Sub:	Natural Language Processing					Sub Code:	15CS741	
Date:	12.10.19	Duration:	90 min's	Max Marks:	50	Sem / Sec:	VII/A,B,C	

Answer any FIVE FULL Questions

MARKS

1 (a) Explain Rule based tagger and hybrid tagger in POS tagging. [04]

These rules use a lexicon to obtain a list of candidate and then use rules to discard incorrect tags.

- have a 2 stage architecture:
- First stage is simply a dictionary look up proc which returns a set of potential tags and appropriate syntactic features for each word.
- Second stage uses a handcoded rules to discard contextually illegitimate tags to get a single part of sp for each word.

eg: The noun-verb ambiguity

The show must go on

The potential tags for the word show in this sentence {VB, NN}.

IF preceding word is determiner THEN eliminate

This rule simply disallows verb after a determiner. this rule the word show is noun only.

In addition to contextual information, many taggers use morphological information to help in the disambiguation process.

IF word ends in -ing and preceding word is a ver
THEN label it a verb (VB)

- Advantage: speed, Deterministic than Stochastic
- Arguments against them is the skill and effort required in writing disambiguation rules.
- Stochastic taggers require manual work if good performance is to be achieved.
- Rule based, time is spent in writing a rule set
- Stochastic, time is spent developing restrictio on transitions and emissions to improve tagger performa
-> Disadvantage: it's usable for only one language.
- using it for another one requires a rewrite of m of the programs.

③ Hybrid taggers:

Hybrid combines both: rule based — use rules to assign tags to words; Stochastic

Like the Stochastic taggers, this is a machine learning technique and rules are automatically induced from data

eg: Hybrid approach - Transformation based learning (TBL) of tags, known as Brill tagging.

b) Explain Stochastic tagger and write the derivation of Maximum likelihood.[6]

② Stochastic tagger :

- Standard stochastic tagger is HMM tagger algo

- Markov model applies the simplifying assum that the probability of a chain of symbols can be approximated in terms of its parts of n-grams.

- Simplest n-gram model is unigram model, w assigns the most likely tag to each token.

eg: It will assign the tag JJ for ... fast is used as an adjective than used as noun, verb or adverb.

She had a fast → Noun

Muslims fast (verb) during Ramadan

Those who were injured in the accident need to b helped fast → adverb.

Let w be the sequence of words.

$W = w_1, w_2, \ldots w_n$

The task is to find the tag sequence

$T = t_1, t_2, \ldots t_n$

which maximizes $P(T|W)$ i.e.,

$T' = \text{argmax}_T \, P(T|W)$

Applying Bayes Rule, $P(T|W)$ can be the estimated using the expression:

Using these assumptions, we obtain

$$P(W|T) = P(w_1|t_1) * P(w_2|t_2) \ldots P(w_i|t_i) * \ldots P(w_n|t_n)$$

i.e) $$P(W|T) \approx \prod_{i=1}^{n} P(w_i|t_i)$$

So, $$P(W|T) * P(T) = \prod_{i=1}^{n} P(w_i|t_i) \times P(t_1) * P(t_2|t_1) * P(t_3|t_1 t_2) * \ldots P(t_n|t_1 \ldots t_{n-1})$$

$$P(T) = P(t_1) \times P(t_2|t_1) \times P(t_3|t_1 t_2) \times \ldots \times P(t_n|t_{n-2} t_{n-1})$$

Hence, $P(T|W)$ can be estimated as

$$P(W|T) * P(T) = \prod_{i=1}^{n} P(w_i|t_i) \times P(t_1) \times P(t_2|t_1) * P(t_3|t_2 t_1) \times \ldots \times P(t_n|t_{n-2} t_{n-1})$$

$$= \prod_{i=1}^{n} P(w_i|t_i) \times P(t_1) \times P(t_2|t_1) \times \prod_{i=3}^{n} P(t_i|t_{i-2} t_{i-1})$$

we estimate these probabilities from relative ...
via maximum likelihood estimation.

$$p(t_i|t_{i-2}t_{i-1}) = \frac{c(t_{i-2}, t_{i-1}, t_i)}{c(t_{i-2}, t_{i-1})}$$

$$p(w_i|t_i) = \frac{c(w_i, t_i)}{c(t_i)}$$

where $c(t_{i-2}, t_{i-1}, t_i)$ is the number of occurences of t_i, followed by t_{i-2}, t_{i-1}.

