human language into more formal representations that are easier for computer programs to manipulate.

NATURAL LANGUAGE PROCESSING:

- 1. Understanding written text (written programs, search on internet, Email & chat, Microsoft word,etc).
- 2. understanding spoken language(commands, robotics S/W & H/W, modern car's,…etc)
- 3. signs (describing objects)

NLP GOAL

- The basic goal of (NLP) is to enable a person to communicate with a computer in a Language that they use in their everyday life.

STAGE OF LANGUAGE ANALYSIS:

- **Generally Five Processing Stages in a NLP System**
- **Phonological Analysis**
- **Morphological Analysis (lexical)**
- **Syntactic Analysis**
- **Semantic Analysis**
- **Pragmatic Analysis**

WRITTEN TEXT PROCESSING:

- 1. Formal method.
- 2. Informal method

- **Informal method:**
- Example:
- Computers milk drinks.
- Computer drinks milk.
- Computers use data.

- **Formal method:**
- 1. lexical analysis.(word)
- 2. syntactical analysis.(grammars)
- 3. semantic analysis.(meanings)

INTELLIGENT ROBOT:

- The robot would have to know:
- 1. The meaning of the words.
- 2. Relationship of one word to another.
- 3. Knowledge of grammars.
- 4. Associate descriptions and objects.
- 5. Analyze sentence in relation to another sentences.

- e.g
- - John drank milk
- -he then put on his coat.

• WHAT UNDERSTANDS?

- To understand something is transform it from one representation into another, where this second representation has been chosen to correspond to a set of available actions that could be performed and where the mapping has been designed. So that for each event, an appropriate action will be done.

STAGE OF UNDERSTANDING NLP:

- Generally Processing Stages in a NLP System

1. Phonological Analysis
2. Morphological Analysis (lexical Analysis)
3. Syntactic Analysis
4. Semantic Analysis
5. Discourse analysis
6. Pragmatic Analysis

THE PHONOLOGICAL ANALYSIS

- Analysis of speech sounds of the world's languages, with a focus on both their articulator and acoustic properties. An introduction to phonetic alphabets, including practice in transcribing a variety of language samples. Analysis of the systematic organization of speech sounds, with reference to features and supra segmental.

THE LEXICAL ANALYZER

- Reads the input text of source language character and produces tokens such as (*names , keywords, punctuation marks ,discards white space and comments*), *which are* the basic lexical units of the language. The process of breaking-up a text into its constituent tokens is known as tokenization. Tokenization occurs at a number of different levels: a text could be broken up into paragraphs, sentences, words, syllables, or phonemes.

SYNTACTIC ANALYSIS

- Is concerned with the construction of sentences. Syntactic structure indicates how the words are related to each other. Syntax tree is assigned by a grammar and a lexicon.

SEMANTIC ANALYSIS

- Which produce a representation of the meaning of the text, the representation that commonly used include conceptual dependency, frames, and logic base representation. and it uses the knowledge about the meaning of the word and linguist structure, such as case roles of nouns or the transitivity.

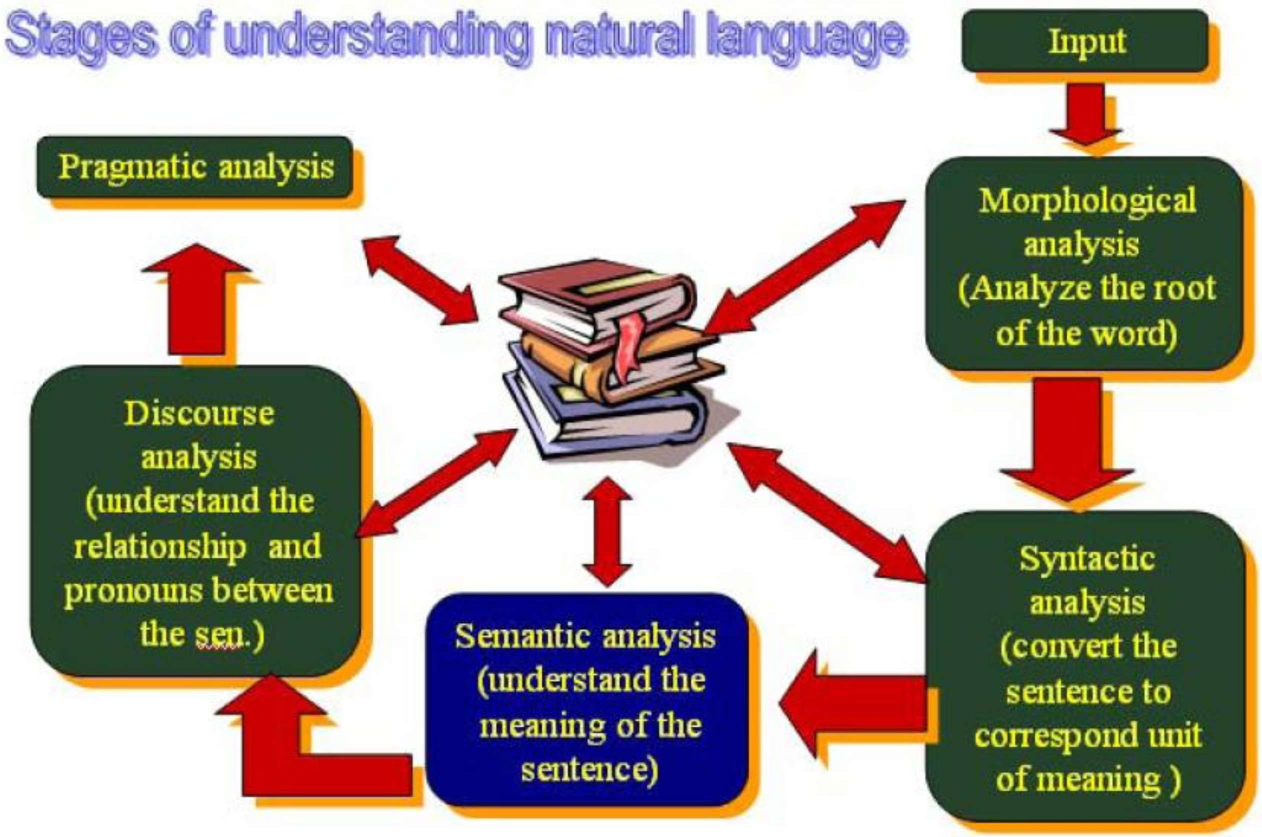
DISCOURSE ANALYSIS

- Analyze the effect of previous sentence on the next sentence.

PRAGMATIC ANALYSIS

- Interpret the sentence according to its meaning.

STAGES OF UNDERSTANDING NL


Stages of understanding natural language
Input
Morphological analysis (Analyze the root of the word)
Syntactic analysis (convert the sentence to correspond unit of meaning)
Semantic analysis (understand the meaning of the sentence)
Discourse analysis (understand the relationship and pronouns between the sen.)
Pragmatic analysis

• WHAT MAKES UNDERSTANDING HARD?

1. The complexity of the target representation into which the matching is being done.

Example 1:

- I want to read all books of computer.

Search = book ^ computer

Example 2:

- Bill told his mother he would not go to school today, She feels sorry about him.

2. Types of mapping:

a. one-to –one :

Example: A:= B+C*D

b. One-to-many:

Example: √16 = 4*4 or -4 *-4.

c. Many – to-one:

Example: The chickens are ready to eat.

d. Many-to-many.

Example: Tell me about the last computer books.

I am interested in computer books.

There are many computer books.

3. Level of interaction among component.

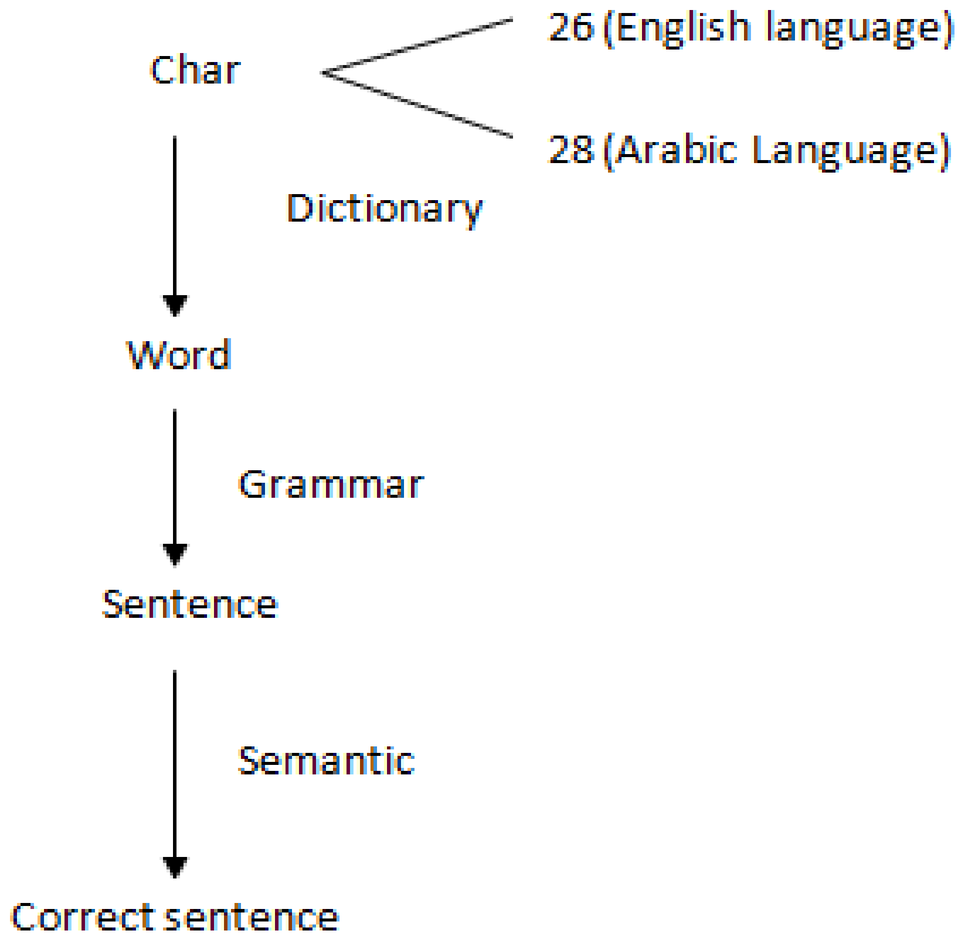
Example:

```
X:= A*B +C * sin (D)
x:= A*B+C *cos (D)
```

THE DICTIONARY AND THE MORPHOLOGY

- The Dictionary: contains all the available words in a language with its classification such as (nouns, pronouns, verbs, adjectives, adverbs, and .. etc).

- The Morphology: is the basic unit in NLP , and it is the process of breaking a word into its smallest meaningful components or morphemes is concerned with breaking a word in to its root , suffix or prefix.


Char
26 (English language)
28 (Arabic Language)
Dictionary
Word
Grammar
Sentence
Semantic
Correct sentence

MORPHOLOGY ANALYSIS WITH ENGLISH LANGUAGE

- Dictionary words can be defined as facts such as:

Word (“noun, name, plural, human, move,…)
Word (“boy”, name, “boys”, human, move,…).

Word (the verb, verb, past, continues, past participle)
Word (“go”, verb”, “went”, “going”, gone”,…).
Word (tall, adjective, human, type of adjective,…).

In Morphology analysis, the prefix or suffix either removed from the word or added to it in order to extract the root at the word.

Prefix	Suffix
in	ing
im	ed
un	er
re	es
	s
	tion
	sion
	ment

- Sometimes the suffix and the prefix is a part from the sentence like:
- (im & in isn't prefix here)

important,

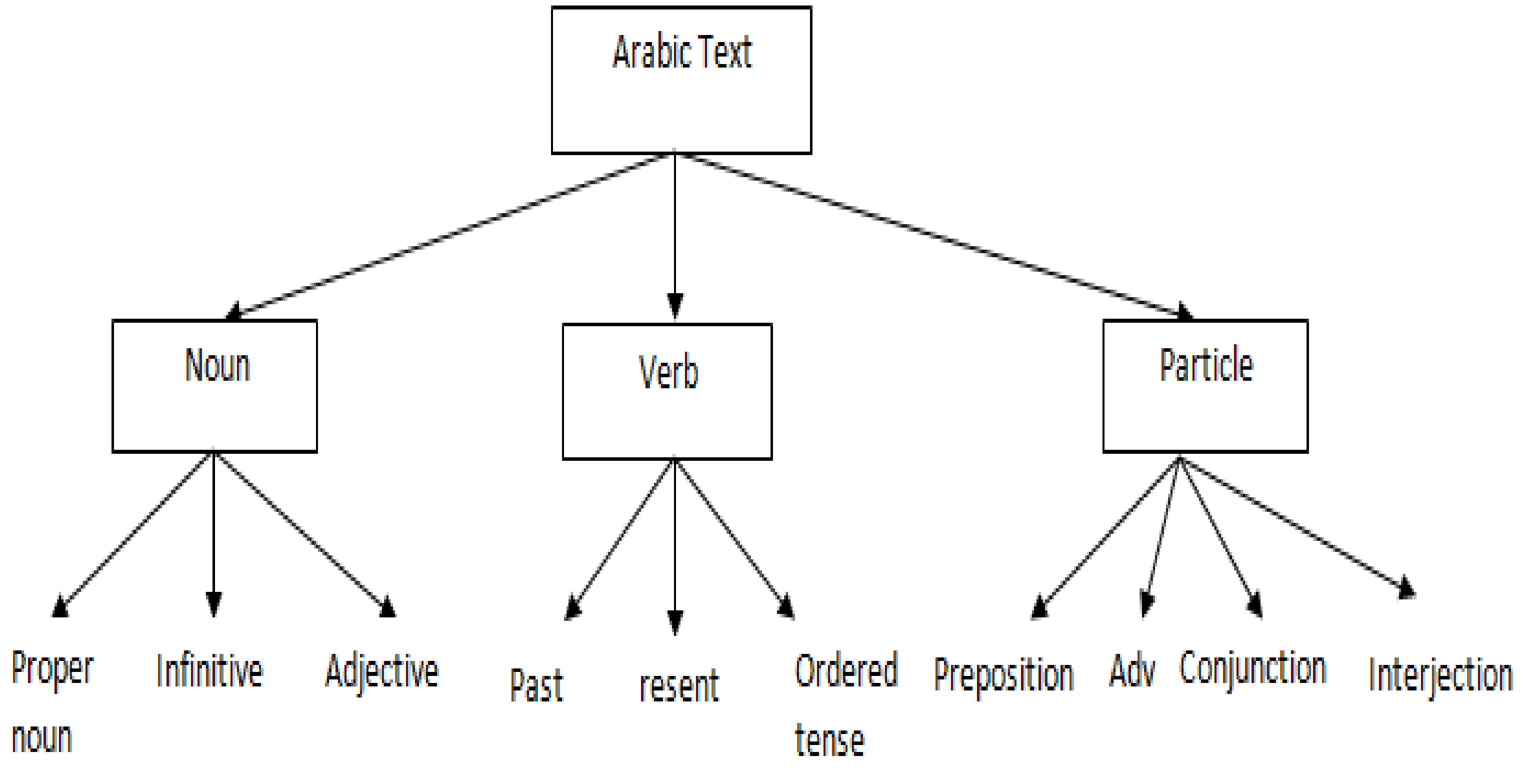
input,

- (sion & ing isn't suffix here)

mission, (sion isn't suffix here)

sing…, etc.

MORPHOLOGY ANALYSIS WITH ARABIC LANGUAGE:



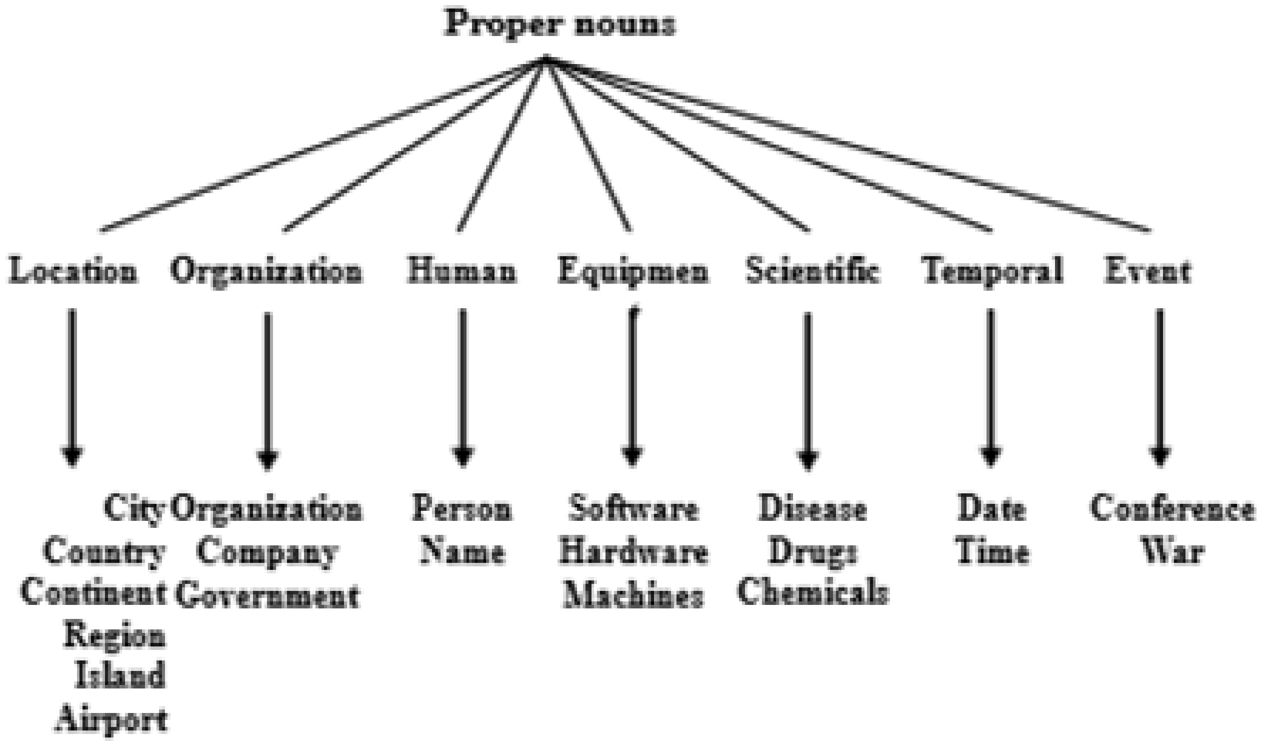
A Classification of Arabic words according to the part of speech

- Nouns: is sub classified into three sub categories: proper noun , infinitive and adjectives.

