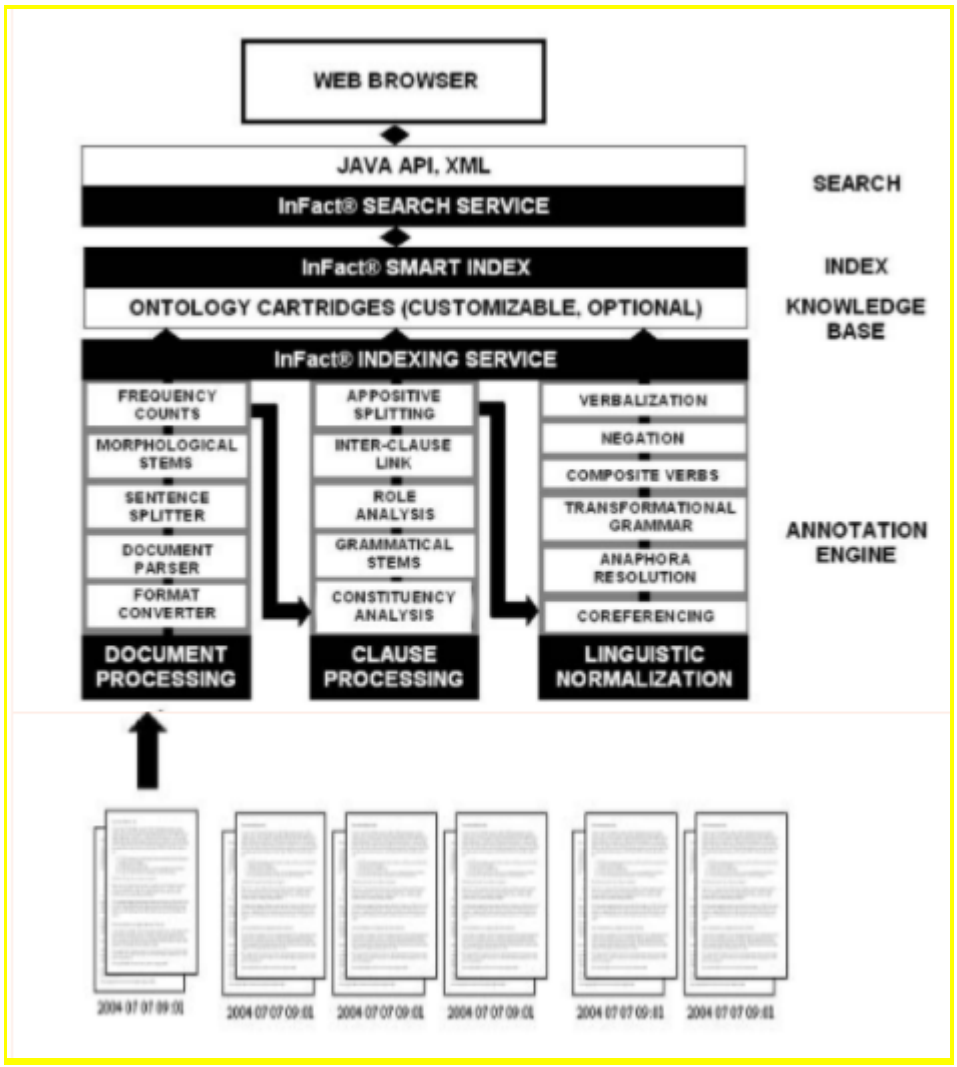
Internal Assessment Test III Solution– Jan 2022

Sub:	NLP	Sub Code:	18CS743

Q.1 With a neat diagram explain a functional overview of the InFact system

Ans:

Functional overview of the InFact System:



- InFact is a natural language search engine
- It combines:
  - the speed of keyword search with the power of natural language processing.
  - Performs clause level indexing
  - offers a full spectrum of functionality that ranges from Boolean keyword operators to linguistic pattern matching

InFact System Overview

- InFact consists of indexing and a search module.

- InFact models text as a complex multivariate object using a unique combination of deep parsing, linguistic normalization, and efficient storage.
- The storage schema addresses the fundamental difficulty of reducing information contained in parse trees into generalized data structures that can be queried dynamically.
- InFact handles the problem of linguistic variation by mapping complex linguistic structures into semantic and syntactic equivalents.

