UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería Informática

Extracción de información de imágenes de pasaportes a partir de los índices probabilísticos

Trabajo Fin de Grado

Grado en Ingeniería Informática

AUTOR/A: Dionís Ros, Alejandro

Tutor/a: Benedí Ruiz, José Miguel

Cotutor/a: Sánchez Peiró, Joan Andreu

CURSO ACADÉMICO: 2021/2022

Resum

Al llarg dels segles s'han produït grans quantitats de text manuscrit. Malgrat els innombrables esforços realitzats per a fer accessibles aquests documents, les imatges en brut són en gran manera inútils per al seu propòsit principal l'extracció d'informació continguda en el text de les imatges del document. A causa d'això, existeix un creixent interés en els mètodes automàtics que permeten als usuaris buscar informació textual en aquestes imatges.

Recentment, s'ha introduït un nou enfocament per a buscar paraules en col·leccions massives d'imatges històriques de documents manuscrits. Per a reduir el temps de cerca en la fase d'explotació, es proposa una solució en dues fases. En la primera fase ("offline"), es calculen les probabilitats a posteriori de les paraules, índexs probabilístics (IP), a partir del procés de reconeixement de text manuscrit. En una segona fase ("en línia"), els IP s'utilitzen per a la indexació i cerca de paraules en la col·lecció. Els IP són una representació extraordinàriament més compacta que les pròpies imatges i per tant és l'únic resultat que es pot mantindre.

En aquest treball es proposa la construcció d'un sistema que permeta l'extracció de la informació rellevant a partir dels IP obtinguts d'un corpus. El procés d'extracció de la informació es realitzarà mitjançant informació geomètrica continguda en els IP. El corpus està format per imatges extretes del control de passaports i data dels anys 30 del segle passat. Comprovarem el seu rendiment mesurant la taxa d'error comparant-ho amb la referència.

Paraules clau: Reconeixement de text manuscrit, Extracció d'informació, Índexs probabilístics

Resumen

A lo largo de los siglos se han producido grandes cantidades de texto manuscrito. A pesar de los innumerables esfuerzos realizados para hacer accesibles estos documentos, las imágenes en bruto son en gran medida inútiles para su propósito principal la extracción de información contenida en el texto de las imágenes del documento. Debido a ello, existe un creciente interés en los métodos automáticos que permitan a los usuarios buscar información textual en estas imágenes.

Recientemente, se ha introducido un nuevo enfoque para buscar palabras en colecciones masivas de imágenes históricas de documentos manuscritos. Para reducir el tiempo de búsqueda en la fase de explotación, se propone una solución en dos fases. En la primera fase ("offline"), se calculan las probabilidades a posteriori de las palabras, índices probabilísticos (IP), a partir del proceso de reconocimiento de texto manuscrito. En una segunda fase ("online"), los IP se utilizan para la indexación y búsqueda de palabras en la colección. Los IP son una representación extraordinariamente más compacta que las propias imágenes y por tanto es el único resultado que se puede mantener.

En este trabajo se propone la construcción de un sistema que permita la extracción de la información relevante a partir de los IP obtenidos de un corpus. El proceso de extracción de la información se realizará mediante información geométrica contenida en los IP. El corpus está formado por imágenes extraídas del control de pasaportes y data de los años 30 del siglo pasado. Comprobaremos su rendimiento midiendo la tasa de error comparándolo con la referencia.

Palabras clave: Reconocimiento de texto manuscrito, Extracción de información, Índices probabilísticos

Abstract

Large amounts of handwritten text have been produced over the centuries. Despite countless efforts to make these documents accessible, raw images are largely useless for their primary purpose of information retrieval given in the text of the document images. Due to this, there is a fast-growing interest in automatic methods that allow users to search for textual information in these images.

Recently, a new approach has been introduced to searching words in massive collections of historical handwritten document images. To reduce the search time in the exploitation phase, a two-phase solution is proposed. In the first phase ("offline"), the posterior probabilities of the words, probabilistic indices (PI), are calculated from the handwritten text recognition process. In a second phase ("online"), the IPs are used for indexing and searching for words in the collection. The IPs are an extraordinarily more compact representation than the images themselves and therefore it is the only result that can be maintained.

This work proposes the construction of a system that allows the extraction of relevant information from the IPs obtained from a corpus. The information extraction process will be carried out using geometric information contained in the IPs. The corpus is made up of images taken from passport control and dates back to the 1930s. We will check its performance by measuring the error rate comparing it with the reference.

Key words: Handwritten text recognition, Information retrieval, Probabilistic indexes

Índice general

Índice de figuras

Índice de tablas

CAPÍTULO 1
Introducción

En la actualidad, contamos con cantidades enormes de imágenes de documentos manuscritos, es decir aquellos "documentos que contienen información escrita a mano en un soporte flexible y manejable (por ejemplo: el papiro, el pergamino o el papel), con materias como la tinta de una pluma, de un bolígrafo o de un lápiz de grafito" [1].

Estos documentos han servido a lo largo de la historia para agrupar, conservar y transmitir conocimientos o relatos a lo largo del paso del tiempo. Existen documentos de todo tipo, registros de nacimientos, muertes o bodas, documentos notariales, cartas, registros de viaje y de fronteras o cuadernos de bitácora.

Hoy en día contamos con cantidades inmensas de documentos manuscritos, repletos de información de alto valor pero cuyo acceso no es nada sencillo.

El hecho de procesarlos nos aportaría información muy valiosa sobre distintos temas como podrían ser cambios sociales o económicos, evolución del clima a lo largo de los años, movimientos comerciales, migraciones, entre muchos otros. Este es el motivo principal por el que las técnicas de transcripción automática de imágenes y extracción de información generan gran interés en la comunidad.

La extracción de información para corpus textuales está prácticamente resuelta, sin embargo no podemos decir lo mismo de los corpus de imágenes sin transcribir, siendo de este último tipo gran parte de los corpus masivos con los que contamos.

Desafortunadamente, la obtención del contenido de estas imágenes sin transcribir es inaccesible. Hasta el momento, esta transcripción se realizaba manualmente por paleógrafos, expertos altamente cualificados en la transcripción de documentos antiguos. Esta reproducción, además de ser muy complicada es muy costosa en cuanto a tiempo y dinero y, debido a esto, solamente se transcribían pequeños conjuntos de documentos ya que para grandes colecciones era inviable.

Hoy en día, con los avances en tecnologías como el HTR (del inglés, *Handwritten Recognition*) es posible la transcripción de estos documentos de manera automática, a partir de su digitalización. Por desgracia, la tasa de error de estos sistemas sigue siendo más alta de lo esperado siendo este el motivo por el que se descarta su uso.

Recientemente se ha introducido una nueva tecnología para abordar el problema de la extracción de información en conjuntos masivos de imágenes sin necesidad de transcribir las mismas. Esta tecnología se le conoce como índices probabilísticos y se busca realizar la extracción de información sobre estos índices.

La forma de obtener estos índices es la siguiente, para cada página del conjunto de imágenes aplicamos técnicas de HTR y obtenemos como resultado un grafo multietapa, denominado *trellis*. El *trellis* representa todas las posibles interpretaciones asociadas con la imagen, es por esta razón por la cual no es posible almacenarlo puesto que, por el gran tamaño del corpus, es muy pesado. A partir de este *trellis* se obtiene un grafo de palabras o WG y, a partir de este, los índices probabilísticos que son una representación mucho más ligera y es finalmente lo que se guarda para toda la colección.

La aplicación de esta nueva tecnología está obteniendo unos resultados muy alentadores y se actualmente se hace uso de esta en grandes colecciones de muchos archivos nacionales y privados.

1.1 Motivación

La utilidad de la extracción de información en textos manuscritos, reside en la recopilación y almacenamiento de la información contenida, con el fin de mejorar la accesibilidad a esta. Además, permite elaborar búsquedas cuya finalidad sea comprender y analizar hechos sociales, demográficos o económicos, así como inspirar nuevos temas de estudio en dichas áreas.

Disponemos de una colección de imágenes de documentos, asociados con los trámites de los pasaportes, del consulado de España en Buenos Aires en los años 30, conocida como "Libros de registros. Índice de pasaportes (1933-1938)", perteneciente a los Archivos Nacionales de España.

Además, disponemos de los índices probabilísticos correspondientes a cada uno de los documentos mencionados anteriormente, los cuales contienen las pseudopalabras detectadas así como alguna información geométrica adicional y su relevancia.

Por último, contamos también con unos documentos en formato PAGE [2](del inglés, (Page Analysis and Ground-truth Elements), donde figura la transcripción real del documento y nos permitirá evaluar el sistema.

La idea principal del trabajo consiste en, empleando únicamente los índices probabilísticos, extraer la información asociada a cada pasaporte, clasificándola por categorías, para, de esta forma, regenerar el documento en un formato que facilite tanto el acceso como la búsqueda de información. Cabe la posibilidad que varias palabras coincidan en la misma región de la imagen, por este motivo tendremos en cuenta la relevancia de las mismas a la hora de presentarlas, permitiendo al usuario establecer un umbral específico

adecuado a sus necesidades.

1.2 Objetivos

Uno de los objetivos principales del trabajo consiste en aprender la tecnología de los índices probabilísticos, aplicados en el contexto de un problema de HTR (del inglés, *Handwritten Recognition*), así como familiarizarse con su software y sus demostradores. Además, debemos comprender el proceso de obtención de estos índices, así como conocer los métodos empleados para su obtención.

