Benemérita Universidad Autónoma de Puebla

Facultad de Ciencias de la Computación

Extracción y manejo de terminología a partir de un corpus lingüistico de textos especializados

Tesis presentada para obtener el título de:
Maestro en Ciencias de la Computación

Presenta:
Oswaldo Jair García Franco

Asesor de Tesis:
Dr. David Eduardo Pinto Avendaño

Co-asesor de Tesis:
Dra. Angeles Belém Priego Sánchez

Enero 2023

Dedicatoria

Me gustaría dedicar esta tesis a Mónica mi amada esposa. Por su paciencia, comprensión y apoyo en todo momento a lo largo de este emocionante pero igualmente complicado proceso, además de motivarme para mejorar como persona y como investigador.

A mis padres por ayudarme a afrontar las adversidades, por su apoyo incondicional y por enseñarme que el conocimiento es un camino de aprendizaje infinito el cual requiere de esfuezo y pasión.

Agradecimientos

1. A mis asesores El Dr. David Pinto y la Dra. Belém Priego por su ayuda, paciencia, por fomentar la pasión por la investigación y por sus conocimientos brindados.

2. A mi comité tutorial por presentar sus importantes evaluaciones en los avances del proyecto.

3. A mis compañeros del área porque hicieron de la maestria un lugar increiblemente sano y ameno para la investigación.

4. A mis profesores de la maestría por compartir su conocimiento con dedicación durante este proceso de aprendizaje.

....

Resumen

El desarrollo de sistemas de recuperación de la información refleja una rápida progresión que se aleja de los enfoques manuales de adquisición, indexación y búsqueda basados en bibliotecas información a métodos cada vez más automatizados.
Para este proyecto se cuenta con un corpus de obras literarias, otro de leyes universitarias y uno más pequeño de chistes. Por lo que era necesario el desarrollo de herramientas computacionales que permitan extraer y manejar terminología a partir de estos corpus lingüisticos; Por ello se desarrollaron dos herramientas. La primera es un buscador que permite realizar consultas sobre un índice que contiene todos los documentos, para llevar a cabo el objetivo primero se aplicó un preprocesamiento sobre todos los documentos; posteriormente, fueron indexados. Las consultas se llevan a cabo a través del indice por medio de un buscador en formato web con un modelo cliente-servidor: en el lado del cliente el usuario puede ejecutar una consulta que será enviada al servidor, éste ejecutará las operaciones correspondientes para poder retornar los resultados obtenidos al cliente. Los resultados son mostrados con un número preciso de documentos encontrados, tiempo de ejecución de la consulta realizada, metadatos con información relevante de los textos como el nombre del autor y año de publicación. Además, cada resultado contará con un segmento breve de texto que hace referencia a la consulta realizada. También, se desarrolló una aplicación de escritorio que permite la extracción automática de colocaciones y locuciones de un texto seleccionado las cuales son enviadas aun archivo de texto.

Índice general

Capítulo 1

Introducción

1.1. Planteamiento del problema

En la época anterior a los buscadores, las personas preferían obtener información que fuera transmitida de persona a persona, al igual que utilizar un agente humano para hacer una reservación de un vuelo, comprar boletos para una función de cine ó buscar un título en una biblioteca . Sin embargo actualmente la mayor parte de la información puede ser indexada y consultada en tiempo real lo que conlleva al objetivo de este trabajo de investigación .
Para este proyecto se cuenta con un corpus diverso de obras literarias en diversos formatos por lo que se planteó desarrollar herramientas computacionales que permitan indexar todas las obras para realizar diversos tipos de consultas y extraer terminología (colocaciones y locuciones) automáticamente.

1.2. Objetivos

Los objetivos definidos plantean dos tipos: general y objetivos especificos que nos permiten tener una visión de como debe desarrollarse

1.2.1. Objetivo General

Diseñar una propuesta computacional que sea capaz de extraer y manejar terminología a partir de un corpus lingüistico de textos especializados.

1.2.2. Objetivos Específicos

1. Extraer colocaciones y locuciones verbales del corpus lingüistico de textos especializados.

2. Seleccionar las técnicas y métodos actuales para la extracción automática de terminología, a partir de textos planos (sin ningún tipo de etiquetado manual).

3. Diseñar la plataforma computacional para llevar a cabo el proceso de búsqueda de información. En este caso, se considera el uso de técnicas de recuperación de información.

4. Proponer un método para el descubrimiento automático de términos relacionados , a partir de textos planos.

5. Uso de diversos textos para enriquecer la extracción de terminología.

1.3. Antecedentes

A finales del 2020 se tuvo conocimiento de un interés particular por parte de la universidad de Alicante por el desarrollo de sistemas para el manejo eficiente de información relacionada con textos literarios.
Durante 2021 se ha venido desarrollando un prototipo de un sistema de recuperación de información que permite realizar búsquedas rápidas sobre dichos textos literarios, por lo cual también se requeria la extracción de colocaciones y locuciones.

1.4. Justificación

Se dispone de un conjunto de obras literarias en diferentes formatos por lo cual se requiere de herramientas computacionales capaces de procesarlos e indexarlos para

efectuar consultas de manera rápida.
La extracción automática de terminología es de suma importancia , ya que nos permite tener en consideración las frases más utilizadas de cada texto para su uso en estudios posteriores.

1.5. Metodología

El presente proyecto se relizó en diversas etapas: Comenzando por la elaboración del estado del arte lo que permitió analizar diversos planteamientos , propuestas de solución y resultados que sirvieron como parámetros para llevar a cabo la implementación.
Posteriormente se desarrolló una propuesta de solución al problema planteado por medio de dos herramientas computacionales: una que permite realizar consultas y otra que permite extraer colocaciones y locuciones. Esto se cumplió mediante el diseño de un prototipo que permitió programar el modelo de solución propuesto y ponerlo a prueba para validar el código.
Las pruebas se realizaron sobre un conjunto de diversas obras literarias que fueron indexadas. En la etapa final se muestran los resultados obtenidos con las herramientas desarrolladas, mostrando el cumplimiento de los objetivos planteados anteriormente.

1.6. Hipótesis planteada

Tomando en cuenta lo desarrollado anteriormente, se formula la siguiente hipótesis a comprobar:

Se pueden desarrllar herramientas computacionales que permitan realizar consultas rapidamente y además extraer colocaciones y locuciones de manera automática.

1.7. Distribución del trabajo de tesis

El presente trabajo de tesis se organiza de la siguiente manera:

- Capítulo 1, Introducción. Se muestra el planteamiento del problema, justificación, objetivos, metodología seguida e hipótesis planteada.

- Capítulo 2, Estado del Arte. Se describen las contribuciones realizadas por diversos autores en las tareas de "Extracción y validación de conceptos ontológicos" y "Extracción de relaciones semánticas entre conceptos" puntos clave de desarrollo de este trabajo de tesis.
- Capítulo 3, Marco teórico. Se exponen las bases teóricas de la investigación, útil para comprender el significado del contenido expuesto.
- Capítulo 4, Diseño. Se muestran los modelos propuestos para la resolución del problema expuesto.
- Capítulo 5, Resultados. Se exponen a detalle los resultados obtenidos por los modelos propuestos.
- Capítulo 6, Conclusiones. En esta sección son mostradas las conclusiones obtenidas al realizar este trabajo de tesis y exponemos el trabajo a futuro.

Capítulo 2

Estado del Arte

En este capítulo, se presenta un estudio detallado del estado del arte con el objetivo de conocer investigaciones anteriores y trabajos relacionados con la extracción de terminología y la recuperación de información basada en índices invertidos y posicionales.

Iniciamos con la importancia de la recuperación de la información a través del tiempo. Sanderson y Croft. [15] describen la evolución de este tipo de sistemas desde 1950. El objetivo de la RI es encontrar información que es relevante para la consulta de un usuario en colecciones de información normalmente no estructurada o semiestructurada. Desde antes del uso de internet ya existía un manejo convencional de grandes cantidades de información como los libros o periódicos que eran indexados usando esquemas de catálogos facilitando su búsqueda. Posteriormente, se usaron sistemas mecánicos de búsqueda como la máquina de Goldberg que buscaba patrones de puntos o letras a través de entradas de catálogo almacenadas en un rollo de microfilme. Después, con el gradual crecimiento de información científica disponible Holmstrom describió que la UNIVAC era capaz de buscar referencias de texto asociadas a un asunto, que se almacenaban en cinta magnética. Después se utilizó un híbrido entre la recuperación de información mecánica y computacional llamada booleana que resultaba en un conjunto de documentos que coincidían exactamente con la consulta realizada. En los 70's se desarrolló la búsqueda basada en frecuencia de palabras conocida como *tf-idf*. De los 80's a hasta mediados de los 90's con la indexación semántica latente (LSI), por sus siglas en inglés, permitió encontrar palabras

con signficados semánticos similares y así obtener un rango más amplio de documentos relevantes.Finalmente todo evolucionó a gran escala con la búsqueda web y la creación del PageRank por los creadores de Google, añadiendo análisis de enlaces y multiples representaciones de documentos haciendo que la RI sea más compleja.

S. Ibrihich et al. [20] explican el incremento exponencial de información de los últimos años lo que ha fomentado innovar en la búsqueda de información relevante en menos tiempo y describen una estructura general que debe tener un sistema de recuperación de información. Dicha estructura se compone de la colección de documentos, un rastreador (crawler) que recupera información a partir de un conjunto de sitios web donde se obtienen los documentos y un sistema de ranking que permite definir cuáles son los resultados más relevantes. Sin embargo, para que ésto sea posible es necesario realizar un procedimiento de preprocesado al texto que es crucial, el cual consiste en analizar los documentos, tokenizar, identificar las palabras cerradas en caso de ser necesario, llevar a cabo el truncamiento de las palabras, extracción de características y evaluar los resultados por medio de algún mecanismo de representación como *tf-idf*.

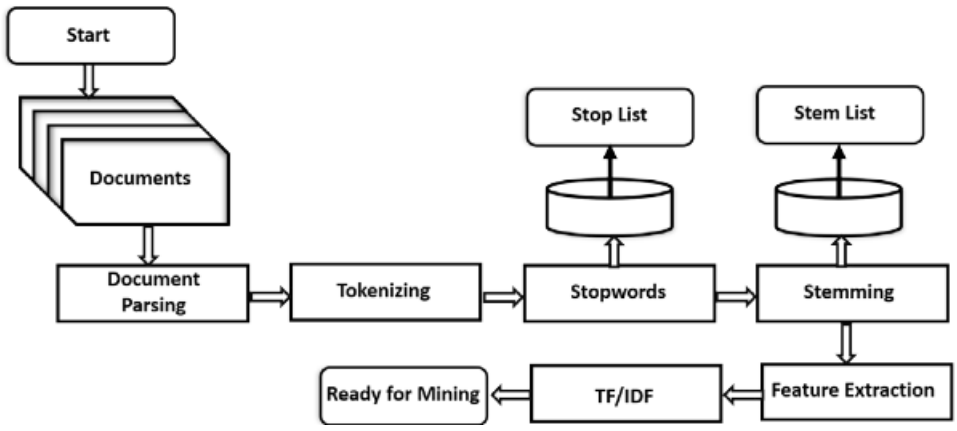

Figura 2.1: Preprocesamiento de documentos de texto [20].

Guillen et al [2] hacen énfasis en la gran cantidad de información existente en la actualidad, lo cual hace menos eficientes a los sistemas de recuperación de información, cuando el usuario requiere información especifica dependiendo de sus necesidades y preferencias. Proponen diseñar un perfil de documento capaz de re-

presentar metadatos semánticos extraídos utilizando diferentes tecnologías de PLN, demostrando que diversas de estas tecnologías pueden converger en un ecosistema único.Para ello compararon algunas de estas tecnologías por medio de la tarea realizada, medida, puntuación y tipo de prueba a partir de experimentos realizados en investigaciones anteriores.

Chiranjeevi y Shenoy [5] implementaron un buscador web que permite subir un documento de texto de preferencia del usuario para preprocesarlo y posteriormente realizar consultas sobre éste. Proponen preprocesar, tokenizar el texto, recuperar la información usando *tf-idf*, además de agregar la identificación por partes de la oración, lo cual mejora el contexto de las palabras con información detallada de si misma y de sus vecinos. La entrada a un algoritmo de etiquetado es una secuencia de palabras y un conjunto de etiquetas, y la salida es una secuencia de etiquetas y una etiqueta para cada palabra. Kochmar [14] presenta el esquema de preprocesamiento de la librería Spacy (ver Figura 2.2).

