Natural Languages Processing:

Chapter 1 and 2

- Introduction To NLP
- Challenges/Open Problems of NLP
- Characteristics of NLP
- Application of NLP
- Word Segmentation
- Parsing – Parsing Tree, Top down parsing and Bottom up parsing
- Chunking,
- NER
- Sentiment Analysis
- Web 2.0 application

Chapter 3

- HMM
- CRF
- Naïve Bayes

Chapter 4
- Pos Tagging – Difficulty
- Morphology Fundamentals - Types
    - Automatic Morphology Learning,
    - Finite State Machine Based Morphology
    - Shallow Parsing

Chapter 5

- Dependency Parsing
- Malt Parser

Chapter 6

- Lexical Knowledge Networks
- WordNET Theory
- Semantic Roles
- Metaphors;
- Word Sense – Application

Chapter 1 and 2

➢ Introduction To NLP:

1. Natural language processing (NLP) can be defined as the automatic (or semi-automatic) processing of human language.
2. Natural Language processing (NLP) is a field of computer science and linguistics concerned with the interactions between computers and human (natural) languages.
3. In theory, natural-language processing is a very attractive method of human-computer interaction.
4. Natural language processing is the task of analyzing and generating by computers, languages that humans speak, read and write.
5. NLP is concerned with questions involving three dimensions: language, algorithm and problem.
6. Figure 1 expresses this point. On the language axis are different natural languages and linguistics.
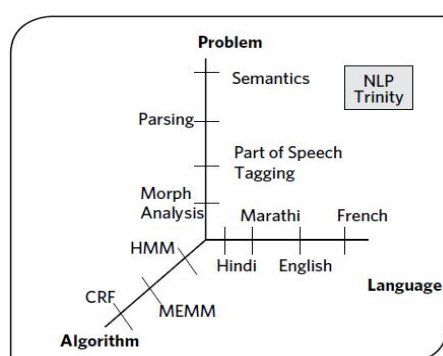

Fig. 1: Three dimensions of NLP

7. The problem axis mentions different NLP tasks like morphology, part of speech tagging etc.
8. The algorithm axis depicts mechanisms like HMM, MEMM, CRF etc. for solving problems.
9. The goal of natural language analysis is to produce knowledge representation structures like predicate calculus expressions, semantic graphs or frames. This processing makes use of foundational tasks like morphology analysis, Part of Speech Tagging, Named Entity Recognition, both shallow and deep Parsing, Semantics Extraction, Pragmatics and Discourse Processing.

➢ Challenges/Open Problems of NLP:

- Natural Language Processing (NLP) is the process of computer analysis of input provided in a human language (natural language), and conversion of this input into a useful form of representation.
- The field of NLP is primarily concerned with getting computers to perform useful and interesting tasks with human languages. The field of NLP is secondarily concerned with helping us come to a better understanding of human language.
    - The input/output of a NLP system can be:
        - **written text**
        - **speech**
    - We will mostly concerned with written text (not speech).
    - To process written text, we need:
        - **lexical, syntactic, semantic knowledge about the language**
        - **discourse information, real world knowledge**
    - To process spoken language, we need everything required to process written text, plus the challenges of speech recognition and speech synthesis.
- There are two components of NLP.
    - **Natural Language Understanding**
        - Mapping the given input in the natural language into a useful representation.
        - Different level of analysis required:
            morphological analysis,
            syntactic analysis,
            semantic analysis,
            discourse analysis, …

- **Natural Language Generation**
  - Producing output in the natural language from some internal representation.
  - Different level of synthesis required:
    deep planning (what to say),
    syntactic generation
- NL Understanding is much harder than NL Generation. But, still both of them are hard.
- The main challenges of NLP are –
  1. Ambiguity - Syntactic ambiguity and polysemous words and

  2. Models to represent Linguistic Knowledge
  3. Algorithms to Manipulate Linguistic Knowledge
- The difficulty in NL understanding arises from the following facts:
  1. Natural language is extremely rich in form and structure, and very ambiguous.
     i. How to represent meaning,
     ii. Which structures map to which meaning structures.
  2. One input can mean many different things. Ambiguity can be at different levels.
     i. Lexical (word level) ambiguity -- different meanings of words
     ii. Syntactic ambiguity -- different ways to parse the sentence
     iii. Interpreting partial information -- how to interpret pronouns
     iv. Contextual information -- context of the sentence may affect the meaning of that sentence.
  3. Much input can mean the same thing.
  4. Interaction among components of the input is not clear.

