ZoomBooks Smart: Sistema Digital para la Recuperación de Información relevante a escuelas de nivel medio superior

ZoomBooks Smart: Digital System for information retrieval relevant to upper-middle schools

Cupertino Lucero-Álvarez[1], Mariano Larios-Gómez[2], Pascual Pérez-Cruz[1], Carlos Ortiz-Ramírez[1], Brian Manuel González-Contreras[3], Juventino Montiel-Hernández[3]

[1]Universidad Tecnológica de Izúcar de Matamoros, Departamento de Tecnologías de la Información
[2]Benemérita Universidad Autónoma de Puebla, Facultad en Ciencias de la Computación
[3]Universidad Autónoma de Tlaxcala, Facultad de Ciencias Básicas, Ingeniería y Tecnología

* Correo-e: clucero@utim.edu.mx

PALABRAS CLAVE:
Modelo booleano extendido, recuperación de información, aprendizaje automático, máquina de búsqueda.

RESUMEN

En este artículo se presenta una propuesta para la construcción del prototipo de software ZoomBooks Smart, basado en un Sistema de Recuperación de Información orientado a sin-conexión (SRI- off line) para el almacenamiento y recuperación de documentos relevantes a la carga curricular de los estudiantes de nivel medio superior, especialmente para los bachilleratos apartados de las tres Mixtecas de la República Mexicana que carecen de recursos de información digitales. La metodología planteada para su implementación tiene como base el modelo Booleano Extendido (BE) para la recuperación, además de aplicar aprendizaje automático para recomendar lecturas a los usuarios con base en sus perfiles, y técnicas del Procesamiento del Lenguaje Natural (PLN) para el tratado automático de tareas específicas como las sugerencias de frases mediante n-gramas y la expansión de consultas. Se presenta un primer prototipo de software que tiene la funcionalidad de almacenar, de manera semiautomática los documentos en formato PDF (formato de documento portátil) que son relevantes a las diferentes asignaturas de la carga curricular de los estudiantes u otros usuarios en el ambiente académico. Los documentos pueden ser recuperados en orden de relevancia consulta-documento, mediante consultas de texto libre.

KEYWORDS:

Extended boolean model, information retrieval, machine learning, search engine.

ABSTRACT

This article presents a proposal for the construction of the ZoomBooks-Smart framework prototype, based on an offline-oriented Information Recovery System (SRI-offline) for the storage and retrieval of documents relevant to the curricular load of students upper level, especially for high school graduates of the three Mixtecas in the state of Puebla-Mexico that lack digital information resources. The methodology proposed for its implementation is based on the Extended Boolean (BE) model for recovery, in addition to applying machine learning to recommend readings to users based on their profiles, and Natural Language Processing (PLN) techniques for the treaty Automatic specific tasks such as n-gram phrase suggestions and query expansion. As a result, the system has the functionality to store semi-automatically, documents in PDF format (portable document format) that are relevant to the different subjects of the curriculum load of students or other users in the academic environment, and of this from can be retrieved in order of relevance query-document, through free text queries.

• **Recibido:** 10 de noviembre 2019 • **Aceptado:** 8 de marzo de 2020 • **Publicado en línea:** 30 de junio de 2020

1. Introducción

En la actualidad, gracias a los avances en las Tecnologías de la Información (TI), existe una gran variedad de herramientas digitales que facilitan la búsqueda de información en casi todas las áreas del conocimiento humano. Tal es el caso de las Bibliotecas digitales, tecnologías Wiki, máquinas de búsqueda como Google, Yahoo, Bing entre otras, en [1] se puede encontrar más información de la situación de los buscadores WEB. Si se cuenta con acceso al Internet mediante los motores de búsqueda, se puede consultar la información requerida, y el buscador mostrará una lista de páginas en orden de relevancia consulta-documento. Sin embargo, en muchas regiones apartadas de nuestro país como ocurre en las tres Mixtecas (Puebla, Guerrero y Oaxaca), hay muchas comunidades que no cuentan con acceso al Internet o éste es ineficiente, y peor aún, las escuelas marginadas de estas regiones no brindan las herramientas y recursos de información necesarios para que la comunidad estudiantil pueda usar para hacer investigaciones documentales y realizar tareas, lo cual va en detrimento del proceso de enseñanza-aprendizaje. Para tratar de solucionar esta problemática, se pudieran usar recursos de información documentales offline como Wikipedia (Mediante XOWA o kiwix se puede descargar la biblioteca completa para español) o incluso alguna biblioteca digital, pero al ser recursos de propósito general, no contienen la información específica de la bibliografía recomendada en el plan curricular de las escuelas, y en el caso de las bibliotecas digitales la mayoría no son de acceso gratuito ni de acceso off-line, en [2] se presenta un estudio del uso de bibliotecas digitales para la educación básica en el que se pone en evidencia la falta de especialización en el lenguaje y en los contenidos. Es por eso que desarrollar un software de recuperación de información, con una base de datos que contenga documentos relevantes al plan curricular de las escuelas de nivel medio superior es una tarea importante a considerar, y de esta forma apoyar a los estudiantes en la búsqueda de la información. En [3] se plantean metodologías para el desarrollo de bibliotecas digitales para tesis doctorales que pudieran ser de interés para el desarrollo de máquinas de búsqueda para otros dominios.

