3.3.6 The Verb Phrase and Subcategorization

- English Verb Phrase consist of a head verb along with 0 or more following constituents which we'll call ***arguments***. In the simple rules we have built so far, these other constituents include NP's and PP's and combinations of the two:

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

- But even though there are many valid VP rules in English, not all verbs are allowed to participate in all those VP rules. We can subcategorize the verbs in a language according to the sets of VP rules that they participate in. This is a modern take on the traditional notion of transitive/intransitive.

4.1.3 Elements of Lexical Semantic Analysis

Followings are some important elements of lexical semantic analysis –

1. **Hyponyms :** They are specific lexical items of a general lexical item (hypernym) e.g., apple is a hyponym of fruit (hypernym).
2. **Meronomy :** It is a logical arrangement of text and words that represent a part of or member of something e.g., a segment of an apple
3. **Polysemy :** It a relationship between the meanings of words or phrases, although slightly different, they share a common basic meaning e.g. I read a book, and I wrote a book
4. **Synonyms :** They are words that have the same sense or mostly the same meaning as another, e.g., sad, unhappy, depressed, etc.
5. **Antonyms :** They words that have close to opposite meanings e.g., good, bad
6. **Homonyms :** They are two words that sound the same and are spelled alike but have a different meaning e.g., right (correct), right (turn)

1. Homonymy

- They are lexemes that share a form, but unrelatedmeanings.
- Homonyms are words that have a same syntax or same spelling or same form but their meaning are different and unrelated to each other.
- Two or more words become homonyms if they either sound the same (homophones), have the same spelling (homographs), or if they both homophones and homographs, but do not have related meanings.

- Examples:

  i) bat (wooden stick thing) vs bat (flying scary mammal)

  ii) bank (financial institution) vs bank (riverside)

2. Polysemy

- Polysemy refers to words or phrases with different, but related meanings.
- A word becomes polysemous if it can be used to express different meanings.
- The difference between these meanings can be obvious or subtle.
- It is sometimes difficult to determine whether a word is polysemous or not because the relations between words can be vague and unclear.
- But, examining the origins of the words can help to decide whether a word is polysemic or homonymous.
- The following sentences contain some examples of polysemy.

  i) He drank a glass of milk.

  i) He forgot to milk the cow.

  iii) The enraged actor sued the newspaper.

  iv) He read the newspaper.

4. Antonymy

- Antonyms are words that have opposite or contrasting meanings.
- For example, the antonym of hot is cold; similarly, the antonym of day is night.
- Antonyms are actually the opposite of synonyms.
- Furthermore, there are three types of antonyms as gradable, complementary, and relational antonyms.
- Gradable antonyms are pairs of words with opposite meanings that lie on a continuous spectrum.
- For example, if we take age as a continuous spectrum, young and old are two ends of the spectrum.
- Complementary antonyms are pairs of words with opposite meanings that do not lie on a continuous spectrum.
- For example, interior: exterior, true: false, and inhale: exhale
- Relational antonyms are pairs of words that refer to a relationship from opposite points of view.
- For example, doctor: patient, husband: wife, teacher: student, sister: brother.

5. Hypernymy and Hyponymy

- Hyponym is the sense which is a subclass of another sense
  - i) car is a hyponym of vehicle
  - ii) dog is a hyponym of animal
  - iii) mango is a hyponym of fruit
- Hypernym is the sense which is a superclass
  - i) vehicle is a hypernym of car
  - ii) animal is a hypernym of dog
  - iii) fruit is a hypernym of mango
- In simpler terms, a hyponym is in a type-of relationship with its hypernym.
- Hypernyms and hyponyms are asymmetric.
- Hyponymy can be tested by substituting X and Y in the sentence "X is a kind of Y" and determining if it makes sense.
- For example, "A screwdriver is a kind of tool" makes sense, but not "A tool is a kind of screwdriver".
- Strictly speaking, the meaning relation between hyponyms and hypernyms applies to lexical items of the same word class (or parts of speech), and holds between senses rather than words.
- Hyponymy is a transitive relation, if X is a hyponym of Y, and Y is a hyponym of Z, then X is a hyponym of Z.
- For example, violet is a hyponym of purple and purple is a hyponym of color; therefore, violet is a hyponym of color.
- A word can be both a hypernym and a hyponym: for example, purple is a hyponym of color but itself is a hypernym of the broad spectrum of shades of purple between the range of crimson and violet.
- The hierarchical structure of semantic fields can be mostly seen in hyponymy.
- They could be observed from top to bottom, where the higher level is more general and the lower level is more specific.
- Hyponymy is the most frequently encoded relation among synsets used in lexical databases such as WordNet.
- These semantic relations can also be used to compare semantic similarity by judging the distance between two synsets and to analyse anaphora.

3.3.3(A)Top-down parsing

A top-down is goal oriented. Parser starts root and then with a list of constituents to be built. It rewrites the goals in the goal list by matching one against the LHS of the grammar rules, and expanding it with the RHS in attempting to match the sentence to be derived. If a goal can be rewritten in several ways, then there is a choice of which rule to apply (search problem). For Top-down parsing we can use depth-first or breadth-first search, and goal ordering.

Top-down parsing example (Breadth-first) :

S	$\rightarrow$	$NP\ VP$	Det	$\rightarrow$	$that \mid this \mid a \mid the$
S	$\rightarrow$	$Aux\ NP\ VP$	$Noun$	$\rightarrow$	$book \mid flight \mid meal \mid man$
S	$\rightarrow$	VP	$Verb$	$\rightarrow$	$book \mid include \mid read$
NP	$\rightarrow$	$Det\ NOM$	Aux	$\rightarrow$	$does$
NOM	$\rightarrow$	$Noun$			
NOM	$\rightarrow$	$Noun\ NOM$			
VP	$\rightarrow$	$Verb$			
VP	$\rightarrow$	$Verb\ VP$			

Fig. 3.3.5: *L0* Grammar rule

Example sentence: Book that flight.

From Fig. 3.3.5 we can understand that *S* is start symbol of tree, which will be expand for *NP VP*, *Aux NP VP* and *VP*. Further as we expand using breadth first search shown in Fig. 3.3.6 for finding sequence for "book that flight" as *Verb Determiner Nominal Noun*.