➤ **Characteristics of NLP**

➤ **Application of NLP**

The applications can be divided into two major classes: Text-based applications and Dialogue-based applications.

Text-based applications:

Text-based applications involve the processing of written text, such as books, newspapers, reports, manuals, e-mail messages, and so on. These are all reading-based tasks. Text-based natural language research is ongoing in applications such as

- finding appropriate documents on certain topics from a database of texts (for example, finding relevant books in a library)
- extracting information from messages or articles on certain topics (for example, building a database of all stock transactions described in the news on a given day)
- translating documents from one language to another (for example, producing automobile repair manuals in many different languages)
- summarizing texts for certain purposes (for example, producing a 3-page summary of a 1000-page government report)
- One very attractive domain for text-based research is story understanding. In this task the system processes a story and then must answer questions about it. This is similar to the type of reading comprehension tests used in schools and provides a very rich method for evaluating the depth of understanding the system is able to achieve.

Dialogue-based applications:

It involves human-machine communication. Most naturally this involves spoken language, but it also includes interaction using keyboards.

Typical potential applications include

- question-answering systems, where natural language is used to query a database (for example, a query system to a personnel database)
- automated customer service over the telephone (for example, to perform banking transactions or order items from a catalogue)
- tutoring systems, where the machine interacts with a student (for example, an automated mathematics tutoring system)
- spoken language control of a machine (for example, voice control of a VCR or computer)
- general cooperative problem-solving systems (for example, a system that helps a person plan and schedule freight shipments)

The following list is not complete, but useful systems have been built for:

- spelling and grammar checking
- optical character recognition (OCR)
- screen readers for blind and partially sighted users
- augmentative and alternative communication (i.e., systems to aid people who have difficulty communicating because of disability)
- machine aided translation (i.e., systems which help a human translator, e.g., by storing translations of phrases and providing online dictionaries integrated with word processors, etc)
- lexicographers' tools
- information retrieval
- document classification (filtering, routing)
- document clustering
- information extraction
- question answering
- summarization
- text segmentation
- exam marking
- report generation (possibly multilingual)
- machine translation
- natural language interfaces to databases
- email understanding
- dialogue systems

➢ **Some NLP Task**

There are following NLP Task:
- Word segmentation
- Topic segmentation and recognition
- Part-of-speech tagging
- Word sense disambiguation
- Named entity recognition (NER)
- Parsing

➢ **Word Segmentation**
- Word segmentation is the problem of dividing a string of written language into its component words.
- In English and many other languages using some form of the Latin alphabet, the space is a good approximation of a word divider (word delimiter).

➢ **Parsing – Parsing Tree, Top down parsing and Bottom up parsing**
   What is Parsing?
- Parsing is the process of taking a string and a grammar and returning a (or multiple) parse tree(s) for that string
- It is completely analogous to running a finite-state transducer with a tape
- It's just more powerful - there are languages we can capture with CFGs that we can't capture with finite-state machines.
- **Example 1 - John ate the cat**

   A **top-down strategy** starts with S and searches through different ways to rewrite the symbols until it generates the input sentence (or it fails). Thus S is the start and it proceeds through a series of rewrites until the sentence under consideration is found.

       S
       NP VP
       NAME VP
       John VP
       John V NP
       John are NP
       John are ART N
       John ate the N
       John ate the cat

   In a **bottom-up strategy,** one starts with the words of the sentence and used the rewrite rules backward to reduce the sentence symbols until one is left with S.

