1 PROCESAMIENTO DE LENGUAJE NATURAL

El *procesamiento de lenguaje natural (PLN)* es la función de componentes de software o hardware en un sistema de cómputo que analiza o sintetiza el lenguaje hablado o escrito (Jackson y Moulinier, 2002). Para tener una mejor comprensión del término procesamiento de lenguaje natural es necesario saber qué es un lenguaje natural. Un *lenguaje natural* es aquel propiamente usado por los humanos para comunicarse entre ellos, como el español, el inglés, el francés, entre otros. Se diferencia de los *lenguajes artificiales*, los cuales son creados para que haya una comunicación humano-máquina, como los lenguajes de programación.

Las aplicaciones y usos de PLN se encuentran en muchas de las cosas que empleamos en nuestra vida diaria. Casos en los cuales se usa el PLN son los motores de búsqueda en línea como Google o Yahoo!, los traductores y resumidores automáticos, los correctores de estilo y ortografía de los procesadores de texto, los reconocedores de voz, entre otros.

Además, las capacidades del procesamiento de lenguaje natural son muy grandes: permiten disminuir y/o facilitar tareas que anteriormente se realizaban de manera manual. Pero también han logrado que se lleven a cabo en menor tiempo o que sean realizables; por ejemplo, una persona jamás podría analizar toda la cantidad de información que existe en la Biblioteca del Congreso de los Estados Unidos de América, al menos de manera manual. Con el PLN, esto es posible.

Pero detrás de las aplicaciones basadas en el PLN hay varios procesos en los que es necesario realizar un tratamiento de la lengua, escrita o hablada, para que pueda ser analizada por una computadora. Los procesos realizados tratan de simular el proceso que lleva a cabo el ser humano para comprender y analizar la lengua. La diferencia entre este proceso y el efectuado por los seres humanos, es que una computadora puede analizar enormes masas de datos a velocidades muy rápidas, aunque no de manera tan exacta y precisa como lo hacemos los humanos.

En este capítulo se abordará el PLN, también conocido como tratamiento de lenguaje natural. Para ello en una primera instancia se darán a conocer algunas de las herramientas básicas que son empleadas dentro del PLN; posteriormente, dado que el objetivo es el uso del PLN para la extracción de la terminología, se dará a conocer lo que es la recuperación de información como una de sus aplicaciones, así como las diversas técnicas de la recuperación de información que son utilizadas en esta tesis.

1.1 Recursos y herramientas empleadas en PLN

A continuación se darán a conocer algunas de las herramientas y recursos más importantes empleados en PLN. Si bien son muchas las herramientas y los recursos existentes, los principales son los corpus lingüísticos, los tokenizadores, las listas de palabras vacías, entre otros.

1.1.1 Corpus lingüísticos

Uno de los recursos básicos empleados en el PLN son los *corpus lingüísticos*. El nombre de corpus proviene del latín y según el Diccionario de la Real Academia Española (DRAE), en su edición 22, indica que un corpus "es un conjunto lo más extenso y ordenado posible de datos o textos científicos, literarios, etc., que pueden servir de base a una investigación". Aunque esta definición da a conocer, de manera general, lo que es un corpus, la definición de corpus lingüístico es un poco más específica.

Un *corpus lingüístico*, según Sierra (2008), consiste en la recopilación de un conjunto de textos de materiales escritos y/o hablados, agrupados bajo un conjunto de criterios mínimos, para realizar ciertos análisis lingüísticos. Su función principal es establecer la relación entre la teoría y los datos (Torruella y Llisterri, 1999); es decir, un corpus debe cumplir los modelos teóricos con datos reales.

Todo corpus lingüístico es creado con base en determinados requerimientos, según nuestras necesidades y el tipo de análisis que se vaya a llevar a cabo con él. Aun así, existen una serie de parámetros, o de criterios mínimos, que se deben seguir para que pueda cumplir su función principal de manera exitosa, las cuales se explican a continuación:

Variedad: Este parámetro indica que los recursos que conforman el corpus deben ser diversos. En este sentido, los documentos pueden provenir de distintas fuentes, épocas, hablantes, lugares, entre otros. Por ejemplo, si se busca crear un corpus de ciencia ficción, no se puede basar solamente en la serie de libros de "Dune" de Frank Herbert, se debe de incluir de igual manera libros como "Yo robot" de Isaac Asimov, "La guerra de los mundos" de Herbert George Wells, hasta "20 mil leguas de viaje submarino" de Julio Verne.

Representatividad: Esto se refiere a que el corpus abarque, de la manera más amplia posible, todas las formas y variedades que existen en la lengua en determinada área, tema o tiempo. Es como en probabilidad, cuando se busca analizar una población sumamente grande se toma una muestra, la cual debe representar de manera general a todo el conjunto poblacional. En el caso de corpus lingüísticos, la población es la lengua a analizar mientras que el conjunto de documentos es la muestra a emplear.

Equilibrio: El equilibro de un corpus está relacionado con la existencia de una neutralidad en todos los aspectos que se buscan analizar. Y aunque este parámetro es difícil de cumplir, se debe mantener lo más posible, de lo contrario los análisis que se lleven a cabo con el corpus podrían estar sesgados o limitar el uso del corpus en otras investigaciones.

Tamaño: Un corpus debe tener un tamaño que permita obtener resultados significativos. Según MacMullen (2003) los corpus muy grandes no son necesariamente representativos; de hecho pueden causar problemas para encontrar elementos menos abundantes pero más importantes.

Manejable por la computadora: En la actualidad muchos de los corpus lingüísticos se encuentran de manera digital; esto se recomienda ya que se pueden obtener beneficios de ello, por ejemplo se pueden realizar análisis o búsquedas de manera más rápida, ya que antes se realizaban estos procesos de forma manual. Esto ha dado lugar al término de corpus informatizado, es decir, un conjunto de textos elegidos y anotados con ciertas normas y criterios para el análisis lingüístico, de forma que se sirve de la tecnología y de las herramientas computacionales para generar resultados más exactos (Sierra, 2008).

Derechos de autor: Este se debe de tener en cuenta cuando se realiza un corpus. La razón de ello se debe a que es necesario no violar las leyes o normatividades con respecto al uso o reproducción de documentos, ya que un corpus está conformado por documentos creados por distintas personas.

Existe una gran cantidad de tipos de corpus y según Torruella y Llisterri (1999), se pueden clasificar de la siguiente manera: por el porcentaje y la distribución de los diferentes tipos de textos que lo conforman, por la especificidad de los textos que lo componen, por la cantidad de texto que se recoge en cada documento, por la codificación y las anotaciones añadidas a los textos y por la documentación que lo acompañe.

Algunos ejemplos de corpus lingüísticos son el Brown Corpus (Francis y Kučera, 1979), el Corpus Técnico del IULA (Vivaldi, 1995), el Corpus Histórico del Español en México (Medina y Méndez, 2006), el Corpus de Referencia del Español Actual (CREA) y el Corpus Diacrónico del Español (CORDE), estos dos últimos creados por la Real Academia Española.

