Universidad Tecnológica Nacional
Facultad Regional Santa Fe

INGENIERÍA EN SISTEMAS DE INFORMACIÓN

PROYECTO FINAL DE CARRERA

"Análisis de factibilidad para implementación de software basado en técnicas de resumen automático aplicado a documentos legales"

Autores
Grosso, Juan Ignacio. Legajo Universitario: 23795
Montiel, Facundo. Legajo Universitario: 24153

Directores
Dra. Gutiérrez, María de los Milagros
Dr. Rubiolo, Mariano

26/09/2022

Agradecimientos

Queremos agradecer a todas las personas que de alguna manera u otra manera colaboraron y fueron parte de la realización de este proyecto, realizado con el fin de obtener el título de Ingeniería en Sistemas de Información.

A nuestros profesores, que con su paciencia, esfuerzo y vocación lograron transmitir los conocimientos y la pasión por este desafiante camino. A nuestros Directores de Proyecto, por su tiempo invertido y sus invalorables aportes al presente trabajo.

Agradecer a nuestros compañeros y amigos, quienes en momentos buenos y malos ofrecieron su sostén y supieron hacer de este recorrido una experiencia gratificante e inolvidable. Gracias por transitarlo con nosotros.

A nuestras familias, por su incondicional soporte y constante motivación. Por ser la fuerza motora detrás de los constantes esfuerzos para ser mejores personas y profesionales.

Gracias a la Universidad Tecnológica Nacional, institución que nos acogió y nos abrió sus puertas para convertirnos en profesionales y hacernos formar parte de una comunidad tan grande y especial como lo es Sistemas.

Sobre todo, queremos agradecer a nuestra más grande fuente de inspiración y ejemplos a seguir, nuestros padres. Gracias por enseñarnos el valor de la perseverancia, la integridad y tantas otras cualidades que nos llevaron a ser quienes somos hoy. Es a ellos a quienes le dedicamos el presente trabajo. Todo ha sido posible gracias a ellos.

Resumen

Este Proyecto Final de Carrera (PFC) se realiza con el objetivo de poder identificar una posible solución para un problema latente en el ámbito judicial. En este contexto, un fallo es el resultado oficial de una demanda en un tribunal y como tal, forma parte de la jurisprudencia. Dado que éstos suelen ser muy extensos, se utilizan los sumarios como una herramienta que resume la parte más significativa de éstos. Generalmente estos sumarios explicitan la interpretación de la ley que hizo el tribunal que dictó el fallo. Para la generación de los mismos, intervienen sumariantes, es decir personal especialmente entrenado para identificar las partes del fallo que forman el sumario. En este trabajo se presenta un estudio de factibilidad acerca de la posibilidad de aplicar técnicas de resumen automático sobre fallos judiciales y así obtener los correspondientes sumarios. Se presentan diferentes modelos aplicando técnicas extractivas. Se muestran los resultados obtenidos y su análisis correspondiente, de acuerdo a métricas definidas que miden la calidad o grado de aceptación de los resultados.

Contenido

1. Introducción

Día a día, personas e instituciones se ven afectadas por grandes volúmenes de información provenientes de distintas fuentes o simplemente generados por ellos mismos. Históricamente la información formó parte de la vida de los seres humanos en distintos formatos, pero fue, sin dudas, siempre un recurso clave en diferentes aspectos. Si se piensa, por ejemplo, en las organizaciones y su operación habitual se podría decir que **saber dónde encontrar la información y cómo usarla** son claves para el éxito.

El crecimiento de internet y de la tecnología en general fueron grandes facilitadores para nutrir de información, a tal punto que es tanto lo que un individuo puede captar que lo excede ampliamente, lo que hace pensar una vez más lo importante de saber identificar qué información es realmente relevante.

Para poder solventar este exceso de información, la tecnología vuelve a jugar un papel clave en un escenario que ella misma generó. La idea de procesar automáticamente información para detectar lo relevante y descartar el resto resulta de gran utilidad.

Si se analiza dicha utilidad, se pueden identificar varios contextos en donde un proceso automático de información podría aplicarse. Por ejemplo, es común que estudiantes recurran a apuntes resumidos de sus compañeros para poder captar lo relevante, y de esa manera estudiar en un tiempo menor al que requeriría revisar todo el material de estudio sin procesar. Otro ejemplo puede ser la industria financiera, si sus dirigentes fueran capaces de procesar información que distintas fuentes emiten a lo largo de todo el mundo, probablemente sus decisiones sean más acertadas, o al menos la información necesaria para la toma de decisiones esté disponible con más anticipación, lo que podría resultar en un accionar más eficiente. Como este par de ejemplos, podrían nombrarse muchos más, pero a lo largo de este proyecto se tratará en particular con un solo tipo de información, que se enmarca en un ámbito judicial y legal.

Entrando en dicho contexto, procesar documentos legales, a diferencia de otros tipos, presenta un desafío adicional ya que la información con la que normalmente se trata es de carácter sensible. Por lo tanto, considerando un posible escenario, donde una empresa haga uso de herramientas que manipulen este tipo de documentos, no puede haber pérdidas ni mucho menos filtraciones de los mismos.

Dentro de los documentos legales, uno de los elementos principales son los fallos judiciales. Estos se caracterizan por poseer una longitud considerable. Es por ello que existe otro elemento de gran relevancia asociado a ellos, los sumarios. Cada uno de estos es una descripción breve de los contenidos más importantes de su correspondiente fallo. La obtención de un sumario no es algo trivial, sino que, para su generación o construcción intervienen los sumariantes. Cabe destacar que este es un proceso laborioso que demanda mucho tiempo y esfuerzo. Por otro lado, un sumario, si bien es un resumen del fallo,

generalmente no se corresponde con uno o varios párrafos exactamente que aparecen en el original, sino que es considerado un resumen elaborado.

Bajo el contexto de un proyecto desarrollado en el CIDISI, convenio específico número 707 de transferencia de conocimiento, la empresa "Rubinzal y asociados S.A.", de ahora en más "La Empresa", demostró interés por la posibilidad de utilizar una herramienta de software que permita facilitar dicha tarea a los sumariantes.

Dicho esto, el **objetivo general** de este PFC consiste en **realizar un análisis de factibilidad de la aplicación de técnicas de minería de textos existentes para la generación automática de resúmenes de documentos legales.**

1.1. Objetivos específicos

Dentro del marco descrito y para establecer en detalle el alcance del PFC, se establecen los siguientes objetivos:

- Identificar el contexto y fundamento teórico relacionado a la sumarización de textos.
- Investigar las principales técnicas de resumen automático de textos.
- Preparar el conjunto de datos etiquetados ofrecido por La Empresa para ser utilizado con las técnicas seleccionadas.
- Evaluar las técnicas seleccionadas en función de los conjuntos de datos estandarizados.
- Seleccionar y definir si fuera necesario, métricas que pueden ser usadas para medir la adecuación de los resultados obtenidos.
- Evaluar la eficiencia de las métricas utilizadas para calificar las técnicas seleccionadas
- en el dominio particular de aplicación.
- Analizar la factibilidad de aplicación de las técnicas contempladas para documentos de naturaleza legal.

2. Marco Teórico

A continuación, se presentan los conceptos más importantes que se emplearon a lo largo de todo el PFC, con el fin de ofrecer los fundamentos teóricos adecuados.

2.1. Documentos legales

Un documento legal es un escrito público, emitido por una autoridad legítima pública de un país, que da fe o garantiza la autenticidad de un hecho, evento, actividad, acto, solicitud o afirmación. Dependiendo de la naturaleza del documento, puede cumplir con distintas funciones, pero todos cumplen como prueba o testimonio de aquello que declaran.

Los documentos legales a los que pretende dar tratamiento este proyecto son fallos judiciales y sus respectivos sumarios. Un fallo judicial es el resultado oficial de una demanda en un tribunal. Un fallo completo es una sentencia íntegra, tal como fue emitida por el Tribunal y de la cual se extractan los sumarios de jurisprudencia, uno por cada unidad de información relevante tratada en ella. Asimismo, un sumario es una descripción resumida de las doctrinas contempladas en la sentencia judicial, dado que en ella pueden tratarse diversas cuestiones jurídicas. La obtención de un sumario no es algo trivial, es realizado por personas especialistas (sumariantes) y no es necesariamente una copia textual de uno o más párrafos del fallo.

Los documentos y textos jurídicos, particularmente los legislativos, poseen unas características que les hacen aptos para su tratamiento informático estandarizado: la estructura, la articulación, las posibilidades de comparabilidad e interrelación, la referenciación entre ellos, la jerarquía, la complementariedad, etc.

Desde la tradición de la imprenta, la estructura de los documentos legislativos se despliega a través de títulos, capítulos, artículos, párrafos, números, etc. que organizan la información contenida en el texto legal de forma sistemática, junto con un conjunto de reglas que forman parte de la técnica legislativa (1). Está explícitamente expresado cómo deben realizarse, por ejemplo, las citas, las referencias bibliográficas, referencias a otras leyes, etc. Incluso hay tablas de referencias para el uso de abreviaciones y cómo deben ser escritas (2). Todas estas normativas tienen el objetivo de evitar incoherencias, contradicciones, redundancias, y demás.

Estas reglas y normativas para estructuras del documento en sí y para sus contenidos, son extremadamente útiles para tenerlas en cuenta a la hora de hacer un análisis del texto mediante software informático.

2.2. Resumen automático de textos

"Un resumen es una transformación reductiva de un texto fuente a un texto resumen por reducción de su contenido mediante selección y/o generalización de lo que es importante en el texto fuente." (3)

Esta definición permite entender cuál es la idea fundamental detrás de los generadores automáticos de resúmenes. Aquí se identifica un patrón clave y es que la entrada siempre contendrá más información que la salida. Por ello, el hecho de interpretar la forma de la entrada y la salida de dicho proceso deja solo pendiente definir la forma en que se va a procesar la información para obtener los resultados esperados.

La generación automática de resúmenes se define como el proceso de destilar la información más importante de una fuente o varias fuentes para producir una versión abreviada destinada a un usuario (o conjunto de usuarios) determinado y para una tarea (o conjunto de tareas) determinada. (4)

En conclusión, cuando se habla de resúmenes "automáticos", se hace referencia a máquinas que utilizan métodos para procesar información que percibe como entrada y generar nueva información que es recibida por un usuario como salida. En la *Figura 1* se presenta un esquema de alto nivel de un generador de resúmenes de textos, donde se muestran las tres etapas: análisis, transformación y síntesis.

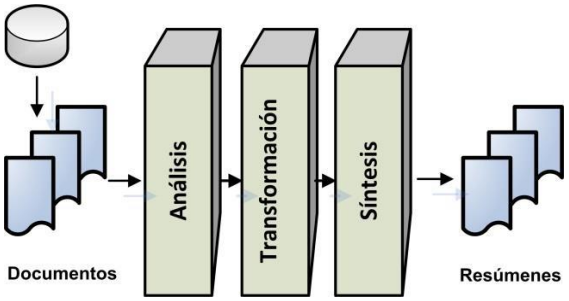

Figura 1. Arquitectura tradicional de un sistema de generación automática de resúmenes. (5)

Está claro que hay una gran variedad de técnicas existentes, es por ello que resulta de gran utilidad clasificar a las mismas. Existen distintos tipos de clasificaciones, pero la primera y más importante es según el tipo de salida que producen:

- Técnicas de resúmenes extractivas: El enfoque extractivo implica recolectar las frases y oraciones más importantes de los documentos. Luego combinarlas a todas ellas para así generar el resumen. Entonces, en este caso, cada oración y palabra que compone el resumen pertenece también al documento original. (6)
- Técnicas de resúmenes abstractivas: El enfoque abstractivo, por su parte, utiliza nuevas frases y terminologías, diferentes del documento actual, pero manteniendo siempre la misma idea. (6) Este enfoque demuestra ser más complejo que el previamente mencionado.

Existe, a su vez, otra clasificación que permite diferenciar a las técnicas de resumen según el tipo de entrada que reciben:

- Técnicas de resumen con un único documento: Un solo documento produce un resumen que proviene de un documento fuente y el contenido descrito trata del mismo tema (7). En otras palabras, el output se produce a través de una relación uno a uno (1:1).
- Técnicas de resumen con múltiples documentos: Por otro lado, los resumidores multi documentos tienen distintas fuentes con las cuales producen un solo resumen, pero siempre teniendo en cuenta que todas ellas tienen un tópico en común. Haciendo la misma analogía que en la clasificación anterior, se dice que el output aquí se produce a través de una relación muchos a uno (n:1).

En el caso de documentos legales, se identifican como las más adecuadas en términos de cantidad de inputs, a las técnicas de resumen con un único documento, ya que por cada documento que exista se espera un resumen distinto debido a que cada uno de ellos tratan individuos, causas y contextos distintos.

Las clasificaciones descritas anteriormente representan una forma de caracterizar a las técnicas teniendo en cuenta la entrada o la salida que las mismas perciben. Existen otras clasificaciones que están relacionadas al propósito que cumple cada técnica en particular y se detallan a continuación:

- Genéricas: Aquí el modelo no hace suposiciones acerca del dominio o contenido del texto y trata todos los inputs de manera homogénea.
- Específicas del dominio: La técnica utiliza conocimiento específico del dominio para generar un resumen más preciso.
- Query-based (o Basadas en consultas): Este tipo de técnicas solamente poseen un dominio que permite responder preguntas definidas en lenguaje natural acerca del texto que se percibe como entrada.

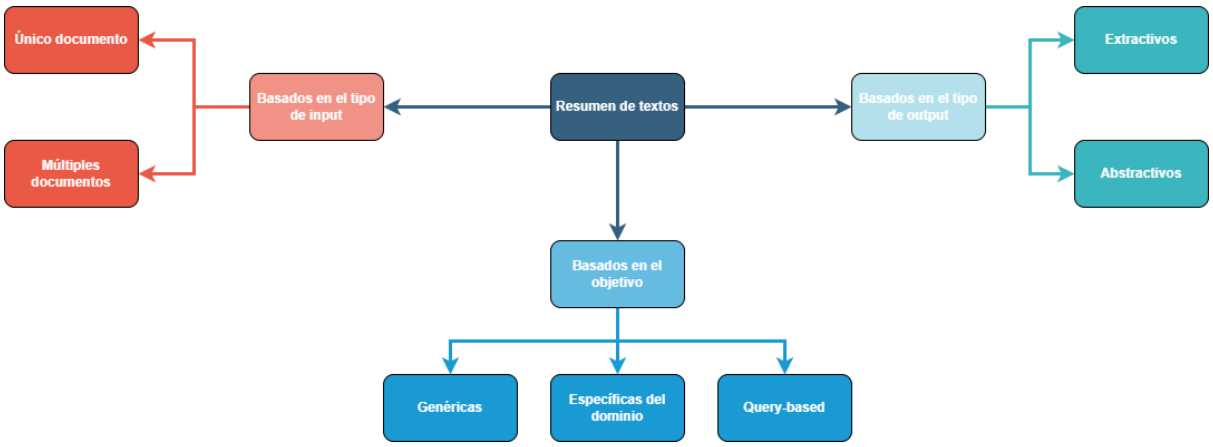

Figura 2. Tipos de técnicas de resumen de textos. (8)

2.3. Inteligencia Artificial

La inteligencia artificial (de ahora en más, IA) es un concepto no tan simple de definir. Esto se debe a que existen distintas formas de interpretar lo que se considera como inteligente. Es por ello que varios autores han tomado distintos enfoques para definirla.

Siendo más precisos, en (9) se plantean cuatro enfoques que definen a la IA. Estos enfoques brindan conceptos distintos en base al aspecto de la IA que se quiera definir. Existen dos aspectos claves dentro de los cuales es posible encapsular los cuatro enfoques mencionados.

El primero de ellos está asociado a la forma de medir el éxito. Una implementación de IA será más exitosa si su comportamiento se corresponde a alguno de los que se mencionan a continuación:

- Comportamiento similar al de un humano: Un sistema se comporta como un humano si piensa o actúa como uno de ellos.
    - Sistemas que piensan como **humanos**:
        - *"El nuevo y excitante esfuerzo de hacer que los computadores piensen como máquinas con mentes, en el más amplio sentido literal"*
    - Sistemas que actúan como **humanos**:
        - *"El arte de desarrollar máquinas con capacidad para realizar funciones que cuando son realizadas por personas requieren de inteligencia"*
- Comportamiento Racional: Un sistema es racional si hace lo correcto en función de su conocimiento:
    - Sistemas que piensan **racionalmente**:
        - *"El estudio de las facultades mentales mediante el uso de modelos computacionales"*
    - Sistemas que actúan **racionalmente**:
        - *"La Inteligencia Computacional es el estudio del diseño de agentes inteligentes"*

El segundo aspecto tiene que ver con lo que se espera de una IA:

- Aluden a procesos mentales y al razonamiento:
    - Sistemas que **piensan** como humanos:
        - *"La automatización de actividades que vinculamos con procesos de pensamiento humano, actividades como la toma de decisiones, resolución de problemas, aprendizaje"*
    - Sistemas que **piensan** racionalmente:
        - *"El estudio de los cálculos que hacen posible percibir, razonar y actuar"*
- Aluden a la conducta:
    - Sistemas que **actúan** como humanos:
        - *"El estudio de cómo lograr que los computadores realicen tareas que, por el momento, los humanos hacen mejor"*
    - Sistemas que **actúan** racionalmente:

- *"IA está relacionada con las conductas inteligentes en artefactos"*

En conclusión, la definición de IA queda sujeta a los propósitos y objetivos con los que se realice su implementación.

Asociando esto al resumen automático de documentos legales, se pretende que la IA implementada tenga un comportamiento más similar al de un ser humano, ya que la interpretación e identificación de partes más importantes de un texto son cuestiones muy subjetivas a la persona realice dichas tareas.

2.3.1. Aprendizaje Automático

El aprendizaje puede ser definido como el proceso de cambio y mejora en el comportamiento a través de la exploración y reconocimiento de nueva información en el tiempo. (10) Cuando dicho proceso es ejecutado por una máquina, se define como **aprendizaje automático** o bien como es conocido por su traducción al inglés: **Machine Learning (ML)**.

El aprendizaje automático consiste entonces en dotar a las computadoras con la habilidad de aprender basándose en datos y experiencia tal como los humanos lo hacen. (10)

Su principal objetivo es crear modelos que puedan ser entrenados para mejorar la toma de decisiones, percibir patrones complejos y encontrar soluciones a nuevos problemas, basándose en información previa. (10)

El aprendizaje automático a su vez puede clasificarse en 4 tipos, que facilitan su estudio y desarrollo:

Aprendizaje supervisado

En este tipo, los algoritmos trabajan con datos "etiquetados", intentado encontrar una función que, dadas las variables de entrada, les asigne la etiqueta de salida adecuada. (11) Existen dos tipos de problemas en donde este tipo de aprendizaje puede ser aplicado:

- Problemas de clasificación: Los datos se deben clasificar en categorías en base características de los mismos. Ejemplos: Identificación de dígitos, diagnósticos, detección de fraude de identidad.
- Problemas de regresión: Ciertas características de los datos se deben predecir a partir de las características disponibles al comienzo del entrenamiento. Ejemplos: Predicciones meteorológicas, de mercado, de expectativa de vida.

Aprendizaje no supervisado

El mismo tiene lugar cuando no se dispone de un conjunto de datos etiquetados para el entrenamiento. (11) Solo se conocen los datos de entrada, pero no existen datos de salida que

correspondan a una determinada entrada. De hecho, es frecuente y apropiado afirmar que para este tipo de aprendizaje no existe un conjunto de datos de entrenamiento. (10)

El proceso de aprendizaje ocurre haciendo uso de las relaciones y conexiones entre los datos. Se busca describir la estructura de los mismos para intentar encontrar algún tipo de organización que simplifique el análisis. (10) Algunos casos de uso para este tipo de aprendizaje se mencionan a continuación:

- Tareas de clustering: Se busca agrupar datos que son similares entre ellos, pero no se conocen agrupaciones inherentes a los mismos.
- Problemas de asociación: Se necesita determinar relaciones y conexiones entre los datos pertenecientes a un mismo conjunto.

Aprendizaje semi supervisado

La diferencia con el aprendizaje supervisado se basa en los datos etiquetados. En el supervisado, el conjunto de datos etiquetados es más grande que el conjunto de datos que debe ser procesado. Por el contrario, el semi supervisado cuenta con un conjunto de datos sin etiquetar mayor a los etiquetados, y en estos casos se utiliza una porción de ellos para tratar de deducir información y características acerca de los mismos. (10)

Aprendizaje por refuerzo

En este tipo, el modelo aprende a través de un sistema de retroalimentación, observando el mundo que lo rodea. La información de entrada es el feedback o retroalimentación que obtiene del mundo exterior como respuesta a sus acciones. Por lo tanto, el modelo aprende a base de ensayo-error. (11)

El objetivo del modelo es usar los caminos más cortos y correctos para alcanzar su objetivo. Cuando el mismo realiza acciones que lo llevan por estos caminos, obtiene una retroalimentación positiva. Lo contrario ocurre cuando se toman decisiones erróneas, la retroalimentación es negativa. (10)

2.3.2. Aprendizaje Profundo

El aprendizaje profundo (en inglés, Deep Learning) es un tipo o subdominio de aprendizaje automático. Como se mencionó en la *sección 2.3.1*, el aprendizaje es una forma o parte de la IA que permite a un sistema aprender de los datos en lugar de aprender mediante la programación explícita.

