Anna University
B.E.(Computer Science and Engineering) – 2017 Regulations
VI SEM CSE CS8691 Artificial Intelligence

UNIT V APPLICATIONS 9

AI applications – Language Models – Information Retrieval- Information Extraction – Natural Language Processing - Machine Translation – Speech Recognition – Robot – Hardware – Perception – Planning – Moving

UNIT-V Question and Answers

(1) Define communication.

Communication is the intentional exchange of information brought about by the production and perception of **signs** drawn from a shared system of conventional signs. Most animals use signs to represent important messages: food here, predator nearby etc. In a partially observable world, communication can help agents be successful because they can learn information that is observed or inferred by others.

(2) What is speech act?

What sets humans apart from other animals is the complex system of structured messages known as **language** that enables us to communicate most of what we know about the world. This is known as speech act.

Speaker, hearer, and **utterance** are generic terms referring to any mode of communication. The term **word** is used to refer to any kind of conventional communicative sign.

(3) **What are the capabilities gained by an agent from speech act?**.

- **Query** other agents about particular aspects of the world. This is typically done by asking questions: *Have you smelled the wumpus anywhere?*
- **Inform** each other about the world. This is done by making representative statements: *There's a breeze here in 3 4.* Answering a question is another Pund of informing.
- **Request** other agents to perform actions: *Please help me carry the gold.* Sometimes **indirect speech act** (a request in the form of a statement or question) is considered more polite: *I could use some help carrying this.* An agent with authority can give commands *(Alpha go right; Bravo and Charlie go lejl),* and an agent with power can make a threat *(Give me the gold, or else).* Together, these kinds of speech acts are called **directives.**
- **Acknowledge** requests: **OK.**
- **Promise** or commit to a plan: *I'll shoot the wumpus; you grab the gold.*

(4) Define formal language.

A formal language is defined as a (possibly infinite) set of' **strings.** Each string is a concatenation of **terminal symbols,** sometimes called words. For example, in the language of first-order logic, the terminal symbols include A and P, and a typical string is "P A Q." . Formal languages such as first-order logic and Java have strict mathematical definitions. This is in contrast to **natural languages,** such as Chinese, Danish, and English, that have no strict definition but are used by a community.

(5) Define a grammar.

A **grammar** is a finite set of rules that specifies a language. Formal languages always have an official grammar, specified in manuals or books. Natural languages have no official grammar, but linguists strive to discover properties of the language by a process of scientific inquiry and then to codify their discoveries in a grammar.

(6) What are the component steps of communication? Explain with an example.

The component steps of communication

A typical communication episode, in which speaker S wants to inform hearer H about proposition P using words W, is composed of seven processes:

1) Intention. Somehow, speaker S decides that there is some proposition P that is worth saying to hearer H. For our example, the speaker has the intention of having the hearer know that the wumpus is no longer alive.

2) Generation. The speaker plans how to turn the proposition P into an utterance that makes it likely that the hearer, upon perceiving the utterance in the current situation, can infer the meaning P (or something close to it). Assume that the speaker is able to come up with the words "The wumpus is dead," and call this *W.*

3) Synthesis. The speaker produces the physical realization *W'* of the words *W.* This can be via ink on paper, vibrations in air, or some other medium. In Figure 22.1, we show the agent synthesizing a string of sounds *W'* written in the phonetic alphabet defined on page 569: "[thaxwahmpaxsihzdehd]." The words are run together; this is typical of quickly spoken speech.

4) Perception. H perceives the physical realization W' as *Wi* and decodes it as the words *W2.* When the medium is speech, the perception step is called **speech recognition;** when it is printing, it is called **optical character recognition.**

5) Analysis. H infers that *W2* has possible meanings *PI, . . . , P,.*

We divide analysis into three main parts:

a) **syntactic interpretation (or parsing**),

b) **Semantic interpretation,** and

c)**Pragmatic interpretation**.

Parsing is the process of building a **parse tree** for an input string, as shown in Figure 22.1. The interior nodes of the parse tree represent phrases and the leaf nodes represent words.

Semantic interpretation is the process of extracting the meaning of an utterance as an expression in some representation language. Figure 22.1 shows two possible semantic interpretations: that the wumpus is not alive and that it is tired (a colloquial meaning of dead). Utterances with several possible interpretations are said to be **ambiguous.**

Pragmatic interpretation takes into account the fact that the same words can have different meanings in different situations.