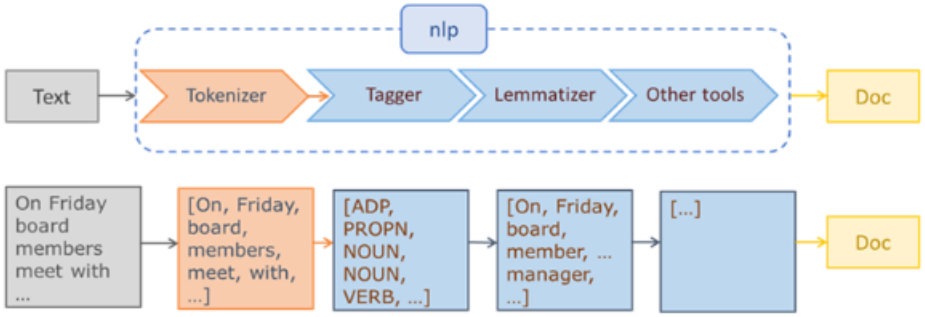

Figura 2.2: Procesamiento con resultados intermedios usando la libreria Spacy [6].

Finalmentes fueron implementadas las medidas de precision, recall y F-measure para analizar sus resultados.

Rafique y Hassan [18] plantean que casi todas las funciones de recuperación de la información usan la técnica de bolsa de palabras ignorando la importancia de la proximidad. Ellos encontraron oraciones cortas formadas por distintas palabras o términos en su primera aparición en los documentos para calcular la proximidad

y lo incorporaron en una función.Demostraron que hay un número significativo de oraciones cortas formadas por palabras distintas que se pueden utilizar para explotar la proximidad y búsqueda de frases utilizando un indice posicional invertido.

Sodel et al. [22] desarrollaron un sistema de recuperación de información ya que consideran que es importante para la toma de decisiones en las compañías en el cual almacenaron e indexaron información de una compañía de aviación en un índice invertido sobre el cual se pueden realizar consultas y los resultados son ordenados por relevancia. Su sistema fue llamado *be intelligent* y da soporte a la administración en el uso de documentos digitales usados durante inspecciones de aeronaves reduciendo el tiempo de búsqueda de información.

Alia Hassan en [13] propone un método de representación de información enfocado en el preprocesamiento de los documentos para los sistemas de recuperación de información, centrándose así en una tokenización eficaz basada en un método llamado índice invertido mejorado. Así la tokenización en documentos ayuda a satisfacer la necesidad de información del usuario con mayor precisión y así reducir el tiempo de busqueda y tiempo de preprocesamiento usando un índice invertido.

Manning y Schütze [6], definen a las colocaciones y sus características además de algunas técnicas para su extracción como: frecuencia, media y varianza, t-test, chi-square test e información mutua o PMI. Además, hacen enfásis en la noción de colocación refiriendo a que existen más de una definición, además definen la no composicionalidad,no sustituibilidad y no modificabilidad que caracterizan principalmete a las colocaciones.

Silas y Demberg [21], sugieren medidas para identificar automáticamente aquellas expresiones de varias palabras donde la primera parte es particularmente predictiva del resto. Evaluaron sus medidas contra los datos de asociación humana recopilados en una prueba cloze que consiste en quitar ciertos elementos, palabras o signos y se le pide al participante que reemplace el elemento del idioma que falta, utilizando las técnicas de extracción mencionadas en [6]. Seleccionaron un grupo de verbos altamente predictivos y generaron pares verbo-sustantivo donde el verbo es altamente predictivo del sustantivo. Posteriormente calcularon el rango promedio de esos pares para cada una de las medidas utilizadas.

Thanopoulus y Fakotakis [23], utilizaron algunas medidas para la extracción automática de colocaciones como: t-score, Pearson X-square, PMI y una medida teórica llamada dependencia mutua. La extracción se realizó utilizando unicamente bigramas y se compararon los resultados de cada una de las técnicas mencionadas.

Sulema Torres [19] presenta la extracción automática de un diccionario de colocaciones en español a partir de un corpus con las estructuras sintácticas marcadas manualmente. Las relaciones de dependencias encontradas en tal corpus, junto con sus frecuencias, constituyen el diccionario de colocaciones. Para el desarrollo de este proyecto se utilizó el corpus Cast3LB en español éste cuenta con más de cien mil palabras (aproximadamente 3500 oraciones) y la extracción del diccionario de colocaciones puede ser descrita en los siguientes pasos:

1. Transformación del corpus de constituyentes a corpus de dependencias: Se extrajeron las reglas grmáticales del corpus Cast3LB, después de determinaron los rectores o cabezas de cada regla grmatical mediante el uso de heurísticas finalmente se utilizó la información de rectores o cabezas, de forma recursiva, para determinar cuáles reglas y componentes se subirán de nivel en un árbol de dependencias .

2. Extracción de colocaciones: se Recorre el árbol de dependencias en profundidad de izquierda a derecha, comenzando de la raíz y por cada nodo hijo del nodo visitado, se extrae el nodo padre, el nodo hijo y la relación de dependencia entre ellos. Si el nodo hijo es una preposición entonces éste se considera como la relación de dependencia y el nodo hijo de la preposición se considera el nodo hijo de la colocación. No se consideran colocaciones donde existen determinantes. Por ejemplo, se considera la siguiente oración: *"Los policias velarán por la seguridad de los líderes"*. El árbol de dependencias es el siguiente:

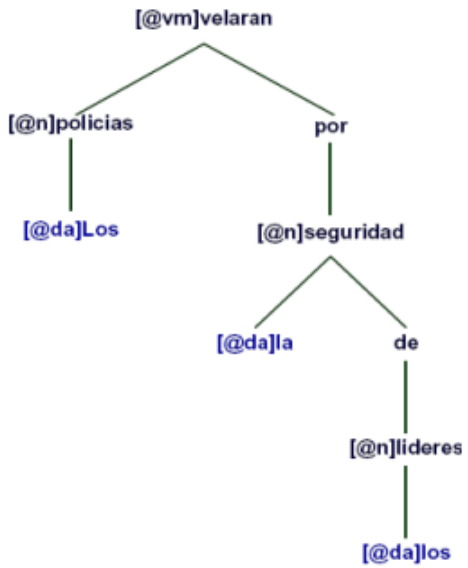

Figura 2.3: Árbol sintáctico de dependencias para la oración "Los policías velarán por la seguridadde los líderes" [19]

Después de recorrer el árbol, las colocaciones extraídas son las siguientes:

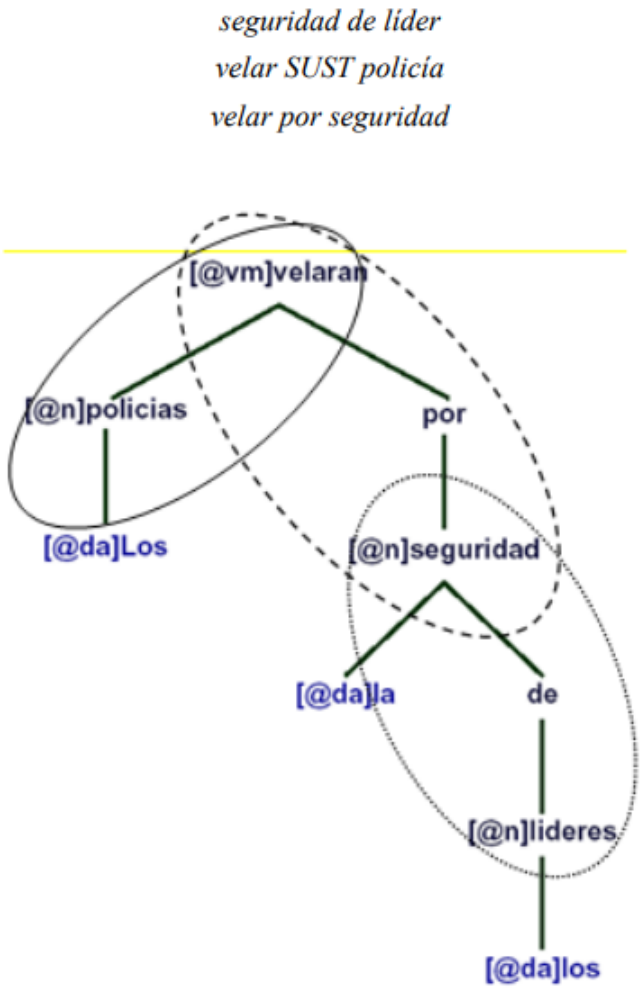

Figura 2.4: Colocaciones extraídas del árbol de dependencias de la oración "Los policías velaránpor la seguridad de los líderes" [19]

3. Agregar información estadística: para las frecuencias del diccionario se ordenaron las colocaciones obtenidas, se contaron las frecuencias de las colocaciones y se eliminaron las colocaciones repetidas.

Jose de Lucca en [8] desarrolló una herramienta que permite extraer unidades fraseológicas de un corpusa partir de una selección de corpus formada por diccionarios monolingües, bilingües, de fraseología y tesis relativas a fraseología dónde se extrajeron las unidades fraseológicas más relevantes para la posterior construcción de un diccionario que es construido mediante un peso asignado a las palabras de la

oración que es determinado por su frecuencia.

Capítulo 3

Marco teórico

En este capítulo, explican los conceptos teóricos usados en el presente trabajo de tesis.
Se inicia con la definición de Procesamiento de Lenguaje Natural, más adelante se define la recuperación de la información; Posteriormente, se profundiza en el uso de índices invertidos. A continuación se explican algunos métodos para identificar colocaciones de manera automática. Finalmente se explican las herramientas de trabajo utilizadas para el desarrollo de este proyecto.

3.1. Procesamiento de Lenguaje Natural

Es un campo de las ciencias de la computación, la inteligencia artificial y de la lingüística que estudia las interacciones entre la computadora y el lenguaje humano. Según Augusto et al. [16], el conocimiento científico es el resultado de muchos años de investigación sobre temas aparentemente no relacionados por lo que se incrementa en forma de documentos, libros, artículos que se almacenan en diferentes formatos. Sin embargo, lo que es conocimiento para nosotros no lo es para las computadoras. El estudio del lenguaje natural tiene dos objetivos:

- Faclitar la comunicación con la computadora para que usuarios no especializados puedan acceder a ella.
- Modelar procesos cognoscitivos que entran en juego en la comprensión del

lenguaje y así diseñar sistemas que realicen tareas lingüísticas complejas como: traducción, resumenes de textos, recuperación de información, etc.

El Procesamiento de Lenguaje Natural consiste en la utilización de un lenguaje natural para comunicarnos con la computadora. El uso de éste facilita el desarrollo de programas que realicen tareas relacionadas o desarrollar modelos que ayuden a comprender los mecanismos humanos relacionados en el lenguaje.

3.1.1. Ejemplos de aplicaciones del PLN

- Recuperación de la información (usada para este proyecto): trata con la representación, almacenamiento, organización y el acceso a la información. Es el conjunto de tareas mediante las cuales, el usuario localiza y accede a los recursos que son pertinentes para la resolución de un problema planteado [11].

- Traducción automática: es una rama de la lingüística aplicada,es importante desde el punto de vista científico, pues sirve como campo experimental de la lingüística y la informática, especialmente en el ámbito del procesamiento y análisis automático del lenguaje natural. Esta disciplina aplicada permite establecer vínculos con otras disciplinas de la lingüística aplicada como la traductología, la terminología, la sicolingüística y la pragmática, entre otras [9].

- Extracción de información y resúmenes: tienen como meta extraer información precisa a partir de textos no estructurados y presentarla al usuario de forma consistente. La extracción de resúmenes muestra esta información mediante frases en lenguaje natural y la extracción de información a través de representaciones estructuradas de los datos [10].

- Reconocimiento de voz: Es una rama de la inteligencia artificial que tiene como objetivo posibilitar la comunicación entre humanos y sistemas informáticos. Un sistema de reconocimiento de voz puede entender las palabras emitidas por un ser humano de forma natural [12].

3.2. Recuperación de la información

Para Bordigon y Tolosa [4], la recuperación de información es la Ciencia de la búsqueda de información en documentos electrónicos o colección digital, y tiene como objetivo la recuperación de textos, imágenes, sonidos o datos de otras características de manera relevante. Con la consolidación de Internet como principal medio de consulta de información, se hace necesario tener herramientas que permitan que la recuperación de información sea efectiva en términos de tiempo invertido por los usuarios.