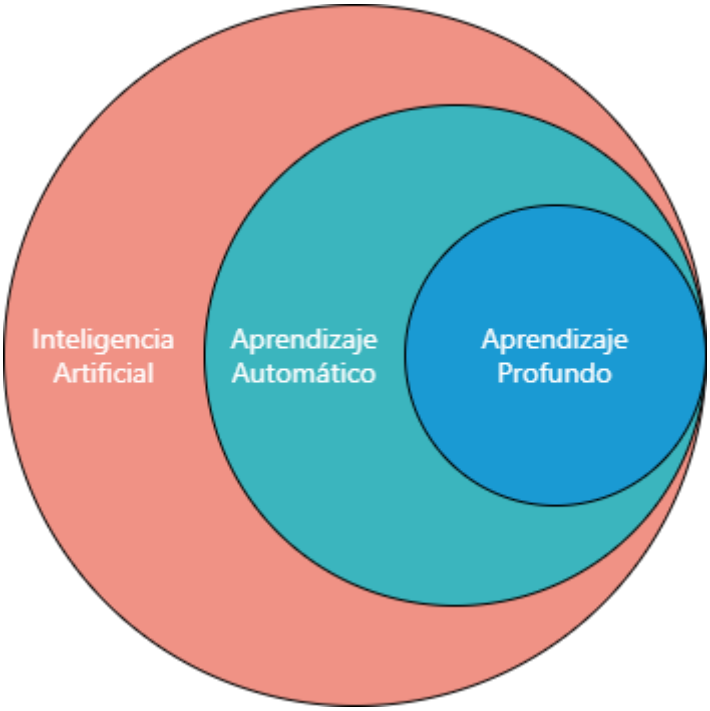

Figura 3. Dominios de la Inteligencia Artificial.

Por su parte, el aprendizaje profundo es un subconjunto de machine learning donde las redes neuronales (algoritmos inspirados en cómo funciona el cerebro humano) aprenden de grandes cantidades de datos. (12)

Los algoritmos de deep learning realizan una tarea repetitiva que ayuda a mejorar de manera gradual el resultado a través de ''deep layers'' lo que permite el aprendizaje progresivo. Este proceso forma parte de una familia más amplia de métodos de ML basados en redes neuronales. (12)

Debido a que los modelos de aprendizaje profundo procesan la información de manera similar al cerebro humano, se pueden aplicar a muchas tareas que realizadas por humanos. Algunos de los casos de usos más relevantes pueden ser las herramientas de reconocimiento de imágenes, el procesamiento del lenguaje natural y el software de reconocimiento de voz.

2.3.2.1. Restricted Boltzmann Machines

Una Boltzmann Machine (de ahora en adelante, BM) es un modelo de perceptrones que interactúan entre ellos donde cada uno actualiza su estado a lo largo del tiempo de forma probabilística dependiendo de los estados de sus nodos vecinos. (13) A su vez, este modelo era capaz de contar con una capa oculta de nodos de las mismas características antes mencionadas.

Este primer modelo propuesto tenía una adversidad: El aprendizaje de los parámetros para la BM era computacionalmente costoso.

Para reducir la complejidad de este proceso de aprendizaje, la conectividad fue restringida y de esta manera nacen las Restricted Boltzmann Machines (de ahora en adelante, RBM). La restricción consistió en la cantidad de interconexiones del modelo, eliminando las interacciones entre perceptrones de una misma capa, pero manteniendo la conectividad entre nodos de distintas capas.

Definiendo formalmente: RBM es un modelo basado en energía probabilística (del inglés, Energy Based Model) con una arquitectura de dos capas en la que las unidades estocásticas visibles están conectadas a las unidades estocásticas ocultas. No hay conexiones de nodos visibles a visibles u ocultos a ocultos (13). En la *Figura 4* se puede apreciar la diferencia entre ambos modelos.

Además de las capas visible y oculta, una RBM posee tiene un tercer componente fundamental: los pesos (del inglés, weights) de cada relación. De esta forma, el input 'i' provisto para la capa visible, se multiplica por los pesos asociados a la capa oculta. Los pesos 'w' y un cierto factor de desviación (del inglés, bias) 'b' son posteriormente utilizados en una función de activación. Dicha función es la encargada de producir el output para un nodo. En la *Figura 5*, es posible identificar dichos componentes.

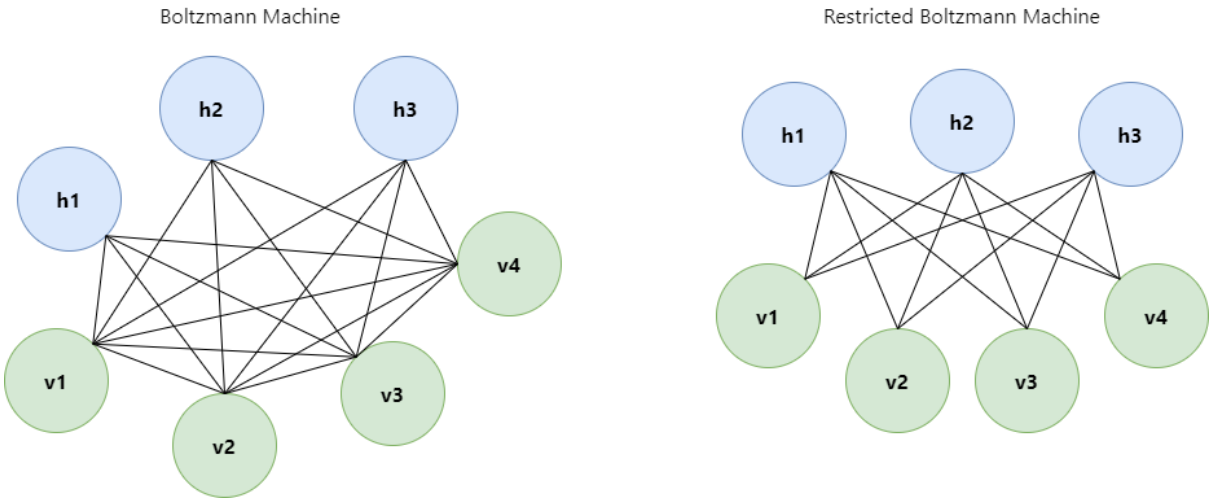

Figura 4. Diferencias entre Boltzmann Machines y Restricted Boltzmann Machines.

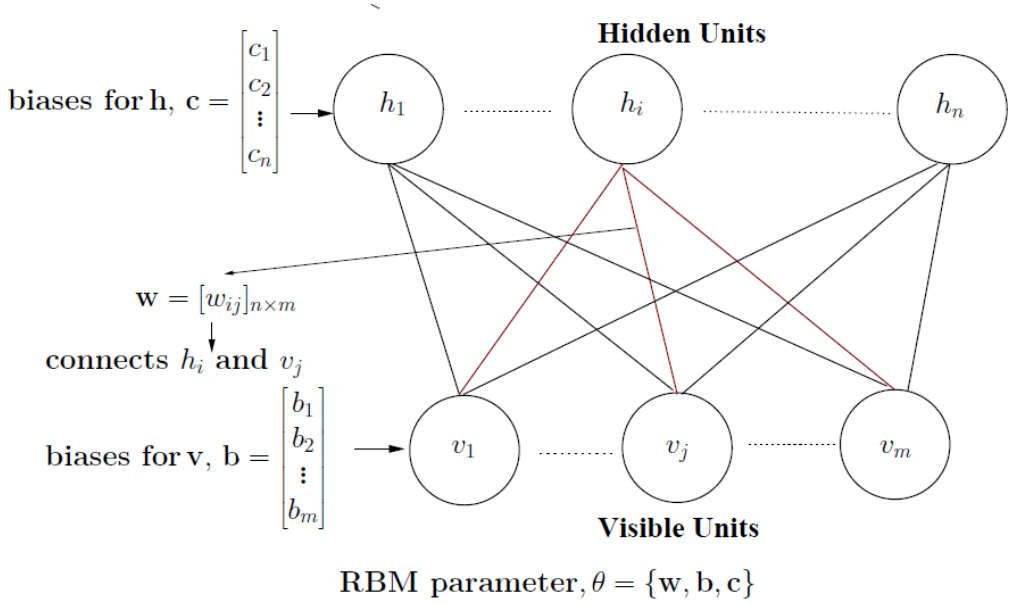

Figura 5. Componentes de una Restricted Boltzmann Machine. (13)

En la vida real, existen varios casos de uso para las RBM. A continuación, se presentan algunos de los más comunes:

- Reconocimiento de patrones: Son utilizadas para extraer características de un conjunto de datos y comprender posibles patrones. La interpretación de escritura a mano es un claro ejemplo, aunque también pueden tratarse patrones aleatorios.
- Sistemas de recomendación: Son ampliamente utilizadas en técnicas de filtración colaborativas donde intentan predecir cual es la mejor recomendación para un usuario final. Los más conocidos pueden ser sistemas de recomendación de películas o de libros.

2.4. Procesamiento del Lenguaje Natural

El Procesamiento del Lenguaje Natural (de ahora en más PLN) es un campo de investigación y aplicación de la Inteligencia Artificial que explora cómo pueden ser utilizadas las computadoras para entender y manipular el lenguaje natural, en formato texto o audio, y realizar así tareas útiles. Las investigaciones del PLN buscan generar conocimiento acerca de cómo los seres humanos entienden y utilizan el lenguaje, de manera de poder desarrollar herramientas y técnicas que hagan que los sistemas de computadora puedan comprender y manipular los lenguajes naturales y así llevar a cabo tareas deseadas (14).

El campo de PLN engloba una gran variedad de tópicos que involucran el procesamiento computacional y entendimiento del lenguaje humano. Diversos acercamientos para resolver los problemas planteados por este campo han sido tratados, desde el uso de estadísticas, probabilidad y, más recientemente, machine learning.

Si bien el PLN está en auge y ha experimentado avances muy significativos en las últimas décadas, especialmente en la más reciente, existen diversos desafíos inherentes a los lenguajes que todavía no se han podido sortear completamente. Los lenguajes humanos son increíblemente complejos, fluidos y plagados de inconsistencias. Por ejemplo, una misma frase u oración puede tener distintos significados dependiendo del contexto en el que se esté expresando. La ironía y los sarcasmos son otro buen ejemplo, ya que, si bien las palabras en una oración que expresa ironía pueden ser positivas, su significado es exactamente el contrario. (15)

Avances recientes en capacidad de procesamiento y paralelismo han hecho posible la aplicación de aprendizaje profundo (o deep learning) a cuestiones de PLN, utilizando redes neuronales con una gran cantidad de parámetros entrenables. Adicionalmente, el acceso contemporáneo a colecciones de datos masivos, facilitados por sofisticados procesos de recolección de datos, permiten el entrenamiento de tales arquitecturas profundas. En años recientes, investigadores y practicantes del PLN han aprovechado las oportunidades ofrecidas por el poder de las redes neuronales modernas con resultados muy prometedores (16). El uso de estas redes ha aumentado considerablemente en el último tiempo, lo que llevó a avances significativos en áreas esenciales del PLN, directamente aplicadas para lograr objetivos prácticos y útiles.

Diversos ejemplos de usos comunes del PLN pueden ser apreciados en la vida cotidiana moderna. Desde predictores de texto como se puede encontrar en Gmail o Google, que buscan predecir la palabra siguiente en base a las palabras anteriores en una oración, hasta asistentes virtuales, como Alexa o Siri, que ejecutan comandos simples basándose en instrucciones habladas verbalmente por parte de un humano. Dentro del contexto de este Proyecto, sin embargo, el PLN ofrece herramientas y algoritmos para la generación de resúmenes automáticos de textos que se describen a continuación.

2.4.1. Stopwords

Las stopwords (o palabras vacías de contenido) son las palabras más comunes en un lenguaje que no proveen un significado útil. Son palabras que modifican a otras palabras o sirven para indicar relaciones gramaticales. Ignorar este tipo de palabras puede mejorar considerablemente la eficiencia del resumidor, sin sacrificar el significado de la oración. Por ejemplo, utilizando como forma de representar a un documento el modelo de espacio vectorial, donde cada oración es un vector en el espacio y la proximidad entre vectores representa la semejanza entre las oraciones, si no se excluyen las palabras vacías, va a haber proximidad entre oraciones simplemente por que contienen gran cantidad de artículos en común, lo cual no es representativo en absoluto. (17)

Para eliminar este tipo de palabras se utilizan herramientas o librerías de PLN que contienen una lista de palabras vacías de contenido. Así, al ejecutar el algoritmo correspondiente se analiza el texto en busca de estas palabras, y cuando se las encuentra, simplemente se las suprime.

No todas las stopwords serán las mismas para todos los lenguajes, ni todas las herramientas de PLN utilizan la misma lista de ellas. También pueden construirse estas listas desde cero si así lo desea el usuario. Un acercamiento efectivo puede basarse en una lista de palabras conocidas para el lenguaje en cuestión, y agregar las que el usuario considere necesarias a medida que avanza en el proceso.

2.4.2. Tokenización

En PLN la tokenización es el proceso de convertir el texto en cuestión en porciones más pequeñas (de párrafos a oraciones, de oraciones a palabras) y darles un formato para que funcionen como entradas para la computadora. Se puede pensar al token como la unidad para procesamiento semántico.

Estos tokens ayudan a entender el contexto para el modelo de PLN. La tokenización es importante para lograr entender el significado del texto al analizar la secuencia los tokens, es decir, de la palabras u oraciones presentes en el texto.

2.4.3. Named Entity Recognition (NER)

El reconocimiento de entidades nombradas (Named Entity Recognition, NER por sus siglas en inglés) es el proceso de identificar información clave (entidades) en un texto y clasificarlas según un conjunto de categorías predefinido. Una entidad puede ser una palabra o un conjunto de ellas que se refiera a la misma cosa.

Cualquier procedimiento de NER sigue dos pasos básicos:

- Identificar entidades en el texto. Cada entidad puede estar compuesta por una palabra o varias. Cada palabra es un token, por lo tanto, cada entidad puede estar formada por uno o más tokens.
- Categorizar entidades. A cada entidad identificada en el paso anterior se le asigna una categoría. Algunas de las categorías más comunes para las entidades son: personas, organizaciones, lugares, fechas, medidas numéricas (peso, valor monetario, etc.), direcciones de correo, entre otras. Éstas dependen de cada modelo en particular, y debe ser entrenado para poder clasificar con eficacia.

2.5. Evaluación de técnicas

A la hora de evaluar la performance de una técnica de resumen automático, se pueden utilizar evaluaciones de las siguientes clasificaciones.

2.5.1. Evaluación humana

El análisis y puntuación de un determinado resumen generado es realizado por una persona, generalmente experta en el dominio del texto, y determina de manera subjetiva con que grado de asertividad el resumen cubre los temas principales del texto. Otros criterios que se pueden tener en cuenta son: determinar si el resumen responde preguntas claves del usuario, factores gramaticales y la no redundancia del resumen.

2.5.2. Evaluación automática

Como insinúa su nombre, las evaluaciones automáticas son aquellas que se realizan a través de un algoritmo y determinan a través de diferentes cálculos la calidad de los resúmenes generados. Los resultados de estos cálculos se expresan a través de **métricas** de evaluación, es decir, índices o medidas que evalúan distintos aspectos del texto para generar una calificación.

Evaluar la calidad de un resumen generado automáticamente es una tarea ambiciosa, debido a que las bases de comparación son variadas (18). Se puede comparar el resumen generado con el texto original, donde la calidad del resumen es juzgada, generalmente, en base a cuantas ideas del documento fuente están cubiertas en el resumen generado. O bien, se puede comparar con un resumen hecho por un humano (considerado un resumen ideal o resumen objetivo). Es posible ver como en este último caso un resumen ideal es subjetivo, ya que, al ser de naturaleza humana, un resumen puede ser considerado ideal por una persona, pero no así por otra.

2.5.2.1. ROUGE

Una de las métricas más reconocidas y utilizadas en el ámbito de la generación automática de resúmenes de textos se conoce como **ROUGE** (Recall-Oriented Understudy for Gisting Evaluation), y posee diversas variantes. Es, en esencia, un conjunto de métricas orientadas a la evaluación de textos generados automáticamente, comparándolos con un conjunto de referencias (generalmente hechas por un humano, consideradas ideales).

ROUGE hace uso de dos conceptos que vale la pena aclarar. El primero, denominado **recall** (o sensibilidad) se encarga de medir cuánto del resumen ideal se encuentra dentro del resumen generado.

Se obtiene de la siguiente manera:

$$recall = \frac{cantidad\ de\ palabras\ coincidentes}{cantidad\ de\ palabras\ en\ resumen\ ideal}$$

Fórmula 1.

En otras palabras, el recall representa la proporción de palabras en X (resumen ideal) que se encuentran también presentes en Y (resumen generado). (19)

Esta medida puede generar ciertos problemas. Supongamos que el resultado del recall para un par de resúmenes generado e ideal devuelve un 1, es decir, un 100%. Esto significa que todas las palabras del resumen ideal fueron encontradas en el resumen generado, dando la impresión de que el resultado es muy bueno. Sin embargo, puede darse el caso (y de hecho es algo frecuente) de que el resumen generado conste de, por ejemplo y dando valores arbitrarios, 7 palabras y el resumen generado conste de 100 palabras. Es esperable que dentro de las 100 palabras del resumen generado se encuentren las 7 del ideal. Pero 100 palabras contra 7 resulta excesivo, y si bien el recall sería de un 100%, la calidad del resumen generado es cuestionable. Aquí es donde entra el segundo concepto.

La **precisión** se encarga de medir la proporción de palabras *relevantes* dentro del resumen generado respecto del resumen esperado. Se calcula de la siguiente manera:

$$precisión = \frac{cantidad\ de\ palabras\ coincidentes}{cantidad\ de\ palabras\ en\ resumen\ generado}$$

Fórmula 2.

Al igual que el recall, a la precisión se puede interpretarla entonces como la proporción de palabras en Y (resumen candidato) que se encuentras también presentes en X (resumen de referencia) (19).

La precisión resulta crucial cuando se están tratando de generar resúmenes de naturaleza concisa. Es una medida que proporciona información sobre cuanto del contenido del resumen generado es realmente relevante, y cuál es la proporción de "relleno" o contenido no relevante presente. Queda claro entonces que estas dos medidas son complementarias, y apuntan a evitar los problemas que genera cada una respectivamente (20).

El problema mencionado para el recall puede darse con la precisión, pero a la inversa. Cuando se obtiene una precisión muy alta, es posible estar frente a un caso en donde el resumen candidato se encuentra contenido mayoritariamente en el resumen ideal, pero si los tamaños difieren ampliamente, siendo de tamaño menor el resumen candidato, la calidad del mismo es nuevamente cuestionable.

Con estos conceptos, es posible profundizar en lo que se conoce como ROUGE-N, que se define como superposición de N-gramas entre el sistema y los resúmenes de referencia. Por ejemplo:

ROUGE–1 se refiere a la superposición de unigramas (cada palabra individual) entre los resúmenes generados y los resúmenes de referencia.

ROUGE-2 se refiere a la superposición de bigramas (segmentos de dos palabras consecutivas) entre los resúmenes generados y los resúmenes de referencia.

Debajo se proporciona un ejemplo para ilustrar mejor estos conceptos.

Sumario generado por la máquina:

```
el gato fue encontrado bajo la mesa
```

Sumario ideal (generado por un humano):

```
el gato estaba bajo la mesa
```

En este caso y haciendo uso de ROUGE-1, el recall sería:

$$ROUGE\ 1_{recall} = \frac{cantidad\ de\ palabras\ coincidentes}{cantidad\ de\ palabras\ en\ resumen\ ideal} = \frac{5}{6} = 0.83$$

Fórmula 3.

Y la precisión:

$$ROUGE\ 1_{precisión} = \frac{cantidad\ de\ palabras\ coincidentes}{cantidad\ de\ palabras\ en\ sumario\ generado} = \frac{5}{7} = 0.71$$

Fórmula 4.

Para el caso de ROUGE-2, haciendo uso de los bigramas, el ejemplo se divide en:

Sumario generado dividido en bigramas:

```
el gato
gato fue
fue encontrado
encontrado bajo
bajo la
la mesa
```

Sumario ideal dividido en bigramas:

```
el gato
gato estaba
estaba bajo
bajo la
la mesa
```

Basados en estos bigramas, se puede calcular:

$$ROUGE\ 2_{recall} = \frac{cantidad\ de\ bigramas\ coincidentes}{cantidad\ de\ bigramas\ en\ resumen\ ideal} = \frac{3}{5} = 0.6$$

$$ROUGE\ 2_{precisión} = \frac{cantidad\ de\ bigramas\ coincidentes}{cantidad\ de\ bigramas\ en\ sumario\ generado} = \frac{3}{6} = 0.5$$

Éstas son dos de las métricas más conocidas dentro de ROUGE. Otra muy utilizada se conoce como **ROUGE-L.** Esta variante es bastante diferente de las anteriores, ya que lo que hace es buscar la cadena común más larga contenida dentro de ambos resúmenes. Es decir, se tiene en cuenta la similitud de estructura a nivel de oración de forma natural e identifica automáticamente los n-gramas de secuencia más largos que se superpongan.

La precisión y el recall pueden combinarse para otorgar una única métrica que permita evaluar de manera "global" los resultados obtenidos.

El conocido **F-Score** (o puntuación F en español) es el promedio entre la precisión y el recall, y se calcula de la siguiente manera (21):

$$F\ Score\ = 2 * \frac{Precisión * Recall}{Precisión + Recall}$$

Fórmula 5.

De la *Fórmula 5* se puede concluir que:

- El valor de la métrica será elevado si ambas, precisión y recall, son elevadas.
- El valor de la métrica será medio si alguna de las dos es elevada.
- El valor de la métrica será bajo si ambas, precisión y recall, son bajas.