1.1.2 Tokenizadores

Antes de realizar cualquier análisis lingüístico o de procesar un documento es necesario encontrar y separar cada uno de los elementos que lo conforman, para ello se emplean los tokenizadores. Los *tokenizadores* (también conocidos como analizadores léxicos o segmentadores de palabras) segmentan un conjunto de caracteres en unidades con significado llamados *tokens* (Jackson y Moulinier, 2002), que se pueden llamar igualmente *casos*. A continuación se presenta un ejemplo:

Frase: La niña juega con la pelota

Tokens:	La	niña	juega	con	la	pelota

Como se puede observar en el ejemplo anterior, la frase es segmentada en seis tokens, empleando como delimitador de cada token el espacio en blanco. De igual manera podemos ver que en los tokens se cuentan las repeticiones, en el caso anterior, la palabra "la" aparece dos veces y se contabilizan ambas. En el caso en que nos refiramos a la clase de todos los casos contenidos con la misma secuencia de caracteres (Manning et al., 2008), en otras palabras, el número de distintos tokens, entonces estamos dando a conocer lo que se define como *type* o *tipo*. Por tanto, en el ejemplo anterior existen 5 tipos, es decir, "la" se contabiliza una sola vez.

Este proceso, aunque aparenta ser fácil de llevar a cabo, es una tarea compleja. Existen casos en el cual la regla de espacios en blanco como delimitador no es suficiente, algunos de ellos son los siguientes (Ananiadou y McNaught, 2006):

Fronteras ambiguas: Existen lenguas, como el alemán, que son aglutinantes, donde un conjunto de palabras se unen y no existe algún espacio en blanco entre ellas. Por ejemplo, estación de autobús en alemán es "busbahnhof" que es la unión entre "bus" que significa autobús y "bahnhof" que significa estación. Por tanto, en este tipo de casos es difícil encontrar la frontera entre las dos palabras ya que no existe un espacio de por medio. De igual forma pasa en lenguas como el japonés y el chino, donde no se escribe ningún tipo de espacio entre los caracteres.

Formatos: Fechas, números, teléfonos, entre otros, se escriben de distinta manera dependiendo del lugar de donde sea un documento, por tanto se debe prevenir esto para realizar la correcta tokenización. Por ejemplo, la notación que se emplea para separar los números, en algunos países se emplea la coma decimal en lugar del punto decimal; de igual manera algunos usan el punto, la coma o un espacio en blanco para separar los millares.

Guiones: Otro de los casos son los elementos unidos por guiones, donde estos se puede considerar como un solo token o como varios. Algunos ejemplos son los siguientes, compra-venta, México-68, entre otros. De igual manera, en otros idiomas se emplea de manera distinta el guión, por ejemplo en el francés se emplea el guión para indicar una posición inversa a la normal de los elementos como "dis-moi!". En portugués peninsular se unen con guión los clíticos "parece-me".

Abreviaturas y siglas: Además del espacio en blanco como delimitador, se podría considerar a los signos de puntuación como límite entre palabras gráficas, pero el caso de las abreviaturas y de las siglas complica esto. Por consiguiente hay que diferenciar entre los puntos del final de una oración y los de una abreviatura o sigla, de lo contrario habría problemas en casos como F. C. (ferrocarril) o S.C.T. (Secretaría de Comunicaciones y Transportes).

Apóstrofos: Aunque en el español no se emplean los apóstrofos (‘), en otras lenguas sí se emplean, como en el inglés y en el francés. Ejemplo: "L'hôpital" (en francés el hospital) o "isn't" (en inglés, contracción de is not).

Según Jackson y Moulinier (2002) los tokenizadores usualmente dependen de reglas, máquinas de estados finitos, modelos estadísticos, y lexicones para identificar abreviaturas o palabras de varios elementos.

1.1.3 N-gramas

La definición más simple de un *n-grama* es la unión de uno o varios tokens o caracteres. La construcción de los n-gramas se lleva por medio de combinaciones entre tokens o caracteres vecinos. Para llevar esto a cabo se crea una ventana del tamaño del n-grama deseado: si se quiere un unigrama el tamaño de la ventana será de 1, si son bigramas (o digramas) la ventana será de tamaño 2 y así sucesivamente hasta llegar al tamaño de n-grama final deseado. Esta ventana se mueve a través del texto y abarca la cantidad de tokens o caracteres que el tamaño de la ventana indica. Un ejemplo de formación de n-gramas de tokens es el siguiente:

Frase: El termómetro de mercurio se rompió

Tokens

El	termómetro	de	mercurio	se	rompió

Unigramas

El	termómetro	de	mercurio	se	rompió

Bigramas

El termómetro	termómetro de	de mercurio	mercurio se	se rompió

Trigramas

El termómetro de	termómetro de mercurio	de mercurio se	mercurio se rompió

Se emplean los n-gramas de tokens para abarcar la mayor cantidad de construcciones que se encuentran en un texto y no emplear en el análisis solamente construcciones de un elemento. Como se puede observar en el ejemplo anterior, en trigramas obtenemos una construcción que es importante en la frase y expresa claramente un objeto, esta es "termómetro de mercurio".

Los n-gramas de caracteres se emplean para obtener construcciones que frecuentemente se emplean dentro de los tokens en una determinada lengua. Por ejemplo, con el uso de los n-gramas de caracteres se puede conocer el idioma de un texto sin haberlo leído (Cavnar y Trenkle, 1994).

1.1.4 Etiquetadores de partes de la oración

Los etiquetadores de partes de la oración, también conocidos como etiquetadores POS por sus siglas en inglés "Part-of-Speech", son herramientas que realizan el proceso de asignar partes de la oración u otra clase de marcador léxico a cada palabra en un corpus (Jurafsky y Martin, 2008), en otras palabras, son sistemas que ayudan en la determinación de la categoría gramatical (por ejemplo, si es verbo, sustantivo, preposición, etcétera) de cada una de las palabras de un texto o conjunto de ellos. Asimismo, el etiquetado que realizan estas herramientas se aplica de igual forma a signos de puntuación, números, cantidades, entre otros.

Para llevar el proceso del etiquetado POS es necesario primeramente realizar un análisis morfológico. Un *análisis morfológico* o *análisis estructural* es el proceso de descomponer palabras complejas en sus componentes morfológicos (partes significantes de las palabras) (Bellomo, 2009), este análisis provee información sobre la semántica de la palabra y el papel sintáctico que juega en una oración (Goyal y Singh Lehal, 2008). En otras palabras, un análisis morfológico permite conocer las posibles categorías gramaticales de cada una de las palabras que se encuentran en una oración. A continuación se muestran los casos que se obtendrían al analizar morfológicamente la frase "El gato come pescado":

Frase:		El	gato	come	pescado
Categoría gramatical	**Caso 1:**	Artículo masculino singular	Nombre común masculino singular	Verbo principal indicativo presente tercera persona del singular	Sustantivo común masculino singular
	Caso 2:	-	-	Verbo principal imperativo segunda persona del singular	Verbo principal participio singular masculino

Como se puede observar en el ejemplo anterior, para "come" y "pescado" existen dos posibles casos; "come" puede representar un verbo en imperativo o en presente en tercera persona del singular, la razón es porque para los verbos terminados en –er estas dos formas terminan en –e. En cambio, la palabra "pescado" puede representar un sustantivo o un verbo en participio por su terminación –ado.

