Prototipo de asistente corrector gramatical y ortográfico

Alumnos: Aguirre Hernández Leonardo Miguel, Álvarez Hernández Zeth,
Dr. Suarez Castañón Miguel Santiago
Escuela Superior de Cómputo I.P.N. México D.F.
Tel. 57-29-6000 ext. 52000 y 52021. e-mail: leonardomaguirreh@gmail.com, zethalvarezh@hotmail.com

Resumen-- Desarrollo de un prototipo de software, basado en una arquitectura orientada a servicios, en particular la denominada software como servicio o SaaS por sus siglas en inglés. Esta arquitectura es atractiva porque combina el poder de los sistemas orientados a servicios, con el poder y disponibilidad de Internet. El prototipo ha sido definido como: "Prototipo de asistente corrector gramatical y ortográfico en la redacción de protocolos correctos", el cual para fines de presentación será nombrado: Prototipo de asistente corrector gramatical y ortográfico.

Palabras clave - software como un servicio, corrección ortográfica y gramatical automatizada, machine learning, procesamiento de lenguaje natural.

I. Introducción

En la ESCOM, de acuerdo con una encuesta realizada en el mes de septiembre de 2019, aproximadamente el 45% de los alumnos piensan que las unidades de aprendizaje de formación institucional, como lo es Comunicación Oral y Escrita, son materias sin relevancia.

La falta de interés en este tipo de materias ocasiona que el alumno no adquiera adecuadamente los conocimientos necesarios para redactar de forma correcta un documento, lo cual, se ve reflejado en la pobre redacción de los diversos trabajos escritos (tareas, reportes, trámites, etc.) que realiza durante su estancia en la ESCOM.

Una mala redacción tiende a volverse una mala costumbre que afecta al alumno, principalmente al cursar unidades de aprendizaje como Trabajo Terminal, en la que la claridad y ortografía al escribir son de suma importancia.

Lo expuesto, nos motivó a desarrollar un prototipo de software que actúe como asistente para la redacción de protocolos gramatical y ortográficamente correctos. Para esto, utilizaremos técnicas de inteligencia artificial, en particular algoritmos de machine learning enfocados al procesamiento y análisis de textos, mejor conocidos como algoritmos de procesamiento de lenguaje natural (PLN), que nos permitirán detectar, en su mayoría, los errores ortográficos y gramaticales que se presenten. [1]

El software estará disponible como un servicio en la nube, con el objetivo de ponerlo a disposición del mayor número de usuarios posible. Disponer de un asistente en tiempo real, que ayude a pulir la escritura, además de ayudar a escribir correctamente, ayudará a conocer mejor las reglas gramaticales y ortográficas, porque el asistente al aconsejar justificará la sugerencia señalando la o las reglas que se estarían infringiendo, o las que el asistente estaría aplicando.

II. Metodología

El prototipo de asistente corrector gramatical y ortográfico está compuesto por cuatro módulos: módulo de descargas, el cual permite descargar el contenido analizado en formato .pdf o .txt, módulo de correcciones

ortográficas, que se encarga de detectar mayoritariamente los errores de ortografía, módulo de corrección gramatical, que se enfoca en analizar cada oración para determinar si está bien estructurada o no y por último, módulo de sugerencias que, a partir de los errores detectados en los módulos anteriores, muestra sugerencias de lo que el usuario podría hacer para corregir esos errores.

Un factor importante a considerar es la estructura de las oraciones, dado que a partir de esto se podrán identificar sus componentes.

La oración es la unidad máxima del análisis sintáctico. Se caracteriza por los dos rasgos siguientes:

- Es una unidad sintáctica formada por la unión de un predicado y su sujeto. Es decir, la oración constituye el marco sintáctico en el que se establece la relación predicativa.
- Posee necesariamente un verbo; salvo en las oraciones atributivas, este verbo constituye el núcleo del predicado. [3]

El prototipo se realizó basado en la metodología Kanban que, por sus características fue considerada la más adecuada para el desarrollo.

Esta metodología se basa en el desarrollo incremental, dividiendo el trabajo en partes. Normalmente, cada una de esas partes se escribe en una nota y se pega en un tablero. Las notas suelen tener información variada, si bien, aparte de la descripción, debieran tener la estimación de la duración de la tarea. El tablero tiene tantas columnas como estados por los que puede pasar la tarea. [2]

Primero hay que conocer los principios fundamentales:

- Iniciar con lo que haces ahora.
- Estar de acuerdo en seguir un cambio incremental evolutivo.
- Al inicio, respetar los roles actuales, responsabilidades y títulos de trabajo.

Después adoptar las prácticas básicas:

- Visualizar.
- Limitar el trabajo en progreso.
- Gestionar el flujo.
- Realizar explicitas las políticas de proceso.
- Implementar ciclos de retroalimentación.
- Mejorar colaborativamente, evolucionar experimentalmente.

Para el desarrollo se utilizaron los siguientes métodos:

- Tokenización: para separar palabras de signos de puntuación y de espacios durante el análisis del texto ingresado.
- Pilas: para detectar el emparejamiento de los signos que tienen apertura y cierre, como lo son el paréntesis, la doble comilla, el signo de admiración y el de interrogación.
- Anotaciones: para que se pudiera hacer la detección de errores en la estructura de una oración, fue necesario asignar valores manualmente a un aproximado de 4 mil palabras, tales como sujeto, verbo, adjetivo, adverbio, entre otros, y almacenarlas como un diccionario no muy robusto pero lo suficientemente útil para nuestras pruebas.
- Autómatas: en este caso se implementaron 3, de los cuales uno se utilizó para la detección de errores de mayúsculas y minúsculas, como se puede ver en la figura 1, otro para la detección de errores en signos de puntuación y finalmente el más complejo se utilizó para determinar si la estructura de las oraciones era correcta.