       John ate the cat
       NAME ate the cat
       NAME V the cat
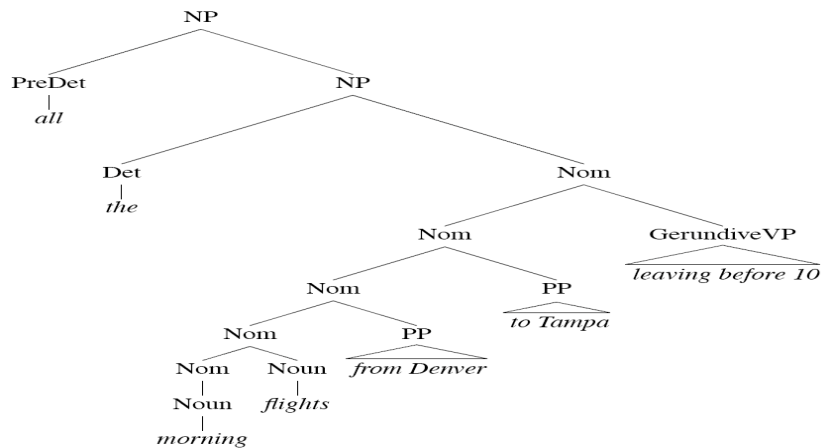   NAME V ART cat
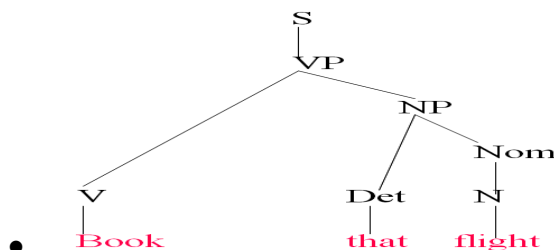
NAME V ART N
NP V ART N
NP V NP
NP VP
S

➢ **Example 2** Construct the Parse Tree for following sentence
"All the morning flights from Denver to Tampa leaving before 10."

➢ **Top Down Parsing** – **Construct the Parse Tree** – **Book that flights**
- Top-down parsing is a strategy of analyzing unknown data relationships by hypothesizing general parse tree structures and then considering whether the known fundamental structures are compatible with the hypothesis. It occurs in the analysis of both natural languages and computer languages.
- A top-down parser searches for a parse tree by trying to build from the root node S down to the leaves.
- The top-down strategy never wastes time exploring trees that cannot result in an S, since it begins by generating just those trees.
- Example -

➢ **Different between Top down parsing and Bottom up parsing**
  ➢ Top down never explores options that will not lead to a full parse, but can explore many options that never connect to the actual sentence.
  ➢ Bottom up never explores options that do not connect to the actual sentence but can explore options that can never lead to a full parse.

- ➢ Relative amounts of wasted search depend on how much the grammar branches in each direction
- ➢ **Chunking,**
- ➢ **NER (Named-entity recognition)**
  - It is also known as entity identification, entity chunking and entity extraction.
  - Named-entity recognition is the problem of segmenting and classifying proper names, such as names of people and organization, in text.
  - An entity is an individual person, place, or thing in the world, while a mention is a phrase of text that refers to an entity using a proper name.
  - The problem of named-entity recognition is in part one of segmentation because mentions in English are often multi-word.
  - It is a subtask of information extraction that seeks to locate and classify elements in text into pre-defined categories such as the names of persons, organizations, locations, expressions of times, quantities, monetary values, percentages, etc.
  - Most research on NER systems has been structured as taking an unannotated block of text, such as this one:
  - Example –

    Jim bought 300 shares of Acme Corp. in 2006.

    And producing an annotated block of text that highlights the names of entities:

    [Jim]Person bought 300 shares of **[Acme Corp.]**Organization in   Time.
  - In this example, a person name consisting of one token, a two-token company name and a temporal expression have been detected and classified.
- ➢ **Sentiment Analysis**
  - Sentiment analysis (also known as opinion mining) refers to the use of natural language processing, text analysis and computational linguistics to identify and extract subjective information in source materials.
  - Sentiment analysis is widely applied to reviews and social media for a variety of applications, ranging from marketing to customer service.
  - Sentiment analysis aims to determine the attitude of a speaker or a writer with respect to some topic or the overall contextual polarity of a document.
  - Types of Sentiment Analysis –
    1. Subjectivity/objectivity identification –
        - This task is commonly defined as classifying a given text (usually a sentence) into one of two classes: objective or subjective.
        - The subjectivity of words and phrases may depend on their context and an objective document may contain subjective sentences (e.g., a news article quoting people's opinions).
    2. Feature/aspect-based sentiment analysis –
        - It refers to determining the opinions or sentiments expressed on different features or aspects of entities, e.g., of a cell phone, a digital camera, or a bank.

- The advantage of feature-based sentiment analysis is the possibility to capture nuances about objects of interest. Different features can generate different sentiment responses, for example a hotel can have a convenient location, but mediocre food.