- Proper noun can be either Singular noun like (Ali , Ahmed, etc) or Composite noun like (Abu Jaafar, Abdul Rahman, etc).

PROPER NOUN CLASSIFICATION

Proper noun classification



EXTRACTING PROPER NOUN ALGORITHM

- Algorithm steps to extract proper noun in Arabic language is described as follows:

* Remove diacritics

 - Diacritics: special marks are put above or below the characters to determine the correct pronunciation. Such as (ٌ, ُ, ّ, ِ) e.g. to Arabic (language).

* Remove punctuation and non letters. Such as “ . ,!,? ".

* Search in keyword file and special verbs using set of rule.

* Check for the prefix and strips off " ال, لل, اللا...الخ" and remove it.

* Check for suffixes, “ ية, ين, ون, ان, ات, تان."

* Extract the words that follow keywords and save them in the proper noun database.

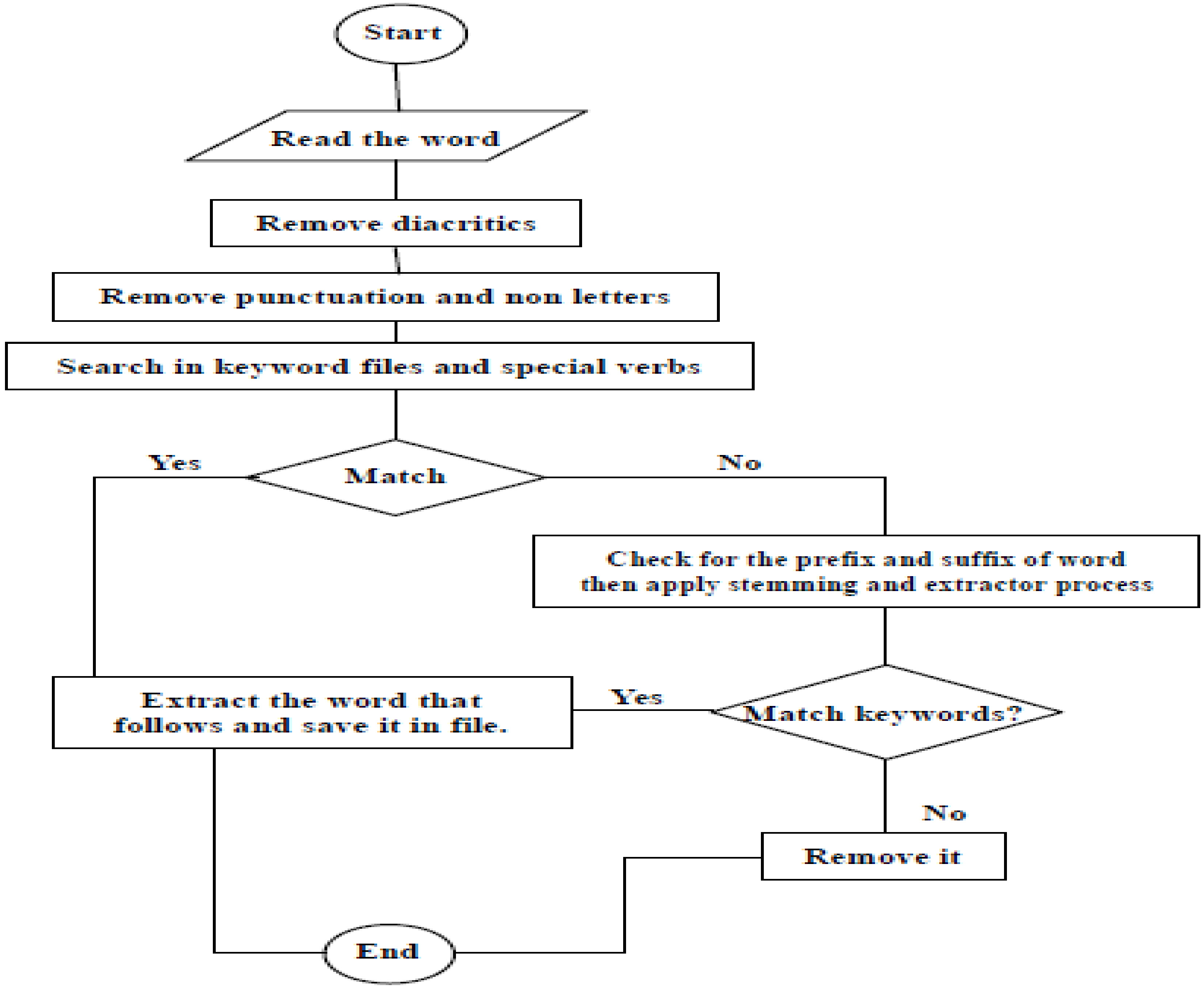
EXTRACTING PROPER NOUN ALGORITHM

- The rule for word that always followed by proper noun:
- - Any word follows any of the following words "بنت , بن , ابن "((must be a proper noun.
- -Any word follows any of the following words " ماعدا ,خلا , حاشا“((must be a proper noun Such as: جاء الطلاب ماعدا سعيد
- - Any word follows any of the following words " , عمة , عم ,اخت , اخ , اب , ام خالة ,خال" which means kunai must be a proper noun.
- - Any word follows any of the following words "يا , أيا" must be proper noun.
- - Any word follows any of the following words "عبد " or " " must be noun.
- - the combination of followed by often one of the Muslim 99 names of god such as must be a proper noun.
- - Any word follows any of the following words Prepositions" , في , على ,عن من وغيرها من حروف الجر". And has the pattern " فاعل " such as

الكاتبان

Suffix فاعلprefix

THE AUTOMATIC ALGORITHM FOR EXTRACTING PROPER NOUN


Start
Read the word
Remove diacritics
Remove punctuation and non letters
Search in keyword files and special verbs
Match
Yes
No
Check for the prefix and suffix of word then apply stemming and extractor process
Match keywords?
Yes
Extract the word that follows and save it in file.
No
Remove it
End

KEYWORD /SPECIAL VERB		KEYWORD/ SPECIAL VERB
Mr.		Announced
President		Newspaper
Professor		Bank
Country		Sea
City		Mother
Conference		Father
Exhibit		Son
War		Republic
Said		Ministry

SYNTACTIC PARSING

- Syntactic parsing is the step in which a flat input sentence is converted into a hierarchical structure that correspond to the units of meaning in the sentence, this process is called *parsing*.
- Parsing is the problem of constructing a derivation or a parse tree for an input string from a formal definition of a grammar. Parsing algorithms falls into two classes: top-down parsers, which begin with top level sentence symbol and attempt to build a tree whose leaves match the target sentence. And bottom-up-parsers, which starts with the words in the sentence (the terminal and attempt to find a series of reductions that yield the sentence symbol.

TYPES OF LANGUAGES AND GRAMMARS

- The grammar of Chomsky is consist of
- G = (N, T, S, P)
- Where:
- N = non terminal
- T = terminal
- S = start symbol
- P = production rule
- Type 0: phrase structure grammar (PSG)

- This step plays an important role in many natural languages understanding system for two reasons:
- 1. Semantic processing must operate on sentence components. If there is no parsing step then the semantics system must decide on its own Component. If the parsing is done, then its constrain the number of component that semantic can considered, additionally parsing is computationally less expensive than is semantic processing, and reducing overall system complexity.

- 2. It is possible to extract the meaning of sentence using grammatically facts, it is not always possible to do so , for example

- the satellite orbited mars (correct sentence)

- mars orbited the satellite

in the second sentence, syntactic facts need an interpretation in which a plant (mars) move around a satellite .

1. G is also called a **Type-0** grammar or an **unrestricted** grammar.

2. G is a **Type-1** or **context-sensitive** grammar if each production $\alpha \to \beta$ in P satisfies $|\alpha| \leq |\beta|$. By "special dispensation," we also allow a Type-1 grammar to have the production $S \to \epsilon$, provided S does not appear on the right-hand side of any production.

3. G is a **Type-2** or **context-free** grammar if each production $\alpha \to \beta$ in P satisfies $|\alpha| = 1$; i.e., α is a single nonterminal.

4. G is a **Type-3** or **right-linear** or **regular** grammar if each production has one of the following three forms:

$$A \to cB, \quad A \to c, \quad A \to \epsilon,$$

where A, B are nonterminals (with $B = A$ allowed) and c is a terminal.

CONTEXT FREE GRAMMAR

- a grammar is a set of rules that define the legal structure in a language.
- A grammar is said to be context – free if the structure of each one of its parts has no influence on that on any other part.
- The general form of CFG is as follows:-

$$A \rightarrow B$$

- Where A is non terminal (only one capital letter in the left side)
- B is a sequence of terminal and/or non terminal.

EXAMPLE:

- “The president has a cat”
- CFG:

- Sentence → NP.VP
- NP→ article . noun
- VP→ V NP
- noun→ president / cat
- artical→ the / a
- V → has

- Parse tree

  The parsing process takes the rules of the grammar and compares them against the input sentence. Each rule that matches the sentence adds something to a complete structure that is being build for the sentence; the simplest structure build is the parse tree.

$$\begin{tikzcd}
 & & \text{Sentence} \arrow[dl] \arrow[dr] & & & & \\
 & \text{NP} \arrow[dl] \arrow[d] & & \text{VP} \arrow[d] \arrow[dr] & & & \\
\text{artical} \arrow[d] & \text{noun} \arrow[d] & & \text{v} \arrow[d] & \text{NP} \arrow[dr] \arrow[drr] & & \\
\text{the} & \text{President} & & \text{has} & & \text{artical} \arrow[d] & \text{noun} \arrow[d] \\
 & & & & & \text{a} & \text{cat}
\end{tikzcd}$$

Top – down derivation

Sentence → NP.VP

NP→ article . noun

VP→ V NP

noun→ president / cat

artical→ the / a

V → has

Sentence
NP VP
Article Noun VP
The Noun VP
The President VP
The president V NP
The president has NP
The president has article Noun
The president has a noun
The president has a cat

Bottom-up derivation

The president has a cat
Article president has a cat
Article noun has a cat
Article noun V a cat
Article noun V article cat
Article noun V article noun
NP V article noun
NP V NP
NP VP
Sentence

- Advantage of CFG
- 1. its ability to describe, build and design structural language (programming language)

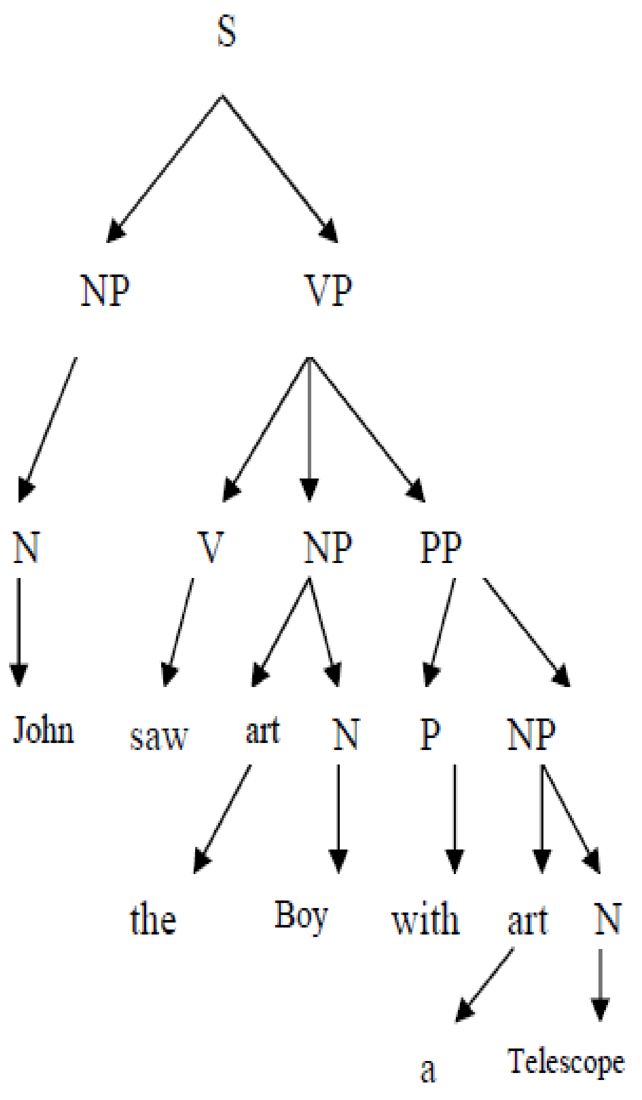
like pascal , FORTRAN.

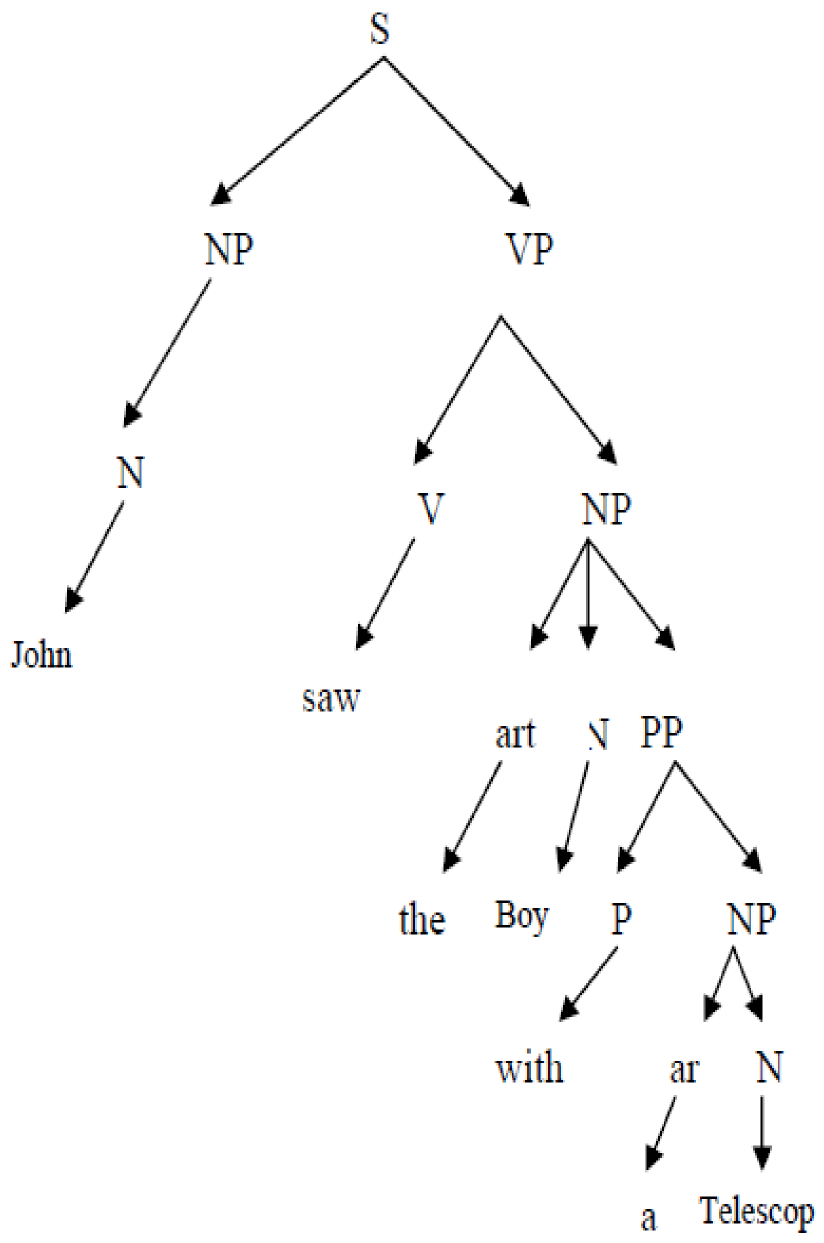
- 2. it is easy to be programmed and combined to automatic processing system.

- Disadvantage of CFG
- 1. limited capability when concerned with NLP, natural languages have too many complicated rules in order to define them that would require too many CFG rules to represent them all.
- 2. CFG is unable to describe discrete structural sentences, such as: Arthur, Barry, and David are husbands of Jane, Joan and Jill.

- Grammar ambiguity
- Ambiguity is the case in which a grammar produces more than one parse tree for some given sentence, such a grammar is called ambiguous grammar. In this case we can not determine which parse tree to select for a given sentence.