(b)

2 (a) Explain the concept of constituency. [04]

Constituency :

- words in a sentence are not tied together as a sequence of part of speech.
- Language puts constraints on word order.

ex:
Certain words go together with each other m... than with others and seem to behave as a uni...

- The fundamental idea of syntax is that wor... group together to form constituents, each of wh... acts as a single unit.
- They combine constituents to form larger constituents,
- For example, they can all function as the subject or the object of a verb.
- These constituents combine with others to form a sentence constituent.

Eg: The noun phrase, The bird can combine with the verb phrase, flies to form the sentence, The bird flies.

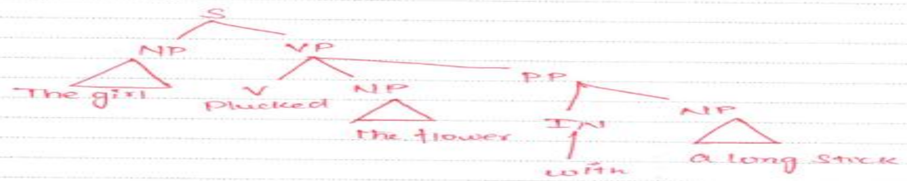
- Different types of phrases have different internal structures.

i) Phrase level constructions
ii) Sentence level construction.

Phrase Level Constructions :

- Fundamental notion in natural language is that certain group of words behave as constituents.
- These constituents are identified by their abilit... to occur in similar contexts.
- Simplest way to decide whether a group of wo... is a phrase, is to see if it can be substituted with some other group of words without changing the meaning. So, if such a substitution is possible then the set of ...
This is called the substitution

— If the head is a noun _ the phrase is called Noun phrase.
If the head is a verb — the phrase is called Verb phrase



Noun Phrase:

— A noun phrase is a phrase whose head is a noun or a pronoun, accompained by a set of modifiers.

— It can function as subject, object or complement.

— The modifiers of a noun phrase can be determiners or adjective phrase.

Verb phrase

Analogous to the noun phrase, is the verb phrase, which is headed by a verb.

- There are wide ranges of phrases that can modify a verb. This makes verb phrases a bit more complex.

Prepositional phrase

Prepositional phrases are headed by a preposition. They consist of a preposition, possibly followed by some other constituent, usually a noun phrase.

We played Volleyball on the beach.

we can have a preposition phrase that consists of just a preposition.

John went Outside.

Adjective phrase:

The head of an adjective phrase is an adjectiv

AP consists of an adjective, which may be preced by an adverb and followed by a PP.

ex:

Ashish is clever

The train is very late

My sister is fond of animals.

Adverb phrase:

- An adverb phrase consists of an adverb, pos
preceded by a degree adverb. Here is an example.

Time passes very quickly

(Deg...) Adv.

The four commonly known structures are
* Declarative structure
* Imperative structure
* Yes-No question structure
* wh-question structure.

(*) Sentences with a declarative structure have a subject followed by an predicate.
The subject of a declarative sentence is a noun phrase and predicate is verb phrase
eg: I like horse riding.
The phrase structure rule for declarative sentences is

S → NP VP

(*) Sentences with an imperative structure usually begin with a verb phrase and lack subject. The subject of these types of sentences implicit and is understood to be 'you'. These types of sentences are used for commands +

(*) Sentences with wh-question structure are mor complex. These sentences begin with a wh-words.
who, which, where, what, why and how.
- It have a wh-phrase, as a subject or may include another subject.

Which team won the match?

S → wh-NP VP
- another type involves more than one NP. the auxiliary verb comes before the subject NP.
... show me in your sho

(b) Describe the concept of probabilistic parsing. [03]

Probabilistic Parsing

- Statistical parser like stastical tagging requires a corpus of hand parsed text.
- Penn tree bank is one of the corpora.
- The Penn-tree bank is a large corpus of articles from the Wall Street Journal have been tagged with Penn tree-bank tags and then parsed according to a sim set of phrase structure rules conforming to Chomsky government and binding syntax.
- The parse trees or sentences are represented in the form of bracketed trees.

$$\phi' = \operatorname{argmax}_{\phi \in \tau(s)} P(\phi|s)$$

$$= \operatorname{argmax}_{\phi \in \tau(s)} P(\phi, s)$$

$$= \operatorname{argmax}_{\phi \in \tau(s)} P(\phi)$$

=> PCFG is a CFG in which every rule is ~~αep~~ assigned a Probability (Charniak 1993). It extends the CFG by augmentin each rule $A \rightarrow \alpha$ in set of Productions P, with a conditional Probability P:

$$A \longrightarrow \alpha\ [P]$$

gives the probability of expanding a constituent using rule. $A \rightarrow \alpha$.

