Revista Politécnica
ISSN: 1390-0129
editor.rp@epn.edu.ec
Escuela Politécnica Nacional
Ecuador

Hernández, M.; Gómez, J.
Aplicaciones de Procesamiento de Lenguaje Natural
Revista Politécnica, vol. 32, julio-diciembre, 2013, pp. 87-96
Escuela Politécnica Nacional

Aplicaciones de Procesamiento de Lenguaje Natural

Hernández M. * **Gómez J.** **

* *Escuela Politécnica Nacional, Facultad de Ingeniería en Sistemas*
Quito, Ecuador (e-mail: myriam.hernandez@epn.edu.ec)
** *Universidad de Alicante, Departamento de Lenguajes y Sistemas Informáticos*
Alicante, España (e-mail: jmgomez@ua.es)

Resumen: *El campo de procesamiento de lenguaje natural (PLN), ha tenido un gran crecimiento en los últimos años; sus áreas de investigación incluyen: recuperación y extracción de información, minería de datos, traducción automática, sistemas de búsquedas de respuestas, generación de resúmenes automáticos, análisis de sentimientos, entre otras. En este artículo se presentan conceptos y algunas herramientas con el fin de contribuir al entendimiento del procesamiento de texto con técnicas de PLN, con el propósito de extraer información relevante que pueda ser usada en un gran rango de aplicaciones. Se pueden desarrollar clasificadores automáticos que permitan categorizar documentos y recomendar etiquetas; estos clasificadores deben ser independientes de la plataforma, fácilmente personalizables para poder ser integrados en diferentes proyectos y que sean capaces de aprender a partir de ejemplos. En el presente artículo se introducen estos algoritmos de clasificación, se analizan algunas herramientas de código abierto disponibles actualmente para llevar a cabo estas tareas y se comparan diversas implementaciones utilizando la métrica F en la evaluación de los clasificadores.*

Palabras clave: *Procesamiento de lenguaje natural, clasificadores, categorizar, etiquetar, aprendizaje supervisado, aprendizaje no supervisado, aprendizaje automático.*

Abstract: *The field of natural language processing (NLP) has grown tremendously in recent years, its research interests include: information retrieval and extraction, data mining, machine translation systems, question answering systems, automatic summarization, sentiment analysis, among others. In this paper we present some concepts and tools in order to contribute to the understanding of text processing with NLP techniques, to extract relevant information that can be used in a wide range of applications. Automatic classifiers can be developed to categorize documents and recommend labels, these classifiers should be platform independent, easily customizable in order to be integrated in different projects and to be able to learn from examples. In this article we introduce the algorithms for classification, we discuss some open source tools currently available to perform these tasks and different implementations are compared using F metrics to evaluate classifiers.*

Keywords:*Natural language processing, classify, categorize, tagging, supervised learning, unsupervised learning, machine learning.*

1. INTRODUCCIÓN

El instrumento que los seres humanos utilizamos para comunicar el conocimiento es el lenguaje natural. Actualmente, buena parte del saber humano se encuentra en forma digital en distintos tipos de colecciones de datos. Los volúmenes de información son inmensos, según la International Data Corporation [8], el mundo generó 1,8 zettabytes de información digital en 2011 y "en 2020 el mundo va a generar 50 veces [esa cantidad]" [13]. Las computadoras archivan esta información pero sin el Procesamiento de Lenguaje Natural (PLN) es difícil aprovecharla. El procesamiento del lenguaje involucra una transformación a una representación formal, manipula esta representación y por último, si es necesario, lleva los resultados nuevamente a lenguaje natural. Los campos de desarrollo de PLN incluyen la recuperación y extracción de información, traducción automática, sistemas de búsquedas de respuestas, generación de resúmenes automáticos, minería de datos, análisis de sentimientos, entre otras. Este artículo está organizado de la siguiente manera: se presentan conceptos de los campos de PLN

mencionados, con énfasis en recuperación y extracción de información con el propósito de conocer métodos de extracción de información relevante que pueda ser usada en un gran rango de aplicaciones. Se introducen algoritmos de clasificación, se analizan herramientas de código abierto disponibles actualmente para llevar a cabo estas tareas y se comparan diversas implementaciones utilizando resultados reportados por distintos autores.

2. CONCEPTOS DE PROCESAMIENTO DE LENGUAJE NATURAL

2.1 Recuperación y extracción de información

La recuperación de información (RI), es el proceso de encontrar en un repositorio grande de datos, material (usualmente documentos) de naturaleza no estructurada (usualmente texto) o semiestructurada (páginas Web, por ejemplo), que satisfaga una necesidad de información [22].

Los datos no estructurados no tienen un esquema claro, no están listos para procesar y son lo opuesto a los datos con un esquema estructurados como los que se encuentran en bases de datos. Los datos semiestructurados están en documentos esquema estructurados como los que se encuentran en bases de datos. Los datos semiestructurados están en documentos con marcas explícitas como el código HTML. La información encontrada debe ser pertinente y relevante. La relevancia es la medida de cómo una pregunta se ajusta a un documento y la pertinencia es la medida de cómo un documento se ajusta a una necesidad informativa [15].

Las estrategias de recuperación de información involucran la transformación del texto en representaciones adecuadas de acuerdo a modelos específicos que cumplan con los propósitos de las búsquedas.

Como se muestra en la Figura1, los modelos pueden ubicarse en categorías de acuerdo a dos posibles dimensiones: sus bases matemáticas y sus propiedades [16].

En la dimensión de bases matemáticas, el texto puede ser representado como: conjuntos de palabras o frases en donde las coincidencias se logran realizando operaciones de álgebra booleana; modelos algebraicos que introducen parámetros e índices para recuperar información con metadatos, calificar y clasificar documentos en respuesta a una consulta, lo que lleva a modelos en espacios vectoriales, matriciales o agrupamientos irregulares; modelos probabilísticos que enfocan la solución de los problemas de búsqueda desde el punto de vista probabilístico, aplicando teoremas como el de Bayes; modelos basados en características que se eligen y combinan y califican la relevancia de las similaridades usando métodos de aprendizaje automático.