Document Processing

- The first step in document processing is format conversion, which is handled through native format converters,which can convert 370 different input file types.
- Customized document parsers address the issue that a Webpage may not be the basic unit of content, but it may consist of separate sections with an associated set of relationships and metadata.
    - For instance a blog post may contain blocks of text with different dates and topics.
    - The challenge is to automatically recognize variations from a common style template, and segment information in the index to match zones in the source documents, so the relevant section can be displayed in response to a query.
- Next we apply logic for sentence splitting in preparation for clause processing.
- Last, we extract morphological stems and compute frequency counts, which are then entered in the index.

Clause Processing

- The indexing service takes the output of the sentence splitter and feeds it to a deep linguistic parser.
- Indices are created automatically, without using predefined extraction rules, and it captures all information, not just predefined patterns. The parser performs a full constituency and dependency analysis, extracting part-of-speech (POS) tags and grammatical roles for all tokens in every clause.
- Next it captures inter-clause links, through:
    - Explicit tagging of conjunction or pronouns that provide the link between the syntactic structures for two adjacent clauses in the same sentence
    - pointing to the list of annotated keywords in the antecedent and following sentence.

Example:

Sentence: "**Appointed commander of the Continental Army in 1775, George Washington molded a fighting force that eventually won independence from Great Britain"**

- It consists of three clauses, each containing:
- A governing verb (appoint, mold, and win). InFact decomposes it into a
- Primary clause **("George Washington molded a fighting force")**
- and two secondary clauses, which are related to the primary clause by an
- appositive construct **"Appointed commander of the Continental Army in 1775"** and a
- pronoun **"that eventually won independence from Great Britain"**
- Each term in each clause is assigned a syntactic category or POS tag (e.g., noun, adjective, etc.) and a grammatical role tag (e.g., subject, object, etc.).
- InFact then utilizes these linguistic tags to extract relationships that are normalized and stored in an index, as outlined in the next two sections.

Linguistic Normalization

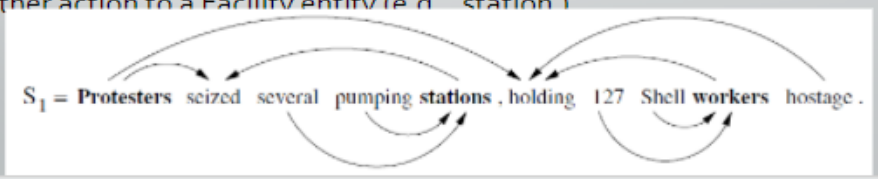
- Apply normalization rules at the syntactic, semantic, or even pragmatic level.
- Our approach to coreferencing and anaphora resolution make use of syntactic agreement and/or binding theory constraints.
- Binding theory places syntactic restrictions on the possible coreference relationships between pronouns and their antecedents.
    - Example:
        - "**John works by himself,**" "himself" must refer to John,
        - "**John bought him a new car,**" "him" must refer to some other individual mentioned in a previous sentence.
        - ""**You have not been sending money," John said in a recent call to his wife from Germany,**"binding theory constraints limit pronoun resolution to first and second persons within a quotation (e.g., you),

Q.2. Write short notes on

a. The shortest path hypothesis

2.1. The Shortest Path Hypothesis

- If *e1* and *e2* are two entities mentioned in the same sentence such that they are observed to be in a relationship *R*. *R*(*e1*, *e2*) is almost exclusively concentrated in the shortest path between *e1* and *e2*.
- If entities *e1* and *e2* are arguments of the same predicate, then the shortest path between them will pass through the predicate, which may be connected directly to the two entities, or indirectly through prepositions. If *e1* and *e2* belong to different predicate-argument structures that share a common argument, then the shortest path will pass through this argument.
- For the first path, it is reasonable to infer that if a Person entity (e.g., 'protesters') is doing some action (e.g., 'seized') to a Facility entity (e.g., 'station'), then the Person entity is Located at that Facility entity.
- The second path captures the fact that the same Person entity (e.g., 'protesters') is doing two actions (e.g., 'holding' and 'seized') , one action to a Person entity (e.g., 'workers'), and the other action to a Facility entity (e.g., 'station')