- Example:
- “ john saw a boy with a telescope”


S
NP
VP
N
V
NP
PP
John
saw
art
N
P
NP
the
Boy
with
art
N
a
Telescope


S
NP
VP
N
V
NP
John
saw
art
N
PP
the
Boy
P
NP
with
ar
N
a
Telescop

THERE ARE SEVERAL WAYS TO SOLVE AMBIGUITY:

1. by using punctuation marks:

For our example using comma, “john saw the boy, with a telescope”

This will yield the first parse tree.

2. Consider the parse tree with the smallest number of syntactic nodes (fewer nodes at each tree level). This yield the second parse tree.

3. Consider the parse tree with the smallest number of parse tree levels. This will give the first parse tree (the first parse tree has only 5 levels, while the second parse tree has 6).

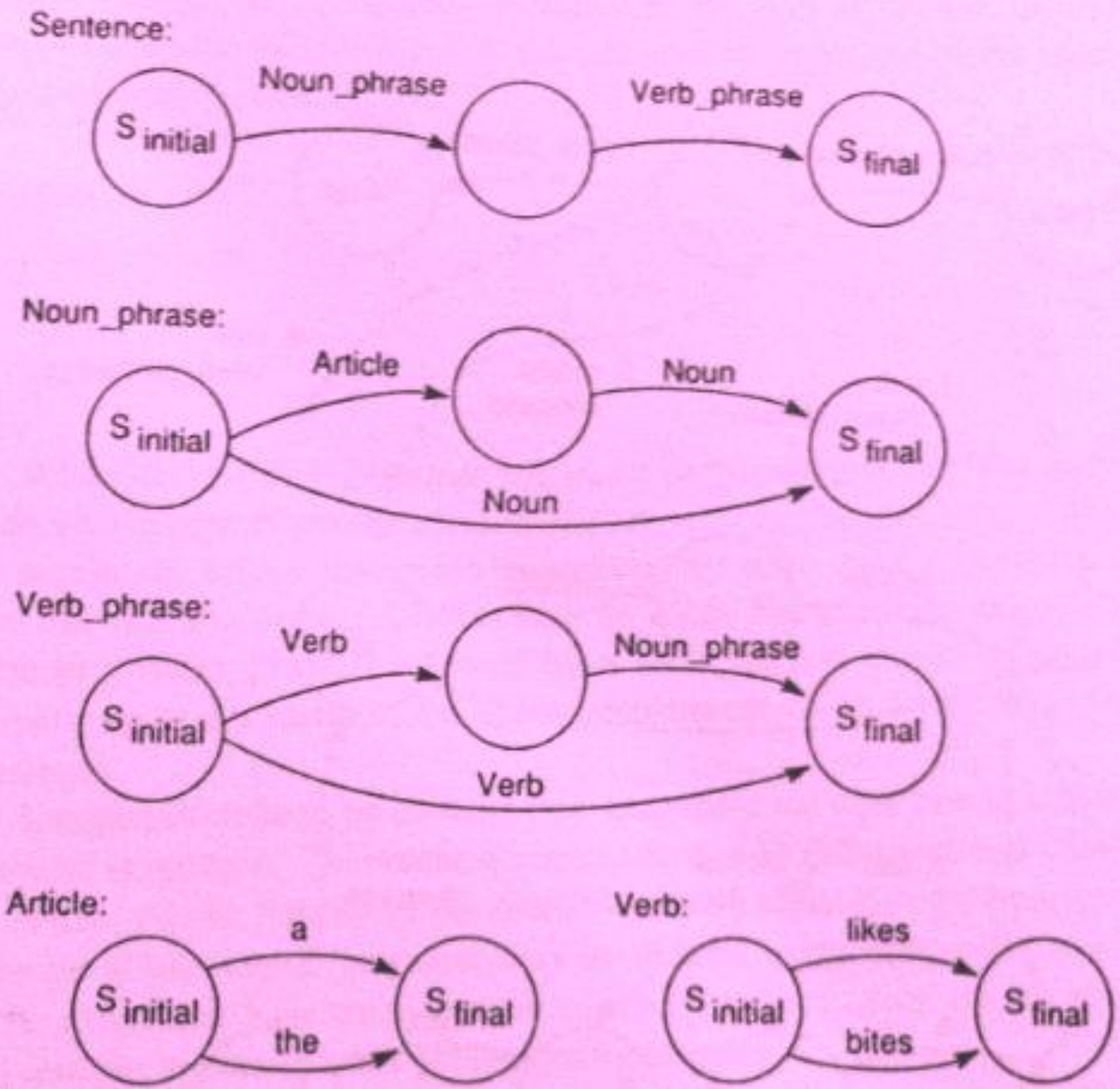
4. Allowing interaction with other processing stage: in our example “the telescope “is a device that used to see with, so it can be attached to the verb “saw” yielding the first parse tree.

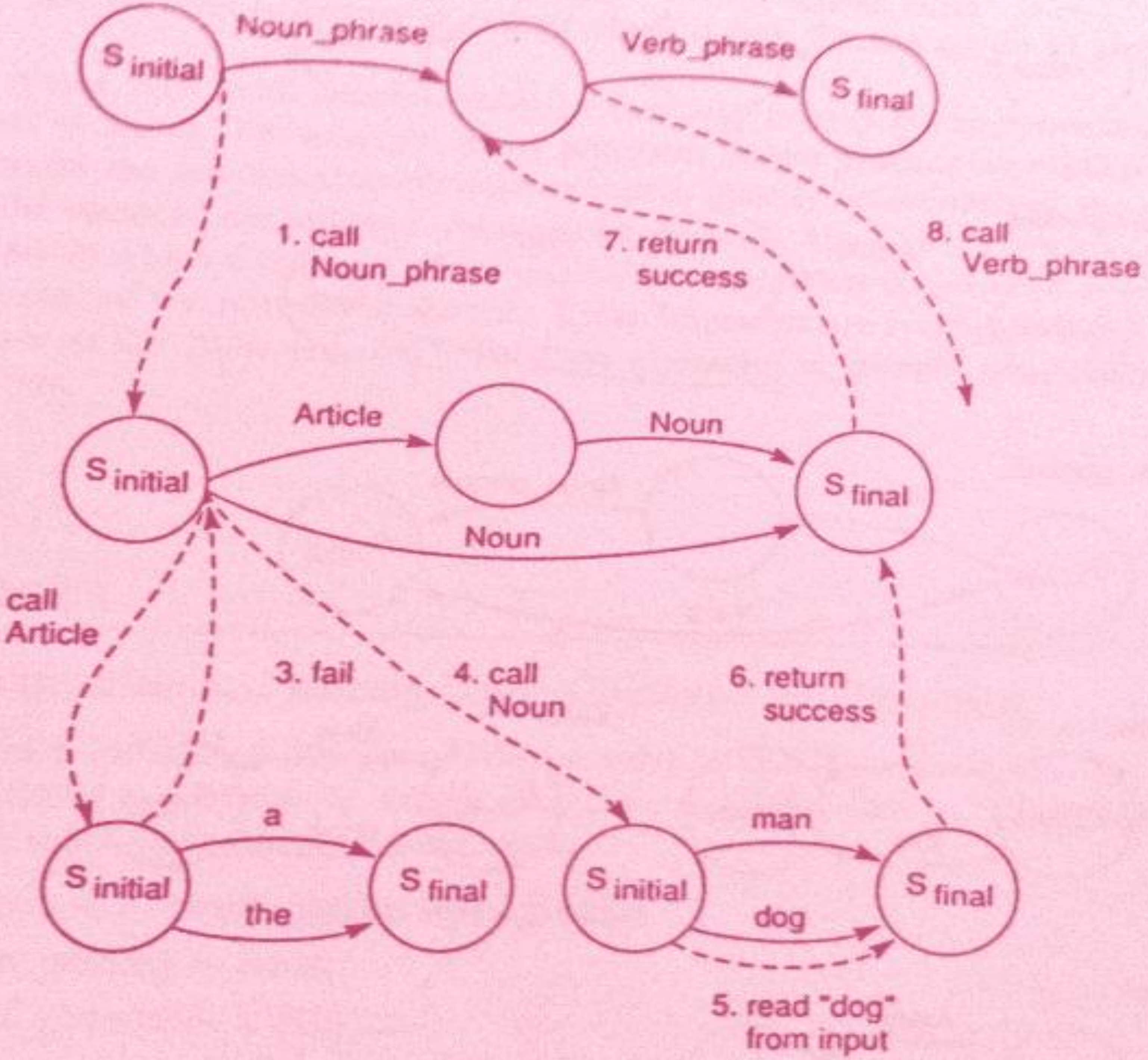
TRANSITION NETWORK PARSER

- A transition network parser represents the grammar as a set of finite state machine or transition network. Each network corresponds to a single non terminal in the grammar. Arc in the networks are labeled with either terminal or non terminal symbols. Each path in the network from the start state to the final state corresponds to some rule for that no terminal; the sequence on arc labels on the path is the sequence of symbols on the right hand side of the rule. The previous grammar is represented by the transition networks of figure (1). When there is more than one rule for non terminal, the corresponding network has multiple paths from the start to the goal. Finding a successful transition through the network for a no terminal correspond to the replacement of the non terminal by the right hand side of grammar rule. Consider a simple sentence “the man drives the car”. The steps in parsing the sentence are illustrated in figure (2).

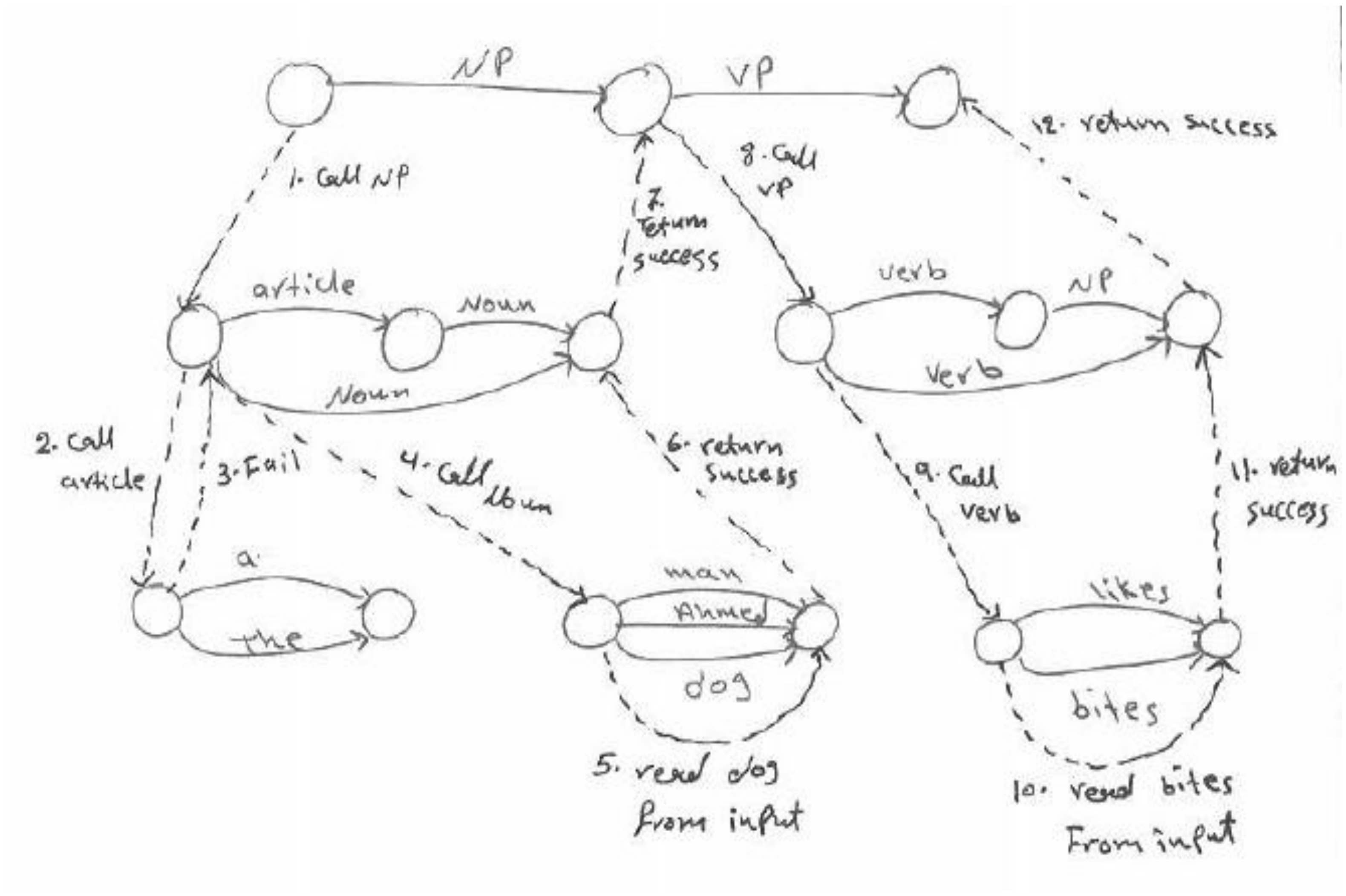
- Ex: ahmed likes the dog
- the dog bites a man

- Sentence → NP VP
- NP→ noun |article Noun
- VP→ verb |V NP
- noun→ ahmed | man| dog
- Article → the | a
- V → likes| bites


Sentence:
Noun_phrase
Verb_phrase
S initial
S final
Noun_phrase:
Article
Noun
S initial
S final
Noun
Verb_phrase:
Verb
Noun_phrase
S initial
S final
Verb
Article:
a
S initial
S final
the
Verb:
likes
S initial
S final
bites



S initial
Noun_phrase
Verb_phrase
S final
1. call Noun_phrase
7. return success
8. call Verb_phrase
S initial
Article
Noun
S final
Noun
call Article
3. fail
4. call Noun
6. return success
S initial
a
the
S final
S initial
man
dog
S final
5. read "dog" from input



Trace of a transition network parse of the sentence “ dog bites”

RULES OF ENGLISH GRAMMAR

- Viewing from the ***highest level, we can see a sentence as an ordered*** arrangement of noun phrases, verb phrases, and prepositional phrases. The purpose of a grammar is to define a complex object, such as a sentence that is built up from containment parts.
- Here is some acceptable patterns for ***grammatical sentences:***

GRAMMATICAL SENTENCES

- Sentence → NP , VP
- “ the big man cried”

- Sentence → NP , VP,NP
- “The lawyer read the will”

- Sentence → NP , VP,PP
- The cat slept on the important papers

NOUN PHRASE PATTERN

- NP → noun
- “Animals”

- NP → proper noun
- “John”

- NP → adjective, noun
- “Large salaries”

- NP → determiner, noun
- “The broker”

NOUN PHRASE PATTERN

- NP $\rightarrow$ determiner, adjective, noun
- “A spicy pizza”

- NP $\rightarrow$ determiner, adjective, adjective, noun
- “The big red apple”

PREPOSITIONAL PHRASE PATTERN

- PP → preposition , NP
- “ in the bank”

VERB PHRASE PATTERN

- VP → verb
- “Thinks”

- VP → verb, adverb
- “Schemed obsessively”

- At some point in the grammar, the lowest level units, words, must be cataloged. Every word that the grammar will recognize must be recorded along with its type.

- ***Suppose that all the legal words are shown below:***
- Determiner → a/ the/ that/ those
- Verb → thinks/ cries/ drinks/ sings
- Adverb → happily
- Noun → power/ money/ food
- Adjective → popular/ red
- Preposition → about/ in/ over

PROLOG PROGRAM FOR PARSE SENTENCES USING APPEND MECHANISM

- ***Example1: Sentence "The big man cried"***
- *Sentence → NP, VP*
- *NP → determiner, adjective, noun*
- *VP → verb*
- *Determiner → the*
- *Adjective → big*
- *Noun →man*
- *Verb → cried*

- ***Domains***
- *S= symbol*
- *Slist=s**
- ***Predicates***
- *Sentence(slist)*
- *NP(slist)*
- *VP(slist)*
- *Append(slist)*
- *Determiner(slist)*
- *Noun(slist)*
- *Adjective(slist)*
- *Verb(slist)*

- ***Clauses***
- *Determiner([the])*
- *Noun([man])*
- *Adjective([big])*
- *Verb([cried])*
- *Sentence(S):- append(S1,S2,S), NP(S1), VP(S2).*
- *NP(S):-append(S1,S2,S),append(S3,S4,S1),determiner(S2), adjective(S3),noun(S4).*
- *VP(S):- verb(S).*
- *Append ([ ], L , L).*
- *Append ([H | T], L, [H|T1]):- append (T,L,T1).*

- ***Goal***
- *Sentence([the,big,man,cried]).*

SINGULAR AND PLURAL

- *For the following sentences,*

  The careless inspector falls down the stairs.

  The careless inspectors fall down the stairs.

- There is a problem; however, it is number agreement. What this mean is, if the subject of a sentence is singular, so must be the verb, if the subject of the sentence is plural, the verb must be plural also.