6) Disambiguation. H infers that *S* intended to convey P, (where ideally *P,* = P). Most speakers are not intentionally ambiguous, but most utterances have several feasible interpretations.. Analysis generates possible interpretations;if more than one interpretation is found, then disambiguation chooses the one that is best.

7) **Incorporation. H** decides to believe *P,* (or not). A totally naive agent might believe everything it hears, but a sophisticated agent treats the speech act as evidence for P,, not confirmation of it.

Putting it all together, we get the agent program shown in Figure 22.2. Here the agent acts as a robot slave that can be commanded by a master. On each turn, the slave will answer a question or obey a command if the master has made one, and it will believe any statements made by the master. It will also comment (once) on the current situation if it has nothing more pressing to do, and it will plan its own action if left aloine.

Here is a typical dialog:

ROBOT SLAVE MASTER
I feel a breeze. Go to 12.
Nothing is here. Go north.
I feel a breeze and I smell a stench
and I see a glitter. Grab the gold.
Fig 22.1 shows the seven processes involved in communication,using the example sentence “The wumpus is dead”.

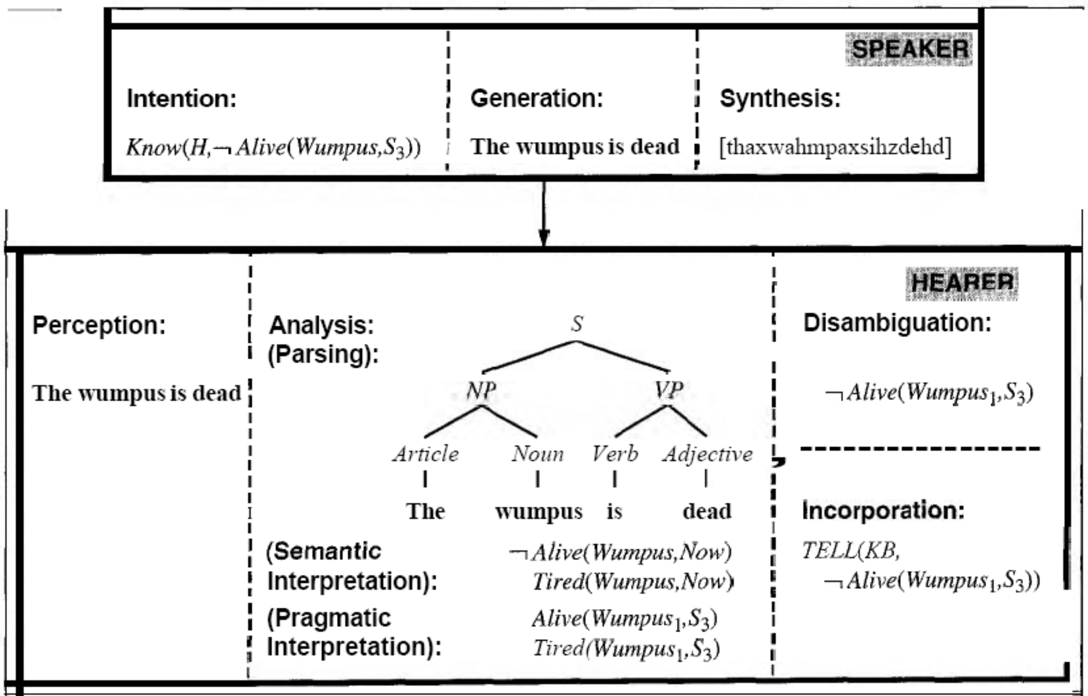

Figure 22.1 *Seven processes involved in communication, using the example sentence "The wumpus is dead."*

(7) Define a Lexicon and grammar for language consisting of a small fragment of English.

The Lexicon of $\mathcal{E}_0$

First we define the **lexicon,** or list of allowable words. The words are grouped into the categories or parts of speech familiar to dictionary users: nouns, pronouns, and names to denote

things, verbs to denote events, adjectives to modify nouns, and adverbs to modify verbs. Categories that are perhaps less familiar to some readers are articles (such as the), prepositions (in), and conjunctions (and). Figure 22.3 shows a small lexicon.