Otro de los objetivos consiste en construir un nuevo software capaz de extraer toda la información disponible en los índices probabilísticos, para cada uno de los pasaportes presentes en las imágenes, y que sea capaz de agrupar en la categoría correspondiente cada una de las pseudopalabras. La eficiencia del programa debe de estar garantizada mediante el uso de estructuras de datos convenientes, permitiendo procesar cientos de documentos en tiempos moderados.

A continuación, buscamos generar el resultado del proceso de extracción de información en un formato adecuado, que permita ser empleado por los demostradores del PRHLT.

Por último, comprobar, mediante un experimento controlado, la tasa de error obtenida respecto a la referencia.

1.3 Estructura de la memoria

Este documento se divide en un total de 6 capítulos, los cuales se distribuyen como sigue:

En el capítulo 2, se realizará una revisión al estado del arte de este trabajo, donde comentaremos las principales técnicas que se utilizan actualmente, además de comentar algunos trabajos de distintos autores donde se hayan empleado tecnologías similares.

Para finalizar el capítulo, presentaremos la solución que hemos implementado para abordar el problema.

Continuaremos con el capítulo 3, donde explicaremos el origen del corpus con el que hemos trabajado a lo largo del proyecto, comentaremos su estructura y analizaremos cuáles son las principales dificultades con las que vamos a encontrarnos a la hora de procesarlo.

Posteriormente, en el capítulo 4, se comentarán brevemente los aspectos teóricos en los que se basa este trabajo y se procederá a explicar de manera exhaustiva la solución implementada para abordar el problema de este trabajo. En este último punto se comentará la aplicación que se ha desarrollado, los problemas que han surgido durante su desarrollo y las soluciones propuestas para solventarlos.

A continuación, en el capítulo 5, se explicarán las métricas de evaluación escogidas y se comentarán los experimentos llevados a cabo para evaluar el rendimiento de la solución.

Por último, en el capítulo 6, se revisarán los objetivos planteados al comienzo del proyecto y se analizarán las metas alcanzadas al final del mismo, además de comentar posibles futuras líneas de trabajo derivadas de este proyecto. Para finalizar, propondremos unas posibles líneas de trabajo futuras y relacionaremos este proyecto con los conocimientos proporcionados por el grado.

CAPÍTULO 2
Estado del arte

En este apartado expondremos los métodos y los resultados obtenidos por otros autores que han tratado de resolver problemas similares al que se presenta en este proyecto.

El interés por el reconocimiento de textos manuscritos está en auge, tanto en su faceta académica como en la indústria, lo que conlleva que una gran cantidad de autores investigue acerca de nuevos métodos, más eficientes y eficaces, con el fin de digitalizar y extraer la información contenida en imágenes de texto manuscrito.

Tras años de investigación en el área del reconocimiento del habla, el uso de modelos ocultos de Markov (HMM) con mixturas gaussianas embebidas se ha establecido como un estándar. Sin embargo, en [7] se nos propone el uso de mixturas de Bernoulli con ventana deslizante, a lo largo del ancho de la imagen. Esta ventana nos permite extraer en cada punto el mayor contexto horizontal de la imagen.

Esto implica que, la probabilidad de que observemos un determinado vector característico D-dimensional viene dado por la función de probabilidad basada en mixturas de Bernoulli, siendo D el producto de la altura de la imagen binaria por el tamaño de la ventana deslizante.

En este trabajo se disponía del corpus IfN/ENIT, formado por nombres de ciudades tunecinas manuscritas en árabe. En cuanto a los resultados de los experimentos, los autores reportan haber conseguido resultados por encima de lo obtenido, hasta ese momento, con un modelo con una ventana deslizante, de anchura igual a 9, y 32 mixturas de Bernoulli; siendo el WER (del inglés, Word Error Rate) obtenido igual a 12.3 %, comparado al modelo empleado en [5] que obtiene un WER del 14.6 %.

En [4], se nos presenta una competición sobre métodos de reconocimiento y análisis de páginas. Los retos de esta competición comprendían las áreas de la segmentación de páginas, la detección de líneas y el reconocimiento de caracteres. En este trabajo se evalúan 6 métodos, 3 pertenecientes a participantes y el resto se presentan como modelos base.

El primer método participante fue el proporcionado por Google, que presenta un método enfocado exclusivamente al tercer reto, el reconocimiento de caracteres.

En este se emplea la API de Google Cloud Vision, la cual basa su proceso de reconocimiento de caracteres en 5 fases: detección de texto, identificación de la dirección, identificación de guiones, decodificación de líneas y análisis del diseño. Como salida, la API devuelve un archivo en formato JSON que incluye las palabras detectadas, información acerca de las *bounding boxes* y sus probabilidades.

El siguiente método es *KFCN*, presentado por Berat Kurar. Este método trata los retos de segmentación de páginas y detección de líneas. Para el primero emplea una *Fully Convolutional Network*, que consiste en una red neuronal, cuyas capas no se encuentran completamente conexas, capaz de llevar a cabo solamente operaciones de convolución. Esta red fue entrenada durante 2372 *patches*, usando una ventana deslizante de 800x800 píxeles con un tamaño de paso de 400 píxeles.

Una vez entrenado el modelo, predice el diseño de la página y recorta los párrafos detectados, para, mediante un método basado en suavizado Gaussiano anisotrópico.

El último método fue presentado por Hany Ahmed, de la empresa RDI. Este método esta enfocado a los retos 2 y 3. El sistema es capaz de extraer las líneas de una imagen mediante un algoritmo de segmentación de instancias, el cual precisa de las localizaciones y las clases de todas las líneas presentes en la imagen. Después de esto, comienza el reconocimiento línea por línea.

Los métodos base proporcionados son *Tesseract OCR* versiones 3.04 y 4.0, basado en LSTM (del inglés, Long Short-Term Memory); y *ABBYY FineReader Engine 11*.

Tras la evaluación de los distintos métodos se puede observar que, KFCN ha obtenido los mejores resultados para el reto de segmentación de páginas y el método presentado por RDI ha obtenido los mejores resultados para los otros dos retos.

En [3], se emplean redes convolucionales recurrentes junto con técnicas de aprendizaje transductivo para el reconocimiento de textos manuscritos en tibetano. La particularidad principal de esta solución se encuentra en que la red esté entrenada a partir de datos sintéticos y que, al utilizarse técnicas de aprendizaje transductivo, la red tiene accesibles para el entrenamiento tanto los datos de entrenamiento como los de test, aunque estos últimos estén sin etiquetar.

En [6] se propone un modelo basado en *filler HMM* y en *keyword HMM* para la detección de palabras clave, o *keyword spotting*, en los corpus “Cristo-Salvador” y “IAMDB”. Un modelo *filler HMM* está constituido por varios modelos HMM de caracteres, organizados de forma paralela.

En [8], se comentan las tecnologías empleadas para la obtención de los índices probabilísticos para la colección de registros de la parroquia de Passau. Para la construcción

de estos índices se emplean redes convolucionales y recurrentes y, una vez entrenadas, al aplicar el algoritmo de Viterbi se obtiene un *character lattice*. De este lattice se obtienen un conjunto de n mejores caminos, siendo este conjunto el índice probabilístico.

2.1 Propuesta

En este trabajo se propone una aplicación capaz de regenerar en formato texto los pasaportes de las imágenes proporcionadas, a partir de sus índices probabilísticos. Además permitirá al usuario personalizar los niveles de confianza de los resultados devueltos por el sistema, con el fin de poder ajustarse de manera concreta a sus necesidades.

Como resultado el programa almacenará, para cada uno de los pasaportes presentes en la imagen, un archivo en formato JSON conteniendo la transcripción del mismo con el umbral especificado previamente por el usuario. En este archivo encontraremos los términos del pasaporte agrupados según el campo del formulario al que pertenezcan.

El modus operandi para la extracción de información comienza con la división de la imagen con el fin de conseguir localizar cada uno de los pasaportes. A continuación, agruparemos los términos de los índices por pasaporte y los agruparemos por campos. Finalmente, se aplicará el umbral deseado por el usuario y se crearán los archivos JSON pertinentes.

CAPÍTULO 3
Corpus

En el capítulo que se nos presenta se introducirá el corpus disponible para la realización de este trabajo.

La colección de datos que emplearemos para realizar los experimentos está compuesta por imágenes pertenecientes a los Archivos Nacionales de España. Esta colección es conocida como "Libros de registros. Índice de pasaportes (1933 - 1938)" y consiste en los registros de pasaportes del Consulado de España en Buenos Aires, Argentina.

Figura 3.1: Imagen de ejemplo del corpus utilizado. Fuente: Archivos Nacionales de España.

Dentro de cada imagen (véase la figura 3.1) encontramos cuatro pasaportes, donde cada uno contiene alguna foto así como los distintos campos escritos a mano.

Los campos en los que se divide cada pasaporte son: número de orden, fecha, nombre y apellidos, localidad y provincia de nacimiento, edad, estado civil, profesión, residencia habitual, motivos del viaje y personas que le acompañan, lugares de validez, documentos

presentados, clase, derechos cobrados y observaciones.

Esta colección posee diferentes dificultades en diferentes niveles del procesado. Podemos encontrarnos las fotos tanto arriba como abajo del pasaporte, algunas líneas ocupan varias líneas o están escritas de manera que invaden otros campos del pasaporte. Asímismo, nos encontramos con una cantidad elevada de nombres propios y fechas y, además, al tratarse de textos manuscritos antiguos, se emplea el vocabulario y las abreviaturas típicas de la época.