2.5.2.1.1. ROUGE en Python

Para poder utilizar la métrica explicada, fue necesario transpolar los conceptos de la misma a una implementación en código. En *ROUGE: A Package for Automatic Evaluation of Summaries* (19) se presenta la implementación de ROUGE en forma de paquete para Python. De esta forma, es posible realizar evaluaciones haciendo uso de una implementación de la métrica con el aval científico correspondiente y disminuir el margen de errores a la hora de puntuar.

La librería ofrece funcionalidades simples de utilizar, que permiten evaluar directamente archivos JSON o bien variables dentro del código, pero lo fundamental es que la misma provee mecanismos para tres tipos de ROUGE: **Rouge 1, Rouge 2 y Rouge L**. Para cada uno de ellos obtenemos, al realizar una evaluación, el **Recall**, la **Precisión** y el **F-Score**.

A continuación, se presenta un breve ejemplo del output que se obtiene al evaluar con esta librería:


```
[
  {
    "rouge-1": {
      "f": 0.4786324739396596,
      "p": 0.6363636363636364,
      "r": 0.3835616438356164
    },
    "rouge-2": {
      "f": 0.2608695605353498,
      "p": 0.3488372093023256,
      "r": 0.20833333333333334
    },
    "rouge-l": {
      "f": 0.44705881864636676,
      "p": 0.5277777777777778,
      "r": 0.3877551020408163
    }
  }
]
```


En secciones posteriores se procede a mostrar en detalle las evaluaciones realizadas para cada una de las técnicas implementadas en el proyecto. Para dichas evaluaciones las métricas aplicadas son: ROUGE 1, ROUGE 2 y ROUGE L, calculando para cada una de ellas el recall, la precisión y el F1-Score.

Un detalle acerca de la implementación es que todas las evaluaciones correspondientes se persisten automáticamente una vez realizadas, en dos formatos distintos, para facilitar análisis posteriores:

- Un archivo JSON el cual detalla los resultados de cada métrica y cada tipo de ROUGE para cada documento.
- Gráficos de barras para cada métrica y cada tipo de ROUGE, agrupando todos los documentos en una misma figura.

3. Herramientas utilizadas

A continuación, se presentan las herramientas que resultaron fundamentales para el desarrollo del PFC. Las mismas se encuentran diferenciadas por tres categorías:

Comunicación y planificación:

- **Discord:** es un servicio de mensajería instantánea freeware de chat de voz VoIP, video y chat por texto. Resultó de gran utilidad para los investigadores a la hora de trabajar de modo virtual.
- **Microsoft Teams:** es una plataforma unificada de comunicación y colaboración que combina chat persistente en el lugar de trabajo, reuniones de video, almacenamiento de archivos (incluida la colaboración en archivos) e integración de aplicaciones. Dicha plataforma es un estándar para la virtualidad en la UTN Santa Fe. Por ello, fue utilizada para llevar a cabo todo tipo de reuniones con los directores del proyecto, como también para compartir documentos e información.
- **Asana:** es una plataforma web y móvil de gestión del trabajo diseñada para ayudar a los equipos a organizar, realizar un seguimiento y gestionar su trabajo. En ella se gestionaron las tareas y objetivos que los investigadores llevaban a cabo.

Repositorios remotos y documentación:

- **Microsoft Word:** Es un software de procesamiento o tratamiento de textos. Con este se realizó la escritura completa del informe del PFC.
- **OneDrive:** Es un servicio en la nube de Microsoft que permite almacenar y proteger archivos, compartirlos con otros usuarios y acceder a ellos desde cualquier lugar/dispositivo. El mismo fue utilizado para la gestión y almacenamiento de todos los archivos referentes al PFC.
- **Github:** GitHub es una plataforma para alojar proyectos utilizando el sistema de control de versiones Git. La misma fue utilizada para gestionar el código asociado al proyecto.

Programación: Debajo se listan las herramientas, lenguajes, frameworks y librerías mas relevantes asociadas al trabajo de desarrollo del PFC.

- **IDE:** Pycharm
- **Lenguaje:** Python 3.7
- **Frameworks y librerías:**
    - Pandas
    - Spacy
    - NLTK
    - PyTorch
    - Matplotlib

4. Metodología de trabajo

4.1. Desarrollo de la metodología

Para la elección y establecimiento de una metodología de trabajo, fueron tomadas como referencias las etapas del proyecto definidas en el Plan de Proyecto. Las mismas consisten en:

- Etapa 1: Revisión bibliográfica
- Etapa 2: Investigación de técnicas/estrategias
- Etapa 3: Preparación de datos
- Etapa 4: Implementación de técnicas
- Etapa 5: Análisis de Factibilidad

Alineándonos con esta estructura, adoptamos una estrategia de naturaleza cíclica para el desarrollo del proyecto, donde un ciclo consiste en un período de tiempo a lo largo del cual se llevan a cabo las primeras cuatro etapas. Cada técnica se investiga e implementa a lo largo de la ejecución de un ciclo, con las iteraciones que sean necesarias. La quinta etapa se trata de forma separada, ya que para su realización resultó indispensable completar en su totalidad las cuatro etapas anteriores.

4.2. Ciclos de la metodología

Un ciclo en la metodología propuesta consiste entonces en un proceso iterativo donde se llevan a cabo las primeras cuatro etapas.

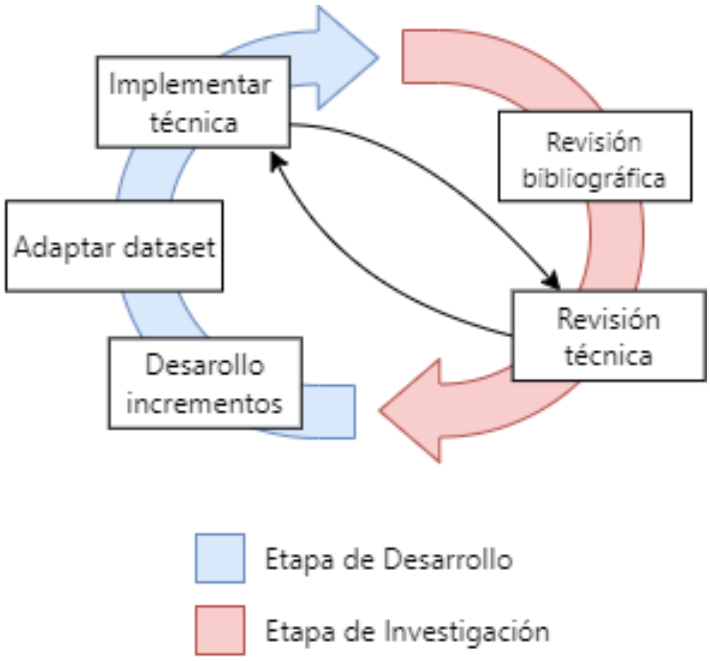

Figura 6. Ciclo de metodología.

El mismo comienza obligatoriamente con tareas relacionadas a la investigación, abordando principalmente las primeras dos etapas. Generalmente las mismas parten desde bibliografía más general o amplia para dirigirse hacia documentos cada vez más específicos. Para la ejecución de dichas actividades resultó fundamental mantener una comunicación activa, tanto entre investigadores como entre investigadores y directores del PFC.

Se llevaron a cabo dos tipos de iteraciones:

En primer lugar, iteraciones de una semana, donde los investigadores se reunían semanalmente para presentar material leído/investigado, detalles relevantes y pautar objetivos para próximas reuniones.

Por otro lado, se realizaron iteraciones de duración variable, dos o tres semanas, según la necesidad, en donde investigadores se reunían con los directores para realizar una presentación de los temas analizados, discutir acerca de los mismos y pautar objetivos para próximas iteraciones.

Estas iteraciones fueron realizadas de forma continua hasta llegar a un punto de conformidad con los directores, punto en el cual una técnica es aprobada para proceder con su debida implementación.

Continuando con el ciclo, la siguiente gran parte del mismo se destina a las tareas de desarrollo, para las cuales se adoptó una metodología inspirada en el desarrollo bajo incrementos.

La misma consiste en realizar tareas que estén ligadas a la implementación de la técnica seleccionada a partir de la primer parte del ciclo, alineándose así con las etapas tres y cuatro previamente mencionadas. Es destacable mencionar que a pesar de que esta fase esté ligada más que nada a tareas de desarrollo, se realizan también tareas de investigación como las de la primera fase, aunque de una índole distinta, más ligadas a la tecnología y técnica en cuestión.

El desarrollo en incrementos radica en tomar el objetivo completo del sistema y dividirlo en partes o incrementos. Cada incremento busca lograr un objetivo más pequeño y susceptible de ser cumplido en el tiempo que se estime necesario. La suma de todos los incrementos produce el sistema final (22). En la siguiente imagen se puede apreciar una ilustración del modelo incremental, con sus secuencias lineales que suceden a medida que avanza el calendario de actividades. Cada secuencia produce un incremento de software susceptible de entregarse o evaluarse.

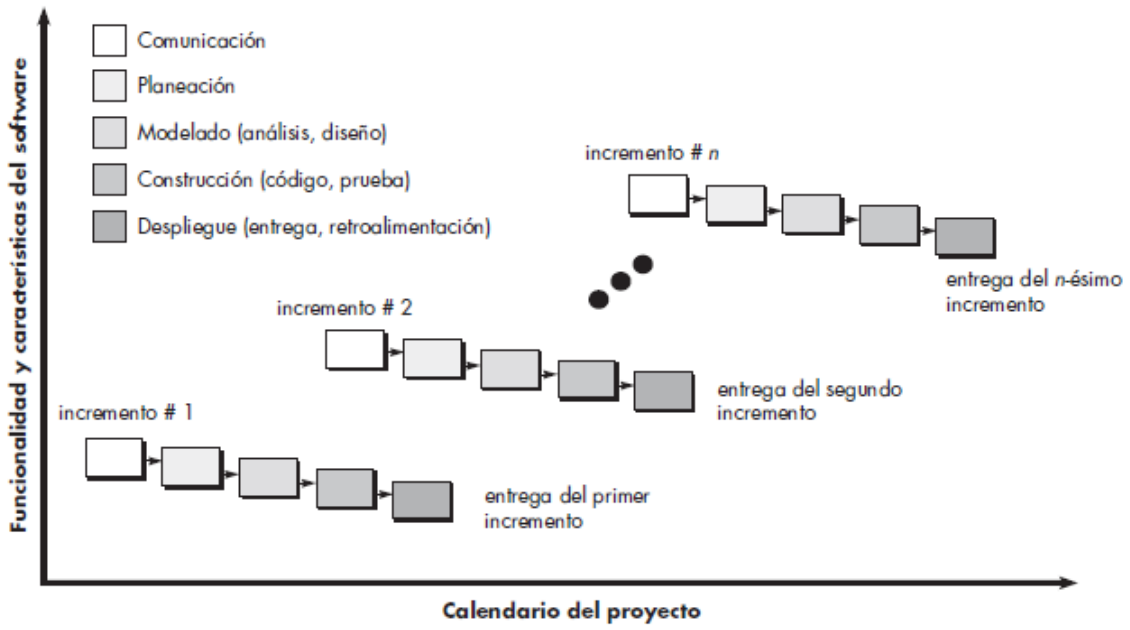

Figura 7. Modelo Incremental (22)

Para el contexto de este proyecto, un incremento consiste en una funcionalidad asociada específicamente a la técnica implementada en cuestión o bien una funcionalidad de carácter más integral (leer dataset, limpiar dataset, procesar dataset, obtener métricas, graficar resultados) que aporte al desarrollo de un sistema completo y pueda cumplir las expectativas de implementación. Se mantuvo una mentalidad de desarrollo basada en la reutilización de

componentes, de forma que, por ejemplo, el preprocesamiento del dataset sea el mismo, independientemente de la técnica en cuestión.

La duración de cada incremento se estableció por defecto en 15 días. Cumplido este plazo se realizaban reuniones con los directores para validar los avances y el código desarrollado. Cabe aclarar que hubo incrementos en los que este período de tiempo no pudo ser respetado por distintas cuestiones, lo que se resolvió comunicando los inconvenientes a los directores y pautando nuevas fechas de entrega para continuar con el mismo.

Paralelamente, en cada ciclo se realizó una actividad constante e indistinta a cada etapa: la documentación. Esta se iba elaborando sobre la bibliografía investigada o bien sobre lo implementado para mantener así consistencia en cada progreso realizado.

Resulta necesario mencionar que cada ciclo se repitió al menos una vez para cada técnica. Es decir, se pasaba de la investigación al desarrollo, y luego de vuelta a la investigación (para reforzar conceptos o resolver dudas emergentes) para posteriormente volver a seguir con el desarrollo. Esto se refleja en la naturaleza iterativa de los ciclos, y en las flechas internas que se aprecian en la *Figura 6* (de Revisión de Técnica a Implementación de técnica y viceversa).

Trabajar bajo incrementos resultó particularmente útil debido a que, al principio del proyecto, si bien estaba claro el objetivo final (disponer de un prototipo de software que permita evaluar objetivamente distintas técnicas de resumen automático de textos legales a partir del mismo conjunto de datos), los pasos para llegar a ella resultaban difíciles de contemplar. No era trivial trazar un camino lineal para llegar al destino, debido a la falta de familiaridad con los temas a tratar, dificultad para estimar plazos, herramientas de desarrollo que conllevan su propia curva de aprendizaje, etc.

El enfoque orientado a incrementos permitió que, en cada momento de la fase de desarrollo, la atención esté situada en implementar correctamente una funcionalidad particular, y hacer que la misma esté disponible para usarse en futuros incrementos. De esta manera, al obtener sucesivamente nuevas partes del sistema completo, éste resultaba cada vez más fácil de ver y contemplar.

Una vez que transcurrieron los primeros incrementos ya se contaba en el ambiente de trabajo con funcionalidades del software que funcionaban correctamente. Esto hizo que las mismas se encuentren disponibles para su uso en incrementos posteriores, y el desarrollo se haga más ágil.

La metodología desarrollada a lo largo del Proyecto fue particularmente desafiante. Debido a haber comenzado durante 2021, año a lo largo del cual continuó el período de cuarentena por COVID-19, los medios de comunicación tanto entre investigadores como entre investigadores y directores para validación y retroalimentación fueron puramente virtuales. Cabe destacar que el uso de herramientas de planificación, división y asignación de tareas

(como Asana) resultaron especialmente útiles para llevar un registro y control sobre avances, tareas pendientes, detectar cuellos de botella, etc. Asimismo, herramientas de gestionado de versiones (Git para código, OneDrive para documentos) también fueron claves para hacer posible un uso compartido y paralelo de recursos evitando conflictos.

Además, al tratarse de un proyecto de investigación, resultó fundamental contar con flexibilidad a la hora de responder a cambios y ajustes en las metas parciales y tiempos estimados. Todas estas cuestiones fueron posibles de manejar y gestionar gracias a las herramientas mencionadas.

5. Desarrollo del proyecto

5.1. Estado actual de los resúmenes automáticos de textos

Para dar comienzo al proyecto, fue necesario realizar una recopilación de información que nos permita comprender conceptos generales de los resúmenes automáticos como también los estudios y trabajos que la comunidad científica haya realizado.

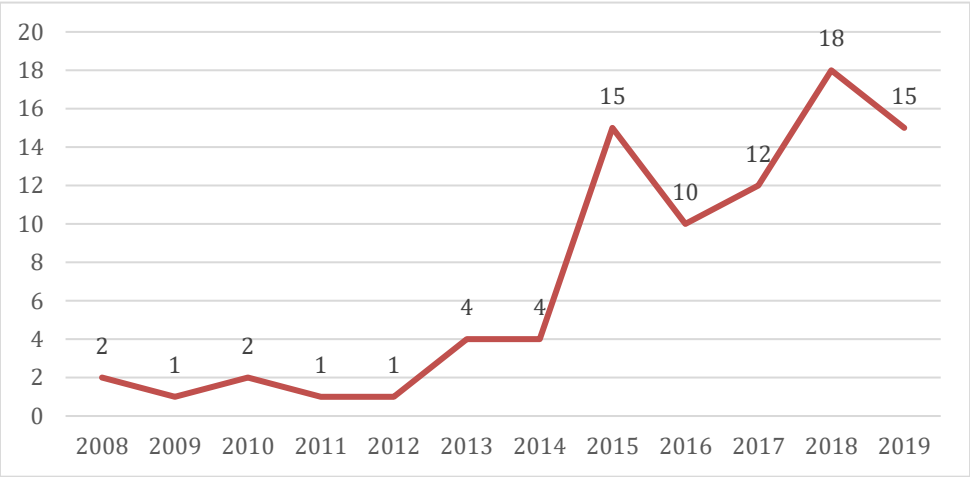

Figura 8. Distribución de estudios a lo largo de 10 años.

Se puede confirmar que, a lo largo de los años, los estudios relacionados a la extracción automática de resúmenes y el procesamiento del lenguaje fue incrementando. En la *Figura 8* se puede observar la cantidad de investigaciones publicadas alrededor de este tópico entre el año 2008 y 2019 (7).

Podemos observar del gráfico una tendencia alcista que nos indica que el campo de estudio está en pleno crecimiento y nos genera cierta motivación para continuar indagando y estudiando al mismo. En la actualidad existen diversos métodos para generar resúmenes, los cuales pueden agruparse según su tipo:

Las técnicas extractivas que resultaron más relevantes para el análisis y desarrollo de este proyecto fueron:

- Utilización de modelos semánticos y estadísticos: Frecuencia de palabras, palabras claves, ubicación, títulos, son algunas de las características utilizadas en este enfoque. (23)
- Métodos basados en Machine Learning: Técnicas de ML con aprendizaje supervisado o no-supervisado se incluyen aquí. La principal diferencia reside en el conjunto de datos utilizado. Para las primeras el conjunto de datos está etiquetado, mientras que pare el resto no. (23)

- Métodos basados en grafos: Aquí la construcción de un grafo a partir del texto, y la deducción de una matriz de similaridad a partir de un algoritmo son la clave para obtener un resumen automáticamente. El más relevante de esta clase es el algoritmo TextRank.
- Métodos probabilísticos: El objetivo de estos métodos es encontrar oraciones destacadas, conceptos claves y relaciones entre estos conceptos a través de modelos probabilísticos. Los más conocidos son los modelos Bayesianos y el Modelo Oculto de Markov (Hidden Markov Model). (23)
- Métodos basados en Redes Neuronales: Este enfoque consiste en aprender acerca del texto a través del uso de redes neuronales. La utilización de deep learning resulta en una de las técnicas más interesantes. (23)

Por otro lado, existen técnicas de tipo abstractivas que también fueron consideradas relevantes para llevar a cabo el proyecto:

- Métodos Encoder – Decoder: La estrategia en estos casos consiste en leer y codificar el input, dándole una longitud previamente conocida, y luego decodificar dicho output para obtener así un resumen. (23)
- Utilización de modelos de jerarquía: La idea principal en este tipo de técnicas se basa en comprender el texto y darle una estructura jerárquica para luego, a partir de ella, generar el resumen correspondiente.

5.1.1. Resumen automático de documentos legales

Los documentos legales, al igual que otros tipos de textos como son las noticias, textos literarios, entre otros, tienen características propias que hacen a las distintas técnicas eficientes en mayor o menor medida.

En la actualidad, varios estudios se realizaron en torno a este tópico. Por ello, en la *Tabla 1* se presentan los estudios más relevantes que fueron analizados a lo largo del proyecto.

En esta etapa de investigación fue posible destacar la tendencia de la comunidad hacia tecnologías basadas en Machine Learning. Este hecho permitió también poder enfocar este proyecto hacia un horizonte más innovador.

Estudio/Técnica/Publicación	Métricas utilizadas	Año
LetSum, an automatic Legal Text Summarizing system	ROUGE-2, ROUGE-L (Solo recall)	2004
pytextrank: Module for TextRank for phrase extraction and text summarization	ROUGE-1, ROUGE-2, ROUGE-L	2020
SummCoder: An unsupervised framework for extractive text summarization based on deep auto-encoders	ROUGE-1, ROUGE-2, ROUGE-L	2019
Effective deep learning approaches for summarization of legal texts	ROUGE-1, ROUGE-2, ROUGE-L	2019
Combining Different Summarization Techniques for Legal Text	ROUGE-1	2012
Automatic Text Summarization with Machine Learning - An overview	-	2020
Automatic Text Document Summarization Based on Machine Learning	ROUGE-2	2015
Automatic text summarization using fuzzy inference	F1-Score	2017
Combining Different Summarization Techniques for Legal Text	ROUGE-1	2012
Text Summarization Using Unsupervised Deep Learning	ROUGE-1, ROUGE-2, ROUGE-SU4	2016

Tabla 1. Investigaciones analizadas a lo largo del proyecto.

En la siguiente *Figura 9* extraída de (7) se observa la tendencia previamente mencionada. En ella se muestran el tipo de técnicas utilizadas para resumir documentos a partir de una recopilación de 87 publicaciones.