La dimensión de propiedades tiene que ver con la forma como se presentan las relaciones entre los términos/palabras del documento en el espacio vectorial, relaciones que pueden ser ortogonales e independientes o dependientes [22].

Una vez que se ha logrado acceso a los contenidos buscados y se tiene control sobre los datos, en muchas aplicaciones se hace necesario el siguiente proceso que es el de extracción de la información (EI), que consiste en la obtención de las partes que interesan en el texto para pasarlas a un formato de base de datos, es decir a un formato estructurado.

Los sistemas de EI pueden ser de utilidad aún si no presentan una puntuación perfecta en las medidas de recuperación y precisión. En el caso de colecciones muy grandes, es preferible tener resultados parcialmente correctos antes que realizar la extracción por métodos no automáticos. Las medidas utilizadas en estos sistemas para medir su rendimiento son:

Cobertura = (número de documentos recuperados) / (número total de documentos relevantes)

Precisión = (número de documentos recuperados que son relevantes) / (número total de documentos recuperados)

Los clasificadores se evalúan usando la métrica F, que es igual a la media armónica de la precisión y la cobertura. La forma de calcular las medidas de micro promedio y el macro promedio de la métrica F (micro-F1 y macro-F1), se puede consultar en [24].

En estos casos los resultados deben pasar por un sistema de auditoría que permita corregir manualmente los errores en cuanto a precisión (que es una tarea relativamente fácil puesto que sólo hay que comprobar los resultados recuperados extraídos) y cobertura (tarea más compleja al no poder saber exactamente cuál es la cantidad real de documentos relevantes en una gran colección de textos).

En general los sistemas de EI son útiles si: la información a ser extraída está especificada explícitamente; o el documento puede resumirse con un número pequeño de plantillas; o la información que se necesita está incluida completamente en el texto.

Cada documento se procesa para encontrar entidades y relaciones con significado y contenido. Primero se definen los tipos de información semántica que van a ser extraídos.

La jerarquía entre marcos y características se presentan en forma de árbol, con los marcos como raíces y las características como hijos que se van añadiendo conforme se las va descubriendo.

La salida del motor de EI es un conjunto de marcos anotados es decir etiquetados que son extraídos de los documentos. Los marcos pueblan una tabla en la que los campos del marco son las filas de la tabla [7]. Hay cuatro tipos

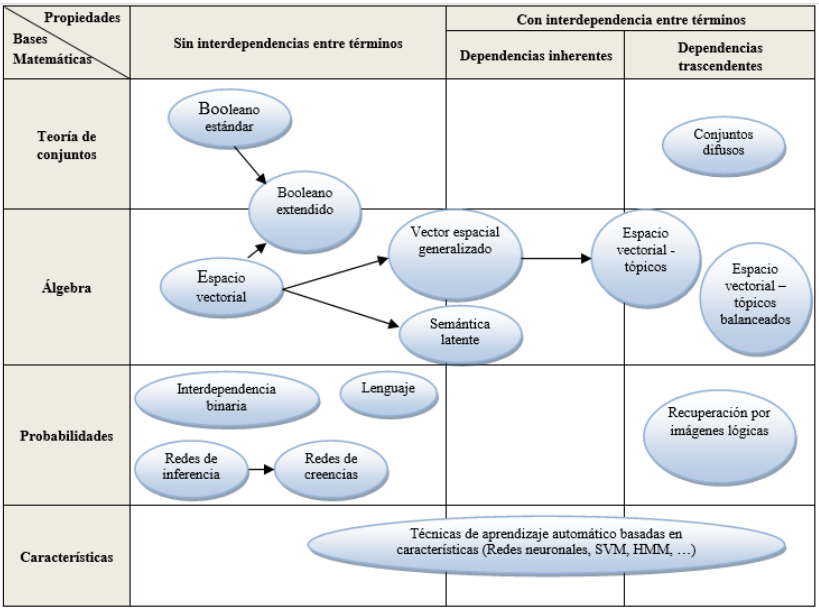

Figura 1. Categorización de modelos de Recuperación de Información. Adaptada a partir de [16]

básicos de elementos que podrían ser extraídos: entidades, atributos (características de las entidades extraídas), hechos (que relacionan entidades) y eventos (actividades u ocurrencias en las que participan las entidades) [7].

En la Figura 2, se muestra un ejemplo de documento etiquetado con características que podrán ser extraídas usando un motor de EI.

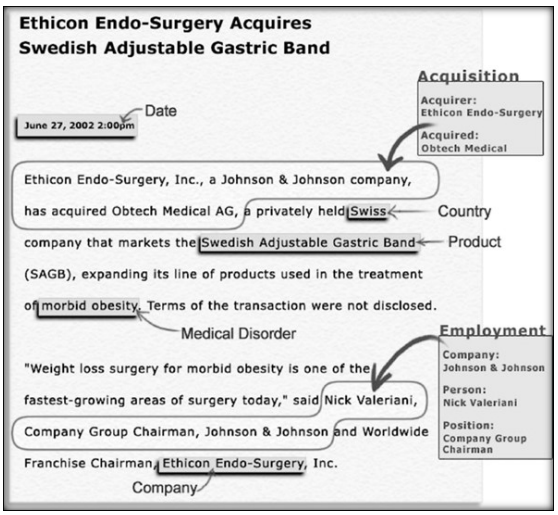

Figura 2: Artículo de noticias etiquetado [7]

En [7] se listan algunas herramientas de código abierto que se utilizan para etiquetado:

- Etiquetador de Eric Brill [5] (C code)
- Etiquetador de Lingua-EN-Tagger [19]
- Etiquetador de Illinois POS T [22]
- Etiquetador Demo [3]
- Etiquetador OpenNLP [1] [24]

[1] `http://opennlp.sourceforge.net/models-1.5/`

2.2 Minería de datos

La minería de datos proporciona herramientas poderosas para descubrir patrones ocultos y relaciones en datos estructurados. Este proceso asume que los datos ya se encuentran almacenados en un formato estructurado. Por esta razón su pre-procesamiento consiste en la limpieza y normalización de los datos y la generación de numerosos enlaces entre las tablas de las bases de datos. La minería de datos usa técnicas y metodologías de RI, EI y corpus procesados con técnicas de lingüística computacional [7].