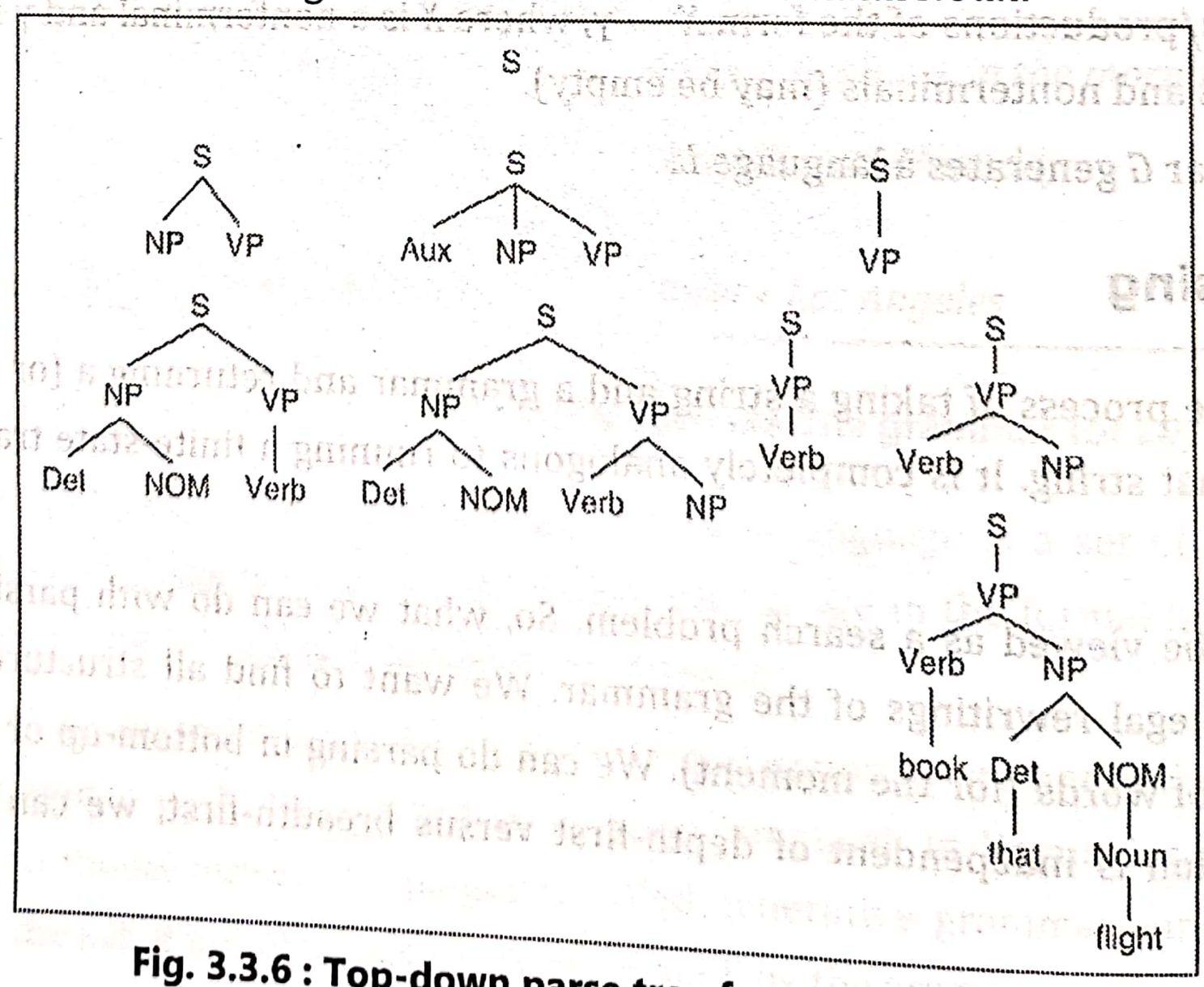


Fig. 3.3.6 : Top-down parse tree for given sentence

There are some problems with top-down parsing. In case of left recursive rules. e.g.*NP* → *NP PP* can be lead to infinite recursion. Top-down is sometimes waste of time for rewriting parts of speech (preterminals) with words (terminals). In practice that is always done bottom-up as lexical lookup.

4.4 WordNet

4.4.1 WordNet and Synsets

- Wordnet is a big collection of words from the English language that are related to each other and are grouped in some way.
- It is also called as a lexical database.
- In other words, WordNet is a database of English words that are connected together by their semantic relationships.
- It is like a superset dictionary with a graph structure.
- WordNet groups nouns, verbs, adjectives, etc. which are similar and the groups are called synsets or synonyms.
- In a wordnet a group of synsets may belong to some other synset.
- For example, the synsets "Stones" and "cement" belong to the synset "Building Materials" or the synset "Stones" also belongs to another synset called "stonework".
- In the example given, stones and cement are called hyponyms of synset building materials and also the synsets building materials and stonework are called synonyms.
- Every member of a synset denotes the same concept but not all synset members are interchangeable in context.
- The membership of words in multiple synsets or concepts mirrors polysemy or multiplicity of meaning.

I received a Gift present.

4. WordNet

WordNet is a large lexical database of English. Nouns, verbs, adjectives and adverbs are grouped into sets of cognitive synonyms (synsets), each expressing a distinct concept. Synsets are interlinked by means of conceptual-semantic and lexical relations. The resulting network of meaningfully related words and concepts can be navigated with the browser. WordNet is also freely and publicly available for download. WordNet's structure makes it a useful tool for computational linguistics and natural language processing.

WordNet superficially resembles a thesaurus, in that it groups words together based on their meanings. However, there are some important distinctions. First, WordNet interlinks not just word forms—strings of letters—but specific senses of words. As a result, words that are found in close proximity to one another in the network are semantically disambiguated. Second, WordNet labels the semantic relations among words, whereas the groupings of words in a thesaurus does not follow any explicit pattern other than meaning similarity.

Structure

The main relation among words in WordNet is synonymy, as between the words shut and close or car and automobile. Synonyms--words that denote the same concept and are interchangeable in many contexts--are grouped into unordered sets (synsets). Each of WordNet's 117 000 synsets is linked to other synsets by means of a small number of "conceptual relations." Additionally, a synset contains a brief definition ("gloss") and, in most cases, one or more short sentences illustrating the use of the synset members. Word forms with several distinct meanings are represented in as many distinct synsets. Thus, each form-meaning pair in WordNet is unique.

There are only few adverbs in WordNet (hardly, mostly, really, etc.) as the majority of English adverbs are straightforwardly derived from adjectives via morphological affixation (surprisingly, strangely, etc.)

Applications of WordNet

WordNet has found numerous applications in problems related with IR and NLP. Some of these are given below:

Concept Identification in Natural Language: WordNet can be used to identify concepts pertaining to a term, to suit them to the full semantic richness and complexity of a given information need.

Word Sense Disambiguation: WordNet combines features of a number of the other resources commonly used in disambiguation work. It offers sense definitions of words, identifies synsets of synonyms, defines a number of semantics relations and is freely available. This makes it the (currently) best known and most utilized resource for word sense disambiguation.

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 Summarization: WordNet has found useful application in text summarization. Few approaches utilize information from WordNet to compute lexical chains.

4.5.1 Word-sense Disambiguation(WSD)

- Word-sense disambiguation (WSD) is a well-known problem in NLP.
- WSD is used in identifying what the sense of a word means in a sentence when the word has multiple meanings.
- When a single word has multiple meaning, then for the machine it is difficult to identify the correct meaning and to solve this challenging issue we can use the rule-based system or machine learning techniques.
- WSD is a natural classification problem: Given a word and its possible senses, as defined by a dictionary, classify an occurrence of the word in context into one or more of its sense classes.

- The features of the context such as neighbouring words provide the evidence for classification.
- A famous example is to determine the sense of pen in the following passage.

"Little John was looking for his toy box. Finally, he found it. The box was in the pen. John was very happy."

WordNet lists five senses for the word pen:

- pen — a writing implement with a point from which ink flows.
- pen — an enclosure for confining livestock.
- playpen, pen — a portable enclosure in which babies may be left to play.
- penitentiary, pen — a correctional institution for those convicted of major crimes.
- pen — female swan.

- There are four conventional approaches to WSD :
  1. **Dictionary- and knowledge-based methods :** These rely primarily on dictionaries, thesauri, and lexical knowledge bases, without using any corpus evidence.
  2. **Supervised methods :** These make use of sense-annotated corpora to train from.
  3. **Semi-supervised or minimally-supervised methods :** These make use of a secondary source of knowledge such as a small annotated corpus as seed data in a bootstrapping process, or a word-aligned bilingual corpus.
  4. **Unsupervised methods :** These eschew (almost) completely external information and work directly from raw unannotated corpora. These methods are also known under the name of word sense discrimination.

4.5.4 Difficulties in WSD

Followings are some difficulties faced by word sense disambiguation (WSD) :

1. Differences between dictionaries

a) The major problem of WSD is to decide the meaning of the word because different contextual meaning can be very closely related.

b) Even different dictionaries and thesauruses can provide different parts of words into meanings.

2. Inter-judge variance

a. One problem of WSD is that WSD systems are generally tested by having their results of a task compared against the task of human beings.

b. This is called the problem of inter-judge variance.

3. Different algorithms for different applications

a. Another problem of WSD is that a different algorithm may be needed for different applications.

b. For example, in machine translation, it takes the form of target word selection; and in information retrieval, a sense inventory is not required.

4. Word-sense discreteness

One of the difficulty in WSD is that words cannot be easily divided into discrete sub meanings.

4.5.5 Applications of WSD

Word sense disambiguation (WSD) is applied in almost every application of language technology.

Followings are some applications of WSD :

1. Machine Translation :

- Machine translation is the most common application of WSD.
- In machine translation, Lexical choice for the words that have different translations for different senses, is done by WSD.
- The senses in machine translation are represented as words in the target language.

2. Text Mining and Information Extraction (IE)

- In most of the text mining and IE, WSD is necessary for accurate analysis of text.
- WSD helps in flagging of the correct words.
- For example, email intelligent system might need flagging of "http links" rather than "https links"

3. Information Retrieval (IR)

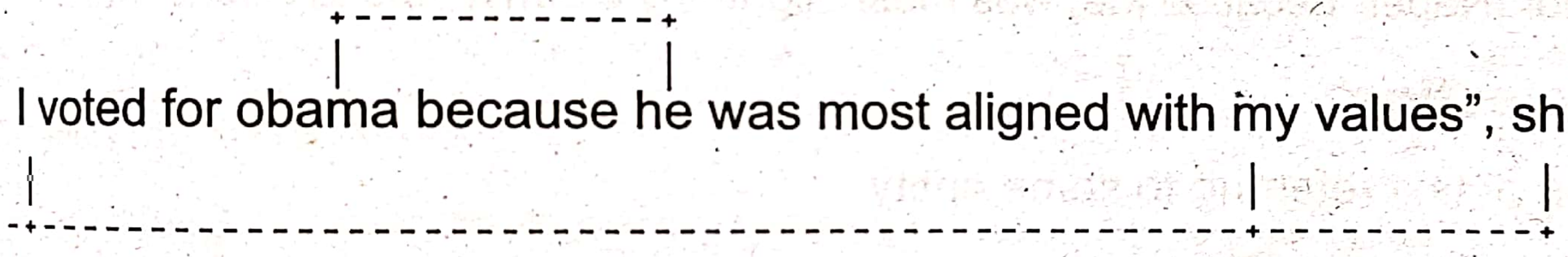
- Information retrieval (IR) is defined as a software that deals with the organization, storage, retrieval and evaluation of information from document repos particularly text based information.
- The system mostly assists users in finding the information they require but it does not particularly return the answers of the questions.
- WSD is used to solve the ambiguities of the queries provided to IR system.
- Similar to machine translation, current IR systems do not especially use WSD module and they depend on the concept that user would type enough context in the query to only retrieve relevant documents.