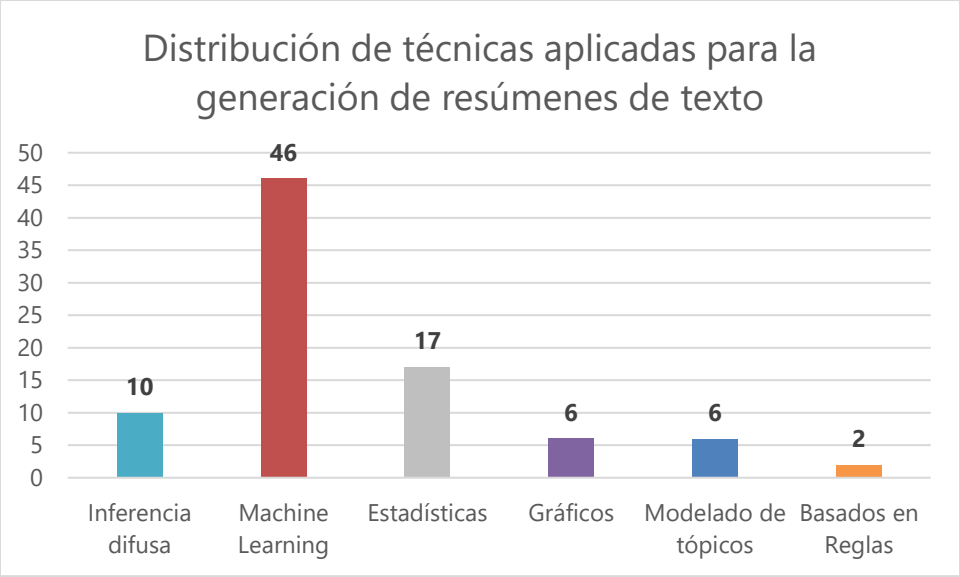

Figura 9. Distribución de técnicas aplicadas para la generación de resúmenes de texto.

Una de las conclusiones que se obtienen luego de realizar un análisis del estado del arte del resumen automático de textos es que las técnicas de resumen extractivas son aún las más elegidas a la hora de atacar diversas problemáticas asociadas al resumen automático. También es posible observar que uno de los puntos claves para producir buenos resúmenes es el preprocesamiento, es decir, identificar palabras claves, frecuencia de aparición y similaridad de palabras, posicionamiento y longitud de oraciones, y cuestiones semánticas.

En cuanto a los tipos de técnicas, las de machine learning suelen ser de las más elegidas en conjunto con técnicas basadas en estadísticas. Los enfoques estadísticos muchas veces suelen ser combinados con técnicas de machine learning o lógica difusa. (7)

5.2. Elección de estrategias - Fundamentación

En secciones anteriores se definen conceptos asociados a las formas de generar resúmenes automáticamente, se revisan gran variedad de técnicas propuestas por la comunidad científica y también se mencionan métricas a través de las cuales un resumen automático puede ser evaluado.

Todo este proceso de revisión y aprendizaje permitió construir un criterio a la hora de seleccionar técnicas para implementar a lo largo del proyecto.

Dos técnicas fueron seleccionadas para su posterior implementación, ambas coincidiendo en el tipo de salida que producen: **Extractiva**.

La elección de técnicas de resumen extractivas se ve justificada por un previo análisis realizado con el conjunto de datos brindado. El objetivo del mismo es determinar si los sumarios ideales pertenecientes al dataset se realizan en su mayoría a partir de extracciones de fragmentos de textos.

Para llevar a cabo dicho análisis se decidió utilizar conceptos y herramientas inherentes al proyecto. Por ello, se calculó el **recall (o sensibilidad)** entre los textos que se utilizarán como fallos judiciales (que representa el input para las técnicas) y su resumen ideal (o bien, target). Como se mencionó previamente y para este estudio en particular, **el recall indica qué proporción de un texto A (el resumen ideal) está contenida en un texto B (el fallo judicial a partir del cual fue generado el sumario ideal)**. De esta forma, es posible inferir que, si el recall entre estos dos documentos es alto, los sumarios se realizan extrayendo fragmentos textuales del fallo judicial, lo cual a su vez implica que el tipo de técnicas extractivas es el más adecuado para generar este tipo de resúmenes automáticamente.

Una cuestión a considerar es que dicho análisis solo indica si los sumarios contienen o no extracciones textuales del fallo, pero, en caso de contenerlas, no brinda ningún tipo de información acerca de cómo se realizan las mismas.

A la hora de calcular la sensibilidad, se utilizaron los tres tipos de ROUGE previamente mencionados. De manera más explícita, el cálculo realizado se presenta en la siguiente ecuación:

$$recall = \frac{cantidad\ de\ palabras\ coincidentes\ entre\ el\ fallo\ y\ el\ sumario}{cantidad\ total\ de\ palabras\ en\ el\ sumario}$$

Fórmula 6.

El análisis se realizó utilizando un script de Python encargado de leer los fallos judiciales con sus respectivos sumarios, calcular el recall entre ellos y mostrar gráficamente los resultados del mismo.

El muestreo fue realizado con la totalidad del conjunto de datos, buscando resultados que avalen o refuten la hipótesis. El mismo consta de **50.792 casos**.

A continuación, se presentan los resultados como fueron obtenidos: en 5 lotes de 10000 documentos cada uno, y uno restante de 792, división que hizo su procesamiento más ágil:

- Lote Nº 1

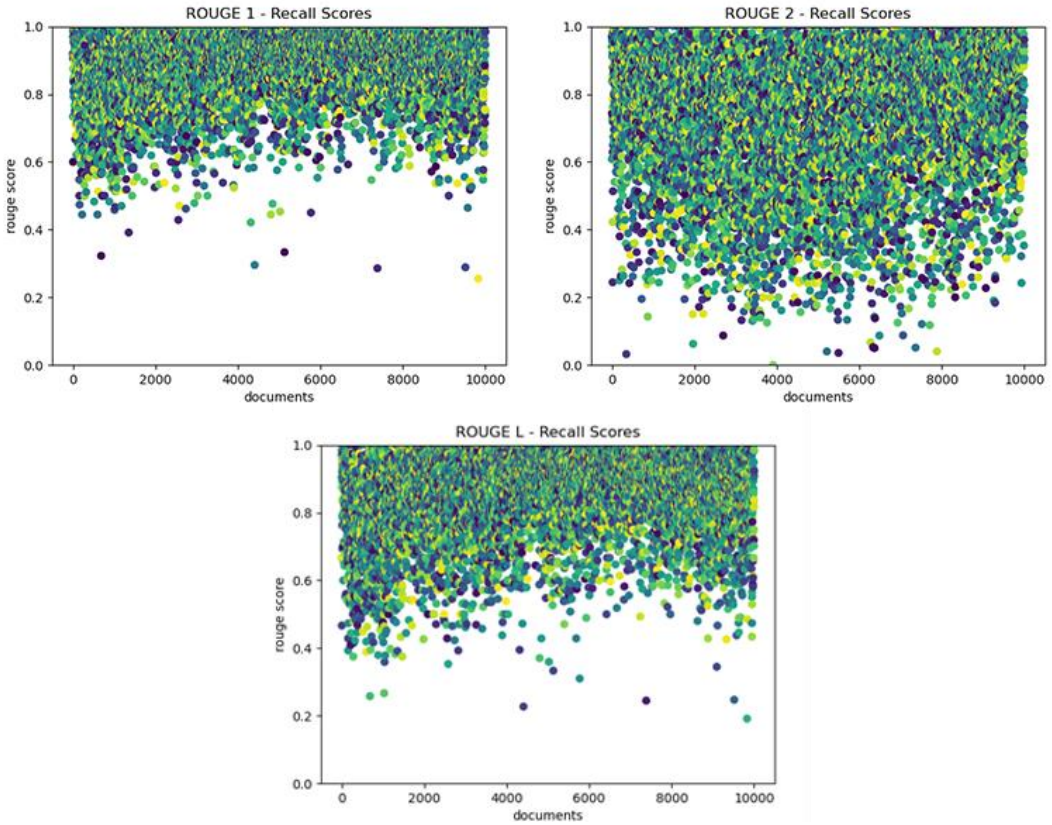

Figura 10. Distribución de puntuación de recall, Lote 1.

- Lote Nº 2

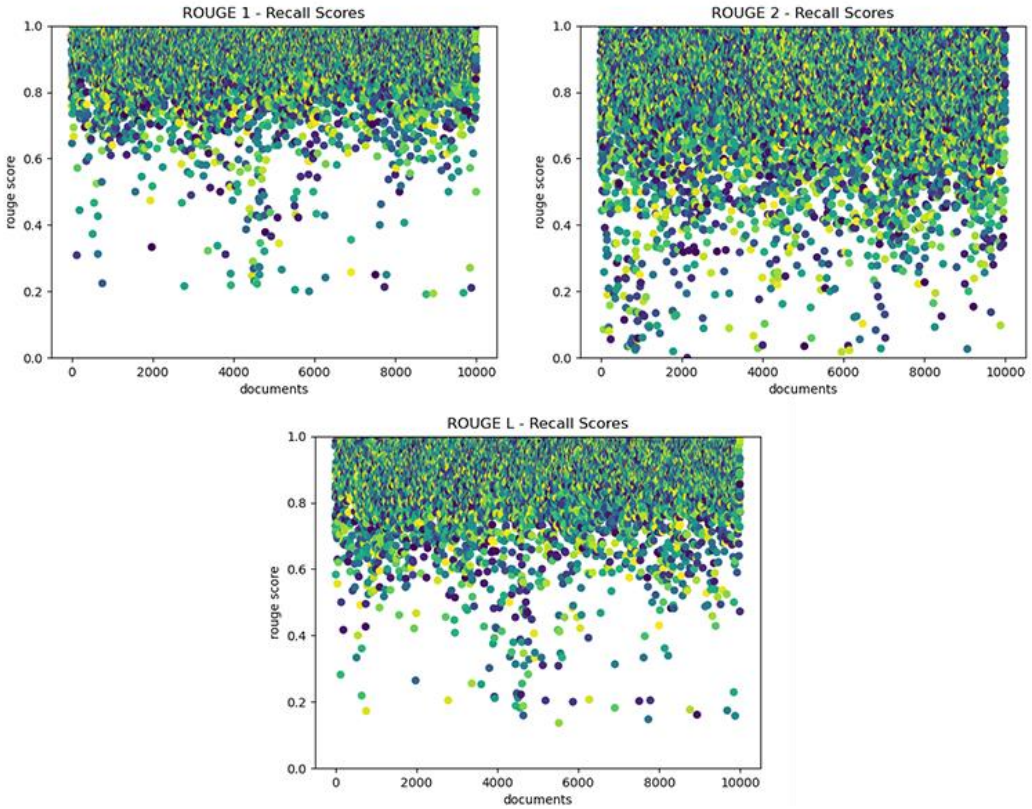

Figura 11. Distribución de puntuación de recall, Lote 2.

- Lote Nº 3

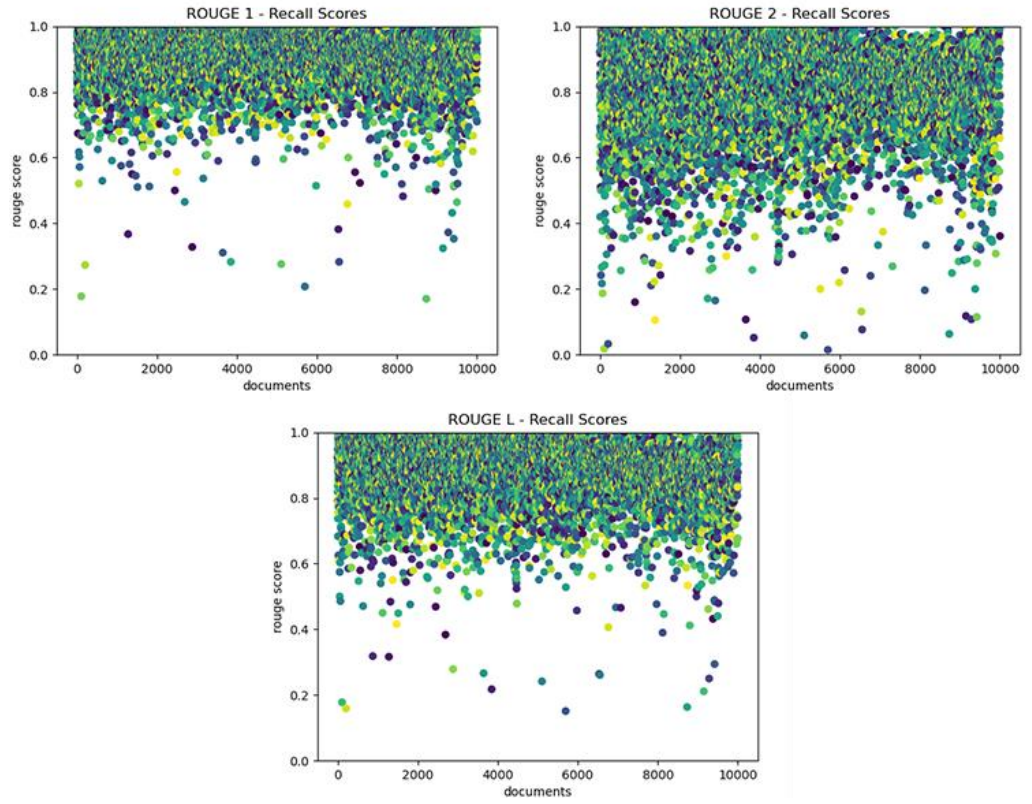

Figura 12. Distribución de puntuación de recall, Lote 3.

- Lote Nº 4

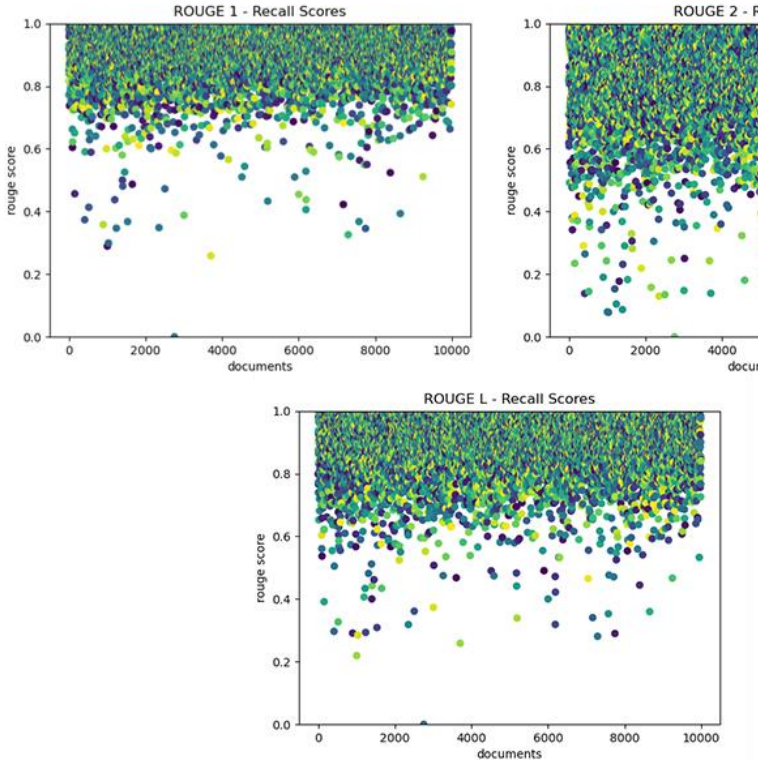

Figura 13. Distribución de puntuación de recall, Lote 4.

- Lote Nº 5

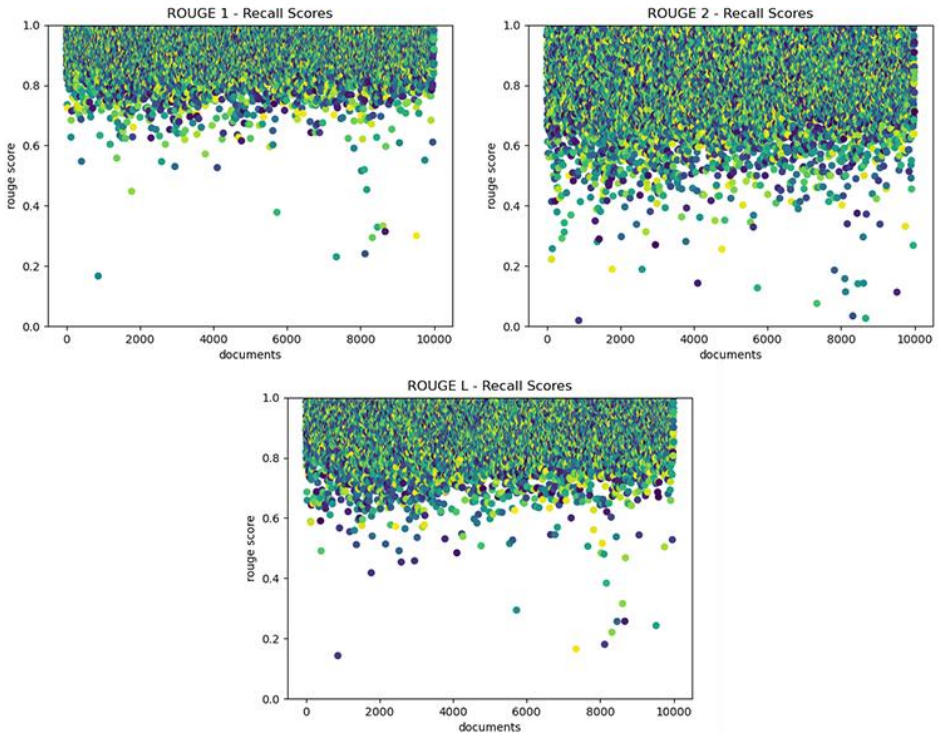

Figura 14. Distribución de puntuación de recall, Lote 5.

- Lote Nº 6

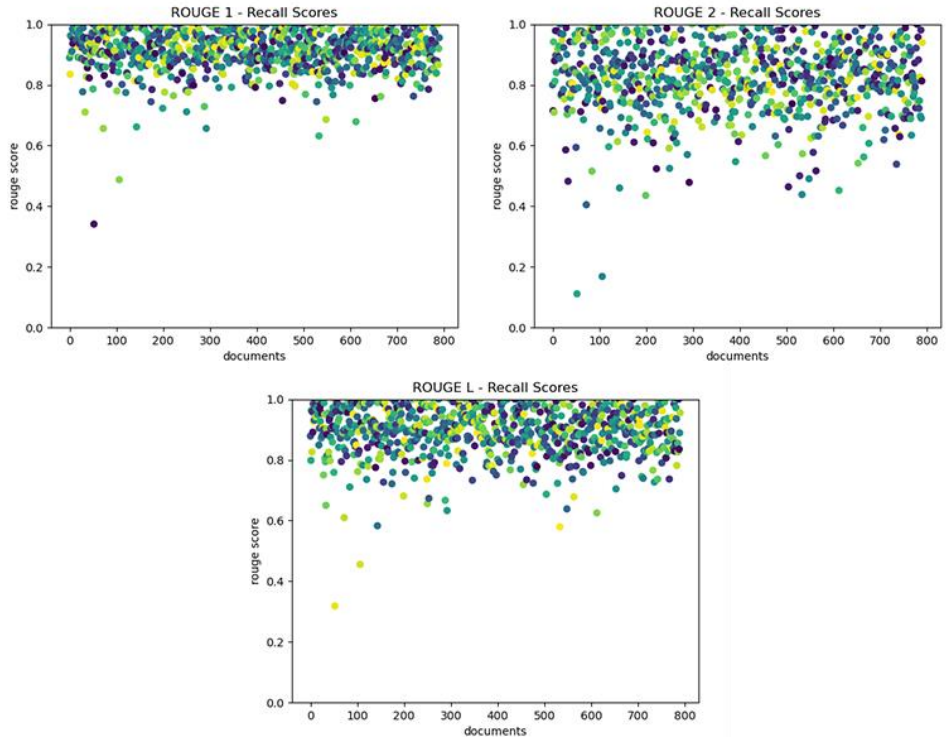

Figura 15. Distribución de puntuación de recall, Lote 6.

Una vez presentados los gráficos, es posible interpretarlos a través de lo que representa ROUGE y cada una de sus variantes:

ROUGE 1: permite conocer solo la proporción de palabras coincidentes entre fallo y sumario. Esta medida por sí sola no garantiza que las palabras presentes en el sumario sean una réplica de fragmentos del fallo, ya que pueden tener un orden completamente distinto, pero aun así coincidir. A pesar de esto, el alto valor de esta métrica es un buen indicio y requiere analizar las variaciones restantes de ROUGE. Los documentos que arrojen un valor pequeño para esta medida aseguran que el sumario no fue generado extractivamente y no requieren posterior análisis.

ROUGE 2: determina la proporción de bigramas (secuencias de dos palabras) coincidentes entre fallo y sumario. Un alto valor de esta métrica implica que las palabras en cuestión son consecutivas. Si en el sumario se encuentran palabras extra (conectores, artículos, etc.) se produce una disminución en el valor de la medida.

ROUGE L: da a conocer la proporción de secuencias de palabras coincidentes entre fallo y sumario. Por ejemplo: si todas las palabras que coinciden al medir con ROUGE 1 están en secuencia en el texto, entonces el valor de ROUGE L será muy similar al de ROUGE 1. Cabe aclarar que, que las palabras estén en secuencia implica que siguen el mismo orden de aparición, pero no implica que sean consecutivas.

A partir de este procesamiento, es posible brindar un resultado promedio para el recall de cada uno de los lotes como también un promedio para el recall global:

Lote	Recall		
	ROUGE 1	ROUGE 2	ROUGE L
1	0.876	0.724	0.852
2	0.899	0.789	0.881
3	0.899	0.805	0.881
4	0.903	0.804	0.884
5	0.902	0.801	0.881
6	0.910	0.821	0.892
TOTAL	**0.898**	**0.790**	**0.878**

Tabla 2. Puntuaciones promedio de recall.

Se puede concluir que los sumarios ideales, en general, no fueron generados cien por ciento extractivamente. Sin embargo, es posible afirmar, observando los valores promedio, que gran proporción de ellos se construyeron a partir de extracciones. Esto valida la hipótesis planteada al comienzo de la sección, y convierte a las técnicas extractivas en el enfoque más apropiado para este contexto.