El segundo paso para hacer un etiquetado POS consiste en desambiguar los casos otorgados por el análisis morfológico y poner la etiqueta de la parte de la oración más probable. Para llevar a cabo esto es necesario utilizar etiquetadores que emplean algoritmos que pueden ser de dos tipos (Jurafsky y Martin, 2008):

Basados en reglas: Este tipo de algoritmo se basa en un conjunto de reglas escritas a mano que desambiguan los casos. Por ejemplo, una palabra será un sustantivo y no un verbo si está antecedido de un artículo.

Basado en aprendizaje: Este tipo de algoritmo se basa en las probabilidades que tiene una etiqueta en un determinado contexto. Para calcular estos valores es necesario utilizar previamente un corpus de aprendizaje, es decir, un corpus que fue etiquetado a mano lo suficientemente grande para que un programa computacional puede calcular las probabilidades de las etiquetas según el contexto.

Existen diversos formatos de etiquetas, incluso para la misma lengua; esto se debe a que no todos clasifican o anotan de la misma manera las partes de la oración. Sin embargo, existen diversos estándares, siendo el más conocido el desarrollado por el grupo EAGLES[1]

[1] http://www.ilc.cnr.it/EAGLES96/home.html

(Expert Advisory Group on Language Engineering Standards) ya que trata de establecer un mismo formato de etiquetas para las lenguas de la Unión Europea (español[2], inglés, francés, etcétera). Las etiquetas de EAGLES tienen una estructura variable según la categoría gramatical y el idioma, ya que no en todas las lenguas europeas las categorías gramaticales tienen los mismos atributos; en la Tabla 1 se muestra un ejemplo de esta estructura para el caso de adverbios.

Adverbios			
Posición	Atributo	Valor	Código
1	*Categoría*	*Adverbio*	*R*
2	*Tipo*	*General*	*G*
		Negativo	*N*

Tabla 1. Estructura de la etiqueta de adverbios para el formato EAGLES

Con base en la Tabla 1, adverbios como "rápido" o "siempre" obtendrían una etiqueta "RG" mientras que "jamás" o "no" tendrían una etiqueta "RN".

1.1.5 Lematizadores

En todo documento escrito en una lengua flexiva, como el español y el italiano, existen variaciones léxicas de las palabras, es decir, se pueden tener casos como "escribimos", "democracias", "industrialización", etcétera. Pero dentro del procesamiento de lenguaje natural es necesario disminuir la cantidad de variaciones léxicas que existan en los documentos a analizar. Para ello se debe obtener el *lema* o *forma canónica*, es decir, la base o la forma de diccionario de una palabra (Manning et al., 2008).

El proceso de lematización se lleva a cabo de manera automática por parte de los humanos; cuando queremos buscar las palabras "encontramos" o "niñas" en un diccionario las pasamos a "encontrar" y "niño", para ello empleamos nuestro conocimiento del mundo.

[2] Para información sobre las etiquetas EAGLES con ejemplos en español se puede consultar http://nlp.lsi.upc.edu/freeling/doc/userman/parole-es.pdf

Pero para las computadoras realizar este procedimiento es más complicado, ya que no tienen acceso a este conocimiento, por tanto, es necesario darle una serie de reglas y de recursos.

Para reducir el número de variaciones léxicas, ya sea por flexiones (*caminar* → *caminamos*) o por derivaciones (*activar* → *activación*), existen dos herramientas que se emplean en PLN, que son los *lematizadores* y los *truncadores* o *stemmers*. Aunque frecuentemente se confunden ambos términos, cabe aclarar que son dos métodos distintos.

Los lematizadores son herramientas que emplean diccionarios o tesauros, al igual que reglas, que buscan obtener el lema de las palabras; además estas herramientas realizan un etiquetado POS (sección 1.1.4) para conocer la categoría gramatical de las palabras.

A continuación se muestra un ejemplo de los lemas obtenidos por la lematización en dos frases:

Frase 1: El cuidado de las obras de arte es un trabajo arduo

Lemas:	el	cuidado	de	el	obra	de	arte	ser	un	trabajo	arduo

Frase 2: El museo fue cuidado por los policías

Lemas:	el	museo	ser	cuidar	por	el	policía

Se puede observar que en la frase 1 tanto la palabra “cuidado” como “trabajo”, al lematizarse quedan sin cambios, la razón de ello es que ambas palabras hacen referencia a un sustantivo, y no a un verbo en participio y a un verbo en presente en primera persona, respectivamente. En cambio, en la frase 2, la palabra “cuidado” al ser un verbo conjugado en participio, su lema es “cuidar”.

Los truncadores, en cambio, son herramientas que emplean la heurística, es decir, ciertas reglas, para cortar las partes finales de las palabras, con el fin de llegar a su lema; estas herramientas no usan ni analizadores morfológicos o etiquetadores POS, aunque hay truncadores que incluyen reglas para eliminar afijos derivacionales, como “-ción”, “-ía”. Al no realizar un análisis morfológico o un etiquetado POS los stemmers no encuentran la diferencia entre “cuidado” que proviene del verbo y “cuidado” que es un sustantivo, como lo hacen los lematizadores. De igual forma, el problema de los stemmers es que el hecho de

cortar las partes finales de las palabras no siempre implica que se obtendrá la forma canónica correcta. A continuación en la Tabla 2 se muestran algunos ejemplos de la truncación:

Palabra	Lema por un stemmer[3]
chicharrones	chicharron
torres	torr
torreón	torreon

Tabla 2. Ejemplos de lemas obtenidos por un stemmer

Como se puede observar en la tabla anterior, existen casos en el cual el lema es el correcto (siempre y cuando se omita la falta de acentos), pero en otros casos, como el de "torres", su lema dado es incorrecto, la razón de este caso es que la regla empleada en el truncador indica que cuando una palabra termina en "-es" es un plural y por tanto se debe de quitar; en casos como "meses" o "celulares" sí funciona la regla anterior. La mayor ventaja que tienen los stemmers sobre los lematizadores es la velocidad de procesamiento.

Algunas herramientas lematizadoras son el FreeLing (Padró et al., 2010) y el TreeTagger (Schmid, 1994); mientras que para truncar existen sistemas basados en el algoritmo de Porter (Porter, 1980), que comenzó para el idioma inglés, pero se ha llevado a otros idiomas.