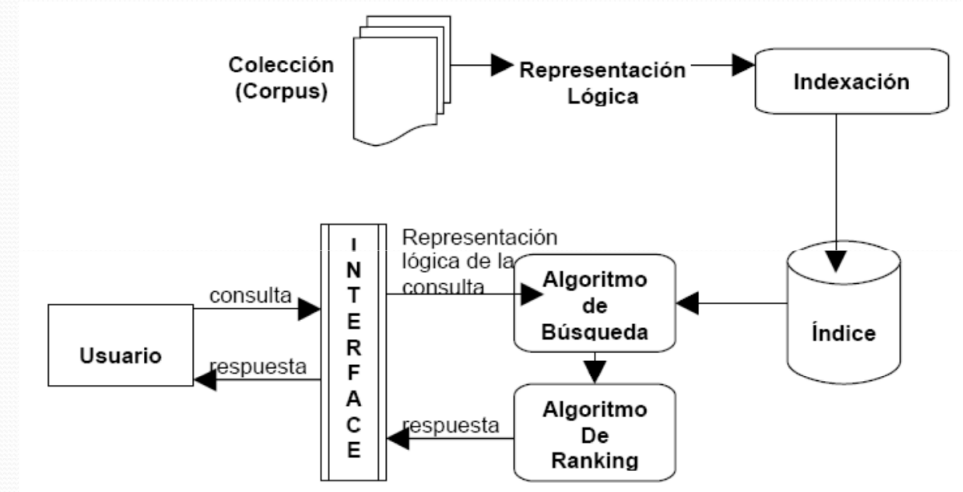

Figura 3.1: Arquitectura básica de un *SRI*(sistema recuperación de información) [11].

A continuación se describen los componentes de la arquitectura básica de un sistema de recuperación de información.

- Colección o corpus: Conjunto de textos o de datos destinados a la investigación
- Indexación: Consiste en construir estructuras de datos (denominadas índices) que almacenen y soporten búsquedas eficientes.
- Algoritmo de búsqueda: Acepta como entrada una expresión o consulta de un usuario para verificar en el índice, qué documentos pueden satisfacer la

búsqueda, posteriormente un **algoritmo de ranking** determinará la relevancia de cada documento.

- Interface: Permite que el usuario especifique la consulta mediante una expresión en un lenguaje preestablecido; además, permite mostrar los resultados obtenidos. También puede mostrar algunos datos extra como metadatos, tiempo de ejecución y número de resultados.

3.3. Índices invertidos

Según Cortez [24], un índice invertido es una estructura de datos que contiene los términos del vocabulario de una colección de documentos. Por cada término almacena una lista de registros con información referente al número de identificación (ID) de cada uno de los documentos que contienen al término, y de ser necesario las posiciones convirtiendose en un índice invertido posicional aunque una de sus desventajas es el alto uso de memoria RAM al cargar en memoria un gran número de documentos.
En la figura 3.2 se observa la estructura de un índice invertido ,del lado izquierdo se encuentran las palabras y del lado derecho en que documentos se encuentran.

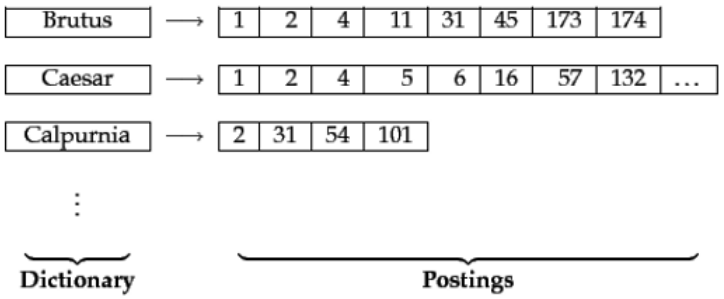

Figura 3.2: Estructura de un índice invertido [7].

3.4. Índices posicionales

De acuerdo a Manning [7], el índice posicional es el más comúnmente empleado. Aquí, para cada término del vocabulario, se almacenan listas de la forma docID:

<*position1*, *position2*, ...> donde cada posición es un token en el documento. Cada lista usualmente almacena la frecuencia del término, como se muestra.

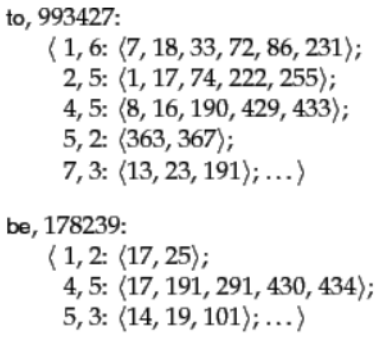

Figura 3.3: Estructura de un índice posicional [6].

Para procesar una consulta por frase, aún necesita acceder a las entradas del índice invertido para cada término distinto. Se comienza con el término menos frecuente y luego restringir aún más la lista de posibles candidatos. Se verifica que ambos términos estén en un documento, también se debe verificar que sus posiciones de aparición en el documento sean compatibles con la consulta de frase que se está evaluando.

```
INTERSECT(p1, p2)
 1  answer ← ⟨ ⟩
 2  while p1 ≠ NIL and p2 ≠ NIL
 3  do if docID(p1) = docID(p2)
 4        then ADD(answer, docID(p1))
 5             p1 ← next(p1)
 6             p2 ← next(p2)
 7        else if docID(p1) < docID(p2)
 8                then p1 ← next(p1)
 9                else p2 ← next(p2)
10  return answer
```

Figura 3.4: El algoritmo encuentra lugares en donde aparecen los dos términos por k palabras de separación, regresa una lista con el docID y la posición del término en *p1* y *p2* [6].

3.5. Etiquetado de partes de la oración

El etiquetado de partes del discurso (POS) es importante para identificar el papel que tiene cada palabra dentro de una oración. Es la identificación de la clase morfológica de cada forma de las palabras utilizando información léxica y contextual. Esta es una tarea fundamental en el procesamiento del lenguaje natural. Sin embargo, el lenguaje es ambiguo y una palabra puede tener varias interpretaciones morfosintácticas dependiendo del contexto en que se encuentre. Por ejemplo, *casa* puede ser un sustantivo común femenino y también una forma de presente o imperativo del verbo *casar*. Para esta tarea, la librería Spacy ofrece un etiquetador el cual funciona en diversos idiomas incluyendo inglés y español. A continuación se muestra un ejemplo del etiquetado que ofrece esta herramienta con la frase *tomar el toro por los cuernos*.

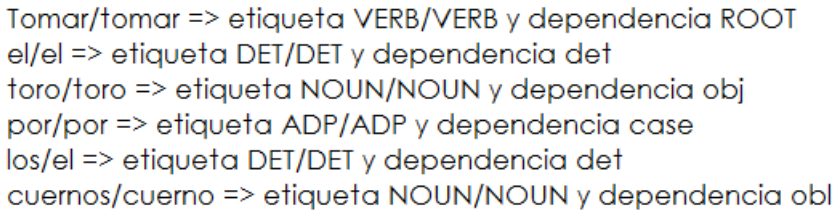

Figura 3.5: Ejemplo de etiquetado con la librería Spacy tomado de la herramienta de extracción de colocaciones del presente proyecto.

El etiquetador divide la frase en tokens y asigna una categoría a cada palabra. Las categorías estan definidas de la siguiente manera:

POS	DESCRIPTION	EXAMPLES
ADJ	adjective	*big, old, green, incomprehensible, first*
ADP	adposition	*in, to, during*
ADV	adverb	*very, tomorrow, down, where, there*
AUX	auxiliary	*is, has (done), will (do), should (do)*
CONJ	conjunction	*and, or, but*
CCONJ	coordinating conjunction	*and, or, but*
DET	determiner	*a, an, the*
INTJ	interjection	*psst, ouch, bravo, hello*
NOUN	noun	*girl, cat, tree, air, beauty*
NUM	numeral	*1, 2017, one, seventy-seven, IV, MMXIV*
PART	particle	*’s, not,*
PRON	pronoun	*I, you, he, she, myself, themselves, somebody*
PROPN	proper noun	*Mary, John, London, NATO, HBO*
PUNCT	punctuation	*., (,), ?*
SCONJ	subordinating conjunction	*if, while, that*
SYM	symbol	*$, %, §, ©, +, −, ×, ÷, =, :), 😝 *
VERB	verb	*run, runs, running, eat, ate, eating*
X	other	*sfpksdpsxmsa*
SPACE	space	

Figura 3.6: categorías utilzadas por la librería Spacy extraídas de su documentación oficial.

Finalmente, se presenta, en la figura 3.7, el preprocesamiento general que usa la librería Spacy para llevar a cabo el etiquetado.

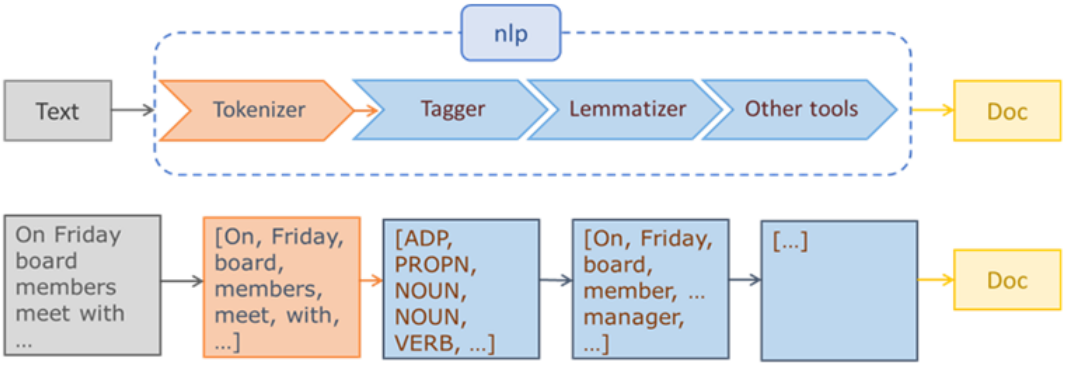

Figura 3.7: Procesamiento del etiquetador Spacy [8].

3.6. Lematización y truncamiento

El truncamiento generalmente se refiere a un proceso heurístico crudo que corta los extremos de las palabras. Con la esperanza de lograr este objetivo la mayor parte del tiempo y, a menudo, incluye la eliminación de afijos derivativos; por ejemplo, la raíz de romper y rompiendo sería *romp*. La lematización se refiere a hacer las cosas correctamente con el uso de un vocabulario y un análisis morfológico de las palabras, normalmente con el objetivo de eliminar solo las terminaciones flexivas y devolver la forma base o de diccionario de una palabra, que se conoce como el lema [6] por ejemplo el lema de *desarrollo* es *desarrollar* .

3.7. Colocaciones y locuciones

Manning y Schütze [6] definen una colocación como una expresión de dos o más palabras que corresponden a una forma convencional de decir las cosas.

3.7.1. Características de las colocaciones

Para identificar una colocación se suele considerar la adyacencia de palabras como: *mercado negro*. Sin embargo, una frase puede ser una colocación incluso si no es consecutiva [6], por ejemplo: *mantener la compostura*, por ello además de la definición debe considerarse lo siguiente:

- No composicionales: su significado no puede determinarse a partir del significado de sus partes, por ejemplo: hot dog, sacapuntas, limpia parabrisas.

- No sustituibilidad: No se puden sustituir por sinónimos o palabras relacionadas, por ejemplo: White wine-yellow wine, mercado negro- mercado oscuro, memoria interna-memoria intrínseca.

- No modificabilidad: Muchas colocaciones no pueden ser libremente modificadas con material léxico adicional, por ejemplo: estructura ósea- estructura óseas, fútbol americano-fútbol americana.

Una locución es una expresión característica de una lengua que esta formada por un conjunto de palabras con una estructura fija y que tiene un significado que no puede deducirse del significado de las palabras que lo forman. Además tienen un grado de fijación, es decir, que tanto pueden cambiar los componentes de la unidad, por ejemplo: *me costo un ojo de la cara* no es lo mismo que *me costo un ojo del pie*.

3.7.2. tipos de locuciones

- Adjetiva: Se utiliza como adjetivo por ejemplo: me compré ropa de ***segunda mano***(usada).

- Adverbial: Se usa como adverbio por ejemplo: el delincuente disparó ***a quemarropa***(directamente).

- Nominal: Se usa como sustantivo por ejemplo: Raúl encontró a su ***media naranja***(amor ideal).

