Sistema de texto predictivo e indexación de nombres por sonido

Alex Montoya Franco José Orlando Rengifo Caicedo

Universidad Eafit Universidad Eafit

CR 55 #95 A15 4to Medellín Colombia Cll 65 #90-90 Apto: 25-08 Medellín-Colombia

+57 3226595682 +57 3197319112

[amonto69@eafit.edu.co](mailto:amonto69@eafit.edu.co) [jrengif9@eafit.edu.co](mailto:jrengif9@eafit.edu.co)

**ABSTRACT**

In this paper some algorithms and data structures will work to create an efficient solution to the following problem: stored in a data structure a large set of English words and implement algorithms and selected structures by problems investigated and discussed in this work, a program by entering a word returns words that are prefixed by the word entered, and further that given a word entered return other words with a phonetic English Orthoepy like. In this way it will work in this paper search algorithms and data structures that fit in an efficient manner claimed by this problem.

**Resumen**

En este paper se trabajarán algunos algoritmos y estructuras de datos para crear una solución eficiente al siguiente problema: almacenar en una estructura de datos un conjunto grande de palabras en Inglés e implementar con los algoritmos y estructuras escogidas mediante problemas investigados y mencionados en este trabajo, un programa que al ingresar una palabra retorne palabras que tengan como prefijo la palabra ingresada, y además que dado una palabra ingresada retorne otras palabras con una fonética ortoepía inglesa similar. De esta manera se trabajará en este paper en la búsqueda de algoritmos y estructuras de datos que se acoplen de una manera eficiente a lo demando por este problema.

**keywords**

algoritmos, estructuras de datos, indexación de nombres por sonido, algoritmos de fonética ortoepía inglesa.

**Introducción**

Este trabajo plantea unas estructuras de datos y algoritmos para lo que en síntesis es un autocompletador de texto del idioma inglés, pero antes de plantear la solución realizamos una investigación sobre problemas similares reconociendo así los algoritmos que suelen usarse para resolver este tipo de problemas y poder tener una mejor idea de cómo proceder con la búsqueda e implementación de algoritmos idóneos para la solución del problema.

**Problemas Similares**

Una vez se tiene dominio del problema la búsqueda del algoritmo idóneo y la estructura ideal que pueda almacenar los datos que se usaran para la solución del problema y que se acople bien al algoritmo escogido es crucial, dado que para cada tipo de problema hay un algoritmo que se acopla a sus necesidades y puede desempeñarse mejor. Por tal motivo, hemos investigado algunos algoritmos que se usaron para la solución de cierto tipo de problemas, algunos con un grado de complejidad mayor pero que nos pueden dar una idea de cuales se pueden desempañar mejor para el programa esperado. Estos son algunos de los problemas investigados.

**1-** **Algoritmos y Métodos para el Reconocimiento de Voz en Español Mediante Sílabas**

Aunque este problema se trata de un reconocimiento de voz necesita mediante un algoritmo decidir mediante la frecuencia del habla del locutor a que palabra se asemeja en el idioma e interpretarla, este problema es bastante complejo y necesita de conocimiento previos como lógica difusa y redes neuronales artificiales que son temas bastante profundos, pero para la solución de este problema los investigadores del *Centro de Investigación en Computación-IPN* decidieron utilizar como TDA(tipo de dato abstracto) una lista enlazada “*Las cuales nos permiten ir accediendo a cada uno de los elementos de la palabra para posteriormente realizar la segmentación adecuada de la misma según las reglas silábicas del idioma.”.* Una solución bastante brillante para un problema tan complejo.

**2- ALGORITMO FONÉTICO PARA DETECCIÓN DE CADENAS DE TEXTO DUPLICADAS EN EL IDIOMA ESPAÑOL**

*PhoneticSpanish* así nombraron los investigadores el algoritmo para la detección de cadenas de texto en el idioma español el cual considera la presencia de errores ortográficos en el idioma español. Algoritmo pensado para organizaciones que manejan grandes volúmenes de datos como hospitales que tienen que registrar a un paciente una y otra vez pero que por error pueden enviar un registro incorrecto. Para la realización de este algoritmo los investigadores se basaron en un algoritmo inventado por Robert C. Russell y Margaret Rey llamado Soundex el cual es un algoritmo fonético para la indexación de los nombres de sonido, como se pronuncia en Inglés, el objetivo de Soundex es que homófonos que se codifican para la misma representación de manera que se pueden emparejar a pesar de diferencias menores en la ortografía. Este algoritmo bastante conocido y creado en la década de los 30 fue crucial para la investigación de este problema y dio cabida a su solución.

**3- Un corpus de bígamas utilizado como corrector ortográfico y gramatical destinado a hablantes nativos de español**

Se trata de un algoritmo de corrección ortográfica pensado para hablantes nativos del idioma español, el algoritmo detecta y corrige errores gramaticales que usan normalmente los nativos de la lengua española el algoritmo se encarga de analizar la frecuencia en el corpus de cada una de las palabras utilizadas en el texto y, por otra parte, de cada bigrama. Además se calcula el número de veces que cada combinación de dos palabras aparecería en el corpus si estas se dieran de manera aleatoria. Es una solución bastante interesante dado que usa estadística para la corrección de errores ortográficos.