- There is a standard solution to the number a agreement problem and to other similar but more obscure agreement problems. ***It is to make two sets of all rules one for singular situation and one for plural situations. This can be*** done efficiently by introducing a second argument into all the prolog rules stemming from the grammar. The argument is a symbol and will take on only the values ***singular or plural.***

EXAMPLE:

- *sentence (S) :- append (S1,S2,S), NP(S1), VP(S2).*
- We can write this rule with another argument that carries number information

- *Sentence(S,X):- append (S1,S2,S), NP(S1,X), VP(S2,X).*

- The variable X could be bound to any symbol, but the rule actually will be used in only two ways:

- *Sentence(S, singular) :- append(S1,S2,S),NP(S1,singular),VP(S2,singular).*
- *Sentence(S, plural):-*
- *append (S1,S2,S),NP(S1,plural),VP(S2,plural).*
- ***The VP will be interpreted as follow:***
- *VP(S,X):- append (S1,S2,S), verb(S1,X),adverb(S2,_).*

- ***Verbs and nouns must generally be cataloged in both a singular and plural form, for example:***

- *Verb ([thinks],singular)*
- *Verb([think],plural).*
- *Noun([doctor],singular).*
- *Noun([doctors],plural).*

- ***For determiner,***
- *Determiner([that],singular).*
- *Determiner([those],plural).*
- *Determiner([the],_).*
- *Determiner([a],_).*

DIFFERENCE PAIR IDEA

- There is a disadvantage to writing prolog rules with a heavy dependence on the append predicate. Append is fairly inefficient. In every case it will try many candidates splitting before it discovers one that is effective. Grammars rules written this way may run between five and fifty times slower than if we encode the rules with a different method, which is called ***difference pair parsing.***
- Difference pair parsing adding stills another argument to all the prolog
- rules just as we needed to do to get a agreement between subject and object.

- Every predicate that use the difference pair idea has two lists as arguments.
- The first list is input, and the second, which is shorter, is output. Take a simple example. The meaning of
- NP(S0,S).
- In this: for the predicates to succeed, it must be possible to examine the whole list S0 and found a noun phrase at the front of it. If the noun phrase is striped off the list S is what would be left. For istance this use of the predicate will succeed:
- NP([the, big, politician, talked, slowly],[talked, slowly]).

- Consider the following definition of the NP:
- NP $\longrightarrow$ determiner, adjective, noun
- ***Using append predicate it will be:***
- *NP(S):- append(S1,S2,S),append(S3,S4,S1),determiner(S2),adjective(S3),noun(S4).*

- ***Using the difference pair idea it will be:***
- *NP(S0,S):- determiner (S0,S1), backnp1(S1,S).*
- *Backnp1(S0,S):- Adjective(S0,S1),backnp2(S1,S).*
- *Backnp2(S0,S):- noun(S0,S).*

- ***For the lowest level entities, word, should be defined as follow:***

Noun([man|S]),S).

Verb([reads|S],S).

- ***At the highest level there is a rule that will look like this:***

Sentence (S0,S) :- NP(S0,S1), backsen1(S1,S).

- The goal predicates for any sentence must carry an empty list as the second argument since all the word in the first list will be stripped off.
- For the following sentence “ the young engineer smiled” , will be as follow
- *Sentence ([the, young, engineer, smiled],[ ]).*

THE DOCUMENT INFORMATION SYSTEM

- Programs that trigger of keywords have some functions that is supported by natural language in a simple way. This function could be many different things. Documentation information system is a skimmer program that forms a rudimentary expert system that is very useful.
- The problem to solve is this: many big company produce manuals about their products by the ton. This is particularly are a problem in the computer industry where each software product has to have its own manual which need frequent revisions.
- The idea behind (DOCSYS) is simple. The indexes from all the manuals of interest will be known to the program. Each index citation just becomes a data item for DOCSYS. User then can ask such question as:

- "Tell me about EDLIN" or "do you have any information about EDLIN".

The system will determine that EDLIN is what is being asking about, it find out all references to EDLIN and return them with suitable syntactic.

- for example:
- *"EDLIN can found in reference BOOKS1 on pages ["41-46","57"]*
- *"EDLIN can found in reference BOOKS1 on pages ["100-102"]*
- *"EDLIN can found in reference BOOKS1 on pages ["100","110"]*

DATA STRUCTURE OF DOCSYS

- The basic data are the index entries. In this simple program they consist of predicate name, *ind, with the argument bearing the information.*
- The first argument is the name of the publication. The second argument is the name of the topic. And the third is a list of page references indicating where the topic can be found. The first two arguments in *ind are strings,* and the third is a list of strings.

- *Ind (“BOOK1”,”LOGOFF”,[“99”]).*
- *Ind (“BOOK1”,EDLIN”,[“41-46”,”57”]).*

- Frequently when the user asks about a particular term, it will be talked about in a way that is different from the exact index citation. It is useful to maintain a stock of synonyms that the system will recognize.
- For instance, there is only one index entry for LOGOFF, but since a user might try to access this information under LOGOUT.

- *Dsyn("LOGOUT","LOGOFF")*.
- *Dsyn("LOGIN","LOGON")*

- The last data item used by the system is a group of words commonly occurring in questions but that have nothing to do with the keyword strategy. They should be ignored.

- *Reject(“HOW”).*
- *Reject(“GO”).*
- *Reject(“WHO”).*

PROGRAMMING STRATEGY USED IN DOCSYS

The highest level predicate. It starts off all the work that the system can do:

- *Docdriver: - repeat, nl, getquery(X),findref(X,Y),produceans(Y),fail.*

- All the work of this system takes place between the repeat and fail predicates.
- The *getquery predicate prompts the user for his question and brings it* back as a list of symbols bound to the predicate’s argument.

- *Getquery(Z):-write(“please ask your question.”),*
- *nl,readln(Y),upper_lower(Y1,Y),changeform(Y1,Z).*

- Upper-lower change string to all uppercase letters. This make program insensitive to capitalization. *Changeform change the query to a list of* symbols.

- *Changeform(S,[H|T]):-*
  fronttoken(S,H,S1),!,changeform(S1,T).
- *Changform(”“,[ ]).*

- The first argument is string and the second argument is the equivalent list of symbol.
- After getquery does its work, the next thing is to isolate the keyword that the query is relay about. By use *findref*.

- *Findref(X,Y):-memberof(Y,X),not(reject(Y)),!.*

- After isolate the proper keyword, we pass it on to *produceans predicate. It* has the responsibility for constituting and returning an answer to the user.

- *Produceans(X):-ind(X1,X,Y),putflag,*

write(X,”can be found in the refrence”,X1,”on pages” ,Y,”.”),nl.

- *produceans(X):-syn(X,Z),ind(X1,Z,Y),putflag, write(X,”can be found in the refrence”,X1,”on pages” ,Y,”.”),nl.*
- *Produceans(_):-not(flag), write(”we have no information on that”),nl.*
- *Produceans(_):-remflag.*

- The first two clauses setup answers to queries, and the third clause writes a message that says no answer.
- Put flag predicate used to put a *flag in the data base when either the first* two rules succeed. This means that the answer was found. The third clause tests the flag to decide if it should print “nothing found” message.
- The last clause removes the flag from the database so that it dose not interfere with the operation of second query.

- *Putflag:- not(flag),assert(flag),!.*
- *Putflag.*
- *Remflag:- flag,retract(flag),!.*
- *Remflag.*

- Finally, the following rules treat the synonyms. One of the rules is for the purpose of stating that if A is synonym of B. then B is synonym of A.
- there is also rules that force the system to recognize predictable singular and plural forms of words as being synonyms of each other.

- *Syn(X,Y):-dsyn(X,Y).*
- *Syn(Y,X):-dsyn(Y,X).*
- *Dsyn(Y,X):-concat(X, ”S”, Y).*
- *Dsyn(Y,X):-concat(X, ”ES”, Y).*
- *Dsyn(Y,X):-concat(X, ”’S”, Y).*

ANALYZING THE SEMANTIC STRUCTURE OF THE SENTENCES

- The most powerful strategy for understanding the meaning of the sentences is thematic analysis case grammars. Sentences can be classified into two types,

1. object oriented sentences: deal solely with description of object, for example,

“The roof is old”, it is simple to analyze for meaning because all they ever do is state that a property has become associated with an object.

2. action sentences: deal with actions that objects can undergo or participate in:

“ the old roof leaked”

“ the rain pelted the old roof”

- There are many cases:
- 1. Object case: is the noun group that receives the action of the
- verb.
- “The realtor and his assistant in inspected <u>a house </u>for their
- client”
- 2. Agent case: is the entity that applies the action to the object.
- “<u>The realtor </u>and his assistant in inspected a house for their
- client”
- 3. co-agent case: identifies another agent how shares in applying
- the action, it is possessive noun followed by noun,
- “The realtor and <u>his assistant </u>in inspected a house for their
- client”
- 4. Beneficiary case: the entity, whose behalf the action in the
- sentence was performed,
- “The realtor and his assistant in inspected a house <u>for</u> <u>their</u>
- <u>client”</u>
- 5. Location case: express where the action take place.
- “In the school”.

- 6. Time case: express when the action took place,” at 5 o’clock” .
- 7. Instrument case: identify something used by the agent to apply
- the action carried by the verb,” with a knife”.
- 8. Source and destination cases: its describe the movement from
- one place to another, “the dog cashed the insurance agent <u>out of</u>
- <u>the yard</u> and into <u>his car</u>”.
- 9. trajectory case: it describes the path over which action occurred,”
- the man drove in his car <u>through the woods</u> to his next client”.
- 10. Conveyance case: describe action occurs in some kind of
- carrier, “in the car”.
- All these cases work with noun group.

THE CASE ANALYSIS PARSER

- To set up a case grammar, one must inspect typical sentences that will be of interest, for instance, in simple action sentences with active verbs, there are form like this:
- ***He hit the ball.***
- ***The big man hit the ball.***
- ***John hit the ball with the bat.***
- ***A boy hit the ball through the window during the game.***

- We can easily write down the grammar (the grammar are just rules that describe something that is complex in terms of a catalog of more basic parts) that describe what happing in these sentence using thematic analysis.

- *Sentence → agent, verb1, object*
- *Sentence → agent, verb1, object, instrument*
- *Sentence → agent, verb1, object, trajectory, time*

- *Agent→ pronoun*
- *Agent→ proper-noun*
- *Agent→ noun-phrase*

- *Object→ pronoun*
- *Object→ proper-noun*
- *Object→ noun-phrase*

- *Instrument → with, noun-phrase*

- *Trajectory → through, noun-phrase*

- *Time → during, noun-phrase*
- *Time →before, noun-phrase*
- *Time → after, noun-phrase*

- *Pronoun → he, ect*

- *Noun-phrase → determiner, noun*
- *Noun-phrase → determiner, adjective, noun*

- *Adjective → big*
- *verb1 → hit*
- *verb1 → made, ect*

- This is the type of grammar that grows fro this kind of sentence analysis.
- A hundred structures rules are required to determine the semantic meaning variation way in simple sentences. They will have to be collected to find the best one. This will be arising when it is not possible to completely to classify the aspect of the verb from the verb by itself.

For the above example,

When the verb is "hit" what is the aspect of this verb?

aspect	example
First person singular	I hit
Second person singular	You hit
Third person singular	He hit
First person plural	We hit

These difficulties would be resolved using closer inspection of the agent each of candidate parsing. So the aspect of the verb will match the aspect of agent.

ALL THE RULES FOR THE SENTENCES CAN BE CATEGORIZED USING DIFFERENCE PAIR IDEA:

- *Sentence(S,S0) :- agent (S,S1), backparta (S1,S0).*
- *Backparta (S,S0):- verb (S,S1), object (S1,S0).*
- *Sentence(S,S0):- agent(S,S1),backpartb(S1,S0).*
- *Sentence(S,S0):- verb(S,S1),backpartc(S1,S0).*
- *Backpartc(S,S0):-object(S,S1), instrument(S1,S0).*
- *Sentence(S,S0):-agent(S,S1),backpartd(S1,S0).*
- *Backpartd(S,S0):- verb(S,S1),backparte(S1,S).*
- *Backparte(S,S0):-object(S,S1),backpartf(S1,S).*
- *Backpartf(S,S0):- trajectory(S,S1),time(S1,S0).*
- The lower level rules would all be set up in a similar manner.

MACHINE TRANSLATION

Rule Based Machine Translation

DEFINITION OF (MT):

The use of computers to automate <u>*some*</u> *or* <u>*all*</u> *of the process of* translating from one language to another.

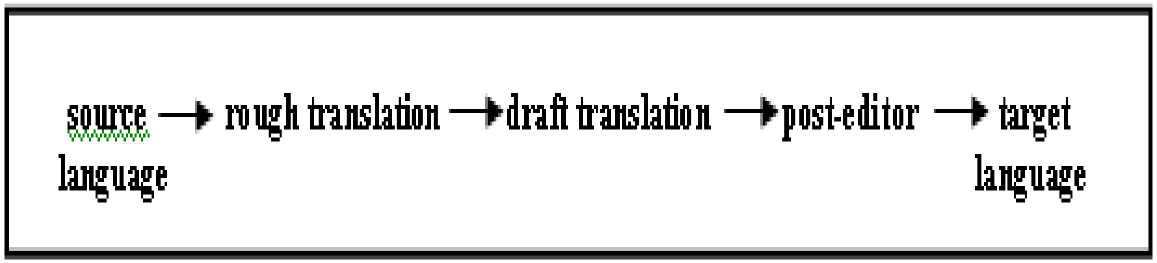
IN GENERAL, WE NEED TRANSLATION FOR MANY PURPOSES:

- 1. To read documents of many applications written in different languages which the reader does not know that language.
- 2. To increase scientific & technical information to be transferred.
- 3. The need for military, economically, trading … etc. information.
- 4. Scientific researches.
- 5. To provide fast translator.

- Translation, in its full generality, is a difficult, fascinating and intensely human endeavor, as rich as any other area of human creativity

- Translation requires a deep and rich understanding of the source language and the input text, and sophisticated, poetic, and creative command of the target language.

- ***The problem of (automatically producing a high-quality translation of an arbitrary text from one language to another)*** *is far too hard to automate* completely.
- But certain simpler translation tasks can be addressed with current computational models. In particular machine translation systems often focus on:
- **1. Tasks for which a rough translation is used.**
- **2. Tasks where a human post-editor can be used to improve MT output.**


source language
rough translation
draft translation
post-editor
target language

- This is called “Computer-Aided Human Translation”, (CAHT), rather than (fully automatic) machine translation.
- This model of MT usage is effective especially for high volume jobs and those requiring quick turn-around:
- • The translation of S/W manuals for localization to reach new markets.
- • The translation of market-moving financial news, for example from Japanese To English for use by stock traders.

- **3. Tasks limited to small sublanguage domains in which fully automatic high-quality translation is achievable.**
- The following domains have a limited vocabulary and only a few basic phrase types:-

LIMITED VOCABULARY

- 1.Weather forecasting is an example of a sublanguage domain that can be modeled completely enough to use raw MT output even without postediting.
- 1- Weather forecasting consists of phrases like:
- -Cloudy today and Thursday,
- -Low tonight 4,
- -Outlook for Friday: Sunny.
- 2. Equipment maintenance manuals.
- 3.Air travel queries.
- 4. Appointment scheduling.
- 5.Restaurant recommendations.

LANGUAGE SIMILARITIES & DIFFERENCES:-

- Even when language differ, these differences often have systematic structure. ***The study of systematic cross-linguistic similarities and differences is called Topology.***
- Machine Translation in that the difficulty of translating form one language to another depends a great deal on how similar the languages are in their vocabulary, grammar, and conceptual structure.

- **1. Syntactically, languages are different in the basic word order of verb, subject, and object in simple declarative clauses.**
- - English, French, German: are all SVO languages, meaning that verb tends to come between the subject & object.
- -Hindi and Japanese: are SOV languages, meaning that the verb tends to come at the end of clauses.
- -Arabic: is VSO language.
- Similarities:-
- SVO languages have prepositions.
- SOV languages have postposition.

- **2. sometimes, rather than a single word, there is a fixed phrase in the target language.**
- نشأ (Arabic) grow up (English)
- Informatique (French) computer science (English)

3. Overlap problem: One language may divide up a particular conceptual domain in more detail than other

English –to- Japanese

brother otooto (younger)

oniisan (older)

German

wall wand (inside)

mauer (outside)

- The way that language differ in lexically dividing up conceptual space may be more complex than this one-to-many translator problem, leading to many-to-many mapping.

- **4. One language may have a (lexical gap), where no word or phrase can express the meaning of a word in the other language.**

- As a conclusion, there can be no perfect translation, since speakers of the source and target languages necessarily have different conceptual systems.

1. DIRECT TRANSLATION

- In **direct translation, we proceed word-by-word through the source** language text translating each word as we go. We make use of no intermediate structures, except for shallow morphological analysis; each source word is directly mapped onto some target word. Direct translation is thus based on a large bilingual dictionary; each entry in the dictionary can be viewed as a small program whose job is to translate one word.
- After the words are translated, simple reordering rules can apply, for example for moving adjectives after nouns when translating from English to French.

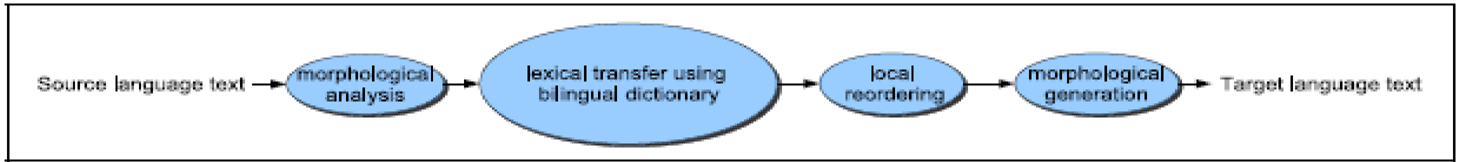


Figure Direct machine translation. The major component, indicated by size here, is the bilingual dictionary.