El objetivo es construir un SRI off-line para el almacenamiento y búsqueda de documentos en formato PDF, con base en los planes de estudio de las materias de nivel bachillerato. Como una meta subyacente al primer objetivo se planteó utilizar el modelo Booleano extendido propuesto en [4]. Para la sugerencia de frases en la caja de búsqueda se plantea el uso de la técnica de n-gramas. Se usará un *thesaurus* de información para hacer expansión de consultas por medio de palabras en relación semántica. Otra forma de llevar a cabo esta tarea es el uso de ontologías para dominios específicos, [5]. En este trabajo se plantea el diseño e implementación de un algoritmo de aprendizaje automático como núcleo de un subsistema de recomendación basado en el conocimiento, para sugerir documentos alternos a la salida del SRI que pudieran ser de interés para el usuario con base

en su perfil. El perfil de usuario será construido tomando en cuenta la actividad en su historial de búsqueda y los contenidos, así como en la identificación automática de patrones. Otro objetivo es que el sistema de cómputo sea instalable para ser usado de manera local en computadoras personales. Cabe hacer notar, que el sistema que aquí se propone está pensado para ser usado en los bachilleratos de las comunidades marginadas de la Mixteca, en los cuales no se cuenta con acceso al Internet, por lo que no existen las condiciones para un enfoque en la nube.

2. Metodología para el desarrollo del SRI

Un Sistema de Recuperación de Información tiene como funciones básicas el almacenamiento y recuperación de documentos relevantes a la consulta de los usuarios, la relevancia de un documento a una consulta tiene que ver con el contenido del documento y la necesidad de información del usuario expresada en los términos de la consulta, por ejemplo, en un contexto financiero, si la consulta del usuario fuera “la privatización de la banca en México”, los documentos que hablan sobre “la banca” como mueble no son relevantes. Por otro lado, la efectividad de la recuperación depende de la precisión (número de todos los documentos recuperados que son relevantes) y el recall (proporción de documentos relevantes recuperados del total de documentos relevantes de la colección). En [6] se hace una evaluación de los SRI clásicos mediante el cálculo de la precisión y el recall, para poder estimar su desempeño en dominios de aplicación específicos. La metodología que se plantea para el desarrollo del SRI que aquí se presenta pasa por varias etapas: preparación de los textos de los documentos que se van a almacenar en la base de datos; representación de la información en memoria y recuperación; procesamiento de las consultas, algoritmos y su justificación.

En la Fig. 1 se muestra, de manera general, una aproximación de la estructura del motor de recuperación de información que aquí se propone. Los módulos representados por rectángulos representan bloques de tareas conectados de manera secuencial. Así, por ejemplo, en el módulo "Consulta" se engloban tareas como la sugerencia de frases y mantenimiento del historial. En el módulo "Normalización" se engloban tareas como stemming y reducción de términos. Como puede observarse en las flechas, posteriormente se realiza la extracción de los tokens y se procesan mediante la unión o la intersección de los postings. En el centro se puede apreciar un mecanismo que representa a los algoritmos encargados de la optimización de la consulta. El resultado de los algoritmos se usa para consultar por los documentos en la base de datos y generar la respuesta.

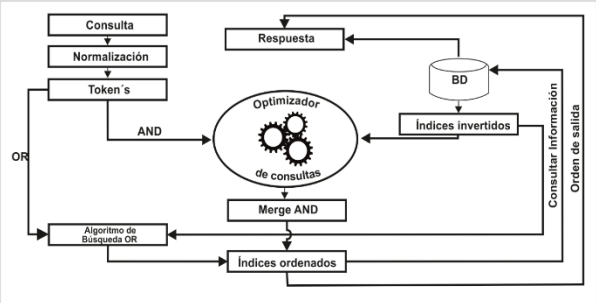


Fig. 1 Motor de Recuperación de Información

En el centro se puede apreciar un mecanismo que representa a los algoritmos encargados de la optimización de la consulta. El resultado de los algoritmos se usa para consultar por los documentos en la base de datos y generar la respuesta.

2.1. La normalización del texto

Mediante lematización o *stemming* se conduce a mejorar los SRI, en particular la cobertura (recall), característica que tiene que ver con el número de documentos relevantes recuperados en una consulta [7]. El *stemming* y la lematización son técnicas ligeramente distintas ya que mientras que en la lematización se hace uso de diccionarios para tratar de eliminar las diferentes flexiones de las palabras mediante análisis morfológicos y sintácticos, en los procesos basados en *stemming* se busca, mediante listas de prefijos y sufijos, encontrar la raíz morfológica. Un algoritmo clásico para la eliminación de sufijos se presenta en [8], un ejemplo de su aplicación, para normalizar documentos recuperados en la WEB, puede encontrarse en [9]. Dado que los algoritmos de *stemming* han sido ampliamente usados en los SRI, en [10] se presentan varios experimentos para medir su efectividad para el español, en su investigación comparan algoritmos basados en *n-gramas*, *s-stemmer* mejorado (algoritmo mediante el cual se eliminan las terminaciones en *s* y *es* para tratar de reducir el plural al singular, y las vocales *a*, *e* y *o* para tratar de sortear el género), también se compara la lematización flexiva y la derivativa. Se concluye que los algoritmos más complejos, que incluyen conocimiento lingüístico no consiguieron mejorar los resultados del *s-stemmer* mejorado que es más fácil de implementar.