También nos encontramos con un problema muy típico en el área del reconocimiento de texto manuscrito, los textos están escritos por varias personas cada una con estilos de escritura propios por tanto, nos encontramos con una alta variabilidad en el estilo de escritura.

Para la fase experimental del trabajo utilizaremos un conjunto transcrito de 99 imágenes del total de la colección, disponible tanto en formato PAGE como en índices.

PAGE es un *framework* para la representación de imágenes de páginas, basado en XML, que almacena información acerca de las características de la imagen, como pueden ser: los bordes de la imagen, distorsiones geométricas y sus correcciones, binarización, entre otros; además de la estructura del diseño de la hoja y el contenido de la página.

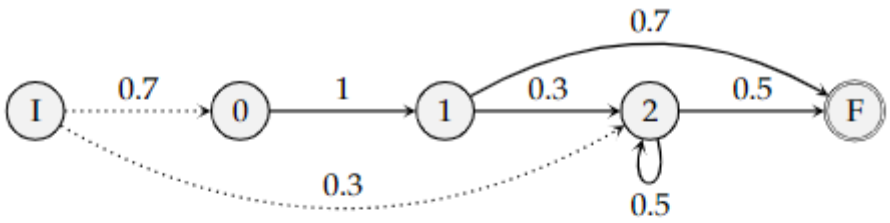
La colección ha sido utilizada y procesada por la compañía tranSkriptorium AI para el proyecto europeo European Digital Treasures en el cual participan los Archivos Nacionales de España [18].

CAPÍTULO 4
Metodología

En el capítulo que se presenta se comentarán algunos conceptos esenciales para comprender tanto el contexto de este trabajo así como la solución propuesta.

4.1 Reconocimiento de texto manuscrito

Un HMM [9, 10] (del inglés, *Hidden Markov Model*) es un modelo probabilístico para procesos estocásticos de Markov, es decir, sistemas cuyo estado en el instante de tiempo $t+1$ solo depende del estado en el instante t, y solamente se puede observar la cadena generada x puesto que la secuencia de estados visitados permanece oculta al observador.



$$Q = \{0, 1, 2, F\}$$
$$\Sigma = \{a, b\}$$

π	0	2
	0.7	0.3

A	0	1	2	F
0		1		
1			0.3	0.7
2			0.5	0.5

B	a	b
0	0.2	0.8
1	0.6	0.4
2	0.9	0.1

Figura 4.1: Ejemplo de HMM. Fuente: Elaboración propia.

Su definición formal es la siguiente (véase la figura 4.1), un HMM es un modelo $M = (Q, \Sigma, \pi, A, B)$ donde:

- Q es un conjunto finito de estados, incluyendo el estado final F.
- Σ es un conjunto finito de símbolos, también conocido como alfabeto.
- π pertenece $[0,1]^Q$ es un vector que incluye las probabilidades iniciales del sistema. Estas probabilidades han sido calculadas mediante ...
- A pertenece $[0,1]^{QXQ}$ es una matrix que incluye las probabilidades de transición del modelo.
- B pertenece $[0,1]^{Qx\Sigma}$ es una matrix que incluye las probabilidades de emisión del modelo.

Los HMM nos permiten calcular la probabilidad de que una cadena de caracteres, $x = x_1 x_2 \dots x_T$, sea generada por un modelo M.

$$P_M(x) = \sum_{\mathbf{q}=q_1 q_2 \dots q_T} P_M(x, \mathbf{q}) \tag{4.1}$$

donde:

$$P_M(x, \mathbf{q}) = [\pi_{q_1} B_{q_1,x_1}] \cdot [A_{q_1,q_2} B_{q_2,x_2}] \cdot \dots \cdot [A_{q_{T-1},q_T} B_{q_T,x_T}] \cdot A_{q_T,F} \tag{4.2}$$

Un modelo de lenguaje de n-gramas [11, 12] consiste en un modelo que es capaz predecir cuál es la siguiente palabra de un cadena teniendo en cuenta el contexto de la oración, a partir de los $n-1$ términos anteriores de la secuencia.

Para el proceso de reconocimiento de texto se ha empleado un sistema que consiste en HMM ópticos de caracteres y un modelo de lenguaje de n-gramas.

Este sistema de reconocimiento, dada una línea de texto manuscrito, representado como una secuencia de vectores característicos, $\mathbf{x}$, encuentran la secuencia de palabras más probable $\hat{w} = \hat{w_1}, \hat{w_2} \dots \hat{w_l}$, de acuerdo a:

$$\hat{\mathbf{w}} = \arg\max_{\mathbf{w}} P(\mathbf{w}|\mathbf{x}) = \arg\max_{\mathbf{w}} p(\mathbf{x}, \mathbf{w}) = \arg\max_{\mathbf{w}} p(\mathbf{x}|\mathbf{w}) \cdot P(\mathbf{w}) \tag{4.3}$$

Donde $p(\mathbf{x}|\mathbf{w})$ puede ser estimado mediante el modelo óptico, es decir mediante los HMM. También podemos aproximar $P(\mathbf{w})$ mediante el modelo de lenguaje de n-gramas.

Este proceso se realiza en tres niveles. En el primero, los HMM modelan el contorno de los caracteres. Seguidamente, se moldean las palabras mediante la unión de varios HMM, lo cual representa las posibles combinaciones de caracteres para formarlas. Por último, las líneas de texto se modelan mediante el modelo de lenguaje de n-gramas.

El modelo empleado es el utilizado en [13]. Con este modelo somos capaces de obtener tanto la mejor secuencia de palabras junto con su posible posición dentro de una línea dada x, además de obtener un grafo de palabras donde aparecerán gran cantidad de estas hipótesis.

4.2 Extracción de los índices probabilísticos

A continuación se muestra el proceso de obtención de los índices probabilísticos. Este proceso nos permitirá representar una cantidad de información elevada de gran tamaño mediante una estructura de datos, en este caso un grafo, de forma compacta reduciendo así su tamaño.

Para comenzar con el proceso de extracción debemos obtener las líneas de las imágenes que forman la colección. Este procedimiento está prácticamente resulto mediante el uso de técnicas de detección y segmentación de líneas, como las comentadas en [14].

Una vez finalizada la extracción de líneas procedemos a la obtención del *trellis*, un grafo multietapa. Este grafo representa todas las posibles interpretaciones asociadas con la imagen. Debido al gran tamaño de este y al volumen del corpus, es inviable almacenarlo por lo que, mediante el algoritmo de Viterbi, obtenemos un grafo de palabras.

Un grafo de palabras [15] se define como un grafo dirigido acíclico y etiquetado, donde cada arco tiene como etiqueta una palabra y una puntuación, y cada nodo un instante de tiempo.

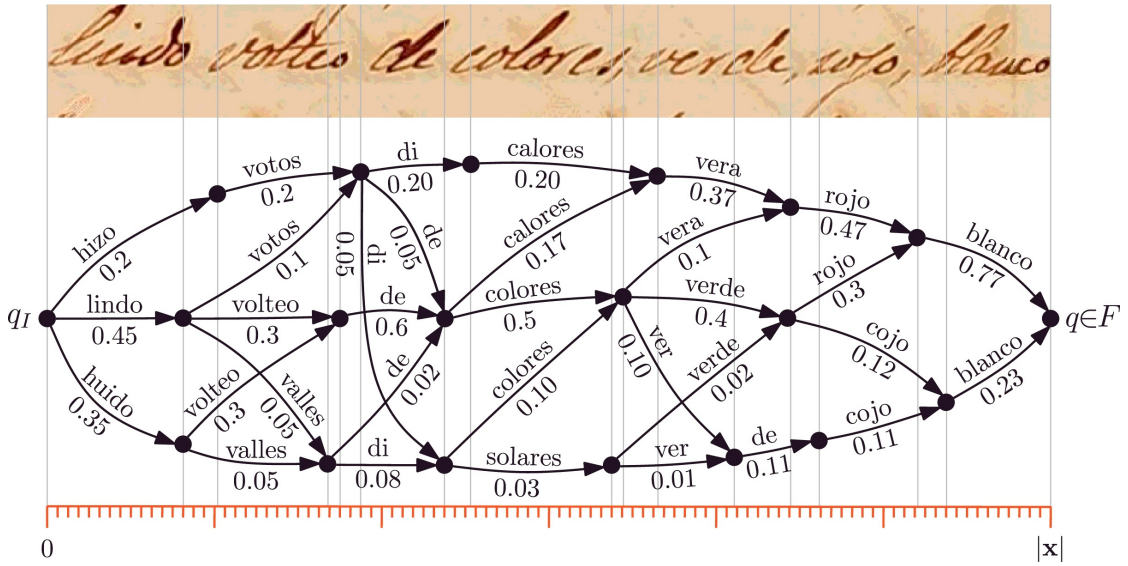


Figura 4.2: Ejemplo simplificado de un grafo de palabras normalizado. Fuente: [13]

El grafo (véase la figura 4.2), tiene un único nodo inicial en tiempo cero y un único nodo final. Las palabras, también denominadas pseudopalabras, no tienen por qué representar palabras reales, si no que representan las posibles cadenas de caracteres encontradas. Las pseudopalabras de los caminos formados entre los nodos inicial y final, forman frases hipotéticas. Tras la obtención de este grafo se normalizan las puntuaciones de los arcos, conocidas como la probabilidad a posteriori del arco.