DIRECT TRANSLATIN EXAMPLE

- Let's look at a simplified direct system on our first example,

step 1 translating from English into Spanish:

Mary didn't slap the green witch

Maria	***no***	***dio***	***una***	***bofetada***	***a***	***la***	***bruja***	***verde***
Mary	**not**	**gave**	**a**	**slap**	**to**	**the**	**witch**	**green**

- Step 2 presumes that the bilingual dictionary has the phrase *dar una*
- *bofetada a* as the Spanish translation of English *slap.*

- The local reordering step 3 would need to switch the adjective-noun ordering from *green witch* to *bruja verde*. And some combination of ordering rules and the dictionary would deal with the negation and past tense in English *didn't*. While the direct approach can deal with our simple Spanish example, and can handle single-word reordering, it has no parsing component or indeed any knowledge about phrasing or grammatical structure in the source or target language.

- It thus cannot reliably handle longer-distance reordering, or those involving phrases or larger structures.
- This can happen even in languages very similar to English, like German, where adverbs like *heute* ('today') occur in different places, and the subject (e.g., *die griine Hexe)* can occur after the main verb.

- ***Input: Mary didn’t slap the green witch***
- ***After 1: Morphology*** Mary DO-PAST not slap the green witch
- ***After 2: Lexical Transfer*** Maria PAST no dar una bofetada a la verde bruja
- ***After 3: Local reordering*** Maria no dar PAST una bofetada a la bruja *verde*
- ***After 4: Morphology*** Maria no di´o una bofetada a la bruja verde
-
- Finally, even more complex reorderings occur when we translate from SVO to SOV languages, as we see in the English-Japanese example ,

He adores listening to music

kare ha ongaku wo kiku no ga daisuki desu

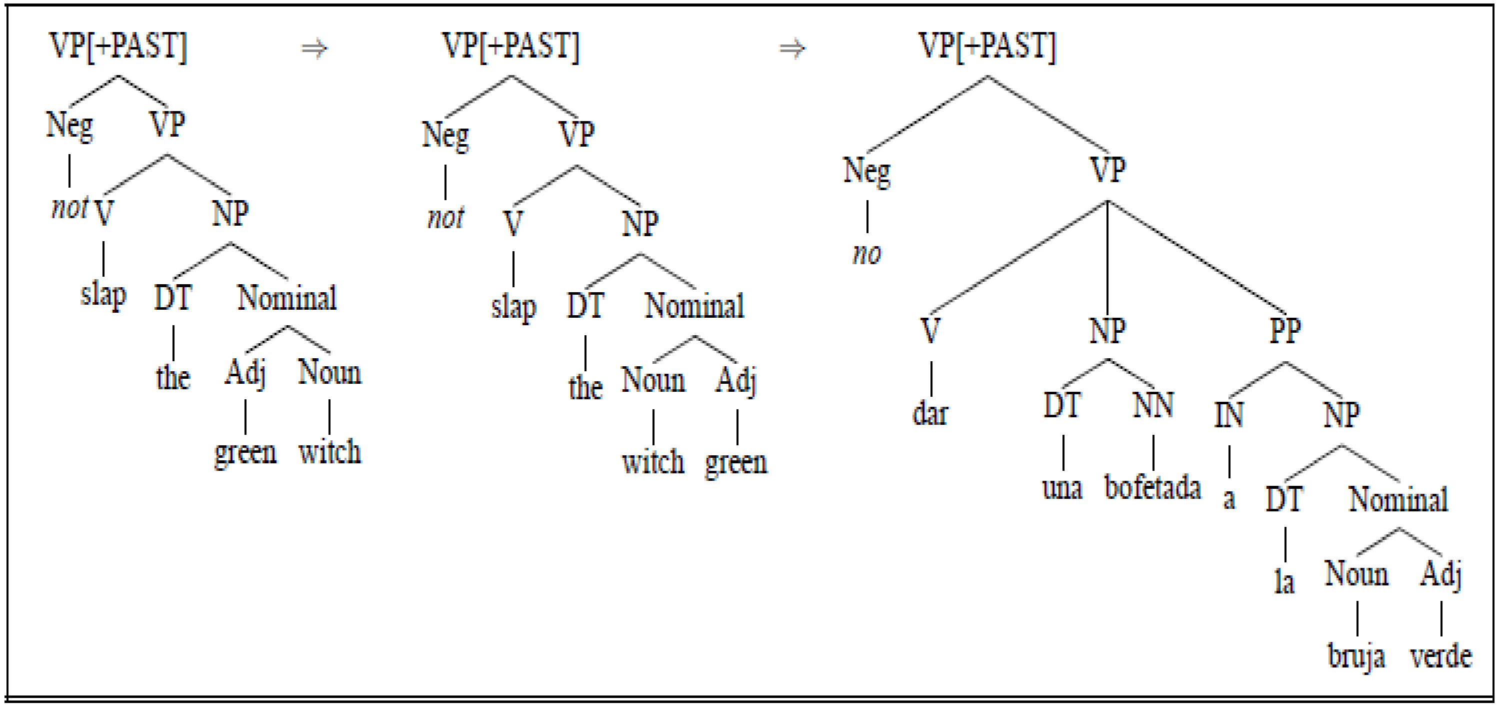
- suggest that the direct approach is too focused on individual words, and t*he music to listening adores* ,These examples hat in order to deal with real examples we'll need to add phrasal and structural knowledge into our MT models.

2. TRANSFER

- languages differ systematically in structural ways. One strategy for doing MT is to translate by a process of overcoming these differences, altering the structure of the input to make it conform to the rules of the target language. This can be done by applying **contrastive knowledge,** that is, knowledge about the differences between the two languages. Systems that use this strategy are said to be based on the **transfer model.**
- The transfer model presupposes a parse of the source language, and is followed by a generation phase to actually create the output sentence.
- Thus, on this model, MT involves three phases: **analysis, transfer,** and **generation,** where transfer bridges the gap between the output of the source language parser and the input to the target language generator. It is worth noting that a parse for MT may differ from parses required for other purposes.
- For example, suppose we need to translate *John saw the girl with the binoculars* into French. The parser does not need to bother to figure out here the prepositional phrase attaches, because both possibilities lead to the same French sentence. Once we have parsed the source language, we'll need rules for **syntactic transfer** and **lexical transfer**. The syntactic transfer rules will tell us how to modify the source parse tree to resemble the target parse tree.

$$\begin{tikzcd}
 & \text{Nominal} \arrow[dl] \arrow[dr] & \Rightarrow & & \text{Nominal} \arrow[dl] \arrow[dr] & \\
\text{Adj} & & \text{Noun} & {} & & \text{Adj}
\end{tikzcd}$$

A simple transformation that reorders adjectives and nouns (English to Spanish)

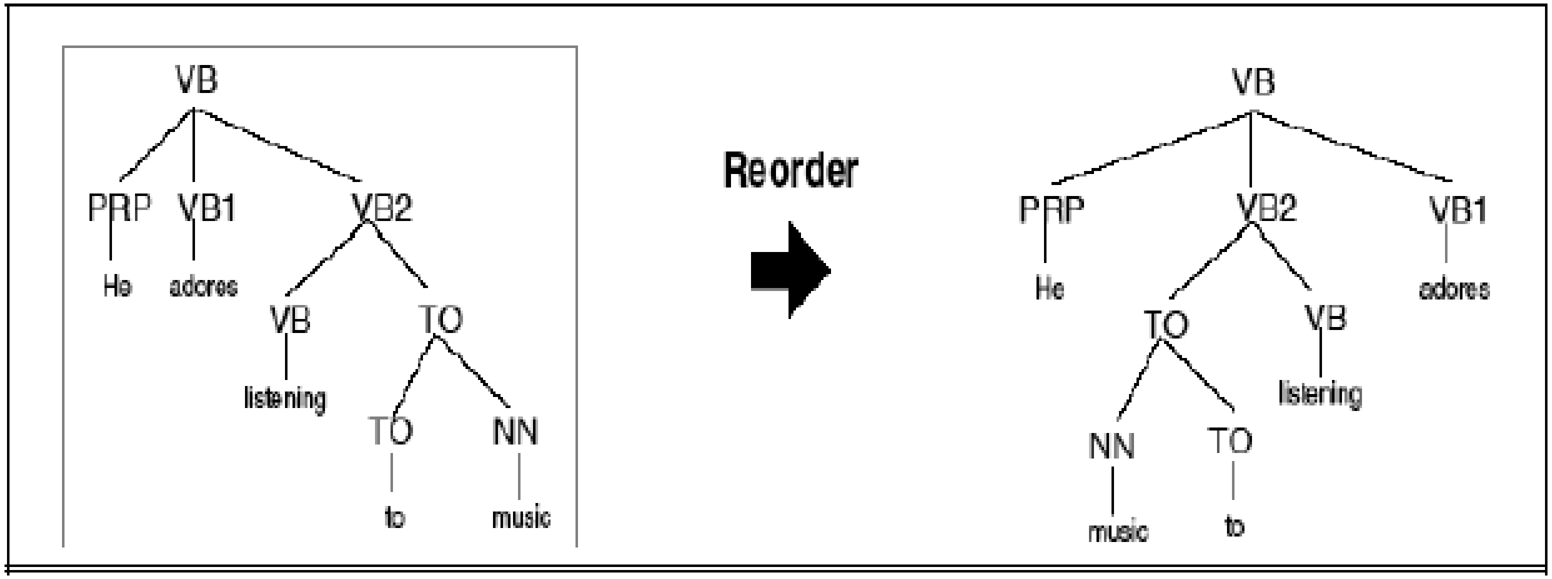

VP[+PAST]
⇒
VP[+PAST]
⇒
VP[+PAST]
Neg
VP
not
V
NP
slap
DT
Nominal
the
Adj
Noun
green
witch
Neg
VP
not
V
NP
slap
DT
Nominal
the
Noun
Adj
witch
green
Neg
VP
no
V
NP
PP
dar
DT
NN
una
bofetada
IN
NP
a
DT
Nominal
la
Noun
Adj
bruja
verde

- The transfer approach and this rule can be applied to our example *Mary did not slap the green witch*. Besides this transformation rule, we'll need to assume that the morphological processing figures out that *didn 't* is composed of *do-PAST plus not*, and that the parser attaches the PAST feature onto the VP. Lexical transfer, via lookup in the bilingual dictionary, will then remove *do*, change *not* to *no*, and turn *slap* into the phrase *dar una bofetada a*, with a slight rearrangement of the parse tree .

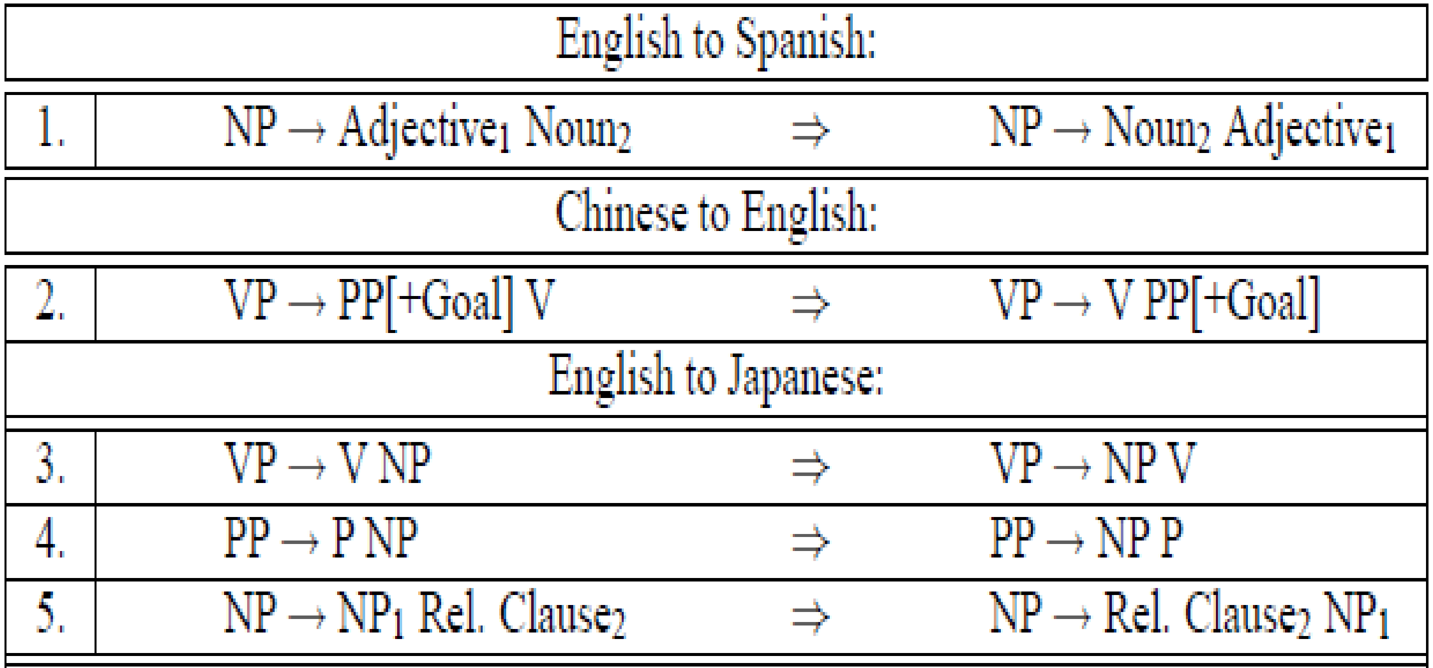
He adores listening to music

kare ha ongaku wo kiku no ga daisuki desu

he music to listening adores


VB
PRP
VB1
VB2
He
adores
VB
TO
listening
TO
NN
to
music
Reorder
VB
PRP
VB2
VB1
He
TO
VB
adores
listening
NN
TO
music
to

AN INFORMAL DESCRIPTION OF SOME TRANSFORMATIONS



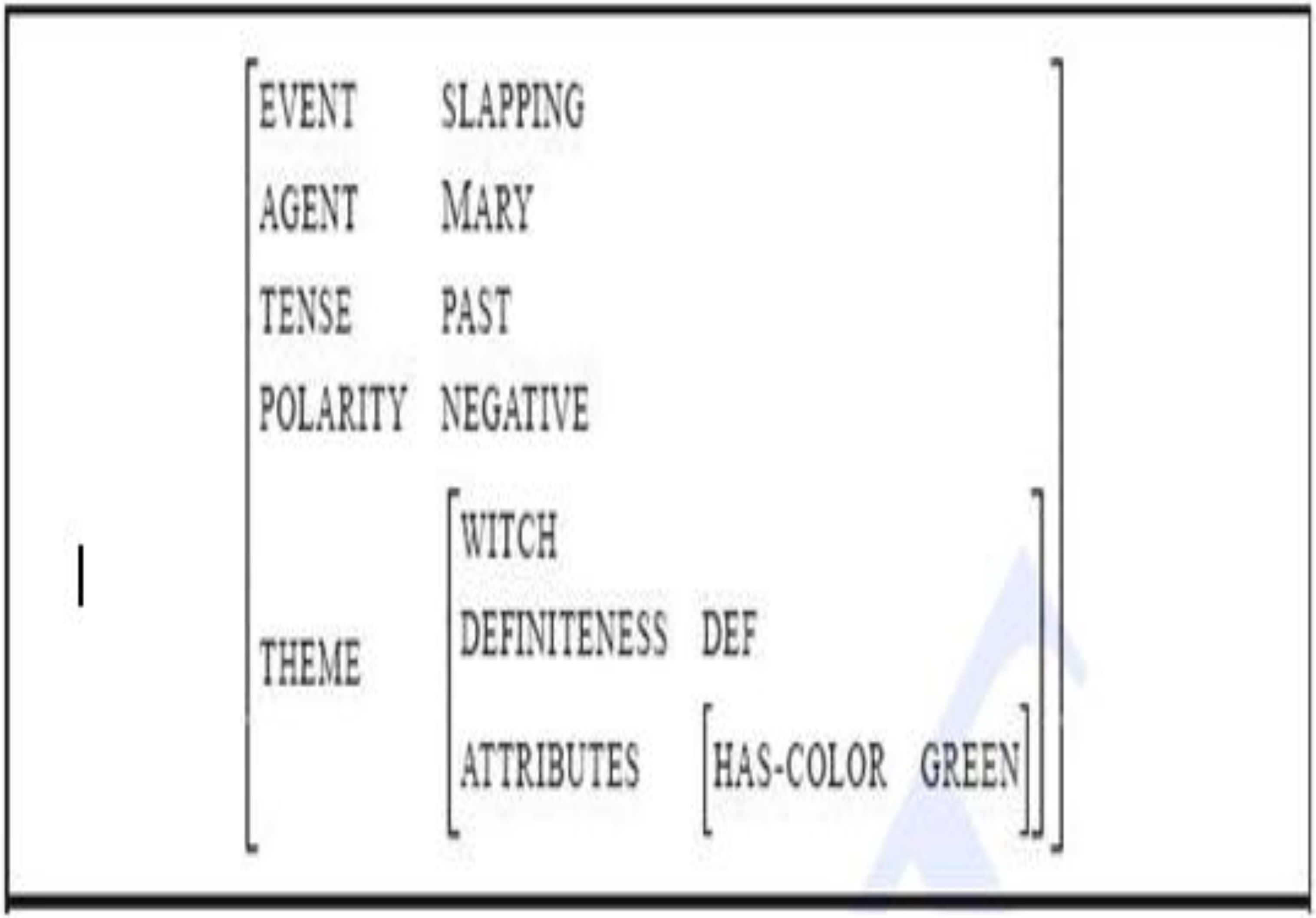
	English to Spanish:		
1.	NP → $Adjective_1$ $Noun_2$	⇒	NP → $Noun_2$ $Adjective_1$
	Chinese to English:		
2.	VP → PP[+Goal] V	⇒	VP → V PP[+Goal]
	English to Japanese:		
3.	VP → V NP	⇒	VP → NP V
4.	PP → P NP	⇒	PP → NP P
5.	NP → NP_1 Rel. $Clause_2$	⇒	NP → Rel. $Clause_2$ NP_1

Figure 24.13 An informal description of some transformations.