- Conjuntiva: Se utiliza como conjunción por ejemplo: Te explicaré el ejercicio ***a condición de que*** prestes mas atención (si).

- Pronominal: Se utiliza como pronombre por ejemplo: No encontré ningún error grave, ***solo alguno que otro*** (error).

- Proposicional: Se utiliza como una preposición por ejemplo: Hubo un error en la venta de entradas y quedé ***en medio de*** una pareja (entre).

- Verbal: Se utiliza como verbo por ejemplo: Los ***eché de menos*** todos estos meses (extrañé).

3.8. Generación de N-gramas

Segun Moore y Quirk [17] un n-grama es una subsecuencia de n elementos de una secuencia dada. El estudio de los n-gramas es interesante en diversas áreas del conocimiento. Por ejemplo, es usado en el estudio del lenguaje natural, en el estudio de las secuencias de genes y en el estudio de las secuencias de aminoácidos.

La forma en la que extraemos los gramas se tiene que adaptar al ámbito que estamos estudiando y al objetivo que tenemos en mente. Por ejemplo en el estudio del lenguaje natural podríamos construir los n-gramas sobre la base de distintos tipos de elementos como por ejemplo fonemas, sílabas, letras, palabras. Algunos sistemas procesan las cadenas de texto eliminando los espacios. Otros no. En casi todos los casos, los signos de puntuación se eliminan durante el preproceso.

Se puede usar gramas para casi todos los ámbitos. Por ejemplo, se han usado n-gramas para extraer características comunes de grandes conjuntos de imágenes de la Tierra tomadas desde satélite, y para determinar a qué parte de la Tierra pertenece una imagen dada.

Para ciertos valores de n los n-gramas tienen nombres especiales. Por ejemplo:

Los 1-gramas también se llaman unigramas. Los 2-gramas también se llaman bigramas o digramas. Los 3-gramas también se llaman trigramas.

3.9. Técnicas de extracción de colocaciones y locuciones

Para identificar colocaciones y locuciones automáticamente es necesario el uso de diversas técnicas que se describen a continuación:

- Cálculo de colocaciones por frecuencia de ocurrencia: Consiste en encontrar colocaciones contando el número de ocurrencias, usualmente resulta en muchos pares de palabras que necesitan ser filtrados.

- Cálculo por hipótesis de asumpción de distribución normal (T-test): La prueba busca la diferencia entre los valores observados y esperados, escalado por la varianza de los datos, y nos dice qué tan probable es obtener una muestra de esa media y varianza, suponiendo que la muestra se extrae de una distribución normal con media μ. Se debe pensar en el corpus como una larga secuencia de N-gramas, y las muestras son entonces variables aleatorias con valor 1 si el bigrama de interés ocurre y 0 en caso contrario.

$$t = \frac{\overline{x} - \mu_o}{\frac{s^2}{\sqrt{n}}}$$

- Cálculo de colocaciones por hipótesis de asumpción de distribución normal Pearson X^2: Compara las frecuencias observadas con las frecuencias esperadas por independencia. Si la diferencia entre frecuencias observadas y esperadas es grande, podemos rechazar la hipótesis nula de independencia.

$$X^2 = \sum_{i,j} \frac{(O_{i,j} - E_{i,j})^2}{E_{i,j}}$$

- Cálculo por información mutua (PMI): Es una medida de cuanto nos dice una palabra sobre otra.

$$PMI = \log(\frac{P(w1, w2)}{P(w1)P(w2)})$$

3.9.1. Filtrado de colocaciones por patrones

Aún con el uso de las técnicas ya mencionadas es necesario filtrar los resultados obtenidos por medio de patrones definidos por las etiquetas que utiliza la librería Spacy, por ejemplo: *tomar el toro por los cuernos* se conforma por el siguiente patron: VERB, DET, NOUN, ADP, DET, NOUN. finalmente se considera un número de n-gramas como límite, es decir, se establece un número máximo de palabras que pueden tener las colocacones extraídas.

3.10. Herramientas utilizadas para el desarrollo del proyecto

Para llevar a cabo este proyecto se utilizaron diversas herramientas orientadas al procesamiento del lenguaje natural y desarrollo web.

- Lenguaje de programación Python: Seleccionado por sus amplias opciones de librerías en al ámbito del Procesamiento del Lenguaje Natural que además incluye algunas que utilizan el idioma español.
- Librería Spacy-Stanza: Es desarrollada y mantenida por la universidad de Stanford, la cual permite etiquetar textos en diversos idiomas, para este proyecto en específico en Español.
- Flask: Es una librería utilizada para realizar una conexión cliente-servidor, y fue seleccionada debido a que es muy ligera comparada con frameworks como Django.
- Pickle : Permite almacenar en archivos binarios diccionarios que pueden tener un gran tamaño lo que facilita su carga en memoria.
- Pyqt5: permite tener interfaces de escritorio modernas , además de que se pueden ejecutar en cualquier sistema operativo de escritorio.

Capítulo 4

Diseño

Usando los conceptos teóricos y el estudio del estado del arte, se proponen soluciones para la realización de diversos tipos de consultas mediante un buscador, además de la extraccción automática de colocaciones y locuciones.

Por lo tanto este capítulo se divide en dos secciones: una enfocado en la realización de consultas con sus diversos tipos y posteriormente se profundiza más en las técnicas de extracción de colocaciones.

4.1. Consultas del buscador

Las búsquedas o consultas son una parte fundamental de los buscadores modernos. La información tiene que ser indexada incluyendo las palabras cerradas según Avishek et al. [3]. A partir de esto debe permitirse al usuario realizar consultas por palabras o por frases en las que se puedan incluir las opciones de búsqueda por lemas y su versión truncada.
Las consultas pueden realizarse sobre tres corpus que fueron seleccionados porque se habian realizado búsquedas independientes con cada uno: uno que contiene obras literarias proporcionado por la universidad de Alicante, otro que contiene chistes y un corpus sobre las leyes universitarias de la BUAP.

4.1.1. Preprocesamiento del texto

El preprocesamiento del texto es una parte fundamental para este proyecto ya que en esta etapa se realizaron las siguientes tareas de limpieza del texto:

- Eliminación de signos de puntuación: Por medio de algunas expresiones regulares se removierón todos los signos de puntuación de las obras literarias ya que estos no son necesarios en el proceso de consultas de información.

- Tokenización: separar palabras del texto en entidades llamadas tokens, después de la eliminación de los signos de puntuación en este proyecto los tokens son principalmente palabras.

- Conversión de texto a minúsculas: Para que el texto sea más uniforme se decidió convertir todo el texto de las obras literarias a minúsculas.

4.1.2. Estructura para indexar la información

La búsqueda puede realizarse sobre tres corpus: uno que contiene obras literarias, otro que contiene chistes y un corpus sobre las leyes universitarias de la BUAP. Finalmente el resultado de la consulta muestra las siguientes características:

- Número de resultados obtenidos.

- textos en donde se encuentran las consultas con algunos metadatos como el título del texto. En el caso de las obras literarias también cuenta con el año de publicación, autor y año de publicación.

- Tiempo de ejecución de la búsqueda.

- Valor k: que permite definir la cercanía de las palabras, es decir, que se puede buscar una frase como ***la primera guerra***. que son palabras consecutivas. Sin embargo si utilizo un valor de k de tres esto permitirá obtener resultados que contienen la frase original y además resultados como ***la primera*** *de una* ***guerra*** , es decir hay mas flexibilidad en el número de palabras que puede haber entre la frase.

Para realizar consultas se procedió a indexar la información y para ello se estudiaron dos métodos.

- Matrices término-incidencia: presentan filas que corresponden a términos y columnas a documentos. Sin embargo su creación se vuelve menos eficiente al contar con un gran número de documentos por lo que el uso de memoria es excesivo [5].

- **Índices invertidos o índices invertidos posicionales:** es la estructura seleccionada para indexar todas las obras literarias del presente trabajo. Tienen un posting list que es una lista de documentos en donde aparece el término, generalmente ordenados por un ID de documento y almacenados en una forma de lista[5].

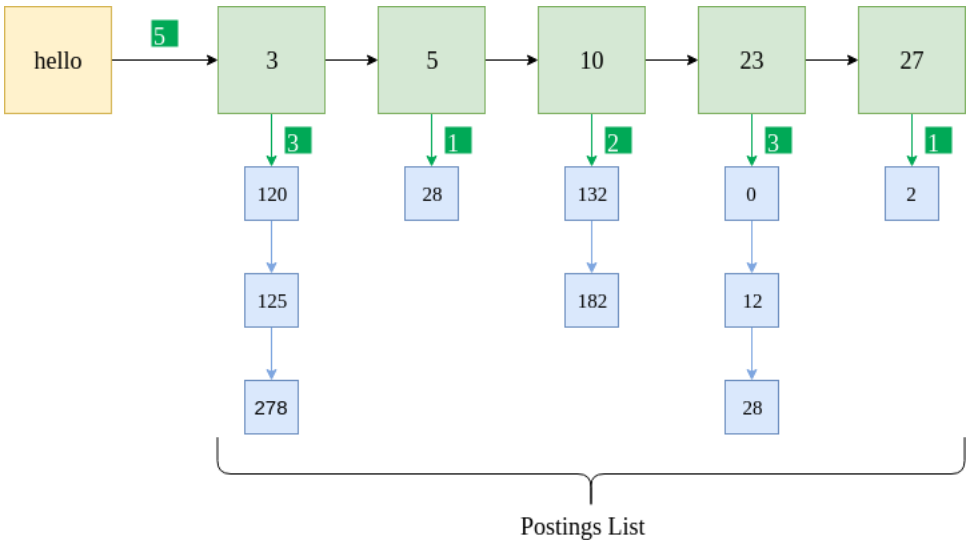

Figura 4.1: Estructura de un índice invertido posicional.

La figura anterior muestra la misma lista implementada para un índice posicional. Los recuadros azules indican la posición del término "*hello*" en los documentos correspondientes. Por ejemplo, "*hello*.ªparece en el documento cinco en tres posiciones: 120, 125 y 278. Además, la frecuencia del término se almacena para cada documento.
.

4.1.3. Consulta sobre índice posicional

Para realizar una búsqueda por frases es necesario el uso del índice invertido mencionado anteriormente. Se debe verificar que los términos se encuentren en el documento, así como sus posiciones de aparición y que estas sean compatibles con la frase consultada. En [7] se ejemplifica de la siguiente forma: suponiendo que la consulta es *"to be or not to be"*. Las listas para acceder son: to, be, or, not. Entonces se examina la intersección de las listas para *to* y *be*. Primero buscamos documentos que contengan ambos términos. Luego, buscamos lugares en las listas donde haya una ocurrencia de *be* con un índice de token uno más alto que una posición de *to*, y luego buscamos otra ocurrencia de cada palabra con un índice de token 4 más alto que la primera ocurrencia.

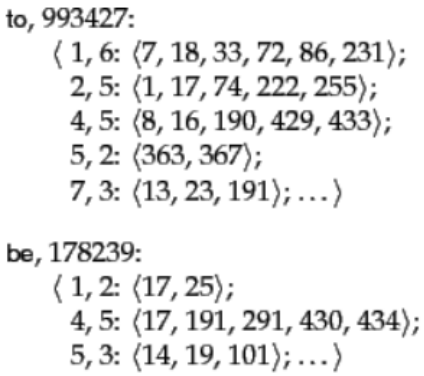

Figura 4.2: Ejemplo de índice posicional [7].

Por lo tanto para este ejemplo en particular ls resultados serián los siguientes:
to: {...; 4..,429, 433; ...}
be: {...; 4..,430, 434; ...}

4.1.4. Búsqueda por similitud de coseno

La similitud del coseno mide la similitud entre dos vectores de un espacio de producto interno. Se mide por el coseno del ángulo entre dos vectores y determina si dos vectores apuntan aproximadamente en la misma dirección [1]. Se representa de la siguiente forma:

$sim(x, y) = \frac{x.y}{|x||y|}$ Donde $|x|$ es la norma euclidiana del vector $x = (x_1, x_2, ..., x_p)$ definido como:
$\sqrt{x_1^2 + x_2^2 + ... + x_p^2}$
Conceptualmente, es el tamaño del vector. De forma similar $|y|$ es la forma euclidiana del vector y. La medida calcula el coseno del ángulo entre vectores x y y. Un valor de coseno de 0 significa que los dos vectores están a 90 grados cada uno. Cuanto más cercano sea el valor del coseno a 1, menor será el ángulo y mayor será la coincidencia entre los vectores. A continuación se muestra el siguiente ejemplo:

Document	team	coach	hockey	baseball	soccer	penalty	score	win	loss	season
Document1	5	0	3	0	2	0	0	2	0	0
Document2	3	0	2	0	1	1	0	1	0	1
Document3	0	7	0	2	1	0	0	3	0	0
Document4	0	1	0	0	1	2	2	0	3	0

Figura 4.3: Matriz ejemplo de término-frecuencia [1].