Como podemos observar en la figura 4.2, cada nodo está situado en un instante de tiempo. En el caso del nodo inicial este se encuentra en el instante 0 y el final en el instante $|x|$, siendo $|x|$ la longitud de la línea analizada.

En el grafo (véase la figura 4.2), las aristas se encuentran etiquetadas por la pseudopalabra que se ha encontrado entre dos nodos, así como la relevancia que tiene esta en ese espacio. Como podemos observar en la figura, los instantes de tiempo asignados a los nodos de una arista representan dónde se ha encontrado dicha pseudopalabra, es decir, nos está indicando la *bounding box* a la que pertenece dicha cadena de caracteres.

Pese a pertenecer a la misma palabra del texto manuscrito, las pseudopalabras no tienen por qué abarcar el mismo instante de tiempo, es más puede darse el caso en el que encontremos una pseudopalabra duplicada, donde la única diferencia entre estas resida en el instante en el que fueron detectadas. A modo de ejemplo, en la figura podemos observar que, para la segunda palabra del texto manuscrito, se encuentran dos pseudopalabras que hacen referencia a "votos", una con relevancia igual a 0.1 y la otra con relevancia igual a 0.2. Podemos observar como la referencia con relevancia 0.1, comienza antes que la otra relevancia pero termina en el mismo punto.

Por último, cabe destacar que, la suma de las relevancias de las aristas salientes de un nodo debe ser igual a la suma de las relevancias de las aristas entrantes.

El tamaño de este grafo, pese a haber reducido su magnitud sigue siendo considerable. Por este mismo motivos surgen los índices probabilísticos, lo que nos permitirá reducir aún más el peso de estos grafos.

Un índice probabilístico consiste en una lista de palabras obtenidas de una imagen. En esta lista se indica el término encontrado, la probabilidad de encontrar dicho término, así como la posición de la *bounding box* donde se ha encontrado. Como veremos más adelante, existe la posibilidad de que se incluya información adicional.

La probabilidad a posteriori de encontrar un término v en una imagen X se define como la probabilidad de que una palabra v este escrita en una región de X que incluya al píxel (i, j). A la región en la que se encuentra dicho píxel se le denomina *bounding box*. Esta probabilidad se calcula como se muestra a continuación:

$$P(v|X,i,j) = \sum_{b \epsilon B(i,j)} P(v,b|X,i,j) = \sum_{b \epsilon B(i,j)} P(b|X,i,j) \cdot P(v|X,b,i,j). \tag{4.4}$$

Donde X es una imagen de texto de dimensiones IxJ, V es el vocabulario, (i, j) se corresponde con píxeles de la imagen, tal que $i \epsilon [1, I]$, $j \epsilon [1, J]$; y $B(i, j)$ representa el conjunto de todos los *bounding boxes* de la imagen que contienen el píxel (i, j). En este cálculo se considera que cada una de las palabras de V están escritas en todos los posibles *bounding boxes* de la imagen X.

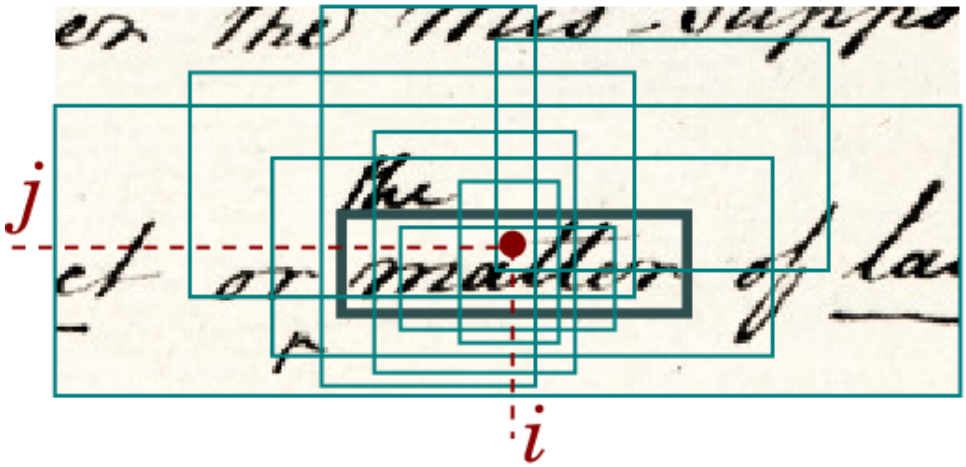


Figura 4.3: Ejemplo *bounding boxes* $b \epsilon B(i, j)$, para la palabra v="matter". Fuente: [16].

En la figura 4.3 se observa el proceso que se lleva a cabo para realizar el cálculo comentado anteriormente. Podemos observar que destaca una *bounding box* por encima del resto puesto que tiene una línea más gruesa, esto nos indica que se trata de la región donde la probabilidad de $P(v|X, b)$ es máxima. El resto de bounding boxes contribuyen a la suma.

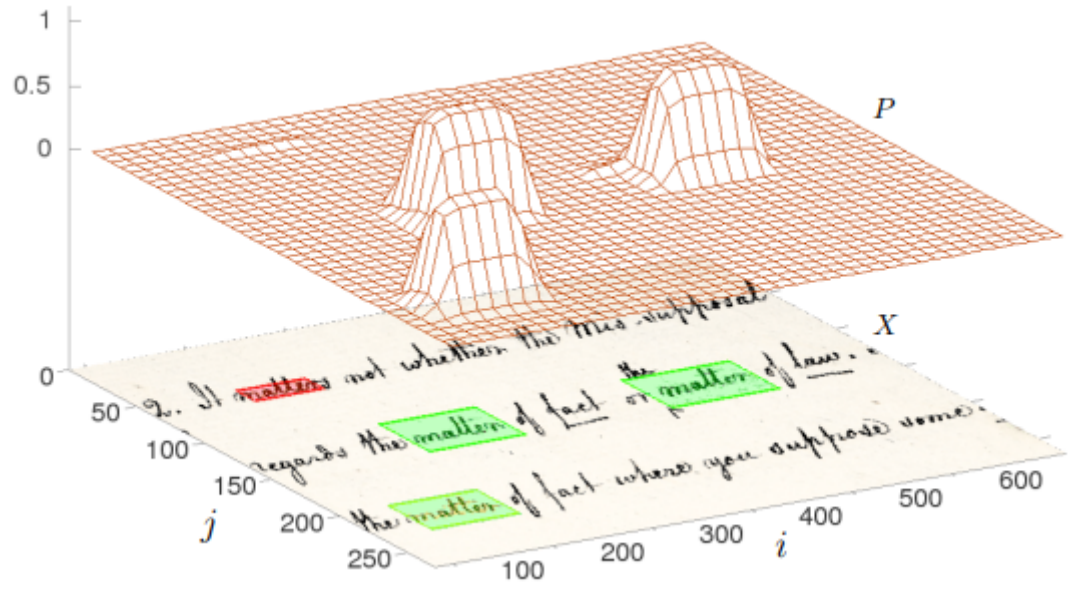


Figura 4.4: Ejemplo del cálculo de la probabilidad a posteriori, mediante un posteriorgrama para la palabra v="matter". Fuente: [17].

En la figura 4.4, se presenta un posteriorgrama a nivel de píxel, un concepto imprescindible en la indexación probabilística. Representa un mapa de probabilidad generado para una imagen dada X y una posible palabra v. Para cada posición (i, j) de X, el posteriorgrama nos devuelve la probabilidad a posteriori de que una palabra v se encuentre escrita en una subimagen de X, que contiene al píxel (i, j).

Para obtener esta probabilidad mediante marginación estadística y para ello, debemos suponer que la palabra v se ha podido escribir en cualquiera de las *bounding boxes* de la imagen X, que incluyen al píxel (i, j). Este proceso de marginación consiste en sumar las probabilidades de reconocer la palabra, para cada *bounding box*.

Tras la extracción de los índices probabilísticos podemos realizar búsquedas de aquellos términos presentes en los manuscritos que superen un cierto umbral. Esta exploración es conocida como *Keyword Spotting*.

PSEUDOWORD	GT	SCORE	[INFO_BB]	LINE_ID	[POSITION]
natural<print>	0	0.00345156	3320 1277 186 62	line_1626683753618_1293	5 0.00344945 6 2.10619e-06