3. THE INTERLINGUA IDEA: USING MEANING

- One problem with the transfer model is that it requires a distinct set of transfer rules for each pair of languages. This is clearly suboptimal for translation systems employed in many-to-many multilingual environments like the European Union. This suggests a different perspective on the nature of translation. Instead of directly transforming the words of the source language sentence into the target language, the interlingua intuition is to treat translation as a process of extracting the meaning of the input and then expressing that meaning in the target language. If this could be done, an MT

- system could do without contrastive knowledge, merely relying on the same syntactic and semantic rules used by a standard interpreter and generator for the language.
- The amount of knowledge needed would then be proportional to the number of languages the system handles, rather than to the square.
- The idea is for the interlingua to represent all sentences that mean the "same" thing in the same way, regardless of the language they happen to be in. Translation in this model proceeds by performing a deep semantic analysis on the input from language X into the interlingual representation and generating from the interlingua to language Y.
- Semantic decomposition into some kind of atomic semantic primitives is the approach used to implement Interlingua idea.



$$\begin{bmatrix} \text{EVENT} & \text{SLAPPING} \\ \text{AGENT} & \text{MARY} \\ \text{TENSE} & \text{PAST} \\ \text{POLARITY} & \text{NEGATIVE} \\ \text{THEME} & \begin{bmatrix} \text{WITCH} & \\ \text{DEFINITENESS} & \text{DEF} \\ \text{ATTRIBUTES} & \begin{bmatrix} \text{HAS-COLOR} & \text{GREEN} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

SPEECH RECOGNITION

SPEECH RECOGNITION

- **1. Understanding Spoken Language**

  Speech: consists of a continuously varying sound wave which links speaker to listener. Sound requires a medium through which to travel and the most usual medium is air.

- The process of extracting the information content from the spoken word is a formidable task from the point of view of the computer and borders on the area of artificial intelligence.

SPEECH RECOGNITION PROBLEMS

- **There are several major problem areas in speech recognition:**

1. Continuous speech has to be segmented in order to obtain the correct information.

2. Speech patterns vary not only between speakers but also within an

individual speaker, even when identical words are spoken.

3. A word an vary in loudness, pith, stress and pronunciation rate.

4. The geographical origin of the speaker is also an important factor

when words are pronounced.

5. Different words can sound very similar.

6. Background noise and other interference can distort the original

signal.

WORD RECOGNITION

- There have been two major drives at speech recognition; the simplest one being ***speaker-dependent isolated word recognition*** and the other being ***speaker-independent isolated word recognition***. Isolated word recognition nicely sidesteps the problem of correctly segmenting continuous speech, by demanding that appropriate pauses are inserted between each word spoken.

Speaker-dependent recognition, on the other hand, helps to overcome such Problems as regional, accent, the sex of the speaker, etc.

SPEAKER-INDEPENDENT RECOGNITION

- Speaker-independent recognition involves converting the spoken word into an electrical signal of some. Sort via a microphone the signal is the processed further; to obtain a set of identifying features which are then compared with those held in the computer's vocabulary: The vocabulary will consist of a set of reference templates, which have been chosen to represent the average speaker, or speakers with similar accents. Dynamic speaker adaptation is also a viable means for effective speech recognition. However, the vocabulary size (i.e. number of recognizable words/ utterances) in Speaker- independent recognition systems is much smaller than in speaker-dependent systems.

- ***Speaker-dependent isolated-word recognition*** involves training the computer to recognize words by getting the speaker repeatedly to say certain words. From the results of the initial training the computer is able to formulate an average template for individual words, which are then used for reference. Obviously, the more training a machine receives, the better it becomes at choosing the right word(s). Each word must be spoken separately, as opposed to continuously, which is the case in normal speech. The problem here is that the user may not always be prepared to help the machine in this way, since the process can be very time-consuming.
- In both cases (*speaker-dependent and speaker-independent recognition*) some degree of pattern matching is necessary in order to recognize a word. The system compares the incoming signal with a stored template, thereby generating some sort of score on the basis of the degree of similarity. The template with the highest score is then chosen.

- The problems which arise with both speaker-dependent and speaker- independent systems *are that words will not be pronounced at the same rate by all speakers*. This lead to non-linear distortion of the incoming template compared to the stored template. Consequently, some sort of algorithm is needed to match the incoming template with the stored template, either by shortening or lengthening them accordingly. One such mathematical method is known as ***Dynamic Time Warping D.T.W*** where the incoming template is warped in order to align it with the stored a score is obtained, on the basis of which the best alternative template): (template) is then selected. By applying various grammatical rules the number of templates tested can be reduced significantly, with a consequent reduction in processing time. An alternative approach, adopted by low-cost systems, is to alter both incoming and stored templates to fit a standard length.
- Many of the tasks required for voice recognition systems can be handled adequately by isolated word recognition, but continuous speech recognition is much faster and more satisfying to the user, since this is the most natural and efficient way of communication. However, continuous speech recognition systems are much more difficult to implement than isolated word recognition systems. In the former case, the incoming sound wave is usually segmented into small units such as phonemes, diphones, syllables, etc.;

2. BASIC STEP OF SPEECH RECOGNITION

1. Sampling.
2. Detection of start and end of incoming signal.
3. Calculations of speech spectra directly from digitization, or indirectly through filters.
4. Pitch contour evaluation
5. Segmentation.
6. Word detection, either by applying 'bottom-up' techniques to associate phonemic features with the required lexicon, or 'top-down' techniques in order to identify complete units within the template database.
7. Responding to a message.

2.1 SAMPLING

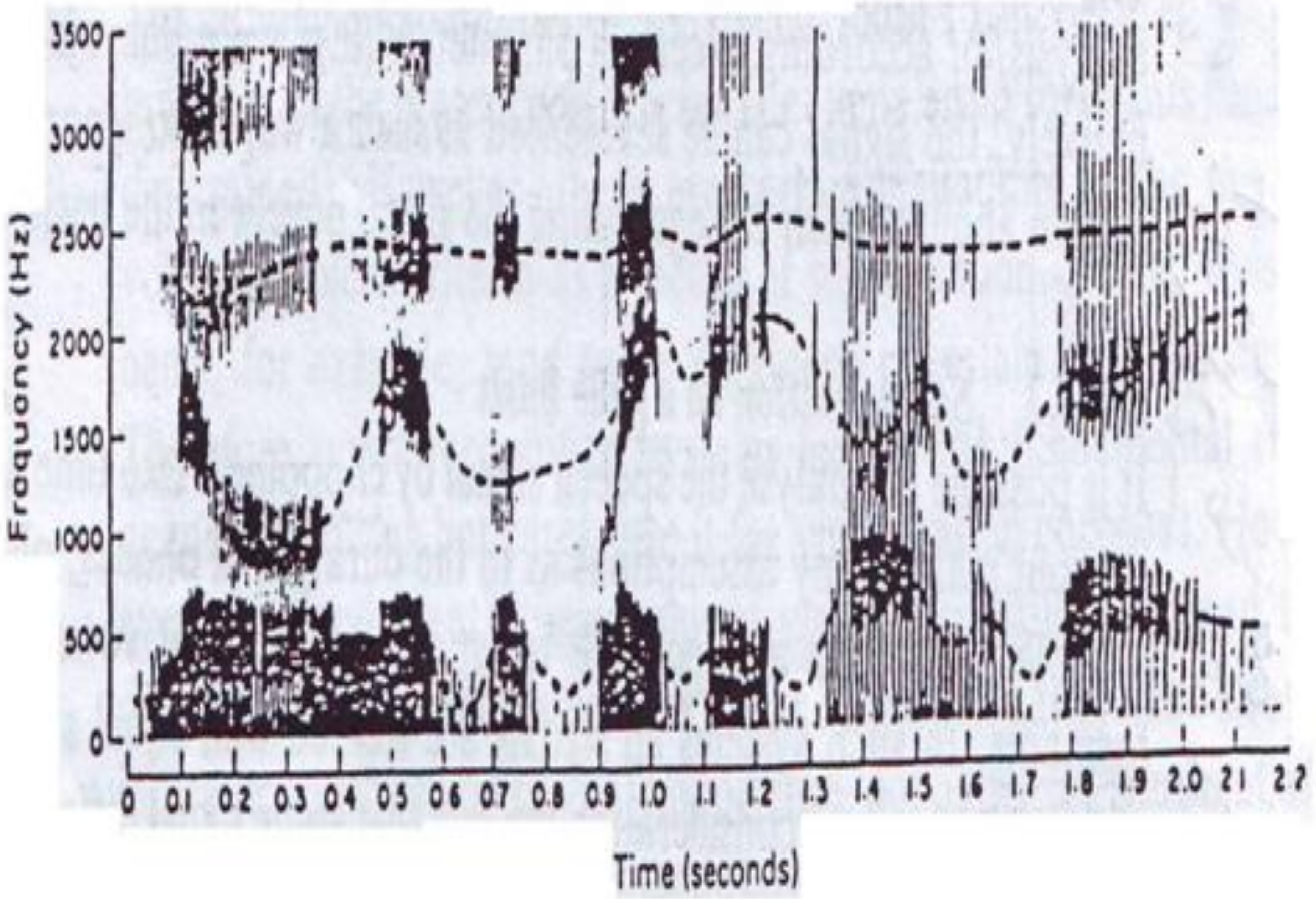
- The speech signal is first sampled and then digitized using an analogue to digital converter. The result from this stage is then stored in the computer.
- Having obtained the digital representation of the incoming signal, various parametric features can then be obtained. These parametric features then form the basis for the segmentation of the speech signal. Further, to avoid distortion of the incoming signal by background noise, it is desirable to have the microphone as close as possible to the speaker, if it is not possible to operate in a noise-free environment.

.2 DETECTION OF START AND END OF INCOMING SIGNAL

- The signal processor remains in a ‘waiting’ state when no voice signals high enough to be detected are coming from the receiver (*microphone*).
- Variation in this silent environment is automatically detected by the program in execution and the speech recognition process is activated. A similar principle applies when the end of an incoming signals Detected, that is, when there i s a variation from a signals silence, and the Signal is not of sufficient strength to energize the filters; the speech recognition program recognizes this as the end of the utterance.

2.3 CALCULATION OF SPEECH SPECTRA

- A sound spectrograph utilizes a special frequency analyzer to produce a three-dimensional plot of the variation in the speech energy spectrum with time.



3500
3000
2500
2000
1500
1000
500
0
Fraquancy (Hz)
0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0 1.1 1.2 1.3 1.4 1.5 1.6 1.7 1.8 1.9 2.0 2.1 2.2
Time (seconds)

2.4 PITCH CONTOUR EVALUATION

- The fundamental frequency of voicing is of use for determining stress and intonation. The contours of the fundamental frequency (as it rises and falls) can also be used as an indication of major syntactic boundaries. Additionally, stress patterns, rhythm and intonation carry some clues as to the Phonetic identify of some speech sounds.

2.5 SEGMENTATION/IDENTIFICATION OF PHONEMIC FEATURES

- Segmentation can be done either on the basis of some form of phonetic Analysis, or according to certain parameters (e.g. voiced/unvoiced). Alternatively, the signal can be segmented in such a way as to ignore anything known about speech by segmenting the signal purely on the basis of time.

2.5.1 SEGMENTATION ON A TIME BASIS

- It is possible to analyze speech signal by chopping into time segments without making any assumptions as to the duration of phones, phonemes, syllables, etc) It is also possible to ignore the fact that vowels produce formants. In such systems all signals are treated with equal significance, without taking into consideration the fact that some signal differences may be more important than others. [1]The obvious drawback of this approach is the large amount of storage space required. In addition, [2]several different templates may have to be stored for each word in order to take into account the differences which are a direct result of the varying speech rates of individual speakers, the channel noise, etc. Thus, some form of speech processing is desirable to eliminate redundant information.

SOME OF THE TECHNIQUES TO ELIMINATE REDUNDANT INFORMATION INCLUDE:

- **(a)** Warping the incoming signal to match certain features of the stored Templates. The individual segments for each unit can then be compared with the corresponding units in the stored template. Each segment will be compressed/stretched, as appropriate, until it matches. The amount of distortion required for each segment is then recorded. Finally, the template requiring the least distortion will be selected.

- **(b)** Detecting certain features of the incoming signal, so that they can be used to adjust the signal to a closer match with the stored template. For example, the fundamental frequency of the incoming signal can be determined and the difference this makes to the stored template can be taken into account. Therefore, irrelevant differences between the stored template and the incoming signal can be reduced.

2.5.3 SEGMENTATION ON A PHONETIC BASIS

- Segmentation can take place at the level of allophone, phoneme; diphone, syllable or word.

- **Allophone**

  Certain allophones can provide word or syllable boundary information which can be very useful in recognition systems. For example, some allophonic variations occur only at the end or beginning of words/syllables.

- **Phoneme**

The phoneme represents the smallest number of distinctive phonological classes for recognition and is substantially less than the number of allophones, syllables, words, etc. (there are approximately 43 phonemes in the English language). Although the number of phonemes is small, their automatic recognition by a computer system is still a major problem, since there are no acoustically well-defined patterns or templates for phonemes.

DIPHONE

- The term diphone is used to represent a vowel-consonant sequence, such that the segment is taken from the centre of the vowel to the centre of the consonant; the segment is often referred to as ***transeme***.
- One of the advantages of considering the diphone is that it includes transitional information which is necessary for recognition in many cases. Furthermore, since it is taken across two sounds, it contains within itself some of the co-articulation information which is not present in other units such as the phoneme.
- One of the disadvantages with diphones is the large number of diphones (running into thousands). Furthermore, the phonological rules as they are written at preset are not easily applied to diphones.

SYLLABLE

- Generally speaking, a syllable consists of a vowel and its neighboring consonants. The theoretical basis for deciding just how to segment a word into its component syllables has yet to be fully agreed by linguists. The advantages are that the nuclei of syllables are relatively easy to locate and identify acoustically. Consequently, they can also form the basis for continuous speech recognition.

WORD

- Although we as humans can claim to have a good understanding of just what a word is, its acoustic form is very difficult to define. This difficulty arises because when we speak we have a tendency to blend one word into the next. This forms a sort of hybrid word which is neither the first nor the second, but could be construed as a separate entity.

- The main disadvantages are that when dealing with a very large lexicon, the scanning of the templates to find the best match can be very time-consuming. There are many algorithms to reduce the number of templates scanned; an example here is the use of partial phonetic information, in lexical, (word) access. There is also the problem of recognizing word boundaries, because when we speak we tend to anticipate the next word. This results in the beginnings and in particular the ends, of words being distorted from the stored templates of words spoken in isolation. The problem then becomes one of determining how a template may be distorted in these situations.

2.6 WORD RECOGNITION

- If segmentation takes place at a level lower than a word, some means of getting from the lower level to that of stored words has to be included in the system .There is a certain degree of uncertainty when trying to assign words to segmented data, and in many cases it is possible to assign more than one word to the data . When this occurs, an algorithm has to be included in the system which can generate the most likely word. The algorithm may include some form of syntactical analysis in order to weed out ungrammatical sequences, as well as some knowledge of the most likely words in a particular context. Thus, some knowledge of the semantics and pragmatics of a language can be incorporated into a speech recognition system. This can, however, make the recognition system very task specific.If the speech signal is viewed as a composite pattern in which a relatively few primitive patterns are combined in a multi-level hierarchy (e.g. bit, segment, word, etc.) then the patterns can be combined probabilistically and deterministically to form strings at the semantic level.

- Syntactical analysis can also be used to restrict the recognition of the next word on the basis of previously stored words. This is desirable, because with large lexicons the task of searching for the best match can become very expensive computationally. However, due to the vagaries of the English language, this syntactical approach has certain limitations, including the difficulty in distinguishing between well-formed and poorly formed sentences. A statistical approach can be adopted at all levels of decision making where a score can be assigned to each of the alternatives on the basis of past history; the alternatives with the highest score being the one selected for further processing. There are many ways of obtaining the highest score. The ‘breadth first search' computes the score at each alternative and selects the route with the highest score. The depth first search' selects the highest score at the initial level and then pursues this initial choice in subsequent levels, in a 'depth first' manner. The problem with the 'depth first‘ technique is that the system **is** committed to the consequences of the first choice. There are also searching techniques which area hybrid of 'breadth first' and 'depth first' techniques.

2.7 RESPONDING TO THE MESSAGE

- Assuming that all the words and sentences have been correctly identified, the computer must then be able to respond in the appropriate manner. The response can be in the form of an execution of an operating system command, the display of a string (message) on the screen, the lexical display of the words actually spoken, etc.