Suponiendo que se toman los dos primeros vectores de término-frecuencia como x y y.Se utiliza la fórmula para calcular la similitud coseno entre los dos vectores.
$x^t \cdot y = 5 \times 3 + 0 \times 0 + 3 \times 2 + 0 \times 0 + 2 \times 1 + 0 \times 1 + 0 \times 0 + 2 \times 1 + 0 \times 0 + 0 \times 1 = 25$
$|x| = \sqrt{5^2 + 0^2 + 3^2 + 0^2 + 2^2 + 0^2 + 0^2 + 2^2 + 0^2 + 0^2} = 6.48$
$|y| = \sqrt{3^2 + 0^2 + 2^2 + 0^2 + 1^2 + 1^2 + 0^2 + 1^2 + 0^2 + 1^2} = 4.12$
$sim(x, y) = 0.94$
Por lo tanto,estos documentos se considerarían bastante similares.

4.1.5. Tipos de consultas

Dentro del buscador el usuario puede realizar diferentes tipos de consultas:

- Normal: Se realiza la consulta por medio de una palabra o frase en el buscador y esta devuelve los documentos en donde se encuentra y muestra algunos metadatos relacionados .

- Por raíz: consiste en agrupar las palabras por su raíz (Unidad morfológica que no posee afijos, ni flexivos, ni derivativos). Para ello existe un indice posicional

que contiene todas las palabras en su forma truncada por ejemplo las palabras: trapo, trapero. trapear y trapito tienen como raíz ***trap*** .Este proceso realizado por medio de una función de la líbreria snowballStemmer. Es decir que el resultado de la consulta puede tener todas las palabras con la raíz ya mencionada.

- Por lema: La lematización es un proceso lingüístico que consiste en, dada una forma flexionada (es decir, en plural, en femenino, conjugada, etc), hallar el lema correspondiente. El lema es la forma que por convenio se acepta como representante de todas las formas flexionadas de una misma palabra. Es decir, el lema de una palabra es la palabra que nos encontraríamos como entrada en un diccionario tradicional: singular para sustantivos, masculino singular para adjetivos, infinitivo para verbos. Por ejemplo, *decir* es el lema de *dije*, pero también de *diré* o *dijeramos*; *guapo* es el lema de *guapas*; *mesa* es el lema de *mesas*.De esta forma los resultados de la búsqueda tendrán el lema de las palabras consultadas.

4.2. Extracción automática de colocaciones y locuciones

Las técnicas usadas para la extracción fueron mejoradas agregando patrones de las estructuras de colocaciones y locuciones que se desean obtener como los que se muestran a continuación:

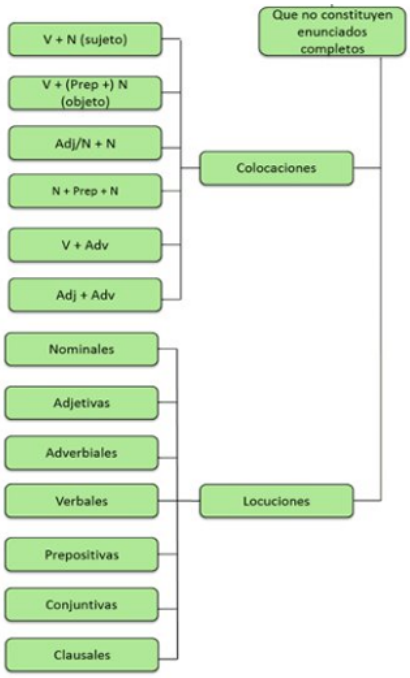

Figura 4.4: Unidades fraseológicas relevantes para el presente proyecto [6].

A partir de esta estructura básica se llevó a cabo el etiquetado por partes de la oración usando la librería Spacy lo que permite definir patrones específicos de las locuciones deseadas, las etiquetas usadas por esta librería son las que se muestran a continuación:

POS	DESCRIPTION	EXAMPLES
ADJ	adjective	*big, old, green, incomprehensible, first*
ADP	adposition	*in, to, during*
ADV	adverb	*very, tomorrow, down, where, there*
AUX	auxiliary	*is, has (done), will (do), should (do)*
CONJ	conjunction	*and, or, but*
CCONJ	coordinating conjunction	*and, or, but*
DET	determiner	*a, an, the*
INTJ	interjection	*psst, ouch, bravo, hello*
NOUN	noun	*girl, cat, tree, air, beauty*
NUM	numeral	*1, 2017, one, seventy-seven, IV, MMXIV*
PART	particle	*'s, not,*
PRON	pronoun	*I, you, he, she, myself, themselves, somebody*
PROPN	proper noun	*Mary, John, London, NATO, HBO*
PUNCT	punctuation	*., (,), ?*
SCONJ	subordinating conjunction	*if, while, that*
SYM	symbol	*$, %, §, ©, +, −, ×, ÷, =, :), 😍*
VERB	verb	*run, runs, running, eat, ate, eating*
X	other	*sfpksdpsxmsa*
SPACE	space	

Figura 4.5: etiquetas de la libreria Spacy.

Para obtener mejores resultados en la extracción de colocaciones y locuciones fue necesario utilizar patrones que definen diversas formas de unidades fraseológicas, estos patrones pueden ser modificados . Esto servirá como un filtro para solo obtener frases que contengan las estructuras deseadas, por ejemplo :

- PRON, VERB, NOUN, ADP, ADJ, NOUN : Me compré ropa de **segunda mano** (ropa usada-locución adjetiva).

- DET, NOUN, PRON, VERB, ADP, PROPN: El delincuente le disparó a **quemarropa** (directamente-locución adverbial).

- PROPN, VERB, ADP, DET, NUM, NOUN: David encontró a su **media naranja** (amor ideal- locución nominal).

- PRON, VERB, ADP, ADV, DET, DET, NOUN: Los **eché de menos** todos estos meses (extrañar- locución verbal).

- PROPN, AUX, DET, NOUN, ADP, DET, NOUN: Román es un hombre de **pocas palabras** (locución adjetiva).
- VERB, DET, NOUN, ADP, NOUN: Compró el auto **a hurtadillas** (locución adverbial).
- ADP, NOUN, ADP, SCONJ, VERB, ADV, ADJ, DET, NOUN: **A pesar de que** estudié mucho, reprobé el examen (locución conjuntiva)
- VERB, ADP, DET, NOUN, ADJ, ADP, NOUN, ADJ: Asesinó a una familia completa **a sangre fria** (locución adverbial)
- Lucía es una persona **difícil de leer** (locución adjetiva).

En conjunto con la definición de patrones, la búsqueda se realiza por n-gramas y para ellos se establece un límite que puede variar en este caso establecido en siete. La extracción de colocaciones y locuciones se lleva a cabo en una herramienta de escritorio con una interfaz que permite extraer colocaciones y locuciones automáticamente partir de la selección de un texto y esta se divide en cuatro secciones:

- Preprocesamiento: En donde se seleciona un texto y se lleva a cabo la limpieza de este , este es explicado en la sección anterior.
- Extracción de terminología: En esta sección se inicia la extracción automática, una vez terminada se pueden seleccionar los métodos de frecuencia, t-test, x^2 y PMI explicados en el marco teórico.
- Etiquetado PoS: Es una sección experimental en donde se pueden ingresar frases y como resultado obtendremos su etiquetado.
- Búsqueda: Se puede realizar una búsqueda de una frase en específico lo que devolverá la posición del texto en que se encuentra y un fragmento del texto.

Capítulo 5

Resultados

En este capítulo se muestran los resultados obtenidos al realizar consultas sobre el buscador y sobre la extracción automática de colocaciones y locuciones.

5.1. Consultas en el buscador

El buscador esta conformado por los siguientes elementos:

- Barra de búsqueda: Permite al usuario ingresar consultas por palabra o frase.
- Tipo de corpus: Permite seleccionar entre tres corpus: Literario, chistes y leyes de la BUAP.
- Tipo de consulta: Permite seleccionar el tipo de búsqueda: normal , por raíz,la consulta es truncada en automático y busca los resultados sobre un índice en donde se encuentran todas las palabras en la misma forma o por lemas. Ocurre algo similar que con el truncamiento, pero en este caso la consulta es lematizada y se buscan los resultados en un índice donde todas las palabras han sido lematizadas.
- valor k : Permite establecer una distancia entre palabras, si la ocnsulta es *la guerra* y el valor es de uno se mostrarán resultados donde esas palabras se encuentren consecutivas, pero si cambia el valor de k a dos se puede obtener el resultado de *la guerra* y algún otro como *la primera guerra*.

- Botón buscar: el cual da inicio a la búsqueda.

- Sección de resultados: la cual muestra todos los documentos en donde se localiza la consulta realizada, además del número de resultados, tiempo de ejecución, el tipo de búsqueda realizada, que corpus se utilizó, un fragmento de texto donde aparece la consulta y algunos metadatos.

Figura 5.1: Interfaz del buscador Alisearch

A continuación se muestra la estructura de los resultados de las consultas:

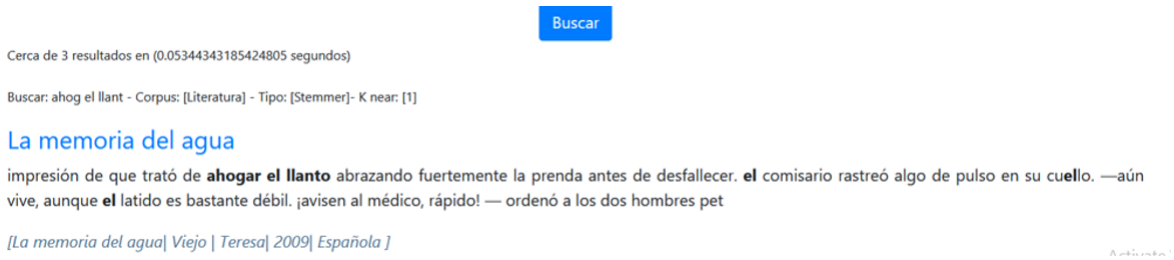

Figura 5.2: Estructura de los resultados en la interfaz

Más ejemplos de resultados de diversas consultas:

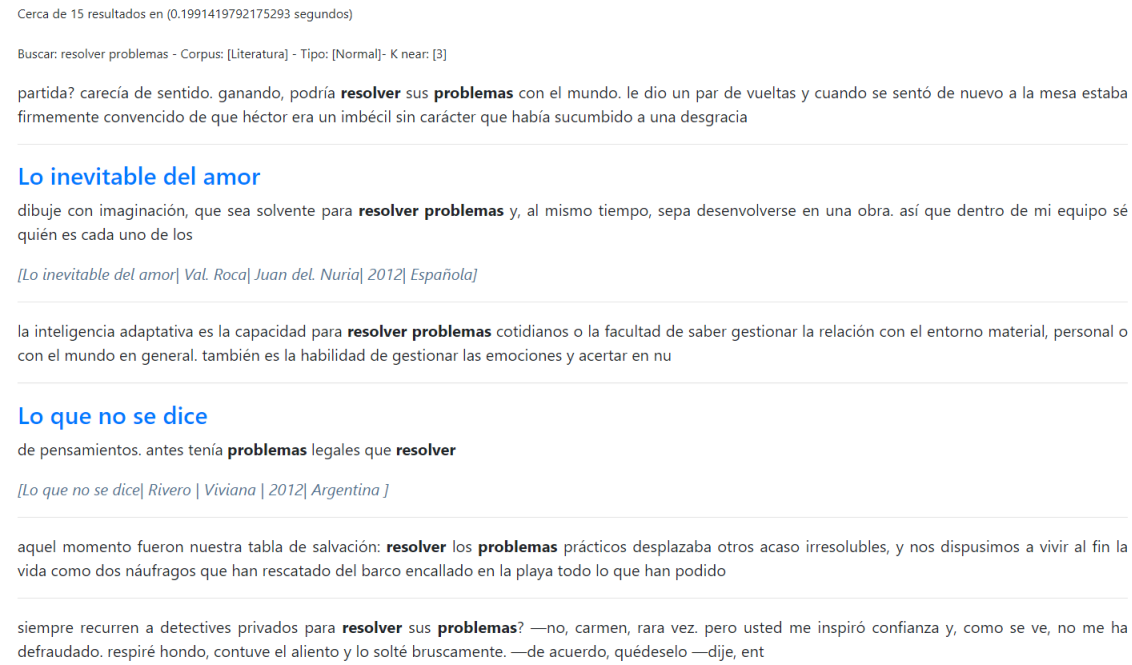

Figura 5.3: se obtuvieron 15 resultados en el corpus de literatura con un tipo de búsqueda normal y el valor de $k = 3$