➢ **Web 2.0 application**

- Web 2.0 is the term given to describe a second generation of the World Wide Web that is focused on the ability for people to collaborate and share information online.
- Web 2.0 basically refers to the transition from static HTML Web pages to a more dynamic Web that is more organized and is based on serving Web applications to users.
- Web 2.0 is the current state of online technology as it compares to the early days of the Web, characterized by greater user interactivity and collaboration, more pervasive network connectivity and enhanced communication channels.
- One of the most significant differences between Web 2.0 and the traditional World Wide Web (WWW, retroactively referred to as Web 1.0) is greater collaboration among Internet users, content providers and enterprises. Originally, data was posted on Web sites, and users simply viewed or downloaded the content. Increasingly, users have more input into the nature and scope of Web content and in some cases exert real-time control over it.
- The foundational components of Web 2.0 are the advances enabled by Ajax and other applications such as RSS and Eclipse and the user empowerment that they support.
- **Application :**
  - ❖ **Trading** - Buying, selling or exchanging through user transactions mediated by internet communications
  - ❖ **Media sharing** - Uploading and downloading media files for purposes of audience or exchange
  - ❖ Conversational arenas - One-to-one or one-to-many conversations between internet users
  - ❖ **Online games and virtual worlds** - Rule-governed games or themed environments that invite live interaction with other internet users
  - ❖ **Social networking** - Websites that structure social interaction between members who form subgroups of 'friends' (Eg. Facebook, Orkut, etc)
  - ❖ **Blogging** - An internet-based journal or diary in which a user can post text and digital material while others can comment
  - ❖ **Social bookmarking** - Users submit their bookmarked web pages to a central site where they can be tagged and found by other users
  - ❖ **Recommender systems** - Websites aggregate and tag user preferences for items in some domain and thereby make novel recommendations
  - ❖ **Collaborative editing** - Web tools are used collaboratively to design, construct and distribute a digital product
  - ❖ **Wikis** - A web-based service allowing users unrestricted access to create, edit and link pages

❖ **Syndication** - Users can "subscribe" to RSS feed-enabled websites so that they are automatically notified of any changes or updates in content via an aggregator

Chapter 3

➢ **HMM - Hidden Markov Model**

- A hidden Markov model (HMM) is a statistical Markov model in which the system being modeled is assumed to be a Markov process with unobserved (hidden) states. A HMM can be presented as the simplest dynamic Bayesian network. The mathematics behind the HMM were developed by L. E. Baum and coworkers.

- In simpler Markov models the state is directly visible to the observer, and therefore the state transition probabilities are the only parameters. In a hidden Markov model, the state is not directly visible, but the output, dependent on the state, is visible. Each state has a probability distribution over the possible output tokens. Therefore the sequence of tokens generated by an HMM gives some information about the sequence of states. The adjective 'hidden' refers to the state sequence through which the model passes, not to the parameters of the model; the model is still referred to as a 'hidden' Markov model even if these parameters are known exactly.

- Hidden Markov models are especially known for their application in temporal pattern recognition such as speech, handwriting, gesture, part-of-speech tagging, musical score following, and bioinformatics.

- A hidden Markov model can be considered a generalization of a mixture model where the hidden variables (or latent variables), which control the mixture component to be selected for each observation, are related through a Markov process rather than independent of each other. Recently, hidden Markov models have been generalized to pair wise Markov models and triplet Markov models which allow consideration of more complex data structures and the modelling of nonstationary data.

- Example of HMM:

  Consider two friends, Alice and Bob, who live far apart from each other and who talk together daily over the telephone about what they did that day. Bob is only interested in three activities: walking in the park, shopping, and cleaning his apartment. The choice of what to do is determined exclusively by the weather on a given day. Alice has no definite information about the weather where Bob lives, but she knows general trends. Based on what Bob tells her he did each day, Alice tries to guess what the weather must have been like.
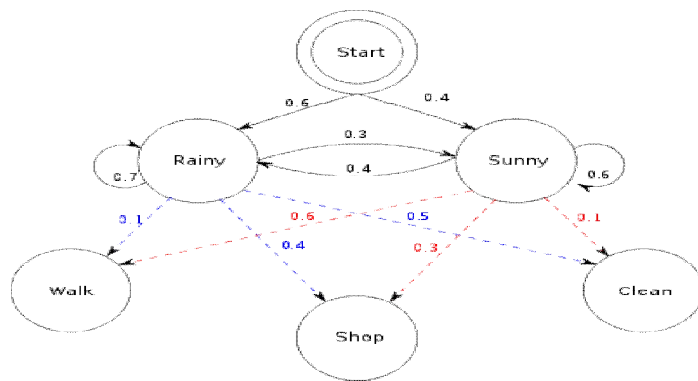
  Alice believes that the weather operates as a discrete Markov chain. There are two states, "Rainy" and "Sunny", but she cannot observe them directly, that is, they are hidden from her. On each day, there is a certain chance that Bob will perform one of the following activities, depending on the weather: "walk", "shop", or "clean". Since Bob tells Alice about his activities, those are the observations. The entire system is that of a hidden Markov model (HMM).