Figura 4.5: Ejemplo de la estructura del índice probabilístico. Fuente: Elaboración propia.

```
Sodel 0 0.000101089 1893 1343 310 62 TextLine_GSq4  3 0.000101089
Tod 0 0.00085465 1814 1343 153 62 TextLine_GSq4  3 0.00085465
Toda 0 0.0134223 1850 1343 225 62 TextLine_GSq4  3 0.0134223
Todal 0 0.00108568 1893 1343 310 62 TextLine_GSq4  3 0.00108568
Todal<print> 0 0.000105249 1899 1343 322 62 TextLine_GSq4  3 0.000105249
Tode 0 0.0444114 1850 1343 225 62 TextLine_GSq4  3 0.0444114
Todel 0 0.129 1893 1343 310 62 TextLine_GSq4  3 0.129
Todela 0 0.000809437 1897 1343 318 62 TextLine_GSq4  3 0.000809437
Todell 0 0.000558061 1893 1343 310 62 TextLine_GSq4  3 0.000558061
Todella 0 0.000242452 1893 1343 310 62 TextLine_GSq4  3 0.000242452
Todel<country> 0 0.000209054 1899 1343 322 62 TextLine_GSq4  3 0.000209054
Todel<print> 0 0.0336696 1899 1343 322 62 TextLine_GSq4  3 0.0336696
Tode<print> 0 0.000989718 1899 1343 322 62 TextLine_GSq4  3 0.000989718
Todl 0 0.000101024 1897 1343 318 62 TextLine_GSq4  3 0.000101024
Todo 0 0.000823968 1850 1343 225 62 TextLine_GSq4  3 0.000823968
Todo<print> 0 0.000194311 1899 1343 322 62 TextLine_GSq4  3 0.000194311
de 0 0.00217989 1868 1343 261 62 TextLine_GSq4  3 0.00217989
del 0 0.0242214 1893 1343 310 62 TextLine_GSq4  3 0.0242214
del<print> 0 0.00642145 1899 1343 322 62 TextLine_GSq4  3 0.00642145
de<print> 0 0.000680409 1899 1343 322 62 TextLine_GSq4  3 0.000680409
mmundo 0 0.000885662 2261 1343 381 61 TextLine_GSq4  4 0.000885662
mudo 0 0.0190171 2265 1343 389 61 TextLine_GSq4  4 0.0190171
mundo 0 0.755854 2261 1343 381 61 TextLine_GSq4  4 0.755854
odel 0 0.000273859 1893 1343 310 62 TextLine_GSq4  3 0.000273859
para<print> 0 0.999985 1678 1342 73 61 TextLine_GSq4  2 0.999985
tod 0 0.00292322 1814 1343 153 62 TextLine_GSq4  3 0.00292322
toda 0 0.00795452 1850 1343 225 62 TextLine_GSq4  3 0.00795452
```

Figura 4.6: Ejemplo del índice probabilístico "AGA_TOP-55-79-LIB-05975-002.idx" . Fuente: Elaboración propia.

En este trabajo, los índices probabilísticos tienen una estructura fija (véase las figuras 4.5 y 4.6), la cual procedemos a comentar:

PSEUDOWORD GT SCORE [INFO_BB] LINE_ID [POSITION]

PSEUDOWORD Suele ser una cadena de caracteres. Esta representa que en dicha región de imagen, comprendida dentro de la *bounding box* indicada, puede que se encuentre este término. Cabe destacar que puede que haya pseudopalabras repetidas pero que estas se encuentren en *bounding boxes* distintas.

GT Consiste en un valor lógico e indica si dicha pseudopalabra pertenece al *ground truth*, es decir a la referencia.

SCORE Consiste en un valor numérico que indica la probabilidad a posteriori de que se encuentre dicho término en dicha región de la imagen. Más en profundidad, este valor representa la suma de las probabilidades a posteriori de las diferentes posiciones en las que puede aparecer el término dentro de la misma línea, es por esto que puede ser, en algunos casos, mayor que 1.

[INFO_BB] Representa la localización geométrica dentro de la imagen. Está formada por 4 valores siendo estos el valor del centro de la caja en el eje X, el valor del centro de la caja en el eje Y, la anchura de la caja y la altura de la caja, en este orden.

LINE_ID Consiste en un identificador que indica la línea en la que se encuentra dicha región de imagen.

[POSITION] Contiene duplas formadas por la posición esperada dentro de la línea y su correspondiente relevancia o probabilidad a posteriori. Hay que tener en cuenta que puede haber más de una posición esperada y por tanto más de una dupla. La suma de estas relevancias nos proporciona el valor de SCORE.

4.3 Extracción de información a partir de los índices probabilísticos

En este apartado se tratará el objetivo principal del trabajo, la extracción de información de los índices probabilísticos para cada campo del formulario de la imagen. Durante el desarrollo de esta solución nos hemos encontrado con distintos problemas los cuales procedemos a enunciar y comentaremos, además, como los hemos solventado.

En cada imagen nos encontramos con 4 pasaportes. El primer paso consiste en separarlos y para ello nos apoyaremos en las características del pasaporte. Utilizaremos un término, en nuestro caso "Nº" perteneciente al campo "Nº de orden", para poder distinguir entre los distintos pasaportes. Con esto nos surgen varios problemas que debemos controlar para el correcto funcionamiento del programa.

Uno de los problemas es la detección del término empleado para la separación en un campo distinto al de "Nº de orden". Para evitar que esto ocurra, utilizaremos las ordenadas de estos términos para comprobar que se encuentran alineados con una separación de máximo un 10 % respecto a la media de las ordenadas. Los términos que no se encuentren alineados pertenecen a otros campos, y por tanto, no se deben de tener en cuenta para la separación.

Otro de los problemas es cuando se detectan menos de 4 términos. Esto significa que no se va a detectar un pasaporte y, por consiguiente, se perderá la información relacionada con este. Con el fin de evitarlo comprobaremos a qué pasaportes se corresponden los términos que faltan y crearemos una geometría artificial para cada uno de estos.

El último de los problemas encontrados en este punto consiste en la detección de más de 4 términos alineados. Esto se debe a que se ha duplicado la detección de uno de estos términos, por ejemplo se detecta un término "Nº" con una puntuación de 0.8 y otro con una puntuación de 0.2 para el mismo pasaporte. Como sabemos que estos se encuentran alineados, puesto que se ha comprobado debido al primer problema, nos quedaremos con aquel que tenga una puntuación mayor siempre que se encuentre en la posición correcta, con respecto a las abscisas, del pasaporte al que hace referencia.

El siguiente paso en la extracción es la obtención de la posición del resto de campos para cada pasaporte. Cada vez que detectemos un campo almacenaremos su información para, posteriormente, continuar con la extracción.

En este punto, podemos encontrar que no todos los campos de todos los pasaportes han sido localizados. Para solventar este problema, crearemos una geometría artificial para proseguir con la extracción de información.

Para finalizar la extracción, agruparemos todos los términos del índice por pasaportes y, a continuación, los separaremos según el campo al que pertenecen. Podemos observar, en la figura 4.7, el resultado de la agrupación por campos para un pasaporte, en un formato visual. A continuación, procedemos a la explicación del proceso llevado a cabo para su obtener las agrupaciones por campo.

Figura 4.7: Representación visual de la extracción de información realizada. Fuente: Elaboración propia.

Para la segmentación por campos (véase la figura 4.7), obtendremos aquellos términos que se encuentren entre la parte superior e inferior de la *bounding box* del campo. Al emplear esta solución, nos surge un problema en las líneas que incluyen varios términos.

Como solución a este problema trataremos estas líneas por separado, haciendo uso de las propiedades geométricas de la línea y de los campos incluídos en esta.

Para los campos de “Estado” y “Clase”, agruparíamos todos los términos que se encuentran entre el borde superior e inferior de la *bounding box*, en cuanto a las ordenadas, y entre el borde izquierdo de la *bounding box* y el borde izquierdo de la *bounding box* de la categoría siguiente, “Profesión” y “Derechos cobrados $” respectivamente, en cuanto a las abscisas.

Para los campos “Profesión” y “Derechos cobrados $”, agruparemos los términos que se encuentren entre el borde superior e inferior de la *bounding box*, en cuanto a las ordenadas, y que además se encuentren a partir del borde izquierdo de la misma, en cuanto a las abscisas.

En cuanto al campo “años”, agruparemos los términos que se encuentren entre el borde superior e inferior de la *bounding box*, en cuanto a las ordenadas, y que se encuentren antes del borde derecho de la misma, en cuanto a las abscisas.

Otra dificultad que podemos encontrar en la segmentación tiene relación con los campos que abarcan más de una línea de texto. En este caso, con la estrategia empleada esto no nos ocasiona ningún problema puesto que se resuelve de manera trivial.

Por último, aplicaremos aplicaremos el umbral deseado por el usuario y se generará un archivo JSON por cada pasaporte.

