Laboratorio en línea para el procesamiento automático de documentos

Julio C. Torres López, Christian Sánchez-Sánchez, Esaú Villatoro-Tello

Departamento de Tecnologías de la Información,
División de Ciencias de la Comunicación y Diseño,
Universidad Autónoma Metropolitana, Unidad Cuajimalpa, D.F., México

210368282@alumnos.cua.uam.mx, {csanchez, evillatoro}@correo.cua.uam.mx

Resumen. Las grandes cantidades de información textual que actualmente se generan y almacenan digitalmente, junto con la dificultad que existe para analizarla, hace necesario el desarrollo de herramientas que faciliten este trabajo. Existen diferentes campos en las Ciencias de Computación y la Lingüística que en conjunto posibilitan el desarrollo de este tipo de herramientas; en particular una de estas áreas del conocimiento es el Procesamiento de Lenguaje Natural (PLN). El PLN investiga y formula mecanismos computacionalmente efectivos que facilitan la interacción hombre-máquina permitiendo una comunicación mucho más fluida y menos rígida que los lenguajes formales. Sin embargo, para usuarios poco experimentados en este campo, asimilar este tipo de procesos no es algo trivial, situación que desmotiva al uso de las mismas. Con la finalidad de apoyar el desarrollo y la investigación en áreas afines al PLN, en este artículo se presenta un Laboratorio Virtual en Línea para el Procesamiento Automático de Documentos desarrollado en la Universidad, donde se puedan realizar experimentos y ver resultados de forma inmediata, en diferentes tareas relacionadas con el procesamiento automático del lenguaje.

Palabras clave: Preprocesamiento, normalización, etiquetado POS, entidades nombradas, análisis sintáctico, clasificación de textos.

1. Introducción

El avance en la tecnología al día de hoy, así como su bajo costo, ha fomentado que cualquier persona, organismo o empresa pueda almacenar de forma digital grandes cantidades de información textual. El análisis que se pueda realizar a esta información puede ayudar a obtener una mejor perspectiva su contenido y así aportar más y mejores elementos durante los procesos de toma de decisiones.

Dada esta situación, contar con herramientas que ayudan a realizar un procesamiento automático o semi-automático de la información, en particular texto; en grandes volúmenes, ayuda a ahorrar tiempo y recursos.

Existen diferentes campos en las Ciencias de Computación y la Lingüística que en conjunto posibilitan el desarrollo de este tipo de herramientas; en particular una de estos áreas es el Procesamiento de Lenguaje Natural (PLN).

El Procesamiento de Lenguaje Natural o PLN, es una rama de la Inteligencia Artificial, que dentro de sus objetivos tiene el habilitar a las computadoras a procesar y "entender" el texto. El PLN investiga y formula mecanismos computacionalmente efectivos que facilitan la interacción hombre-máquina y permiten una comunicación mucho más fluida y menos rígida que los lenguajes formales, facilitando, teóricamente, la comprensión o análisis de grandes cantidades de información digital. Así entonces, el PLN propone varias técnicas que ayudan a procesar, clasificar y entender (hasta cierto punto) grandes volúmenes de información obtenida [1].

El Procesamiento de Lenguaje Natural representa un área de investigación muy variada. Entre algunos de los temas más representativos se tiene como campos de investigación importantes a: la extracción de información, la generación automática de resúmenes, la búsqueda de respuestas, la recuperación de información monolingüe y multilingüe, técnicas automáticas de clasificación de textos temática y no temática, identificación de perfiles de usuarios, análisis de sentimientos, etc., mostrando grandes avances en todas ellas [2]. Es importante mencionar que a pesar de los avances logrados, el PLN sigue aún en desarrollo, y la necesidad de cada vez más y mejores herramientas motiva el desarrollo de nuevas técnicas y/o enfoques que contribuyan en esta área del conocimiento cada vez más útil en los diferentes sectores tanto sociales como industriales.

En ese sentido, dentro de este trabajo se describe el desarrollo realizado para el laboratorio en línea, especializado en diversas tareas involucradas en el procesamiento automático de documentos, el cual tiene como principal objetivo convertirse en un sitio de referencia y de gran utilidad para estudiantes, desarrolladores e investigadores involucrados en el área de la Lingüística Computacional.

El resto del documento se encuentra organizado de la siguiente manera: en la sección 2 se presenta el marco teórico, más relevante, relacionado a la temática en cuestión y algunas de las herramientas existentes. En la sección 3 se describe el sistema propuesto así como su arquitectura. Finalmente, la sección 4 muestra las conclusiones obtenidas y define las líneas de trabajo futuro.

2. Trabajo relacionado

2.1. Marco teórico

Dentro de esta sección se describen algunas de las principales tareas que se utilizan de manera regular en muchos de los problemas que aborda el área del PLN (Véase Sección 1).

