Functional Parsers For Indian Languages

AIM

To investigate various Natural Language Processing tools that are written in functional languages, chiefly Haskell. Following the above, to use said tools to implement a Paninian Grammar framework for Indian languages.

INTRODUCTION

The construction of tools to parse natural languages is a multi-layered challenge for programmers. The question we pursue here is: are functional languages suitable for building tools to parse Indian languages?

The process of parsing a language involves several steps, and in this study we mainly look towards syntactic parsing, i.e: parsing a sentence in terms of the syntactic roles of the words in a sentence.

METHODS

1) We implemented a dependency parser engine in Haskell, following the model laid down in the paper Dependency Grammars as Haskell programs by Tomasz Obrebski. This project demonstrates that a parser engine as well as the associated grammar for the parser can be directly expressed in the Haskell language as functions, types or relations between them.

2) Exploration of the Grammatical Framework programming language- a functional, strongly-typed language created to write grammars for rule-based NLP applications. GF is based on and written in Haskell.

CONCLUSIONS

Functional languages are well-suited for implementing NLP interfaces for Indian languages.

Haskell, with it’s properties of higher-order abstraction, lazy evaluation and it’s type system, is ideal for building dependency parsers and also for expressing dependency grammars directly as Haskell code.

More work needs to be done in the direction of Grammatical Framework, to expand it to fit a Paninian Grammar framework that is more suited for Indian languages.

Intern Name: Atreyee Ghosal
Mentor: Venkatesh Choppella