b. Learning with dependency path

2.2. Learning with Dependency Paths

- A dependency path is represented as a sequence of words interspersed with arrows that indicate the orientation of each dependency.
- "protesters → seized ← stations," "Noun → Verb ← Noun," "Person → seized ← Facility," or "Person → Verb ← Facility." The total number of features generated by this dependency path is 4 × 1 × 3 × 1 × 4.
- We also use an additional suffix '(-)' to indicate a negative polarity item. For example, the phrase "He never went to Paris" is associated with the dependency path "He → went(-) ← to ← Paris."
- The number of common features between ***x*** and ***y*** is computed as:

$$K(\mathbf{x}, \mathbf{y}) = \mathbb{1}(m = n) \cdot \prod_{i=1}^{n} c(x_i, y_i) \qquad (3.4)$$

$$\begin{bmatrix} \text{protesters} \\ \text{NNS} \\ \text{Noun} \\ \text{PERSON} \end{bmatrix} \times [\rightarrow] \times \begin{bmatrix} \text{seized} \\ \text{VBD} \\ \text{Verb} \end{bmatrix} \times [\leftarrow] \times \begin{bmatrix} \text{stations} \\ \text{NNS} \\ \text{Noun} \\ \text{FACILITY} \end{bmatrix}$$

- As an example, let us consider two instances of the Located relationship, and their corresponding dependency paths:

1. '**his** actions in **Brcko**' (**his** → actions ← in ← **Brcko**).
2. '**his** arrival in **Beijing**' (**his** → arrival ← in ← **Beijing**).

Their representation as a sequence of sets of word classes is given by:

1. $\boldsymbol{x}$ = [x_1 x_2 x_3 x_4 x_5 x_6 x_7], where x_1 = {his, PRP, Person}, x_2 = {→}, x_3 = {actions, NNS, Noun}, x_4 = {←}, x_5 = {in, IN}, x_6 = {←}, x_7 = {Brcko, NNP, Noun, Location}
2. $\boldsymbol{y}$ = [y_1 y_2 y_3 y_4 y_5 y_6 y_7], where y_1 = {his, PRP, Person}, y_2 = {→}, y_3 = {arrival, NN, Noun}, y_4 = {←}, y_5 = {in, IN}, y_6 = {←}, y_7 = {Beijing, NNP, Noun, Location}

Based on the formula from Equation 3.4, the kernel is computed as $K(\boldsymbol{x}, \boldsymbol{y}) = 3 \times 1 \times 1 \times 1 \times 2 \times 1 \times 3 = 18$.

Q.3. Explain the functioning of Latent Semantic Analysis (LSA) feedback system

Ans:

Latent Semantic Analysis (LSA) Feedback Systems

- Latent Semantic Analysis uses statistical computations to extract and represent the meaning of words. Meanings are represented in terms of their similarity to other words in a large corpus of documents.
- LSA begins by finding the frequency of terms used and the number of co-occurrences in each document throughout the corpus and then uses a powerful mathematical transformation to find deeper meanings and relations among words.
- When measuring the similarity between text-objects, LSA's accuracy improves with the size of the objects. Hence, LSA provides the most benefit in finding similarity between two documents.
- The method, unfortunately, does not take into account word order; hence, very short documents may not be able to receive the full benefit of LSA.

LSA corpus matrix

To construct an LSA corpus matrix, a collection of documents are selected. A document may be a sentence, a paragraph, or larger unit of text. A term-document frequency (TDF) matrix X is created for those terms that appear in two or more documents. The row entities correspond to the words or terms (hence the W) and the column entities correspond to the documents (hence the D). The matrix is then analyzed using Singular Value Decomposition, that is the TDF matrix X is decomposed into the product of three other matrices:

1. vectors of derived orthogonal factor values of the original row entities W
2. vectors of derived orthogonal factor values of the original column entities D
3. scaling values (which is a diagonal matrix) S

The product of these three matrices is the original TDF matrix.