Adicionalmente, al investigar acerca enfoques abstractivos, se pudo observar la escasa puntuación que dichas estrategias obtienen tratando de resumir documentos de diversas naturalezas. Al adicionar esto a la gran complejidad que se identificó en los documentos de carácter legal, sea por su estructura, formalidad o bien el mismo idioma, se decidió dejar de lado las mismas y continuar adelante con enfoques extractivos. (7)

5.3. Conjunto de datos

En cualquier proyecto de investigación, el dataset o conjunto de datos es una de las piezas claves para el éxito de un proyecto. Es por ello que la etapa tres del proyecto estuvo enfocada específicamente en estudiar y preparar los datos a ser utilizados.

Dado que este proyecto se desarrolló bajo el contexto del CIDISI, en un convenio específico con La empresa, el conjunto de datos fue provisto por esta última.

El mismo se encuentra comprimido en un archivo .rar, en donde se listan una gran cantidad de fallos judiciales (exactamente 50.792) con sus respectivos sumarios, cada uno encapsulado en un archivo de formato JSON.

Para trabajar de manera más dinámica con este gran conjunto de datos, se codificó un script que construye, bajo un criterio previamente definido, un dataset de tamaño N en un único archivo JSON, apto para luego ser procesado por los distintos módulos desarrollados.

Una de las ventajas logradas con este script es la generación de subconjuntos de fallos seleccionados aleatoriamente. Al trabajar con este criterio, evitamos trabajar siempre con los mismos casos, lo que implicaría perder una visión general del problema o bien sesgar los resultados.

Otro aspecto beneficioso es que se puede ajustar la cantidad de casos a procesar según lo que se está evaluando. De esta manera se controlan los recursos y tiempos destinados al procesamiento.

El archivo final generado por dicho script se estructura de la siguiente manera:

```
{
    "lines":
    [
        {
            "text_id": 00001,
            "text": "Fallo Judicial 00001",
            "summary": "Sumario 00001"
        },
        .
        .
        .
        {
            "text_id": 99999,
            "text": "Fallo Judicial 99999",
            "summary": "Sumario 99999"
        }
    ]
}
```

Donde:

- Text_id: Representa un identificador único para el documento.
- Text: Representa el fallo judicial completo.
- Summary: Representa el sumario generado por un humano, a ser considerado ideal, asociado al fallo judicial.

Cabe aclarar que se decidió utilizar archivos JSON en vez de CSV para aprovechar el formato original en el que los datos fueron brindados y minimizar los riesgos al realizar cualquier tipo de transformación.

5.3.1. Datasets y documentos legales

A medida que se fueron analizando y estudiando los datos, se identificaron algunas particularidades que en nuestro caso están ligadas a la naturaleza de los documentos legales. Pudimos detectar que estos escritos poseen ciertas características que los hacen particularmente complejos al ser procesados por un software. Entre ellas, podemos destacar:

- **Sensibilidad de los datos:** Al tratarse de fallos judiciales, la información contenida es sumamente sensible, por lo que la manipulación de estos debe realizarse de manera cuidadosa. Como regla de trabajo se estableció almacenar los datos aportados por el cliente de manera local a lo largo de todo el proyecto. De esta manera logramos, no solo respetar un acuerdo con el cliente, sino también evitar cualquier peligro que el Internet puede suponer cuando de información se trata.

- **Estructuras:** A lo largo de las tareas de investigación, nos encontramos distintos conjuntos de datos, de distintas naturalezas y países de origen. Resulta relevante mencionar que, al tratarse de documentos legales, muchas veces se cuenta con estructuras léxicas. Esto quiere decir que ciertos tipos de textos, al ser de carácter legal, tienen que respetar una serie de pautas al ser desarrollados, lo cual a la hora de preprocesarlos aporta una guía y facilita la tarea. En Argentina, los fallos judiciales suelen seguir una estructura o jerarquía para organizar la información contenida en ellos (2). Sin embargo, existen casos en los cuales no se respeta o no se pudo identificar una estructura clara estándar, por ello las tareas de preprocesamiento suponen un desafío adicional.

- **Tamaño:** suelen ser más largos que, por ejemplo, artículos de revistas o textos informativos.

- **Idiomas:** Asociado al ítem anterior, el idioma resulta también como una característica particular del conjunto de datos utilizado para el proyecto. Al tratarse con textos, el concepto de Procesamiento Natural del Lenguaje es algo que no puede evitarse. En la

actualidad existen muchos frameworks y herramientas que están enfocadas en el lenguaje más hablado del mundo, el inglés. Debido a esto, la adaptabilidad al idioma español es algo que se tuvo en cuenta a la hora del desarrollo de las técnicas.

- **Abreviaciones y formalidades:** Debido al ámbito en que se escriben y utilizan los documentos legales, existen ciertos criterios a la hora de escribirlos. Las abreviaciones o formalidades que se utilizan pueden suponer un problema a la hora de generar los sumarios, es por ello que durante las tareas de preprocesamiento fue necesario identificar la mayor cantidad posible y transformarlas para que los resúmenes sean generados de la manera más óptima posible.

- **Subjetividad:** Anteriormente fue mencionado que los sumarios generados a partir de los fallos judiciales no son construidos trivialmente. De hecho, son construidos por un especialista, lo cual implica que dicha persona, a la hora de documentar esta información, adopta un criterio que es subjetivo. Esta subjetividad es una característica que impacta altamente a toda pieza de software que intente realizar la misma tarea, ya que en la mayoría de los casos es muy complejo replicarla.

5.3.2. Segmentación del conjunto de datos

A partir del análisis realizado en la *sección 5.2*, se presenta a continuación un estudio más profundo de las métricas obtenidas sobre el conjunto de datos disponible.

Para comenzar, es importante recordar que las tres variantes de ROUGE (1, 2 y L) son complementarias a la hora de determinar la calidad extractiva de un sumario. A su vez, en dicho análisis es de gran relevancia, dentro de ROUGE, la medida de recall.

Es bajo esta afirmación que se toma la decisión de segmentar el conjunto de datos en subconjuntos según sus puntuaciones de recall para cada una de las variantes.

Lote	**ROUGE 1, ROUGE 2, ROUGE L**		
	Recall >= 0,7	**Recall >= 0,9**	**No aplica**
1	6187	1952	3813
2	7610	2994	2390
3	8069	2756	1931
4	8159	2563	1841
5	8048	2505	1952
6	693	206	99
TOTAL	**38766**	**12976**	**12026**

Tabla 3. Distribución de puntuaciones de recall.

Los valores presentados en la *Tabla 3* representan la **cantidad de documentos**, de cada lote, que cumplen la condición de recall establecida. Mas específicamente, las condiciones propuestas para segmentar el conjunto de datos pueden interpretarse de la siguiente manera:

Recall >= 0.7: Aquellos documentos que poseen un recall mayor o igual a 0,7 para las tres variantes de ROUGE.

Recall >= 0.9: Aquellos documentos que poseen un recall mayor o igual a 0,9 para las tres variantes de ROUGE.

No aplicables: Aquellos documentos que poseen un recall menor a 0,7 para al menos una de las tres variantes de ROUGE.

El subconjunto de la segunda columna se ve contenido por el primero, por ello, si se consideran tres subconjuntos en donde ninguno contenga a otro, es posible observar las segmentaciones sobre el conjunto de datos original y los valores totales quedarían conformados de la siguiente manera:

Recall >= 0.7	Recall >= 0.9	No aplica
25790	12976	12026
50792		

Tabla 4. Cantidad de documentos según puntuaciones de recall.

Este análisis indica que existen 12026 sumarios, un **23,68%** del conjunto de datos original, que fueron construidos a partir de pocos o ningún tipo de extracto textual del fallo judicial. Estos sumarios son muy difíciles de imitar a través del uso de técnicas extractivas.

Por otro lado, los sumarios que arrojaron un **recall mayor o igual a 0,7 pero menor a 0,9** se consideran mayormente extractivos. Esto quiere decir que poseen gran cantidad de fragmentos extraídos del texto original, pero a la hora de su construcción se utilizaron palabras u oraciones adicionales. Este subconjunto representa un **50,77%** del conjunto de datos original.

Por último, se establece que los sumarios con un **recall mayor o igual a 0,9** son altamente extractivos, es decir, son construidos casi o en su totalidad a partir de extracciones de texto del fallo judicial y pueden o no tener algunas pocas palabras o frases adicionales. Este subconjunto representa un **25,55%** del conjunto de datos original.

En base a estos resultados se decidió segmentar el conjunto de datos original para trabajar con los casos en los cuales los sumarios fueron generados principalmente de manera extractiva. El subconjunto resultante se define entonces como aquellos documentos que tienen un recall mayor o igual a 0,9 para las 3 variantes de ROUGE, conformado por un **25,55%** del total de documentos, y es el que se utilizará para la evaluación de las técnicas.

Esta segmentación se realiza a partir de la premisa de que **lo más valioso del dataset será aquello que las técnicas puedan realizar**. De esta forma es posible evaluar las técnicas en base a lo que ellas pueden generar. No se alinea a los fines del proyecto evaluar los resultados de técnicas extractivas utilizando sumarios de referencia que fueron generados a partir de técnicas no extractivas.

Desde otra perspectiva, resulta de gran utilidad segmentar el conjunto de datos para agilizar su procesamiento en la etapa de evaluaciones. El procesamiento de un documento es una tarea computacionalmente costosa, y reducir la cantidad a procesar en aproximadamente un 75% ofrece una gran ventaja en términos de recursos y tiempo utilizados.

Finalmente, se presenta la conformación del conjunto de datos por lotes, que será utilizado para las evaluaciones de las técnicas implementadas:

Lote	Cantidad de documentos
1	1952
2	2994
3	2756
4	2563
5	2505
6	206
Total	**12976**

Tabla 5. Conformación del conjunto de datos por lotes.

5.4. Preprocesamiento – "Text Preprocessing"

El objetivo principal de preprocesar los datos de entrada tiene que ver con obtener mejores resultados a la hora de generar resúmenes. El texto debería estar segmentado y formateado de forma que funcione mejor con los métodos de procesamiento.

A su vez, también está relacionado con no desperdiciar recursos, tanto de tiempo de procesamiento como de capacidad de almacenamiento, en evitar procesar palabras o expresiones que no agregan valor a la semántica ni al entendimiento del texto por parte del algoritmo, eliminándolas de antemano.

Con dichos objetivos en mente, se realizaron distintas tareas para lograr obtener un texto más limpio y bien estructurado que permita generar un sumario de mayor calidad. Resulta importante destacar que cada una de las siguientes secciones se corresponde con una de estas tareas, implementadas en un bloque de código altamente reutilizable.

5.4.1. Reemplazos de expresiones

El primer paso a la hora de preprocesar el conjunto de datos y prepararlo para las siguientes etapas consiste en reemplazar todas las abreviaturas por su correspondiente palabra, de manera de evitar separar oraciones (con el delimitador de un punto) incorrectamente.

Esto es logrado mediante un script que detecta las expresiones y hace su correspondiente reemplazo, como se puede ver ilustrado en la *Tabla 7*. En ella se presentan algunas de estas abreviaciones y formalidades que fueron detectadas.

Expresión	Reemplazo
1er.	Primer
1era.	Primera
Art. / Arts.	Artículo / Artículos
As.	Aires
Avda.	Avenida
Bs.	Buenos
C. N.	Código Nacional
C. N. Civ	Código Nacional Civil
C.P.	Código Penal
C.P.C.	Código Procesal Constitucional
C.P.P.	Código Procesal Penal
cctes.	Consecuentes
Cit.	Citada
Civ.	Civil
Cód./Cod	Código
Const.	Constitución
D. L.	Decreto Ley
Direc.	Dirección
Dr.	Doctor
Dra.	Doctora
Dres.	Doctores
E.D.	El Derecho
Ed.	Edición
Excma.	Excelentísima
Excmo.	Excelentísimo
Expte.	Expediente
f. / fs.	Foja / Fojas
Inc.	Inciso
L.L.	La Ley
L.O.	Ley Orgánica
Nac.	Nacional
Nro.	Número
Ob.	Obra
Pág.	Página
Prov.	Provincial
R.J.N	Reglamento para la Justicia Nacional
sgtes.	Siguientes
Sr.	Señor
Sra.	Señora
Sres.	Señores
ss.	Siguientes
vta.	Vuelta
cfr.	Conforme

Tabla 7. Significado de abreviaciones y formalidades.

Además de esto, se identificaron en el dataset números con puntos en su expresión, como por ejemplo 16.233. Esto también fue tenido en cuenta y se eliminan dichos puntos, resultando en, siguiendo con el ejemplo, 16233.

Seguidamente, se eliminan otros casos particulares que no aportan semánticamente al texto: Entre ellos encontramos caracteres especiales (#%&*\+/<=>@[\]^{|}~_), tabuladores y/o espacios extras (*palabras separadas por más de un espacio u oraciones tabuladas en el comienzo*), y por último oraciones que no contengan ninguna palabra (". .").

El recurso principal utilizado para realizar estas tareas fue una librería de Python llamada "re", que permite definir, encontrar y reemplazar expresiones regulares dado un texto particular.

A continuación, se muestran algunos ejemplos del uso de esta librería:

```
DOT_BETWEEN_NUMBERS_re = re.compile(r"\b[0-9]{1,2}(?:\.[0-9]{3})+\b")
BAD_PUNCT_re = re.compile(r'([%s])' % re.escape('"#%&\*\+/<=>@[\]^{|}~_'),
re.UNICODE)

# Elimina puntos entre números
refined_text = re.sub(DOT_BETWEEN_NUMBERS_re, lambda x:
x.group().replace(".", ""), refined_text)

# Elimina caracteres especiales
refined_text = BAD_PUNCT_re.sub('', refined_text)
```

5.4.2. Tareas de tokenización

Esta etapa consiste en dividir el texto en porciones más cortas, a partir de uno o varios delimitadores. En este caso, comenzamos dividiendo el texto a partir de delimitadores como '\r\n' o '\r\r\n' que se encuentran presente en la totalidad de los documentos y representan un salto de línea en el documento físico. Esto permitirá, en etapas consecuentes, realizar una segmentación adicional en oraciones. Estos delimitadores podrían considerarse como un separador de párrafos y fueron de gran utilidad ya que reducen mucho los riesgos de inducir un error a la hora de segmentar en oraciones. Dependiendo de la técnica usada y, como es posible que queden párrafos irrelevantes con longitudes muy pequeñas, se pueden eliminar aquellos que tengan 2 o menos palabras. Ejemplo: Oraciones que comienzan en "VISTOS:" o "CONSIDERANDO:".

Para la tokenización en oraciones se utiliza una solución provista por una librería de PLN, que ha sido entrenada y optimizada para realizar esta tarea de la manera más efectiva posible.

5.4.3. Eliminación de "stop-words"

Para efectuar este proceso también se hace uso de una librería de PLN, que identifica a partir de una base de conocimiento en español aquellas palabras consideradas vacías de contenido.

En el siguiente ejemplo extraído de nuestro conjunto de datos se puede apreciar el concepto mencionado:

Oración original:

"Apela la actora conforme los agravios de fojas 798805 contestados a fojas 821826, y por la accionada según los que expresa a fojas 816817."

Oración sin stopwords:

"Apela actora conforme agravios fojas 798805 contestados fojas 821826, accionada según expresa fojas 816817."

Esta tarea implica un desafío a la hora de construir un resumen, ya que las oraciones sin este tipo de palabras carecen de sentido para un lector humano, haciendo que el mismo sea difícil de comprender. Es por ello que se adaptó una solución donde se almacenan paralelamente las oraciones originales y las oraciones sin las stopwords, utilizándose las primeras a la hora de construir el resumen y las segundas como entradas para la generación del mismo.

5.4.4. Estructura de datos y persistencia

En las etapas posteriores al preprocesamiento, resulta fundamental mantener consistencia a la hora de acceder a los datos y utilizarlos. Para ello, construimos una estructura genérica para todo el conjunto de datos que al completarse las tareas previamente mencionadas queda constituido de la siguiente manera:

Cabe aclarar que la siguiente estructura representa un solo documento. Cada uno de ellos son a su vez almacenados en un DataFrame que es el utilizado a lo largo de todo el proceso.

```
"text_id":
    [
        "ArrayParrafosConOraciones-SinStopWords": [
            "Párrafo1": [Oración1, Oración2],
            "Párrafo2": [Oración3, Oración4, Oración5],
            "PárrafoN": [OraciónN-1, OraciónN]
        ],
        "OracionesSinStopWords": [Oración1, Oración2, OraciónN],
        "OracionesConStopWords": [Oración1, Oración2, OraciónN]
    ]
```

Esta estructura no solo es de gran utilidad para realizar todas las tareas que una técnica generación de resúmenes supone, sino que también es esencial para reconstruir un sumario una vez identificadas las oraciones a incluir en el mismo.

5.5. Técnicas revisadas

A lo largo del PFC se revisaron distintas estrategias para generar los resúmenes previamente mencionados. En este apartado se detallan los conceptos teóricos más relevantes relacionados a las técnicas seleccionadas y se profundiza acerca de cada respectiva implementación.

5.5.1. TextRank

En muchas aplicaciones relacionadas con el Procesamiento del Lenguaje Natural los grafos suelen representar de manera muy adecuada un texto. De hecho, desde el momento en el que un texto es fragmentado en palabras y se establece algún tipo de relación entre ellas, se dispone de una representación en forma de grafo. (24)

TextRank consiste en una metodología basada en grafos y hace uso principalmente de un conocido algoritmo: PageRank.

La finalidad de PageRank es medir la importancia de cualquier página web en Internet en función de los enlaces que dicha página recibe, aunque también se ha utilizado en ideas similares en otros contextos como el análisis de redes sociales o de redes de referencias bibliográficas. (24)

Los algoritmos de clasificación basados en grafos son esencialmente una manera de decidir la importancia de un vértice, basándose en información recursivamente extraída del grafo.

Formalmente, el algoritmo de PageRank se conceptualiza como:

Sea ***G = (V,E)*** un grafo dirigido donde ***V*** es un conjunto de vértices o nodos y ***E*** un conjunto de arcos, siendo ***E*** es un subconjunto de ***V x V***. Para un vértice dado ***Vi***, se define a ***E(Vi)*** como el conjunto de vértices que apuntan hacia ***Vi***, y, por otro lado, ***S(Vi)*** como el conjunto de vértices hacia donde apunta ***Vi***. A partir de estas dos operaciones básicas, se define la puntuación (o PageRank) de un determinado vértice con la siguiente fórmula:

$$P(V_i) = (1-d) + d \sum_{j \in E(V_i)} \frac{1}{|S(V_j)|} P(V_j)$$

Fórmula 7.

donde ***d*** es un factor de amortiguación que tiene como objetivo incluir en el modelo la probabilidad de que haya un salto aleatorio de un vértice del grafo a cualquier otro. (24)

En el contexto de la navegación en Internet, dicho factor representa la probabilidad de que un usuario acceda a una página a través de un enlace situado en la página actual, siendo por tanto ***(1 − d)*** la probabilidad de que dicho usuario salte a una página aleatoria no enlazada con la página actual. (25)

Partiendo de valores arbitrarios para las puntuaciones asignadas a cada nodo del grafo, se alcanza un **punto de convergencia** aplicando iterativamente la fórmula hasta que la mayor diferencia de las puntuaciones obtenidas para cada nodo, entre dos iteraciones, es menor que un determinado umbral. Una vez finalizado el algoritmo, la puntuación alcanzada por cada nodo representa la importancia de este dentro del grafo. Cabe destacar que los valores asignados inicialmente no condicionan el resultado, sino que con la cantidad de iteraciones la convergencia puede variar.

Sin embargo, al hablar de textos, pueden existir múltiples vínculos entre nodos, por lo que introducir el peso de una conexión entre vértices resulta útil.

En este caso, la puntuación de cada nodo se calcularía con la siguiente fórmula:

$$P(V_i) = (1-d) + d \sum_{j \in E(V_i)} \frac{p_{ji}}{\sum_{k \in S(V_j)} P_{jk}} P(V_j)$$

Fórmula 8.

donde $\boldsymbol{p_{ji}}$ es el peso del arco que va del vértice ***Vj*** al ***Vi***. (24)

A partir de la generalización del algoritmo se puede intuir el funcionamiento de **TextRank**. Al obtener un grafo a partir de un texto y calcular, a partir del mismo, la puntuación de cada nodo, se logran identificar las partes más importantes del texto que luego serán consideradas un resumen del mismo.

Cabe aclarar que para distintos casos la forma de construcción del grafo es distinta, lo que deriva en resultados distintos que se adecuan a una problemática y/o toma de decisiones distinta. Por ejemplo, pueden existir grafos donde los nodos sean simplemente palabras o algunos en donde los mismos representen oraciones.

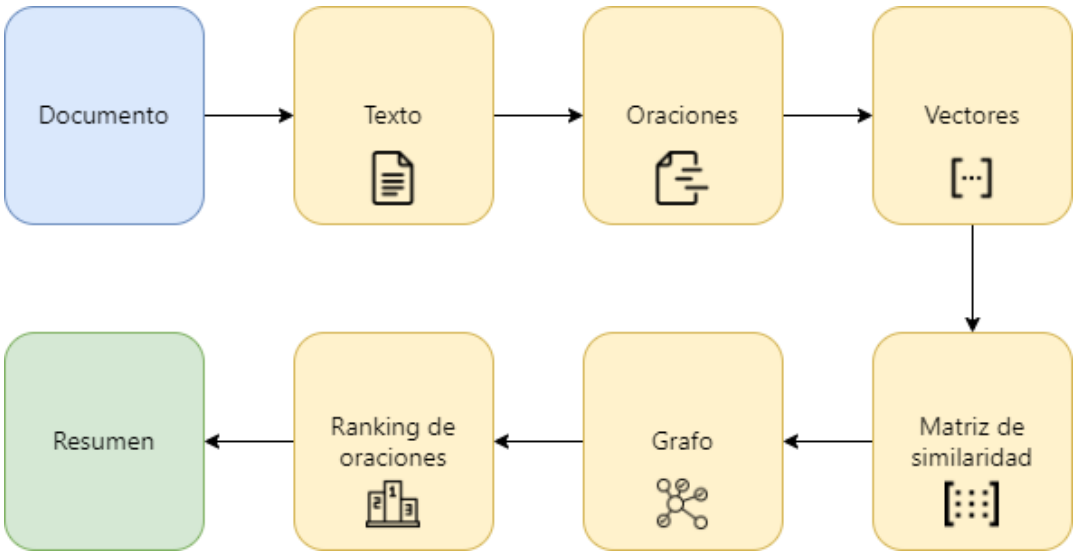

Figura 16. Secuencia de procesamiento de TextRank.