Noun	→	stench | breeze | glitter | **nothing** | agent
		| wumpus | pit | **pits**| **gold** | east | ...
Verb	→	is | see | smell | shoot | feel | stinks
		| go | grab | carry | kill | turn | ...
Adjective	→	right | left | east | dead | back | smelly | ..
Adverb	→	here | there | nearby | ahead
		| right | left | east | south | **back** | ...
Pronoun	→	me | you | I | it | ...
Name	→	John | Mary | Boston | **Aristotle** | ...
Article	→	the | a | an | ...
Preposition	→	to | in | on | near | ...
Conjunction	→	and | or | but | ...
Digit	→	**0| 1| 2| 3| 4| 5| 6| 7| 8| 9**

Figure 22.3 The lexicon for $\mathcal{E}_0$.

The Grammar of $\mathcal{E}_0$

The next step is to combine the words into phrases. We will use five nonterminal symbols to define the different kinds of phrases: sentence *(S)*, noun phrase (NP), verb phrase (VP), prepositional phrase (PP), and relative clause e el clause).' Figure 22.4 shows a grammar for l o , with an example for each rewrite rule. lo generates good English sentences such as the following:

John is in the pit

The wumpus that stinks is in 2 2

S	→	*NP VP*	*I* + feel a breeze
	|	*S Conjunction S*	I feel a breeze + and + I smell a wumpus
NP	→	*Pronoun*	*I*
	|	*Name*	John
	|	*Noun*	pits
	|	*Article Noun*	the + wumpus
	|	*Digit Digit*	3 4
	|	*NP PP*	the wumpus + to the east
	|	*NP RelClause*	the wumpus + that is smelly
VP	→	*Verb*	stinks
	|	*VP NP*	feel + a breeze:
	|	*VP Adjective*	is + smelly
	|	*VP PP*	turn + to the east
	|	*VP Adverb*	go + ahead
PP	→	*Preposition NP*	to + the east
RelClause	→	that *VP*	that + is smelly

Figure 22.4 The grammar for $\mathcal{E}_0$, with example phrases for each rule.

(8) What is parsing? Explain the top down parsing method.

Parsing is defined as the process of finding a **parse tree** for a given input string.

That is, a call to the parsing function PARSE, such as

PARSE("the wumpus is dead", $\mathcal{E}_0$, S)

should return a parse tree with root S whose leaves are "the wumpus is dead" and whose internal nodes are nonterminal symbols from the grammar Eo.

Parsing can be seen as a process of searching for a parse tree.

There are two extreme ways of specifying the search space (and many variants in between).

First, we can start with the *S* symbol and search for a tree that has the words as its leaves. This is called **top-down parsing**

Second, we could start with the words and search for a tree with root S. This is called **bottom-up parsing.**

Top-down parsing can be precisely defined as a search problem as follows:

- The initial state is a parse tree consisting of the root S and unknown children: [S: ?I. In general, each state in the search space is a parse tree.

The successor function selects the leftmost node in the tree with unknown children. It then looks in the grammar for rules that have the root label of the node on the left-hand side. For each such rule, it creates a successor state where the ? is replaced by a list corresponding to the right-hand side of the rule..

(9) Formulate the bottom-up parsing as a search problem.

The formulation of bottom-up parsing as a search is as follows:

The **initial state** is a list of the words in the input string, each viewed as a parse tree that is just a single leaf node-for example; **[the, wumlpus, is, dead].** In general, each state in the search space is a list of parse trees.

The **successor function** looks at every position ***i*** in the list of trees and at every righthand side of a rule in the grammar. If the subsequence of the list of trees starting at *i* matches the right-hand side, then the subsequence is replaced by a new tree whose category is the left-hand side of the rule and whose children are the subsequence. By "matches," we mean that the category of the node is the same as the element in the righthand side. For example, the rule *Article* + **the** matches the subsequence consisting of the first node in the list **[the, wumpus, is, dead],** so a successor state would be *[[Article:***the], wumpus, is, dead].**

The **goal test** checks for a state consisting of a single tree with root S.