En este trabajo, para normalizar los textos de los documentos (entradas del Thesaurus, consultas y textos en general) se plantea el uso de un algoritmo *s-stemmer* mejorado, así como también la eliminación de las palabras que aportan poco valor informativo como artículos, conectores lógicos, números, símbolos especiales, etc. Este proceso de normalización de los textos también se usará en las consultas y en los términos del Thesaurus.

2.2. Proceso de recuperación de información

Las técnicas para la recuperación de información se pueden clasificar en los modelos: clásicos (booleano, espacio vectorial y probabilístico), alternativos (basados en la lógica Fuzzy), lógicos (basados en la lógica formal), basados en la interactividad (se retroalimentan por medio de la relevancia de los documentos recuperados), y basados en la inteligencia artificial (Redes neuronales, algoritmos genéticos, y PLN), una descripción más

detallada de algunos de ellos se puede encontrar en los trabajos de [11] y [1].

Dada la naturaleza de la información que se pretende representar y recuperar en esta aplicación específica, se tuvo que discernir entre los modelos clásicos. En el modelo vectorial, para calcular la relevancia entre consulta y documento, cada documento y cada consulta son representados en vectores, donde los índices de los vectores son todos los términos de la colección, y para cada termino se le asigna un peso con base en estadística básica, para de esta forma comparar los vectores de representación y establecer el grado de cercanía mediante la medida del coseno del ángulo [12]. El modelo booleano fue el modelo más utilizado en los primeros buscadores bibliográficos debido a que se basa en la teoría de conjuntos y en el álgebra booleana, donde las búsquedas de los términos de la consulta en los documentos son binarias, es decir solo hay dos estados posibles: el documento contiene al término o el documento no contiene al término, de esta manera, es posible usar los operadores booleanos básicos (*AND, OR y NOT*) en la expresión que representa a la consulta, y así incluir condiciones sencillas de búsqueda, en [4] se plantea el uso del modelo Booleano extendido con una propuesta de representación de los documentos que facilita las búsquedas. Por otro lado, Los modelos probabilísticos usan técnicas de probabilidad como las búsquedas bayesianas, y probabilidad condicional e independencia para calcular la probabilidad de que los términos de las consultas se encuentren en la colección de los documentos. En [13] se propuso el uso del primer modelo probabilístico con fines de la RI.

En SRI locales, son los modelos probabilísticos los que han mostrado ligeramente mejores resultados que los modelos: vectorial y booleano, sin embargo, lograr esa ligera ventaja requiere de mayor complejidad en su implementación. Por otro lado, no hay la suficiente evidencia de que el modelo vectorial arroje mejores resultados que el modelo booleano extendido [4]. En este proyecto, en el que la colección de documentos es relativamente pequeña, y debido también a el poder de computo de las maquinas modernas, aunado a la sencillez y efectividad del modelo booleano extendido, se ha elegido a este como la parte central de la metodología.

3. Estrategia de almacenamiento

Se plantea construir una base de datos en la que se almacenen todos los documentos, tanto en formato PDF como en texto plano. También se almacenarán, el diccionario de términos y las listas invertidas. En un archivo de texto plano se almacenará el *thesaurus* de palabras en relación semántica, para su construcción se usará el método de vecinos cercanos para dominios específicos, propuesto en [14]. Por último, en una BD se almacenarán los perfiles de los usuarios identificados mediante su matrícula o alguna clave especial.

3.1. Construcción de la estructura de representación

Para implementar el modelo booleano extendido es necesario crear una estructura de datos tipo hash para almacenar la representación. El método básicamente consiste en dos etapas: construcción del diccionario de términos (*tokens*) y creación de los *posting lists* o simplemente *postings*. En la Fig. 1. Se pueden observar las dos partes fundamentales de una estructura de índice invertido (*inverted index*). Como puede observarse en la estructura de índice invertido, cada término o *token* es la llave de acceso a su respectiva lista o valor, la que a su vez representa a todos los documentos de la colección que contienen al *token* actual.

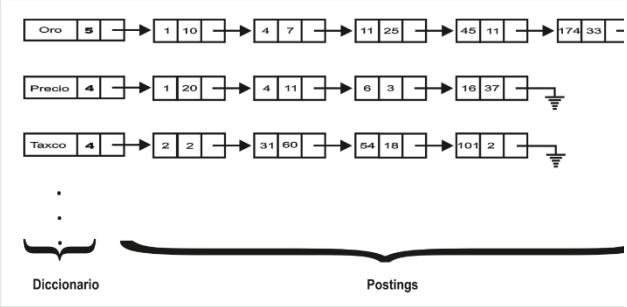


Fig. 2. Índice invertido

En la Fig. 2, del lado izquierdo del índice invertido se tiene el diccionario de términos ordenados de manera alfabética, y del lado derecho sus *posting lists*, los cuales mantienen, en orden ascendente, los *IDs* de los documentos que contienen al *token* correspondiente en el diccionario de términos, y la frecuencia del término en cada documento. El número que aparece en cada nodo del diccionario representa al número de documentos que contienen al *token*.