2.3 Traducción automática

La traducción automática tiene objetivos claros: tomar el texto escrito en un lenguaje y traducirlo a otro, manteniendo el mismo significado. En general el proceso de traducción automática sigue tres pasos: primero, el texto en el lenguaje original se transforma a una representación intermedia, luego, de acuerdo a la morfología del lenguaje destino, se realizan modificaciones a esta representación intermedia y por último ésta se transforma al lenguaje destino. La manera de evaluar si la traducción es correcta es un problema no trivial. Definir exactamente la palabra "significado" y luego poder medirlo presenta complicaciones. ¿Cómo saber que dos expresiones significan lo mismo o algo parecido? Normalmente, se encarga este tipo de tarea a traductores humanos, pero distintas personas realizan distintas traducciones de un mismo texto y diferentes evaluadores de una traducción pueden no coincidir sobre si el texto está bien traducido o no. Con un enfoque estadístico en la evaluación de la traducción automática se puede llegar, con suficientes muestras, a una distribución verdadera con lo que se lograrán evaluaciones válidas. Se detectan reglas de traducción extrayendo traducciones de

palabras en distintos contextos dentro de corpus. De esta manera se establece alineamiento de palabras, un paso fundamental en cualquier modelo estadístico de traducción automática. Un corpus con palabras alineadas permite la estimación de modelos basados en frases y árboles, que son los enfoques más comunes hoy en día [4].

A pesar de los avances en esta tecnología, todavía hay muchos retos en la traducción especialmente entre lenguajes con distinto orden de palabras y una morfología compleja. La investigación en esta área tiene muchas direcciones no exploradas: métodos de aprendizaje avanzado en modelos estadísticos de traducción, modelos sintácticos y sus representaciones, datos paralelos para entrenamiento de modelos estadísticos, integración de traducción del habla con otras aplicaciones como reconocimiento de voz y traducción automática. Afortunadamente, se dispone de muchas herramientas de código abierto que pueden ser usadas por los investigadores en estos campos [13].

2.4 Sistemas de búsquedas de respuestas

Son sistemas diseñados para tomar una pregunta en lenguaje natural y proporcionar una respuesta. De esta manera los usuarios no tendrían que navegar y leer una o varias páginas de resultados de búsqueda. Estos sistemas se construyen sobre motores de búsqueda y requieren contenido como fuente para descubrir las respuestas. Deben tener métodos para entender las preguntas del usuario y determinar el tipo de respuesta que debe dar, generar una búsqueda significativa de la consulta, y finalmente calificar los resultados obtenidos. De estos tres problemas el más difícil de enfrentar es determinar el tipo de respuesta. Para hacerlo se realizan tres pasos: entrenamiento, fragmentación y solo entonces la determinación del tipo de respuesta.

Para construir un sistema de respuestas se necesita aplicar técnicas de PLN como: RI, EI con algoritmos de reconocimiento de entidades y etiquetado, comparación de secuencias de caracteres, entre otras [22].

2.5 Generación de resúmenes automáticos

En [2] se define el problema de la generación de resúmenes automáticos a dos diferentes niveles: a nivel de documento y a nivel de grupos de documentos. Los resúmenes pueden ser con enfoque extractivo o abstractivo. Los métodos extractivos se basan en los mismos principios usados en la identificación de términos, consisten en una colección de términos, frases o párrafos significativos que definen el significado del texto original. Los abstractivos depende de técnicas de parafraseo para producir las síntesis, las técnicas aún están siendo desarrolladas. Un problema común es el de la existencia de múltiples documentos sobre un mismo tema, en este caso se habla de resúmenes a nivel de colecciones de documentos que agrupan o separan los documentos por tópicos y destacan las similitudes y diferencias de la información contenida en ellos.

Los contenidos se relacionan entre ellos en un sentido semántico: cubren el mismo tópico, tienen similares categorías semánticas o conceptos estrechamente relacionados.

2.6 Análisis de sentimientos

De acuerdo a [7] el análisis de sentimientos en textos es la identificación y extracción de información subjetiva. También llamado "minería de opiniones", ese proceso generalmente involucra el uso de herramientas de PLN y software de análisis de textos para automatizar el proceso. La forma básica de análisis de sentimientos es una clasificación polarizada de sentimientos que puede asignar calificaciones de en un rango de -10 a 10 que se basa en el aprendizaje para evaluar emociones tanto negativas como positivas en corpus etiquetados de entrenamiento.

Técnicas avanzadas permiten analizar gramaticalmente y descomponer la oración. La minería de opiniones tiene un mercado ávido de conocer, indexar y resumir opiniones en grandes volúmenes de texto con fines de mercadeo y manejo de imagen.

Los algoritmos heurísticos tienen el inconveniente de la dificultad de recopilar manualmente todos los patrones posibles que expresan sentimientos. Por ello, la siguiente fase de investigación usa la información creciente existente en Internet sobre con comentarios de distinta naturaleza. Se usan reglas gramaticales, tipo compiladores, para extraer inferencias.

El motor de reglas se aplica varias veces para transformar el texto etiquetado en oraciones que definen la asociación entre una palabra y una parte del habla con un sentimiento calificado. Para la implementación se usan herramientas para etiquetar y una base de datos con claves / frases con evaluaciones de polaridad de emociones. Esta información proviene de fuentes etiquetadas o por aprendizaje en corpus. Dos fuentes de datos disponibles son: HDCUS [11] y el WordNet-Affect [6]. En [13] se listan librerías de código abierto que pueden ser usadas para construir un modelo de análisis de sentimientos: Gate [9], Balie [1], Mallet [20].