$$\{X\} = \{W\}\{S\}\{D\}$$

These documents consist of terms, which are represented by term vectors; hence, the document can be represented as a document vector which is computed as the sum of the term vectors of its terms:

$$D_i = \sum_{t=1}^{n} T_{ti}$$

where Di is the vector for the i th document D, Tti is the term vector for the term t in Di, and n is number of terms in D. The similarity between two documents (i.e., the cosine between the two document vectors) is computed as

$$Sim(D1, D2) = \frac{\sum_{i=1}^{d}(D1_i \times D2_i)}{\sum_{i=1}^{d}(D1_i)^2 \times \sum_{i=1}^{d}(D2_i)^2}$$

Q.4. Explain the functioning of the word matching feedback system used in iSTART

Ans:

Word Matching Feedback Systems

Word matching is a very simple and intuitive way to estimate the nature of a self explanation. In the first version of iSTART, several hand-coded components were built for each practice text.

For example, for each sentence in the text, the “important words” were identified by a human expert and a length criterion for the explanation was manually estimated.

Important words were generally content words that were deemed important to the meaning of the sentence and could include words not found in the sentence.

For each important word, an association list of synonyms and related terms was created by examining dictionaries and existing protocols as well as by human judgments of what words were likely to occur in a self-explanation of the sentence. In the sentence “All thunderstorms have a similar life history,” for example, important words are thunderstorm, similar, life, and history. An association list for thunderstorm would include storms, moisture, lightning, thunder, cold, tstorm, t-storm, rain, temperature, rainstorms, and electric-storm. In essence, the attempt was made to imitate LSA.

Literal word matching

Words are compared character by character and if there is a match of the first 75% of the characters in a word in the target sentence (or its association list) then we call this a literal match.

This also includes removing suffix -s, -d, -ed, -ing, and -ion at the end of each words. For example, if the trainee's self-explanation contains 'thunderstorm' it still counts as a literal match with words in the target sentence since the first nine characters are exactly the same. On the other hand, if it contains 'thunder,' it will not get a match with the target sentence, but rather with a word on the association list.

Soundex matching

This algorithm compensates for misspellings by mapping similar characters to the same soundex symbol. Words are transformed to their soundex code by retaining the first character, dropping the vowels, and then converting other characters into soundex symbols. If the same symbol occurs more than once consecutively, only one occurrence is retained.

For example,

'thunderstorm' will be transformed to 't8693698';

'communication' to 'c8368.'

If the trainee's self-explanation contains 'thonderstorm' or 'tonderstorm,' both will be matched with 'thunderstorm' and this is called a soundex match. An exact soundex match is required for short words (i.e., those with fewer than six alpha-characters) due to the high number of false alarms when soundex is used. For longer words, a match on the first four soundex symbols suffices. We are considering replacing this rough and ready approach with a spell-checker.

Q.5. Explain Boolean and Vector space information retrieval model.

Ans:

Boolean Model:

- 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 the document contains the query terms.

Given a finite set

$T = \{ t_1, t_2, ..., t_i, ..., t_m \}$ of index terms,

a finite set

$D = \{ d_1, d_2, ..., d_j, ..., d_n \}$ 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:

$R_i = d_j \mid \theta i \in d_j \quad \theta i \in \{t_i, \neg t_i\}.$

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: $\cap R_i$

Boolean Model - Example 1:

Which plays of Shakespeare contain the words ***Brutus*** *AND* ***Caesar*** but *NOT* ***Calpurnia***?

Document collection: A collection of Shakespeare's work.

	Antony and Cleopatra	Julius Caesar	The Tempest	Hamlet	Othello	Macbeth
Antony	1	1	0	0	0	1
Brutus	1	1	0	1	0	0
Caesar	1	1	0	1	1	1
Calpurnia	0	1	0	0	0	0
Cleopatra	1	0	0	0	0	0
mercy	1	0	1	1	1	1
worser	1	0	1	1	1	0

1 if play contains word, otherwise 0

- So we have a 0/1 vector for each term.
- To answer query: take the vectors for ***Brutus, Caesar*** and ***Calpurnia*** (complemented)
- 110100 *AND* 110111 *AND* 101111 = 100100
- The plays *Antony and cleopatra* and *Hamlet* contain the words Brutus and Caesar but not Calpurnia.