Ex:

$P(S \rightarrow NP\ VP) + P(S \rightarrow VP) = 1$

$P(NP \rightarrow Det\ Noun) + P(NP \rightarrow Noun) + P(NP \rightarrow pronoun) + P(NP \rightarrow Det\ Noun\ PP) =$

$P(VP \rightarrow Verb\ NP) + P(NP \rightarrow Verb) + P(VP \rightarrow VP\ PP) = 1.0$

$P(Det \rightarrow this) + P(Det \rightarrow that) + P(Det \rightarrow a) + P(Det \rightarrow The) = 1.0$

$P(Noun \rightarrow paint) + P(Noun \rightarrow door) + P(Noun \rightarrow bird) + P(Noun \rightarrow hole) = 1.0$

The MLE estimate for a rule $A \rightarrow \alpha$ is given by the expression.

$$P_{MLE}(A \rightarrow \alpha) = \frac{Count(A \rightarrow \alpha)}{\sum_{\alpha} Count(A \rightarrow \alpha)}$$

Rule	Count(A→α)	Count A	MLE estimate
S → VP	2	2	1
NP → Det Noun PP	1	4	0.25
NP → Det Noun	3	4	0.75
VP → Verb NP	2	3	0.66
VP → VP PP	1	3	0.33
Det → The	2	2	1
Noun → hole	2	4	0.5
Noun → door	2	4	0.5
Prep → with	1	1	1
Verb → Paint	1	1	1

Paint the door with the hole

$P(t_1) = 0.2 * 0.5 * 0.2 * 0.2 * 0.35 * 0.25 * 1.0 * 0.25 * 0.4 * 0.35 * 0.25$

$= 0.0000030625$

$P(t_2) = 0.2 * 0.2 * 0.5 * 0.2 * 0.4 * 0.35 * 0.25 * 1 * 0.2 * 0.4 * 0.35 * 0.25$

$= 0.000001225$

The first tree has a higher probability leading to correct interpretation.

Probabilistic CYK algorithm

Initialization:

for i = 1 to n do
 for all rules $A \rightarrow w_i$ do
  $\varphi[i, 1, A] = P(A \rightarrow w_i)$

Recursive step:

for j = 2 to n do
 for i = 1 to n-j+1 do
 begin
  $\varphi[i, 1, A] = \varphi$

Problems with PCFG

① Independence assumption. We calculate the probability of a parse tree assuming that the rules are independent of each other.

But if a node wants to expand depends on its location in the parse tree.

eg: pronoun occurs more frequently as subject rather than object.

3. Explain briefly about Earley parser with an algorithm and example [10]

Earley Parser:

- Implements an efficient parallel top down search using dynamic programming.

- It builds a table of sub-trees for each of the constituents in the input. This way, the algorithm eliminates the repetitive parse of a constituent which arises from backtracking, and successfully reduces the exponential-time problem to polynomial time.

- The Earley parser can handle recursive rules such as $A \rightarrow Ac$ without getting into an infinite loop.

The algorithm uses three operations to process states in the chart.

1) Predictor
2) Scanner
3) Completer

The algorithm sequentially constructs the sets for each of the n+1 chart entries. chart[0] is initialized with a dummy state $S' \rightarrow \cdot S, [0,0]$

Earley Parsing:

Input : Sentence and the Grammar

Output : Chart

```
Chart [0] ← S' → S [0,0]
n ← length (sentence) // no of words in the sentence
for i=0 to n do
    for each state in chart [i] do
    if (incomplete (state) and next category is not a part of
                    speech) then
        Predictor (state)
    else if (incomplete (state) and next category is a
                    part of speech)
        Scanner (state)
    else
        Completer (state)
    end-if
    end-if
    end for
end for
return
Procedure Predictor (A → X1 .... •B ... Xm [i,j])
for each rule (B → α) in G do
insert the state B → •α [i,j] to chart [j]
End
Procedure Scanner (A → X1 .... •B ... Xm [i,j])
If B is one of the part of speech associated with word [j] the
```

Predictor (state) and next category is a
else if (incomplete (state) and next category is a
end for
return
Procedure Predictor (A → X1 .... •B ... Xm [i,j])

Insert the State $B \rightarrow word[j] \bullet [j, j+1]$ to chart [j+1]
End
Procedure Completer ($A \rightarrow X_1 \ldots \bullet [j, k]$)
for each $B \rightarrow X_1 \ldots \bullet A \ldots [i, j]$ in chart [j] do
Insert the State $B \rightarrow X_1 \ldots A \bullet [i, k]$ to Chart [k]
End.