{"orden" 1 "Nº<print>",1,2613,393,"TextLine_71Iz","1",1.00002],["de<print>",1,2667,395,"TextLine_71Iz","2",1.00002],
2 ["Orden<print>",1,2751,397,"TextLine_71Iz","3",1.00002],["2427<print>",0.765522,3028,397,"TextLine_71Iz","4",0.765522]],
"fecha" 3 "Fecha<print>",1,2834,470,"TextLine_bKGk","1",1],["30<date>",0.993339,3010,470,"TextLine_bKGk","2",0.993339],
4 ["MAYO<date>",0.999969,3243,471,"TextLine_bKGk","3",0.999969],["1936<date>",0.989995,3375,471,"TextLine_bKGk","4",0.989995],
5 ["de<print>",1,3413,471,"TextLine_bKGk","5",1.00002],["193<print>",0.948762,3461,471,"TextLine_bKGk","6",0.948762]],
"nombre 6 ["apellidos<print>",1,2822,539,"TextLine_WV9Q","3",1],["Maria<gname>",0.994006,3039,539,"TextLine_WV9Q","4",0.994006],
7 ["Dolores<gname>",0.984842,3361,544,"TextLine_WV9Q","5",0.984842],["Nombre<print>",0.999985,2646,546,"TextLine_WV9Q","1",0.999985],
8 ["y<print>",0.999985,2728,546,"TextLine_WV9Q","2",0.999985],["Cazán<surname>",0.402424,3109,614,"TextLine_IrOw","2",0.402424],
9 ["Gartinez<surname>",0.390424,2773,619,"TextLine_IrOw","1",0.390424]],
"naturc 10 [["Natural<print>",1,2645,693,"TextLine_8HB2","1",1],["de<print>",1,2725,693,"TextLine_8HB2","2",1.00002],
11 ["La<place>",0.999985,2971,694,"TextLine_8HB2","3",0.999985],["Coruña<place>",0.995205,3381,697,"TextLine_8HB2","4",0.995205]],
"provir 12 ":[["Provincia<print>",1,2657,767,"TextLine_g07n","1",1],["de<print>",1,2755,767,"TextLine_g07n","2",1],
13 ["idem<state>",0.993339,3078,768,"TextLine_g07n","3",0.993339]],
"años<p 14 t>":[["50<age>",0.937838,2660,841,"TextLine_MY4o","2",0.937838],["años<print>",1,2735,841,"TextLine_MY4o","3",1.00002],
15 ["De<print>",0.999985,2610,842,"TextLine_MY4o","1",0.999985]],
"estado 16 ["Estado<print>",0.999985,2837,842,"TextLine_MY4o","4",0.999985],["soltera<civilstate>",0.772257,3002,842,"TextLine_MY4o","5",0.772257]],
"profes 17 ":[["Profesión<print>",1,3178,843,"TextLine_MY4o","6",1],["labores<job>",0.831106,3373,843,"TextLine_MY4o","7",0.831106]],
"reside 18 nt>":[["Reside<print>",0.999985,2635,915,"TextLine_GYfY","1",0.999985],["habitualmente<print>",0.999985,2796,916,"TextLine_GYfY","2",0.999985],
19 ["en<print>",1,2932,916,"TextLine_GYfY","3",1],["Marino<residence>",0.938754,3181,916,"TextLine_GYfY","4",0.938754],
20 ["327<residence>",0.762689,3470,916,"TextLine_GYfY","5",0.762689]],
"motivo 21 [["del<print>",1,2737,989,"TextLine_RBfS","2",1],["viaje<print>",0.999992,2807,989,"TextLine_RBfS","3",0.999992],
22 ["familia",0.99955,3006,989,"TextLine_RBfS","4",0.99955],["Motivos<print>",0.999908,2644,990,"TextLine_RBfS","1",0.999908]],
"persor 23 :[["que<print>",1,2759,1063,"TextLine_FEha","2",1.00001],["le<print>",1,2817,1063,"TextLine_FEha","3",1],
24 ["Personas<print>",0.967121,2655,1064,"TextLine_FEha","1",0.967121],["acompañan<print>",1,2923,1064,"TextLine_FEha","4",1]],
"dado": 25 ado<print>",1,2630,1341,"TextLine_LPe5","1",1.00001],["para<print>",1,2712,1342,"TextLine_LPe5","2",1.00002],
26 ["España<country>",0.986557,2910,1342,"TextLine_LPe5","3",0.986557]],
"docume 27 :[["1239-36",0.392018,1473,"TextLine_VY7U","6",0.392018],["exhibido<print>",0.999985,2949,1483,"TextLine_VY7U","4",0.999985],
28 ["C",0.999512,3043,1483,"TextLine_VY7U","5",0.999512],["Documento<print>",1,2672,1484,"TextLine_VY7U","1",1],
29 ["que<print>",1,2792,1484,"TextLine_VY7U","2",1],["ha<print>",1,2851,1484,"TextLine_VY7U","3",1],["5667",0.365563,3366,1602,"TextLine_KaIB","1",0.365563]],
"clase' 30 '2#",0.999542,2841,1685,"TextLine_Mv9q","2",0.999542],["Clase<print>",0.999741,2636,1689,"TextLine_Mv9q","1",0.999741]],
"derecl 31 :[["cobrados<print>",1,3215,1679,"TextLine_Mv9q","4",1],[">print>",1,3320,1679,"TextLine_Mv9q","5",1],
32 ["Derechos<print>",0.999878,3065,1682,"TextLine_Mv9q","3",0.999878],["6",0.990811,3436,1682,"TextLine_Mv9q","6",0.990811]],
"observ 33 nes":[["Observaciones<print>",1,2702,1768,"TextLine_06mk","1",1]]}

Figura 4.8: Ejemplo archivo JSON obtenido para el índice “AGA_TOP-55-79-LIB-05975-002.idx”, con umbral 1.0. Fuente: Elaboración propia.

JSON (o *JavaScript Object Notation*) es una notación para la transferencia de datos que sigue un estándar específico.

Los datos contenidos en un archivo en formato JSON deben estructurarse por medio de una colección de pares con nombre y valor o deben ser una lista ordenada de valores.

La estructura seleccionada (véase la figura 4.8) para los ficheros obtenidos en nuestro sistema es la siguiente:

- "orden": Incluirá toda la información extraída correspondiente al campo "Nº de Orden".
- "fecha": Incluirá toda la información extraída correspondiente al campo "Fecha".
- "nombre": Incluirá toda la información extraída correspondiente al campo "Nombre y apellidos".
- "natural": Incluirá toda la información extraída correspondiente al campo "Natural de".
- "provincia": Incluirá toda la información extraída correspondiente al campo "Provincia de".
- "años<print>": Incluirá toda la información extraída correspondiente al campo "Años"
- "estado": Incluirá toda la información extraída correspondiente al campo "Estado".
- "profesión": Incluirá toda la información extraída correspondiente al campo "Porfesión".
- "reside<print>: Incluirá toda la información extraída correspondiente al campo "Reside habitualmente en".
- "motivos": Incluirá toda la información extraída correspondiente al campo "Motivos del viaje".
- "personas": Incluirá toda la información extraída correspondiente al campo "Personas que le acompañan".
- "dado": Incluirá toda la información extraída correspondiente al campo "Dado para".
- "documento": Incluirá toda la información extraída correspondiente al campo "Documentos que ha exhibido".
- "clase": Incluirá toda la información extraída correspondiente al campo "Clase".
- "derechos": Incluirá toda la información extraída correspondiente al campo "Derechos cobrados $".
- "observaciones": Incluirá toda la información extraída correspondiente al campo "Observaciones".

Con el fin de localizar de manera más sencilla los pasaportes, los archivos JSON se nombran indicando la imagen a la que pertenecen, el pasaporte al que hacen referencia dentro de la imagen, y el valor de confianza o umbral empleado para su obtención.

Por ejemplo, el fichero "AGA_TOP-55-79-LIB-05979-014-r_passport_2_confidence_1-0.json" contiene la información extraída con umbral 1.0 para el tercer pasaporte del índice "AGA_TOP-55-79-LIB-05979-014-r.idx".

CAPÍTULO 5
Marco experimental

En este capítulo se presentarán las medidas de evaluación utilizadas en este así como los experimentos realizados.

5.1 Descripción de las métricas

Antes de comenzar debemos definir qué entendemos por verdadero positivo, verdadero negativo, falso positivo y falso negativo, para poder así identificar cuáles son los términos que debemos utilizar al calcular las métricas.

- Verdadero positivo, TP. Nos referimos como TP a los términos que han sido encontrados por nuestro sistema y que aparecen en la referencia.
- Verdadero negativo, TN. En nuestro caso no tiene sentido hablar de TN, puesto que esto significa que los términos que no han sido encontrados por el sistema no se encuentran en la referencia.
- Falso positivo, FP. Nos referimos como FP a los términos que han sido encontrados por nuestro sistema pero no se encuentran en la referencia.
- Falso negativo, FN. Nos referimos como FN a los términos que no ha sido encontrados por nuestro sistema pero, sin embargo, aparecen en la referencia.

Para el experimento que se llevará a cabo se han seleccionado varias métricas [19] que procedemos a comentar a continuación.

Precisión

Esta métrica nos permite conocer el ratio entre el número de muestras clasificadas correctamente y el número de muestras clasificadas, es decir, nos permite medir la calidad del sistema. Se calcula de acuerdo a la siguiente fórmula:

$$P(\tau) = \frac{h(\tau)}{d(\tau)} = \frac{TP}{TP + FP}$$

Donde h(τ) representa el número de ítems generados relevantes, d(τ) el número de ítems generados, relevante o no, y τ es el umbral con el que se han obtenido estos ítems.

Exhaustividad o *recall*

Esta métrica nos permite conocer el ratio entre el número de muestras clasificadas correctamente y el número total de muestras correctas del corpus, es decir, nos informa sobre la cantidad de palabras que el sistema es capaz de identificar correctamente. Se calcula de acuerdo a la siguiente fórmula:

$$R(\tau) = \frac{h(\tau)}{r} = \frac{TP}{TP + FN}$$

Donde h(τ) representa el número de ítems generados relevantes, r el número de ítems relevantes en el corpus, es decir aquellos que se encuentran en la referencia y τ es el umbral con el que se han obtenido estos ítems.

Valor-F1

Esta métrica se define como la media armónica de la precisión y el *recall*, por lo que combina ambas métricas en una sola. Se calcula de acuerdo a la siguiente fórmula:

$$F1(\tau) = 2 \cdot \frac{P(\tau) \cdot R(\tau)}{P(\tau) + R(\tau)}$$

Siendo P(τ) y R(τ) los valores de la precisión y *recall*, respectivamente, para un umbral τ.

El valor-F1 representa que, tanto la precisión como el *recall* nos importan de igual manera. Según tengamos que proporcionar más o menos importancia a alguna de estas métricas existen distintos valores-F, como puede ser el valor-F2, que valorará en mayor medida el *recall*.