1.1.6 Palabras funcionales

Las *palabras funcionales*, *palabras vacías* o *stop words*, son las palabras que "carecen" de significado. Estas palabras son las de mayor frecuencia y las que aportan la menor cantidad de información, entre ellas se encuentran los artículos, las preposiciones, las conjunciones, entre otras. De las palabras funcionales se crean *listas de paro* o *stoplists*.

Según Pazienza et al. (2005), las palabras funcionales son automáticamente extraídas de un corpus genérico como aquellas con la más alta frecuencia, y posteriormente son

[3] Los ejemplos fueron obtenidos de la página del proyecto "Snowball" el cual es un conjunto stemmers basados en algoritmo de Porter. La dirección web es la siguiente: http://snowball.tartarus.org/algorithms/spanish/stemmer.html

validadas por expertos humanos. De igual manera, se pueden agregar algunas otras palabras que se desean eliminar en PLN.

El objetivo de emplear stoplists en PLN es reducir la cantidad de datos a analizar. De igual manera disminuye el espacio en memoria o en disco empleado por las herramientas que analizan lenguaje natural. En el siguiente ejemplo se puede observar que se quitan las palabras funcionales:

Frase 1: El monitor de esa computadora se descompuso

Tokens:	el	monitor	de	esa	computadora	se	descompuso

Eliminando las palabras funcionales

Resultado:	monitor	computadora	descompuso

1.2 Recuperación de información

Existen múltiples tareas dentro de PLN, una de las más importantes y más utilizadas es la recuperación de información. La *recuperación de información*, también conocida como *búsqueda de información* o *information retrieval (IR)*, es el proceso por el cual se otorgan documentos relevantes (o información sobre ellos) a un usuario, según la consulta que se haya realizado (Kageura y Umino, 1998). Esta tarea es una de las más obvias que existe en PLN porque es una con la que se tiene mayor contacto; por ejemplo, se emplea la recuperación de información en buscadores en línea, en bibliotecas digitales, en el programa para buscar dentro de la computadora y prácticamente en cualquier medio electrónico.

Como se vio en el párrafo anterior, para llevar a cabo una recuperación de información, es necesaria una *consulta* o *query*, que puede ser la correcta o no para buscar la información, de igual manera puede estar mal escrita, con faltas de ortografía, con exceso de conectores o de datos inútiles, entre otros; lamentablemente, lo que se encuentre en esa consulta es la única pista que da el usuario para llevar a cabo la búsqueda necesaria. Por tanto, es necesario que todo sistema de recuperación de información tenga en cuenta las consideraciones anteriores.

Según Frakes y Baeza-Yates (1992) casi todos los sistemas de recuperación de información utilizan operadores booleanos o patrones de texto. Los primeros son empleados en sistemas de búsqueda donde existe una gran colección de documentos, como en el internet o en una biblioteca digital; en estos sistemas, cada documento es representado por una lista de palabras claves o de identificadores. De igual manera, en los sistemas booleanos, el usuario puede conectar los elementos de la consulta por medio de conectores lógicos. En cambio, en los sistemas basados en patrones, las búsquedas se basan en cadenas de texto o en expresiones regulares, estos sistemas se emplean dentro de documentos, o en colecciones pequeñas de archivos.

Existen diversos métodos para la creación de las listas de palabras claves que se emplean en los sistemas de recuperación de información booleanos. Uno de ellos es TF-IDF o term frequency–inverse document frequency, el cual se explicará en el siguiente apartado.

1.2.1 Term frequency – Inverse document frequency (TF-IDF)

Uno de los métodos empleados para la creación de listas de palabras clave para los sistemas de búsqueda de información es *TF-IDF* (*Term Frequency – Inverse Document Frequency*), el cual es la unión de dos métricas, la primera de ellas es la de *frecuencia del término*, *term frequency* o *TF* y la segunda que es la *frecuencia inversa de los documentos*, *inverse document frequency* o *IDF*.

El método de TF-IDF genera listas de palabras clave con una calificación o peso que indica qué tan relevante es la palabra con respecto al documento seleccionado y al corpus en general. Además, estas listas permiten calificar a los documentos del corpus con base en estas palabras clave, es decir, si las palabras clave tienen una gran peso, entonces el documento está más relacionado con ellas que uno con las mismas palabras clave pero con menor peso. Por tanto, cuando un usuario ingrese una consulta, los documentos que tengan las palabras de esa consulta con mayor peso serán los que muestre el sistema de búsqueda de información.

La primera medida, TF, es un sistema de pesos basado en la idea de que construcciones (palabras, frases, grupos de palabras) que frecuentemente ocurren en el texto

de documentos tienen alguna relación con el contenido de los textos (Salton y McGill, 1986). El cálculo de estos pesos se lleva a cabo calculando la frecuencia relativa[4] de cada una de las construcciones (también llamados términos) en cada uno de los documentos a analizar; esto se puede representar por medio de la siguiente fórmula:

$$TF_{i,j} = f_i^j \qquad (1)$$

Donde f es el número de ocurrencias del término i en el documento j.

La segunda métrica, llamada Inverse Document Frequency o IDF, está basada en contar el número de documentos de la colección en donde se busca, que contienen (o están indizadas por) el término en cuestión (Robertson, 2004), en otras palabras, es saber el número de documentos de un corpus en donde las construcciones de palabras se encuentran. El nombre inicial de IDF fue *term specificity* y su fórmula fue la siguiente (Spärck-Jones, 1972):

$$term\ specificity = f(N) - f(n) + 1 \qquad (2)$$

Donde la función $f(x) = y$ tal que $2^{y-1} < x \leq 2^y$, en la Figura 1 se puede observar los valores que podría tomar y; N es el número de documentos que existen en el corpus y n es el número de documentos donde el término analizado se encuentra.

[4] La frecuencia relativa es el número de ocurrencias de un caso en determinado evento. Se diferencia de la frecuencia absoluta, que es la frecuencia relativa entre la suma de las ocurrencias de todos los casos en el evento.