Preprocesamiento: La etapa de preprocesamiento de los documentos consiste fundamentalmente en preparar los documentos para su análisis, eliminando aquellos elementos que se consideran no necesarios para algunas tareas. El preprocesamiento tiene procesos tales como:

a) Eliminación de los términos o partes del documento que no son objeto de indexación, por ejemplo la eliminación de palabras funcionales[1] (preposiciones, artículos, etc.), las etiquetas XML o cabeceras de los documentos.
b) La normalización de textos, consiste en homogeneizar todo el texto de una colección de documentos sobre la que se trabajará, y que afecta, por ejemplo, a la consideración de los términos en mayúscula o minúscula; el control de determinados parámetros como cantidades numéricas o fechas; el control de abreviaturas y acrónimos [3].
c) La lematización es un proceso que busca encontrar la raíz léxica de las palabras, en este sentido las múltiples derivaciones y/o inflexiones de una palabra son llevadas a una sola, el morfema original. Cuando un proceso de lematización no es posible de realizar, se recurre a un proceso de truncado, cuya finalidad es aproximar lo más posible las palabras a su raíz léxica.

En general, el preprocesamiento de los documentos tiene como objetivo principal facilitar el manejo de los documentos en etapas posteriores, dado que ayuda a reducir el costo requerido tanto en espacio como en tiempo de cómputo al momento de procesarlos.

Etiquetado POS: El etiquetado POS[2], también conocido como etiquetado de parte de la oración es el proceso de asignar (o etiquetar) a cada una de las palabras de un texto su categoría gramatical (sustantivos, verbos, artículos, etc.). Este proceso se puede realizar de acuerdo con la definición de la palabra o considerando el contexto en que aparece dentro de algún texto.

El etiquetado gramatical muestra, hasta cierto punto, la estructura de un documento, brinda una gran cantidad de información sobre una palabra y sus vecinas. Una etiqueta gramatical puede ofrecer información relacionada con la pronunciación, el reconocimiento de sustantivos, adjetivos, tipo de derivación y/o inflexión.

Dependiendo del tipo de problema que se quiera resolver, el contar con etiquetas POS dentro de un documento puede resultar de gran utilidad.

Identificadores de Entidades Nombradas (NER): Permite identificar en un texto personas, organizaciones, lugares o fechas.

El NER puede ser utilizado por sistemas que necesiten conocer de quiénes se está hablando y en qué momento para extraer información de los textos que procesan. Es muy útil en sistemas cuya funcionalidad consiste en filtrar información no deseada, o en sistemas que requieren llevar información no estructurada a un formato estructurado, así por ejemplo, construir una base de datos a partir de información identificada por un reconocedor NER [12]. Otro ejemplo puede ser aquel que filtra contenido para adultos, con la finalidad de que dicha información no sea mostrada en las aulas de una escuela de educación básica.

[1] A las palabras funcionales también se les conoce como palabras cerradas o *stopwords*.
[2] Por sus siglas en inglés, *Part-Of-Speech*

Analizador Sintáctico o *Parse tree*: Es una forma ordenada de descomponer una oración según su forma gramatical y estructura. Encontrando las relaciones entra cada una de las palabras que conforman la oración. Teniendo en cuenta que una gramática enlista los principios bajo los cuales se agrupan las palabras, es el conjunto de reglas que describe que es válido en un lenguaje.

Clasificación de Textos (CT): El objetivo de la clasificación de textos es colocar, de forma automática, documentos dentro de un número fijo de categorías (temas o clases) predefinidas, en función de su contenido[13].

En su forma más simple, el problema de clasificación de textos puede formularse de la siguiente manera: Dados un conjunto de documentos de entrenamiento $\mathcal{D}_{Tr} = \{d_1, \ldots, d_n\}$ y un conjunto de categorías predefinidas $\mathcal{C} = \{c_1, \ldots, c_m\}$, el objetivo de la CT es formular la función de aprendizaje que sea capaz de generar el modelo de clasificación adecuado (*i.e.,* una hipótesis) $h : \mathcal{D} \rightarrow \mathcal{C}$ la cual será capaz de clasificar de manera correcta todos aquellos documentos no vistos durante el entrenamiento que pertenecen a $\mathcal{D}$.

Es importante mencionar que en la primera versión de este proyecto, *i.e.,* el Laboratorio en Línea para el Procesamiento de Automático de Documentos, se han considerado, desarrollado e implementado varias herramientas que impactan de manera directa en la resolución de las tareas mencionadas en esta sección.

