CHAPTER 3

CONCEPTS OF LANGUAGE AND PARSING MODELS

Language model is an important component for statistical methods since it records patterns of occurrences of strings in a corpus and predicts a sequence of words in a new context. Similarly parsing model is trained with structural information provided by annotated corpus. It provides structural information for a sentence in a new context. Language and parsing models are the essential components which assist statistical and parsing applications respectively. Language models provide probability values for the reduction of search spaces in predicting a sequence of words. Parsing models help in finding n-best parses for a new sentence. This chapter deals with the basic concepts related to the development of language and parsing models, and the nature of Tamil language.

3.1 NATURAL LANGUAGE PROCESSING AND COMPUTATIONAL LINGUISTICS

NLP and CL are concerned with the use of artificial intelligence to process NL sentences in a wide range of applications. NL processes are carried out by analyzing syntax, semantics and pragmatics. There are various methods available based on linguistic rules, dictionary, statistics, example-based approach, cross-lingual approach etc. Linguistics comprises sub-fields like study of grammar and semantics. The study of grammar includes morphology and syntax. Morphology deals with formation and derivation of words. Syntax enforces the rules that determine the way in which words combine into phrases and phrases into sentences. Semantics deals with the
meaning of words and how word combinations form the meaning of a sentence. Language model covers semantics in terms of unigram, bigram, trigram and n-gram patterns. Parsing model covers syntax, semantics and pragmatics in terms of probabilities among parses.

### 3.2 LANGUAGE MODELS

Language model is the component which aids efficient searching through n-grams to obtain most likely sentences for a sequence of words. It is a mechanism which assists in generating output sentences from the sequence of words generated by any NLP application through the probability over words. The output of any NLP application is the optimal sequence of words which are available in the dictionary. This can be achieved by means of a statistical language model which is primarily data driven and it reduces search spaces of a decoder. So a language model needs a large volume of training corpus for coverage, accuracy and robustness.

In n-gram language model, each word depends probabilistically on n-1 preceding words. This is expressed as shown in equation (3.1).

\[
p(w_{i,n}) = \bigotimes_{i=1}^{d} p(w_i \mid w_{i- (n-1)}, ..., w_{i-1}) \quad (3.1)
\]

When \( n \) is big, memory and processing power requirements are high. Results obtained for \( n=3 \) are quite satisfactory (Jelinek 1997). This is called tri-gram language model, where each word depends probabilistically on previous two words and is shown in equation (3.2).

\[
p(w_{i,n}) = \bigotimes_{i=1}^{d} p(w_i \mid w_{i-2}, w_{i-1}) \quad (3.2)
\]
A trigram language model is more suitable due to capacity, coverage and computational power. For shaping a trigram model into a greater level of suitability advanced and optimizing techniques like smoothing, caching, skipping, clustering, sentence mixing, structuring and text normalization are applied (Joshua Goodman 2003). Through these techniques marginal improvements in perplexity are obtained. Even though a statistical model gives better performance, proper meaning cannot be derived for compound sentences due to tri-gram hits which capture local dependencies (Jelinek 1997). A language model is normally static which is developed with available corpus. To increase coverage, capacity and robustness, it is to be adapted with new training set.

3.3 LANGUAGE MODEL ADAPTATION

English and French have sufficient resources such as text corpora. New documents are used for adaptation to improve topic specific language models after identification of topics. Initial language model can be developed with available corpus for resource deficient languages. New lexicons and their probabilities are to be generated dynamically from languages like English using statistical machine translation or CLSA and used in adaptation by interpolation technique (Kim and Khudanpur 2003, 2004a, 2004b). Adaptation can be done in common or domain independent language model directly or through topic specific models after identification of topics.

3.4 PARSING

Parsing is done with rule based method if the language has a well-defined set of syntactic rules. Parsing NL text is challenging because of ambiguity and inefficiency. A parser helps in disambiguation and interpretation of NL text through context based techniques. It can be done
with the help of parsing models which consist of probability distribution over parses.

3.5 PARSING MODELS

Phrase and dependency structures are the two ways to create parsing model. Phrase structure model is created using phrase structured Treebank by means of immediate head parsing technique. Heads are selected from various constituents and the trigram approach is applied among the heads. Let $c$ be a constituent of a parse. For all features of the constituent $c$, feature files are generated and updated during the training process. The features include pre-terminal, lexical head, expansion and history of the constituent $c$. The history includes label, head and head-part-of-speech for the parent of the constituent $c$. All feature files together constitute this hybrid model.

Dependency model is created using non-projective dependency parsing technique with dependency Treebank. Here, the probabilities are computed over edges which represent dependency relations between the modifier and head word of the edges in a sentence (Chelba and Engle 2000). Features are defined and updated during training which include directions of attachment, distance between words and contextual features.

3.6 TREEBANK

Treebank is a language resource that provides linguistic annotations of NL sentences at word, phrase, sentence and function-argument or dependency level structures. Treebank is crucial for the development of data-driven approaches for NLP, human language technologies, grammar extraction and linguistics. Treebank is the annotated corpus meant for automatic construction of parsing models, and evaluation and comparison of
various parsing models. It is developed with morphological, syntactic, semantic, and contents/pragmatic annotations. Treebanks are highly useful in providing training material for machine learning systems. Some of the classifications of Treebank are phrase structured Treebank and dependency Treebank.