4. Lexicography

- WSD and lexicography can work together in loop because modern lexicography is collection based.
- With lexicography, WSD provides rough actual sense groupings as well as statistically significant contextual measure of sense.

Coreference resolution

Coreference resolution is the task of clustering mentions in text that refer to the same underlying real-world entities. Coreference resolution, is the task of finding all expressions that are coreferent with any of the entities found in a given text. Coreference resolution is the task of resolving noun phrases to the entities that they refer to. Coreference resolution finds the mentions in a text that refer to the same real-world entity. For example, in the sentence, "Andrew said he would buy a car" the pronoun "he" refers to the same person, namely to "Andrew".



"I", "my", and "she" belong to the same cluster and "Obama" and "he" belong to the same cluster.

In computational linguistics, coreference resolution is a well-studied problem in discourse. To derive the correct interpretation of a text, or even to estimate the relative importance of various mentioned subjects, pronouns and other referring expressions must be connected to the right individuals. Algorithms intended to resolve coreferences commonly look first for the nearest preceding individual that is compatible with the referring expression. For example, she might attach to a preceding expression such as the woman or Anne, but not to Bill. Pronouns such as himself have much stricter constraints. Algorithms for resolving coreference tend to have accuracy in the 75% range. As with many linguistic tasks, there is a trade-off between precision and recall.

A classic problem for coreference resolution in English is the pronoun it, which has many uses. It can refer much like he and she, except that it generally refers to inanimate objects (the rules are actually more complex: animals may be any of it, he, or she; ships are traditionally she; hurricanes are usually it despite having gendered names). It can also refer to abstractions rather than beings: "He was paid minimum wage, but didn't seem to mind it." Finally, it also has pleonastic uses, which do not refer to anything specific:

a. It's raining.

b. It's really a shame.

c. It takes a lot of work to succeed.

d. Sometimes it's the loudest who have the most influence.

Approach to coreference resolution

Coreference Resolution in Two Steps

1. *Detect the mentions (easy)*

   "[I] voted for [Nader] because [he] was most aligned with [[my] values]," [she] said

   mentions can be nested!

2. *Cluster the mentions (hard)*

   "[I] voted for [Nader] because [he] was most aligned with [[my] values]," [she] said

Mentions Detection

Mentions : span of text referring to some entity

Three kinds of mentions:

- Pronouns : I, your, it, she, him, etc. Use a part-of-speech tagger
- Named entities : People, places, etc.. Use a NER system
- Noun phrases : "a dog," "the big fluffy cat stuck in the tree" Use a constituency parser

Filter things that look referential but, in fact, are not – e.g.

- geographic names, the United State
- pleonastic "it", e.g. it's 3:45 p.m., it was cold –
- non-referential "it" , "they" , "there" e.g. it was essential, important, is understood, they say, there seems to be a mistake

All noun phrases (both indef. and def.) are considered potential referent candidates. – A referring phrase can also be a referent for a subsequent referring phrases, Example: He had 300 grams of plutonium 239 in his baggage. The suspected smuggler denied that the materials were his. (chain of 4 referring phrases) – All potential candidates are collected in a table collecting feature info on each candidate.

Features

Define features between a referring phrase and each candidate

– Number agreement: plural, singular or neutral

- He, she, it, etc. are singular, while we, us, they, them, etc. are plural and should match with singular or plural nouns, respectively
- Exceptions: some plural or group nouns can be referred to by either it or they IBM announced a new product. They have been working on it …

– Gender agreement:

- Generally animate objects are referred to by either male pronouns he, his) or female pronouns (she, hers)
- Inanimate objects take neutral (it) gender

– Person agreement:

- First and second person pronouns are "I" and "you"
- Third person pronouns must be used with nouns

– Binding constraints

Reflexive pronouns (himself, themselves) have constraints on which nouns in the same sentence can be referred to:

John bought himself a new Ford. (John = himself)

John bought him a new Ford. (John cannot = him)

- Recency

Entities situated closer to the referring phrase tend to be more salient than those further away And pronouns can't go more than a few sentences away

- Grammatical role / Hobbs distance

Entities in a subject position are more likely than in the object position

- Repeated mention

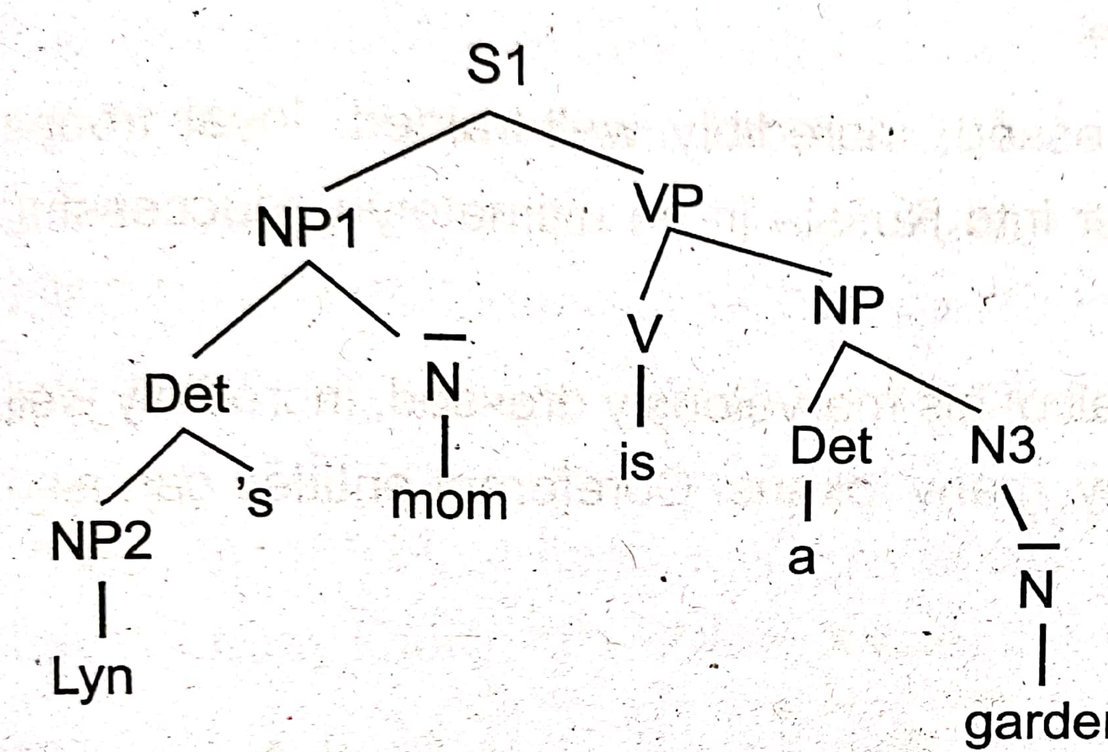
Hobb's Algorithm

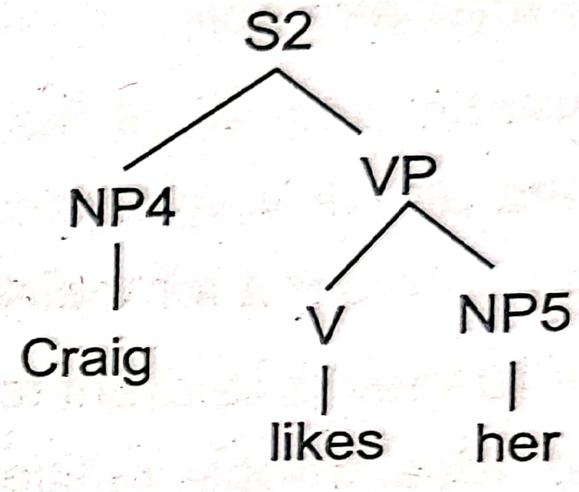
Hobbs' algorithm was one of the earliest approaches to pronoun resolution. The algorithm is mainly based on the syntactic parse tree of the sentences. It makes use of syntactic constraints when resolving pronouns. First, intra-sentential antecedents are proposed - the syntactic tree of the current sentence is searched in a breadth-first left to-right fashion to find antecedents. The contra-indexing constraint is taken care inside the algorithm, by making sure that the path from the NP to the S node of the syntactic tree has at least one another NP on the way. If there are higher-level nodes in the current sentence, then antecedents resulting from a breadth-first left-to-right search of each subtree, are proposed. Then, parse trees of previous sentences in reverse chronological order are searched in the same fashion to propose antecedents. In essence, Hobbs' algorithm prefers entities that are within the same sentence, and entities that are closer pronoun in the same sentence. Depending on the position of the pronoun in the sentence, different entities in a sentence may become more relevant. When looking for antecedents in previous sentences, the antecedents that occur (or are realized) in the subject position are more salient, since a breadth-first left-to-tree search is performed starting at the root S node of the sentence. Depth of a node in the syntactic tree is thus a very important factor to determine discourse prominence. Ref