3. CLASIFICACIÓN, CATEGORIZACIÓN Y ETIQUETADO

Dado un grupo de objetos, la tarea de clasificarlos consiste en asignarlos a un set pre especificado de categorías. Si estamos dentro del dominio de gestión documental, la tarea se la conoce como categorización de texto, y consiste en hallar uno o más tópicos en los que encajen los contenidos de los documentos; teniendo como entrada un grupo de categorías (sujetos – temas) y un conjunto de

documentos de texto. La categorización automática de documentos es una forma de clasificación de patrones, que se es necesaria para la gestión eficiente de sistemas de información de textos. Se aplica en el indexado de texto para entrega comercial personalizada de texto, filtrado de spam, categorización de páginas web bajo catálogos jerárquicos, generación automática de metadatos, detección de género de textos, entre otros [7]. Hay dos enfoque principales a la categorización de textos. Un enfoque de ingeniería del conocimiento en el que mediante reglas de clasificación se introduce conocimiento experto (reglas) y otro el de aprendizaje automático (ML: Machine Learning), en el que procesos inductivos generales construyen un clasificador con aprendizaje basado en ejemplos preclasificados.

Existen diversos resultados en cuanto a rendimiento en el dominio de gestión documental comparando entre ingeniería del conocimiento y sistemas ML, usualmente el primero supera al segundo, pero esta diferencia se va reduciendo debido a que muchas investigaciones, en los últimos tiempo, se concentran en ML. Esto último debido a que el enfoque de ingeniería del conocimiento tiene la desventaja de la dificultad para la creación y mantenimiento de las reglas de codificación del conocimiento mientras ML requiere un conjunto de ejemplos clasificados manualmente que podrían tener un costo menor.

De acuerdo a [7] el enfoque de ingeniería del conocimiento se enfoca en el desarrollo de reglas de clasificación obtenidas en forma no automática. Un experto en el dominio define un conjunto de condiciones suficientes para que un documento sea clasificado en una categoría. El desarrollo de las reglas de clasificación puede ser una labor que toma muchas horas-hombre.

Los sistemas de aprendizaje automático generan etiquetas sobre el contenido en forma automática o semiautomática. Se usan algoritmos para observar como los objetos se etiquetan y se sugieren alternativas para etiquetas existentes o nuevas para contenido no etiquetado.

Los algoritmos de clasificación aprenden con ejemplos usando datos que han sido organizados en clases en forma manual o a través de algún proceso automático. A través del proceso de entrenamiento, los algoritmos de clasificación determinan las propiedades o características que indican que un objeto pertenece a una clase dada. Cuando han sido entrenados, los algoritmos pueden clasificar datos que no tienen todavía etiquetas. En la categorización de documentos se asigna, a un documento, una etiqueta relacionada con una categoría o un tópico.

Un algoritmo de categorización construye un modelo de términos individuales y otras características como longitud o estructura. Al final el modelo puede ser usado para categorizar nuevos documentos. En un sentido computacional, el proceso de clasificación busca asignar etiquetas a datos. Dado un conjunto de características de un objeto, un clasificador intenta asignar una etiqueta a ese objeto. El clasificador hace esto usando el conocimiento derivado de ejemplos de cómo otros objetos han sido etiquetados.

Estos ejemplos, conocidos como datos de entrenamiento, sirven como fuente de conocimiento que el clasificador usa para tomar decisiones sobre objetos no analizados previamente. La categorización trata de asignar una categoría a un objeto. La categorización de documentos es el proceso de categorizar un documento de texto usando alguna(s) características comunes.

En este punto se tratan de categorías basadas en el sujeto, pero otras aplicaciones categorizan documentos usando análisis de sentimientos y tendremos entonces categorías como positividad o negatividad en una revisión de producto, o las emociones ocultas en un mensaje de email o en una solicitud de soporte al cliente. La selección de características determina la calidad y el tipo del clasificador.

3.1 Algoritmos de clasificación

Según [22] los clasificadores binarios indican si un objeto es o no miembro de una clase. A veces se combinan para obtener una clasificación multiclases. Dependiendo del algoritmo, la salida será una sola clase o un número de clases con pesos que describen la probabilidad de que el objeto sea miembro de una clase determinada, como es el caso del algoritmo Mahout Bayes.

A veces los clasificadores jerárquicos están organizados en estructuras tipo árboles. En estos casos un documento que pertenece a la clase A, que tiene como hijos B y C, será evaluado con los clasificadores entrenados para reconocer si está en la clase B o C. Si coincide con B, será evaluado para los hijos de esa clase y así sucesivamente hasta llegar al último nivel del árbol.

Un ejemplo de categorizador multiclases es el de máxima entropía. El categorizador usa las palabras encontradas en documentos como características y los temas como categorías. El proceso de entrenamiento construye un modelo de las relaciones entre las palabras y los temas. En un documento no categorizado el modelo determina los pesos de las categorías y los usa para producir una salida que describe el tema del documento.

En los enfoques que usan el modelo del espacio vectorial, la distancia del vector espacio entre los documentos que han sido clasificados es comparada a un documento que no ha sido clasificado y el resultado se usa para determinar la clasificación apropiada para el documento. El documento no categorizado se convierte en una consulta que se usa para recuperar documentos que son clasificados o documentos que representan los contenidos de cada categoría.

3.2 Proceso de clasificación

De acuerdo a [7] el proceso de clasificación es el mismo para todos los algoritmos. En la Figura 3, se muestra este proceso que consta de las fases de: preparación, entrenamiento, prueba y producción. A menudo este proceso se repite varias veces para ajustar el comportamiento del clasificador y producir los mejores resultados.

Una vez que un clasificador entra en fase de producción, a menudo va a requerir extenderlo para cubrir casos adicionales no cubiertos por los datos de entrenamiento.