Figura 5.4: se obtuvieron 5 resultados en el corpus de leyes universitarias con un tipo de búsqueda normal y el valor de $k = 3$

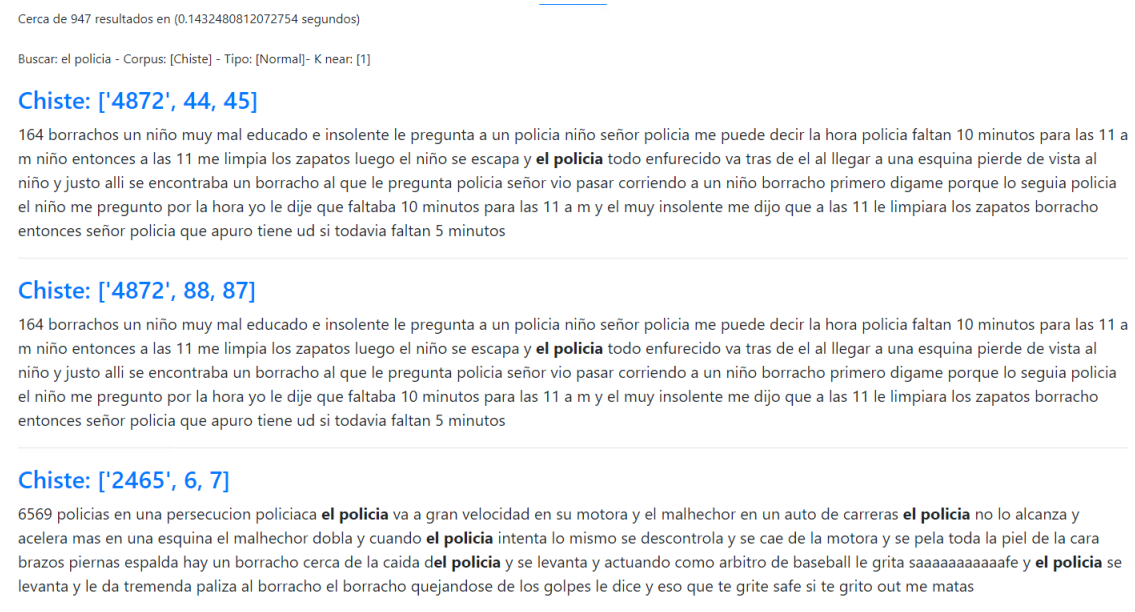

Figura 5.5: se obtuvieron 947 resultados en el corpus chistes con un tipo de búsqueda normal y el valor de $k = 1$

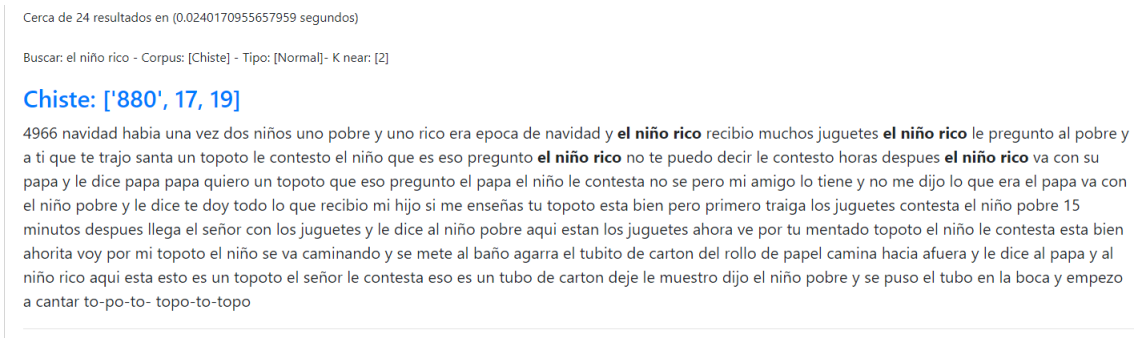

Figura 5.6: se obtuvieron 24 resultados en el corpus chistes con un tipo de búsqueda normal y el valor de $k = 2$

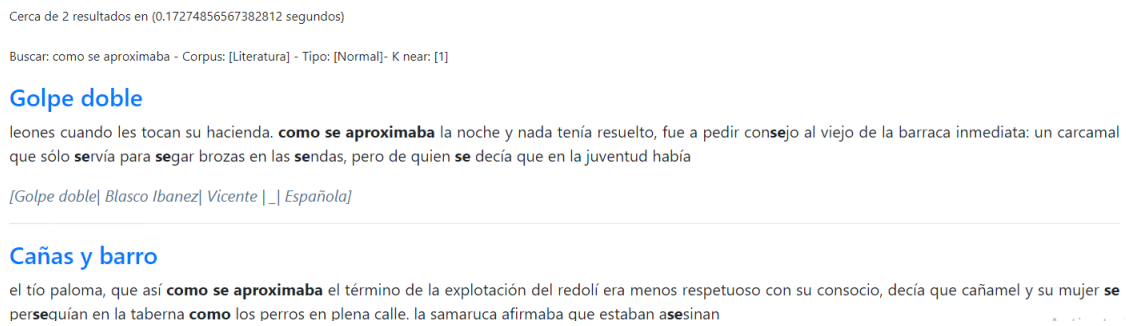

Figura 5.7: se obtuvieron 2 resultados en el corpus literatura con un tipo de búsqueda normal y el valor de $k = 1$

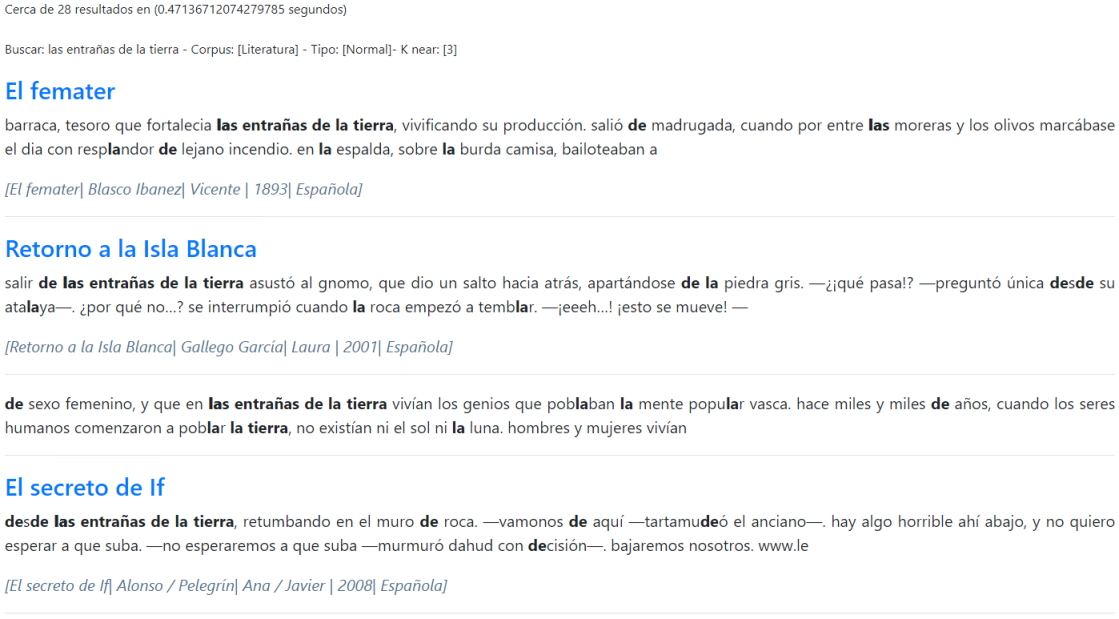

Figura 5.8: se obtuvieron 28 resultados en el corpus literatura con un tipo de búsqueda normal y el valor de $k = 3$

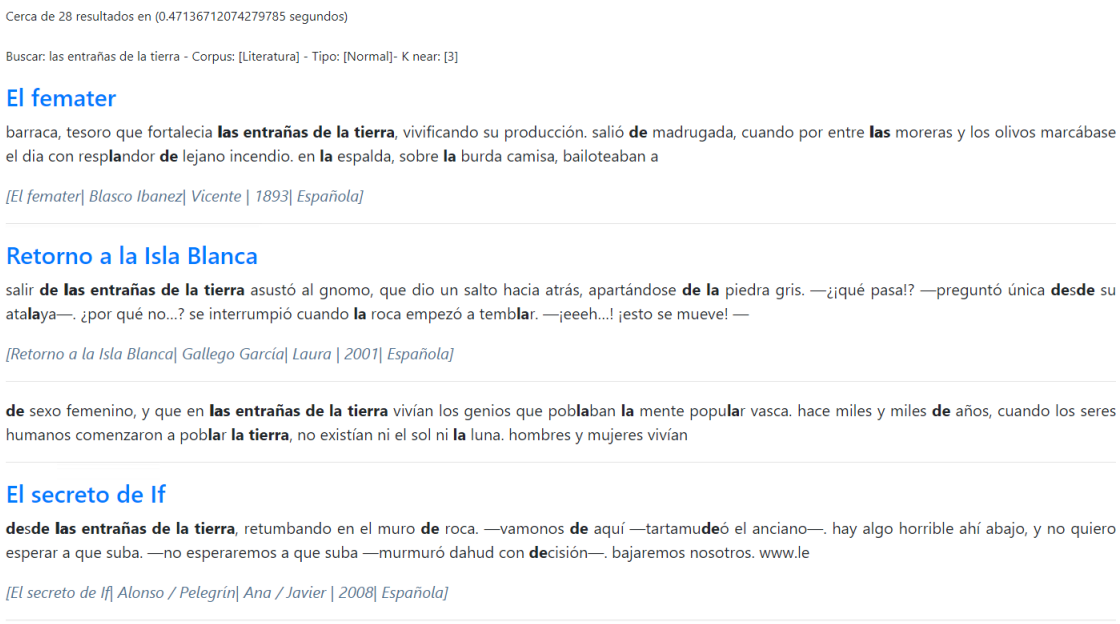

Figura 5.9: se obtuvieron 28 resultados en el corpus literatura con un tipo de búsqueda por lemas y el valor de $k = 3$

5.2. Extracción de colocaciones y locuciones

Como se explicó en el capítulo de Diseño, la extracción automática de locuciones se realiza con una interfaz de escritorio que permite seleccionar algun texto de forma local, preprocesarlo y posteriormente mostrar los resultados obtenidos. Además estos resultados se almacenan en un diccionario y en un archivo de texto para que puedan ser consultados o almacenados posteriormente.

5.2.1. Preprocesamiento

En esta sección se puede seleccionar un texto almacenado localmente para que se lleve a cabo su limpieza (eliminar, signos de puntuación, convertir a minúsculas, tokenizar) además muestra el idioma del texto, el número de tokens y el peso del archivo. Para este ejemplo, se utilizó "El Quijote de la mancha" en formato de texto plano.

