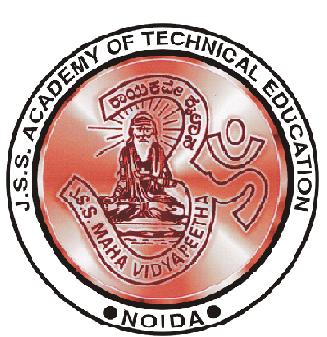
## Synopsis

## Of

## **Shallow parser for Hindi Language**

Bachelor of Technology

Computer Science and Engineering



Submitted by

|  |  |
| --- | --- |
| Akshat Goel | 0909110011 |
| Adiwitiya Singh | 0909110010 |
| Mohit Kumar Singh | 0909110066 |
| Juhi Rajput | 0909110050 |

Submitted to

Mrs. Seema Shukla

Assistant Professor

Department of Computer Science and Engineering

### **Department of Computer Science and Engineering**

### **JSS Academy of Technical Education, Noida**

# Table of Contents

1. Introduction
   1. Motivation
   2. Project Objective
   3. Scope of the Project
   4. Pert Chart / Gantt Chart
   5. Technical Feasibility
   6. References

# Shallow Parser for Hindi Language

## 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.

A shallow parser is different from a parser as well as a Part-of-Speech (POS) tagger. A parser converts a sentence into tree where leaf nodes hold POS tags while rest of the tree constitutes the overall syntactic structure of the sentence on how these words are arranged. A POS tagger gives POS tags of each word in the input sentence. A shallow parser comes in between of these two.

Example:

Sentence: My dog likes his food.

POS tagging output: My/PRP$ dog/NN likes/VBZ his/PRP$ food/NN./.

Chunking output: [NP My Dog] [VP likes] [NP his food]

* 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:

1. To improve the robustness of existing shallow parsers for Hindi language.
2. To design a shallow parser for Hindi language.
   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. Gantt Chart

@To-do Gantt chart comes here.

* 1. Technical Feasibility

The project will be implemented as a web application. Desktop or mobile versions of the project may be included later but are not within the scope of the project as stated in the above section. PHP will be the server language while front-end will be built using open web technologies like HTML5, CSS3 and JavaScript. C and Java libraries will be used for the construction of actual parser. Providing the service as web application may affect the performance of the parser but will provide high availability and distributed nature to the parser.

LAMP will constitute the server-stack for providing infrastructure to the application. Linux will be the development operating system due to availability of large number of open-source libraries that may be needed for the implementation.

* 1. References

1.6.1 Research Papers

* http://users.cis.fiu.edu/~lli003/Sum/RIAO/2007/1.pdf
* <http://www.sriste.com/index.php?journal=IJITT&page=article&op=viewFile&path%5B%5D=6110&path%5B%5D=pdf>
* http://www.cse.msu.edu/~cse842/Classnotes/Lecture2-Morphology.pdf
* http://www.cs.cmu.edu/~madhavi/publications/Ganapathiraju\_11-742Report.pdf

1.6.2 Websites

* http://www-limbio.smbh.univ-paris13.fr/membres/hamon/hybrid/
* http://130.203.133.150/viewdoc/summary;jsessionid=7CD06A2D64AB993AAAFE233FA50FD042?doi=10.1.1.64.7206
* http://ieeexplore.ieee.org/xpl/articleDetails.jsp?reload=true&arnumber=5646317&contentType=Conference+Publications
* https://www.classle.net/projects/project\_ideas/development-auto-
* summarization-tool
* http://www.copernic.com/en/products/summarizer/index.html
* http://www.sourcecodesworld.com/project-bank/project3.asp

1.6.3 Books

* O'Conner, Joseph, Introducing NLP: Psychological Skills for Understanding and Influencing People
* Allen, James., Natural Language Understanding