See Figure 22.5 for an example of bottom-up parsing.

step	*list of nodes*	*subsequence*	*rule*
INIT	**the wumpus is dead**	**the**	Article → **the**
2	*Article* **wumpus is dead**	**wumpus**	Noun → **wumpus**
3	*Article Noun* **is dead**	*Article Noun*	*NP* → *Article Noun*
4	*NP* **is dead**	**is**	*Verb* → **is**
5	*NP Verb* **dead**	**dead**	*Adjective* → **dead**
6	**NP** *Verb Adjective*	*Verb*	*VP* → *Verb*
7	*NP VP Adjective*	*VP Adjective*	*VP* → *VP Adjective*
8	*NP VP*	*NP VP*	*S* → *NP VP*
GOAL	S		

Figure 22.5 Trace of a bottom up parse on the string "The wumpus is dead." We start with a list of nodes consisting of words. Then we replace subsequences that match the right-hand side of a rule with a new node whose root is the left-hand side. For example, in the third line the Article and Noun nodes are replaced by an NP node that has those two nodes as children. The top-down parse would produce a similar trace, but in the opposite direction.

(10) What is dynamic programming?

Forward Chaining on graph search problem is an example of dynamic programming. Solutions to the sub problems are constructed incrementally from those of smaller sub problems and are cached to avoid recomputation.

(11) Construct a parse tree for "You give me the gold" showing the sub categories of the verb and verb phrase.

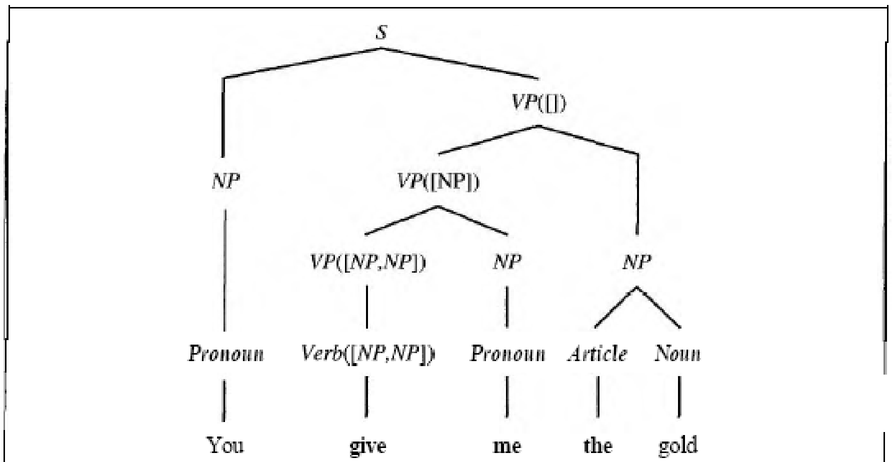

Figure **22.13** Parse tree for "You give me the gold" showing subcategorization of the verb and verb phrase.

(12) What is semantic interpretation? Give an example.

Semantic interpretation is the process of associating an FOL expression with a phrase.

$$
\begin{aligned}
Exp(x) &\rightarrow Exp(x_1)\ Operator(op)\ Exp(x_2)\ \{x = Apply(op, x_1, x_2)\}\\
Exp(x) &\rightarrow (\ Exp(x)\ )\\
Exp(x) &\rightarrow Number(x)\\
Number(x) &\rightarrow Digit(x)\\
Number(x) &\rightarrow Number(x_1)\ Digit(x_2)\ \{x = 10 \times x_1 + x_2\}\\
Digit(x) &\rightarrow x\ \{0 \leq x \leq 9\}\\
Operator(x) &\rightarrow x\ \{x \in \{+, -, \div, \times\}\}
\end{aligned}
$$

Figure 22.14 A grammar for arithmetic expressions, augmented with semantics. Each variable x_i represents the semantics of a constituent. Note the use of the $\{test\}$ notation to define logical predicates that must be satisfied, but that are not constituents.

(13) Construct a grammar and sentence for "John loves Mary"

$$
\begin{aligned}
S(rel(obj)) &\rightarrow NP(obj)\ VP(rel)\\
VP(rel(obj)) &\rightarrow Verb(rel)\ NP(obj)\\
NP(obj) &\rightarrow Name(obj)\\
\\
Name(John) &\rightarrow \textbf{John}\\
Name(Mary) &\rightarrow \textbf{Mary}\\
Verb(\lambda y\ \lambda x\ Loves(x,y)) &\rightarrow \textbf{loves}
\end{aligned}
$$

Figure 22.16 A grammar that can derive a parse tree and semantic interpretation for "John loves Mary" (and three other sentences). Each category is augmented with a single argument representing the semantics.