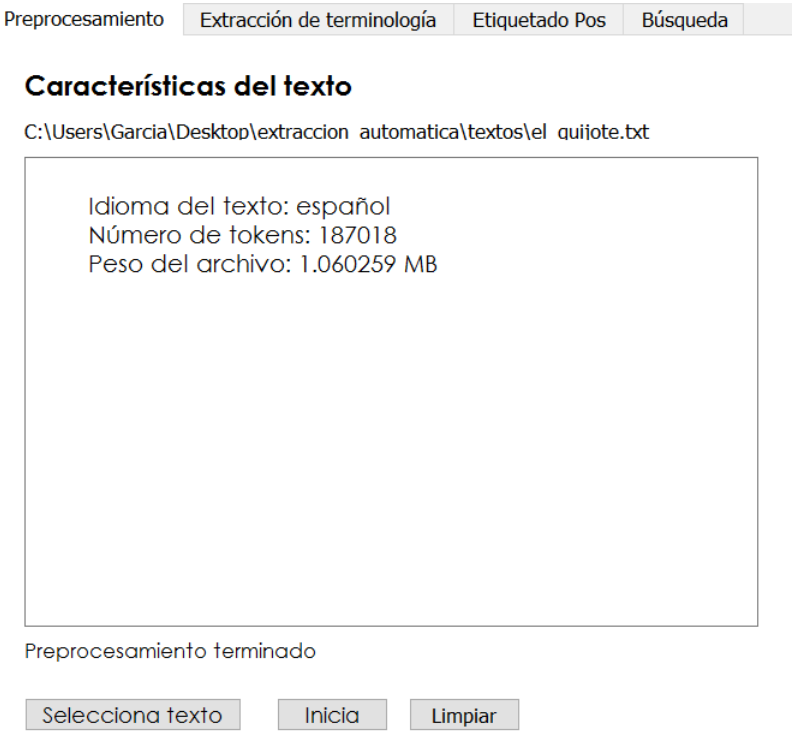

Figura 5.10: sección de preprocesamiento del tetxo

5.2.2. Extracción de terminología

Como se explico en el diseño, hay cuatro métodos utilizados para la extracción de terminología en complemento con una serie de patrones predefinidos que permiten seleccionar la estructura de las colocaciones y locuciones al seleccionar el método, este despliega los resultados obtenidos.

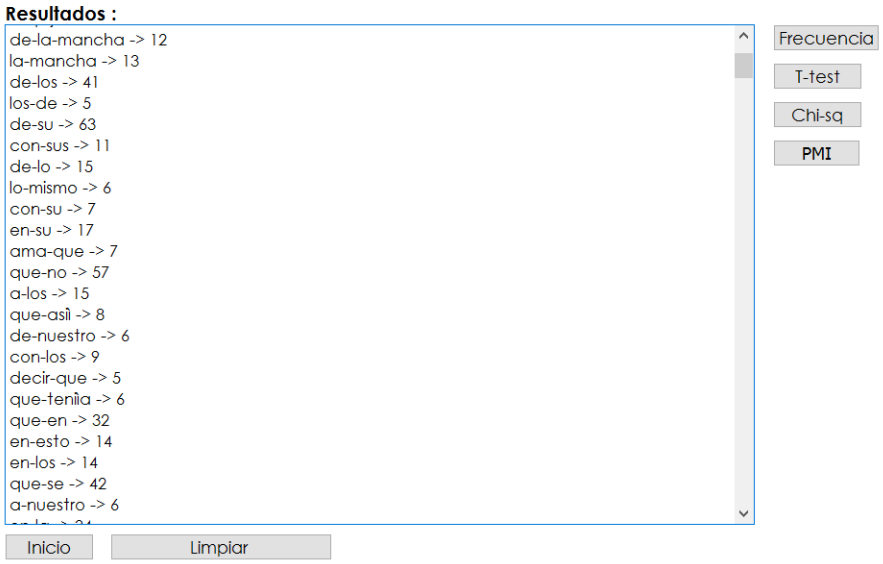

Figura 5.11: resultados por frecuencia con texto de "El Quijote de la mancha"

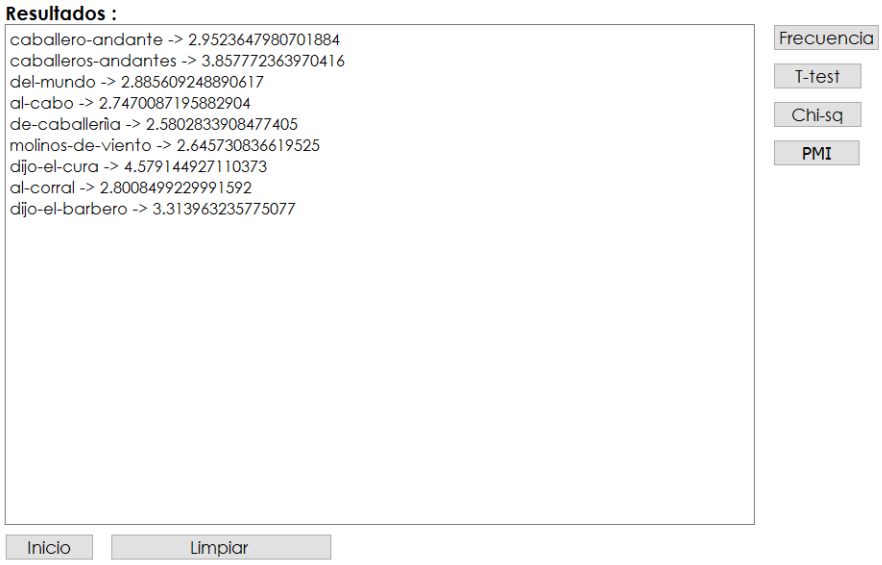

Figura 5.12: resultados con técnica de T-test con texto de "El Quijote de la mancha"

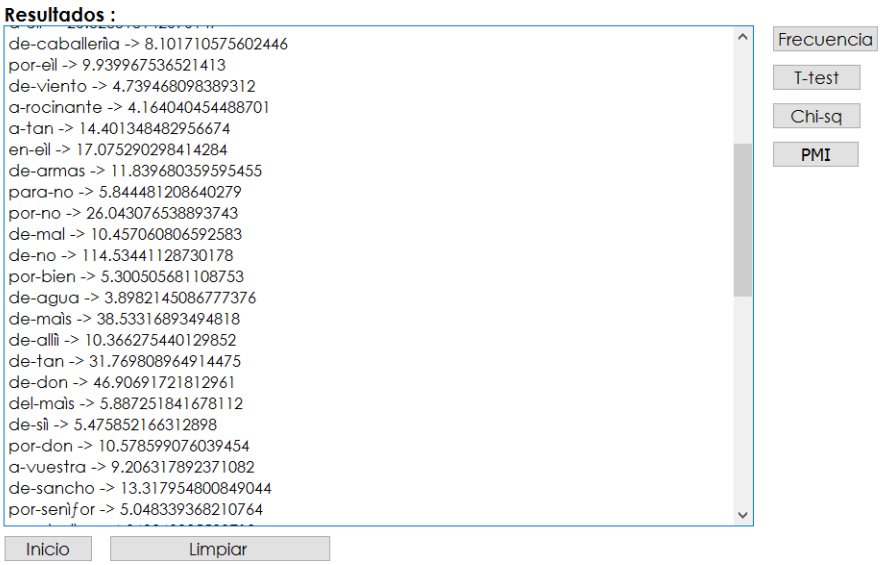

Figura 5.13: resultados con técnica de x^2 con texto de "El Quijote de la mancha"

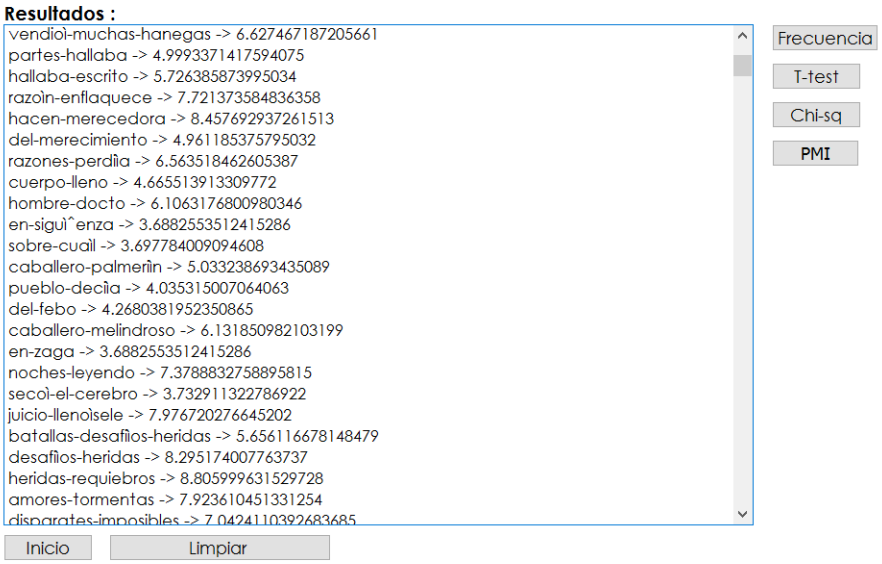

Figura 5.14: resultados con técnica de x^2 con texto de "El Quijote de la mancha"

5.2.3. Tabla comparativa de las cuatro técnicas utilizando bigramas

Column1	Frequency With Filter	PMI With filter	T-test With Filter	Chi-Sq Test with filter
0	('buenos', 'aires')	('abierta', 'inteligencia')	('buenos', 'aires')	('abierta', 'inteligencia')
1	('preguntó', 'si')	('padrenuestro', 'e')	('ramos', 'mejía')	('ochocientos', 'setenta')
2	('sido', 'tropero')	('otras', 'partes')	('sido', 'tropero')	('ocurrido', 'anoche')
3	('evangelio', 'según')	('novela', 'reciente')	('según', 'marcos')	('noches', 'rezara')
4	('según', 'marcos')	('nos', 'parece')	('evangelio', 'según')	('ni', 'siquiera')
5	('baltasar', 'espinosa')	('mientras', 'leía')	('preguntó', 'si')	('mejores', 'jinetes')
6	('ramos', 'mejía')	('mala', 'memoria')	('baltasar', 'espinosa')	('mal', 'jugador')
7	('cuarenta', 'pesos')	('algo', 'decidió')	('jorge', 'luis')	('tendió', 'junto')
8	('isla', 'querida')	('universitaria', 'abundaba')	('primeras', 'gotas')	('sastrería', 'prefirió')
9	('bajel', 'perdido')	('sastrería', 'prefirió')	('primera', 'vez')	('repitió', 'gutrel')
10	('llevaba', 'consigo')	('relatos', 'eróticos')	('prefirió', 'quedarse')	('rastros', 'oscuros')
11	('consigo', 'ahora')	('siesta', 'larga')	('podían', 'importar')	('algo', 'decidió')
12	('dos', 'historias')	('interlocutor', 'tuviera')	('plumas', 'veneraba')	('cuantos', 'relatos')
13	('sangre', 'pensó')	('cuarenta', 'pesos')	('perdidos', 'mientras')	('carne', 'asada')
14	('inmediatamente', 'acatadas')	('altos', 'fuertes')	('partes', 'creyeran')	('averiguar', 'juró')
15	('siempre', 'dos')	('cintita', 'celeste')	('palabras', 'concluido')	('analfabetos', 'desgraciadamente')
16	('repetido', 'siempre')	('averiguar', 'juró')	('ocurrido', 'anoche')	('amainado', 'volvió')
17	('daba', 'órdenes')	('analfabetos', 'desgraciadamente')	('páginas', 'finales')	('altos', 'fuertes')
18	('órdenes', 'tímidas')	('amainado', 'volvió')	('natural', 'complacencia')	('inevitable', 'leyó')
19	('muchacha', 'mimaba')	('gente', 'guardaba')	('mil', 'ochocientos')	('hubieran', 'contestado')

Figura 5.15: Resultados de las cuatro técnicas usando bigramas

Utilizando solo bigramas para la extracción se obtienen resultados similares, sin embargo en los ejemplos anteriores se puede notar que la técnica de frecuencia es la menos eficiente ya que obtiene las frases menos interesantes.