Este último, a diferencia de los otros autómatas tuvo un nivel de complejidad mayor, ya que, para poder analizar el contenido, fue necesario utilizar las anotaciones de nuestro diccionario para comparar palabra por palabra y determinar qué tipo de palabra se trataba y así poder utilizar esos valores dentro del autómata en lugar de las palabras directo. Por cuestiones visuales, la figura 2 muestra la tabla de transición de estados de este autómata.

- Algoritmo de búsqueda: aparte del diccionario de anotaciones, se contó con un diccionario general con aproximadamente 670 mil palabras diferentes que nos permitieron indicar en su mayoría, si las palabras ingresadas al prototipo existen o no.

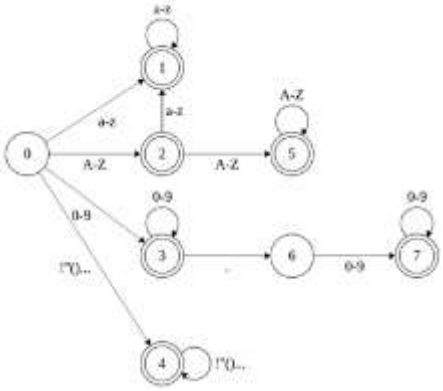

Figura 1. Autómata de detección de errores en mayúsculas y minúsculas.

	A	B	C	D	E	F	G	H
→ 0	1	2	7	12	6	1	11	10
1	-	2	-	-	-	-	-	-
2*	1	2	3	5	-	4	9	-
3*	1	2	3	5	-	-	9	-
4*	1	2	3	5	-	4	-	-
5*	1	2	-	5	-	4	9	-
6	1	2	7	-	-	-	8	2
7*	1	2	7	4	-	-	-	10
8	-	-	7	-	-	-	-	-
9*	-	-	3	-	-	-	-	-
10	1	2	7	-	-	-	8	-
11	-	-	7	-	-	-	-	-
12	1	2	-	12	-	-	-	-

Figura 2. Tabla de transición de estados para detección de orden en las oraciones.

III. Resultados

Se realizaron pruebas sobre oraciones y párrafos de diversos protocolos de la Escuela Superior de Computo. El sistema mostro realizar correcciones sobre símbolos de puntuación, uso de mayúsculas y minúsculas, palabras mal escritas y estructuras gramaticales de forma correcta en la mayoría de los casos. Si bien se depende mucho de un corpus amplio de palabras etiquetadas, damos la posibilidad de que en futuros trabajos este pueda crecer.

Al momento de realizar pruebas de integración se detectaron algunas limitantes.

- Cuando los algoritmos de detección de errores envían la posición del error, en la ventana (front-end) se muestran las palabras de dicho error con un color distinto, sin embargo, para que esto fuera posible se tenían que agregar estilos. Al agregar estos estilos por cada algoritmo, el prototipo no identificaba si eran parte del texto, por lo que, al intentar detectar otro error, tomaba en cuenta los estilos añadidos.

  - Para solucionar esto, se especificó que se debían descartar los estilos del texto al enviarlo a los algoritmos, de tal forma que cuando un texto tuviera diferentes tipos de error, no hubiera problema ni para marcar de otro color las palabras, ni para identificar el texto limpio.

- Al presionar la tecla espacio en el cuadro de texto, todos los espacios que

contenía el texto ingresado eran eliminados.

- Para evitar esto, en los algoritmos no solo se consideraron las palabras, sino también los espacios.

- Al ingresar texto, se encontraron algunos caracteres que no se reconocían.

- Se indicó que el texto ingresado debía tener la codificación UTF-8.

IV. Conclusiones

Éste trabajo resultó más complejo de lo que esperábamos, pues con lo desarrollado hasta este punto, pudimos notar que no es una herramienta cien por ciento precisa, dado que únicamente corrige ciertas reglas de la ortografía y de la gramática del español de México.

Debido al léxico variado del idioma español, utilizado en México, un sistema que realice correcciones procurando que se cumplan todas las reglas ortográficas y de gramática resulta difícil de lograr por la cantidad de reglas y el tiempo limitado que tenemos, debido a esto la problemática que planteamos solo contempla las reglas ortográficas más sencillas, así como una limitada variedad de reglas gramaticales. Sin embargo, abre paso a la posibilidad de perfeccionarlo como trabajo a futuro.

Queda como un trabajo que, a futuro, podría implementarse no únicamente para determinadas personas, sino que podría llevarse incluso a un nivel más general, permitiendo no solo la corrección de protocolos, sino también de otros textos variados.

RECONOCIMIENTOS

Los Autores agradecen a la Escuela Superior de Cómputo del Instituto Politécnico Nacional por el apoyo recibido y las facilidades otorgadas para el desarrollo del presente trabajo terminal.

REFERENCIAS

[1] A. Kulkarni and A. Shivananda, Natural Language Processing Recipes. Berkeley, CA: Apress, 2019, p. 2.

[3] Induráin, J.. (2011). *La oración*. En Sintaxis, lengua española (pp. 45-48). Mallorca, 45 - 08029 Barcelona: LAROUSSE.

[2] Executive master in project management, Universidad de Alcalá, "Gestión ágil de proyectos con kanban", 2014 [En línea]. Disponible en: http://www.uv-mdap.com/programa-desarrollado/bloque-iv-metodologias-agiles/gestion-agil-de-proyectos-con-kanban/ [Accedido: 09 - Julio - 2019].