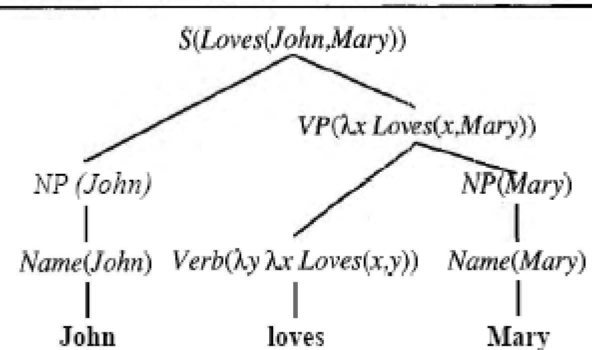

Figure 22.17 A parse tree with semantic interpretations for the string "John loves Mary".

(14) Construct a parse tree for the sentence "Every agent smells a Wumpus"

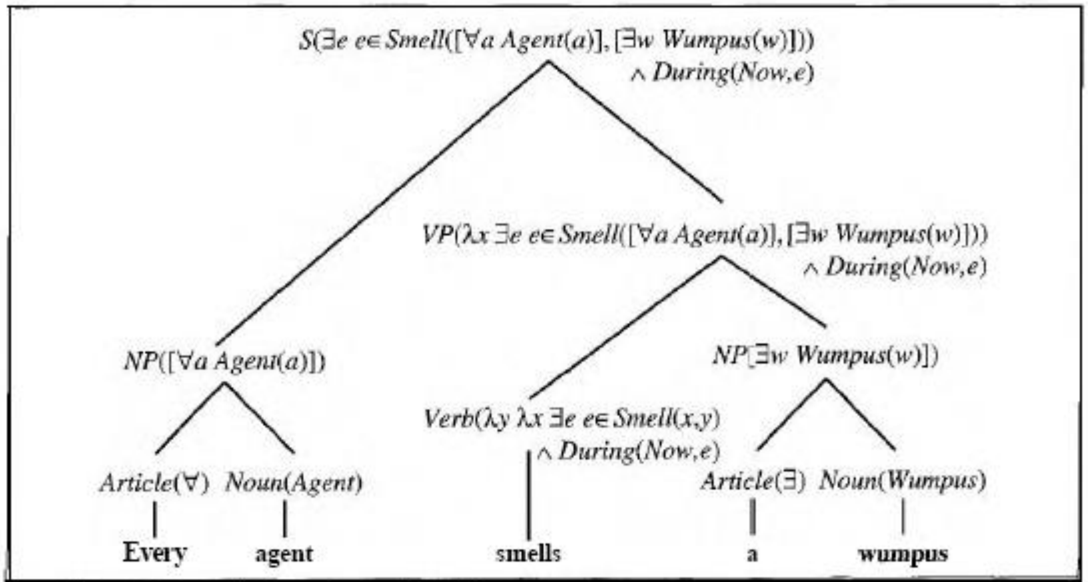

Figure 22.19 Parse tree for the sentence "Every agent smells a wumpus," showing both syntactic structure and semantic interpretations.

(15) Define lexical,syntactic, and semantic ambiguity.

Lexical ambiguity, in which a word has more than one meaning. Lexical ambiguity is quite common; "back" can be an adverb (go back), an adjective (back door), a noun (the back of the room) or a verb (back up your files). "Jack" can be a name, a noun (a playing card, a six-pointed metal game piece, a nautical flag, a fish, a male donkey, a socket, or a device for raising heavy objects), or a verb (to jack up a car, to hunt with a light, or to hit a baseball hard).

Syntactic ambiguity (also known as structural ambiguity) can occur with or without lexical ambiguity. For example, the string "I smelled a wumpus in *2,2"* has two parses: one where the prepositional phrase "in 2,2" modifies the noun and one where it modifies the verb. The syntactic ambiguity leads to a semantic ambiguity, because one parse means that the wumpus is in 2,2 and the other means that a stench is in *2,2.* In this case, getting the wrong interpretation could be a deadly mistake.

Semantic ambiguity can occur even in phrases with no lexical or syntactic ambiguity. For example, the noun phrase "cat person" can be someone who likes felines or the lead of the movie Attack of the Cat People. A "coast road" can be a road that follows the coast or one that leads to it.