Alice knows the general weather trends in the area, and what Bob likes to do on average. In other words, the parameters of the HMM are known.

They can be represented as follows in Python:

```
states = ('Rainy', 'Sunny')
observations = ('walk', 'shop', 'clean')
start_probability = {'Rainy': 0.6, 'Sunny': 0.4}
transition_probability = {
                          'Rainy' : {'Rainy': 0.7, 'Sunny': 0.3},
                          'Sunny' : {'Rainy': 0.4, 'Sunny': 0.6},
                          }
emission_probability = {
                         'Rainy' : {'walk': 0.1, 'shop': 0.4, 'clean': 0.5},
                         'Sunny' : {'walk': 0.6, 'shop': 0.3, 'clean': 0.1},
                         }
```


In this piece of code,

The **start_probability** represents Alice's belief about which state the HMM is in when Bob first calls her.

The **transition_probability** represents the change of the weather in the underlying Markov chain.

The **emission_probability** represents how likely Bob is to perform a certain activity on each day.

- Application of HMM:

  HMMs can be applied in many fields where the goal is to recover a data sequence that is not immediately observable (but other data that depend on the sequence are). Applications include:

  - Single Molecule Kinetic analysis
  - Cryptanalysis
  - Speech recognition
  - Speech synthesis
  - Part-of-speech tagging

- Document Separation in scanning solutions
- Machine translation
- Partial discharge
- Gene prediction
- Alignment of bio-sequences
- Time Series Analysis
- Activity recognition
- Protein folding
- Metamorphic Virus Detection

➢ CRF (Conditional Random Field) –

- Conditional random fields (CRFs) are a class of statistical modelling method often applied in pattern recognition and machine learning, where they are used for structured prediction.
- CRFs are a type of discriminative undirected probabilistic graphical model.
- It is used to encode known relationships between observations and construct consistent interpretations.
- It is often used for labeling or parsing of sequential data, such as natural language text or biological sequences and in computer vision.
- CRFs are essentially a way of combining the advantages of discriminative classification and graphical modeling.
- Specifically, CRFs find applications in shallow parsing, named entity recognition, gene finding and peptide critical functional region finding, among other tasks, being an alternative to the related hidden Markov models (HMMs).
- In computer vision, CRFs are often used for object recognition and image segmentation.
- There are two types of CRFs Model.
  1. Graphical Model
  2. Linear Chain CRFs Model

➢ Naïve Bayes

- Naive Bayes has been studied extensively since the 1950s. It was introduced under a different name into the text retrieval community in the early 1960s.
- Naive Bayes classifiers are highly scalable, requiring a number of parameters linear in the number of variables (features/predictors) in a learning problem.
- Naive Bayes is a simple technique for constructing classifiers: models that assign class labels to problem instances, represented as vectors of feature values, where the class labels are drawn from some finite set.
- It is not a single algorithm for training such classifiers, but a family of algorithms based on a common principle: all naive Bayes classifiers assume that the value of a particular feature is independent of the value of any other feature, given the class variable.
- For example, a fruit may be considered to be an apple if it is red, round, and about 10 cm in diameter.

- A naive Bayes classifier considers each of these features to contribute independently to the probability that this fruit is an apple, regardless of any possible correlations between the color, roundness and diameter features.

- For some types of probability models, naive Bayes classifiers can be trained very efficiently in a supervised learning setting. In many practical applications, parameter estimation for naive Bayes models uses the method of maximum likelihood; in other words, one can work with the naive Bayes model without accepting Bayesian probability or using any Bayesian methods.

- Despite their naive design and apparently oversimplified assumptions, naive Bayes classifiers have worked quite well in many complex real-world situations.

- An advantage of naive Bayes is that it only requires a small amount of training data to estimate the parameters necessary for classification.

Chapter 4
➢ Pos Tagging – Difficulty

- The process of assigning one of the parts of speech to the given word is called Parts Of Speech tagging. It is commonly referred to as POS tagging. Parts of speech include nouns, verbs, adverbs, adjectives, pronouns, conjunction and their sub-categories.