2.2. Herramientas existentes

En la actualidad existen muchas herramientas de apoyo a tareas relacionadas con el PLN, sin embargo, en la mayoría de los casos estas herramientas cuentan con interfaces gráficas complicadas de entender para no expertos en el tema, o simplemente no se cuentan con ningún tipo de interfaces. Por otro lado, normalmente requieren de realizar un proceso de instalación de diferentes componentes, los cuales en muchos casos, generan gran variedad de dificultades para poder utilizar las herramientas debido a las múltiples incompatibilidades que hay entre versiones de compiladores y/o incluso de sistemas operativos. Agregado a esto, muchas de estas herramientas carecen de manuales de usuario comprensibles para usuarios nuevos en la temática; así mismo, no permiten la integración de más herramientas. Todo esto resulta en un problema que desmotiva su uso y provoca que en la mayoría de los casos, tanto estudiantes como investigadores opten por desarrollar desde cero diferentes módulos y/o herramientas que les permitan realizar sus experimentos.

A continuación se muestra un análisis y comparación de algunas herramientas utilizadas en el PLN[3].

Weka. Es una extensa colección de algoritmos de aprendizaje desarrollados por la universidad de Waikato (Nueva Zelanda) implementada en Java. Son útiles para ser aplicados sobre datos mediante las interfaces que ofrece el sistema o para embeberlos dentro de cualquier aplicación. Además Weka contiene las

[3] Las herramientas analizadas se seleccionaron debido a su popularidad entre varios grupos de investigación.

herramientas necesarias para realizar aprendizaje sobre los datos, tareas de clasificación, agrupamiento y visualización. Weka está diseñado como una herramienta orientada a la extensibilidad por lo que añadir nuevas funcionalidades es teóricamente posible.

La licencia de Weka es GPL, lo que significa que este programa es de libre distribución y difusión. Además, ya que Weka está programado en Java, es independiente de la arquitectura, pues funciona en cualquier plataforma sobre la que haya una máquina virtual Java disponible. Sin embargo, y pese a todas las cualidades que Weka posee, tiene un gran defecto, y éste es la escasa documentación orientada al usuario poco experimentado y que junto a una usabilidad bastante pobre, lo convierte en una herramienta difícil de comprender y manejar si no se cuenta con información adicional. Weka es una herramienta muy especializada, y requiere que los usuarios tengan conocimientos amplios sobre técnicas de aprendizaje automático para darle un uso apropiado y por consecuencia poder explotarla correctamente [4][5].

Natural Language Toolkit (NLTK). NLTK proporciona un conjunto de bibliotecas de procesamiento de textos para la clasificación, simbolización, derivado, etiquetado, análisis sintáctico y semántico-razonamiento. NLTK es adecuado para los lingüistas, ingenieros, estudiantes, educadores, investigadores y usuarios de la industria por igual. NLTK está disponible para Windows, Mac OS X y Linux. NLTK es una librería de código abierto, está desarrollado en Python [6]. Es una programa muy poderoso para el PLN ya que contiene un gran conjunto de herramientas, sin embargo, si la persona que lo va a utilizar no tiene conocimientos previos en Phyton, esto representará un gran esfuerzo para poder utilizarlo de manera eficaz, el tiempo que tomará aprendiendo todas sus funcionalidades podría manejarse como una desventaja, además de que todo se realiza desde línea de comandos. .

Text to Matrix Generator (TMG). Text to Matrix Generator (TMG) es un toolbox que se puede utilizar para diversas tareas en la minería de texto. La mayor parte de TMG (versión 6.0; Dic. '11) está escrito en MATLAB, aunque una gran parte de la fase de indexación de la versión actual del TMG está escrito en Perl. TMG es especialmente adecuado para aplicaciones de minería de datos. Originalmente construido como una herramienta de procesamiento previo para la creación de matrices término-documento (TDM) de texto no estructurado, la nueva versión de TMG (Diciembre '11) ofrece una amplia gama de herramientas para las siguientes tareas: indexación, recuperación, reducción de dimensionalidad, agrupamiento y clasificación. El proceso de indexación puede incluir varios pasos, como la eliminación de las palabras cerradas, como artículos y conjunciones, la eliminación de términos muy frecuentes o infrecuentes. TMG acepta como archivos de entrada, archivos en texto ASCII y muchos archivos PostScript y PDF. TMG permite como opción, una variedad de sistemas de expresión de ponderación y normalización. Una de las mayores desventajas de este toolbox es que a pesar de ser de uso libre, MATLAB no lo es [7].