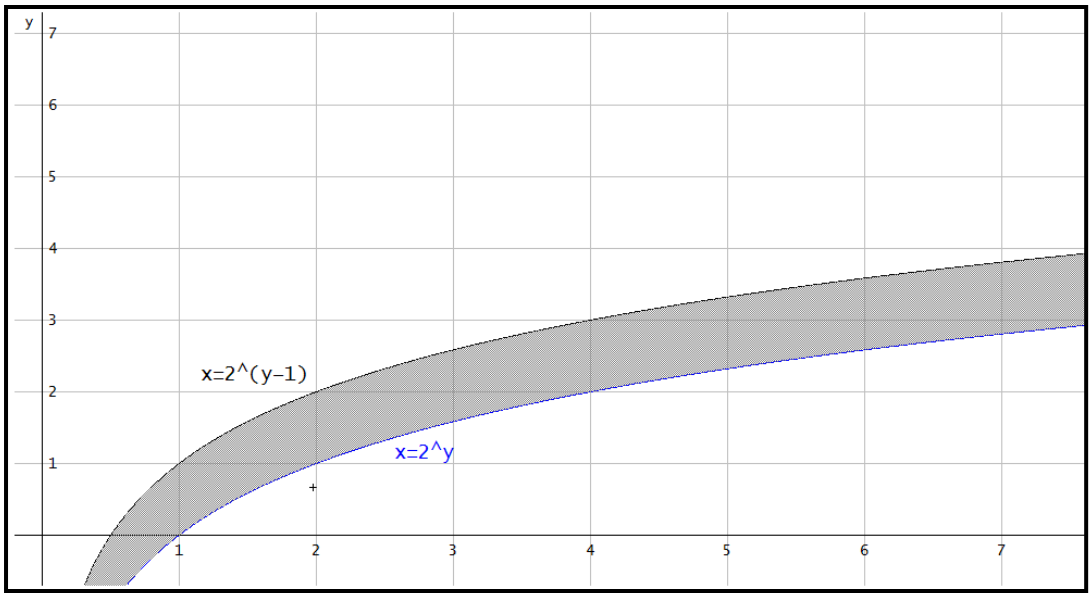

Figura 1. Gráfica de los valores de la función $f(x) = y$ tal que $2^{y-1} < x \leq 2^y$

Posteriormente la fórmula de *term specificity* fue adecuada por Robertson (1972) debido a que la fórmula $f(x)$ se podía aproximar a $f(x) \approx \log_2 x$ (esta aproximación es la función inversa de $x = 2^y$) y el 1 que se encontraba en la fórmula original era para evitar valores iguales a 0 en valores de n muy cercanos al valor de N, por lo tanto se podía prescindir de él. De igual forma le fue cambiado el nombre a inverse document frequency (IDF) y se observó que el logaritmo no tenía que ser forzosamente base 2, esto debido a la siguiente propiedad de los logaritmos:

$$log_b\, a = \frac{log_c\, a}{log_c\, b} \qquad (3)$$

Donde a es el número al cual se le calculará el logaritmo; b es la base del logaritmo original y c es la base de un logaritmo distinto a b. En el caso del IDF, se cambió el logaritmo de base 2 a base 10 debido a que es uno de los más usados.

La fórmula empleada en la actualidad para el cálculo de la métrica Inverse Document Frequency es la siguiente:

$$IDF_i = log \frac{N}{n_i} \qquad (4)$$

Donde N es el número de documentos que existen en el corpus, n es el número de documentos donde el término i se encuentra.

La creación del sistema de pesos basado en IDF se fundó en la teoría de Spärk-Jones que indicaba que los términos con alta frecuencia pueden ser útiles para aumentar la cobertura, pero las correspondencias entre la consulta y los términos del documento que ocurren raramente en una colección de documentos deberían ser tomadas como más importantes que aquellas que ocurren frecuentemente (Salton y Yang, 1973).

A partir de estos dos sistemas de pesos, se desarrolló el TF-IDF (Salton y Yang, 1973), el cual trabaja determinando la frecuencia relativa de las palabras en un documento específico comparado con la proporción inversa de esa palabra en todo el corpus (Ramos, 2003). Su fórmula es la siguiente:

$$TF - IDF_{i,j} = f_i^j * log \frac{N}{n_i} \qquad (5)$$

Donde f_i^j es la frecuencia del término i en el documento j; N es el número total de documentos que conforman el corpus y n_i es el número de documentos donde se encuentra el término i.

En el método de TF-IDF existen, a grandes rasgos, cuatro tipos de pesos que son los siguientes:

Peso grande: Este ocurre cuando un término tiene una alta frecuencia de aparición pero no se encuentra en la mayoría de los documentos analizados.
Peso mediano: En este caso, el número de apariciones tanto dentro de un documento como en los archivos que conforman el corpus no es baja ni alta.
Peso bajo: Su aparición sucede cuando la frecuencia del término es baja y la construcción se encuentra en la mayoría de los documentos.
Peso nulo: Acontece cuando la frecuencia de aparición de un término dentro de un documento es nula o cuando el término aparece en cada uno de los documentos que pertenecen al corpus analizado.

En la Tabla 3 se muestra un ejemplo de los pesos calculados por el método de TF-IDF para una serie de términos en tres documentos:

Término	Documento 1			Documento 2			Documento 3		
	TF	IDF	TF-IDF	TF	IDF	TF-IDF	TF	IDF	TF-IDF
la	10	$\log\frac{3}{3}=0$	0	12	$\log\frac{3}{3}=0$	0	9	$\log\frac{3}{3}=0$	0
geometría	5	$\log\frac{3}{2}=0.17$	0.85	7	$\log\frac{3}{2}=0.17$	1.19	0	$\log\frac{3}{2}=0.17$	0
analítica	5	$\log\frac{3}{1}=0.47$	2.35	0	$\log\frac{3}{1}=0.47$	0	0	$\log\frac{3}{1}=0.47$	0
descriptiva	0	$\log\frac{3}{1}=0.47$	0	6	$\log\frac{3}{1}=0.47$	2.82	0	$\log\frac{3}{1}=0.47$	0
casa	0	$\log\frac{3}{1}=0.47$	0	0	$\log\frac{3}{1}=0.47$	0	8	$\log\frac{3}{1}=0.47$	3.76
de	11	$\log\frac{3}{3}=0$	0	8	$\log\frac{3}{3}=0$	0	10	$\log\frac{3}{3}=0$	0
José	0	$\log\frac{3}{2}=0.17$	0	2	$\log\frac{3}{2}=0.17$	0.34	5	$\log\frac{3}{2}=0.17$	0.85

Tabla 3. Ejemplo de la aplicación de TF-IDF

Como se puede observar, el método de TF-IDF permite la discriminación de palabras frecuentes, como lo son las palabras funcionales, asignándoles pesos nulos o bajos. Por tanto, estas no suelen quedar en las listas de palabras clave que se emplean en los sistemas de búsqueda de información. Pero además califica con pesos altos las palabras con alto valor de importancia para cada uno de los documentos analizados.

Hasta la fecha se han realizado diversas variaciones al método de TF-IDF y se han empleado métodos de normalización, los cuales serán expuestos en el siguiente apartado. De igual manera el uso de TF-IDF ha pasado a otras áreas de PLN.

1.2.2 Normalización de la longitud del documento

Una de las modificaciones que han sido agregadas al método de TF-IDF es el empleo de un factor de normalización que permita equilibrar casos en los que se empleen documentos de distintos tamaños. Es decir, un documento muy extenso tiene una ventaja más alta sobre los

documentos más pequeños a la hora de recuperar información, esto debido a que el tamaño del documento es un parámetro que hasta cierto punto afecta el cálculo de los pesos. Esta ventaja se observa cuando el usuario hace una consulta al sistema de recuperación de información, el cual traerá los documentos que contengan los términos de la consulta con mayor peso.