Algorithm 1 Algoritmo para generar el índice invertido

```
0: procedure GINDICEINV(DocIDs, InvertedIndex)
1: for (i ← 0; Length(docIDs)) do
2:    TokenStream ← GET_DOC(docIDs[i])
3:    while EXISTS_TOKENS(TokenStream) do
4:       TokenActual ← GET_ACTUAL(TokenStream)
5:       DROP_TOKEN(TokenStrem, TokenActual)
6:       if TokenActual ∈ InvertedIndex.keys then
7:          PostingActual ← InvertedIndex[TokenActual]
8:          ADD_DOC(PostingActual.docIDs[i])
9:       else
10:         ADD_TOKEN(InvertedIndex, TokenActual)
11:         ADD_POSTINGS(InvertedIndex, docIDs[i])
12:      end if
13:   end while
14:   REPRESENTED(docIDs[i])
15: end for
16: SORT_TOKENS(InvertedIndex)
17: return InvertedIndex
end procedure
```

El proceso para la creación del índice invertido se plantea de la siguiente forma:

1. Carga de los documentos de la colección.
2. Creación de la lista de *tokens*.
3. Creación de las listas invertidas.

Se propone iniciar con la carga la colección de los documentos en la Base de Datos asignando a cada documento un número entero consecutivo en orden de llegada. Posteriormente se genera la lista de *tokens* de todos los documentos y se almacena en una tabla. Por último, se ligan los *tokens* con los documentos que los contienen y se registra su frecuencia en los nodos de la lista invertida correspondientes a cada documento, también se registra el número de documentos que contienen al *token* en cada nodo del diccionario.

Para las actualizaciones, es necesario saber que documentos ya se han representado anteriormente y cuáles no, esto se debe hacer

para evitar construir desde cero el índice invertido cada vez que la colección de documentos cambie.

El algoritmo 1, se utiliza para la construcción del índice invertido, que como puede observarse necesita como parámetros de entrada: las claves de los documentos a ser representados y el índice invertido (puede ser nulo).

Algorithm 2 Algoritmo para la intersección de los postings

```
0: procedure MERGEAND(p1, p2)
1: apLista ← NIL
2: while (p1 ≠ NILL and p2 ≠ NILL) do
3:    if (docID(p1) == docID(p2)) then
4:       ADD(apLista, docID(p1))
5:       p1 ← next(p1)
6:       p2 ← next(p2)
7:    else if (docID(p1) < docID(p2)) then
8:       p1 ← next(p1)
9:    else
10:      p2 ← next(p2);
11:   end if
12: end while
13: return apLista
end procedure
```

Una vez que se tienen las claves de los documentos no representados (*docIDs*) y el índice invertido, se procesa documento a documento para obtener los *tokens* y verificar si se encuentran dentro de la representación (líneas 1, 2,3). Una vez procesado un *token* se elimina del flujo de *tokens* de cada documento (líneas 4 y 5). Si el *token* actual ya se encuentra dentro de las entradas del índice invertido, entonces se obtiene el respectivo *posting list* y se le agrega al final el *docID* del documento actual (líneas 6,7 y 8). Por ejemplo, en la estructura de índice invertido de la Fig. 2, si *TokenActual* fuese la palabra *Oro* y el *docID* del documento actual fuera el número 200, entonces al final del *posting lis*t del *token Oro* se le anexaría el *docID*: 200. Por otro lado, en caso de tratarse de un nuevo *token*, entonces se agregaría el *TokenActual* como una nueva entrada del índice invertido y un nuevo *posting list* con el *docID* actual como su primer valor (líneas 9,10 y 11). Una vez que el flujo de entrada de cada documento se termina de procesar, se cambia el valor del atributo representado a true para el *docID* actual en la BD (línea 14). Por último, una vez terminada la representación se ordena alfabéticamente el diccionario para prevenir futuros procesamientos y se retorna la estructura para su almacenamiento (línea 16 y 17).