Stanford CoreNLP. Proporciona un conjunto de herramientas de PLN, para documentos de texto los cuales pueden estar en diferentes idiomas. A partir de

texto plano, se pueden ejecutar todas las herramientas programando las líneas de código correspondientes. Stanford CoreNLP contiene las herramientas de PLN: etiquetador de categorías gramaticales (POS), identificación de entidades nombradas (NER), el analizador, y el sistema de resolución de la correferencia, y proporciona los archivos de modelos para el análisis de inglés y algunos otros idiomas. El objetivo de este conjunto de herramientas de Stanford es permitir a usuarios la obtención, de forma rápida y sin problemas de anotaciones lingüísticas completas de textos en lenguaje natural. Está diseñado para ser altamente flexible y extensible y con la opción de que se pueden seleccionar las herramientas que deben estar habilitadas o inhabilitadas. El código CoreNLP Stanford está escrito en Java y bajo la Licencia Pública General de GNU (v2 o posterior) [8]. Una desventaja grande de este conjunto de herramientas es también que si no existen conocimientos previos en programación en lenguaje Java, será difícil su utilización. Otra desventaja es que no existe una visualización gráfica de los resultados de la misma y todo debe manejarse desde línea de comandos.

Estos conjuntos de herramientas son en su mayoría de uso gratuito, algunas ya son ampliamente conocidas y otras más están en proceso de crecimiento. Algunas funcionan con diferentes idiomas, pero el idioma predominante es el inglés. En importante recalcar la variedad de lenguajes de programación empleados (*e.g.*, JAVA, C, Python, etc.) hace difícil la interacción entre todas ellas. Finalmente, es conveniente mencionar que existen una gran diversidad de herramientas de análisis que son ajenas a estos grandes sistemas, pero que pueden ser de utilidad. Ejemplos de estas son herramientas de análisis en redes sociales como Twitter o Facebook [9].

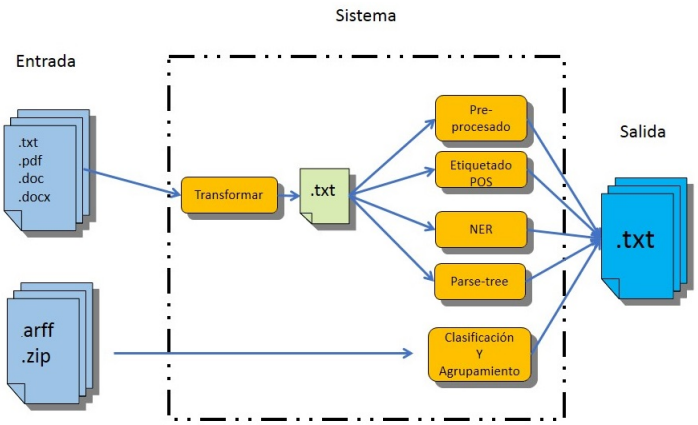

Fig. 1. Arquitectura del sistema propuesto.

3. Sistema propuesto

El sistema propuesto para este proyecto cuenta con características que buscan facilitar el uso de sus herramientas del PLN . Teniendo en cuenta que el laboratorio fue pensado para ser utilizado por estudiantes con poca o ninguna experiencia en el procesamiento de documentos su diseño se simplificó lo más posible. El laboratorio cuenta con diferentes módulos los cuales integran una serie de herramientas para las tareas más utilizadas en el PLN.

Como se muestra en la Figura 1 el sistema tiene una arquitectura modular. El sistema acepta documentos de entrada en diferentes formatos. Estos documentos al ingresarlos al sistema son trasformados a formato de texto plano (.txt). Este archivo podrá utilizar los módulos de preprocesado, etiquetado POS, Identificados de entidades nombradas (NER) y *parse tree*. También acepta archivos de tipo ARFF (.arff), formato exclusivos para poder utilizar la herramienta WEKA, a estos no se le realiza ninguna transformación y se pueden utilizar directamente en tareas de clasificación. También es posible agregar archivos comprimidos ZIP (.zip) que cuenten con una estructura de árbol donde existe una carpeta raíz que almacena más carpetas y estos a su vez otros documentos. Con los archivos comprimidos el sistema puede construir archivos arff, utilizando los nombres de las carpetas como el atributo clase y sus contenidos como sus atributos y/o características.

3.1. Descripción de los módulos

En esta sección realizaremos una descripción de cada uno de los módulos que utilizamos en nuestro sistema.

a) El módulo de transformación de documentos se realiza la modificación de los archivos originales los cuales pueden ser de tipo *.txt, .pdf, .doc* y *.docx*, este módulo los transforma a formato *.txt*, esto se realiza para poder tener un manejo más simple de los archivos. Los archivos *.txt* son guardados en una carpeta que tendrá el nombre del usuario, los archivos con los formatos originales no son guardados por nuestro sistema, para ahorrar espacio en nuestro servidor (Figura 2).

b) El módulo de preprocesado de documentos se realizan las diferentes tareas para preparar los documentos de una manera más rápida y sencilla. En este módulo se utilizaron algunas herramientas de los diferentes programas descritos previamente (Véase Sección 2) y se desarrollaron algunos más (eliminar números, puntos, acentos). Solo basta con seleccionar un archivo que ya previamente se ha cargado al sistema y se elige que preprocesamiento se va a realizar (Figura 3).