Predictor

Generates new states representing potential expansion of the non-terminal in the left most derivation. A predictor is applied to every stat that has a non-terminal to the right of the dot, wh the category of that non-terminal is different from part-of-speech.

If $A \rightarrow X_1 \ldots \bullet B \ldots X_m [i, j]$

Then for every rule of the form $B \rightarrow \alpha$, the operation adds to chart [i, j] the State

$B \rightarrow \bullet \alpha, [j, j]$

Scanner

It's used when a state has part-of speech category to the right of the dot. The scanner examines the i/p to see if the part-of speech appearing to the right of the dot matches one of the part-of-speech associated with the current input.

Completer

The completer is used when the dot reaches the right end of the rule. The presence of such a state signifies successful completion of the parse of some grammatical category. The completer identifies all previously generated states that expect this grammatical category at this positio in the input and creates new state by advancing the dots over the expected category.

4(a) Explain Hybrid tagger and unknown words in parts of speech tagging. [04]

Unknown words:

- Unknown words are words that donot ap in dictionary or a training corpus.
- They create a problem during tagging.
- There are several potential solutions to problem:

(1) One is to assign the most frequent tag (which occurs with word types in the training corpus). to the unknown word.

(ii) Another solution is to assume that the unk words can be a any part of speech and initialize them by assigning them open class tags.

- Then proceed to disambiguate them using the probablities of those tags.
- In this approach, the unknown word is assigned a tag based on the probablity of the word belonging to a specific part of speech in the trainin corpus having the same suffix or prefix.

5 (a) Describe the concept of spelling error detection and correction by computing minimum edit distance algorithm between PEACEFUL and PACEFLU

Spelling Error Detection & Correction:

- In computer-based information s/m, errors of typing and spelling causes variation b/w strings.
- These errors are investigated and that are:

* single character omission
* Insertion
* Substitution
* Reversal ⇒ most common typing mistake.

- Dameau (1964) reported that over 80% of the typ errors were single-error misspellings:

i) Substitution of a single letter
ii) Omission of a " "
iii) Insertion of a " "
iv) Transposition of two adjacent letters.

5. Explain the criteria to evaluate the IR system. [10]

9.7 EVALUATION OF THE IR SYSTEM

The evaluation of IR systems is the process of assessing how system meets the information needs of its users (Voorhees 2001). Ev an IR system is a difficult task involving a number of areas i cognition, statistics, and man-machine interactions. IR evaluation can be broadly classified as system driven models and user- models. System driven models (Cleverdon et al. 1966) measure h a system ranks documents; user-centered models measure user sat Cleverdon listed the following six criteria that can be used for ev

1. *Coverage of the collection*: The extent to which the system
2. *Time lag*: The time that elapses between submission of a qu getting back the response
3. *Presentation format*
4. *User effort*: The effort made by the user to obtain relevant inf
5. *Precision*: The proportion of retrieved documents that are rel
6. *Recall*: The proportion of relevant documents that are retrie

Of these criteria, recall and precision have most frequently bee in measuring IR. Both are related to effectiveness, i.e., the ab system to retrieve relevant documents in response to user query. A of effectiveness measures have been formulated (van Rijsberg We discuss them in the following section. To better under relationship between aspects of retrieval process and different see (Voorhees and Harman 1999), where correlations betwee measures are estimated.

The major goal of IR is to search for documents that are rel user's query. It is necessary to understand what constitutes rel the evaluation of IR systems relies on the notion of relevance.

9.7.1 Relevance

Relevance is subjective in nature (Saracevic 1991), i.e., it depe individual judgements of users. Given a query, the same docu be judged as relevant by one user and non-relevant by anoth possible to measure this 'true relevance' because no human c documents in a collection and provide a relevance assessm evaluations of IR systems have so far been done on test

9.7.2 Effectiveness Measures

Effectiveness is purely a measure of the ability of a system to satisfy the user in terms of the relevance of documents retrieved (Rijsbergen 1979). Aspects of effectiveness include whether the documents returned are relevant to the user, whether they are presented in order of relevance, whether a significant number of relevant documents in the collection are returned to the user, etc. A number of measures have been proposed to quantify effectiveness. As stated earlier, the most commonly used measures of effectiveness are precision and recall. These measures are based on relevance judgments.