En conclusión, TextRank es una técnica de resumen de texto extractiva y no supervisada (ya que no requiere de entrenamiento previo) cuyo requisito primordial es representar un texto como grafo.

Cuando este modelo es extrapolado al caso de documentos legales, las oraciones representan los nodos del mismo y las aristas representan una medida de similitud léxica entre las ellas.

Este modelo posee ciertas características que lo convierten en una gran elección para el proyecto. En primer lugar, es un modelo independiente del lenguaje y las estructuras que un texto, lo cual en el caso de documentos legales es considerado relevante ya que permite abstraerse de cierta manera de dichas complejidades.

Por otro lado, es un algoritmo de alta popularidad, lo cual implica que tiene un aval científico y alta confiabilidad a la hora de ser utilizado. También, a la hora de su implementación se hallan gran cantidad de recursos que incluso están certificados por compañías como Google.

Otra ventaja del modelo tiene que ver con que el costo computacional de la ejecución de este es bastante bajo, lo cual puede considerarse un aspecto importante a la hora de pensar en una herramienta formal.

5.5.1.1. Técnica 1 – "TextRank Summarizer"

La primera estrategia desarrollada en el proyecto consistió entonces en la utilización del algoritmo TextRank, el cual fue explicado con detalles en la *sección 5.5.1*.

Para lograr una implementación exitosa se planteó una arquitectura, que permitió no solo concluir con un código más prolijo y legible, sino que también facilitó la introducción de cambios haciéndola más eficiente y la reutilización de bloques de códigos a la hora de realizar una segunda implementación. En la *Figura 17* se presenta un esquema general de la misma:

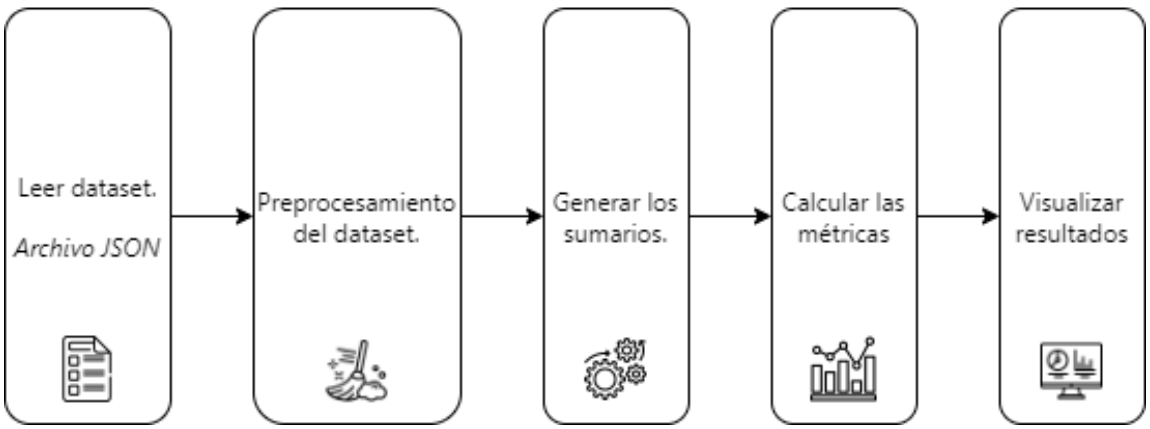

Figura 17. Arquitectura de TextRank Summarizer

Cada uno de los bloques de la *Figura 17* representan un módulo escrito en Python que puede contar con una o varias funciones. A continuación, se presenta una breve descripción de cada uno de ellos:

- El primer bloque del proceso es el encargado de leer el conjunto de datos completo a transformar y resumir. El mismo debe tener el formato mencionado en la *sección 5.3* y su tamaño es indistinto, puede contener de 1 a N documentos.
- El segundo bloque contiene todas las funciones que permiten realizar las tareas de preprocesamiento del input.
- El tercer módulo es el encargado de realizar los sumarios. Este es tal vez el más relevante de mencionar en esta sección ya que aquí es donde efectivamente se pone en funcionamiento el algoritmo estudiado.
- Una vez generados los resúmenes de texto, el cuarto módulo contiene la lógica encargada de calcular las métricas correspondientes para así evaluar la eficiencia del algoritmo. Esto se presenta con más detalles en secciones posteriores.
- El último módulo obtiene los resultados del anteriormente mencionado y los presenta en forma de gráficos para tener una visión más global y práctica de los mismos.

A continuación, se presentan dos recursos utilizados que fueron claves para la implementación exitosa de esta técnica. Ambos se utilizan en el tercer módulo.

5.5.1.1.1. spaCy

SpaCy es una biblioteca de procesamiento de lenguaje natural para Python diseñada específicamente con el objetivo de ser una biblioteca útil para implementar sistemas listos para producción. Los modelos de spaCy de procesamiento de lenguaje natural permiten analizar un texto y extraer información tanto del texto como de las predicciones del modelo sobre su significado por el contexto. (26)

Una de las características más importantes son las tuberías o más bien conocidas en el inglés como "pipelines". Estas permiten tomar como input un texto y transformarlo con distintas herramientas de PNL. Este concepto es de gran relevancia para la implementación de TextRank.

Además, cabe destacar que esta librería admite paquetes de diferentes idiomas, previamente entrenados y diseñados para hacer uso de las herramientas de PNL ofrecidas.

En la *Figura 18* se puede observar el esquema de un pipeline con sus respectivas herramientas.

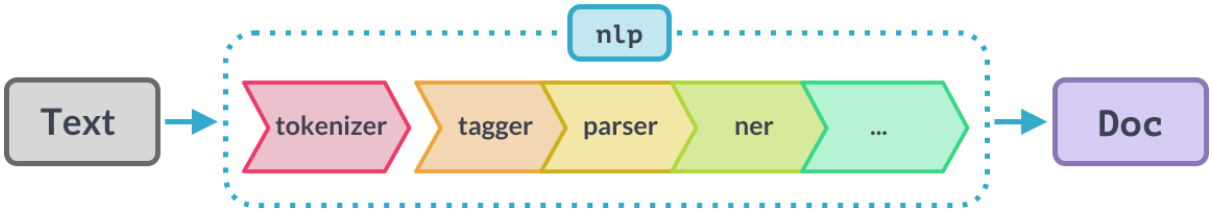

Figura 18. Pipeline de spaCy

Algunas de las más útiles que ofrece spaCy son:

- Tokenizer
- Tagger
- Parser
- Lemmatizer
- Sentencizer
- Textcat

5.5.1.1.2. PyTextRank

PyTextRank es una implementación del algoritmo TextRank como una extensión del pipeline de spaCy. (27)
En términos de desarrollo esto resulta una gran ventaja ya que podemos hacer uso de todas las herramientas de PLN que la librería provee para luego ejecutar el algoritmo de forma eficiente.

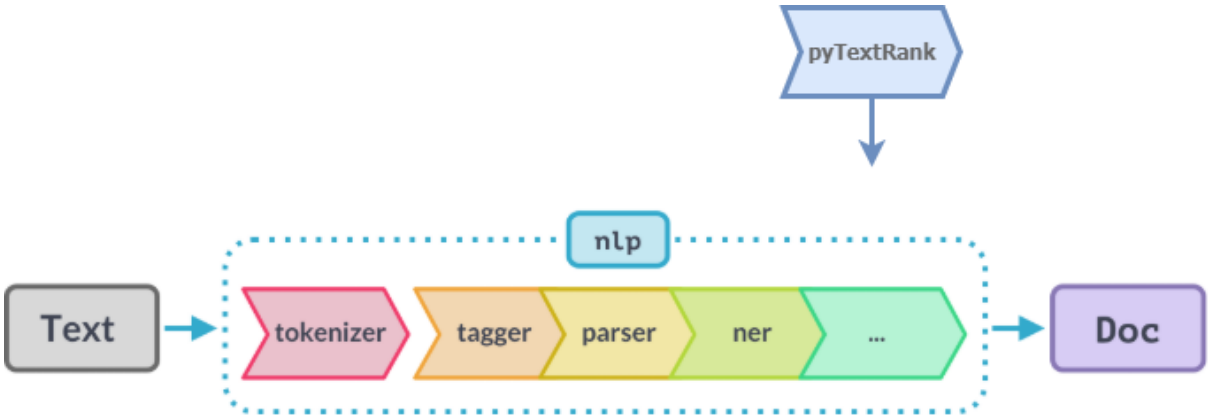

Figura 19. Pipeline de spaCy con el agregado de pyTextRank.

Este algoritmo se ejecuta de manera independiente al idioma que se esté procesando, es por ello que las tareas de preprocesamiento son la pieza clave para que el sumario generado por TextRank sea certero.

5.5.1.1.3. Configuración de TextRank

Como fue mencionado previamente, un módulo encargado específicamente de generar los sumarios haciendo uso del algoritmo TextRank fue desarrollado.

A continuación, se definen los puntos más importantes del mismo:

- Parámetros de entrada:

Este módulo, para mantener simpleza, recibe únicamente como entrada al conjunto de datos que debe ser resumido. Esto se alinea con el concepto de técnica de resumen para documento único.

- Configuración de spaCy:

Para las tareas de tokenización, se utilizó la herramienta sentencizer provista por la librería. Esta es capaz de tokenizar el texto en oraciones de manera eficiente, luego de pasar las etapas de preprocesamiento. Para ello, se seleccionó un modelo de idioma español entrenado previamente, en donde se hace foco en la precisión del mismo para las tareas a realizar y no tanto en la eficiencia computacional. El modelo se nombra como "es_dep_news_trf".

A su vez, el modelo requiere configurar sus parámetros a la hora de generar los resúmenes. Para ello simplemente debe definirse la cantidad de oraciones a incluir en el resumen. De esta forma, el algoritmo solo seleccionará las dos oraciones (o vértices) del grafo con mayor puntuación. El parámetro se nombra como “limit_sentences”.

5.5.2. Resúmenes extractivos basados en características

En secciones anteriores se mencionan distintos trabajos e investigaciones sobre cómo generar un resumen de texto automático. Entre todos ellos, hay un aspecto en común y consiste en que los resúmenes se construyen en torno a una única característica de los textos, es decir, se identifica un patrón o particularidad entre el texto original y un resumen ideal, y se lo utiliza en etapas de procesamiento posteriores. Por ejemplo, en el método presentado que utiliza TextRank, se evalúa la similitud léxica entre oraciones y a partir de esta se generan los sumarios, es decir, el problema se resuelve trivialmente a través de esta característica. Otro caso similar, es el que se realiza en (28), en donde se establece a través de un análisis que el 85% de las oraciones iniciales de un párrafo describen el tópico del mismo.

También, existen otras ocasiones en las que las técnicas están más enfocadas a la frecuencia de palabras o el posicionamiento de las mismas, modelos estadísticos de alta complejidad, o bien orientadas a Machine Learning.

El objetivo de generar un resumen basado en características (o en inglés "feature-based") es recopilar las más importantes de un texto, generar puntuaciones para cada una de ellas y luego generar el sumario de manera extractiva. (29)

5.5.2.1. Técnica 2 – "Feature Based Summarizer"

Al igual que en la Técnica 1, se presenta la arquitectura propuesta para la implementación de la misma. Nuevamente, se hace foco en la construcción de bloques de código que resulten reutilizables para futuras implementaciones. En la *Figura 20* se puede apreciar que varios de los bloques son los mismos que para TextRank:

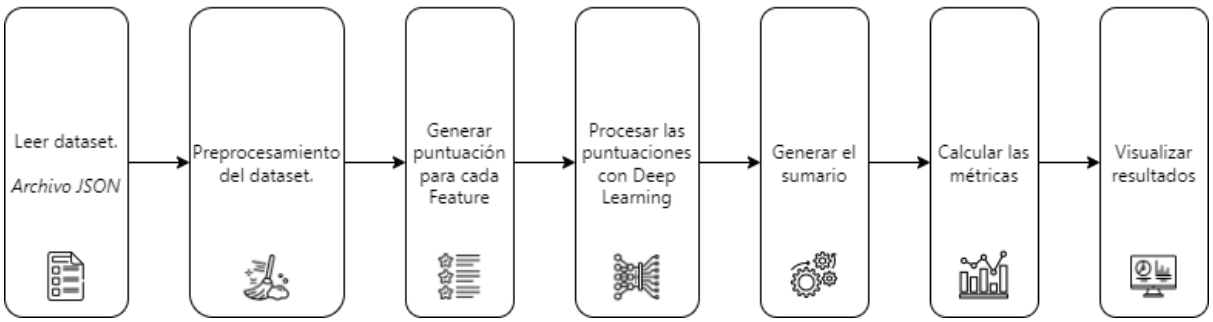

Figura 20. Arquitectura de Feature Based Summarizer.

Presentado la arquitectura general de la técnica, se listan a continuación los detalles más importantes para el desarrollo de la misma.

5.5.2.1.1. Características

Como fue mencionado previamente, la estrategia en esta metodología consiste evaluar distintas cuestiones asociadas a un texto para generar un resumen de manera más certera. A

lo largo de esta sección se presentan las características ("features") que fueron consideradas para su implementación:

- **Número de palabras temáticas**

Se refiere a las 10 palabras que se encuentran en el texto con mayor frecuencia. Para cada oración se calcula, según la *Fórmula 9*, la proporción de palabras temáticas respecto al total de palabras en el documento.

$$palabrasTemáticas \ = \frac{cantidad\ de\ palabras\ temáticas}{cantidad\ total\ de\ palabras}$$

Fórmula 9.

- **Posición de oración**

Esta característica indica la posición de la oración en el texto. Esto resulta relevante ya que las primeras y últimas oraciones del documento suelen introducir o concluir ideas generales del texto. Esto se refleja en el cálculo de la puntuación, dado por la siguiente fórmula:

$$posiciónOración = \begin{cases} 1, & si\ es\ la\ primer\ o\ última\ oración\ del\ texto \\ \cos\left((posición - min) * \left(\left(\frac{1}{max}\right) - \ min\right)\right), & resto\ de\ oraciones \end{cases}$$

Fórmula 10.

Donde:

$posición \ = posición\ de\ la\ oración\ en\ el\ texto$

$min \ = th * N$

$max \ = th * 2 * N$

$th = 0.2 * N$

$N \ = cantidad\ total\ de\ oraciones$

- **Longitud de oración**

Como en el texto pueden encontrarse oraciones que son demasiado cortas como para relevar información importante, esta característica se asegura de no tenerlas en cuenta. Se calcula:

$$longitudOración = \begin{cases} 0, & si\ la\ cantidad\ de\ palabras\ es\ menor\ a\ 3 \\ \text{cantidad de palabras en la oración}, & en\ otro\ caso \end{cases}$$

Fórmula 11.

- **Posición de oración relativa al párrafo**

La relevancia de esta característica surge al observarse que, al comienzo de cada párrafo, una nueva discusión se abre, y al final del mismo, se llega a una conclusión. Esta relevancia se expresa matemáticamente según la fórmula a continuación:

$$posicionEnPárrafo = \begin{cases} 1, & si\ es\ la\ primer\ oración\ del\ párrafo \\ 0, & en\ otro\ caso \end{cases}$$

Fórmula 12.

- **Cantidad de sustantivos propios**

El objetivo de esta característica es darle importancia a aquellas oraciones que contengan una cantidad sustancial de sustantivos propios. Para ello, se cuenta para cada oración la cantidad de este tipo de sustantivos que aparecen.

Para la identificación de sustantivos propios, en términos de desarrollo, se utilizó un paquete llamado "nltk.tag" (nltk.org/api/nltk.tag.html) provisto por la librearía NLTK (Natural Language ToolKit) el cual consiste en analizar palabras, haciendo uso de un modelo pre entrenado, y asignarle a cada una de ellas una etiqueta (o "tag") que denota cierta propiedad de la misma.

La puntuación final para cada oración se calcula de la siguiente manera, de forma que los resultados sean normalizados:

$$sustantivosPropios = \frac{cantidad\ de\ sustantivos\ propios}{cantidad\ de\ palabras}$$

Fórmula 13.

- **Cantidad de entidades nombradas**

Esta característica puede considerarse similar a la anterior, pero con un enfoque un tanto más complejo.

Como se mencionó en la *sección 2.4.3* las entidades nombradas son consideradas un conjunto de elementos importantes para la comprensión de un texto. Oraciones que contengan entidades como empresas, nombres, referencias a grupos de personas, etc. son de gran importancia para el resumen de un texto.

Para encontrar las entidades presentes en cada oración, en este caso se utilizaron nuevamente dos paquetes, ambos provisto por la librería NLTK:

- El primero "nltk.tag" (nltk.org/api/nltk.tag.html), utilizado previamente para la identificación de sustantivos propios, y que permite etiquetar palabras según ciertas propiedades de las mismas.
- El segundo "nltk.chunk" (nltk.org/api/nltk.chunk.html), es un paquete orientado a resolver tareas de "Named Entity Recognition", es decir, permite identificar en un texto

fragmentos del mismo que representen alguna entidad. En la *Figura 21* se presenta una ilustración de cómo NLTK estructura un texto e identifica las entidades nombradas:

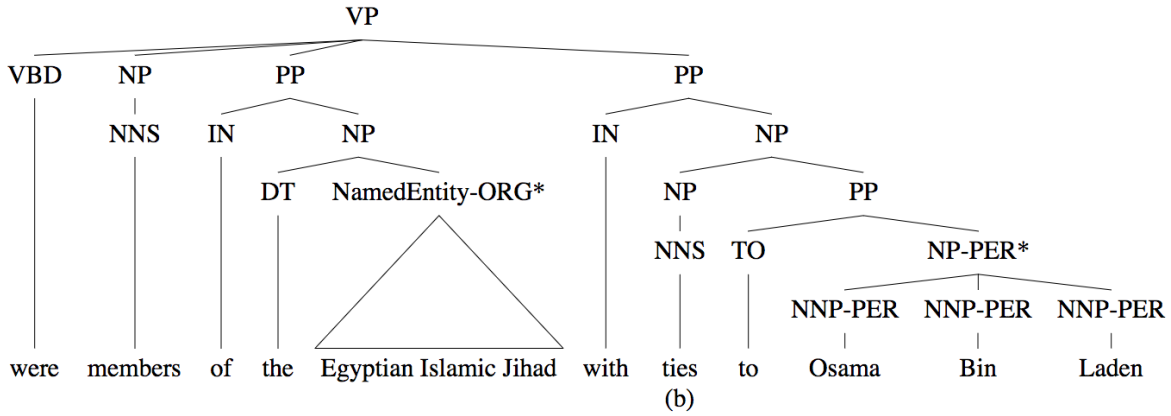

Figura 21. Análisis estructural de entidades nombradas. (30)

Una vez identificadas las entidades para cada oración, la puntuación final se calcula para cada una de ellas de la siguiente manera:

$$entidadesNombradas = \frac{cantidadEntidadesNombradas}{cantidad\ de\ palabras}$$

Fórmula 14.

- **Cantidad de numerales**

Dado que los números son cruciales para presentar hechos, brindar datos, descripciones o resultados, esta característica le otorga importancia a aquellas oraciones que los presenten. Para cada oración, se calcula la proporción de numerales respecto al total de palabras en ella:

$$numeralesEnOración\ = \frac{cantidad\ de\ números}{cantidad\ de\ palabras}$$

Fórmula 15.

- **Frecuencia de Término – Frecuencia Inversa de Oración (TF-ISF)**

Esta característica se deriva de una medida muy utilizada conocida como TF-IDF (Frecuencia de término – frecuencia inversa de documento), es decir, la frecuencia de ocurrencia de un término en una colección de documentos. Indica cuán relevante es una palabra para un documento en una colección de estos.

Para nuestro caso puntual, la misma fue modificada de manera tal que se calcule la frecuencia de ocurrencia de una palabra en una colección de oraciones. De esta forma podemos saber cuán relevante es una palabra para un texto.

Para su cálculo debemos multiplicar la frecuencia de una palabra en una oración, por la cantidad de ocurrencias de la misma a lo largo de todo el texto. Esto se realiza para todas las palabras de una oración obteniendo una puntuación por cada una de ellas:

$$TF - ISF = \frac{\log(\sum_{all\ words} TF * ISF)}{Total\ words}$$

Fórmula 16.

- **Similitud de oración al centroide**

Las oraciones con la puntuación de TF-ISF más alta son consideradas como la oración centroide. Luego, se calcula la similitud coseno del resto de las oraciones respecto a la antes mencionada. Esta última es una medida que, a grandes rasgos, permite encontrar la similitud entre oraciones. La misma es muy utilizada al trabajar con comparaciones de textos. (31)

5.5.2.1.2. Refinamiento con Deep Learning

El proceso más simple para generar un resumen automático consistiría en obtener la matriz de puntuaciones, a partir de la cual se calcularía una puntuación única para cada oración y se generaría el resumen con las oraciones que posean una puntuación mayor a cierto umbral.