Los diferentes preprocesados que realiza el módulo son los siguientes:

- Palabras cerradas. Elimina en el documento las palabras que no tienen relevancia semántica en nuestros documentos como pueden ser preposiciones, determinantes, pronombres y conjunciones del idioma inglés.

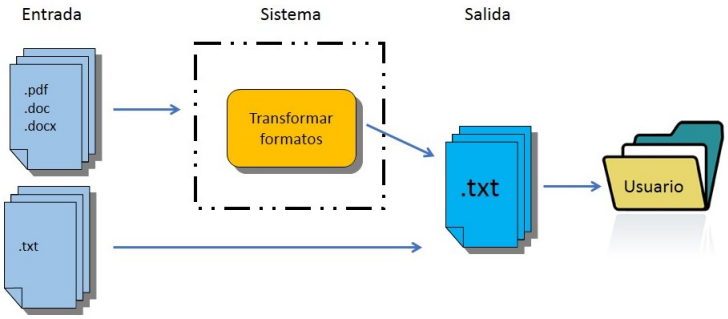

Fig. 2. Módulo de transformación.

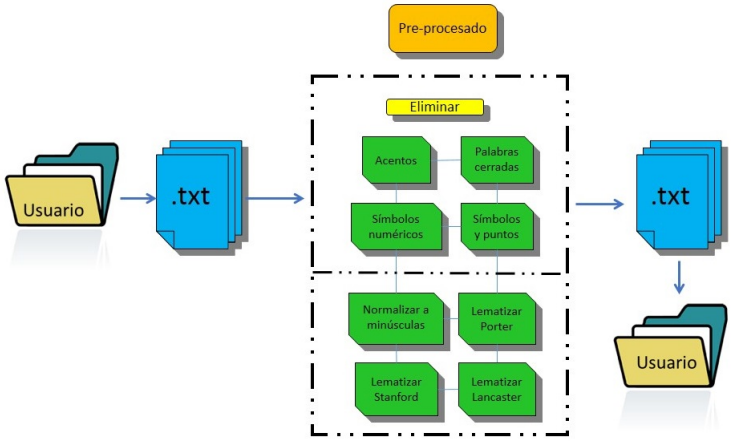

Fig. 3. Módulo de preprocesado.

- Símbolos y puntos. Elimina del documento los símbolos de puntuación y símbolos como: ¡¿ % $, etc.
- Acentos. Elimina en el document los diferentes símbolos de acentuación como tilde, diéresis, etc.
- Números. Elimina en el document todos los símbolos numéricos.
- Normalizar. Hace que nuestro documento sea transformado todo a letra minúscula.
- Lematizador. Transforman cada palabra de nuestro texto a su lema utilizando la herramienta que proporciona NLTK con el algoritmo de Porter y Lancaster [6] respectivamente. También se puede utilizar el Lematizador desarrollado por el Grupo de Procesamiento de Lenguaje Natural de Stanford [8].

La salida de este módulo es un archivo *.txt* con el resultado del preprocesado seleccionado, agregando el nombre del proceso al nombre original del archivo

tratado. Se pueden aplicar las diferentes tareas a un mismo archivo en el orden que se prefiera. El archivo resultante se guarda en la carpeta del usuario, es importante mencionar que el archivo original continúa existiendo sin sufrir ningún cambio.

c) El módulo de etiquetado POS, realiza un etiquetado al archivo original agregando su categoría gramatical (sustantivos, verbos, artículos, etc.) a cada palabra de nuestros archivos. En este módulo está compuesto por herramientas que realizan el etiquetado a todo tipo de archivos y una más que está especializada en documentos de Twitter, ya que reconoce algunos emoticones y anotaciones características usados en esta red social (Figura 4).

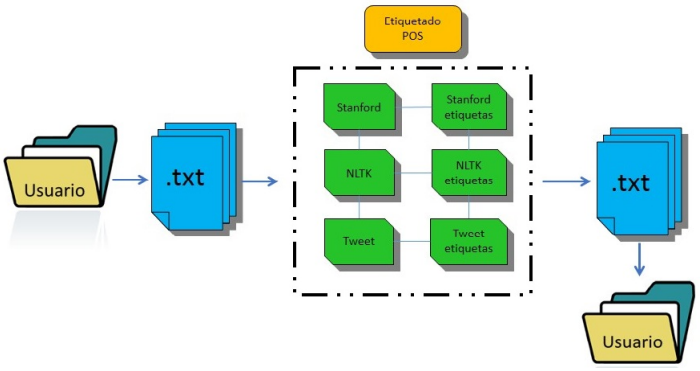

Fig. 4. Módulo de Etiquetado POS.