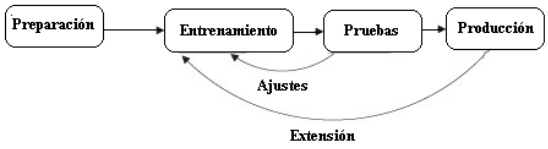

Figura 3: Fases del proceso usado para desarrollar un clasificador automático. Adaptado de [22].

En la fase de preparación se procesan los datos para el proceso de entrenamiento. Se escogen las etiquetas de acuerdo a las características relevantes y se transforma los datos al formato del algoritmo de entrenamiento.

En la fase de entrenamiento cada característica se asocia con la etiqueta asignada al documento y el algoritmo de entrenamiento identifica las características que son importantes para distinguir entre clases y modela las relaciones entre características y etiquetas de clase. En la fase de prueba, el algoritmo evalúa si las clases de los ejemplos corresponden a las asignadas por el clasificador.

Con el número de asignaciones correctas e incorrectas, se calcula la precisión del algoritmo. Algunos clasificadores producen una salida de la fase de entrenamiento que permite conocer cómo están interpretando los datos de entrenamiento para poder ajustar los parámetros. El entrenamiento se puede realizar algunas veces añadiendo o removiendo ejemplos, cambiando la forma como se extraen las características, modificando las clases, o modificando los parámetros del algoritmo. Algunos algoritmos, como el de máxima entropía, repiten el proceso hasta llegar a la mejor respuesta. Cuando el clasificador está listo, se pone en producción, pero puede ser reentrenado después para extender el dominio con nuevas etiquetas o clases.

3.3 Identificación de características

Según [7] la identificación de características es un elemento clave en la precisión de un clasificador automático. Para identificar características el enfoque más simple es el que trata a los documentos como un conjunto de palabras. Cada palabra se considera una característica que se pesa de acuerdo a su frecuencia de ocurrencia. También hay otros esquemas para asignar peso a las palabras con el propósito de escoger características para determinar clases.

Otro tipo de características pueden ser combinaciones de palabras. N-grams se usa para capturar combinaciones frecuentes de palabras que se identifican estadísticamente para eliminar aquellas combinaciones que proporcionan poco valor.

También se pueden usar como características como: metadatos que indican autores, fuentes; longitud para reconocer entre artículos, emails, o tuits.

Se pueden usar fuentes de recursos léxicos para expandir términos y añadir sinónimos de documentos claves como característica.

3.4 La importancia de los datos de entrenamiento

Indica [4] que se requiere un número suficientemente grande de ejemplos para que el clasificador sea capaz de determinar cómo las características se relacionan con las categorías. Éste número de muestras dependerá de la complejidad de la tarea de clasificación como es el número de clases, características, dimensionalidad de las reglas de clasificación, etc. y no puede ser definido a priori [26].

Los datos de entrenamiento se pueden conseguir ya etiquetados en agencias de noticias como Reuters, Freebase y demás proyectos disponibles en Internet. También pueden ser derivados usando procesos automáticos.

4. ALGUNOS ALGORITMOS DE CLASIFICACIÓN

4.1 K. Nearest Neighbor

De acuerdo a [13] el algoritmo de k Nearest Neighbor o clasificador kNN determina el límite de decisión localmente. Por cada 1NN se asigna cada documento a la clase de su vecino más cercano.

Para kNN se asigna cada documento a la clase con los k vecinos más cercanos (menor distancia), donde k es un parámetro. Es un método simple que trabaja bien aún en tareas de clasificación con documentos de múltiples categorías. Su desventaja es que kNN requiere más tiempo clasificando los objetos cuando se tiene un gran número de ejemplos de entrenamiento [29].

Este algoritmo puede implementarse como un método contenido en Waikato Environment for Knowledge Analysis - Weka [29], que es un software de código abierto escrito en Java, desarrollado en la Universidad de Waikato. Weka es un software libre distribuido bajo licencia GNU-GPL.

4.2 Algoritmo de Rocchio

Según [3] este algoritmo los límites entre las clases son hiperplanos en una representación multidimensional del

espacio vectorial de las clases. Por ejemplo, si se tienen tres clases, China, Inglaterra y Kenia.

En el espacio vectorial los límites entre regiones se habrán definido durante el entrenamiento. Se desea clasificar un nuevo documento dibujado como una estrella en la Figura 4, el algoritmo clasificará el documento como China, porque se encuentra en la región correspondiente. Este algoritmo puede utilizar centroides definidos como centros de gravedad de los planos o representaciones vectoriales con diferentes pesos.

El algoritmo es fácil de implementar, eficiente computacionalmente, de rápido aprendizaje y tiene un mecanismo de realimentación de relevancia. El algoritmo de Rocchio falla a menudo en la clasificación de clases multimodales y relaciones. [22].

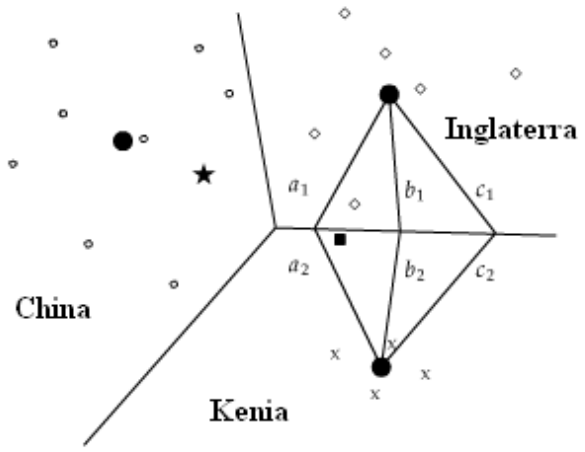

Figura 4: Clasificación de Rocchio. Adaptado de [3].

Rocchio podría implementarse con Weka [29], como un algoritmo propio desarrollado a partir de las clases y esquemas que se encuentran disponibles en esa herramienta.