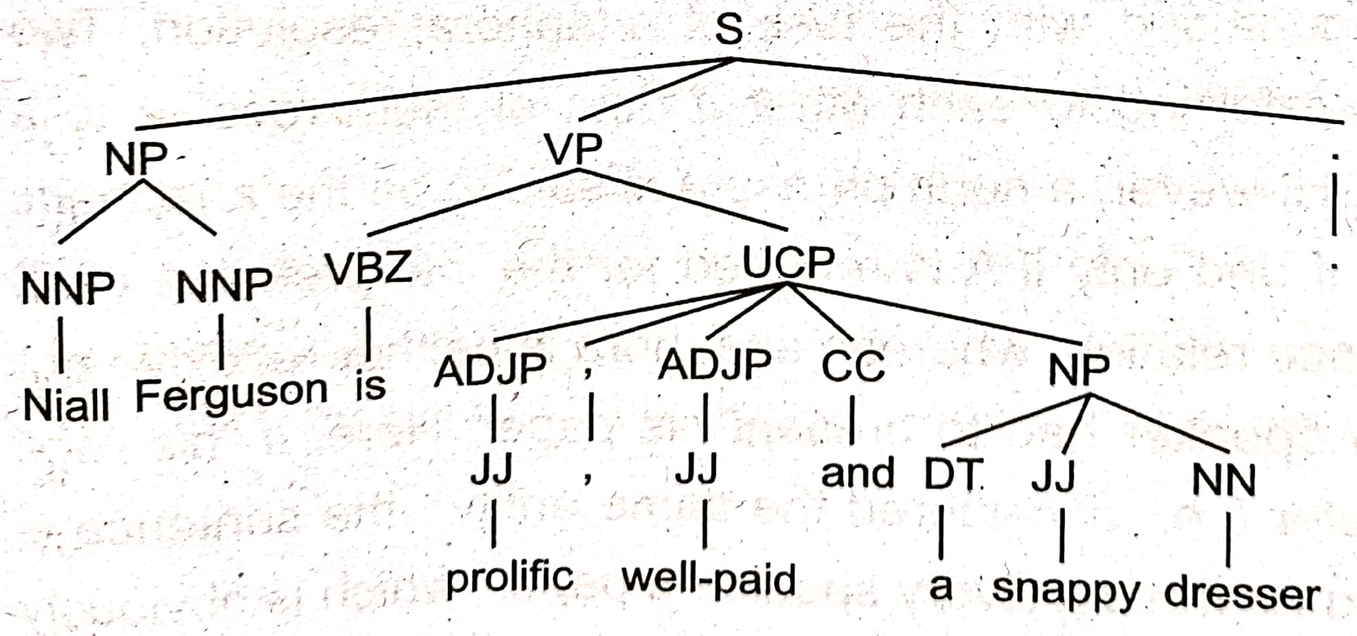
1. Start with target pronoun
2. Climb parse tree to S root
3. For each NP or S
    a. Do breadth-first, left-to-right search of children
    b. Restricted to left of target
    c. For each NP, check agreement with target

4. Repeat on earlier sentences until matching NP found

Example







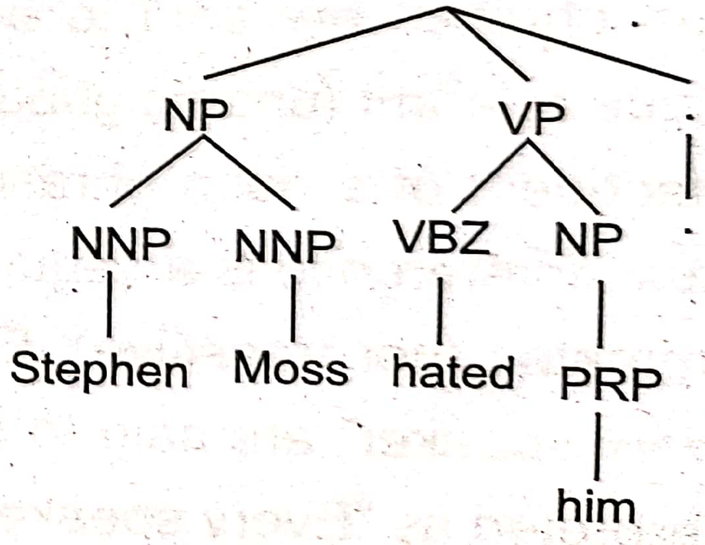


Fig. 6.3.3 : Answer Extraction

6.4 Categorization

- Text classification also known as *text tagging* or *text categorization* is the process of categorizing text into organized groups. By using Natural Language Processing (NLP), text classifiers can automatically analyze text and then assign a set of pre-defined tags or categories based on its content.

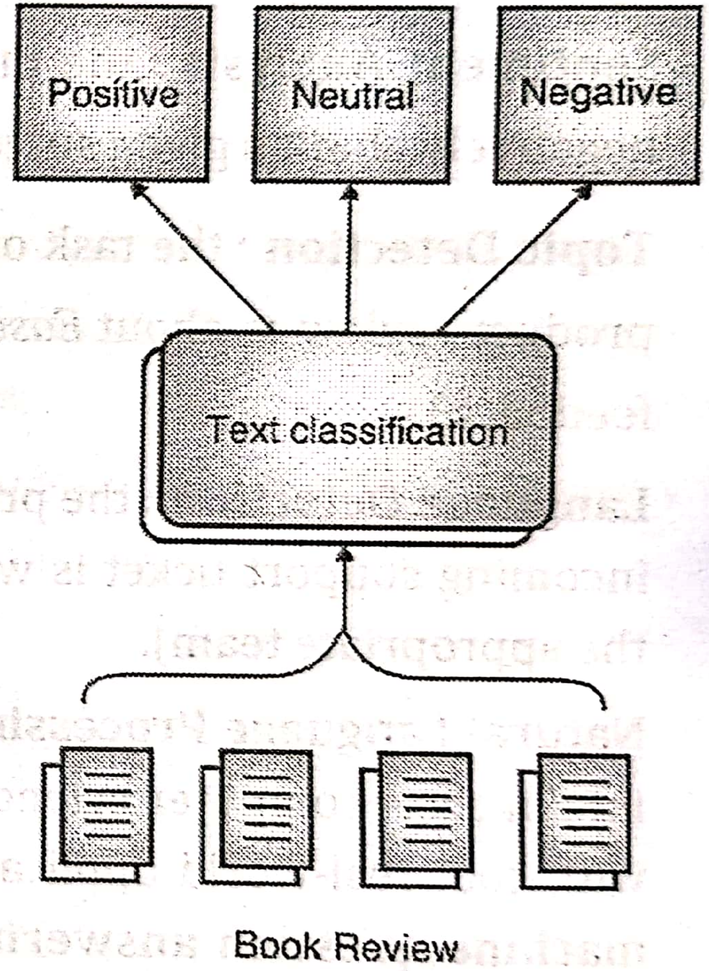


Fig.6.4.1

Approaches

Text Classification can be achieved through three main approaches :

1. **Rule-based approaches :** These approaches make use of handcrafted linguistic rules to classify text. One way to group text is to create a list of words related to a certain column and then judge the text based on the occurrences of these words. For example, words like "fur", "feathers", "claws", and "scales" could help a zoologist identify texts talking about animals online. These approaches require a lot of domain knowledge to be extensive, take a lot of time to compile, and are difficult to scale.
2. **Machine learning approaches :** We can use machine learning to train models on large sets of text data to predict categories of new text. To train models, we need to transform text data into numerical data – this is known as **feature extraction**. Important feature extraction techniques include *bag of words* and *n-grams*. There are several useful machine learning algorithms we can use for text classification. The most popular ones are:
   - Naive Bayes classifiers
   - Support vector machines
   - Deep learning algorithms
3. **Hybrid approaches :**These approaches are a combination of the two algorithms above. They make use of both rule-based and machine learning techniques to model a classifier that can be fine-tuned in certain scenarios.

Some of the most common examples and use cases for automatic text classification include the following :

- **Sentiment Analysis :** the process of understanding if a given text is talking positively or negatively about a given subject (e.g. for brand monitoring purposes).
- **Topic Detection :** the task of identifying the theme or topic of a piece of text (e.g. know if a product review is about *Ease of Use*, *Customer Support*, or *Pricing* when analyzing customer feedback).
- **Language Detection :** the procedure of detecting the language of a given text (e.g. know if an incoming support ticket is written in English or Spanish for automatically routing tickets to the appropriate team).
- **Natural Language Processing (NLP)** is a wide area of research where the worlds of artificial intelligence, computer science, and linguistics collide. It includes a bevy of interesting topics with cool real-world applications, like **named entity recognition, machine translation** or **machine question answering**. Each of these topics has its own way of dealing with textual data.