Las herramientas de etiquetado POS son las proporcionadas por el Grupo de Procesamiento de Lenguaje Natural de Stanford [8] y NLTK [6], de estas herramientas podemos escoger dos modalidades; la primera es que la etiqueta (categoría gramatical) aparezca al lado de la palabra asociada y la segunda es que solo aparezcan las etiquetas. La herramienta especializada en Twitter fue desarrollada por el grupo del Instituto de Tecnologías del Lenguaje, Escuela de Ciencias de la Computación de la Universidad Carnegie Mellon [9].

d) El módulo de reconocimiento de entidades nombradas (NER), reconoce diferentes entidades las cuales pueden ser nombres propios, organizaciones, fechas y lugares. En este módulo se utilizan dos herramientas diferentes, una proporcionada por el Grupo de Procesamiento de Lenguaje Natural de Stanford [8] y la otra de LingPipe [10]. Con la herramienta de Stanford se puede visualizar como salida las entidades junto con los términos o ver solamente las entidades. Con la herramienta de LingPipe solo se visualizan las etiquetas correspondientes al texto (Figura 5).

Este módulo funciona como los anteriores donde se puede seleccionar un archivo precargado en el sistema y aplicarle los diferentes procesos, en el cual se le anexará al final del nombre del archivo el proceso que se le realizó.

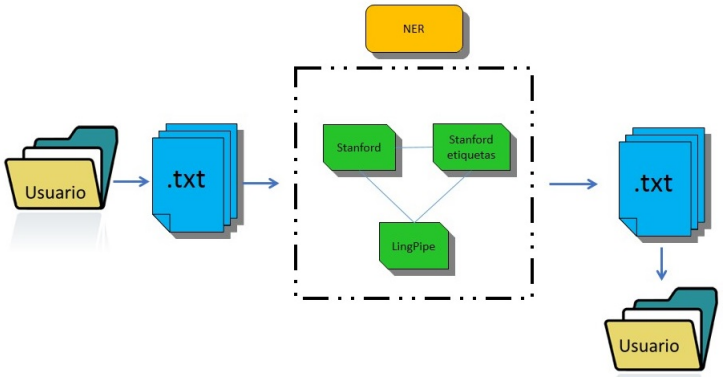

Fig. 5. Módulo de Reconocimiento de Entidades Nombradas.

e) El módulo de *Parse Tree*, descompone una oración según sus funciones gramaticales y nos genera un archivo con éstas por cada oración. Como algunas veces es difícil de leer este tipo de archivo, por ejemplo cuando el texto es muy grande. Para facilitar la visualización se integrá la herramienta llamada DEPENDENSEE [11] que a través de una imagen nos permite ver de una forma gráfica el árbol sintáctico generado. Guarda en una imagen (.png) de cada oración que se tenga en el texto, estas imágenes se guardan en las carpetas personales, y al finalizar la sesión son eliminadas (Figura 6).

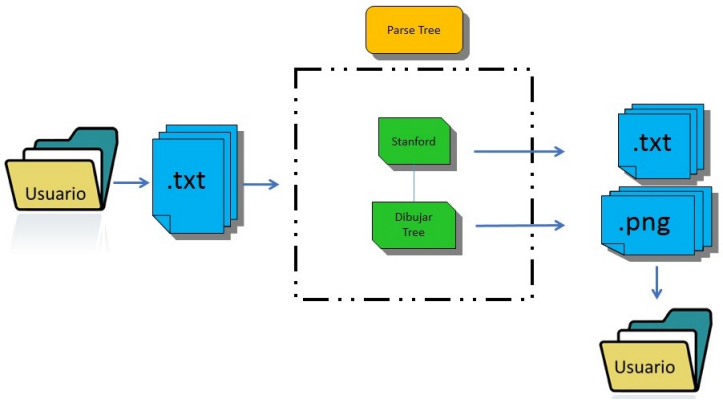

Fig. 6. Módulo de *Parse Tree*.

f) El módulo de clasificación, emplea principalmente módulos de la herramienta WEKA. Debido a esto, es necesario que los documentos a clasificar estén en un archivo *.arff*. Sin embargo, el resultado de este módulo es en un archivo de texto que muestra la matriz de confusión, el porcentaje de casos clasificados

correctamente, el porcentaje de precisión (*precision*), recuerdo (*recall*) y medida F (*f-measure*). De momento solo se pueden utilizar dos algoritmos de clasificación que son los J48 y Naïve Bayes (Figura 7).

