**CHAPTER 1**

**INTRODUCTION**

The purpose of this chapter is to introduce the basic concepts of Shallow parser that is developed throughout the project. Section 1.1 summarizes the introduction to Shallow parser including some examples. It throws light on the motivation behind the project and also lays down objectives behind undertaking the project. Lastly, it describes the scope of the project. Section 1.2 briefly includes previous work carried out in this field, researching the problem studied, summarization of the results obtained etc. At the end, section 1.3 describes the organization of the project and gives an outline of the focus of discussion in further chapters.

**1.1 PROBLEM INTRODUCTION**

Shallow parsing (also chunking, "light parsing") is an analysis of a sentence which identifies the constituents (noun groups, verbs, verb groups, etc.), but does not specify their internal structure, nor their role in the main sentence. Shallow parser is the tool that performs shallow parsing. In other words, a shallow parser extracts syntactically related group of words from a sentence.

Shallow parsing has become an interesting alternative to full parsing. The main goal of a shallow parser is to divide a text into segments which correspond to certain syntactic units. Although the detailed information from a full parse is lost, shallow parsing can be done on non-restricted texts in an efficient and reliable way. In addition, partial syntactical information can help to solve many natural language processing tasks, such as information extraction, text summarization, machine translation and spoken language understanding.

Shallow parsing involves several different tasks, such as text chunking, noun phrase chunking or clause identification. *Text chunking* consists of dividing an input text into non overlapping segments. These segments are non-recursive, that is, they cannot include other segments and are usually called *chunks* as defined by Abney (1991). Noun phrase chunking (*NP chunking*) is a part of the text chunking task, which consists of detecting only noun phrase chunks. The aim of the *Clause identification* task is to detect the start and the end boundaries of each clause (sequence of words that contains a subject and a predicate) in a sentence. For example, the sentence *“You will start to see shows where viewers program the*

*Program”* would be chunked as follows:

(NP You) (VP will start to see) (NP shows) (ADVP where) (NP viewers)

(VP program) (NP the program)

The clauses in the sentence would be:

(S You will start to see shows (S where (S viewers program the program )) .)

Chunks and clause information in a sentence can also be represented by means of tags. In Tjong Kim Sang et al. (2000), there are several equivalent chunk tag sets for representing chunking. The *IOB2* set, which was previously used by Ratnaparkhi (1998), uses three kinds of tags:

B-X for the first word of a chunkof type X;

I-X for a non-initial word in an X chunk;

O for a word outside of any chunk.

For clause identification, each word can be tagged with the corresponding brackets if the word starts and/or ends a clause, or with a null tag if the word is not the start or the end of a clause. The above example can be represented using this notation as follows:

You B-NP (S\*

’ll B-VP \*

start I-VP \*

to I-VP \*

see I-VP \*

shows B-NP \*

where B-ADVP (S\*

viewers B-NP (S\*

program B-VP \*

the B-NP \*

program I-NP \*S)S)

. O \*S)

* + 1. **MOTIVATION**

A shallow parser is important in NLP applications that don’t require full syntactic analysis of the sentence but require more information than POS taggers can provide. Key applications for shallow parser include information retrieval, information extraction and summary generation. Shallow parser answers questions like specific syntactic-semantic relations (agent, object, location, time, etc.) rather than elaborate configurational syntactic analysis.

Application domain of shallow parsers include speech-to-speech translation systems where they are used to add robustness, question answering on internet where they are used to efficiently process ill-formed documents and text-mining applications. They are used to reduce search space for full-blown ‘deep’ parser.

* + 1. **PROJECT OBJECTIVE**

The objective behind the project ‘Shallow Parser for Hindi Language’ is to develop a shallow parser for Hindi Language which can be used as a tool in building more application specific tools like auto-text summarizer, speech-to-speech translators etc.

Key objectives of the project are:

* To improve the robustness of existing shallow parsers for Hindi language.
* To design a shallow parser for Hindi language

**// TODO – need to add more objectives**

* + 1. **SCOPE OF THE PROJECT**

The project comprises construction of shallow parser for Hindi language for a specific subset of the words chosen. Shallow parser shall be able to extract constituents of grammatically simple and unambiguous sentences of Hindi language. Project will be implemented as a web based application and won’t include a desktop or mobile application.

Sentences chosen for testing will comprise of words selected as part of the project and will be grammatically and semantically valid sentences of the language. Any deviation from the optimal and correct structure of the sentence may fail the system.

Input will be a single sentence in Hindi language and Output will be chunked output of the sentence.

**1.2 RELATED PREVIOUS WORK**

Earlier approaches to solving this problem consisted of parsers which are based on a grammar of hand-coded rules. Abney (1996) developed the incremental partial parser *CASS* based on finite state methods for detecting chunks and clauses. A’ıt-Mokhtar and Chanod (1997) also built an incremental architecture of finite-state transducers that identifies chunks and detects subjects and objects. Voutilainen (1993) used a different formalism (constraint grammars) to detect NPs.

In the literature, we can find different learning methods which have been applied to perform shallow parsing:

* Transformation-based Learning
* Memory-based Learning
* Hidden Markov Models
* Maximum Entropy
* Support Vector Machines, etc.

The first works focused mainly on NP detection. Ramshaw and Marcus (1995) used Transformation-based learning and put forward a standard training and testing data set, that has later been used to contrast other approaches. Memory-based learning was used by Daelemans et al. (1999) and Argamon et al. (1998). The main reference for text chunking is the shared task for CoNLL-20002 (Tjong Kim Sang and Buchholz, 2000).

Learning approaches for clause identification have recently been developed. Orasan (2000) applied memory-based learning techniques and corrected the output by applying some rules. In the shared taskfor CoNLL-20013 (Tjong Kim Sang and D´ejean, 2001) other approaches were presented (Hidden Markov Models, Memory-based Learning, Boosting, etc.)

* 1. **ORGANIZATION OF THE REPORT**