1. AUTOMATIC SPEECH RECOGNITION (ASR)

- True spoken language understanding is a difficult task, and it is remarkable that humans do as well at it as we do. The goal of **automatic speech recognition (ASR) research is to address this problem** computationally by building systems that map from an acoustic signal to a string of words. **Automatic speech understanding (ASU) extends this goal** to producing some sort of understanding of the sentence, rather than just the words.
- The general problem of automatic transcription of speech by any speaker in any environment is still far from solved. But recent years have seen ASR technology mature to the point where it is viable in certain limited domains.
- ●

● One major application area is in human-computer interaction. While many tasks are better solved with visual or pointing interfaces, speech has the potential to be a better interface than the keyboard for tasks where full natural language communication is useful, or for which keyboards are not appropriate. This includes hands-busy or eyes-busy applications, such as where the user has objects to manipulate or equipment to control.

●Another important application area is telephony, where speech recognition is already used for example in spoken dialogue systems for entering digits, recognizing “yes” to accept collect calls, finding out airplane or train information, and call-routing.In some applications, a multimodal interface combining speech and pointing can be more efficient than a graphical user interface without speech.

●Finally, ASR is applied to dictation, that is, transcription of extended monologue by a single specific speaker.

SPEECH RECOGNITION TASK

- **1. One dimension of variation in speech recognition tasks is the vocabulary** size. Speech recognition is easier if the number of distinct words we need to recognize is smaller. So tasks with a two word vocabulary, like yes versus no detection, or an eleven word vocabulary, like recognizing sequences of digits, in what is called the **digits task , are relatively easy.** On the other end, tasks with large vocabularies, like transcribing human telephone conversations, or transcribing broadcast news, tasks with vocabularies of 64,000 words or more, are much harder.

- **2.A second dimension of variation is how fluent, natural, or conversational** the speech is. **Isolated word recognition, in which each word is surrounded** by some sort of pause, is much easier than recognizing **continuous speech,** in which words run into each other and have to be segmented. Continuous speech tasks themselves vary greatly in difficulty. For example, human-to machine speech turns out to be far easier to recognize than human-to-human speech. That is, recognizing speech of humans talking to machines, either reading out loud in **read speech (which simulates the dictation task),** or conversing with speech dialogue systems, is relatively easy. Recognizing the speech of two humans talking to each other, in **conversational speech** recognition, for example for transcribing a business meeting or a telephone conversation, is much harder. It seems that when humans talk to machines, they simplify their speech quite a bit, talking more slowly and more clearly.

- **3. A third dimension** of variation is channel and noise. Commercial dictation systems, and much laboratory research in speech recognition, is done with high quality, head mounted microphones. Head mounted microphones eliminate the distortion that occurs in a table microphone as the speakers head moves around. Noise of any kind also makes recognition harder. Thus recognizing a speaker dictating in a quiet office is much easier than recognizing open.
- **4. A final dimension** of variation is accent or speaker-class characteristics. Speech is easier to recognize if the speaker is speaking a standard dialect, or in general one that matches the data the system was trained on. Recognition is thus harder on foreign accented speech, or speech of children (unless the system was specifically trained on exactly these kinds of speech).

3. SPEECH RECOGNITION ARCHITECTURE

- The task of speech recognition is to take as input an acoustic waveform and produce as output a string of words. HMM-based speech recognition systems view this task using the metaphor of the noisy channel. The intuition of the **noisy channel model (Fig.1) is to treat the acoustic waveform as an** "noisy" version of the string of words, i.e.. a version that has been passed through a noisy communications channel. This channel introduces "noise" which makes it hard to recognize the "true" string of words. Our goal is then to build a model of the channel so that we can figure out how it modified this "true" sentence and hence recover it. The insight of the noisy channel model is that if we know how the channel distorts the source, we could find the correct source sentence for a waveform by taking every possible sentence in the language, running each sentence through our noisy channel model, and seeing if it matches the output. We then select the best matching source sentence as our desired source sentence.

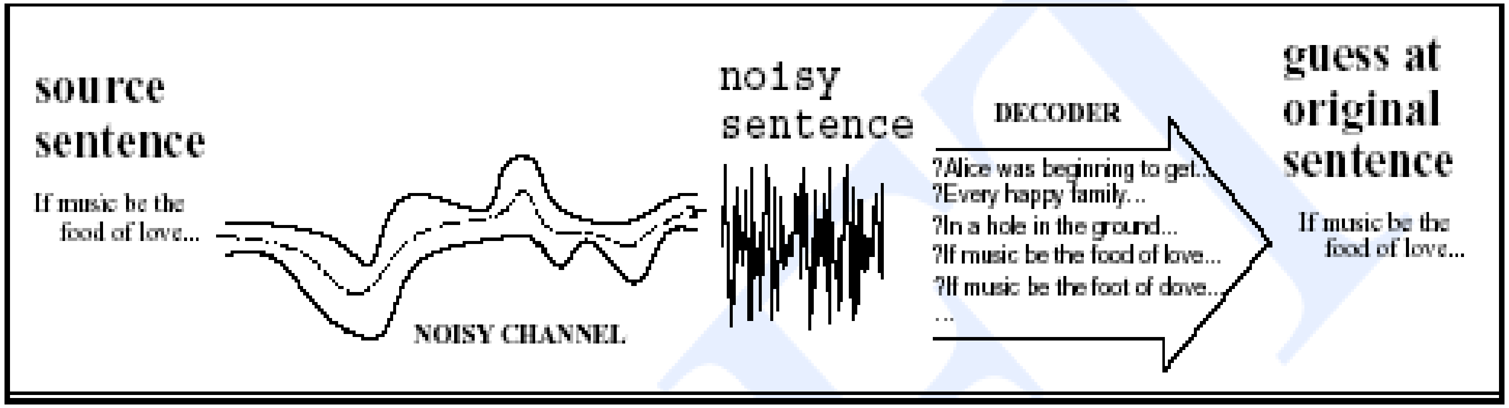

source sentence
If music be the food of love...
NOISY CHANNEL
noisy sentence
DECODER
?Alice was beginning to get...
?Every happy family...
?In a hole in the ground...
?If music be the food of love...
?If music be the foot of dove...
...
guess at original sentence
If music be the food of love...

Figure 1 the noisy channel model.

- We search through a huge space of potential “source” sentences and choose the one which has the highest probability of generating the “noisy” sentence.
- We need models of the prior probability of a source sentence , the probability of words being realized as certain strings of phones (HMM lexicons), and the probability of phones being realized as acoustic or spectral features (Gaussian Mixture Models).
- Implementing the noisy-channel model as we have expressed it in Fig.1 requires solutions to two problems.
- First, in order to pick the sentence that best matches the noisy input we will need a complete metric for a “best match”. Because speech is so variable, an acoustic input sentence will never exactly match any model we have for this sentence. Probability will be used as metric.
- The goal is to combine various probabilistic models to get a complete estimate for the probability of a noisy acoustic observation-sequence given a candidate source sentence. Then search will be made through the space of all sentences, and choose the source sentence with the highest probability.
- Second since the set of all English sentences is huge, an efficient algorithm needed that will not search through all possible sentences, but only ones that have a good chance of matching the input. This is the decoding or search problem**.**

3. HMM Speech Recognizer

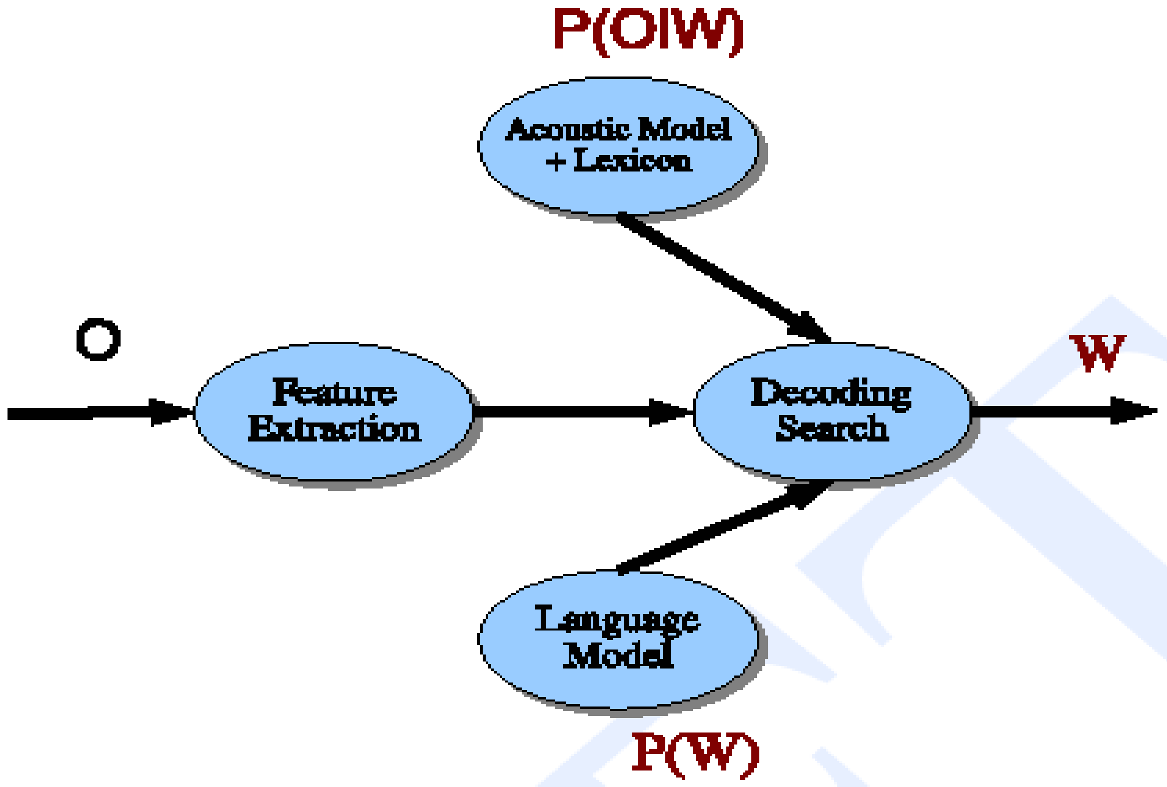


Figure 2 A block diagram of a speech recognizer decoding a single sentence, showing the integration of *P(W) and P(O|W)*.

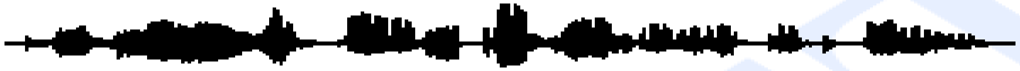
- Fig.3 shows further details of the operation alization in, which shows the components of an HMM speech recognizer as it processes a single utterance. The figure shows the recognition process in three stages. In the **feature extraction or signal processing stage, the acoustic waveform is sampled** into **frames (usually of 10, 15, or 20 milliseconds) which are transformed** into **spectral features. Each time window is thus represented by a vector of** around 39 features representing this spectral information as well as information about energy and spectral change.
- In the **acoustic modeling or phone recognition stage, we compute the** likelihood of the observed spectral feature vectors given linguistic units (words, phones, subparts of phones). For example, we use Gaussian Mixture Model (GMM) classifiers to compute for each HMM state *q, corresponding* to a phone or subphone, the likelihood of a given feature vector given this phone $p(o|q)$. *A (simplified) way of thinking of the output of this stage is as* a sequence of probability vectors, one for each time frame, each vector at each time frame containing the likelihoods that each phone or sub-phone unit generated the acoustic feature vector observation at that time.

- Finally, in the **decoding phase, we take the acoustic model (AM),** which consists of this sequence of acoustic likelihoods, plus an HMM dictionary of word pronunciations, combined with the language model (LM) (generally an *N-gram grammar), and output the most likely sequence of* words. An HMM dictionary, is a list of word pronunciations, each pronunciation represented by a string of phones. Each word can then be thought of as an HMM, where the phones (or sometimes subphones) are states in the HMM, and the Gaussian likelihood estimators supply the HMM output likelihood function for each state. Most ASR systems use the Viterbi algorithm for decoding, speeding up the decoding with wide variety of sophisticated augmentations such as pruning, fast-match, and tree-structuredl exicons.

1. SPEECH SYNTHESIS

- The modern task of **speech synthesis, also called text-to-speech or TTS, is to produce speech (acoustic waveforms) from text input. Modern** speech synthesis has a wide variety of applications.
- Synthesizers are used, together with speech recognizers:
- 1- in telephone-based conversational agents that conduct dialogues with people.
- 2-Synthesizer are also important in non conversational applications that speak **to people, such as in devices that read** out loud for the blind, or in video games or children's toys.
- Finally, speech synthesis can be used to speak **for sufferers of neurological disorders, such** as astrophysicist Steven Hawking who, having lost the use of his voice due to ALS, speaks by typing to a speech synthesizer and having the synthesizer speak out the words.

- The task of speech synthesis is to map a text like the following:
- PG&E will file schedules on April 20. to a waveform like the following: Speech



- Speech synthesis systems perform this mapping in two steps, first converting the input text into a **phonemic internal representation and then converting** this internal representation into a waveform. We will call the first step **text analysis and the second step waveform synthesis although other names are** also used for these steps).
- A sample of the internal representation for this sentence is shown in Fig1. Note that the acronym PG&E is expanded into the words P G AND E, the number 20 is expanded into twentieth, a phone sequence is given for each of the words, and there is also prosodic and phrasing information (the *'s) which we will define later.
- The numbers and acronyms have been expanded, words have been converted into phones, and prosodic features have been assigned. Fig.2 shows the TTS architecture for concatenative unit selection synthesis.

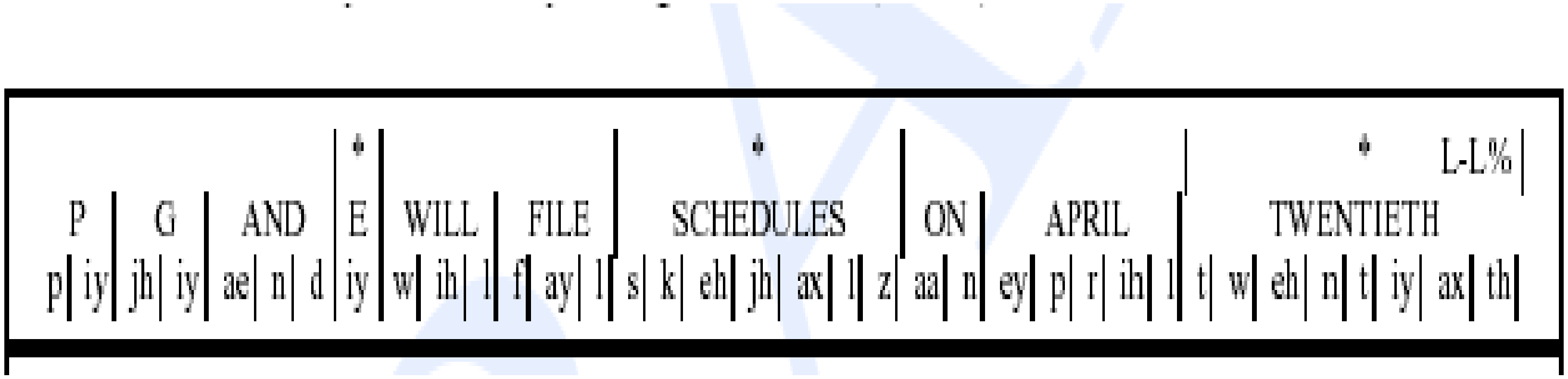

*
*
*
L-L%
P
G
AND
E
WILL
FILE
SCHEDULES
ON
APRIL
TWENTIETH
p iy jh iy ae n d iy w ih l f ay l s k eh jh ax l z aa n ey p r ih l t w eh n t iy ax th

Figure 1 Intermediate output for a unit selection synthesizer for the sentence PG&E will file schedules on April 20.