- **Example:**

  Word : Paper, Tag: Noun
  Word : Go, Tag: Verb
  Word: Famous, Tag:Adjective

- POS tagging exemplas some general issues in NLP evaluation:
  Training data and test data The assumption in NLP is always that a system should work on novel data, therefore test data must be kept unseen. For machine learning approaches, such as stochastic POS tagging, the usual technique is to spilt a data set into 90% training and 10% test data. Care needs to be taken that the test data is representative. For an approach that relies on significant hand-coding, the test data should be literally unseen by the researchers. Development cycles involve looking at some initial data, developing the algorithm, testing on unseen data, revising the algorithm and testing on a new batch of data. The seen data is kept for regression testing.
  Baselines Evaluation should be reported with respect to a baseline, which is normally what could be achieved with a very basic approach, given the same training data. For instance, the baseline for POS tagging with training data is to choose the most common tag for a particular word on the basis of the training data (and to simply choose the most frequent tag of all for unseen words).
  Ceiling It is often useful to try and compute some sort of ceiling for the performance of an application. This is usually taken to be human performance on that task, where the ceiling is the percentage agreement found between two annotators (interannotator agreement). Fot

POS tagging, this has been reported as 96% (which makes existing POS taggers look impressive). However this raises lots of questions: relatively untrained human annotators working independently often have quite low agreement, but trained annotators discussing results can achieve much higher performance (approaching 100% for POS tagging). Human performance varies considerably between individuals. In any case, human performance may not be a realistic ceiling on relatively unnatural tasks, such as POS tagging.

Error analysis The error rate on a particular problem will be distributed very unevenly. For instance, a POS tagger will never confuse the tag PUN with the tag VVN (past participle), but might confuse VVN with AJ0 (adjective) because there's a systematic ambiguity for many forms (e.g., given). For a particular application, some errors 25 may be more important than others. For instance, if one is looking for relatively low frequency cases of demonical verbs (that is verbs derived from nouns . e.g., canoe, tango, fork used as verbs), then POS tagging is not directly useful in general, because a verbal use without a characteristic affix is likely to be massaged. This makes POS-tagging less useful for lexicographers, who are often specifically interested in finding examples of unusual word uses. Similarly, in text categorization, some errors are more important than others: e.g. treating an incoming order for an expensive product as junk email is a much worse error than the converse.

Reproducibility If at all possible, evaluation should be done on a generally available corpus so that other researchers can replicate the experiments.

➢ **Morphology Fundamentals - Types**
♦ Automatic Morphology Learning,
♦ Finite State Machine Based Morphology
♦ **Shallow Parsing :**
  ▪ Shallow parsing is an analysis of a sentence which identifies the constituents (noun groups or phrases, verbs, verb groups, etc.), but does not specify their internal structure, nor their role in the main sentence.
  ▪ It is a technique widely used in natural language processing.
  ▪ It is similar to the concept of lexical analysis for computer languages. Under the name of the Shallow Structure Hypothesis, it is also used as an explanation for why second language learners often fail to parse complex sentences correctly.
  ▪ In this technique, we get hierarchical and grammatical information while preserving robustness and efficiency of the processing.
  ▪ Shallow parsing technique can be seen as a set of production/reduction/cutting rules. ·
    Rule 1: Open a phrase p for the current category c if c can be the left corner of p.
    Rule 2: Do not open an already opened category if it belongs to the current phrase or is its right corner. Otherwise, we can reopen it if the current word can only be its left corner.
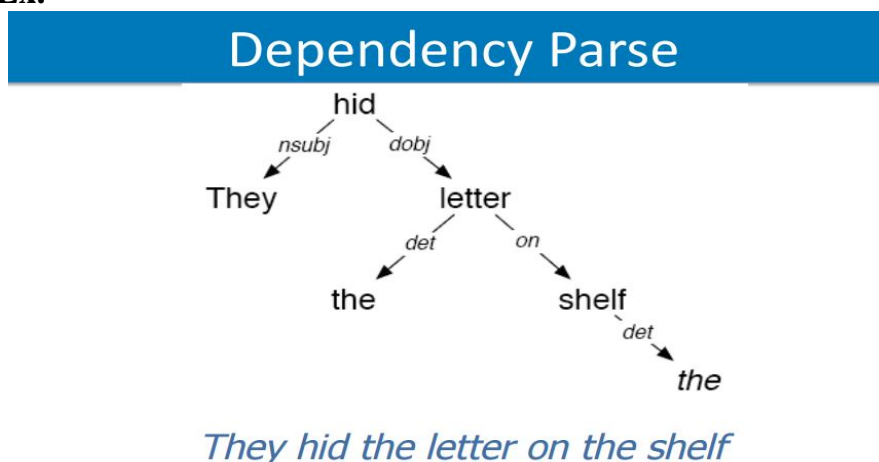   **Rule 3:** Close the opened phrases if the more recently opened phrase can neither neither continue one of them nor be one of their right corners.
    Rule 4: When closing a phrase, apply rules 1, 2 and 3. This may close or open new phrases taking into consideration all phrase-level categories.