Boolean Model - Example 2

Let the set of original documents be D={D1, D2, D3}

Where

D1 =**Information** retrieval is concerned with the organization, storage, **retrieval** and evaluation of information relevant to user's **query**.

D2 = A user having an **information** needs to formulate a request in the form of **query** written in natural language.

D3 = The **retrieval** system responds by retrieving the document that seems relevant to the **query**.

Let the set of terms used to represent these documents be

T = { information, retrieval, query }

Then, the set D of document will be represented as follows:

D = { d1, d2, d3 }

Where

d1 = { information, retrieval, query }

d2 = { information, query }

d3 = { retrieval, query}

Let the Query Q be, Q = **information ^ retrieval**

Step 1: The sets R1 and R2 of documents are retrieved in response to Q, where

R1 = { dj | information ∈ dj } = { d1, d2 }

R2 = { dj | retrieval ∈ dj } = { d1, d3 }

Step 2: Then, the following documents are retrieved in response to the query Q

{ dj | dj ∈ R1 ∩ R2 } = { d1 }

This results in the retrieval of the original document D1 that has the representation d1.

Boolean Model - Advantages and Disadvantages

Advantages:

- They are simple, efficient, easy to implement, perform well in terms of recall and precision if the query is well formulated.

Disadvantages:

1. The model fails to retrieve documents that are only partly relevant to user query.
2. Boolean system cannot rank the retrieved documents.
3. It distinguishes between presence and absence of keywords but fails to assign relevance and importance to keywords in a document.
4. User has to formulate the query in pure boolean expression.

Vector Space Model:

Vector space model

- The vector space model is one of the most well studied retrieval models.
- It represents documents and queries as vectors of features representing terms that occur within them. Each document is characterized by a numerical vector.
- Vectors represented in multidimensional space, in which each dimension corresponds to a distinct term in the corpus of documents.
- Each feature takes a value of either zero or one, indicating the absence or presence of that term in a document or query.
- 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 retrieval documents.

Vector space model

Given a finite set of n documents

$$D = \{d_1, d_2, d_3, \ldots, d_n\}$$

and a finite set of m terms

$$T = \{t_1, t_2, t_3, \ldots, t_m\}$$

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

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

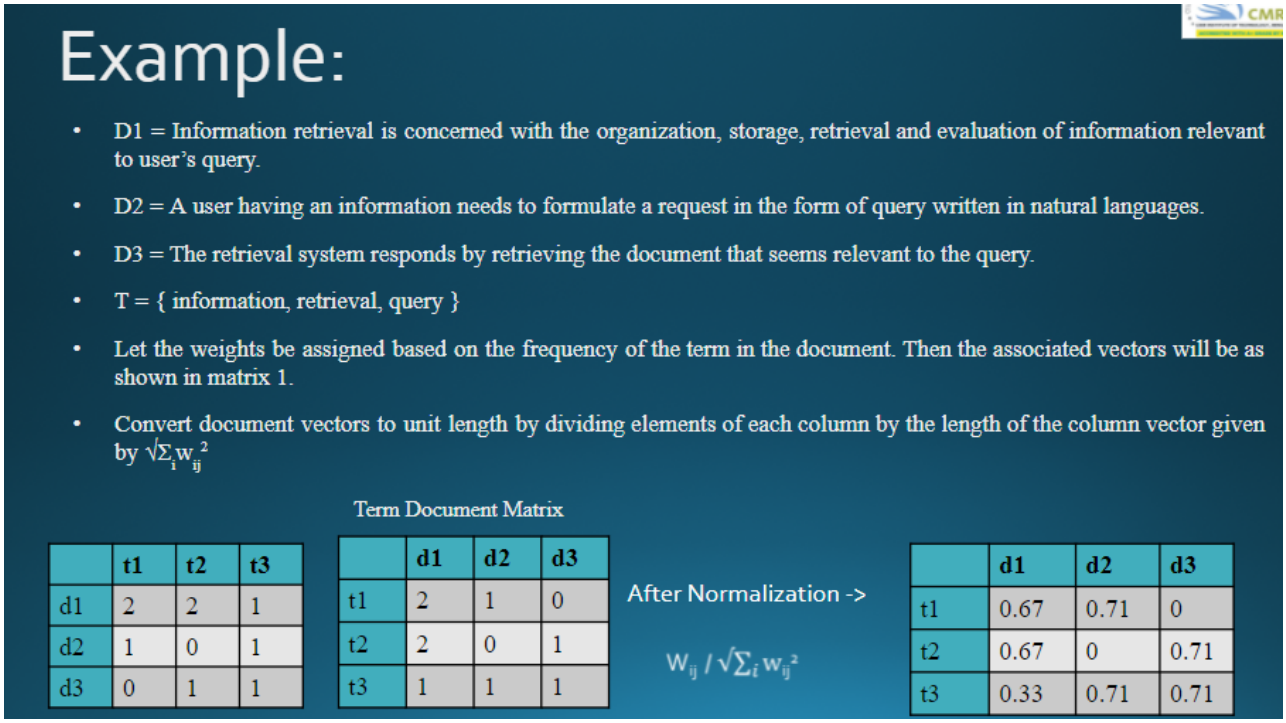
Where W_{ij} is the weight of the term t_i in document d_j.