**4- Un algoritmo lingüístico-estadístico para resumen automático de textos especializados**

Se trata de un algoritmo que resume textos de gran volumen, pero enfocado a tipos de texto médicos, se apoya en la teoría estudiada en los años 70 sobre técnicas estadísticas y las técnicas lingüísticas, y se basa en que tanto una palabra se encuentra en un texto(frecuencia) y de ahí se procede a eliminar, reacomodar el texto para que quede legible y entendible o reemplazar por otra que cumpla igual funcion.

Todos Estos problemas usaron modelos matemáticos para la solución de su problema en específico y trabajaban con algoritmos sobre la frecuencia en la que una palabra se encuentra en un texto, esto nos da pie para poder a empezar a modelar soluciones pertinentes a nuestro problema.

**Estructura de datos usada para la solución del problema**

La estructura principal sobre la que trabaja el algoritmo es un grafo, árboles y tablas hash, las tablas hash permiten recibir un carácter que es validado por el algoritmo como el key o la “llave” y que retorna un value o “valor” que contiene su representación en el árbol, es decir un nodo, en esta parte están siendo procesados por un grafo, que contiene los caracteres iniciales o “padres” los cuales son a la vez la raíz de diferentes árboles que son creados cuando dicho carácter es el prefijo de una palabra que se procesa.

**Algoritmos Eficientes**

El algoritmo seleccionado para la parte del autocompletado es un algoritmo de búsqueda en preorden-profundidad haciendo uso de recursividad que se va hasta el último de los nodos hijos y regresa a su nodo padre si este tiene más hijos, es un algoritmo muy eficiente ya que su complejidad en el peor, mejor y caso promedio es de Orden(1), el método se llama a si mismo con el valor de otro nodo lo que mejora el tiempo de ejecución

El algoritmo para obtener las palabras del diccionario, lee cada palabra y la añade a un árbol, la ventaja frente a otros algoritmos es que este ubica cada carácter en la posición que le corresponde en el árbol, entonces a la hora de buscar solo se recorre dicho árbol

Para el algoritmo de fonética se usó el algoritmo distancia de levenshtein

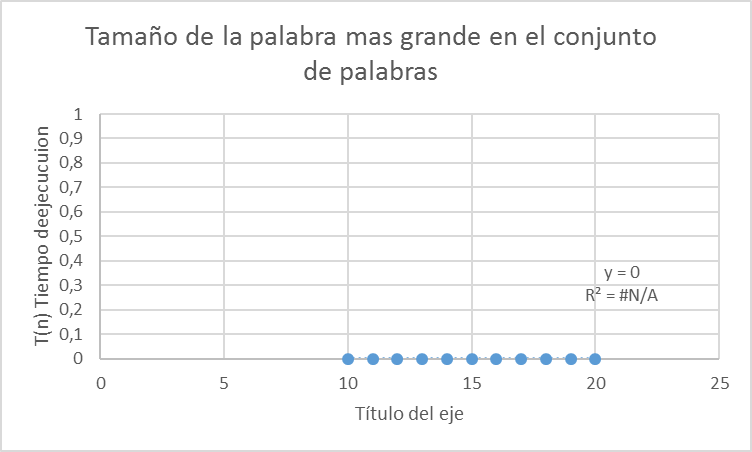
**Complejidad**

La complejidad el algoritmo en el peor de los casos es O(1) ya que siempre realiza el mismo número de operaciones independiente de la cantidad de palabras que tenga el diccionario y de la cantidad de sugerencias que entregue el sistema

longitud máxima de una palabra en el conjunto de palabras (n)

longitud de la palabra que recibe la consulta (m)

numero de palabras del diccionario (p).



**Trabajo del problema**

El trabajo en equipo fue productivo, trabajamos con el repositorio subversión que nos permitió mantener un control de versiones para retroceder en el código de ser necesario y trabajar en paralelo sin afectar el desarrollo del proyecto.

Fue un trabajo en equipo diversificad, en ocasiones hicimos uso de la práctica de desarrollo ágil " programación en parejas", siendo uno el conductor y otro el revisor, en otras ocasiones nos repartíamos el trabajo y programábamos los dos y fusionábamos las versiones en el repositorio

# **Referencias**

|  |  |
| --- | --- |
| [1] | Jose, L.O..R. &. Sergio,.S. , Algoritmos y Métodos para el Reconocimiento de Voz en Español Mediante, Centro de Investigación en Computación-IPN, Mexico, 2006. |
| [2] | Ivan,F. & Francisco,M. ALGORITMO FONÉTICO PARA DETECCIÓN DE CADENAS DE TEXTO DUPLICADAS EN EL IDIOMA ESPAÑOL, Revista Ingenierías Universidad de Medellín, Medellín, 2012. |
| [3] | S. Alicia, Un corpus de bigramas utilizado como corrector ortográfico y gramatical destinado a hablantes nativos de español, Universidad Nacional de Educación a Distancia, 2015. |
| [4] | Iria,C. & Juan,M.T.M. Un algoritmo linguíıstico-estadístico para resumen automático de textos especializados, Linguamática, Portugal, 2009. |