3.6.1 Phrase Structure Treebank

Simple bracketed version of Treebank is generated by phrasing sentences in a text corpus. It is a constituency based format done by manual annotation and bootstrapping or automated tools like morphological analyzer, POS tagger and phrasing tool with manual corrections. Phrase level annotation provides a simple syntax tree for a sentence.

![Figure 3.1 Syntax Tree](image)

Old cotton will not absorb water

Figure 3.1 Syntax Tree

Figure 3.1 shows an example of a syntax tree. Each leaf node contains a word and its POS tag. The leaf nodes are grouped into various syntactic phrases in the next level. Similarly the phrases are grouped into other phrases until the root node is reached. In this figure, ADJ, NNSN, NNSNA, CVPP and VTSNFN are POS tags and NP, PP and VP are noun, prepositional and verb phrases respectively.
3.6.2 Dependency Structure Treebank

A dependency tree is projective when words are in linear order preceded by a root. Edges can be drawn above the words without crossings. In other words, a word and its descendants form a contiguous substring of a sentence. Figure 3.2 is an example of a projective dependency tree. Projective trees are sufficient to analyze most of the sentences in English.

Water should not come out in the tip of the small tap

Figure 3.2 Projective Dependency Tree

Here each word is annotated with POS tag to know the lexical information of the word and each word’s dependent relation is also annotated. Here, the word (small) depends on (of the tap) as DEP (simply dependent), depends on (from the tip) as NP (Noun Phrase), depends on (should not come) as NP, (water) depends on as NP-OBJ (Noun Phrase Object) and finally is the root word (ROOT). In this figure, ADJ, NNSNG, NNSNL, NNSN and VTSNFN are POS tags.

For free-word order languages like Tamil, non-projectivity is a common phenomenon since the relative positional constraints on dependents are less rigid. Tamil reduces reliance on word order to express grammatical relations and allows non-projective dependencies that need to be represented
and parsed efficiently. An example of non-projective dependency graph is shown in Figure 3.3.

![Non-Projective Dependency Graph](image)

**Figure 3.3 Non-Projective Dependency Graph**

In Figure 3.3, the word (Raman) depends on (to study) as NP-SBJ (Subject Noun Phrase), (to school) depends on (went) as NP (Noun Phrase), (subject) depends on as NP-OBJ (Object Noun Phrase), depends on as VP (Verb Phrase) and finally is the root word (ROOT).

Dependency relations between words are analyzed and marked with an index. Since Tamil is a head final language, verb is the root word which is assigned with an index 0. The period (.) is inserted as a dummy root with index of the root word. Similarly other words are assigned with index of their respective head words. Labels are used to identify dependency relations. POS tags provide additional information to words. An example sentence in MST format is shown below:
3.7 PHRASING AND DEPENDENCY ANNOTATIONS

Phrasing is the process of applying morpho-syntactic relations among words in the formation of constituents which in turn build parse trees. A collection of annotated parse trees is called Treebank. It is useful for many applications like information retrieval, spelling correction, text to speech synthesis, terminology extraction and mining. Dependency annotation deals with the application of subject-predicate relationship among words by providing direct edges in order to account for the long distance relationship in a sentence. It is essential to use POS tagging for phrasing and dependency annotations.

3.8 POS TAGGING

POS tagging is used to assign or select correct POS tags to words in a sentence before syntactic analysis. POS tagging needs thorough morphological analysis for the identification of POS for the words (Manish Shrivastava et al 2005). It assists in applying phrases to create various constituent units in phrase structure model and in providing feature information in dependency structure model.

3.9 MORPHOLOGICAL ANALYSIS

Morphological analysis is the process of segmenting a given word into a sequence of morphemes. It is closely related to POS tagging. It is the
process through which inflection of words is studied. Inflectional morphology gives the different forms of a root word. Derivational morphology derives new words by inclusion of affixes. Lexical and surface levels of words are studied through morphological analysis. Based on that, POS tags are applied to words in a sentence. Linguistic rules can be applied through morphological analysis.

3.10 TAMIL LANGUAGE

The grammar of Tamil is agglutinative in nature. A Tamil word may have a lexical root to which one or more affixes are attached. Most of the Tamil affixes are derivational or inflectional suffixes and the length and extent of agglutination is longer in Tamil resulting in longer words with many suffixes. Other issues are morpho-phonology (sandhi – insertion, deletion and substitution of morphemes like , , , , at word boundaries) rules, complex noun and verb patterns and out of vocabulary rate due to inflections. Poetry forms are more complex than prose forms.

In Tamil, nouns are classified into rational and irrational forms. Human comes under rational whereas all other nouns are classified as irrational. Rational nouns and pronouns belong to one of the three classes: masculine singular, feminine singular and rational plural. Irrational nouns belong to one of the two classes: irrational singular and irrational plural. Suffixes are used to mark class, number and cases attached to a noun. Tamil verbs are also inflected through the use of suffixes. The suffix of the verb indicates person, number, gender, mood, tense, voice, negation and interrogation.

The verb comes at the end of the clause with a typical word order of Subject Object Verb (SOV). However, Tamil language allows word order to be changed, making it a relatively free-word-order language. Other features
are plural for honorific noun, frequent echo words, and null subject feature, i.e. some sentences do not have subject, verb and object.

3.11 SUMMARY

The foundational concepts related to the development language and parsing models have been described. Illustrative examples in Tamil have been given for phrase and dependency structure treebanks. The nature of Tamil language and its grammatical structure have also been discussed.