Según Singha et al. (1996) existen dos razones principales por las cuales es necesario emplear la normalización:

Frecuencias de términos más altas: Los documentos grandes usualmente emplean los mismos términos repetidamente. Como resultado, los factores de frecuencia de los términos pueden ser grandes para documentos largos, aumentando la contribución promedio de sus términos a la similitud de la consulta de documentos.

Más términos: Los documentos largos también tienen una gran cantidad de términos diferentes. Esto aumenta el número de coincidencias entre la consulta y un documento largo, aumentando la similitud de la consulta de documentos, y los casos de recuperar documentos largos en vez de documentos cortos.

La normalización, a grandes rasgos, permite tratar de la misma manera a todos los documentos sin importar su longitud.

Existen diversos métodos para llevar a cabo la normalización, algunos de ellos atacan las dos razones principales vistas anteriormente, algunos otros métodos solamente una de ellas.

1.2.2.1 Normalización de coseno

La normalización de coseno es uno de los métodos más empleados para normalizar los pesos de TF-IDF, se basa en el *modelo de espacio vectorial* o *vector space model*. Este modelo es la representación de un conjunto de documentos como vectores en el espacio vectorial común (Manning et al., 2008). Un *vector* es un segmento de línea dirigido[5] con dirección, sentido y

[5] Según Castañeda De Isla Puga (2000) un *segmento de línea dirigido* o *segmento dirigido* es un segmento de recta en el que se ha asignado un punto origen y un punto extremo.

magnitud. Es decir, un documento se puede representar como un vector que sería de la siguiente manera:

$$\overline{D}_i = (w_1, w_2, w_3, \dots, w_{n-2}, w_{n-1}, w_n)$$

Donde D_i es el documento i-ésimo del corpus y w_j es cada uno de los pesos de los términos que conforman el documento, en el modelo de espacio vectorial, los pesos representan cada uno de los componentes escalares[6] de un vector de n dimensiones. De manera gráfica, un documento con tres términos podría representarse como se muestra en la Figura 2.

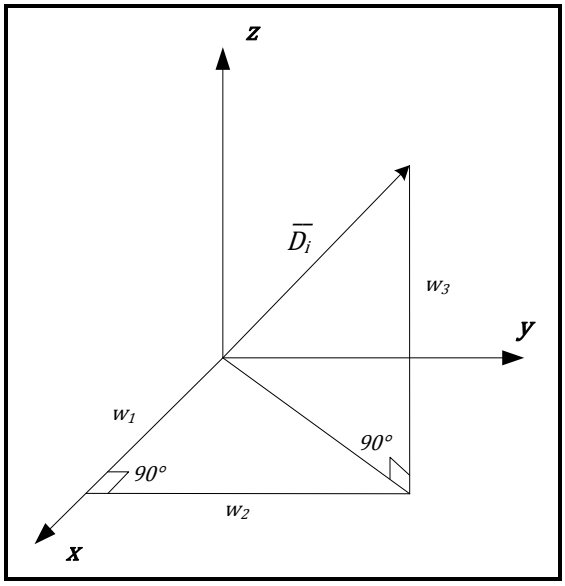

Figura 2. Representación gráfica de un documento de 3 términos en el modelo de espacio vectorial

Al igual que a cualquier otro vector, al vector $\overline{D}$, se le puede sacar su módulo; el *módulo*, *norma*, o *magnitud* de un vector es el tamaño de cualquier segmento dirigido que lo representa (Castañeda De Isla Puga, 2000); este valor es un escalar siempre positivo. Para un vector de la forma $\bar{V} = (a_1, a_2, a_3, \dots, a_{n-1}, a_n)$ el módulo se calcula de la siguiente manera:

[6] Un *escalar* o *cantidad escalar* es un número o medida en que la dirección no interviene o carece de significado (Daintith, 2001). En otras palabras, un escalar sólo indica una magnitud.

$$\left|\overline{V}\right| = \sqrt{a_1^2 + a_2^2 + a_3^2 + \ \ldots + a_{n-1}^2 + a_n^2} = \sqrt{\sum_{i=0}^{n} a_i^2} \qquad (6)$$

Donde $\overline{|V|}$ es el módulo del vector y a_i es cada uno de los componentes escalares que conforman al vector V. Tomando lo anterior en cuenta, para el vector de un documento $\overline{D}$, su módulo sería la raíz cuadrada de la suma de los pesos, es decir:

$$\left|\overline{D}\right| = \sqrt{\sum_{i=0}^{n} w_i^2} \qquad (7)$$

De la misma manera en que se puede obtener el módulo de un vector, también se puede convertir en un vector unitario. Un *vector unitario* es aquel en el que su módulo es igual a la unidad (Castañeda De Isla Puga, 2000). Para ello se emplea la normalización de vectores que para un vector de la forma $\bar{V} = (a_1, a_2, a_3, \ldots, a_{n-1}, a_n)$ consiste en la siguiente:

$$\bar{v} = \frac{\overline{V}}{\left|\overline{V}\right|} \qquad (8)$$

Donde $\overline{v}$ es el vector unitario resultante, $\overline{V}$ es el vector original y $\left|\overline{V}\right|$ es el módulo del vector $\overline{V}$.

A partir de la normalización anterior, se desarrolló la *normalización de coseno* (Salton y Buckley, 1988), el cual consiste en multiplicar el vector del documento por el *factor normalizador*, como se muestra en la siguiente fórmula:

$$\overline{d_i} = \overline{D_i} * \frac{1}{\left|\overline{D_i}\right|} = \frac{\overline{D_i}}{\left|\overline{D_i}\right|} \qquad (9)$$

La normalización de coseno permite eliminar los dos casos por los cuales se normaliza, mismos que fueron vistos en la sección 1.2.2. De igual manera cuando se emplea la normalización de coseno, los pesos de cada uno de los términos tienen una escala que va del cero al uno, debido a que este proceso crea vectores unitarios.

1.2.2.2 Normalización por pivote

Uno de los problemas que ocurren al normalizar es que el factor de normalización penaliza en exceso a los pesos de los documentos grandes, otorgándoles una desventaja a la hora de realizar una búsqueda. Por tanto, es necesario equilibrar el factor de normalización para aumentar la posibilidad de recuperar un documento de cierta longitud.

Para enfrentar este problema, se desarrolló la normalización por pivote (Singha et al., 1996). Este método está conformado a grandes rasgos en dos pasos, el primero de ellos es el cálculo de un factor de normalización por medio de un método de normalización como el de por coseno, y posteriormente el cálculo del nuevo factor de normalización.

El método se basa en la idea de que después de normalizar hay un punto en el cual la probabilidad de recuperar documentos de cierta longitud disminuye cuando su probabilidad de relevancia aumenta. Por ello es necesario rectificar el factor de normalización. En la Figura 3 se puede observar lo explicado antes.