The document collection as a whole is represented by a *m x n* term - document matrix -

Term space

Document space

$$\begin{Bmatrix} w_{11} & w_{12} & \cdots & w_{1j} & \cdots & w_{1n} \\ w_{21} & w_{22} & \cdots & w_{2j} & \cdots & w_{2n} \\ w_{31} & w_{32} & \cdots & w_{3j} & \cdots & w_{3n} \\ w_{m1} & w_{m2} & \cdots & w_{mj} & \cdots & w_{mn} \end{Bmatrix}$$



Q. 6. Explain WordNet and list the applications of WordNet.

Ans:

- WordNet is a large lexical database for the English language.
- Inspired by psycholinguistic theories, it was developed and is being maintained at the Cognitive Science Laboratory, Princeton University, under the direction of George A. Miller.
- WordNet consists of three databases
  - One for nouns
  - One for verbs
  - One for both adjectives and adverbs
- Information is organized into sets of synonymous words called synsets, each representing one base concept.
- The synsets are linked to each other by means of lexical and semantic relations.
- Lexical relations occur between word-forms (senses) and semantic relations between word meanings.
- These relations include synonymy, hypernymy / hyponymy, antonymy, meronymy / holonymy, troponymy, etc.
- A word may appear in more than one synset and in more than one part-of-speech. The meaning of the word is called sense.
- WordNet lists all senses of a word, each sense belonging to a different synset.
- WordNet's sense-entries consist of a set synonyms and a gloss.
- WordNet is freely and publicly available for download from http://wordnet.princeton.edu/obtain

- WordNet for other languages have been developed, Example, **EuroWordNet** and **Hindi WordNet**
- **EuroWordNet** covers European languages, including English, Dutch, Spanish, Italian, German, French, and Czech.
- **Hindi WordNet** has been developed by CFILT (Resource Center for Indian Language Technology Solutions), IIT Bombay.
- Hindi WordNet can be obtained from the URL http://www.cfilt.iitb.ac.in/wordnet/webhwn/
- CFLIT has also developed a **Marathi WordNet** http://www.cfilt.iitb.ac.in/wordnet/webmwn/wn.php
-

Applications of WordNet:

- **Concept identifications in Natural language**
  - WordNet can be used to identify concepts pertaining to a term, to suit them to the full semantic richness.
- **Word sense disambiguation**
  - It offers
    - sense definitions of words
    - identifies synsets of synonyms
    - Defines a number of semantic relations
- **Automatic Query Expansion**
  - WordNet semantic relations can be used to expand queries so that the search for a document is not confined to the pattern-matching of query terms, but also covers synonyms.
- **Document structuring and categorization**
  - The semantic information extracted from WordNet have been used for text categorization.
- **Document summarization**
  - The approach presented by Barzilay and Elhadad uses information from WordNet to compute lexical chains.