4.3 Árboles de decisión

Es un clasificador simbólico. Es un árbol con nodos internos etiquetados como términos, ramas etiquetadas con los pesos que tienen en el documento de texto y hojas etiquetadas con las categorías. Cada nodo en el árbol se asocia con un conjunto de casos. Los árboles contienen decisiones binarias.

En la Figura 5 se muestra un ejemplo que corresponde a una regla de decisión para construcción (enfoque de ingeniería del conocimiento) [22].

Este algoritmo puede implementarse como un método contenido en WEKA [28].

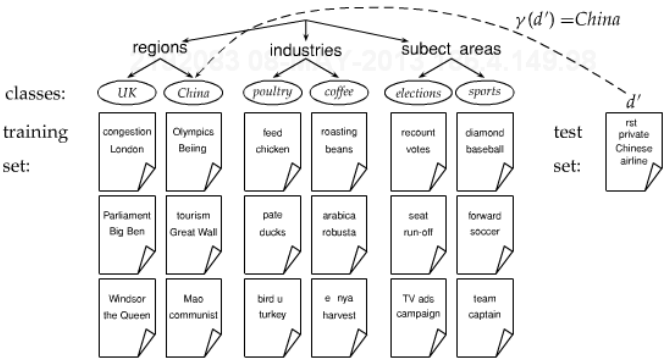

Figura 5: Árboles de decisión [22]

4.4 Algoritmo Iterativo de Naïve Bayes

Según [7], este algoritmo es un clasificador estadístico que se basa en el teorema de Bayes. La idea es encontrar el modelo más probable a partir de documentos etiquetados y no etiquetados.

El algoritmo entrena el modelo sobre documentos etiquetados, luego los siguientes pasos son iteraciones hasta que haya convergencia. En el paso E: los documentos no etiquetados son clasificados en el modelo considerado; en el paso M: el modelo se entrena sobre el corpus combinado. En el paso M, la asignación de categorías de los documentos no etiquetados se asumen como fraccionales de acuerdo a las probabilidades producidas en el paso E.

Este algoritmo puede implementarse como un método contenido en Weka [29].

4.5 Back Propagation Networks

En este método el texto se categoriza por una red neural no lineal alimentada hacia adelante, entrenada con la regla de aprendizaje de Back Propagation. Se aplica a clasificaciones de texto usando aprendizaje supervisado. Es útil para reconocer patrones complejos y realizar funciones no triviales de mapeo [24].

Según [22] las redes neurales pueden construirse para realizar categorización de texto. Los perceptrones multicapa que usan algoritmos de Back Propagation son considerados estándar para procesos de aprendizaje supervisado que permiten soluciones para problemas complejos, el aprendizaje ocurre en el perceptrón evaluando errores y cambiando en consecuencia los pesos de conexión después de que cada dato ha sido procesado.

Como se ve en la Figura 6, usualmente, los nodos de entrada de la red reciben los valores de características, los nodos de salida producen el estado de caracterización, y los pesos de los enlaces representan relaciones de dependencia. Los nodos pueden conectarse en redes con varias capas.

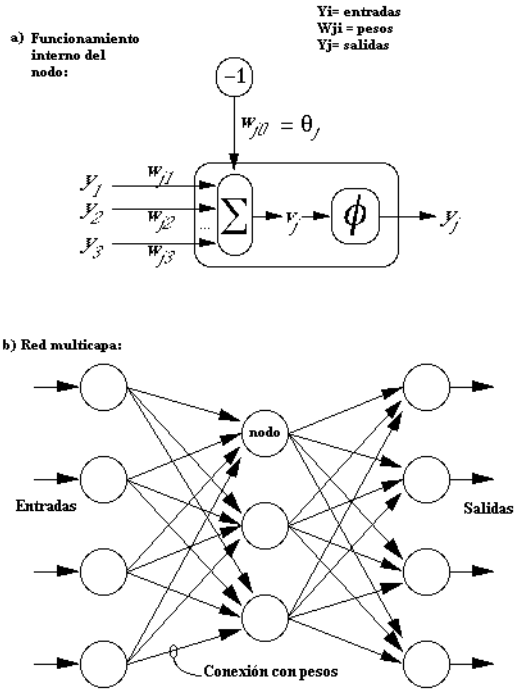

Figura 6: El Perceptrón (a) funcionamiento de un nodo y (b) Red neural multicapa. Adaptado de [29].

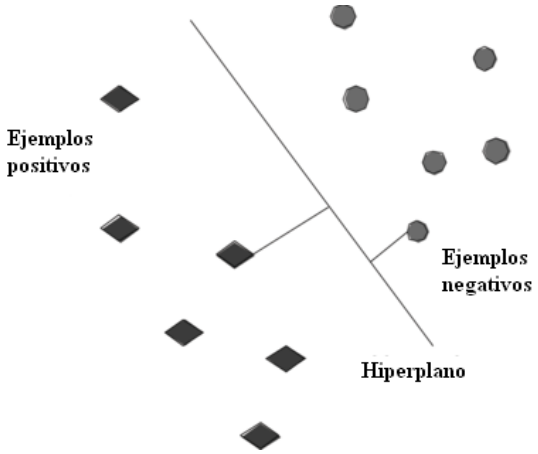

Figura 7: Diagrama de un SVM lineal de dos dimensiones. Adaptado de [7]

Para clasificar un documento, los pesos de las características se cargan con los nodos de entrada; los nodos de activación se propagan hacia adelante a través de la red, y los valores finales de los nodos de salida determinan las decisiones de categorización.

Las redes neurales se entrenan con propagación hacia atrás, donde los documentos de entrenamiento son cargados en los nodos de entrada. Si ocurre un error de clasificación, el error se propaga hacia atrás a través de la red y modifica los pesos de los enlaces para minimizar el error. La clase más simple de red neural es el perceptrón que tiene dos capas (nodos de entrada y salida). Este algoritmo puede implementarse como un método contenido en Weka [29].