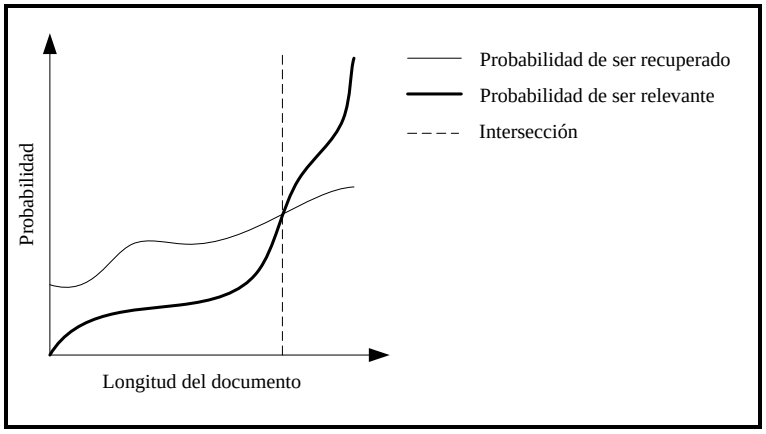

Figura 3. La probabilidad de recuperar un documento normalizado disminuye mientras más relevante sea

La normalización por pivote consiste en rectificar el factor de normalización, disminuyendo su valor en documentos largos, pero aumentándolo en documentos cortos. Para llevar a cabo esta rectificación es necesario considerar lo siguiente:

$$N_1 = n \qquad (10)$$

$$N_2 = m\,(N_1) + b \qquad (11)$$

Donde N_1 es la recta que representa la normalización original (una recta identidad) y N_2es la recta que representa la normalización rectificada; n son los valores de la normalización original, m es la pendiente de la recta y b es la ordenada al origen. En un plano estas dos rectas se verían como se muestra en la Figura 4, el punto donde se intersecan es el pivote (p).

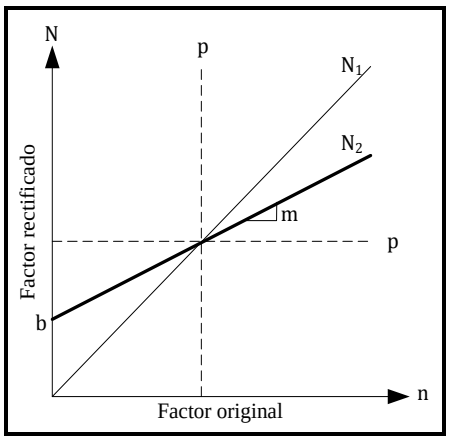

Figura 4. Representación gráfica de las rectas de normalización

La ecuación de la recta normalizada rectificada tiene dos parámetros los cuales se desconocen por defecto. En el caso de la pendiente m, si su valor es igual a 1 entonces se genera una recta paralela a la recta N_1; si su valor es 0 entonces N_2 es una recta horizontal y normalizaría todos los documentos de la misma manera; por tanto, es necesario que el valor de m esté en el intervalo (0,1). Con respecto a b este debe tener un valor que permita a la recta N_2 cruzar por el mismo punto por el cual la recta N_1 cruza el pivote. Por tanto, para obtener el valor de b se sustituye la Ecuación (10) en la Ecuación (11) y queda de la siguiente manera:

$$N_2 = (m * n) + b \qquad (12)$$

Si se considera el punto donde se intersecan las dos rectas de normalización, tiene por coordenadas (p,p), entonces la Ecuación (12) se podría escribir de la siguiente manera para $n = p$:

$$p = (m * p) + b \quad (13)$$

Simplificando la Ecuación (13) y despejando b:

$$b = (1 - m)p \quad (14)$$

Finalmente sustituyendo la Ecuación (14) en la Ecuación (12) se tiene:

$$N_2 = (m * n) + (1 - m)p \quad (15)$$

La ecuación anterior es la que se emplea para obtener el factor de normalización por pivote, el cual se aplica dividiendo cada uno de los pesos de los términos sin normalizar entre el factor de normalización rectificado. El valor de p, es decir, del pivote que se emplea, no tiene una fórmula definida, simplemente los autores del método (Singha et al., 1996) recomiendan emplear como p el promedio de todos los pesos del documento a normalizar. Con respecto a m es un valor que se calcula empíricamente dependiendo del documento y que debe encontrarse entre cero y uno.

1.2.2.3 Normalización por máximo TF

La normalización por la máxima frecuencia de un término (TF) es otro método empleado en los sistemas que emplean TF-IDF. Esta normalización ha sido empleada en sistemas como SMART e INQUERY (Singha et al., 1996).

El método consiste en normalizar la frecuencia de un término por medio de la siguiente fórmula:

$$nTF_{i,j} = \alpha + \left((1 - \alpha) * \left(\frac{TF_{i,j}}{TF_{máx}(j)}\right)\right) \quad (16)$$

Donde α es el valor de suavidad (smoothing), el cual va de 0 a 1, aunque en la práctica se emplea un valor de α= 0.4; $TF_{i,j}$ es la frecuencia del término i en el documento j; mientras que $TF_{máx}(j)$ es la frecuencia máxima de los términos que se encuentran en el documento j. Mientras que $nTF_{i,j}$ es la frecuencia del término normalizada.

Este método limita la frecuencia del término (TF) a un valor máximo de 1, por tanto, sólo compensa la primera razón por la cual se realiza la normalización (frecuencia alta de términos). En la Tabla 4 se muestra un pequeño ejemplo de lo anterior para $\alpha = 0.4$.

TF	nTF
10	$0.4 + \big(0.6 * (10/10)\big) = 0.4 + (0.6 * 1) = 1.00$
8	$0.4 + \big(0.6 * (8/10)\big) = 0.4 + (0.6 * 0.8) = 0.88$
5	$0.4 + \big(0.6 * (5/10)\big) = 0.4 + (0.6 * 0.5) = 0.7$
7	$0.4 + \big(0.6 * (7/10)\big) = 0.4 + (0.6 * 0.7) = 0.82$

Tabla 4. Ejemplo sobre la normalización por máxima frecuencia

1.2.3 Evaluación de sistemas de recuperación de información

En todo sistema, esté relacionado con el PLN o con cualquier otra área, es necesario siempre realizar una evaluación en donde se indique qué tan bueno es el sistema, de esta manera se puede saber cuáles son los puntos fuertes y/o débiles, y en qué se debe mejorar. Pero para llevar a cabo lo anterior es necesario contar con medidas que sean precisas, exactas y reflejen poca subjetividad.

En el caso de la recuperación de información existen dos medidas básicas que se emplean para calificar todo sistema que emplee esta tarea del procesamiento de lenguaje natural. Estas dos medidas son *precisión* o *precision*, por su término en inglés, y *cobertura* o *recall*, aunque de este último término existen otras traducciones al español como, *exhaustividad*, *especifidad* y *recuerdo*. Según Gelbukh y Sidorov (2006) la precisión es la relación entre los resultados correctos sobre los resultados obtenidos en total, mientras que la cobertura es la relación entre los resultados correctos sobre los resultados que deberían haber sido obtenidos. En otras palabras, la precisión indica qué tan preciso fue la recuperación de información, mientras que la cobertura da a conocer si se trajeron todos los resultados que debían ser traídos.