Algorithm 3 Algoritmo para la uniòn de los postings

```
0: procedure MERGEOR(p1, p2)
1: apLista ← NIL
2: if (p1 = NILL and p2 ≠ NILL) then
3:    apLista ← p2
4: else if (p1 ≠ NILL and p2 = NILL) then
5:    apLista ← p1
6: end if
7: while (p1 ≠ NILL and p2 ≠ NILL) do
8:    if (docID(p1) == docID(p2)) then
9:       ADD(apLista, docID(p1))
10:      p1 ← next(p1)
11:      p2(p2)
12:   else if (docID(p1) < docID(p2)) then
13:      ADD(apLista, docID(p1))
14:      p1 ← next(p1)
15:   else
16:      ADD(apLista, docID(p2))
17:      p2(p2)
18:   end if
19: end while
20: return apLista
end procedure
```

3.2. Procesamiento de la consulta

Para procesar los términos de la consulta, después del uso del *Thesaurus* para la expansión automática, se usará un algoritmo de reunión de los *postings* que tengan como llave a los términos de la consulta usando los operadores booleanos AND y OR de manera diferenciada. En el algoritmo 2 se muestra el mecanismo para la intersección de los *postings*

(operador booleano *AND*) el cual se propuso en [4], y en el algoritmo 3 se muestra como se hace la unión de los *postings* (operador booleano *OR*).

El algoritmo para la intersección de los *postings* consiste en tomar por pares los *postings* de los términos de la consulta y buscar los documentos que contengan a ambos, se genera un *posting* para la respuesta el cual se usa como nueva entrada del algoritmo para compararlo con el correspondiente *posting* del siguiente término de la consulta; se usa el número de documentos que contienen a cada *token* para procesar los *postings* en orden ascendente con respecto de su tamaño (optimizador de consultas), esto reduce el número de comparaciones en caso de que un término de la consulta no se encuentre en la representación (este proceso continua siempre y cuando no se encuentre un término que no esté representado, en caso contrario no habrá respuesta). Por ejemplo, si la consulta del usuario fuera "precio del oro en Taxco"; los *tokens* serían (precio, oro y taxco). El SRI haría la intersección y la unión de los *postings* correspondientes a los *tokens* de la consulta. Por medio de la intersección se obtienen las respuestas más relevantes (clasificación A), y por medio de la unión las menos relevantes (clasificación B). Para obtener la intersección se utiliza el algoritmo *MERGEAND(p1,p2),* en el cual, los apuntadores de los *postings p1 y p2* se recorren buscando los documentos que contienen todos los términos de la consulta. Así, para el ejemplo, primero se ejecutaría *MERGEAND(p1,p2)*, donde *p1* apunta al *posting list* del *token* "precio" y *p2* apunta al *posting list* del *token* "Taxco", el resultado se asigna a *apRespAnd* (apuntador a la respuesta), posteriormente se ejecuta *MERGEAND(apRespAnd, p2)*, esto es: se intersecta el resultado de la primer ejecución con el *posting* del *token "*oro". El resultado del algoritmo *MERGEAND(p1,p2)* sería el conjunto de *docIDs* = {2}, posteriormente, el elemento *docID=2* se busca en el *posting* del *token "oro"*, y como se encuentra, la salida final del algoritmo seria el conjunto de *docIDs* = {2}.

Un algoritmo similar se usaría para la unión de los *postings*, para lo cual se propone que los documentos seleccionados para la salida sean los que contengan la mayoría de los términos de la consulta y de la consulta expandida.

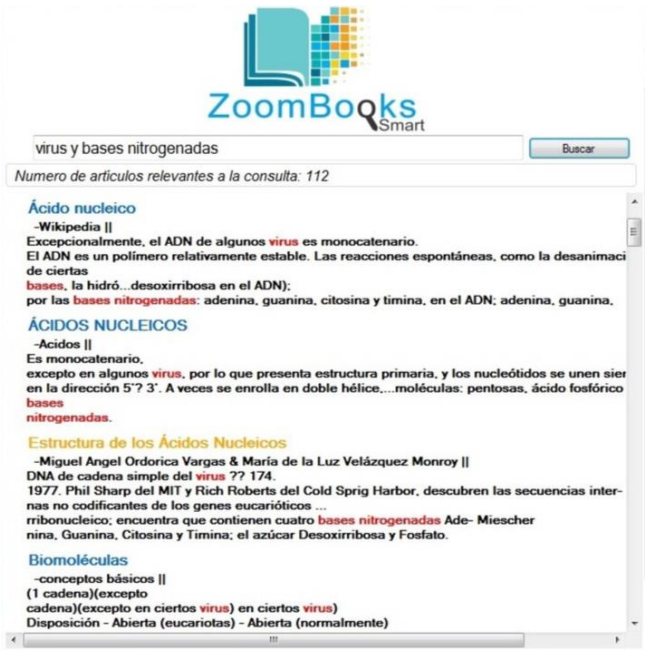
En el algoritmo 3 se muestra la unión de los *postings*. Se inicia con un apuntador hacia la lista de resultados la cual va a contener las claves de todos los documentos que contienen a cualquiera de los términos de la consulta (líneas 1,2,3,4 y 5). Si las claves de los documentos de los *postings* actuales son iguales, entonces se agrega la clave a la lista de respuestas como se muestra en las líneas: 7,8,9,10 y 11, y se incrementan los apuntadores. En caso contrario se agrega a la lista de salida el ID menor y se incrementan los punteros (líneas 12,13,14,15,16 y 17). Por último se regresa la lista de la respuesta para ser usada como la salida de respuestas de la clasificación B.