4.6 Support Vector Machines (SVM)

De acuerdo con [7] los SVM ejecutan algoritmos de clasificaciones supervisadas en forma rápida y efectiva. En términos geométricos un clasificador SVM puede ser visto como un hiperplano en el espacio multidimensional de características que separa los puntos que representan las instancias positivas de la categoría de los puntos que representan las instancias negativas.

El hiperplano se escoge durante el entrenamiento y tiene un margen máximo que es la distancia desde el hiperplano al punto más cercano de los conjuntos positivo y negativo. Según [29] todas las categorías son linealmente separables.

La idea de SVM es encontrar los separadores. Un ejemplo de separador lineal se ve en la Figura 7. Los SVM pueden mapear los datos de entrada en un espacio multidimensional y utilizar distintos métodos para construir un hiperplano óptimo que los clasifique. En esta construcción se pueden usar, por ejemplo: el método de Least Square, las funciones kernel, que pueden ser polinomiales (Figura 8), radial basis function conocidas como rbf, que tienen un amplio campo de aplicación. La Figura 9 muestra que las funciones kernel de mapeo son muy poderosas porque la clasificación es más fácil en un espacio con mayor número de dimensiones.

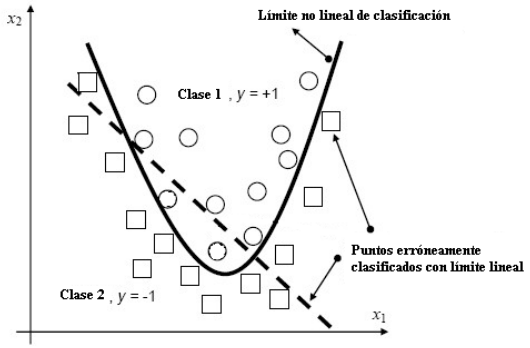

Figura 8: Comparación entre clasificación con límites lineales vs. Polinomiales. Adaptado de [29]

4.7 Comparación de precisión entre distintas implementaciones de clasificadores

En la Tabla 1, se presenta una comparación de diferentes tipos de enfoques en algoritmos de clasificación: vectorial, vectorial ponderada, jerárquica y SVM [28] sobre las colecciones de datos Reuters21578 [2] y 20Newsgroup [3].

El primer corpus contiene 21578 documentos en 135 categorías y el segundo tiene alrededor de 20000 documentos tomados de 20 temáticas noticiosas.

[2] http://kdd.ics.uci.edu/databases/reuters21578/reuters21578.html
[3] http://kdd.ics.uci.edu/databases/20newsgroups/20newsgroups.data.html

Tabla 1. Resultados de la comparación entre diferentes algoritmos de clasificación obtenidos con conjuntos de datos. Adaptado de [28].

Resultados reportados por	Dataset	Esquema de representación	Clasificador usado	Macro F1	Micro F1
Ko et ál, 2004 [14]	20Newsgroup	Representación vectorial con diferentes pesos	Naïve Bayes	83.00	83.00
			Rocchio	78.60	79.10
			K-NN	81.20	81.04
			SVM	86.00	86.10
Tan et ál, 2005 [25]	20Newsgroup	Representación vectorial	Naïve Bayes	83.5	83.5
			Centroid	83.8	84.2
			K-NN	84.6	84.8
			SVM	88.7	88.9
Liang et ál, 2005 [18]	Reuters 21578	Representación vectorial	K-NN	-	79.7
Mubaid and Umair, 2006 [23]	20Newsgroup	Representación vectorial	L Square	83.05	86.45
			SVM	78.19	84.62
	Reuters 21578	Representación vectorial	L Square	94.57	-
			SVM	95.53	-
Hao et ál, 2006 [10]	Reuters 21578	Estructura jerárquica de grafos	SVM (polinomial)	-	86.20
			SVM (rbf)	-	86.50
			K.NN	-	78.8
			Decision Tree	-	87.9
Lan et ál, 2009 [17]	Reuters 21578	SVM con esquemas de ponderación de términos	SVM	90.0	92.1
			K-NN	82.5	84.0
	20Newsgroup	SVM con esquemas de ponderación de términos	SVM	80.8	80.8
			K-NN	69.1	69.1

Se puede observar que el clasificador SVM tiene valores mejores de precisión en clasificación de textos para los dos conjuntos de datos usados en comparación con los otros algoritmos considerados. Estos resultados confirman estudios previos que apuntan en esta misma línea a la hora de clasificar textos [27]. Lamentablemente, los artículos utilizados para obtener los valores de la tabla no aportan información sobre la relevancia estadística de los mismos.

Estos algoritmos pueden implementarse como un método contenido en Weka [29].

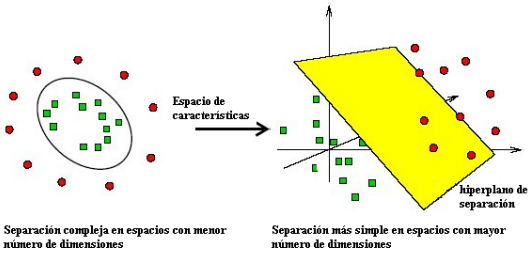

Figura 9: Clasificación más simple en espacios con mayor número de dimensiones. Adaptado de [28]

5. CONCLUSIONES

La Extracción de Información (EI) es la base de los procesos que se realizan con lenguaje natural. Para realizar EI se utilizan distintos modelos para el reconocimiento y etiquetado de entidades que serán comparadas y clasificadas en dos o más clases. La clasificación y categorización de textos son los problemas más investigados en procesamiento de lenguaje natural debido a la creciente cantidad de documentos electrónicos existentes en librerías digitales. Como una medida de comparación del rendimiento global de distintos algoritmos en grupos de datos de entrenamiento ya etiquetados, se recopilan reportes de distintos autores, que aplican clasificadores con representaciones vectorial, jerárquica y SVM en recopilaciones disponibles en Internet: Reuters 21578 y 20Newsgroup. Estas evaluaciones pueden servir como una aproximación inicial para la elección de algoritmos de clasificación para distintos escenarios de implementación de sistemas de EI.