La fórmula para calcular la precisión de un sistema es la siguiente:

$$P = \frac{Número\ de\ elementos\ relevantes\ recuperados}{Número\ de\ elmentos\ recuperados} \qquad (17)$$

Mientras que para el cálculo de la cobertura se emplea la siguiente fórmula:

$$R = \frac{Número\ de\ elementos\ relevantes\ recuperados}{Número\ de\ elementos\ relevantes} \qquad (18)$$

Aunque los desarrolladores de los sistemas de recuperación de información buscan que se cumplan con los mejores niveles de precisión y cobertura, estas dos medidas son hasta cierto punto inversamente proporcionales, es decir, mientras se busque mayor precisión en un sistema, claramente se dejarán pasar los casos que salgan de la norma y por tanto disminuirá la cobertura; en cambio, si se busca que el sistema obtenga todos los casos posibles, es decir, mayor cobertura, la precisión disminuirá porque se deja pasar mayor cantidad de información. En la Figura 5 se muestran dos gráficas, la primera es la curva ideal de la cobertura contra la precisión y la segunda es el comportamiento típico de esta curva en los diversos sistemas relacionados con el PLN.

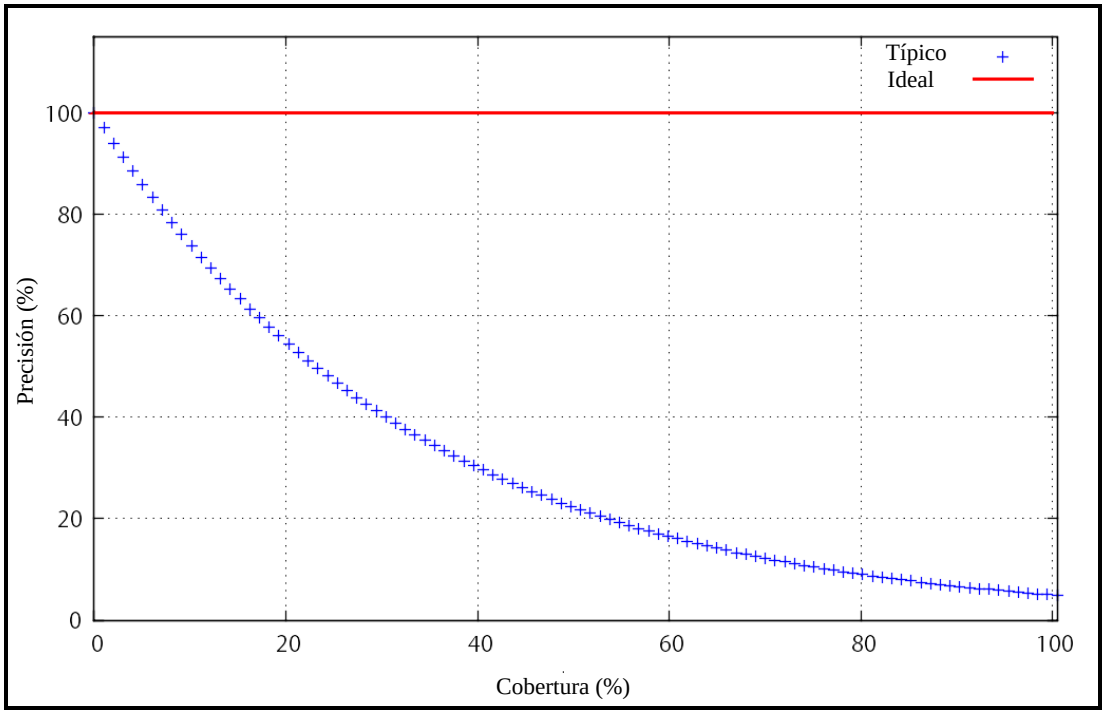

Figura 5. Gráficas típica e ideal del comportamiento de la cobertura contra la precisión

Si bien ambas medidas son las más utilizadas para evaluar los sistemas de recuperación de información, existe otra llamada *F-score* (van Rijsbergen, 1979). Esta medida también conocida como *F-measure* o F_1 es una combinación entre las medidas de precisión y cobertura, su fórmula es la siguiente:

$$F_1 = \frac{(\beta^2 + 1)PR}{\beta^2 P + R} \qquad (19)$$

Donde β es el factor relativo al peso que se le dará a la precisión y a la cobertura, si su valor es igual a 1, ambas medidas reciben el mismo peso, si es mayor a 1 la precisión es favorecida, mientras que si es menor a 1 la cobertura tendrá mayor peso; P es la precisión y R es la cobertura del sistema.

1.2.4 Extracción y recuperación de información

Además de la recuperación de información existe otra tarea dentro de PLN que se llama *extracción de información* o *information extraction (IE)*. La extracción de información es el nombre dado a cualquier proceso que selecciona estructuras y combina datos que son encontrados, de manera explícita o implícita, en uno o más textos (Cowie y Wilks, 2000).

A grandes rasgos la extracción de información se caracteriza por lo siguiente (Ananiadou y McNaught, 2006):

- Toma un texto en lenguaje natural de un documento fuente, y extrae los hechos esenciales acerca de uno o más tipos de hechos predefinidos.
- Representa cada uno de los hechos como una plantilla donde los espacios son llenados con base en lo encontrado en el texto. Según Jurafsky y Martin (2008) sólo una pequeña parte de la información encontrada es relevante para llenar la plantilla; el resto puede ser ignorado.

El propósito de la extracción de información es estructurar el texto posiblemente no estructurado (Pudota et al., 2008). Con esto se pueden obtener resultados que pueden ser otorgados de manera directa al usuario o ser modificados para que puedan ser insertados en una base de datos.

Algunos ejemplos de las tareas de la extracción de información son identificar al hablante en un comunicado, encontrar las proteínas mencionadas en un artículo de una revista de biomedicina y extraer los nombres de las tarjetas de crédito aceptadas por un restaurante desde una reseña en línea (Kauchak et al., 2002).

La recuperación de información y la extracción de información son tareas similares, debido a que traen la información más relevante. De igual manera emplean las mismas métricas para evaluar los sistemas, las cuáles son, precisión, cobertura y F_1. Asimismo, existen métodos que se emplean en ambas tareas, como el TF-IDF que se vio en el apartado 1.2.1. Aunque la recuperación y la extracción de información comparten características existen algunas diferencias: la extracción de información regresa hechos, datos o estructuras de ellos de manera automática, sin basarse en una consulta en concreto, por tanto, permite obtener datos sin hacer un análisis manual de los documentos fuente. En cambio, la recuperación de información busca información con base en una consulta, para ello los documentos fuente se deben encontrar indizados y organizados de tal manera que permitan la búsqueda. Frecuentemente se emplea la extracción de información para alimentar los datos que se emplean en la recuperación de información.