En la técnica propuesta, a la obtención de características se le adiciona una capa más de complejidad que consiste en refinar las puntuaciones obtenidas para cada una de ellas utilizando un modelo de Deep Learning. (29)

El objetivo de dicho refinamiento consiste en utilizar un modelo de aprendizaje profundo que sea capaz de identificar patrones en la matriz de puntuaciones dada y predecir una nueva matriz de puntuaciones, en donde algunas de las características tengan más relevancia que otras debido a dichos patrones que la IA identifica. En otras palabras, las puntuaciones reciben un "peso" adicional dependiendo de su relevancia para los resúmenes.

De esta manera, es posible es mejorar las puntuaciones que se obtienen al evaluar el texto en primera instancia. Este incremento de complejidad deriva en resúmenes de mayor calidad.

El modelo de Deep Learning propuesto consiste en una Restricted Boltzmann Machine con nueve nodos visibles y nueve nodos ocultos, coincidiendo de esta forma con las features previamente mencionadas.

Para la implementación se utilizó un framework llamado **Pytorch**, el cual provee todas las herramientas necesarias para construir una RBM de manera sencilla y configurar todos sus parámetros.

Como fue mencionado en secciones anteriores, las RBM son un tipo de red no supervisada, lo cual implica que no requieren entrenamiento previo, pero es posible evaluar las mismas haciendo uso de un conjunto de datos destinado a entrenamiento. En el caso de este proyecto,

el conjunto de datos no contiene matrices ideales de puntuaciones contra las cuales puedan compararse los outputs generados por la red. Es por ello que, para demostrar y fundamentar el beneficio que implica el uso de la RBM, se presenta una comparación de resultados obtenidos al generar resúmenes con y sin la presencia de la misma.

El conjunto de datos utilizados para este análisis es un subconjunto de datos reducido, construido aleatoriamente, con una cantidad de **2000 documentos**, con el fin de demostrar la mejora que sugiere el uso de una RBM. En cuanto a la técnica de resumen como tal, los parámetros utilizados fueron exactamente los mismos para ambos procesamientos.

	RBM OFF	RBM ON
[ROUGE-1] Recall	0.46257994403171954	0.5134177049503073
[ROUGE-1] Precisión	0.16112872110691223	0.17547268504885155
[ROUGE-1] F-Score	0.2253001744867255	0.2432082073814726
[ROUGE-2] Recall	0.27540365818874857	0.3309232172264241
[ROUGE-2] Precisión	0.09062028012031176	0.10536028090851826
[ROUGE-2] F-Score	0.12546038260594064	0.14751992382132853
[ROUGE-L] Recall	0.40887896768264825	0.4602367378948254
[ROUGE-L] Precisión	0.14264687405166623	0.15801291212758067
[ROUGE-L] F-Score	0.1989591103573935	0.21866344607957974

Tabla 8. Resultados de procesamiento ante la presencia y ausencia de una RBM.

Los resultados expuestos en la *Tabla 8*, son promedios que fueron calculados a partir de los resultados obtenidos para cada documento. Es posible observar que la utilización de la RBM deriva en valores de métricas más altos, lo cual se considera suficiente para justificar el uso de la misma como parte de la técnica.

5.5.2.1.3. Generación del resumen

En esta técnica, al igual que en TextRank, una vez que ha finalizado el procesamiento del input y la generación de las puntuaciones, la generación del resumen podría parecer un proceso trivial, en donde se calcula una puntuación global para cada oración y se genera el resumen haciendo uso de las mejor puntuadas, definiendo un límite de cantidad de oraciones previo.

Pero aquí, el proceso de generación cuenta también con un agregado de complejidad: En primera instancia, se calcula a partir de la matriz "mejorada", una puntuación global para cada una de las oraciones y las mismas son ordenadas descendientemente. Sólo la primera será seleccionada para formar parte del resumen.

Luego, para seleccionar las siguientes oraciones que formarán parte del resumen se calcula la similaridad de Jaccard entre las oraciones restantes con la primera seleccionada, pero con la limitación de que esta medida se calcula solo para la mitad superior del resto de oraciones. A continuación, se selecciona la oración con la similitud más alta. Esta selección se realiza iterativamente hasta llegar a un límite de oraciones previamente establecido.

Finalmente se genera el resumen ordenando las oraciones según su posición original en el texto. Este proceso tiene como objetivo producir un resumen aún más coherente en lugar de un conjunto de oraciones seleccionadas y ordenadas trivialmente. (29)

En la *Figura 22* se presenta una ilustración del proceso de generación del sumario.

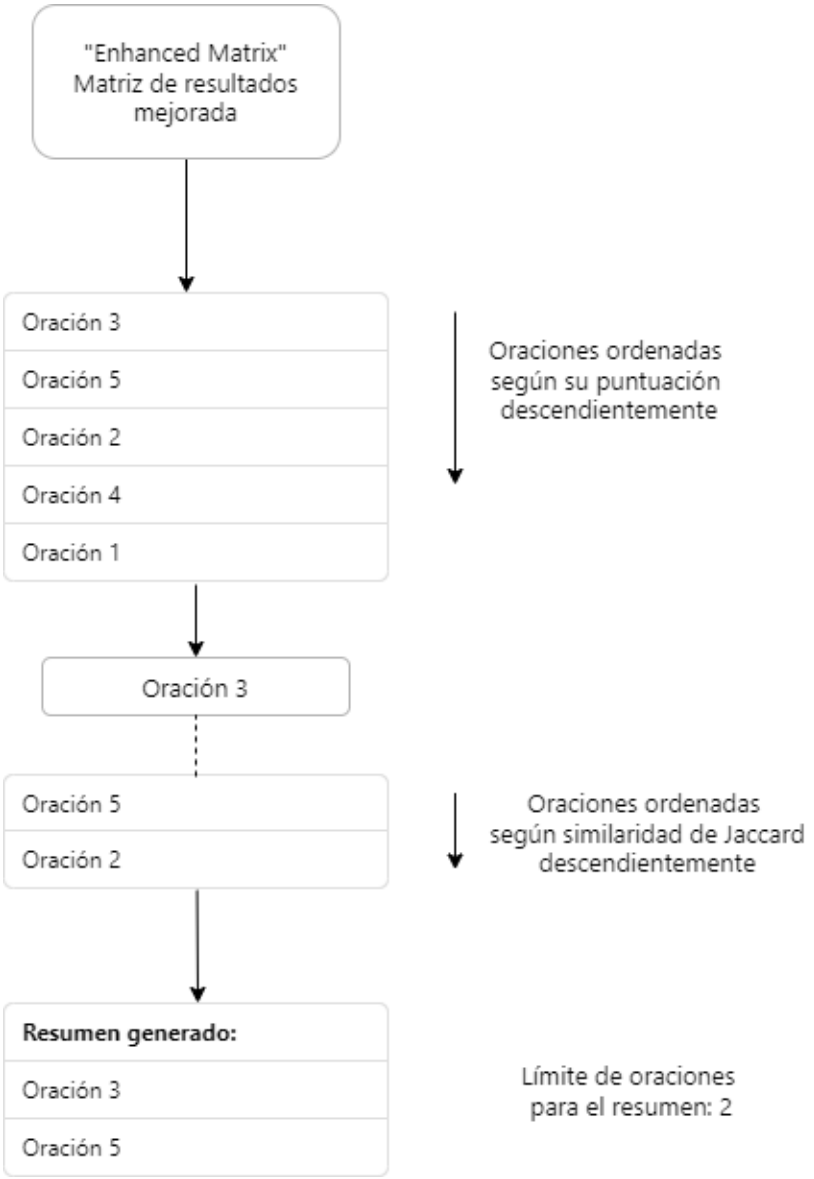

Figura 22. Proceso de generación de sumarios.

5.6. Evaluación de técnicas y resultados

A lo largo de esta sección se presentan todos los detalles asociados a la evaluación de técnicas y métricas utilizadas en el proyecto. Esta etapa resulta fundamental para poder concluir un análisis de factibilidad sobre las técnicas implementadas.

Como fue mencionado en la *sección 2.5* existen dos formas distintivas para evaluar resultados, humana y automática. A lo largo del proyecto se realizaron ambos tipos de evaluaciones, pero haciendo un uso mucho mayor de una de ellas: la evaluación automática.

Dada la naturaleza del proyecto, la evaluación humana es un tanto complicada de aplicar ya que requiere que profesionales del ámbito legal realicen evaluaciones constantes de los resultados, y debido a que los mismos no estuvieron involucrados en el proyecto, esto no fue posible. Pero, por otro lado, existe una evaluación realizada por los mismos desarrolladores, en donde no se utiliza ningún tipo de métrica ni puntuación, sino que se realizan evaluaciones subjetivas a partir de una entrada y salida dadas.

Por su parte, la evaluación automática fue aplicada exhaustivamente a lo largo de todo el proyecto. La métrica utilizada para realizar este tipo de evaluaciones fue ROUGE, en las variantes que fueron introducidas (ROUGE 1, 2 y L) en la *sección 2.5.2.1*. Con la misma es posible determinar automáticamente la calidad de un sumario al compararlo con un sumario "ideal" creado por humanos. (19)

5.6.1. Interpretación de las métricas

Al igual que en la *sección 5.2*, se presenta aquí un análisis acerca de lo que representa ROUGE y sus variantes en esta etapa de evaluaciones.

Con **ROUGE 1** es posible capturar la proporción de palabras individuales coincidentes entre el sumario generado y el sumario ideal. Sin embargo, esta métrica por si sola puede conducir a malinterpretaciones de los resultados ya que muchas palabras pueden coincidir, pero el orden en el texto puede ser distinto, dando interpretaciones semánticas distintas.

Por su parte, **ROUGE 2** determina la proporción de bigramas (secuencias de dos palabras) coincidentes entre sumario generado y el sumario ideal. Un alto valor de esta métrica implica que las palabras en cuestión son, en su mayoría, consecutivas.

Para complementar las anteriores, **ROUGE L** permite evaluar ambos textos de manera distinta, ya que busca la cadena común más larga contenida dentro de ambos sumarios. Es decir, se tiene en cuenta la similitud de estructura a nivel de oración de forma natural y se identifica automáticamente los n-gramas de secuencia más largos que se superpongan.

Independientemente de si se trata de ROUGE 1, 2 o L, al analizar los resultados también se debe tener en cuenta lo que representan **el valor del recall y de la precisión**. Para ello se brinda en la *Tabla 8* que indica distintas interpretaciones:

	Precisión Baja	**Precisión Alta**
Recall Alto	La mayoría de palabras en el resumen ideal fueron capturadas por el resumen generado, pero existen muchas palabras irrelevantes.	El resumen generado es similar al ideal, tanto en palabras coincidentes como en longitud.
Recall Bajo	El resumen generado coincide en muy pocas (o ninguna) palabras con el resumen ideal.	La mayoría de palabras capturadas en el resumen generado son relevantes, pero falta contenido del resumen ideal.

Tabla 9. Interpretaciones del Recall y Precisión.

A su vez, es posible interpretar el recall y la precisión a partir de valores ejemplares:

ROUGE-N – Recall = 0,4 significa que el 40% de los N-gramas en el sumario de referencia están también presentes en el sumario generado.

ROUGE-N - Precisión = 0,4 significa que el 40% de los N-gramas en el sumario generado están también presentes en el sumario de referencia.

Por su parte, **ROUGE-N – F1 Score** es más complejo de interpretar ya que representa una combinación de las dos métricas anteriores. Podría simplificarse su interpretación a que, si el valor es "alto", entonces ambas métricas obtuvieron un valor alto.

5.6.2. Criterio de evaluación

Al tratarse de un análisis de factibilidad es necesario contar con un criterio a la hora de evaluar resultados. Para la construcción del mismo fue necesario determinar algún parámetro que pertenezca a cada una de las técnicas y tenga influencia sobre los resultados. Es por ello que la decisión fue utilizar una característica común a ambas técnicas: la longitud del resumen generado. La misma puede expresarse de distintas maneras, pero acorde a las formas de generar resúmenes de las técnicas implementadas, su valor está representado por la **cantidad de oraciones a extraer**.

Para encontrar el valor óptimo, se realizó un análisis sobre los sumarios ideales provistos en el conjunto de datos, el cual consistió en encontrar aquel sumario con mayor cantidad de oraciones y utilizar esa cantidad como valor inicial del parámetro para ambas técnicas. Este fue de 19 y se definió como **valor base** para comenzar con la búsqueda del parámetro que mejor se adecue al procesamiento de las técnicas.

Dicha búsqueda se llevó a cabo utilizando un subconjunto de datos reducido, de forma que los distintos procesamientos no resulten tan costosos computacionalmente.

Luego de aplicar ambas técnicas utilizando como parámetro el valor base, se realizaron "iteraciones" en donde dicho valor se fue decrementando. El objetivo fue determinar el impacto del incremento o decremento de dicho valor en los resultados finales.

En la *Tabla 10* se presentan los resultados de las iteraciones. Estos se expresan en términos de ROUGE L, ya que dicha variante combina aspectos de ROUGE 1 y 2, y permite realizar un análisis de manera directa sobre los mismos.

Parámetro	**TextRank**		**Feature Based**	
	Recall	**Precisión**	**Recall**	**Precisión**
19	0,649	0,057	0,516	0,165
16	0,607	0,087	0,474	0,163
15	0,612	0,088	0,447	0,171
13	0,569	0,123	**0,386**	**0,179**
10	**0,531**	**0,150**	0,295	0,165
7	0,482	0,162	0,296	0,203

Tabla 10. Resultados de las distintas iteraciones expresados en términos de ROUGE L.

Es posible identificar una relación entre los valores que toma el parámetro y los valores que van adquiriendo la precisión y el recall. A medida que el parámetro disminuye, el recall disminuye, pero la precisión aumenta. Esto se traduce a que, si un sumario generado contiene

menos oraciones, entonces es probable que incluya menos contenido del sumario ideal, por lo tanto, disminuye el recall. A su vez, es probable que contenga menos relleno, aumentando la precisión.

Cabe aclarar que no está contemplado en el alcance de este PFC, definir un procedimiento que permita determinar objetivamente el valor óptimo del parámetro para cada técnica. Por ello, en este caso, la selección de este se basó en el criterio de los investigadores.

Dicho esto, los valores del parámetro seleccionados para cada técnica son:

- **TextRank: 10**
- **Feature Based: 13**

Independientemente de la técnica utilizada y del valor asignado al parámetro, es evidente la diferencia entre el recall y la precisión. Para entender porqué los valores difieren de esta manera, es necesario recordar que la precisión indica que proporción de palabras del sumario generado están también presentes en el sumario ideal. Los valores utilizados para el parámetro parten de la cantidad máxima de oraciones observables en un sumario, lo cual implica que los sumarios generados, en la gran mayoría de casos, se componen de contenido extra, que no está presente en los sumarios ideales.

Por otro lado, dado un conjunto de fallos, resulta muy difícil obtener valores altos para la precisión debido a que es complejo que un sumario se genere con una longitud similar a la de su referencia. La longitud de un sumario está ligada a la subjetividad con la que fue construido.

Por ello, se plantea el objetivo de evaluar principalmente qué proporción de las ideas principales del fallo judicial se encuentran contenidas en el sumario generado por las técnicas seleccionadas. Si bien los valores de la precisión obtenidos serán presentados, el recall será la métrica fundamental que determinará la calidad de los resúmenes generados.

Se concluye entonces, que dentro del contexto de este PFC **el recall es más relevante que la precisión**.

5.6.3. Evaluación "TextRank Summarizer"

A lo largo de esta sección, se presentan en distintas tablas, los resultados obtenidos al evaluar la totalidad del conjunto de datos segmentado, dividido en lotes, haciendo uso de la técnica "TextRank Summarizer". Los valores provistos en las tablas representan el valor promedio para cada métrica, resultado de evaluar un determinado lote.

	Lote 1		
	ROUGE 1	**ROUGE 2**	**ROUGE L**
Recall	0,640	0,438	0,590
Precisión	0,092	0,050	0,085
F1 Score	0,156	0,086	0,143

Tabla 11. Resultados obtenidos al procesar el Lote 1 con TextRank Summarizer.

	Lote 2		
	ROUGE 1	**ROUGE 2**	**ROUGE L**
Recall	0,627	0,433	0,578
Precisión	0,099	0,054	0,090
F1 Score	0,164	0,091	0,150

Tabla 12. Resultados obtenidos al procesar el Lote 2 con TextRank Summarizer.

	Lote 3		
	ROUGE 1	**ROUGE 2**	**ROUGE L**
Recall	0,627	0,433	0,578
Precisión	0,099	0,054	0,090
F1 Score	0,164	0,091	0,150

Tabla 13. Resultados obtenidos al procesar el Lote 3 con TextRank Summarizer.

	Lote 4		
	ROUGE 1	**ROUGE 2**	**ROUGE L**
Recall	0,543	0,354	0,489
Precisión	0,135	0,075	0,121
F1 Score	0,205	0,116	0,184

Tabla 14. Resultados obtenidos al procesar el Lote 4 con TextRank Summarizer.

	Lote 5		
	ROUGE 1	**ROUGE 2**	**ROUGE L**
Recall	0,545	0,343	0,487
Precisión	0,121	0,065	0,108
F1 Score	0,189	0,103	0,169

Tabla 15. Resultados obtenidos al procesar el Lote 5 con TextRank Summarizer.

	Lote 6		
	ROUGE 1	**ROUGE 2**	**ROUGE L**
Recall	0,540	0,347	0,483
Precisión	0,142	0,083	0,127
F1 Score	0,212	0,124	0,190

Tabla 16. Resultados obtenidos al procesar el Lote 6 con TextRank Summarizer.

5.6.4. Evaluación "Feature Based Summarizer"

A continuación, se presentan en distintas tablas, los resultados obtenidos al evaluar la totalidad del conjunto de datos segmentado, dividido en lotes, haciendo uso de la técnica "Feature Based Summarizer". Los valores provistos en las tablas representan el valor promedio para cada métrica, resultado de evaluar un determinado lote.

	Lote 1		
	ROUGE 1	**ROUGE 2**	**ROUGE L**
Recall	0,516	0,305	0,465
Precisión	0,116	0,055	0,103
F1 Score	0,176	0,087	0,157

Tabla 17. Resultados obtenidos al procesar el Lote 1 con Feature Based Summarizer.

	Lote 2		
	ROUGE 1	**ROUGE 2**	**ROUGE L**
Recall	0,508	0,297	0,455
Precisión	0,117	0,057	0,104
F1 Score	0,180	0,090	0,161

Tabla 18. Resultados obtenidos al procesar el Lote 2 con Feature Based Summarizer.

	Lote 3		
	ROUGE 1	**ROUGE 2**	**ROUGE L**
Recall	0,493	0,293	0,437
Precisión	0,142	0,077	0,126
F1 Score	0,209	0,114	0,185

Tabla 19. Resultados obtenidos al procesar el Lote 3 con Feature Based Summarizer.

	Lote 4		
	ROUGE 1	**ROUGE 2**	**ROUGE L**
Recall	0,448	0,256	0,396
Precisión	0,154	0,083	0,137
F1 Score	0,217	0,116	0,192

Tabla 20. Resultados obtenidos al procesar el Lote 4 con Feature Based Summarizer.

	Lote 5		
	ROUGE 1	**ROUGE 2**	**ROUGE L**
Recall	0,453	0,251	0,348
Precisión	0,137	0,068	0,121
F1 Score	0,201	0,101	0,228

Tabla 21. Resultados obtenidos al procesar el Lote 5 con Feature Based Summarizer.

	Lote 6		
	ROUGE 1	**ROUGE 2**	**ROUGE L**
Recall	0,468	0,289	0,419
Precisión	0,167	0,097	0,149
F1 Score	0,233	0,134	0,208

Tabla 22. Resultados obtenidos al procesar el Lote 6 con Feature Based Summarizer.

5.6.5. Análisis de resultados

Una vez presentados los resultados numéricos resulta importante ofrecer una interpretación y análisis más profundo de los mismos.

Dado el caso de estudio de este PFC, y como fue mencionado en la *sección 5.6.2*, el recall es la medida fundamental para la toma de decisiones. Por ello, a continuación, se presenta una ilustración comparando los resultados promedios obtenidos para dicha métrica, dadas las tres variantes de ROUGE.