- But before diving into the deep end and looking at these more complex applications, we need to wade in the shallow end and understand how simpler tasks such as **text classification** are performed.
- Text classification offers a good framework for getting familiar with textual data processing without lacking interest, either. In fact, there are many interesting applications for text classification such as **spam detection** and **sentiment analysis.**
- **Pre-Processing:** A simple approach is to assume that the smallest unit of information in a text is the word (as opposed to the character). Therefore, we will be representing our texts as word sequences. For instance:

  Text: This is a cat. **-->Word Sequence:** [this, is, a, cat]
- In this example, we removed the punctuation and made each word lowercase because we assume that punctuation and letter case don't influence the meaning of words. In fact, we want to avoid making distinctions between similar words such as *This* and *this* or *cat.* and *cat.*
- Moreover, real life text is often "dirty." Because this text is usually automatically scraped from the web, some HTML code can get mixed up with the actual text. So we also need to tidy up these texts a little bit to avoid having HTML code words in our word sequences. For example :

  This is not a sentence.

  <\div> --> [this, is, not, a, sentence]
- Making these changes to our text before turning them into word sequences is called **pre-processing**. Despite being very simple, the pre-processing techniques we have seen so far work very well in practice. Depending on the kind of texts you may encounter, it may be relevant to include more complex pre-processing steps. But keep in mind that the more steps you add, the longer the pre-processing will take.
- A regular expression (or **regex**) is a sequence of characters that represent a search pattern. Each character has a meaning; for example, means any character that isn't the newline character: '\n'. These characters are often combined with quantifiers, such as *, which means zero or more. Combining these two characters, we can make the regex that looks for an expression in the form '<' + 'zero or more' of 'anything but \n' + '>'. This regex is <.*?>. Here the character? indicates a non-greedy search:
- **Input string:** <a>bcd>Difference between greedy and non-greedy search :

  greedy: <.*> --><a>bcd>

  non-greedy: <.*?> --><a>

- Regular expressions are very useful for processing strings. For example, the <.*?> regex we introduced before can be used to detect and remove HTML tags. But we will also be using other regex such as \' to remove the character ' so that words like that's become thats instead of two separate words that and s.
- **Vectorization :** Now that we have a way to extract information from text in the form of word sequences, we need a way to transform these word sequences into numerical features, this is Vectorization.
- The simplest text Vectorization technique is Bag Of Words (BOW). It starts with a list of words called the vocabulary (this is often all the words that occur in the training data). Then, given an input text, it outputs a numerical vector which is simply the vector of word counts for each word of the vocabulary.
- **For example :**

  Training texts: ["This is a good cat", "This is a bad day"] → vocabulary: [this, cat, day, is, good, a, bad].New text: "This day is a good day" → [1, 0, 2, 1, 1, 1, 0]
- As we can see, the values for "cat" and "bad" are **0** because these words don't appear in the original text.
- Using BOW is making the assumption that the more a word appears in a text, the more it is representative of its meaning. Therefore, we assume that given a set of positive and negative text, a good classifier will be able to detect patterns in word distributions and learn to predict the sentiment of a text based on which words occur and how many times they do.

6. Text Summarization

Summarization means to reduce the size of the document without changing its meaning. It is one of the most researched areas among the Natural Language Processing (NLP) community. A good summary should cover the most vital information of the original document or a cluster of documents, while being coherent, non-redundant and grammatically readable. Automatic text summarization is the data science problem of creating a short, accurate, and fluent summary from a longer document. Summarization methods are greatly needed to consume the ever-growing amount of text data available online. In essence, summarization is meant to help us consume relevant information faster. Furthermore, applying text summarization reduces reading time, accelerates the process of researching for information, and increases the amount of information that can fit in an area.

Summarization techniques are categorized into extractive and abstractive techniques on the basis of whether the exact sentences are considered as they appear in the original text

Types of Text Summarization:

A. Extraction-based summarization:

The extractive text summarization technique involves pulling keyphrases from the source document and combining them to make a summary. The extraction is made according to the defined metric without making any changes to the texts.

Here is an example:

Source text: Joseph and Mary rode on a donkey to attend the annual event in Jerusalem. In the city, Mary gave birth to a child named Jesus.

Extractive summary: Joseph and Mary attend event Jerusalem. Mary birth Jesus.

As you can see above, the words in bold have been extracted and joined to create a summary — although sometimes the summary can be grammatically strange.

B. Abstraction-based summarization :

The abstraction technique entails paraphrasing and shortening parts of the source document. When abstraction is applied for text summarization in deep learning problems, it can overcome the grammar inconsistencies of the extractive method.

The abstractive text summarization algorithms create new phrases and sentences that relay the most useful information from the original text — just like humans do.

Therefore, abstraction performs better than extraction. However, the text summarization algorithms required to do abstraction are more difficult to develop; that's why the use of extraction is still popular.

Here is an example:

Abstractive summary: Joseph and Mary came to Jerusalem where Jesus was born.

Text summarization algorithm :

Usually, text summarization in NLP is treated as a supervised machine learning problem (where future outcomes are predicted based on provided data).

Typically, here is how using the extraction-based approach to summarize texts can work:

1. Introduce a method to extract the merited keyphrases from the source document. For example, you can use part-of-speech tagging, word sequences, or other linguistic patterns to identify the keyphrases.
2. Gather text documents with positively-labeled keyphrases. The keyphrases should be compatible to the stipulated extraction technique. To increase accuracy, you can also create negatively-labeled keyphrases.
3. Train a binary machine learning classifier to make the text summarization. Some of the features you can use include:
   - Length of the keyphrase
   - Frequency of the keyphrase
   - The most recurring word in the keyphrase
   - Number of characters in the keyphrase
4. Finally, in the test phrase, create all the keyphrase words and sentences and carry out classification for them.

Example : Abstractive text summarisation for MARATHI documents

The idea is to summarize an input Marathi document by creating semantic graph called rich semantic graph(RSG) for the original document, reducing the generated semantic graph, and further generating the final abstract summary.

2. Information Retrieval

Information Retrieval remains one of the most challenging problems in NLP. Hundreds of millions of people engage in information retrieval everyday while using a web search engine or searching emails. Traditional database searching is becoming obsolete since most of the times the user is unclear what he or she is searching for. Information retrieval system is one that searches a collection of natural language documents with the goal of retrieving exactly the set of documents that pertain to a user's question"

A lot of readily available information is available through the World Wide Web which gets updated every time and is reached out to people all over the world. Thus, searching applications has changed tremendously from systems designed for specific applications belonging to well defined group to systems which are applicable for common people. However the huge and undefined structure of information present over networks have made it difficult for users to search and find relevant information. Many information retrieval techniques have been developed to deal with this problem.

Information retrieval (IR) may be defined as a software program that deals with the organization, storage, retrieval and evaluation of information from document repositories particularly textual information. The system assists users in finding the information they require but it does not explicitly return the answers of the questions. It informs the existence and location of documents that might consist of the required information. The documents that satisfy the user's requirement are called relevant documents. A perfect IR system will retrieve only relevant documents

Traditionally, the information retrieval system techniques are based on keyword. They use lists of keywords to describe the content of information but they do not say reveal semantic relationships between keywords nor consider the meaning of words and phrases. For example, the search engines accept keywords and in return they show a list of links to documents containing those keywords.