(16) What is disambiguation?

Disambiguation

Disambiguation is a question of diagnosis. The speaker's intent to communicate is an unobserved cause of the words in the utterance, and the hearer's job is to work backwards from the words and from knowledge of the situation to recover the most likely intent of the speaker.. Some sort of preference is needed because syntactic and semantic interpretation rules alone cannot identify a unique correct interpretation of a phrase or sentence. So we divide the work: syntactic and semantic interpretation is responsible for enumerating a set of candidate interpretations, and the disambiguation process chooses the best one.

(17) What is discourse?

A discourse is any string of language-usually one that is more than one sentence long. Textbooks, novels, weather reports and conversations are all discourses. So far we have largely ignored the problems of discourse, preferring to dissect language into individual sentences that can be studied *in vitro.* We will look at two particular subproblems: reference resolution and coherence.

Reference resolution

Reference resolution is the interpretation of a pronoun or a definite noun phrase that refers to an object in the world. The resolution is based on knowledge of the world and of the previous parts of the discourse. Consider the passage "John flagged down the waiter. He ordered a hani sandwich."

To understand that "he" in the second sentence refers to John, we need to have understood that the first sentence mentions two people and that John is playing the role of a customer and hence is likely to order, whereas the waiter is not.

The structure of coherent discourse

If you open up this book to 10 random pages, and copy down the first sentence from each page. The result is bound to be incoherent. Similarly, if you take a coherent 10-sentence passage and permute the sentences, the result is incoherent. This demonstrates that sentences in natural language discourse are quite different from sentences in logic. In logic, if we TELL sentences A, *B* and *C* to a knowledge base, in any order, we endl up with Ithe conjunction *A* A *B A C.* In natural language, sentence order matters; consider th~ed ifference between "Go two blocks. Turn right." and "Turn right. Go two blocks."

(18) What is grammar induction?

Grammar induction is the task of learning a grammar from data. It is an obvious task to attempt, given that it has proven to be so difficult to construct a grammar by hand and that billions of example utterances are available for free on the Internet. It is a difficult task because the space of possible grammars is infinite and because verifying that a given grammar generates a set of sentences is computationally expensive.

Grammar induction can learn a grammar from examples, although there are limitations on how well the grammar will generalize.

(19) What is information retrieval?

Information retrieval is the task of finding documents that are relevant to a user's need for information. The best-known examples of information retrieval systems are search engines on the World Wide Web. A Web user can type a query such as [AI book] into a search engine and see a list of relevant pages. An information retrieval (henceforth IR) system can be characterized by:

1) **A document collection.** Each system must decide what it wants to treat as a document: a paragraph, a page, or a multi-page text.

2) A query posed in a query language. The query specifies what the user wants to know. The query language can be just a list of words, such as [AI book]; or it can specify a phrase of words that must be adjacent, as in ["AI book"]; it can contain Boolean operators as in **[AI** *AND* book]; it can include non-Boolean operators such as [AI book SITE:www.aaai.org].

3) A result set. This is the subset of documents that the IR system judges to be **relevant** to the query. By relevant, we mean likely to be of use to the person who asked the query, for the particular information need expressed in the query.

4) **A presentation of the result set.** This can be as simple as a ranked list of document titles or as complex as a rotating color map of the result set projected onto a three dimensional space.

(20) What is clustering?

Clustering is an unsupervised learning problem. Unsupervised clustering is the problem of discerning multiple categories in a collection of objects. The problem is unsupervised because the category labels are not given.

Examples

We are familiar with terms such as “red giant” and “white dwarf”,but the stars do not carry these labels – astronomers had to perform unsupervised clustering to identify these categories.

(21) What is agglomerative clustering?

Agglomerative clustering creates a tree of clusters going all the way down to the individual documents. We begin by considering each document as a separate cluster. Then we find the two clusters that are closest to each other according to some distance measure and merge these clusters into one. The distance measure between two clusters can be the distance to the median of the cluster. Agglomerative clustering takes time $O(n^2)$,where n is the number of documents.

(22) What is K-means clustering?

K-means clustering creates a flat set of exactly k-categories.It works as follows :

i) Pick K documents at random to represent the K categories.

ii) Assign every document to the closest category.

iii) Compute the mean of each cluster and use K-means to represent the new value of the K categories.

iv) Repeat the steps (ii) and (iii) until convergence.