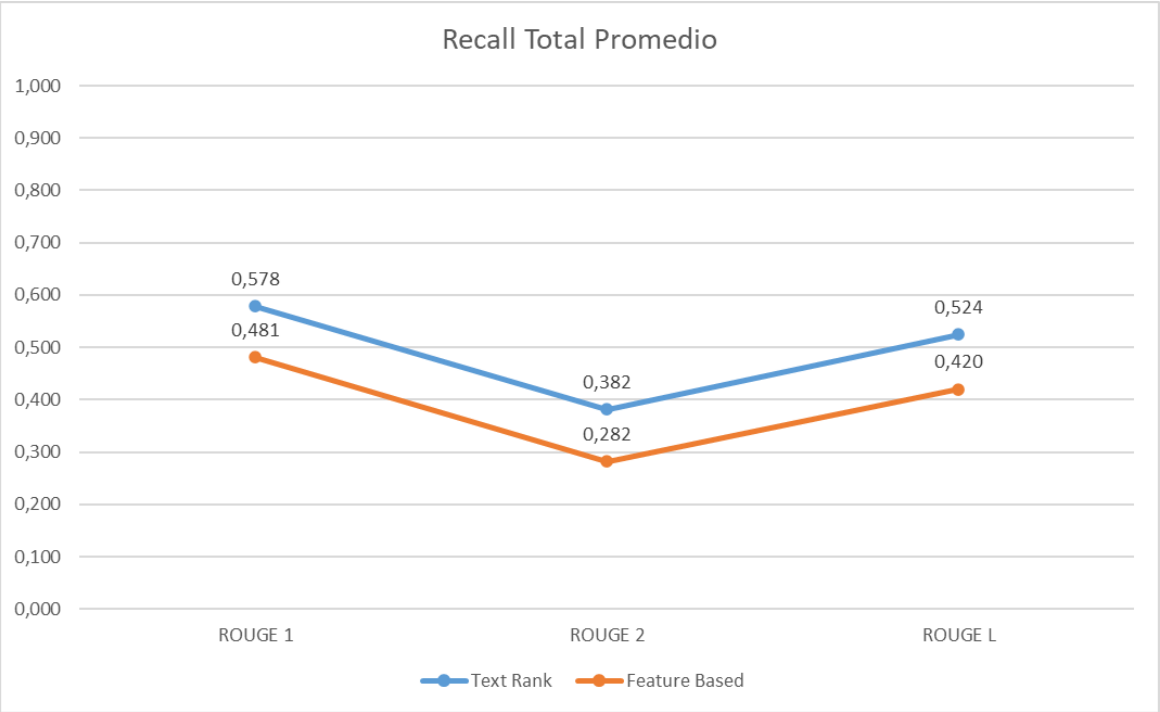

Figura 22. Recall Total Promedio

En la *Figura 22* es posible observar que la puntuación promedio total de recall obtenida con TextRank es, para todas las variantes de ROUGE, mayor a la de Feature Based.

Resulta complejo determinar por qué una técnica funciona mejor que la otra ya que la lógica detrás de ellas es completamente distinta. Sin embargo, hablando de la implementación en concreto, se deduce que al implementar TextRank como parte de un pipeline (spaCy), se dispone de modelos entrenados que permiten realizar tareas de tokenización de manera más eficiente, resultando en una mejor construcción de los sumarios.

Para complementar este análisis de una manera ilustrativa, es interesante observar casos reales de sumarios generados y compararlos con sus sumarios de referencia. Para seleccionar estos casos se recopilaron del conjunto de resultados resúmenes automáticos que abarcan un rango de valores de recall de ROUGE L. Se elige esta variante porque combina los aspectos medidos por ROUGE 1 y 2, es decir, mide la proporción de palabras coincidentes en secuencia. Esto

resulta beneficioso ya que es posible evitar problemas asociados a las puntuaciones, abreviaciones y/o conectores que pueden estar presentes en el sumario de referencia y no en el sumario generado.

En las siguientes figuras se puede observar del lado izquierdo los sumarios de referencia, y a la derecha sus respectivos sumarios generados con TextRank. Los textos han sido difuminados para mantener la confidencialidad de los documentos.

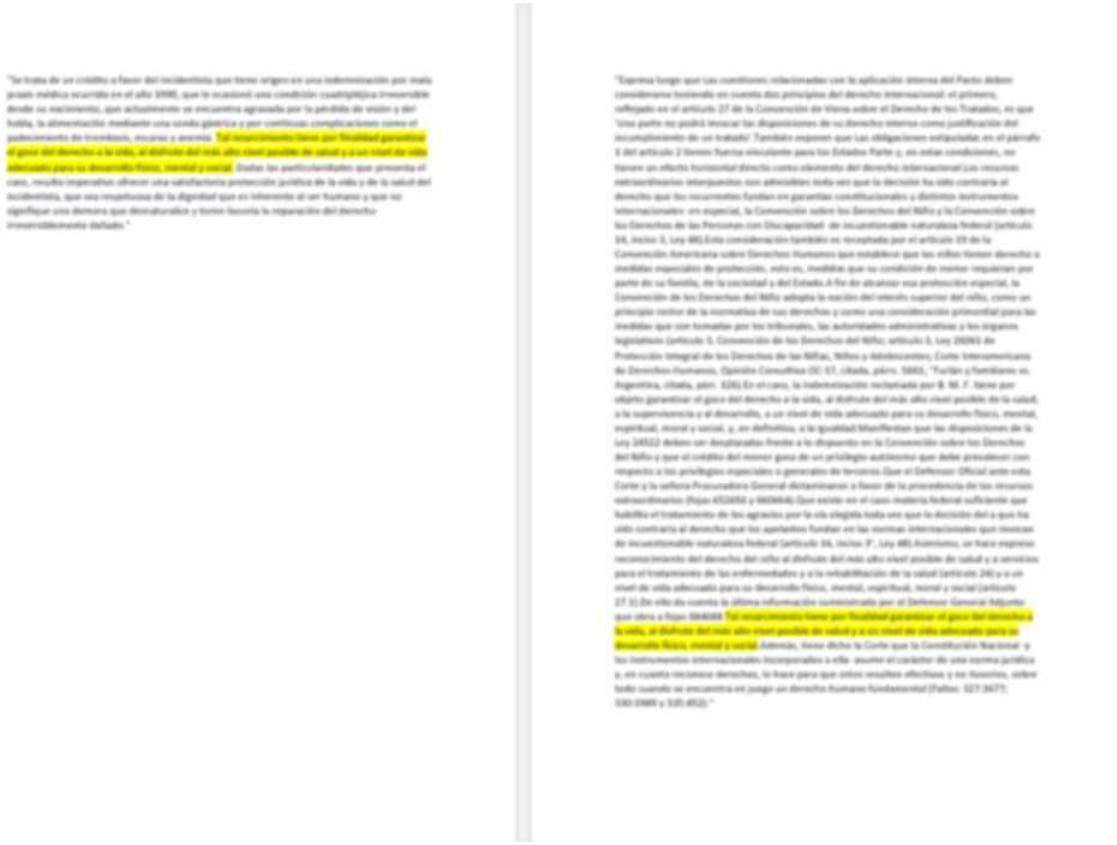

Figura 23. Sumario generado y de referencia – Recall = 0,4.

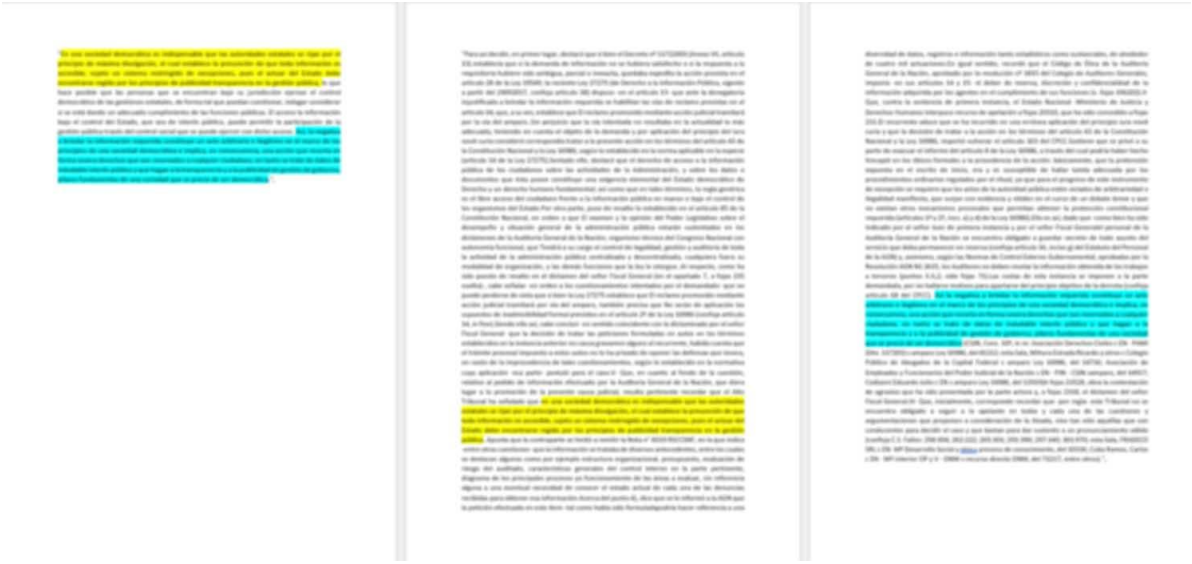

Figura 24. Sumario generado y de referencia – Recall = 0,7.

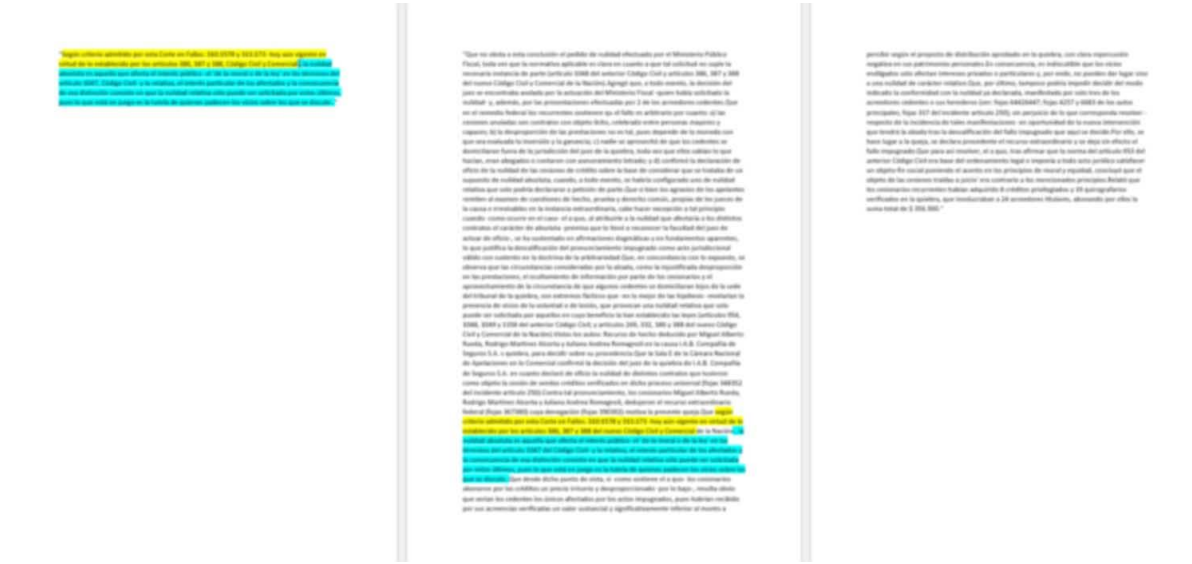

Figura 25. Sumario generado y de referencia – Recall = 0,9.

Analizando las figuras es posible concluir que aquellos sumarios generados que arrojan un recall igual o mayor a 0,7 poseen una proporción del contenido de su referencia aceptable. En base a este criterio de aceptabilidad es posible determinar que un 31,5% de los documentos cumplen con el mismo.

5.7. Análisis de factibilidad

En base al análisis realizado sobre los resultados lo primero a destacar es que, si una técnica es factible de ser usada, la misma es TextRank, simplemente por su rendimiento superior.

Haciendo foco en esta técnica, se observa que solo un **31,5%** de los sumarios generados cumplen con el criterio de aceptabilidad definido. Este número no representa una proporción mayoritaria del conjunto de resultados, por lo que resulta inapropiado recomendar esta técnica como herramienta generadora de sumarios automáticos. Se observa que los resultados no son lo suficientemente completos como para asegurar que ningún concepto fundamental del fallo está ausente. Esto constituye un riesgo que no puede ser ignorado. Los resultados sugieren que las técnicas de resumen automático actuales investigadas no terminan de igualar el criterio de un ser humano con una efectividad admisible.

Sin embargo, teniendo estas limitaciones en cuenta, y bajo el criterio de los investigadores, se concluye que **sí resulta factible el uso de técnicas extractivas como parte de una herramienta de soporte** para los sumariantes. Se considera que los potenciales beneficios de este tipo de tecnologías aportarían una mejora en tiempos de lectura y confección de sumarios, alivianando la pesada carga a la que se enfrentan los sumariantes. Poder controlar las longitudes de los sumarios generados permitiría construir fallos de menor tamaño, sin igualar a los sumarios, pero aumentando la probabilidad de captar las ideas principales. Es más rápido y conveniente leer un texto que represente el 30% o 40% de un fallo judicial y contenga la idea principal, a enfrentarse a leer un fallo entero.

Se propone entonces hacer un uso preliminar de este tipo de técnicas a la hora de generar sumarios. Esto es, utilizar los resúmenes automáticos como una guía a la hora de escribir los sumarios.

6. Conclusiones

A lo largo del proyecto surgieron desafíos de distinta naturaleza, desde aspectos organizativos hasta otros técnicos y conceptuales, que fueron requiriendo de nuestra atención. A la hora de elaborar conclusiones decidimos agruparlas en dos tipos.

6.1. Sobre la planificación

A la hora de establecer una metodología de trabajo se fueron proponiendo distintos enfoques y prácticas, que concluyeron en la metodología detallada anteriormente. Consideramos oportuno destacar varias características que hicieron que el trabajo sea eficiente: definición de metas a corto plazo, herramientas de planificación y división de tareas, trabajo en paralelo y virtualidad, frecuencia pactada para reuniones, entre otras.

A pesar de todos estos puntos fuertes también cabe mencionar que surgieron dificultades que impidieron el desarrollo del proyecto en los tiempos estimados. Revisando el Plan de Proyecto notamos que algunos de los riesgos relacionados a la planificación identificados y pronosticados fueron efectivamente acertados. Además de presenciar una reducción en el tiempo disponible para el desarrollo del proyecto por cuestiones laborales de ambos integrantes, creemos que a la hora de cumplir con los plazos pactados el principal contratiempo se debió a la subestimación de esfuerzos. A su vez, esta subestimación está muy ligada a la falta de familiaridad con los temas tratados, y con la misma gestión del proyecto en general, lo cual se abarca en la siguiente sección.

6.2. Sobre el desarrollo

Lo primero que podemos concluir acerca de las tareas de investigación y desarrollo a lo largo de todo el proyecto es la presencia de un aprendizaje continuo.

Progresivamente fue notable el incremento en el grado de entendimiento acerca de los tópicos tratados, lo cual derivó en una mayor eficiencia a la hora de realizar tareas e incluso mejores estimaciones. Sin embargo, este incremento en el entendimiento resultó en la detección de errores introducidos en etapas temprana del proyecto, lo que llevó a revisiones con mucha frecuencia. Podemos diferenciar este grado de entendimiento en dos aspectos. Primeramente, y yendo al aspecto legal, logramos una familiaridad con términos, lenguaje, procedimientos, y demás cuestiones que eran completamente ajenas a nosotros al comienzo del proyecto. Por otra parte, y quizás con más relevancia personal, notamos también una gran adquisición de conocimientos acerca de cuestiones técnicas, como inteligencia artificial y lenguajes de programación, que están ligadas profundamente a nuestra formación como Ingenieros en Sistemas de Información.

7. Trabajos Futuros

En esta sección se presentan distintos puntos claves que fueron analizados a lo largo del proyecto e implican un desarrollo a futuro. Estas cuestiones pueden dividirse en dos grandes grupos:

7.1. Mejoras a partir del trabajo existente

A lo largo del PFC se presentaron diversas cuestiones asociadas a las limitaciones de las técnicas que derivaron en la decisión de utilizarlas como parte de una herramienta de soporte y no como una herramienta generadora de sumarios. Para poder construir dicho objetivo, se cree que los siguientes conceptos, pueden ser investigados en otro tipo de proyecto a futuro:

- Estudiar la relación entre la longitud de los fallos judiciales y la longitud de los sumarios. Esto permitiría obtener una forma dinámica de determinar la longitud que un sumario debería tener a partir de su respectivo fallo, de forma que el porcentaje de ideas principales capturadas aumente.
- Los fallos judiciales resultan ser textos muy complejos de leer y entender para personas que no pertenecen al ámbito. Esto deriva en que el preprocesamiento de los mismos se dificulte y la limpieza no sea óptima. Se propone entonces continuar con la mejora continua del preprocesamiento de los textos, ampliando la base de conocimientos asociada al ámbito legal.
- La construcción formal de una herramienta de soporte, tal como se propone, es el punto más relevante sobre el cual se puede continuar trabajando a futuro a partir del trabajo existente. Una herramienta que genere sumarios a partir de diversas técnicas aumenta la probabilidad de que todas las ideas principales estén contenidas en los diferentes resúmenes generados y mejora en comparación a la de depender de una sola técnica.

Disponemos el uso del código desarrollado y utilizado en este proyecto para continuar colaborando con la comunidad. El mismo se encuentra disponible en el siguiente repositorio de Github: https://github.com/JIGrosso/Proyecto-Final-Carrera

7.2. Nuevas alternativas

Luego de analizar y revisar gran contenido asociado al PLN y al resumen automático de textos, es notorio que ambos campos están en pleno auge, innovando continuamente. Por ello, resulta interesante explorar nuevas alternativas para la resolución de la problemática planteada, apostando a técnicas distintas, incluso del tipo abstractivas.

8. Bibliografía

1. **Mercedes Martínez González, Dámaso-Javier Vicente Blanco.** *Estructura de los documentos jurídicos y XML.* España : Universidad de Valladolid, 2014.
2. **Suprema Corte de Justicia.** *Manual de estilo.* 2017.
3. **Sparck Jones, K.** *Automatic summarizing: factors and directions.* Cambridge : In: Advances in Automatic Text Summarization. MIT Press, 1999.
4. **Mani, I., Maybury, M.** *Introduction. In: Advances in Automatic Text Summarization.* Cambridge : MIT Press, 1999.
5. **A. C. Cardoso, M. A. Pérez Abelleira.** *Generación automática de resúmenes.* s.l. : Universidad Católica de Salta, Facultad de Ingeniería e IESIING, 2013.
6. **Roy, Abhijit.** *Understanding Automatic Text Summarization: Extractive Methods.* s.l. : Towards Data Science, 2020.
7. **Adhika Pramita Widyassari, Supriadi Rustad.** *Review of automatic text summarization techniques & methods.* Indonesia : Journal of King Saud University – Computer and Information Sciences, 2020.
8. **Chauhan, Kushal.** *Unsupervised Text Summarization using Sentence Embeddings.* s.l. : Medium, 2018.
9. **Stuart J. Russell, Peter Norvig.** *Inteligencia Artificial: Un Enfoque Moderno.* 2009.
10. **ÇELİK, Özer.** *A Research on Machine Learning Methods and Its Applications.* Eskisehir, Turkey : Osmangazi University, 2018.
11. **Simeone, Osvaldo.** *A Very Brief Introduction to Machine Learning.* King's College London, United Kingdom : IEEE, 2018.
12. **Piyush Madan, Samaya Madhavan.** *An introduction to deep learning.* s.l. : IBM, 2020.
13. **Vidyadhar Upadhya, P S Sastry.** *An Overview of Restricted Boltzmann Machines.* Bangalore, India : Journal of the Indian Institute of Science, 2019.
14. **Chowdhury, Gobinda G.** *Natural Language Processing.* Glasgow, UK : Dept. of Computer and Information Sciences, University of Strathclyde, 2003.
15. **Roldós, Inés.** *Major Challenges of Natural Language Processing.* s.l. : MonkeyLearn Blog, 2020.
16. **Daniel W. Otter, Julian R. Medina, and Jugal K. Kalita.** *A Survey of the Usages of Deep Learning for Natural Language Processing.* s.l. : IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, 2019.
17. **Corvi, Julieta Pilar.** *Resumen extractivo de documentos. Un análisis comparativo de técnicas de puntuación.* La Plata, Buenos Aires : Universidad Nacional de la Plata, 2019.
18. **Josef Steinberger, Karel Jezek.** *Evaluation Measures for Text Summarization* . Plzen, Czech Republic : University of West Bohemia, 2009.
19. **Lin, Chin-Yew.** *ROUGE: A Package for Automatic Evaluation of Summaries.* University of Southern California : Information Sciences Institute, 2004.
20. **Gómez, Jorge Verdeguer.** *Generación de resúmenes de textos.* España : Universitat Politècnica de València, 2019.

21. **Korstanje, Joos.** *The F1 score.* s.l. : Towards Data Science, 2021.
22. **Pressman, Roger S.** *Ingeniería de Software: Un Enfoque Práctico, 7ma Edición.* 2010.
23. **Deepali Jain, Malaya Dutta Borah, Anupam Biswas.** *Summarization of legal documents: Where are we now and the way forward.* India : Computer Science Review, Volume 40, 2021.
24. **Fermín Cruz, José A. Troyano, Fernando Enríquez, F. Javier Ortega.** *TextRank como motor de aprendizaje en tareas de etiquetado.* España : Dep. de Lenguajes y Sistemas Informáticos, Universidad de Sevilla, 2014.
25. **Tarau, Rada Mihalcea and Paul.** *TextRank: Bringing Order into Text.* Barcelona, Spain : In Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing, pages 404–411. Association for Computational Linguistics., 2004.
26. **Singh, Taranjeet.** *Natural Language Processing With spaCy in Python.* s.l. : Real Python Tutorials, 2019.
27. **Nathan, Paco.** *PyTextRank - Python implementation of TextRank for lightweight phrase extraction.* s.l. : Spacy.io, 2017.
28. **Baxendale, P.** *Machine-Made Index for Technical Literature: An Experiment.* s.l. : IBM Journal of Research and Development Volume 2 Issue 4, 1958.
29. **Sukriti Verma, Vagisha Nidhi.** *Extractive Summarization using Deep Learning.* Shahbad Daulatpur : Delhi Technological University, 2019.
30. **Ningyu Zhang, et al.** *Named-entity recognition.* s.l. : DeepIA, 2022.
31. **NewsCatcher Engineering Team.** *Ultimate Guide To Text Similarity With Python.* s.l. : NewsCatcher Blog, 2022.