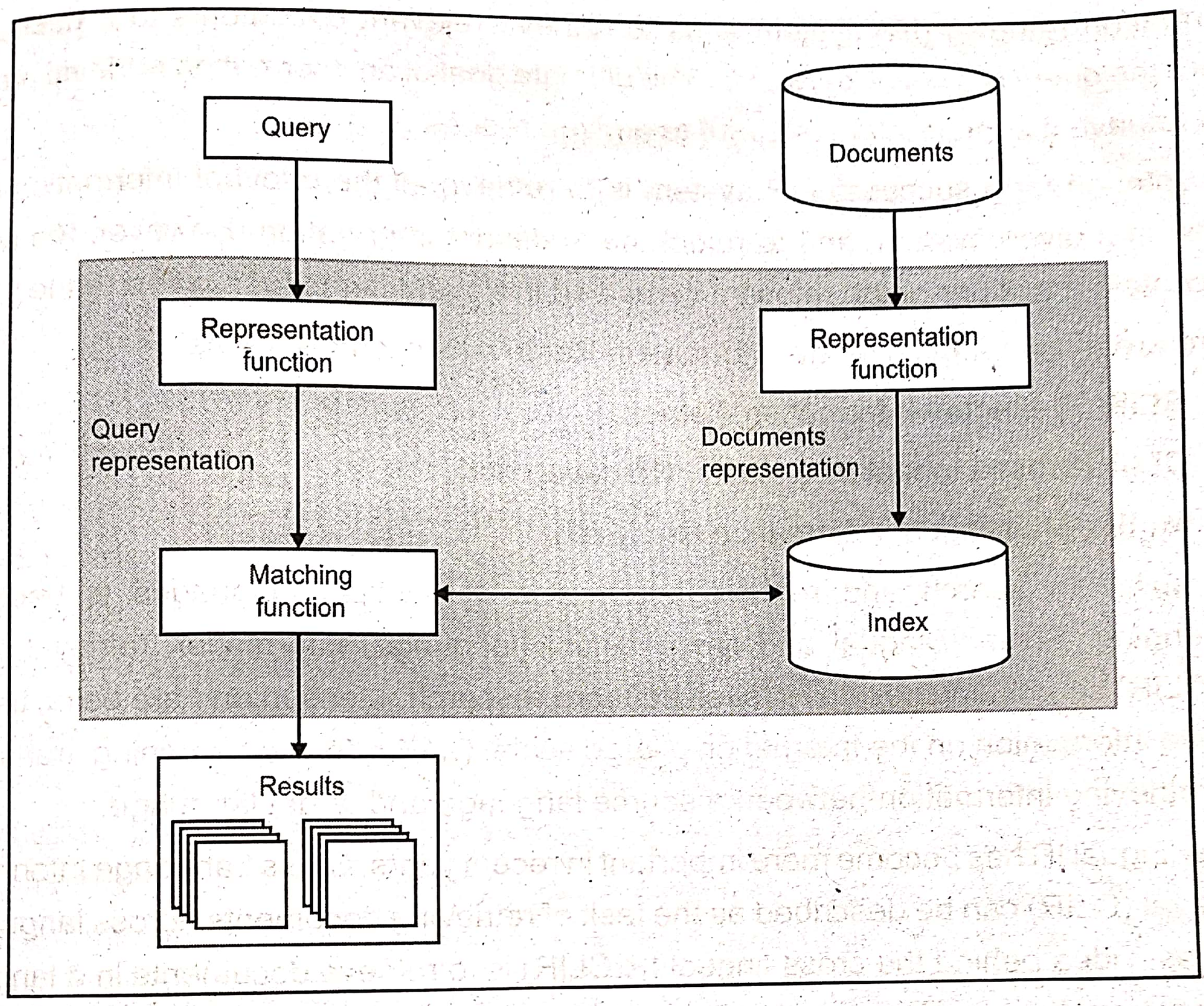


Figure 3: Basic IR system

Basic IR system involves following stages:

1. Indexing the collection of documents.
2. Transforming the query in the same way as the document content is represented.
3. Comparing the description of each document with that of the query.
4. Listing the results in order of relevancy.

In general all information Retrieval Systems consist of mainly two processes as:

1. **Indexing:** Indexing is the process of selecting terms to represent a text which involves tokenization of string, removing frequent words and stemming.
2. **Matching**: Matching is the process of computing a measure of similarity between two text representations. Relevance of a document is computed based on parameters like term frequency and inverse document frequency .

Information retrieval (IR) system aims to retrieve relevant documents to a user query where the query is a set of keywords. The ultimate goal of an information retrieval process is to retrieve the information relevant to a given request.

The criterion for a successful IR system is to retrieve all the relevant information items stored in a given system, and to reject the irrelevant information. However, the results will contain a mixture of the relevant items and irrelevant items for a given request.

There are many variants of the Information Retrieval systems such as :-

1. **BLIR** (Bi-Lingual Information Retrieval)
2. **CLIR** (Cross-Lingual Information Retrieval) and
3. **MLIR** (Multilingual Information Retrieval).

The ability to search and retrieve information in multiple languages is becoming challenging. So multilingual and cross-lingual (language) information retrieval (MLIR and CLIR) search engines have received more research attention and are being used to retrieve information on the Internet on a large scale. CLIR refers to searching, translating and retrieving information between a source language and target language.

Cross-lingual IR has become more important in recent years. Cross Language Information Retrieval (CLIR) can be described as the task of retrieving documents across languages. The basic idea behind the cross-lingual IR(CLIR) is to retrieve documents in a language different from the language used by the user to develop the query. This may be desirable even when the user is not a speaker of the language used in the retrieved documents. CLIR uses different translation approaches to translate queries to documents and indexes in other languages. Some CLIR systems use language resources such as bilingual dictionaries to translate the user's original query, while other systems use machine translation to translate the foreign-language documents beforehand, enabling them to be retrieved by the original query.

This task represents one extreme case of the vocabulary mismatch problem, i.e. The vocabulary of a user query and the vocabulary of relevant documents can differ substantially. The 'bag-of-words (BOW) model seriously suffers from the vocabulary mismatch problem because of different dimensions thus neglecting relations between different words in the same language as well as across languages. Therefore, the challenging task of retrieving documents to queries in other languages requires models other than traditional bag-of-words model.

Maximum Entropy

Maximum entropy modelling is a framework for integrating information from many heterogeneous information sources for classification. The term maximum entropy refers to an optimization framework in which the goal is to find the probability model that maximizes entropy over the set of models that are consistent with the observed evidence.

Maximum Entropy model is used to predict observations from training data. This does not uniquely identify the model but chooses the model which has the most uniform distribution i.e. the model with the maximum entropy. Entropy is a measure of uncertainty of a distribution, the higher the entropy the more uncertain a distribution is. Entropy measures uniformity of a distribution but applies to distributions in general.

The Principle of Maximum Entropy argues that the best probability model for the data is the one which maximizes entropy, over the set of probability distributions that are consistent with the evidence.

Maximum Entropy: A simple example

The following example illustrates the use of maximum entropy on a very simple problem.

- Model an expert translator's decisions concerning the proper French rendering of the English word ***on***
- A model(p) of the expert's decisions assigns to each French word or phrase(f) an estimate, p(f), of the probability that the expert would choose f as a translation of ***on.*** Our goal is to extract a set of facts about the decision-making process from the sample and construct a model of this process

A clue from the sample is the list of allowed translations

on **---> {sur, dans, par, au bord de}**

With this information in hand, we can impose our first constraint on p:

p(sur) + p(dans) + p(par) + p(au bord de) =1

The most uniform model will divide the probability values equally. Suppose we notice that the expert chose either dans or sur 30% of the time, then a second constraint can be added.

p(dans) + p(sur) = 3/10

- Intuitive Principle: Model all that is known and assume nothing about that which is unknown

  A random process which produces an output value y, a member of a finite set Y.

  $y \in$ {sur, dans, par, au bord de}

The process may be influenced by some contextual information x, a member of a finite set X. x could include the words in the English sentence surrounding ***on.***

maximum entropy framework can be used to solve various problems in the domain of natural language processing like sentence boundary detection, part-of-speech tagging, prepositional phrase attachment, natural language parsing, and text categorization .

3.4.3 Conditional Random Fields

- The **conditional random field** (CRF) is a conditional probabilistic model for sequence labeling; just as structured perceptron is built on the perceptron classifier, conditional random fields are built on the logistic regression classifier.
- Conditional random fields (CRFs) are a probabilistic framework for labeling and segmenting sequential data, based on the conditional approach described in hidden markov model.
- A CRF is a form of undirected graphical model that defines a single log-linear distribution over label sequences given a particular observation sequence.
- The primary advantage of CRFs over hidden Markov models is their conditional nature, resulting in the relaxation of the independence assumptions required by HMMs in order to ensure tractable inference.
- Additionally, CRFs avoid the label bias problem; a weakness exhibited by maximum entropy Markov models (MEMMs) and other conditional Markov models based on directed graphical models.
- CRFs outperform both MEMMs and HMMs on a number of real-world sequence labeling tasks.
- The graphical structure of a conditional random field may be used to factorize the joint distribution over elements Yv of Y into a normalized product of strictly positive, real-valued potential functions, derived from the notion of conditional independence.
- The probability of a particular label sequence ***y*** given observation sequence ***x*** to be a normalized product of potential functions, each of the form

$$\exp\left(\sum_{j} \lambda_j t_j (y_{i-1}, x, i) + \sum_{k} \mu_k S_k (y_i, x, i)\right)$$

where ***tj (yi – 1, yi, x, i)*** is a transition feature function of the entire observation sequence and the labels at positions i and i-1 in the label sequence;

Conditional Random Field (CRF)

In POS tagging, the goal is to label a sentence (a sequence of words or tokens) with tags like ADJECTIVE, NOUN, PREPOSITION, VERB, ADVERB, ARTICLE.

For example, given the sentence "Bob drank coffee at Starbucks", the labeling might be "Bob (NOUN) drank (VERB) coffee (NOUN) at (PREPOSITION) Starbucks (NOUN)".

So let's build a conditional random field to label sentences with their parts of speech. Just like any classifier, we'll first need to decide on a set of feature functions fi.

