Universidad de Ingeniería y Tecnología

Ciencia de la Computación

Natural Language Processing with Context Free Grammar

Proyecto - Teoría de la Computación

Autores:
Christian Ledgard Ferrero
Anthony Aguilar
Alexander Baldeon

Mayo 2020

Introducción

Hoy en día la tecnología está presente en muchos ámbitos de nuestra sociedad. Desde vehículos que se manejan por si solos hasta en nuestros celulares que nos permiten estar más comunicados que nunca. Unos de avances más significativos en los últimos tiempos son los asistentes virtuales como Siri, Alexa, Google Assistant, entre otros. ¿Alguna vez se han preguntado cómo estos sistemas organizan y analizan nuestras oraciones para ponerlas en un determinado contexto? Una posible implementación es utilizar **gramáticas independientes del contexto** para organizar y luego generar un árbol de *"parseo"* con la oración. Esto se logra al tener un léxico y una gramática previamente definida.

Alcance

En nuestro proyecto nosotros realizaremos una investigación aplicada en donde desarrollaremos un algoritmo simple para realizar un parser de acuerdo a un léxico y gramática previamente definida. Utilizaremos diversas fuentes (1), (2), (3), (4), (5), (6), (7), (8) y (9).

Definición del Problema

Cómo dividir una oración, utilizando gramáticas independientes del contexto, para que el computador pueda interpretarlas correctamente. ¿Casos de ambigüedad? ¿Qué hacer en determinados casos?

Estado del Arte

El inicio del procesamiento de lenguaje natural suele remontarse a 1950 cuando Alan Turing publico el articulo 'Computing Machinery and Intelligence' donde propone lo que conocemos como el test de Turing, un criterio de inteligencia.
Una de las primera aplicación del NLP ocurrió en 1954 e involucro la traducción de 60 oraciones rusas al ingles.
En los 1960´s ELIZA y SHRDLU fueron sistemas capaces de comunicarce

con humanos.
Existen al menos 4 algoritmos de parsing

1. El algoritmo CYK (Cocke–Younger–Kasami)
   1.1. Solo trabaja con CFG en la forma normal de chumsky
   1.2. El peor de los casos es $O(n^3.|G|)$ time. Donde n es el tamaño de la cadena y $|G|$ es el tamaño de la gramática G lo que lo hace uno de los mas eficientes en el peor de los casos.
2. Earley parser
   2.1. Trabaja con todos las gramáticas libres del contexto.
   2.2. Tiene un tiempo de ejecución $O(n^3)$ en el caso promedio,$O(n^2)$ en gramáticas no ambiguas y $O(n)$ para CFG deterministas.
3. LR parser (Left-to-right, Rightmost derivation in reverse)
   Son tipos de bottom-up parsers que analizan DCFG en tiempo liniar.
   3.1. LALR parser
   3.2. Canonical LR parser
   3.3. Minimal LR parser
   3.4. GLR parser
4. LL parser (Left-to-right, Leftmost derivation)
   Es un top-down parser para un subset de CFG

Propuestas

- Implementar y analizar un algoritmos de parseo para construir una gramática independiente del contexto bajo una gramática y un lenguaje dado.
- Analizar posibles implementaciones y soluciones en casos de ambigüedad y presentar propuestas utilizadas actualmente por grandes empresas como Amazon o Google.

Bibliografía

[1] B. Box, *Natural Language Processing 5 3 Context Free Grammars Part 1 1211*, 2018. Disponible en https://www.youtube.com/watch?v=VkXSpWc_8FM.

[2] A. McCallum, *Context Free Grammars, Introduction to Natural Language Processing*, 2017. Disponible en https://people.cs.umass.edu/~mccallum/courses/inlp2007/lect5-cfg.pdf.

[3] B. College, *Overview of NLP: Issues and Strategies*. Disponible en http://www.bowdoin.edu/~allen/nlp/nlp1.html.

[4] E. Shutova, *Context-free grammars and parsing*, 2015. Disponible en https://www.cl.cam.ac.uk/teaching/1516/L90/slides4-public.pdf.

[5] Wikipedia, *Context-free grammar*. https://en.wikipedia.org/wiki/Context-free_grammar.

[6] Wikipedia, *CYK Algorithm*. https://en.wikipedia.org/wiki/CYK_algorithm.

[7] Wikipedia, *Earley parser*. https://en.wikipedia.org/wiki/Earley_parser.

[8] Wikipedia, *LR parser*. https://en.wikipedia.org/wiki/LR_parser.

[9] Wikipedia, *LL parser*. https://en.wikipedia.org/wiki/LL_parser.