A continuación se muestra una fórmula genérica para el valor-F.

$$F_\beta = (1 + \beta^2) \cdot \frac{P(\tau) \cdot R(\tau)}{(\beta^2 \cdot P(\tau) + R(\tau)}$$

A partir de las métricas precisión y *recall*, podemos generar una gráfica denominada "*Precision-Recall curve*". Esta gráfica nos mostrará el compromiso entre precisión y *recall*, para diferentes umbrales, pudiendo así determinar a partir de qué valor de *recall* tenemos una atenuación de la precisión y viceversa.

La forma de la curva ideal sería aquella que se aproxime lo máximo posible a la esquina superior izquierda de la gráfica, significando esto que se ha conseguido un sistema que permite obtener valores altos de precisión, así como valores altos de recall

5.2 Experimentos comparativos con la referencia

En este apartado se comentarán los resultados de los distintos experimentos realizados.

En los experimentos se han calculado el número de ítems relevantes en el corpus (r), el número de ítems generados relevantes ($h(\tau)$), el número de ítems generados ($d(\tau)$), la precisión, el *recall* y el valor-F1, para distintos valores de umbral τ (0.0, 0.2, 0.4, 0.6, 0.8 y 1.0).

El cálculo de r se ha llevado a cabo mediante el recuento de los términos presentes en la referencia. Este valor no variará con respecto al umbral y, por tanto, se debe calcular una única vez. En nuestro caso, el valor de r es de 25984.

El cálculo de $d(\tau)$ se ha llevado a cabo mediante el recuento de los términos presentes en los índices probabilísticos generados, ya sean estos términos relevantes o no respecto a la referencia.

En la tabla 5.1, podemos observar la variación de este valor respecto al umbral. Para valores poco restrictivos de umbral obtenemos un número elevado de términos y, conforme aumenta el umbral, el número de términos irá en descenso.

En este caso detectamos que, para un umbral de 1.0, obtenemos un $d(\tau)$ igual a 8902.0, lo que supone un decremento de 14005.29 términos con respecto al valor obtenido para un umbral de 0.8. Teniendo en cuenta que la diferencia promedio entre el resto de umbrales es 662.5, se trata de un descenso brusco.

Tras analizar detenidamente los índices buscando posibles causas, hemos supuesto que este descenso es debido a que, en el índice de umbral 1.0, solamente se encuentran los términos impresos del pasaporte y, muy rara vez, aparece algún término manuscrito. Esto supone una reducción importante del tamaño del índice, en algunos casos representando simplemente un tercio de los términos, comparado con el índice de umbral 0.0.

El cálculo de $h(\tau)$ se ha llevado a cabo mediante el recuento de términos presentes tanto en los índices probabilísticos como en la referencia, es decir contabilizaremos los términos que se encuentran en la referencia, los términos relevantes.

Umbral - τ	$d(\tau)$	$h(\tau)$
0.0	25557.30	19986.42
0.2	24546.64	19962.57
0.4	24096.60	19909.12
0.6	23597.31	19784.52
0.8	22907.29	19538.85
1.0	8902.00	8383.00

Tabla 5.1: Resultados de las medidas de $d(\tau)$ y $h(\tau)$. Fuente: Elaboración propia.

En la tabla 5.1 podemos observar que se trata de un valor que decrece conforme aumenta el valor del umbral, de igual manera que $d(\tau)$.

En este caso, detectamos el mismo comportamiento que con $d(\tau)$, esto es, se detecta un descenso rápido en el número de *hits* con respecto a la referencia. Tras analizarlo, hemos supuesto que es debido al mismo motivo que en $d(\tau)$.

En cuanto a la precisión, el *recall* y el valor-F1 estas han sido calculadas mediante las fórmulas explicadas en el apartado anterior.

Umbral - τ	Precisión $P(\tau)$	Recall $R(\tau)$	Valor-F1
0.0	0.78	0.77	0.78
0.2	0.81	0.77	0.79
0.4	0.83	0.77	0.80
0.6	0.84	0.76	0.80
0.8	0.85	0.75	0.80
1.0	0.94	0.32	0.48

Tabla 5.2: Resultados del cálculo de la precisión, el *recall* y el valor-F1. Fuente: Elaboración propia.

En la tabla 5.2, podemos observar que el valor de la precisión es creciente con respecto al umbral, conforme aumenta el umbral también lo hace la precisión. Sin embargo, esto no ocurre con el *recall* debido a que tiene un comportamiento decreciente con respecto al umbral.

Cabe destacar el comportamiento que tiene el *recall* conforme aumenta el umbral. Podemos observar que, para un umbral de 1.0, obtenemos un valor de 0.32 que, comparado con el valor para el umbral anterior (0.8), supone un descenso muy brusco.

Suponemos que este descenso tan brusco es, como en los apartados de $d(\tau)$ y $h(\tau)$, debido al número de términos. Como comentamos en el apartado 5.1, el *recall* se calcula a partir de los *hits*, o $h(\tau)$, y los términos de la referencia, r, siendo este último un valor fijo que no varía con respecto al umbral. Por tanto, al observar para el umbral 1.0 un descenso tan brusco en $h(\tau)$, también lo observaremos en el *recall*.

Sin embargo, en el caso de la precisión no observamos el comportamiento anteriormente comentado, debido a que para su cálculo se emplean $h(\tau)$ y $d(\tau)$, siendo ambas variables con respecto al umbral.

Una vez comentados los resultados para la precisión y el *recall*, podemos generar la curva precisión-*recall*. En la figura 5.1, podemos observar el compromiso entre ambas métricas, obteniendo para valores altos de precisión, valores bajos de *recall* y viceversa.

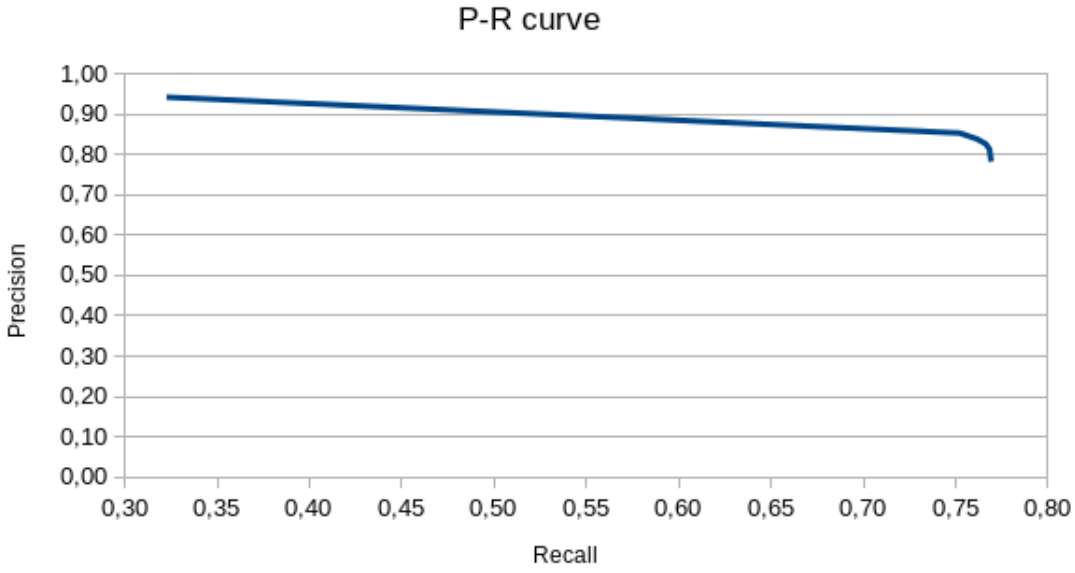


Figura 5.1: Curva precisión-recall. Fuente: Elaboración propia.

Finalmente, el valor-F1 nos permite combinar la precisión y el *recall* en una única medida para, de esta forma, poder evaluar la utilidad combinada de ambas medidas.

En la tabla 5.2, podemos observar nuevamente un descenso repentino para el umbral 1.0. Esto es debido al comportamiento de esta medida, puesto que, cuando una de las dos medidas es inferior a la otra obtenemos valores bajos. Es decir, con esta métrica, asignamos la misma importancia tanto a la precisión como al *recall*.

CAPÍTULO 6
Conclusiones

En el capítulo que se presenta se recapitulará el trabajo realizado para comprobar cuáles de los objetivos propuestos han sido logrados.

Antes de comenzar con el desarrollo de la aplicación hemos tenido que estudiar y comprender las bases de los sistemas de reconocimiento de texto manuscrito, los índices probabilísticos y su proceso de obtención, así como familiarizarse con las técnicas de extracción de información, sus métricas y sus métodos de evaluación. Además hemos analizado trabajos de otros investigadores con el fin de conocer las estrategias que adoptan para atacar distintos problemas. Con esto podemos decir que el primer objetivo principal se ha logrado.

Tras conocer el contexto del trabajo procedemos al desarrollo de la aplicación. Hemos conseguido desarrollar una aplicación que, dado un índice probabilístico de una imagen, nos devuelva la transcripción de cada uno de los pasaportes presentes en dicha imagen. Además el usuario puede ajustar el parámetro de confianza, el cual le permite ajustar según sus necesidades el resultado devuelto por el programa. Por estos motivos, podemos concluir que otra de las metas principales del trabajo se ha alcanzado.

Como formato de almacenamiento del resultado generado hemos elegido el formato JSON. Este formato es compatible con los demostradores del PRHLT y por tanto podemos concluir que el tercer objetivo del trabajo se ha logrado.

Por último, hemos realizado varios experimentos para obtener distintas métricas con las que evaluar el rendimiento del sistema desarrollado. Es por este motivo por el que damos por alcanzado el último objetivo del trabajo.