Feature Functions in a CRF

In a CRF, each feature function is a function that takes in as input:

a sentence s

the position i of a word in the sentence

the label l_i of the current word

the label l_{i-1} of the previous word

and outputs a real-valued number (though the numbers are often just either 0 or 1).

For example, one possible feature function could measure how much we suspect that the current word should be labeled as an adjective given that the previous word is "very".

6.3 Question Answers System

- Question Answer System is a branch of learning of information retrieval and natural language processing, which focuses on building systems that automatically answer questions posed by users in a natural language.
- An understanding of natural language consists of the ability of a system to decode sentences into a representation so the system generates valid answers to questions asked by an user.
- Effective answers mean answers relevant to the questions posed by the user. As the representation of natural language, sentences must effectively map semantics of the statement.
- To form an answer it is necessary to execute the syntax and semantic analysis of a question.
- The process of the system is as follows :
  1. Query Processing
  2. Document Retrieval
  3. Passage Retrieval
  4. Answer Extraction

Step 1: Query Processing

Classify question into seven categories

- **Who** is/was/are/were...?
- **When** is/did/will/are/were ...?
- **Where** is/are/were ...?

a. Category-specific transformation rules

eg "For Where questions, move 'is' to all possible locations"

"Where is the Louvre Museum located"

→ "is the Louvre Museum located"

→ "the is Louvre Museum located"

→ "the Louvre is Museum located"

→ "the Louvre Museum is located"

→ "the Louvre Museum located is"

b. Expected answer "Datatype" (eg, Date, Person, Location, ...)

- **When** was the French Revolution? → DATE

Hand-crafted classification/rewrite/datatyperules
(Could they be automatically learned?)

Send all rewrites to a Web search engine

- Retrieve top N answers (100?)
- For speed, rely just on search engine's "snippets", not the full text of the actual document
- Some query rewrites are more reliable than others

Step 2: Document Retrieval

- In the document retrieval, we will retrieve relevant documents by using the generated query.
- Users submit queries corresponding to their information need
- System returns (voluminous) list of full-length documents
- Then the users find their original information need, within the returned documents

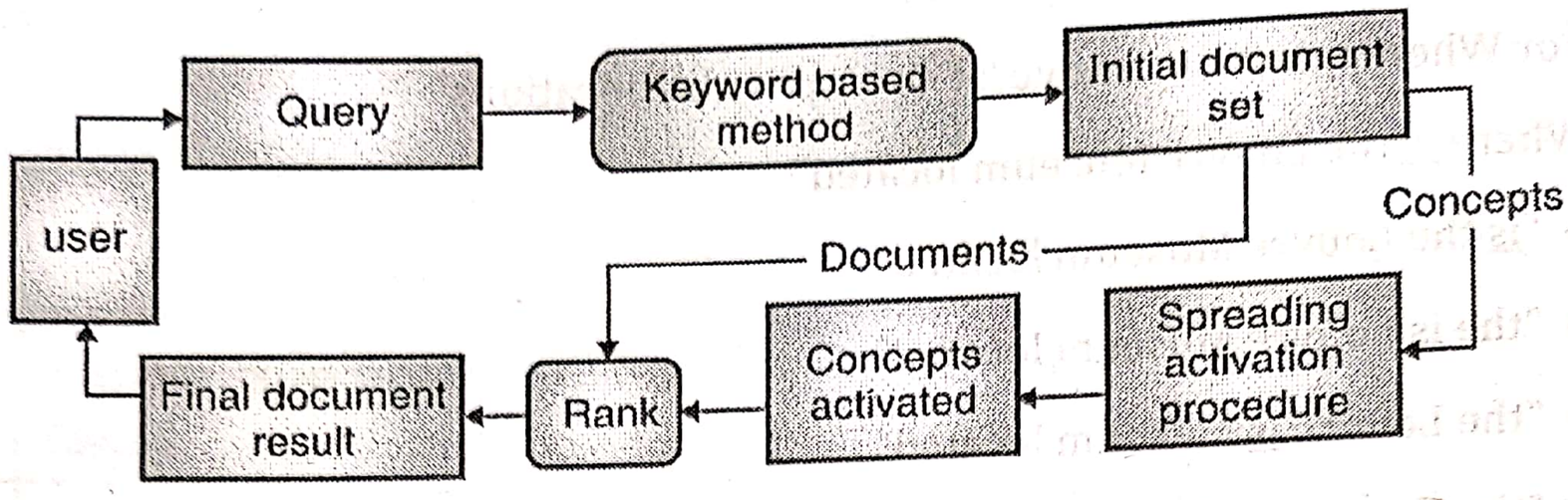


Fig. 6.3.2 : The flow of document retrieval

Step 3: Passage Retrieval

- Passage Retrieval (PR) systems have been proposed for English language. PR systems works on parts of text so that they can limit the relevance of a document to a query, besides detecting document portions that are likely to contain the required answer(rather than the full document).
- The documents are separated into smaller units (passages) such as sentences and paragraphs, and passages that are likely to contain an answer are nominated. If the document is short, it is not needed, but if it is long, passage selection is effective because being uninformed which part of the document the query matched. Generally, the following answer extraction process takes a duration of time to complete, so selecting passages also has the advantage of fast moving up the entire system.

Step 4: Answer Extraction

- Answer Extraction module process the top ranked passages for extracting the final answer to the user's question. Typically, named-entity recognition technique is used to find candidate answers that match the question's named entities.
- For questions that are looking for dates a pattern matching technique is used. Questions that requires descriptive answers like how & why cosine similarity is used to find the answers
- Answer extraction extracts answers from passages. Here, the question and passage are input to the answer extraction model, and the model outputs the answer offset with the score. It then ranks the answers based on the score and presents the answers with the highest scores to the user as the final answer.

6.7 Named Entity Recognition (NER)

- Named entity recognition (NER) also called entity identification or entity extraction is a natural language processing (NLP) technique that automatically identifies named entities in a text and classifies them into predefined categories. Entities can be names of people, organizations, locations, times, quantities, monetary values, percentages, and more.

Ousted WeWork founder Adam Neumann lists his Manhattan penthouse for $3.75 million

[organization] [person] [location] [Monetary value]

- With named entity recognition, you can extract key information to understand what a text is about, or merely use it to collect important information to store in a database. In this guide, we'll explore how named entity recognition works, its applications in business, and how to perform entity extraction using no-code tools.

For example: Mahatma Gandhi is very famous in India as "Bapu" or The full name of him is Mohandas Karamchand Gandhi. He was a great freedom fighter who led India as a leader of the nationalism against British rule. He was born on the 2nd of October in 1869 in Porbandar, Gujarat. He was a leader of Congress, as it was the only national party in India.

Tag Names	**Tagged Entities**
Person	*Mahatma Gandhi, Bapu, Mohandas Karamchand Gandhi*
Location	*India, Porbandar, Gujarat*
Organisation	*Congress*
Date	*2nd, October, 1869*

6.7.1 Types of Named Entity Recognition

- The Named entity hierarchy is divided into three major classes Entity, Name, Time and Numerical expressions.