K-means takes O(n)

(23) What is information extraction?

Information extraction is the process of creating database entries by skimming a text and looking for occurrences of a particular **class of object or event** and for **relationships** among those **objects and events**.

We could be trying to extract instances of addresses from web pages, with database fields for street, city, state, and zip code; or instances of storms from weather reports, with fields for temperature, wind speed, and precipitation. Information extraction systems are mid-way between information retrieval systems and full-text parsers, in that they need to do more than consider a document as a bag of words, but less than completely analyze every sentence.

The simplest type of information extraction system is called an ***attribute-based* system** because it assumes that the entire text refers to a single object and the task is to extract attributes of that object.

For example, the problem of extracting from the text "17in SXGA Monitor for only $249.99" the database relations given by

3 m m E *ComputerMonitors A Size(m, Inches(l7)) A Price(m, $(249.99))*
A Resolution(m, 1280 x 1024) .

Some of this information can be handled with the help of regular expressions, which define a regular grammar in a single text string. Regular expressions are used in Unix commands such as grep, in programming languages such as Perl, and in word processors such as Microsoft Word.

(24) What is machine translation? Explain different types.

Machine translation is the automatic translation of text from one natural language (the source) to another (the target). This process has proven to be useful for a number of tasks, including the following:

1. Rough translation, in which the goal is just to get the gist of a passage. Ungrammatical and inelegant sentences are tolerated as long as the meaning is clear. For example, in Web surfing, a user is often happy with a rough translation of a foreign web page.

Sometimes a monolingual human can post-edit the output without having to read the source. This type of machine-assisted translation saves money because such editors can be paid less than bilingual translators.
2**. Restricted-source translation**, in which the subject matter and format of the source text are severely limited. One of the most successful examples is the TAUM-METEO system, which translates weather reports from English to French. It works because the language used in weather reports is highly stylized and regular.
3. **Pre-edited translation**, in which a human preedits the source document to make it conform to a restricted subset of English (or whatever the original language is) before machine translation. This approach is particularly cost-effective when there is a need to translate one document into many languages, as is the case for legal documents in the European Community or for companies that sell the same product in many countries. Restricted languages are sometimes called "Caterpillar English," because Caterpillar Corp. was the first firm to try writing its manuals in this form. Xerox defined a language for its maintenance manuals which was simple enough that it could be translated by machine into all the languages Xerox deals with. As an added benefit, the original English manuals became clearer as well.
4**. Literary translation**, in which all the nuances of the source text are preserved. This is currently beyond the state of the art for machine translation.

(25) Draw a schematic for a machine translation system for English to French.

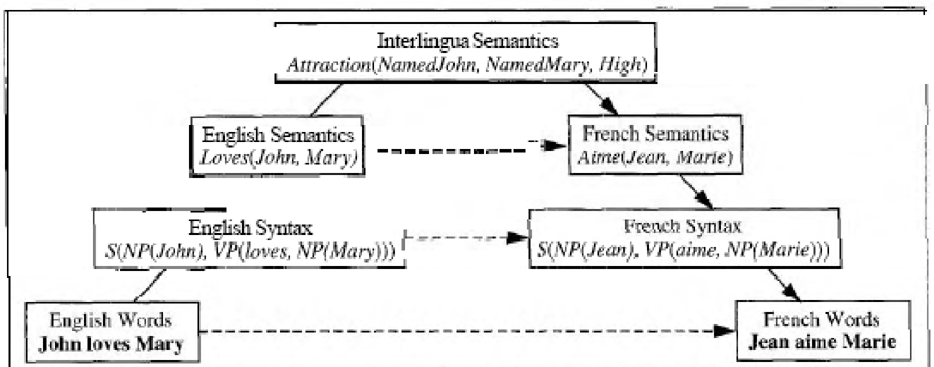

Figure 23.5 A schematic diagram of the choices for a machine translation system. We start with English text at the top. An interlingua-based system follows the solid lines, parsing English first into a syntactic form, then into a semantic representation and an interlingua representation, and then through generation to a semantic, syntactic, and lexical form in French. A transfer-based system uses the dashed lines as a shortcut. Different systems make the transfer at different points; some make it at multiple points.