Chapter 5

➢ **Dependency Parsing**
- The dependency approach has a number of advantages over full phrase-structure parsing.
  - Deals well with free word order languages where the constituent structure is quite fluid
  - Parsing is much faster than CFG-bases parsers
  - Dependency structure often captures the syntactic relations needed by later applications - CFG-based approaches often extract this same information from trees anyway.
  - **Ex. –**

They hid the letter on the shelf

➢ **Malt Parser**

Chapter 6

➢ **Lexical Knowledge Networks**

➢ **WordNET Theory**
- There are several electronic dictionaries, thesauri, lexical databases, and so forth today. WordNet is one of the largest and most widely used of these.
- It has been used for many natural language processing tasks, including word sense disambiguation and question answering.
- This is an attempt to explore and understand the structure of WordNet, and how it is used and for what applications it is used, and also to see where it's strength and weakness lies
- WordNet is the main resource for lexical semantics for English that is used in NLP. Primarily because of its very large coverage and the fact that it's freely available. WordNets are under development for many other languages, though so far none are as extensive as the original.

- The primary organisation of WordNet is into synsets: synonym sets (near-synonyms).
- The following is an overview of the information available in WordNet for the various POS classes:
  - all classes
    - synonyms (ordered by frequency)
    - familiarity / polysemy count
    - compound words (done by spelling)
  - nouns
    - hyponyms / hypernyms (also sisters)
    - holonyms / meronyms
  - adjectives
    - antonyms
  - verbs
    - antonyms
    - hyponyms / hypernyms (also sisters)
    - syntax (very simple)
  - adverbs
- Application or Use of WordNET:
  - WordNet has been used for a number of different purposes in information systems, including word sense disambiguation, information retrieval, classification, automatic, machine translation and even automatic crossword puzzle generation.
  - A common use of WordNet is to determine the similarity between words. Various algorithms have been proposed, and these include measuring the distance among the words and synsets in WordNet's graph structure, such as by counting the number of edges among synsets. The intuition is that the closer two words or synsets are, the closer their meaning. A number of WordNet-based word similarity algorithms are implemented in a Perl package called WordNet::Similarity, and in a Python package called NLTK. Other more sophisticated WordNet-based similarity techniques include ADW, whose implementation is available in Java. WordNet can also be used to inter-link other vocabularies.

Semantic Roles

- Once the computer has arrived at an analysis of the input sentence's syntactic structure, a semantic analysis is needed to ascertain the meaning of the sentence.
- The basic or primitive unit of meaning for semantic will be not the word but the sense, because words may have different senses, like those listed in the dictionary for the same word.
- It is concern with what words mean and how these meaning combine in sentence to form sentence meaning.
-

➢ **Metaphors;**
➢ **Word Sense – Application**

- Needed for many applications, problematic for large domains. Assumes that we have a standard set of word senses (e.g., WordNet)
  - frequency: e.g., diet: the food sense (or senses) is much more frequent than the parliament sense (Diet of Wurms)
  - collocations: e.g. striped bass (the _sh) vs bass guitar: syntactically related or in a window of words (latter sometimes called `cooccurrence'). Generally `one sense per collocation'.
  - selection restrictions/preferences (e.g., Kim eats bass, must refer to fish

- A combination of unsupervised Knowledge-based and supervised Machine Learning techniques that will provide a high-precision system that is able to tag running text with word senses
- A system that acquires a **huge number of examples** per word from the web
- The use of **sophisticated linguistic information**, such as, syntactic relations, semantic classes, selectional restrictions, subcategorization information, domain, etc.
- Efficient margin-based **Machine Learning algorithms**.
- Novel algorithms that combine tagged examples with huge amounts of **untagged examples** in order to increase the precision of the system.