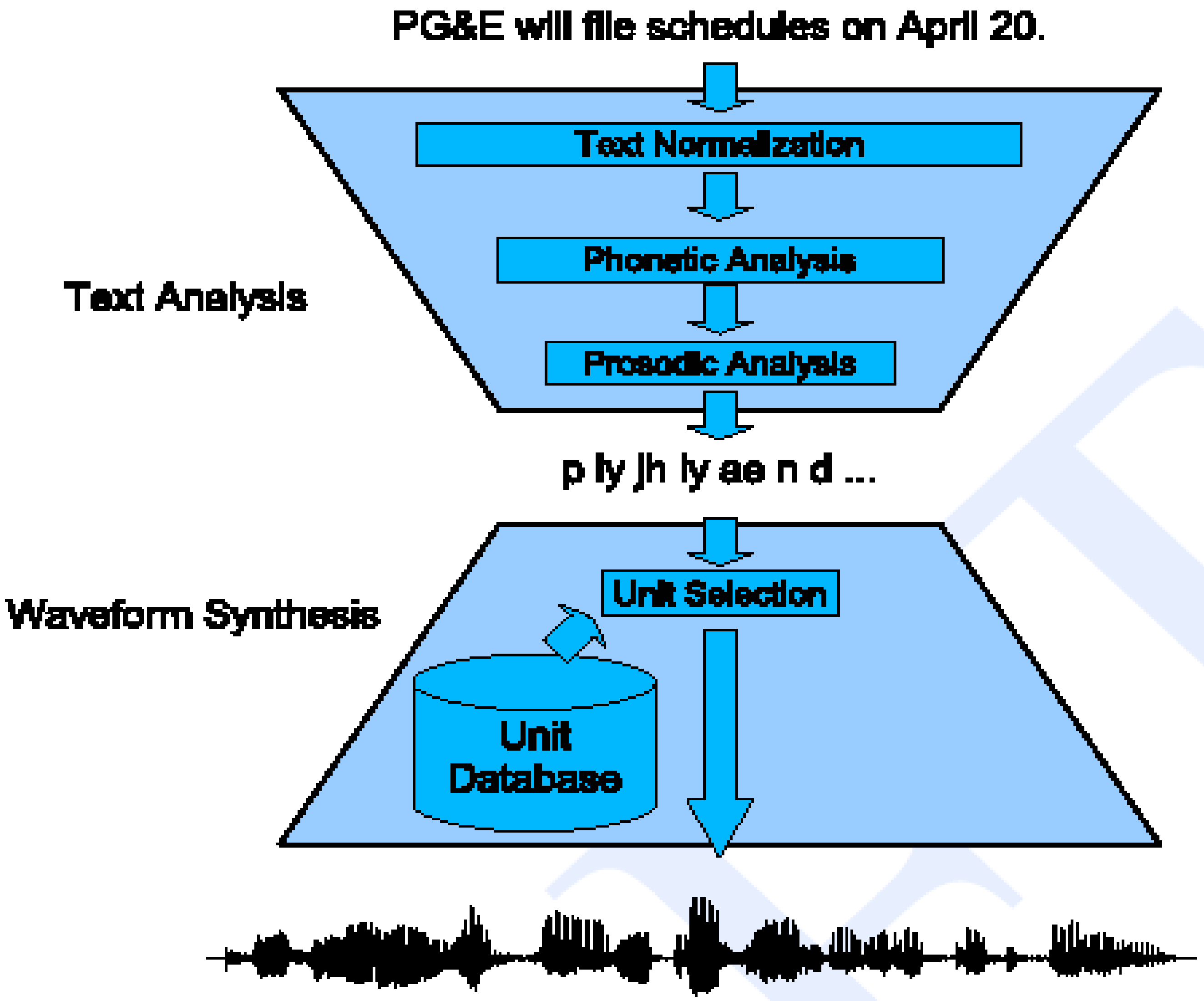


Figure 2 *Architecture for the unit selection (concatenative) architecture for speech synthesis.*

1.1 TEXT NORMALIZATION

- In order to generate a phonemic internal representation, raw text first needs to be preprocessed or **normalized in a variety of ways. We'll need to** break the input text into sentences, and deal with the idiosyncrasies of abbreviations, numbers, and so on. Consider the difficulties in the following text drawn :

  "He said the increase in credit limits helped B.C. Hydro achieve record net income of about $1 billion during the year ending March 31. This figure does not include any write downs that may occur if Powerex determines that any of its customer accounts are not collectible. Cousins, however, was insistent that all debts will be collected: "We continue to pursue monies owing and we expect to be paid for electricity we have sold."

- The first task in text normalization is **sentence tokenization. In order to** segment this paragraph into separate utterances for synthesis, we need to know that the first sentence ends at the period after *March 31, not at the* period of *B.C.. We also need to know that there is a sentence ending at the* word *collected, despite the punctuation being a colon rather than a period.*
- ●The second normalization task is dealing with **non-standard words. Nonstandard** words include number, acronyms, abbreviations, and so on. For example, *March 31 needs to be pronounced March thirty-first, not March three one; $1 billion needs to be pronounced one billion dollars, with the* word *dollars appearing after the word billion.*

1.2 PHONETIC ANALYSIS

- The next stage in synthesis is to take the normalized word strings from text analysis and produce a pronunciation for each word. The most important component here is a large pronunciation dictionary. Dictionaries alone turn out to be insufficient, because running text always contains words that don’t appear in the dictionary.
- Thus the two main areas where dictionaries need to be augmented is in dealing with names and with there unknown words.

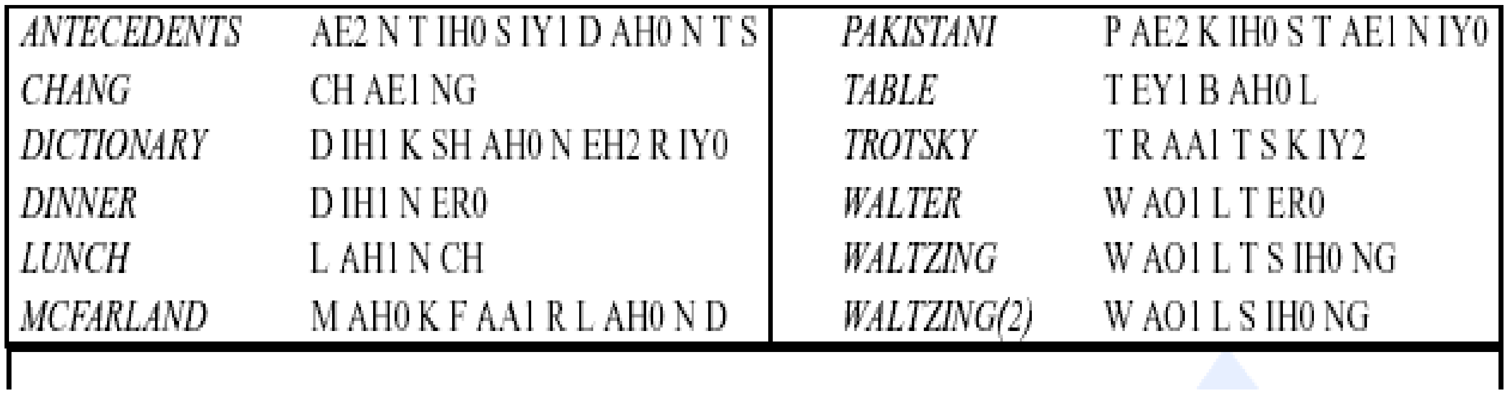
1.2.1 DICTIONARY LOOKUP

- One of the most widely-used for TTS is the freely available CMU Pronouncing Dictionary (CMU, 1993), which has pronunciations for about120,000 words. The pronunciations are roughly phonemic, from a 39-phone ARPAbet-derived phoneme set. Phonemic transcriptions means that instead of marking surface reductions like the reduced vowels [ax] or [ix], CMUdict marks each vowel with a stress tag, 0 (unstressed), 1 (stressed), or 2 (secondary stress). Thus (non-diphthong) vowels with 0 stress generally correspond to [ax] or [ix]. Most words have only a single pronunciation, but about 8,000 of the words have two or even three pronunciations, and so some kinds of phonetic reductions are marked in these pronunciations. The dictionary is not syllabified, although the nucleus is implicitly marked by the (numbered) vowel. Fig.3 shows some sample pronunciations. The CMU dictionary was designed for speech recognition rather than synthesis uses; thus it does not specify which of the multiple pronunciations to use for synthesis, does not mark syllable boundaries, and because it capitalizes the dictionary headwords, does not distinguish between e.g., *US and us (the form US has the two pronunciations [AH1 S] and [Y UW1 EH1 S].*

CMU

- Carnegie Mellon University (134000 Words in the new dictionary)
- 0--- no stress
- 1-- primery stress
- 2-- secondary stress

phoneme	example	translation
AA	odd	AA D
AE	at	AE T



ANTECEDENTS	AE2 N T IH0 S IY1 D AH0 N T S	*PAKISTANI*	P AE2 K IH0 S T AE1 N IY0
CHANG	CH AE1 NG	*TABLE*	T EY1 B AH0 L
DICTIONARY	D IH1 K SH AH0 N EH2 R IY0	*TROTSKY*	T R AA1 T S K IY2
DINNER	D IH1 N ER0	*WALTER*	W AO1 L T ER0
LUNCH	L AH1 N CH	*WALTZING*	W AO1 L T S IH0 NG
MCFARLAND	M AH0 K F AA1 R L AH0 N D	*WALTZING(2)*	W AO1 L S IH0 NG

Figure 3 ***some sample pronunciations from the CMU Pronouncing Dictionary.***

1.2.2 GRAPHEME-TO-PHONEME

- Once we have expanded non-standard words and looked them all up in a pronunciation dictionary, we need to pronounce the remaining, unknown words. **The process of converting a sequence of letters into a sequence of phones is called grapheme-to phoneme conversion, sometimes shortened g2p. The job of a grapheme-to-phoneme algorithm is thus to convert a letter string like *cake into a phone string like [K EY K]*.**
- The earliest algorithms for grapheme-to-phoneme conversion these are often called **letter-to-sound or LTS rules, and they are still used in some** Systems. LTS rules are applied in order, with later (default) rules only applying if the context for earlier rules is not applicable.
- A simple pair of rules for pronouncing the letter *c might be as follows:*
- *c ! [k] / {a,o}V ; context-dependent*
- *c ! [s] ; context-independent*

FINDING A LETTER-TO-PHONE ALIGNMENT FOR THE TRAINING SET

- Most letter-to-phone algorithms assume that we have an **alignment, which** tells us which phones align with each letter. We'll need this alignment for each word in the training set. Some letters might align to multiple phones (e.g., *x often aligns to k s), while other letters might align with no phones at* all, like the final letter of *cake in the following alignment:*
- L: c a k e
- P: K EY K e
- One method for finding such a letter-to-phone alignment is the semiautomatic method. Their algorithm is semi-automatic because it relies on a hand-written list of the **allowable phones that can realize each letter.**
- Here are allowables lists for the letters *c and e:*
- c: k ch s sh t-s e
- e: ih iy er ax ah eh ey uw ay ow y-uw oy aa e

- In order to produce an alignment for each word in the training set, we take this allowable list for all the letters, and for each word in the training set, we find all alignments between the pronunciation and the spelling that conform to the allowables list. From this large list of alignments, we compute, by summing over all alignments for all words, the total count for each letter being aligned to each phone (or multiphone or e). From these counts we can normalize to get for each phone *pi and letter l j*
- a probability *P(pi|l j):*
- *P(pi|l j) =count(pi, l j) /count(l j)*
- We can now take these probabilities and realign the letters to the phones, using the Viterbi algorithm to produce the best (Viterbi) alignment for each word, where the probability of each alignment is just the product of all the individual phone/letter alignments. In this way we can produce a single good alignment *A for each particular* pair *(P,L) in our training set.*

1.3 PROSODIC ANALYSIS

- The final stage of linguistic analysis is prosodic analysis. In poetry, the word **prosody refers to the study of the metrical structure of verse. In** linguistics and language processing, however, we use the term **prosody to** mean the study of the intonational and rhythmic aspects of language. And **energy independently of the phone string.**
- By **sentence-level pragmatic meaning, is referring to a number of kinds of** meaning that have to do with the relation between a sentence and its discourse or external context. **prosody uses:**

- **1. Can be used to mark discourse structure or function, like the difference** between statements and questions, or the way that a conversation is structured into segments or sub dialogs.
- **2. Prosody is also used to mark saliency, such as indicating that a particular** word or phrase is important or salient.
- **3. Prosody is heavily used for affective and emotional meaning, such as** expressing happiness, surprise, or anger.

1.3.1 PROSODIC STRUCTURE

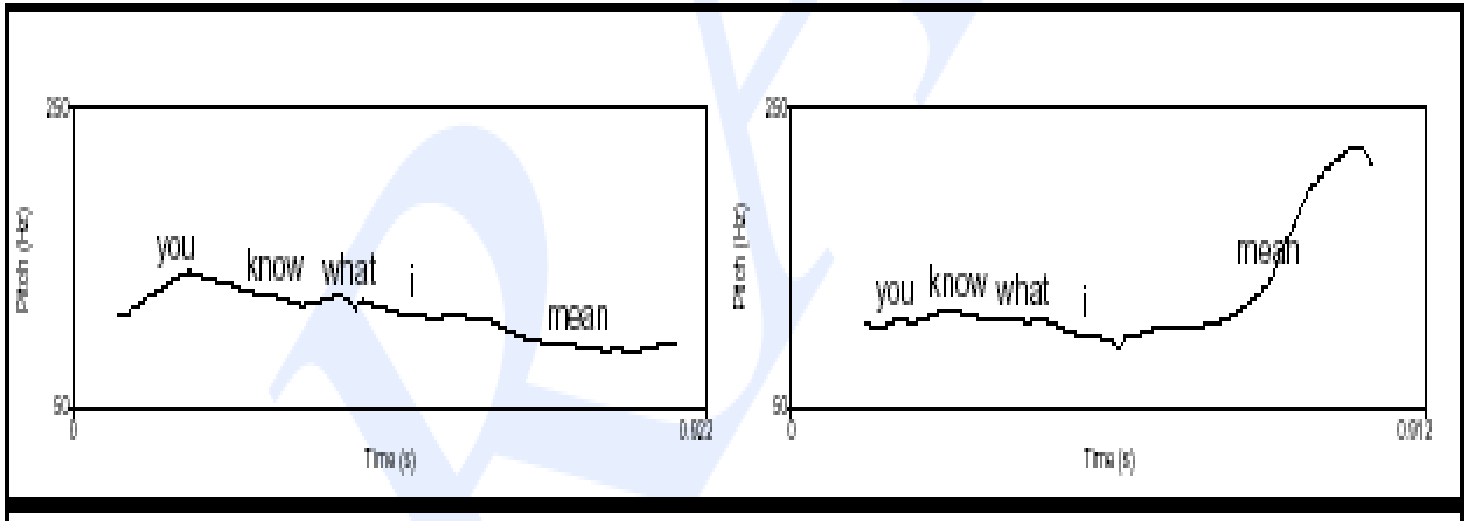
- Spoken sentences have prosodic structure in the sense that some words seem to group naturally together and some words seem to have a noticeable break or disjuncture between them. Often prosodic structure is described in terms of **prosodic phrasing,** meaning that an utterance has a prosodic phrase structure in a similar way to it having a syntactic phrase structure. For example, in the sentence *I wanted to go to London, but could only get tickets for France there seems to be two* main **intonation phrases, their boundary occurring at the comma.**
- Furthermore, in the first phrase, there seems to be another set of lesser
- prosodic phrase boundaries (often called **intermediate phrases that split up** the words as follows *I wanted | to go | to London.*
- Practical phrase boundary prediction is generally treated as a binary classification task, where we are given a word and we have to decide whether or not to put a prosodic boundary after it. a text-only alternative is often used. In this method, a human labeler looks only at the text of the training corpus, ignoring the speech. The labeler marks any juncture between words where they feel a prosodic boundary might legitimately occur if the utterance were spoken.

1.3.2 PROSODIC PROMINENCE

- In any spoken utterance, some words sound more **prominent than** others. Prominent words are perceptually more salient to the listener; speakers make a word more salient in English by saying it louder, saying it slower (so it has a longer duration).We generally capture the core notion of prominence by associating a linguistic marker with prominent words, a marker called **pitch accent.** Words which are prominent are said to bear (be associated with) a pitch accent. Pitch accent is thus part of the phonological description of a word in context in a spoken utterance. Pitch accent is related to **stress. The stressed syllable of a word is where pitch accent is realized.**
- In other words, if a speaker decides to highlight a word by giving it a pitch accent, the accent will appear on the stressed syllable of the word.

1.3.3 TUNE

- Two utterances with the same prominence and phrasing patterns can still differ prosodically by having different **tunes. The tune of an utterance** is the rise and fall of its F0 over time. A very obvious example of tune is the difference between statements and yes-no questions in English. The same sentence can be said with a final rise in F0 to indicate a yes-no-question, or a final fall in F0 to indicate a declarative intonation.
- Fig.4 shows the F0 track of the same words spoken as a question or a statement. Note that the question rises at the end; this is often called a **question rise. The falling intonation of the statement is called a final fall.**



The same text read as the statement You know what I mean. (on the left) and as a question You know what I mean? (on the right). Notice that yes-no-question intonation in English has a sharp final rise in F0.

- It turns out that English makes very wide use of tune to express meaning.
- Besides this well known rise for yes-no questions, an English phrase containing a list of nouns separated by commas often has a short rise called a **continuation rise after each noun. English also has characteristic contours** to express contradiction, to express surprise, and many more. The mapping between meaning and tune in English is extremely complex, and linguistic theories of intonation like ToBI have only begun to develop sophisticated models of this mapping. In practice, therefore, most synthesis systems just distinguish two or three tunes, such as the **continuation rise (at** commas), the **question rise (at question mark if the question is a yes-no** question), and a **final fall otherwise.**

THE RELATIONSHIP BETWEEN NL & SR:

- Spoken language understanding involves two primary component technologies: speech Recognition (SR), and Natural Language (NL) understanding. The integration of speech & natural language has great advantages. NL can bring to SR additional knowledge source (e.g. syntax and semantic). SR to NL is the aim of natural language understanding.

COMPARES BETWEEN WRITTEN TEXT & SPOKEN LANGUAGE

Written Text	Spoken Language
1. Input to computer is done by keyboard.	1.Input to computer is done by source of voice such as human speech using microphone which is connected to computer.
2. People can create more complex sentences than in the spoken form because they have more time to think & plan.	2. Since speech is done in interactive communication more errors are created.
3. A writer an pause for days or months before continuing a though.	3. Every pause, restart, revision of speech would hesitation the listener.
4. Formal and Informal methods are used.	4. Spoken mode must has speech recognition dependent on: text dependent SR sys. & text independent SR sys. And Isolated or continuous speech.
Informal method has three stages to perform: - Lexical analysis. - Syntax analysis. - Semantic analysis.	5. SR sys. Has many stages to convert speech -to- text that stored in computer: - Sampling. - Detecting start & end of signal. - Calculation of speech spectra. - Pitch contour evaluation. - Segmentation. - Word recognition. - Message response.

COMPARES BETWEEN WRITTEN TEXT & SPOKEN LANGUAGE

Written Text	Spoken Language
6. The implementation is easy	6. The Implementation is more difficult
7.More effort and time to use.	7. Easy way in use.
8. Persons must have pre information to use computer.	8. Un trainable persons an use speech to provide commands to computer like children.