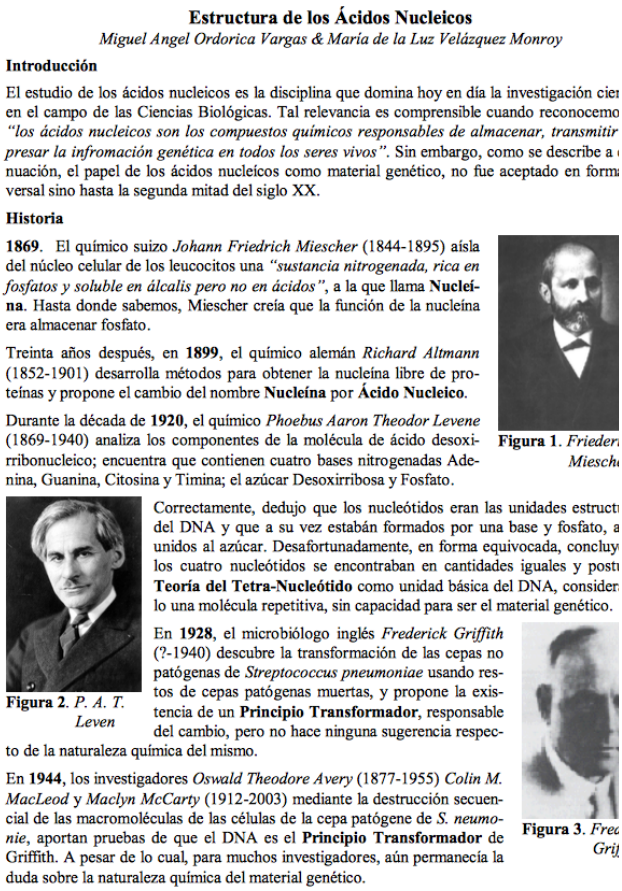
3.3. Sugerencias en la búsqueda

Para las consultas se plantea el uso de n-gramas (especialmente trigramas) para sugerir frases en el momento de la búsqueda. Para lograr esta tarea, también se propone el uso de un algoritmo de aprendizaje automático no supervisado con base en el perfil de usuario y el historial de búsquedas exitosas. También con base en los perfiles y la detección automática de patrones se podrán sugerir obras que pudieran ser de interés para el usuario aunque no se encuentren en el historial, ni se haya consultado por ellas. Un sistema recomendador basado en el contenido será implementado para realizar esta tarea.

4. Resultados y Discusión final

Con base en la metodología planteada, se desarrolló una versión preliminar del sistema ZoomBooks Smart la cual contiene una base de datos de 700 documentos en formato PDF de diferentes áreas del conocimiento, el peso total es de 121.85 MB y contiene 146,650 *tokens.* Las respuestas a las consultas del usuario final salen casi de manera instantánea. En cambio, generar la representación fue relativamente tardado (17.48 minutos) debido a que se hizo una vez cargados todos los documentos, sin embargo, actualizar la representación es rápido puesto que el número de documentos no representados se reduce. En la Fig. 3 se muestra la salida del sistema a la consulta "virus y bases nitrogenadas", las respuestas marcadas en azul refieren a los documentos que contienen todos los términos de la consulta (respuestas más relevantes o "clasificación A"), mientras que las respuestas de color rojo (están abajo y no se alcanzan a apreciar) refieren a los documentos que contienen algún o algunos de los términos de la consulta (respuestas menos relevantes o "clasificación B"); el orden de la salida, dentro de cada clasificación, se calcula mediante la frecuencia de aparición de los términos en los documentos (en proporción directa). Por último, la respuesta marcada en color anaranjado corresponde al documento actual, es decir al documento que el usuario ha seleccionado y que se muestra en el lado derecho.





Estructura de los Ácidos Nucleicos

Miguel Angel Ordorica Vargas & María de la Luz Velázquez Monroy

Introducción

El estudio de los ácidos nucleicos es la disciplina que domina hoy en día la investigación cier en el campo de las Ciencias Biológicas. Tal relevancia es comprensible cuando reconocemo *"los ácidos nucleicos son los compuestos químicos responsables de almacenar, transmitir presar la infromación genética en todos los seres vivos"*. Sin embargo, como se describe a nuación, el papel de los ácidos nucleícos como material genético, no fue aceptado en form versal sino hasta la segunda mitad del siglo XX.

Historia

1869. El químico suizo *Johann Friedrich Miescher* (1844-1895) aísla del núcleo celular de los leucocitos una *"sustancia nitrogenada, rica en fosfatos y soluble en álcalis pero no en ácidos"*, a la que llama **Nucleína**. Hasta donde sabemos, Miescher creía que la función de la nucleína era almacenar fosfato.

Figura 1. *Frieder Miesch*

Treinta años después, en **1899**, el químico alemán *Richard Altmann* (1852-1901) desarrolla métodos para obtener la nucleína libre de proteínas y propone el cambio del nombre **Nucleína** por **Ácido Nucleico**.

Durante la década de **1920**, el químico *Phoebus Aaron Theodor Levene* (1869-1940) analiza los componentes de la molécula de ácido desoxirribonucleico; encuentra que contienen cuatro bases nitrogenadas Adenina, Guanina, Citosina y Timina; el azúcar Desoxirribosa y Fosfato.