Precision and Recall

Precision is defined as the proportion of relevant documents in a retrieved set. This can be seen as the probability that a relevant document is retrieved. *Recall* is the proportion of relevant documents in a collection that have actually been retrieved. Precision measures the accuracy of a system while recall measures its exhaustiveness. Precision and recall can be computed as follows:

$$\text{Precision} = \frac{\text{Number of relevant document retrieved } (NR_{ret})}{\text{Total number of documents retrieved } (N_{ret})}$$

$$\text{Recall} = \frac{\text{Number of relevant documents retrieved } (NR_r)}{\text{Total number of relevant documents in the collection } (NR_{rel})}$$

These definitions ... if all the documents in the collection are relevant (Salton 1983). Fallout is the ratio of non-relevant documents retrieved to non-relevant documents in the collection.

$$\text{Fallout} = \frac{\text{Number of non-relevant documents retrieved } (N_n)}{\text{Number of non-relevant documents in the collection}}$$

For Figure 9.13, the fallout will be computed as

$$\text{Fallout} = \frac{|\bar{A} \cap B|}{|\bar{A}|}$$

The F-measure takes into account both precision and recall. It is defined as the harmonic mean of recall and precision.

$$F = \frac{2PR}{P+R}$$

$$E = \frac{(1+\beta^2)PR}{\beta^2 P + R} = \frac{(1+\beta^2)}{\frac{\beta^2}{R} + \frac{1}{P}}$$

where P is precision, R is recall, and β is the relative importance of P compared to R. The value of β controls the trade-off between precision and recall. Setting β to 1 gives equal weight to precision and recall, resulting in a harmonic mean of recall and precision ($E = F$). $\beta > 1$ gives more weight to precision, and $\beta < 1$ gives more weight to recall.

Swets (1969) developed a model ...

Normalized recall measures how close the set of retrieved documents is to an ideal retrieval, in which the most relevant NR_{rel} document appears in the first NR_{rel} position. Relevant documents are ranked 1, 2, 3, ..., NR_{rel}, where NR_{rel} is the number of relevant documents. The ideal rank is given by

$$\text{IdR} = \frac{\sum_{r=1}^{NR_{rel}} r}{NR_{rel}}$$

Let the average rank (AvR) over the set of relevant documents retrieved by a system be

$$\text{AvR} = \frac{\sum_{r=1}^{NR_{rel}} \text{Rank}_r}{NR_{rel}}$$

Explain in detail about classical information retrieval model.

CLASSICAL INFORMATION RETRIEVAL MODELS

9.4.1 Boolean model

Introduced in the 50s, the Boolean model is the oldest of the three classical models. It is based on Boolean logic and classical set theory. In this model, documents are represented as a set of keywords, usually stored in an inverted file. An inverted file is a list of keywords and identifiers of the documents in which they occur. Users are required to express their queries as a Boolean expression consisting of keywords connected with Boolean logical operators (AND, OR, NOT). Retrieval is performed based on whether or not document contains the query terms.

Given a finite set

$$T = \{t_1, t_2, \ldots, t_i, \ldots, t_m\}$$

of index terms, a finite set

$$D = \{d_1, d_2, \ldots, d_j, \ldots, d_n\}$$

of documents and a Boolean expression—in a normal form—represent a query Q as follows:

$$Q = \wedge(\vee\theta_i),\ \theta_i \in \{t_i, \neg t_i\}$$

The retrieval is performed in two steps:

1. The set R_i of documents are obtained that contain or do not contain the term t_i:

$$R_i = \{d_j \mid \theta_i \in d_j\},\ \theta_i \in \{t_i, \neg t_i\}$$

where $\neg t_i \in d_j$ means $t_i \notin d_j$

2. Set operations are used to retrieve documents in response to Q:

$$\cap R_i$$

9.4.2 Probabilistic Model

The probabilistic model applies a probabilistic framework to IR. It ranks documents based on the probability of their relevance to a given query (Robertson and Jones 1976). Retrieval depends on whether probability of relevance (relative to a query) of a document is higher than that of non-relevance, i.e. whether it exceeds a threshold value. Given a set of documents D, a query q, and a cut-off value α, this model first calculates the probability of relevance and irrelevance of a document to the query. It then ranks documents having probabilities of relevance at least that of irrelevance in decreasing order of their relevance. Documents are retrieved if the probability of relevance in the ranked list exceeds the cut off value.

