Gramáticas Libres de Contexto

Este tipo de gramática se usa mucho en **procesamiento de lenguaje natural (PLN)** y **compiladores** para definir **estructuras válidas de una lengua**, ya sea un lenguaje natural (como el español o inglés) o un lenguaje formal (como el código fuente de un programa). Las gramáticas Libre de Contexto se utilizan principalmente para analizar la lingüística para poder entender la estructura del lenguaje natural hablado o escrito. Se estudia la forma de generar o aceptar cadenas o strings de una forma declarativa.

Una gramática libre o independiente del contexto es una notación formal que sirve para expresar las definiciones recursivas de los lenguajes. La gramática:

R → X R X | S

S → a T b | b T a

T → X T X | X | ε

X → a | b

EXPLICACION DE LAS GRAMÁTICA FORMALES:

Una gramática formal es un conjunto de reglas que nos permite generar cadenas en un lenguaje. Estas reglas nos dicen cómo construir frases o palabras válidas usando símbolos.

Una gramática tiene cuatro componentes principales:

G = (V, T, P, S)

V: Conjunto de variables (también llamadas no terminales).

T: Conjunto de terminales.

P: Reglas de producción.

S: Símbolo inicial.

¿QUÉ ES UN TERMINAL?

Los terminales son los símbolos finales del lenguaje, es decir, los que aparecen en la cadena generada. No se pueden reemplazar por nada más. Son como las letras o palabras reales de un idioma.

X → a | b

Los terminales son 2: a y b.

Una vez que aparece una a o una b, ya no se puede cambiar más.

¿QUÉ ES UN NO TERMINAL O VARIABLE?

Son símbolos auxiliares que se usan para generar otras cosas. Aparecen durante la derivación, pero deben desaparecer al final. Se reemplazan usando las reglas de producción hasta llegar a una cadena compuesta solo por terminales.

R, S, T, X

Son 4 variables se usan para derivar, pero no deberían aparecer en la cadena final.

Supongamos que usamos esta parte:

S → a T b

T → ε

Derivación:

* Comenzamos con S
* Usamos la regla S → a T b
* Ahora tenemos a T b
* Sustituimos T → ε (vacío)
* Resultado: ab

¿QUÉ GENERA ESTA GRAMÁTICA?

Esta gramática genera cadenas formadas solo por las letras a y b, combinadas de formas específicas:

* El símbolo inicial R permite construir cadenas simétricas (espejadas) usando X R X, o bien producir algo a través de S.
* S genera estructuras como a T b o b T a, lo que aporta un núcleo con un contenido variable entre a y b.
* T produce cadenas intermedias formadas por combinaciones de a y b, incluso puede ser vacío (ε).
* X es simplemente un a o b.

Entonces, el lenguaje L(G) contiene cadenas con una simetría parcial o total, con combinaciones equilibradas entre a y b.

3 CADENAS EN L(G)

R -> S R -> XRX R -> S

-> aT b -> aRb -> bT a

-> aX b -> aSb -> bXT X a

-> aab -> aaT bb -> baX b a

-> aaX bb -> bab b a

-> aaa bb

LA CADENA MíNIMA POSIBLE

R -> S

-> aT b

-> aε b

-> a b

Ahora podemos responder las preguntas una por una si son verdaderas o falsas:

V o F: T ⇒ aba

Falso (eso es una derivación completa, no una directa)

V o F: T ⇒∗ aba

Verdadero (por T → X T X → a ε b → aba)

V o F: T ⇒ T

Falso (no hay una regla T → T)

V o F: T ⇒∗ T

Verdadero (por 0 pasos: T ⇒∗ T)

V o F: XXX ⇒∗ aba

Verdadero (X → a o b, entonces es posible generar aba)

V o F: X ⇒∗ aba

Falso (X solo genera a o b, no cadenas largas)

V o F: T ⇒∗ XX

Verdadero (T puede derivar (en cero o más pasos) a dos X seguidas)

V o F: T ⇒∗ XXX

Verdadero (puede generar 3 letras, como aba)

V o F: S ⇒∗ ε

Falso (S siempre produce al menos dos letras: a.…b o b.…a)

¿QUÉ ES UN ÁRBOL DE DERIVACIÓN?

Un árbol de derivación (también llamado árbol sintáctico) es una representación gráfica que muestra cómo una cadena se genera a partir de una gramática.

Muestra, paso a paso, cómo se usan las reglas de producción desde el símbolo inicial hasta obtener una cadena de solo terminales.

Estructura:

* El nodo raíz es el símbolo inicial de la gramática (por ejemplo, R).
* Cada rama representa una regla de producción que se aplica.
* Las hojas (parte inferior del árbol) son los símbolos terminales que forman la cadena final.
* Las ramas intermedias son variables (no terminales).

R

/ | \

X R X

a S a

/|\

a T b

/|\

X T X

b ε a

Un árbol de derivación te muestra de forma visual cómo una cadena se genera paso a paso usando las reglas de la gramática. Es clave para entender el proceso de derivación y para analizar si una cadena pertenece al lenguaje.

CONCLUSION:

En este análisis de texto vemos como se utilizaron diferentes técnicas, para verlas más detalladas, se va a compartir el link del repositorio para poder ser ejecutado:

<https://github.com/VynCrey/beltran.git>

en este repositorio vamos a encontrar dos carpetas, una con el corpus original y otra con el corpus modificado, ambos llegan al mismo resultado

**Trabajo de Procesamiento de Lenguaje Natural (NLP)**

**Tema:**

**TF-IDF (Term Frequency - Inverse Document Frequency) y Canalización de un Pipeline NLP**

**Alumno:**

**Monzón Cristian**

**Profesora:**

**Scudero Yanina Ximena**

**Materia:**

**TECNICAS PROCESAMIENTO HABLA**

**Carrera:**

**Ciencia de Datos e Inteligencia Artificial**

**Objetivo del trabajo:**

Realizar un pipeline completo de análisis de texto utilizando el corpus **CorpusLenguajes.txt**, aplicando las siguientes técnicas y herramientas de Procesamiento de Lenguaje Natural (NLP):

* Tokenización del corpus
* Eliminación de stopwords en inglés
* Lematización utilizando WordNet y etiquetas gramaticales (POS tagging)
* Cálculo de la matriz TF-IDF con TfidfVectorizer
* Visualización del vocabulario generado
* Extracción de:
  + Las **6 palabras más frecuentes** (jerarquía de frecuencia)
  + **La palabra menos utilizada**
  + **Palabras más repetidas por oración**
  + **Gráfico de distribución de frecuencia de palabras**