Tecnológico de Monterrey

Implementación de métodos computacionales

Parseo en LISP

Campus: Toluca

Profesor:

José María Aguilera Méndez

Alumno:

Pedro Enrique Mendoza Garcia ITC | A01772504

Fecha de creación:

29 de marzo del 2024

Índice

Introducción:

Desde la existencia del hombre y su afán de razonamiento este ha buscado siempre la manera de facilitar tareas y otros trabajos que pueden llevar a alguien comun horas, o inclusive días, desde ese pensar se inventaron las primeras generación de las computadoras, que no eran más que nada complejas cajas conectadas basadas en lenguaje máquina para poder solo realizar la tarea específica para la que fueron hechas, sin embargo el costo de estas mismas era altísimo.

Con el paso del tiempo estas mismas fueron evolucionando así como también la ambición del hombre por hacer que la forma de interactuar con ella sea más agradable y sencilla de modo que entienda y que aprenda de lo que lo rodea y reciba, viéndose así un proyecto futurista para la época llamado Inteligencia artificial que consistía en que la máquina tuviera la capacidad de razonar lo escrito. Hecho que marcaría un antes y despues de como la ciencia computacional se desenvolveria y como es que gente con visión empezaban a crear herramientas computacionales para la ayuda del desarrollo de la IA, sin embargo de los principales problemas con los que se encontraron serían ¿Cómo una computadora puede entender lo que yo digo?

Es de esa pregunta que tomaría en cuenta el “Lenguaje natural”, siendo entonces prioridad encontrar una forma en la que una computadora pueda comprender qué es lo que dices, así como entender la estructura de una oración, más tarde a esto se le llamaría “PLN” (Procesamiento de lenguaje natural). La idea era fascinante y el proyecto ambicioso, sin embargo para el momento de pensar esto, la limitante de lenguajes así como de el continuo desarrollo de las computadoras frenó este mismo.

Un joven John McCarthy creo de los primeros lenguajes computacionales llamado Lisp, Lisp tenía la característica singular de poder manejar el uso de lista y cadenas de una manera mas facil, asi como un uso de notación polaca para su sintaxis, de ahi su nombre List Processing.

Este documento redacta los pasos que seguí, la forma de pensar así como también el desarrollo de un “PLN” trabajado en Common Lisp, que tiene como objetivo general reafirmar el entendimiento de una oración escrita y enviada, así como el retorno de un mensaje de error en caso de una mala escritura de la oración enviada

Desarrollo

Se pensó en un programa que pudiera analizar la sintaxis de una oración que se otorga por parte del usuario, esta oración tiene que ser revisada y analizada en otros aspectos, los más vitales es poder separar palabra por palabra, quitar nulos o elementos vacíos, Y como valor agregado a mi codigo hago que la oración contenga un punto final "." para así poder determinar hasta dónde llega, siendo esa mi condición de salida.

Después de que la oración ya tuvo su primer proceso ahora sigue el análisis sintáctico, en donde empezará desde la función que empezará a determinar el camino que idóneamente debe de tener la oración, si esta llega a fallar al encontrar un palabra que no puede reconocer, este deberá de ser capaz de regresarte la palabra en la que no pudo reconocer, asi mismo un mensaje. En el mejor de los casos el sintagma que hizo el último chequeo devolverá un mensaje de que la oración está correcta.

Como extra trate de implementar la concordancia de un artículo con sus respectivos géneros de sujetos, así como si tmb existen neutros y en singular y plural, así como tmb en caso de existir un adjetivo que tenga esa misma concordancia con el artículo, ya que este puede también contener un sustantivo neutro.

Estructura del lenguaje:

El programa busca el camino de acuerdo a las posibilidades de la siguiente estructura

Oración → SN + SV + .
SN:

Artículo + Sustantivo + Adjetivo
Artículo + Sustantivo
Preposición + SN

- Los artículos deben de coincidir con su sustantivo en masculino y en femenino, en plural y singular

SV:

Verbo
Verbo + SN
Verbo + Adverbio
Verbo + Adverbio + SN

Consideraciones:

- Todas las palabras que ingreses deben de estar dentro del diccionario, el programa no es sensible ante mayúsculas, pero sí lo es ante los acentos
- Toda oración debe terminar en un punto
- El programa verifica la concordancia de género, con plurales y singulares del adjetivo con respecto al artículo
- El programa verifica el sustantivo con respecto a su articulo, asi como tambien trata de abarcar ciertas palabras neutras de género en donde el artículo no importa con respecto al sujeto, sin embargo si considera su plural y singular

Restricciones:

En ningún momento se pueden usar comas “,”, tampoco dos puntos “:” etc etc…
El programa no puede verificar preguntas
El programa no puede aceptar dos preposiciones o una serie de preposiciones juntas.
Ejemplo:

que que que a.

El programa no puede ser capaz de identificar conjugaciones de los verbos y tiempos.

Ejecución:

Para ejecutar el programa es necesario abrir nuestra terminal de SBCL y escribir la instrucción (load “path del archivo”), coloque el path en donde se ubica el archivo que ejecutaras, es importante colocar las comillas, asi como abrir y cerrar los parentesis.
ejemplo de path: `"/Users/Hp/Downloads/ProyectoLisp/parseo.lisp"`

De igual manera el programa requiere de un diccionario que contiene dentro las palabras que puede analizar, existen una función que el cambio que se debe realizar el cambiar el path hacia la carpeta diccionario, de ese modo los demás métodos ya podrá acceder al diccionario y sus distantia distribución de palabras para el análisis.

Ejemplos:

**** el motor robusto es peor que el motor liviano***
"A tu oración le hace falta un punto al final"
"ERROR"

**** la motor robusto es peor que el motor liviano.***
'la' y 'motor' no coinciden, verifica que estén bien escritas y que sean coherente

**** el motor robusto es peor que la motor liviano.***
'la' y 'motor' no coinciden, verifica que estén bien escritas y que sean coherente

**** es peor que el motor liviano.***
"SV1: Chido"

**** la piloto rapido maneja el monoplaza.***
'la piloto rapido' no coinciden

**** La piloto rapida maneja el monoplaza.***
SN3: Chido