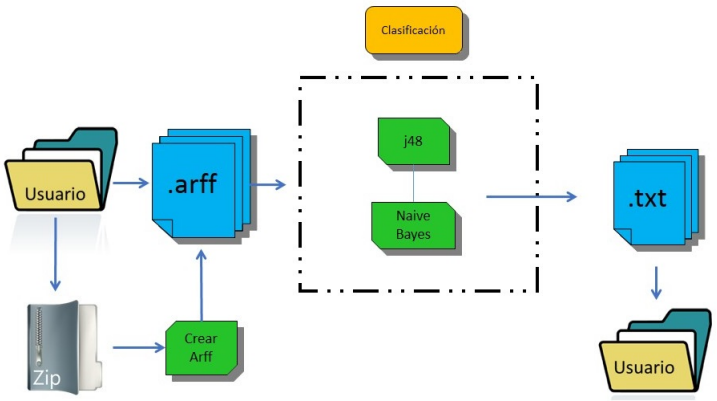

Fig. 7. Módulo de Clasificación. En la versión actual del Laboratorio en línea sólo se han incorporado dos clasificadores: Árboles de decisión y Naïve Bayes, ambos ampliamente utilizados en la comunidad científica.

De forma alternativa, en este módulo el usuario puede crear un archivo *.arff* a partir de un archivo comprimido (.zip). Lo único que necesita es que el archivo *.zip* contenga una carpeta (raíz), que a su vez contenga sub-carpetas que representen las clases de los documentos que estén contenidos en ellas. Por ejemplo, en la Figura 8, se tienen las clases $a, \ldots, n$ donde la clase a tiene los documentos $d_1, ..., d_m$ y así sucesivamente.

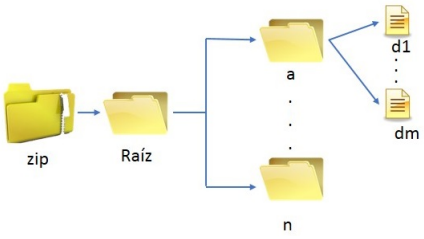

Fig. 8. Estructura del archivo zip.

Cuando se elige esta modalidad, el sistema calcula de manera interna la matriz término-documento (*i.e.,* se realiza el proceso de indexado), permitiendo llevar los documentos contenidos en las diferentes carpetas, a su representación

vectorial, siendo las instancias los documentos contenidos en las diferentes carpetas y los atributos el vocabulario de la colección. Así entonces, bajo este esquema, el *indexado* de los documentos de entrenamiento (Tr), denota la actividad de hacer el mapeo de un documento d_j en una forma compacta de su contenido. La representación más comúnmente utilizada para representar cada documento es un vector con términos ponderados como entradas, concepto tomado del modelo de espacio vectorial usado en recuperación de información [14]. Es decir, un texto d_j es representado como el vector $\overrightarrow{d}_j = \langle w_{kj}, \ldots, w_{|\tau|j} \rangle$, donde τ es el *diccionario*, *i.e.*, el conjunto de términos que ocurren al menos una vez en algún documento de Tr, mientras que w_{kj} representa la importancia del término t_k dentro del contenido del documento d_j.

En ocasiones τ es el resultado de filtrar las palabras del vocabulario, *i.e.,* resultado de un preprocesamiento. Una vez que hemos hecho los filtrados necesarios, el diccionario τ puede definirse de acuerdo a diferentes criterios, sin embargo el que se empleó en este sistema corresponde a la Bolsa de Palabras (BOW).

Cuando esta modalidad es elegida, el usuario tiene la posibilidad de elegir el esquema de pesado que quiere utilizar en la representación vectorial, es decir la importancia w_{kj} de cada término, los cuales son:

- *Ponderado Booleano:* Consiste en asignar el peso de 1 si la palabra ocurre en el documento y 0 en otro caso.

$$w_{kj} = \begin{cases} 1, \text{ si } t_k \in d_j \\ 0, \text{ en otro caso} \end{cases} \tag{1}$$

- *Ponderado por frecuencia de término (TF):* En este caso el valor asignado es el número de veces que el término t_k ocurre en el documento d_j.

$$w_{kj} = f_{kj} \tag{2}$$

- *Ponderado por frecuencia relativa (TF-IDF):* Este tipo de ponderado es una variación del tipo anterior y se calcula de la siguiente forma:

$$w_{kj} = TF(t_k) \times IDF(t_k) \tag{3}$$

  donde $TF(t_k) = f_{kj}$, es decir, la frecuencia del término t_k en el documento d_j. IDF es conocido como la "frecuencia inversa" del término t_k dentro del documento d_j. El valor de IDF es una manera de medir la "rareza" del término t_k. Para calcular el valor de IDF se utiliza la siguiente fórmula:

$$IDF(t_k) = \log \frac{|D|}{\{d_j \in D : t_k \in d_j\}} \tag{4}$$

  donde D es la colección de documentos que está siendo indexada.