Figura 2. *P. A. T. Leven*

Correctamente, dedujo que los nucleótidos eran las unidades estructu del DNA y que a su vez estaban formados por una base y fosfato, a unidos al azúcar. Desafortunadamente, en forma equivocada, concluy los cuatro nucleótidos se encontraban en cantidades iguales y postu **Teoría del Tetra-Nucleótido** como unidad básica del DNA, consider lo una molécula repetitiva, sin capacidad para ser el material genético.

En **1928**, el microbiólogo inglés *Frederick Griffith* (?-1940) descubre la transformación de las cepas no patógenas de *Streptococcus pneumoniae* usando restos de cepas patógenas muertas, y propone la existencia de un **Principio Transformador**, responsable del cambio, pero no hace ninguna sugerencia respecto de la naturaleza química del mismo.

Figura 3. *Fred Grif*

En **1944**, los investigadores *Oswald Theodore Avery* (1877-1955) *Colin M. MacLeod* y *Maclyn McCarty* (1912-2003) mediante la destrucción secuencial de las macromoléculas de las células de la cepa patógene de *S. neumonie*, aportan pruebas de que el DNA es el **Principio Transformador** de Griffith. A pesar de lo cual, para muchos investigadores, aún permanecía la duda sobre la naturaleza química del material genético.

Fig.3. Aplicación ZoomBooks Smart

En la Fig. 3 se muestra el primer prototipo que contempla varias etapas de la metodología como la expansión de consultas ni el sistema recomendado. Tampoco contiene la información relevante a la carga curricular de los estudiantes de bachillerato.

5. Conclusiones

Se plantea el diseño e implementación de un SRI instalable y con una base de datos de obras textuales relativas al nivel bachillerato y que serán una fuente de información documental para los estudiantes de regiones apartadas de las tres Mixtecas. La metodología que se sugiere hace uso del modelo Booleano extendido y Mergin. ZoomBooks Smart podrá ser usado por estudiantes de nivel medio superior que no tiene acceso a internet o que este es ineficiente para investigar al estilo Google, los documentos se mostraran en formato PDF, y en orden de relevancia, para su fácil visualización y distribución. Básicamente se busca realizar un sistema computacional instalable, fácil de usar y eficiente. Se plantea el uso de n-gramas y aprendizaje automático para la sugerencia de frases y para la recomendación de nuevos documentos que pudieran estar relacionados al interés del usuario (la metodología para esta tarea se está valorando).

Se ha desarrollado una prueba piloto de la metodología de recuperación que aquí se plantea. En su primera aproximación el motor de búsqueda está funcionando de manera estable. Se está trabajando en las etapas no consideradas de esta primera versión para terminar de manera adecuada con el producto planteado. Se indagará en el uso de otras técnicas metodológicas que puedan ayudar en la construcción de una segunda versión mejorada. Se podría, por ejemplo, implementar la representación posicional de las palabras para poder brindar al usuario final un buscador con sugerencias de frases nominales y de esta forma tener resultados más precisos.

REFERENCIAS

[1] Rodríguez, M. L. (2005). Modelos de recuperación de la información basados en información lingüística difusa y algoritmos evolutivos: Mejorando la representación de las necesidades de información (Doctoral dissertation, Universidad de Granada).

[2] Zermeño, M. G. G. (2012). Bibliotecas digitales: recursos bibliográficos electrónicos en educación básica. Comunicar: Revista científica iberoamericana de comunicación y educación, (39):119–128.

[3] Orera, L. O. (2003). Bibliotecas digitales de tesis doctorales: metodología para su planificación. Boletín de la Asociación Andaluza de Bibliotecarios, 18(72):55–72.

[4] Manning, C., Raghavan, P., & Schütze, H. (2010). Introduction to information retrieval. Natural Language Engineering, 16(1), 100-103.

[5] Kuna, H. D., Rey, M., Podkowa, L., Martini, E., & Solonezen, L. (2014, October). Expansión de Consultas basada en Ontologías para un Sistema de Recuperación de Información. In XVI Workshop de Investigadores en Ciencias de la Computación.

[6] Zuva, K. and Zuva, T. (2012). Evaluation of information retrieval systems. In- ternational journal of computer science & information technology, 4(3):35.

[7] Sánchez, E. P., Blanco, J. D., González, A. O., and Domínguez, N. A. (2017). Análisis de los procesos de tematización y estatizado en lingüística computacional.

[8] Porter, M. F. (2006). An algorithm for su x stripping. Program.Reyna, Y. C. F. (2012). Recuperación´on de la información: taxonomía de sus modelos. Revista cubana de Ciencias Informáticas, 6(2):1–8.

[9] Eraso, H. A. O. and Lozada, C. A. C. (2011). Stemming en español para documentos recuperados de la web* stemming in the spanish language for documents recovered from the web. Revista Unimar, (58):107–114.