El trabajo me ha permitido conocer en profundidad el área de estudio del reconocimiento de texto, además de permitirme profundizar mis conocimientos de Python y de librerías como Pandas, muy utilizada en la manipulación y análisis de datos.

6.1 Trabajo futuro

Como posibles ampliaciones para trabajos futuros se plantean varias ideas:

- Realizar experimentos más exhaustivos como, por ejemplo, analizar la aportación de las n-mejores soluciones extraídas del GP frente a la 1-mejor.

- Construir una estructura que permita acumular la información obtenido por los índices probabilísticos para, utilizándolo como base, desarrollar un demostrador.

- Desarrollar un demostrador que permita el acceso a la información extraída, de una forma cómoda y sencilla para usuarios ajenos al proyecto.

6.2 Relación con el grado

Los conocimientos adquiridos en el grado, en concreto en la rama de computación han sido esenciales para el desarrollo de este proyecto.

Algunas de estas asignaturas esenciales han sido aprendizaje automático (APR) y percepción (PER), donde se introducen las bases teóricas del aprendizaje automático;

También han sido imprescindibles las asignaturas de estructuras de datos y algoritmos (EDA), algorítmica (ALG) y competición de programación (CACM), donde se estudian las bases para el desarrollo de código eficiente así como estrategias para atacar diversos problemas;

Por último, sistemas de almacenamiento y recuperación de información (SAR), donde se introducen la base teórica sobre la extracción de información en grandes volúmenes de datos y sus métodos de evaluación.

Bibliografía

[1] Manuscrito - Wikipedia Consultado en `https://es.wikipedia.org/wiki/Manuscrito`.

[2] Pletschacher, S. and Antonacopoulos, A. The PAGE (Page Analysis and Ground-truth Elements) Format Framework. *Proceedings of the 20th International Conference on Pattern Recognition (ICPR2010)*, pages 257-260, IEEE-CS Press, August, 23-26, 2010, Istanbul, Turkey.

[3] Sivan Keret, Lior Wolf, Nachum Dershowitz, Eric Werner, Orna Almogi, and Dorji Wangchuk. Transductive Learning for Reading Handwritten Tibetan Manuscripts. *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 214–221. IEEE, 2019.

[4] Clausner, Christian and Antonacopoulos, Apostolos and Mcgregor, Nora and Wilson-Nunn, Daniel Icfhr 2018 competition on recognition of historical arabic scientific manuscripts–rasm2018. *2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, pages 471-476. IEEE, 2018.

[5] P. Dreuw, G. Heigold, and H. Ney. Confidence-based discriminative training for model adaptation in offline arabic handwriting recognition. In *Proc. of the Int. Conf. on Document Analysis and Recognition (ICDAR 200)*, pages 596–600, Barcelona (Spain), July 2009.

[6] Toselli, A. H. and Vidal, E. Fast HMM-Filler Approach for Key Word Spotting in Handwritten Documents. *2013 12th International Conference on Document Analysis and Recognition*, pages 501-505, 2013.

[7] Giménez, A. and Khoury, I. and Juan, A. Windowed Bernoulli Mixture HMMs for Arabic Handwritten Word Recognition. *2010 12th International Conference on Frontiers in Handwriting Recognition*, pages 533-538, 2010.

[8] Lang, E. and Puigcerver, J. and Toselli, A. H. and Vidal, E. Probabilistic Indexing and Search for Information Extraction on Handwritten German Parish Records. *2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, pages 44-49, 2018.

[9] L.R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, pages 257-286 vol.77. IEEE, 1989.

[10] Blunsom, P. Hidden Markov Models. 2004.

[11] Jurafsky, D. and Martin, J. H. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition.* Prentice Hall, borrador tercera edición, 2008.

[12] Speech and Language Processing Consultado en https://web.stanford.edu/~jurafsky/slp3/

[13] Toselli, AH.; Vidal, E.; Romero, V.; Frinken, V. (2016). HMM word graph based keyword spotting in handwritten document images. *Information Sciences*. 370:497-518. https://doi.org/10.1016/j.ins.2016.07.063

[14] Likforman-Sulem L. and Zahour A. and Taconet B. Text Line Segmentation of Historical Documents: a Survey *IJDAR*, 2007.

[15] Oerder, M. and Ney, H. Word graphs: an efficient interface between continuous-speech recognition and language understanding. *1993 IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 119-122 vol.2. IEEE, 1993.

[16] Vidal E. and Toselli A. H. and Puigcerver J. A Probabilistic Framework for Lexicon-based Keyword Spotting in Handwritten Text Images. 2021.

[17] Vidal, E. and Sánchez, J. A. Handwritten text recognition for the EDT project. Part II: textual information search in untranscribed manuscripts. *Manuscript papers prepared for the workshop*, page 32, 2021.

[18] tranSkriptorium AI. Ground-truth Generation through Crowdsourcing with Probabilistic Indexes in the EDT project. En proceso de revisión, 2022.

[19] C.D. Manning, P. Raghavan, H. Schütze. Introduction to Information Retrieval. *Cambridge University Press*, New York, NY, USA, 2008.

ANEXO

OBJETIVOS DE DESARROLLO SOSTENIBLE

Grado de relación del trabajo con los Objetivos de Desarrollo Sostenible (ODS).

Objetivos de Desarrollo Sostenible	**Alto**	**Medio**	**Bajo**	**No procede**
ODS 1. **Fin de la pobreza.**				X
ODS 2. **Hambre cero.**				X
ODS 3. **Salud y bienestar.**				X
ODS 4. **Educación de calidad.**				X
ODS 5. **Igualdad de género.**				X
ODS 6. **Agua limpia y saneamiento.**				X
ODS 7. **Energía asequible y no contaminante.**				X
ODS 8. **Trabajo decente y crecimiento económico.**				X
ODS 9. **Industria, innovación e infraestructuras.**	X			
ODS 10. **Reducción de las desigualdades.**				X
ODS 11. **Ciudades y comunidades sostenibles.**				X
ODS 12. **Producción y consumo responsables.**				X
ODS 13. **Acción por el clima.**				X
ODS 14. **Vida submarina.**				X
ODS 15. **Vida de ecosistemas terrestres.**				X
ODS 16. **Paz, justicia e instituciones sólidas.**	X			
ODS 17. **Alianzas para lograr objetivos.**				X

Reflexión sobre la relación del TFG/TFM con los ODS y con el/los ODS más relacionados.

El trabajo de fin de grado presentado se titula "Extracción de información de imágenes de pasaportes a partir de los índices probabilísticos". La relación que guarda este proyecto con los objetivos de desarrollo sostenibles es muy sútil, esto es, mantiene una relación directa con estos pero, debido a que se trata de un trabajo centrado en un área académica de investigación de la rama de computación, no es capaz de proporcionar una solución precisa para los problemas planteados en estos objetivos de desarrollo sostenible.

Sin embargo, podríamos seleccionar unos pocos, dentro de los 17 objetivos de desarrollo sostenible, en concreto 2 de ellos. Estos objetivos serían "ODS 9. Indústria, innovación e infraestructuras" y "ODS 16. Paz, justícia e instituciones sólidas", siendo estas las opciones que consideramos más apropiadas en el contexto de este proyecto.

El "ODS 9. Indústria, innovación e infraestruturas" tiene como objetivos la construcción de infraestructuras resilientes, impulsar la industrialización inclusiva y sostenible y fomentar la innovación, con el objetivo de favorecer el crecimiento económico y el desarrollo social, además de tomar acción contra el cambio climático.
Nuestro trabajo se relaciona con este en lo que se respecta al fomento de la innovación. Esto se debe a que trabajamos en áreas de investigación donde constantemente se están produciendo innovaciones, siendo el presente trabajo una de estas innovaciones.
En concreto, trabajamos con técnicas recientes como, por ejemplo, los índices probabilísticos. Además nuestro objetivo es obtener un sistema capaz de extraer la información contenida en imágenes de pasaportes de los años 30, cuyo objetivo a largo plazo sea el desarrollo de un demostrador, el cual nos permita un acceso cómodo y directo al resultado de nuestro sistema, fomentando así de nuevo la innovación.

El "ODS 16. Paz, justícia e instituciones sólidas" tiene como objetivos promover las sociedades pacíficas e inclusivas para el desarrollo sostenible, facilitar el acceso universal a la justícia y la creación de instituciones eficaces, responsables e inclusivas, a todos los niveles. Los objetivos mencionados previamente son necesarios para lograr los objetivos de desarrollo sostenible.
Nuestro trabajo puede guardar relación con este objetivo de desarrollo sostenible en cuanto a la creación de instituciones eficaces, responsables e inclusivas.
En concreto, la relación que guarda dicho objetivo con nuestro se debe a que, gracias al sistema desarrollado, podemos llegar a permitir, a todo el mundo, el acceso a datos que, en caso contrario, su acceso sería muy complicado o incluso imposible.
Además, con estas medidas, se está contribuyendo a fomentar el desarrollo de gobiernos abiertos y transparentes basados en la idea de los datos abiertos.

Finalmente, comentar que, pese a no existir una relación directa con los objetivos de desarrollo sostenible, las aplicaciones que se podrán generar, a partir de los resultados obtenidos mediante este sistema, pueden correlacionarse en mayor medida a estos. Es por esto que podemos concluir que, a corto plazo, la relación que tiene el proyecto con los objetivos de desarrollo sostenible es muy débil. Sin embargo, a largo plazo, puede que aumente el grado de relación.