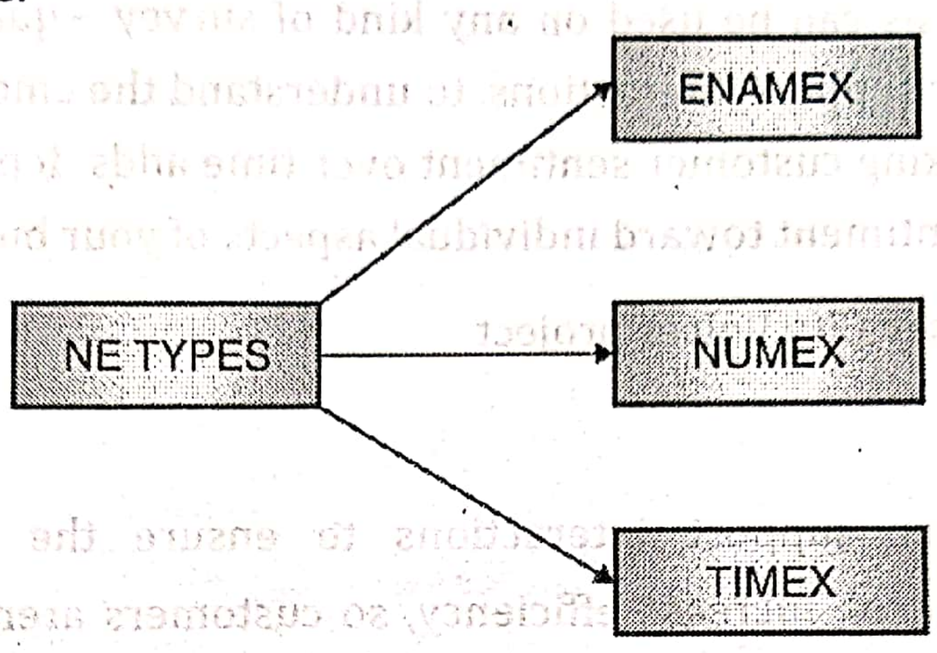


Fig. 6.7.1

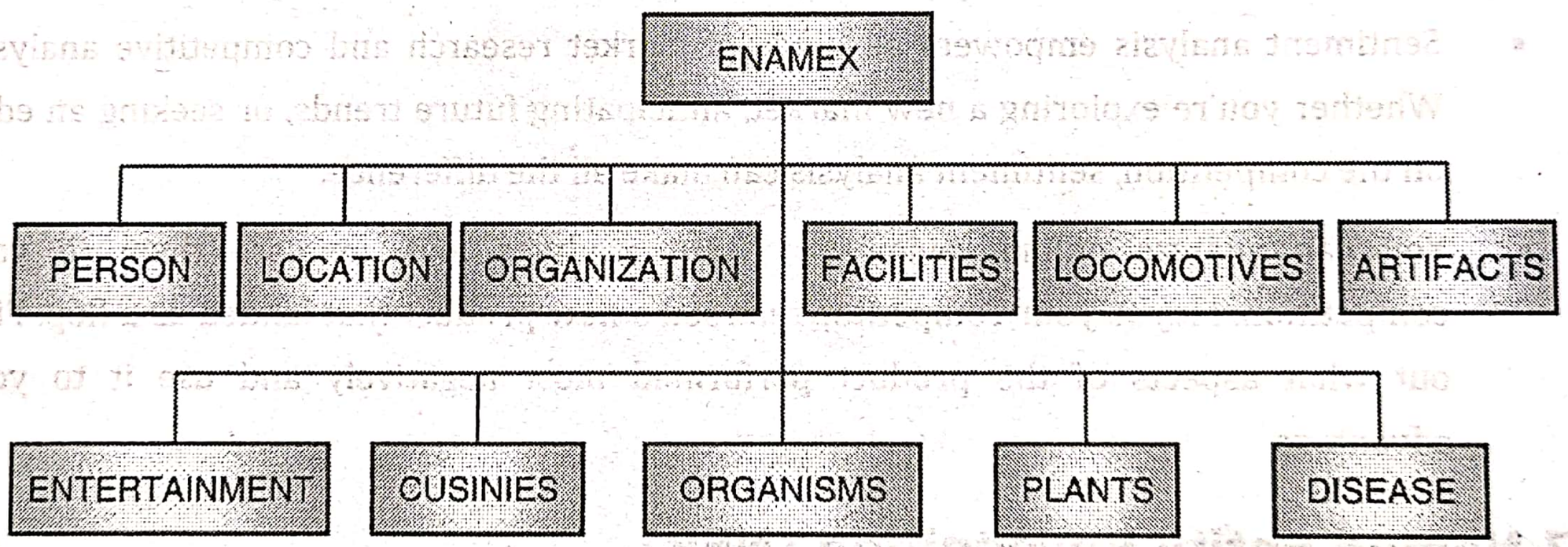


Fig. 6.7.2

1. Entity Name Types

- Persons are entities limited to humans. A person may be a single individual or a group. Individual refer to names of each individual person. Group refers to set of individuals
- Location entities are limited to geographical entities such as geographical areas like names of countries, cities, continents and landmasses, bodies of water, and geological formations.
- Organization entities are limited to corporations, agencies, and other groups of people defined by an established organizational structure

2. Numerical Expressions

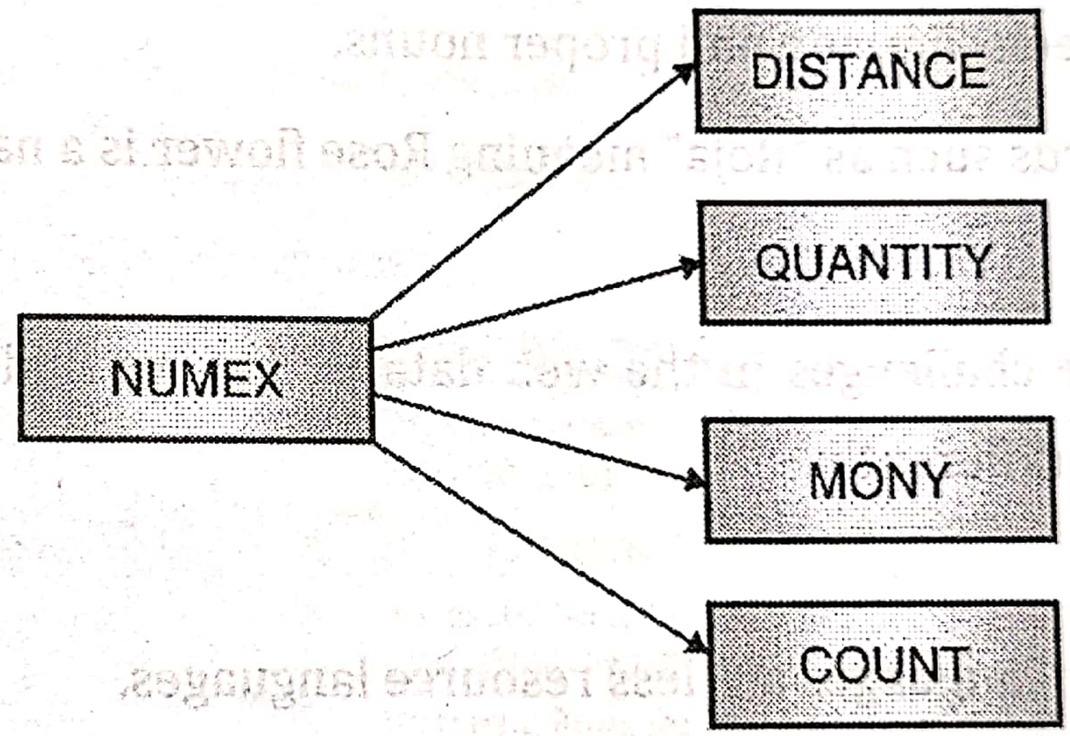


Fig. 6.7.3

3. Time Expressions

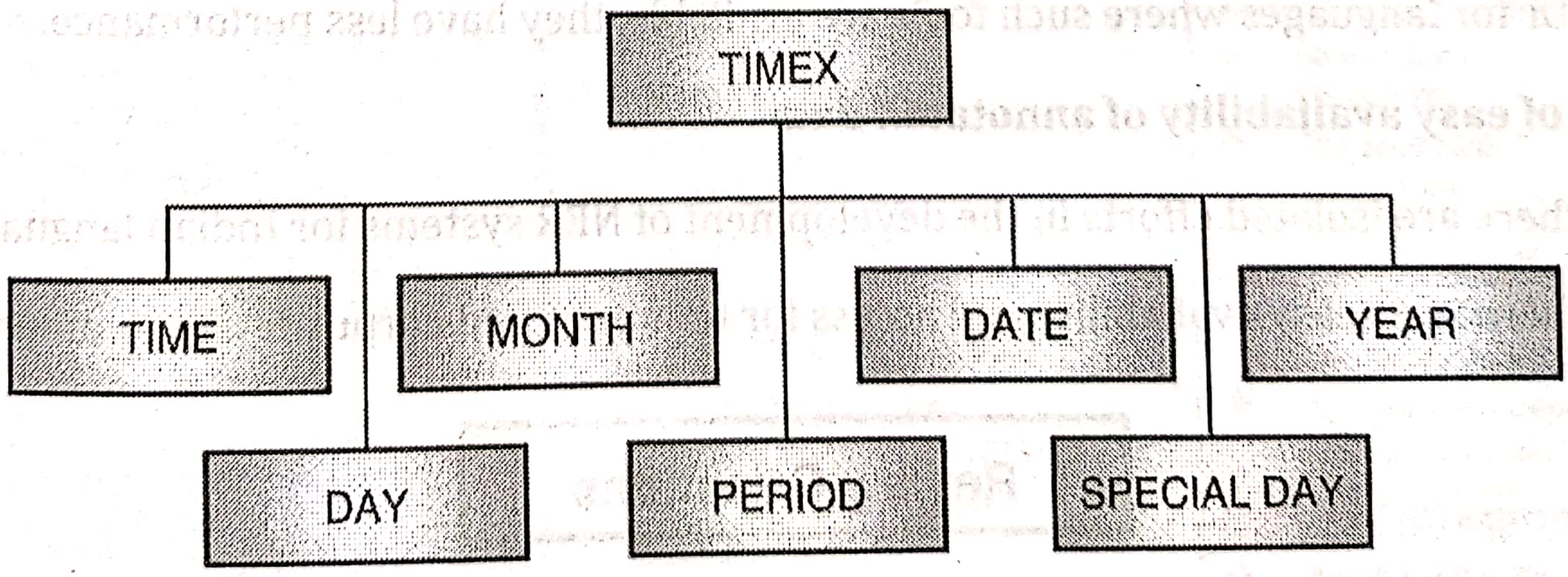


Fig. 6.7.4