[10] Figuerola, C. G., Zazo, Á. F., de Aldana, E. R. V., and Berrocal, J. L. A. (2004). La recuperación de información en español y la normalización de términos. Inteligencia Artificial. Revista Iberoamericana de Inteligencia Artificial, 8(22):135–145.

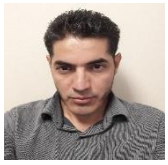
[11] Reyna, Y. C. F. (2012). Recuperación de la información: taxonomía de sus modelos. Revista Cubana de Ciencias Informáticas, 6(2), 1-8.

[12] Salton, G., Wong, A., and Yang, C.-S. (1975). A vector space model for auto- matic indexing. Communications of the ACM, 18(11):613–620.

[13] Robertson, S. E. (1977). The probability ranking principle in ir. Journal of documentation, 33(4):294–304.

[14] Grefenstette, G. (1993). Automatic thesaurus generation from raw text using knowledge-poor techniques.

Acerca de los autores



Cupertino Lucero Álvarez es estudiante de doctorado en Ciencias en Sistemas Computacionales y Electrónicos de la Universidad Autónoma de Tlaxcala UATX. Egresado de la Maestría en Ciencias de la Computación de la FCC-BUAP. Actualmente labora como Profesor de Tiempo Completo en la Universidad Tecnológica de Izúcar de Matamoros en la Ingeniería en Tecnologías de la Información. Ha publicado varios artículos en el área de Recuperación de Información y Procesamiento del Lenguaje Natural. Desde 2015 cuenta con reconocimiento al perfil deseable que otorga PRODEP.

M. Larios-Gómez. Originario de Puebla-México. Profesor investigador tiempo completo en la Benemérita Universidad Autónoma de Puebla (BUAP). Recibió su grado de licenciatura y maestría en ciencias de la computación en la facultad de ciencias de la computación (BUAP) 1997-2001 y 2001-2003 respectivamente. Estudios de doctorado en sistemas en transportes inteligentes en Compiègne-Francia 2013. Desde 2004 es profesor en la facultad de ciencias de la computación (BUAP). Su interés en la investigación incluye tópicos en cómputo distribuido, blockchain, cómputo de alto rendimiento, sistemas de tiempo real y computo pervasivos. Actualmente colabora en proyectos de investigación sobre supercómputo en el laboratorio nacional del suroeste LNS.

Pascual Pérez Cruz, con Maestría en Sistemas Computacionales de la Universidad Popular Autónoma de Puebla en el año 2012. Obtuvo su Licenciatura en Informática en el Instituto Tecnológico Superior de Acatlán de Osorio en 2006 ha participado en varios proyectos de investigación, así como en artículos científicos de índole nacional e internacional.

Brian Manuel González Contreras maestro en ciencias en ingeniería electrónica, con especialidad en control automático. Obtuvo el grado de doctor en ciencias en control automático, procesamiento de señales e ingeniería informática por parte de la Universidad Henri Poincaré, Nancy I, en Francia. Ha participado en diversos proyectos nacionales e internacionales, entre los que destacan: diseño de alimentación eléctrica en alta y baja potencia para subestaciones eléctricas de CFE en Minatitlán, Veracruz; diseño de líneas de suministro de gas y electricidad en alta y baja potencia para PEMEX en Paraíso, Tabasco; diseño del SCADA para PEMEX de la red nacional de refinación, en ciudad de México; diseño de sistemas de control seguros y

confiables para la compañía Neste-Jacobs Oy, en Finlandia. Sus líneas actuales de investigación consideran: el modelado de sistemas, la detección y diagnóstico de fallas, el control tolerante a fallas, la identificación/modelado de procesos industriales y el análisis de confiabilidad de sistemas ciber-físicos.

Carlos A. Ortiz Ramírez. Ingeniero Industrial en electrónica del Instituto Tecnológico de Puebla en 1993, Maestro en Ciencias con especialidad en Optoelectrónica de la FCFM de la BUAP en 1997 y pasante de Maestro en Dirección Escolar en el Colegio Latinoamericano de Posgrados. Participo en proyectos de telecomunicaciones de la sonda de Campeche con el Instituto Mexicano del Petróleo, más de 20 años como PTC en la UTIM donde realiza proyectos de trasferencia de Tecnología con financiamiento de la SDR, SAGARPA y UTIM. Asesor en Estadía Técnica de más de 140 alumnos de TSU e Ingeniería. Ha impartido conferencias en congresos, ponencias y talleres nacionales e internacionales con ISSN ONLINE y/o ISBN. PROMEP le Reconoce con Perfil deseable desde 2008 y vigencia al 2021.

e-mail: carlos.ortiz@utim.edu.mx

Juventino Montiel Hernández, es Maestro en ciencias en Ingeniería en Computación por la Universidad Autónoma de Tlaxcala Obtuvo el grado en diciembre del 2012. Actualmente es Profesor investigador y estudiante de doctorado en la Facultad de Ingeniería y Tecnología. Cuenta ya con varias publicaciones internacionales. Sus intereses de investigación son en las áreas de Inteligencia artificial, interacción Humano-Robot y desarrollo de aplicaciones bio-inspiradas.