6. AGRADECIMIENTOS

Este trabajo ha sido parcialmente financiado por el proyecto LEGOLANG (TIN2012-31224) y el proyecto TEXTMESS 2.0 (TIN2009-13391-C04- 01) del gobierno español.

REFERENCIAS

[1] Balie, Librerías de código abierto, [Online] Available: `http://balie.sourceforge.net/`

[2] S. Bandyopadhyay, S. Naskar and A. Ekbal, “Emerging applications of natural language processing”, IGI Global, October 31, 2012. [Also Online]. Available: `www.safaribooksonline.com`

[3] Cognitive Computation Group, Etiquetador Demo, [Online] Available: `http://cogcomp.cs.illinois.edu/demo/pos/`

[4] D. Bikel and I. Zitouni, “Multilingual natural language processing applications: from theory to practice”, IBM Press, May 10, 2012. [Also Online]. Available:`www.safaribooksonline.com`

[5] E. Brill, Etiquetador, [Online] Available:http://gposttl.sourceforge.net/ (C code)

[6] Fondazione Bruno Kessler, [Online] Available: http://wndomains.fbk.eu/wnaffect.html y http://www.cse.unt.edu/~rada/affectivetext/

[7] R. Feldman and J. Sanger, "The text mining handbook", Cambridge University Press, December 11, 2006. [Also Online]. Available:www.safaribooksonline.com

[8] J. F. Gantz and D. Reinsel, "Extracting value from chaos." International Data Corporation. 2011. [Online]. Available: http://www.emc.com/collateral/analyst-reports/idc-extracting-value-from-chaos-ar.pdf

[9] Gate, Librerías de código abierto, [Online] Available: http://gate.ac.uk/

[10] P. Y. Hao, J. H. Chaing, and Y. K. Tu, "Hierarchically SVM classification based on support vector clustering method and its application to document categorization", International Journal ExpertSystems with Applications, vol. 33, no. 3, October 2007, pp. 1-5.

[11] Hdcus, Fuente de corpus de sentimientos, [Online] Available: http://hdcus.com/ y http://www.hdcus.com/manuals/wdalman.pdf

[12] Illinois POS T, Etiquetador, [Online] Available: - http://cogcomptest.cs.illinois.edu/page/software_view/3

[13] G. S. Ingersoll, T. S. Morton, and A. L. Farris, "Taming text: how to find, organize, and manipulate It", Manning Publications, December 28, 2012. [Also Online]. Available: www.safaribooksonline.com

[14] Y. J. KO, J. Park, and J. Seo, "Improving text categorization using the importance of sentences", on International Journal Information Processing and Management, vol. 40, no. 1, January 2004, pp. 65-79.

[15] R. Korfhage, "Information storage and retrieval", New York: John Wiley, 2007. [Also Online]. Available:www.safaribooksonline.com

[16] D. Kuropka, "Modelle zur repräsentation natürlichsprachlicher dokumente. ontologie-basiertes information-filtering und -retrieval mit relationalen datenbanken", in Advances in Information Systems and Management Science, Bd. 10, 2004, pp. 110.

[17] C. L. Lan, J. Su, and Y. Lu, "Supervised and traditional term weighting methods for automatic text categorization", IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 31, no 4, April 2009, pp. 721 – 735.

[18] C. Y. Liang, L. Guo, Z. H. Xia, F. G. Nie, X. Li, L. Su, and Z. Y. Yang, "Dictionary-based text categorization of chemical web pages", International Journal Information Processing and Management, vol. 42, no. 4, July 2006, pp.1072 – 1029.

[19] Lingua-EN, Etiquetador, [Online] Available:http://search.cpan.org/~acoburn/Lingua-EN-Tagger/Tagger.pm

[20] Mallet, [Online] Available:http://mallet.cs.umass.edu/

[21] S. Manjunath, B.S. Harish, "Representation and classification of text documents : A brief review" IJCA Special Issue on Recent Trends in Image Processing and Pattern Recognition, TIPPR, 2010, pp. 110-119.

[22] C. D. Manning, Prabhakar Raghavan, and Hinrich Schütze, "Introduction to information retrieval", Cambridge University Press, July 7, 2008. [Also Online]. Available: www.safaribooksonline.com

[23] H. A. Mubaid, and L. Umair 2006, "A new text categorization technique using distributional clustering and learning logic", IEEE Trans. on Knowledge and Data Engineering, vol.18, no..9, September 2006, pp. 1156 – 1165.

[24] A, Özgür, L. Özgür and T. Güngör, "Text Categorization with class-based and corpus-based keyword selection", Computer and Information Sciences - ISCIS 2005. Lecture Notes in Computer Science, vol. 3723, pp 606-615, 2005.

[25] P. Y. Pawar and S. H. Gawande, "A comparative study on different types of approaches to text categorization", International Journal of Machine Learning and Computing vol. 2, no. 4, pp. 423-426, 2012.

[26] S. J. Raudys and A. K. Jain, "Small Sample Size Effects in Statistical Pattern Recognition: Recommendations for Practitioners", in IEEE Transactions on Pattern Anaysis and Machine Intelligence, vol. 13, NO. 3. March 1991.

[27] F. Sebastiani, "Machine learning in automated text categorization", ACM computing surveys (CSUR), vol. 34, n. 1, pp. 1-47, 2002.

[28] S.Tan, X. Cheng, M. Ghanem, B,Wang, . and H. Xu, "A novel refinement approach for text Categorization", CIKM. 2005, pp. 469-476.

[29] I. H. Witten, E. F. Mark and A. Hall, "Data mining: practical machine learning tools and techniques", in The Morgan Kaufmann Series in Data Management Systems, Third Edition, January 20, 2011.