5.2.4. Etiquetado PoS

En esta sección se pueden observar los experimentos realizados con diversas frases y conocer el etiquetado de cada palabra y asi poder formar patrones nuevos, por ejemplo se puede utilizar la frase *tomar el toro por los cuernos*:

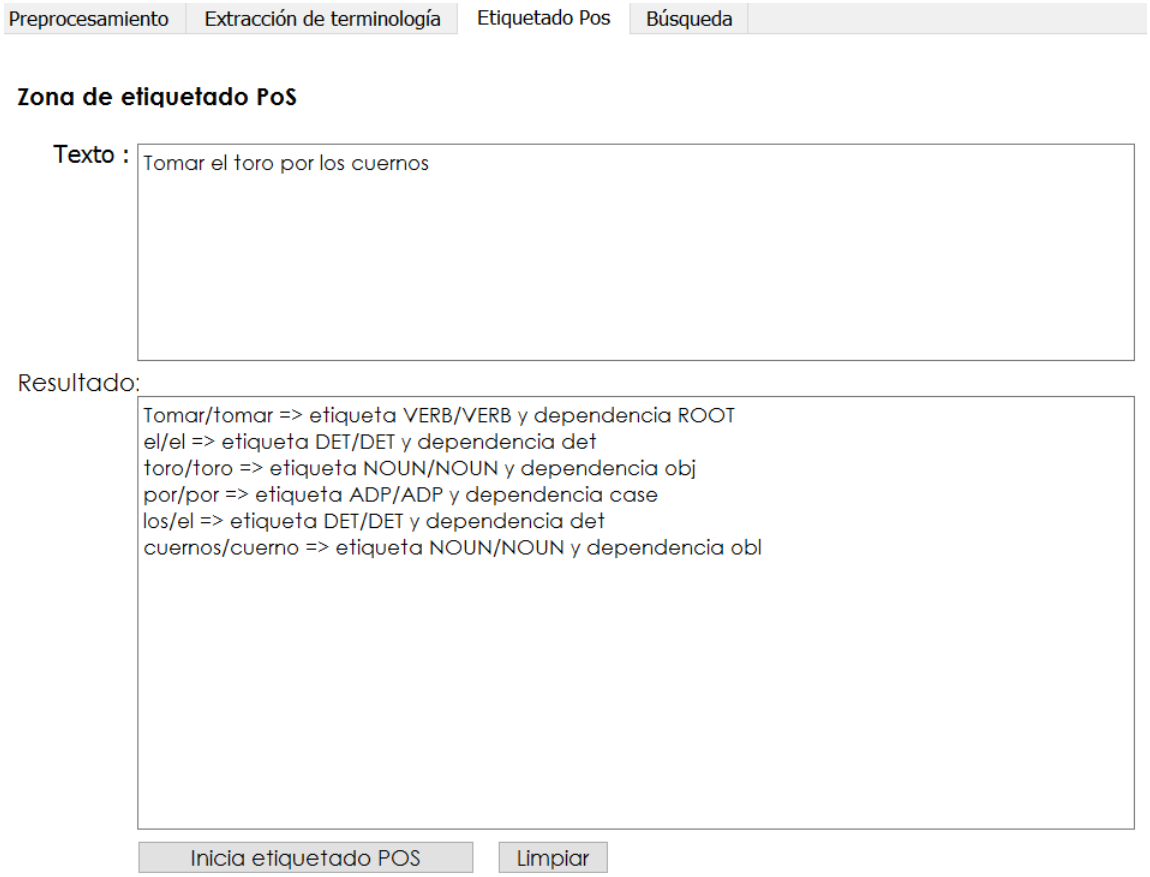

Figura 5.16: sección de extracción de etiquetado Pos

como se observa en la imagen, al oración fue dividida en tokens y se obtuvo la etiqueta correspondiente de cada palabra en el caso de *tomar* se clasificó como verbo, *el* como determinante, *toro* como sustantivo, *por* como adposición, *los* como determinante y *cuernos* como sustantivo.

5.2.5. Búsqueda

La ultima sección de esta herramienta es para realizar una búsqueda simple, no es tan robusta como el buscador pero puede servir como referencia, además es posible en trabajos futuros implementar las funciones de esta herramienta en el buscador implementado.

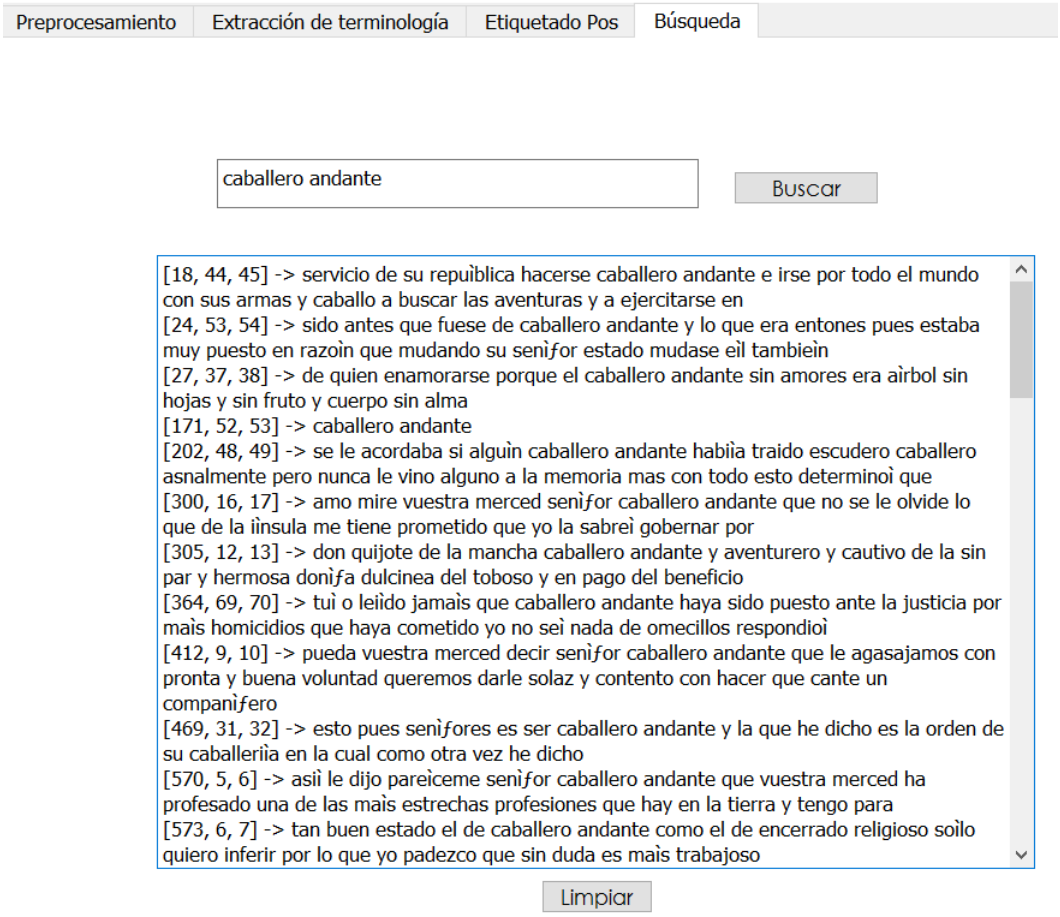

Figura 5.17: sección de búsqueda

En los resultados se nos muestra en que línea y posición se encuentra localizada la frase, también muestra un fragmento pequeño de texto para dar contexto a la oración. Sin embargo esta búsqueda es más básica que la presentada por el buscador por lo que solo funciona como un complemento en esta interfaz de escritorio.

Conclusiones

La recuperación de información implementada en este proyecto permitió crear un índice extenso a partir de textos en diferentes formatos, esto permite consultar información rapidamente a través de ellos. Una tarea futura después de esta implementación es almacenar la información en una base de datos relacional utilizando un manejador como PostgreSQL para poder acceder a ellos de manera más eficiente y que el proyecto pueda escalar de forma más estructurada.

En este trabajo de investigación se presentaron dos herramientas un buscador y una herramienta para la extracción automática de colocaciones y locuciones. La primera nos permitió realizar diversos tipos de consultas y obtener resultados en pocos segundos a partir de un índice invertido posicional que puede seguir creciendo para futuras investigaciones con corpus de diversos tipos. La segunda herramientas permitió la implementación de diversas técnicas para la extracción de colocaciones y locuciones. Sin embargo los resultados obtenidos aunque son extrídos de manera automática son demasiado mecanizados debido al uso de patrones por lo que muchos de los resultados no son interesantes, por ello es necesario implementar nuevas técnicas usando algoritmos de aprendizaje automático que nos permitan obtener resultados relevantes.

Con las técnicas implementadas en este trabajo se podrán comparar los resultados futuros con las técnicas de aprendizaje automático.

Finalmente el uso de Python y sus librerías como herramientas de implementación permitieron un desarrollo ágil de las ambas aplicaciones sobre todo en el buscador ,permitiendo crear un modelo cliente-servidor de manera muy rápida, además el uso de este lenguaje permitirá agregar nuevos modulos y librerías relacionadas al eprendizaje automático que serán totalmente compatibles con todas las herramientas

ya mencionadas.

Bibliografía

[1] *Getting to Know Your Data*. Elsevier, 2012. doi:10.1016/b978-0-12-381479-1.00002-2.

[2] Rafael Muñoz Antonio Guillén, Yoan Gutierrez. A document profile for improving information retrieval systems. *Research in computer Science*, págs. 29–40, 2017.

[3] Srikanta Bedathur Klaus Berberich Avishek Anand, Ida Mele. Phrase query optimization on inverted indexes. *CIKM Proceedings of the 23rd ACM International Conference on information and knowledge*, 17:1807–1810, 2014.

[4] Tolosa Chacón Gabriel Bordignon Fernando. Recuperación de información: un área de investigación en crecimiento. *Ciencias de la información*, 38:13–24, 2007.

[5] Manjula K. Shenoy Chiranjeevi H. Advanced text documents information retrieval system for search services. *Cogent Engineering*, 2020. doi:10.1080/23311916.2020.1856467.

[6] Hinrich Schütze Christopher D. Manning. *Foundations of Statistical Natural Language Processing*. The MIT Press, 1999. ISBN 978-0262133609.

[7] Hinrich Schütze Christopher D. Manning, Prabhakar Raghavan. *Introduction to information retrieval*. Cambridge, 2008. ISBN 0521865719.

[8] Jose Luiz de Lucca. Detección y extracción de unidades fraseológicas a partir de un corpus textual. *A survey of corpus-based research*, págs. 664–674, 2009.

[9] María Isabel Diéguez M. Aciertos y errores en la traducción automática: metodología de la enseñanza-aprendizaje de la traducción humana. *Onomázein*, 2001. ISSN 0717-1285. URL https://www.redalyc.org/articulo.oa?id=134518177011.

[10] Manuel Marcos Aldón Fuensanta M. Guerrero Carmona. Comparación entre extracción de información y extracción de resúmenes. *Proceedings of the Annual Meeting of the Cognitive Science Society*, 10, 2013.

[11] Chacon T. Gabriel M. Recuperación de información un área de investigación en crecimiento. 38:13–24, 2007. URL http://www.redalyc.org/articulo.oa?id=181414865002.

[12] Sergio Suárez Guerra José Luios Oropeza Rodriguez. Algoritmos y métodos para el reconocimiento de voz en español mediante sílabas. *Computación y sistemas*, 9:270–286, 2006.

[13] Alia Karim y Duaa Enteesha. Enhance inverted index using in information retrieval. *Engineering Technology Journal*, 34:302–310, 2016.

[14] Ekaterina Kochmar. *Getting started with Natural Language Processing*. Manning, 2022. ISBN 9781617296765.

[15] W. Bruce Croft Mark Sanderson. The history of information retrieval research. *ciir publications*, págs. 367–375, 2005.

[16] Lic Jaime Pariona Quispe Mg. Augusto Cortez Vázquez, Mg. Hugo Vega Huerta. Procesamiento del lenguaje natural. *RISI*, págs. 45–54, 2009.

[17] Robert C. Moore y Chris Quirk. *Improved Smoothing for N-gram Language Models Based on Ordinary Counts*. Association for Computational Linguistics, Suntec, Singapore, 2009. URL https://aclanthology.org/P09-2088.

[18] Muhammad Imran Rafique y Mehdi Hassan. Utilizing distinct terms for proximity and phrases in the document for better information retrieval. págs. 100–105, 2014. doi:10.1109/ICET.2014.7021024.

[19] Sulema Torres Ramos. Extracción automática de un diccionario de colocaciones en español. *Research in Computing Science*, 70, 2013.

[20] O. Ibrihich M. Esghir S. Ibrihich, A. Oussous. A review on recent research in information retrieval. *Elsevier B.V.*, 2022.

[21] Vera Demberg Silas Weinbach. Phrase query optimization on inverted indexes. *Proceedings of the Annual Meeting of the Cognitive Science Society*, 35, 2013.

[22] Perla Velasco-Elizondo Juan Villa-Cisneros Sandra Briceño-Muro Sodel Vazquez-Reyes, Maria de León-Sigg. The use of inverted index to information retrieval: Add intelligent in aviation case study. 2016. doi:10.1007/978-3-319-48523-2_.

[23] Aristomenis Thanopoulos, Nikos Fakotakis, y George Kokkinakis. Comparative evaluation of collocation extraction metrics. 2002. URL http://www.lrec-conf.org/proceedings/lrec2002/pdf/128.pdf.

[24] Augusto Cortez Vásquez. Recuperación de textos electrónicos mediante índices invertidos. *Perfiles de ingenieria*, 17:119–127, 2005.