More formally, if $P(R/d)$ is the probability of relevance of a document d, for query q, and $P(I/d)$ is the probability of irrelevance, then the set of documents retrieved in response to the query q is as follows.

$$S = \{d_j \mid P(R/d_j) \geq P(I/d_j)\}\ P(R/d_j) \geq \alpha$$

accurately. usually too

9.4.3 Vector Space Model

The vector space model is one of the most well-studied retrieval models. Important contribution to its development was made by Luhn (1959), Salton (1968), Salton and McGill (1983), and van Rijsbergen (1977). The vector space model represents documents and queries as vectors of features representing terms that occur within them. Each document is characterized by a Boolean or numerical vector. These vectors are represented in a multi-dimensional space, in which each dimension corresponds to a distinct term in the corpus of documents. In its simplest form, each feature takes a value of either zero or one, indicating the absence or presence of that term in a document or query. More generally, features are assigned numerical values that are usually a function of the frequency of terms. Ranking algorithms compute the similarity between document and query vectors, to yield a retrieval score to each document. This score is used to produce a ranked list of retrieved documents. Given a finite set of n documents

$$D = \{d_1, d_2, \ldots, d_j, \ldots, d_n\}$$

and a finite set of m terms

$$T = \{t_1, t_2, \ldots, t_i, \ldots, t_m\}$$

each document is represented by a column vector of weights as follows:

$$(w_{1j}, w_{2j}, w_{3j}, \ldots, w_{ij}, \ldots, w_{mj})^t$$

where w_{ij} is the weight of the term t_i in document d_j. The document collection as a whole is represented by an $m \times n$ term-document matrix as

$$\begin{pmatrix} w_{11} & w_{12} & \cdots & w_{1j} & \cdots & w_{1n} \\ w_{21} & w_{22} & \cdots & w_{2j} & \cdots & w_{2n} \\ w_{i1} & w_{i2} & \cdots & w_{ij} & \cdots & w_{in} \\ w_{m1} & w_{m2} & \cdots & w_{mj} & \cdots & w_{mn} \end{pmatrix}$$

9.4.4 Term Weighting

Each term that is selected as an indexing feature for a document, acts as a discriminator between that document and all other documents in the corpus. Luhn (1958) attempted to quantify the discriminating power of the terms by associating the frequency of their occurrence (term frequency) within the document. He postulated that the most discriminating (content bearing) terms are mid frequency terms. This postulate can be refined by noting the following facts:

A simple automatic method for obtaining indexed representation of the documents is as follows.

Step 1 Tokenization This extracts individual terms form a document, converts all the letters to lower case, and removes punctuation marks. The output of the first stage is a representation of the document as a stream of terms.

Step 2 Stop word elimination This removes words that appear more frequently in the document collection.

Step 3 Stemming This reduces the remaining terms to their linguistic root, to obtain the index terms.

Step 4 Term weighting This assigns weights to terms according to their importance in the document, in the collection, or some combination of both.

9.4.5 Similarity Measures

Vector space model represents documents and queries as vectors in a multi-dimensional space. Retrieval is performed by measuring the 'closeness' of the query vector to document vector. Documents can then be ranked according to the numeric similarity between the query and the document. In the vector space model, the documents selected are those that are geometrically closest to the query according to some measure. The model relies on the intuitive notion that similar vectors define semantically related documents. Figure 9.4 gives an example of document and query representation in two-dimensional vector space. These dimensions correspond to the two index terms t_1 and t_2. Document d_1 has two occurrences of t_1, document d_2 has one occurrence of t_2, and document d_3 has one occurrence of t_1 and t_2 each.

Explain the three types of non classical model.

The *information logic model* is based on a special logic technique called logical imaging. Retrieval is performed by making inferences from document to query. This is unlike classical models, where a search process is used. Unlike usual implication, which is true in all cases except that when antecedent is true and consequent is false, this inference is uncertain. Hence, a measure of uncertainty is associated with this inference. The principle put forward by van Rijsbergen is used to measure this uncertainty.

The *situation theory* model is also based on van Rijsbergen's principle. Retrieval is considered as a flow of information from document to query. A structure called *infon*, denoted by ι, is used to describe the situation and to model information flow. An infon represents an *n*-ary relation and its polarity. The polarity of an infon can be either 1 or 0, indicating that the infon carries either positive or negative information.

The *interaction IR* model was first introduced in Dominich (1992, 1993) and Rijsbergen (1996). In this model, the documents are not isolated; instead, they are interconnected. The query interacts with the interconnected documents. Retrieval is conceived as a result of this interaction. This view of interaction is taken from the concept of interaction as realized in the Copenhagen interpretation of quantum mechanics.