Una vez que se ha indexado la colección, el archivo *.arff* es construido de manera automática y el proceso de clasificación se puede realizar de forma tradicional.

Es importante mencionar que cada función incluida dentro del laboratorio posee un botón de ayuda que especifica su funcionamiento y los enlaces necesarios a documentación más extensa sobre las diferentes herramientas utilizadas en este sistema. El laboratorio en línea para el procesamiento automático de documentos estará disponible en el sitio del Grupo de Lenguaje y Razonamiento de la UAM-C cuya página es: http://lyr.cua.uam.mx

4. Conclusiones y trabajo futuro

El Laboratorio en Línea para Procesamiento Automático de Documentos pretende ser una herramienta útil para realizar las diferentes tareas más comunes en el PLN. Hasta el momento, el sistema ha mostrado un funcionamiento estable con documentos de tamaño mediano. Por ejemplo, el preprocesamientos de documentos se realiza de una forma muy eficaz y rápida (segundos) en documentos de tamaño entre 10 000 y 50 000 palabras, funcionando para documentos en inglés o en español.

En el etiquetado POS, NER y *Parse Tree* por el momento solo funciona para documentos escritos en el idioma inglés . El realizar un archivo *.arff* a partir de un *.zip* es prácticamente inmediato (segundos), mientras que el proceso de de clasificación puede ser más tardado dependiendo del tamaño de la matriz proporcionada. En general, gracias al empleo del laboratorio se ha reducido el tiempo empleado para utilizar este tipo de herramientas por usuarios con poca experiencia, ya que se ha minimizado el uso de programación para realizar experimentos. También permite que usuarios sin experiencia puedan utilizar este tipo de herramientas y a través de la visualización de los resultados se ha comprobado que los usuarios que muestran mayor interés por las tareas comunes en el PLN.

Es importante mencionar que el laboratorio se desarrolló con la intención de que la comunidad de PLN en México pueda integrar más herramientas o mejorar las ya existentes. Idealmente el crecimiento del laboratorio propuesto deberá ser orientado a incluir y/o desarrollar herramientas específicas para el idioma español, lo cual permitirá impactar de manera directa en la comunidad de PLN tanto en México como a nivel internacional entre los países de habla hispana.

Agradecimientos. Agradecemos el apoyo otorgado por la Universidad Autónoma Metropolitana Unidad Cuajimalpa, al SNI-Conacyt y al Conacyt por el apoyo otorgado con el proyecto número CB2010/153315.

Referencias

1. Asociación Mexicana para el Procesamiento del Lenguaje Natural, *http://www.ampln.org* (Última visita en Julio de 2013)

2. Vallez, M., Pedraza-Jimenez, R.: El Procesamiento del Lenguaje Natural en la Recuperación de Información Textual y áreas afines [en linea]. "Hipertext.net", núm. 5, 2007. ¡http://www.hipertext.net¿
3. Manning, C.D., Schutze, H.: Foundations of statistical natural language processing. Massachusetts Institute of Technology. Second printing with corrections, 2000
4. Weka Documentation. The University of Waikato. *http://www.cs.waikato.ac.nz/ml/weka/*
5. Witten, I., Frank, E., Hall, M. A.: Data Mining: Practical Machine Learning Tools and Techniques. Morgan Kaufmann Publishers (2011)
6. Bird, S., Klein, E., Loper, E.: Natural Languaje Processing with Python. O'Reilly Media (2009)
7. Zeimpekis, D., Gallopoulos, E.: TMG: A MATLAB Toolbox for Generating Term-Document Matrices from Text Collections. (2005)
8. The Stanford Natural Language Processing Group *http://nlp.stanford.edu/index.shtml*(Última visita en Noviembre de 2013)
9. Twitter NLP and Part-of-Speech Tagging. University, Carnegie Mellon *http://www.ark.cs.cmu.edu/TweetNLP/* (Última visita en Noviembre de 2013)
10. LingPipe *http://alias-i.com/lingpipe/index.html* (Última visita en Noviembre de 2013)
11. Chaoticity *http://chaoticity.com/dependensee-a-dependency-parse-visualisation-tool/* (Última visita en Noviembre de 2013)
12. Téllez Valero, A., Montes y Gómez, M., Villaseñor Pineda, L.: Using Machine Learning for Extracting Information from Natural Disaster News Reports. Computación y Sistemas, 13(1) (2009)
13. Sebastiani, F. Machine Learning in Automated Text Categorization. En *ACM Computing Surveys*, Vol. 34, No. 1, March 2002, pp. 1–47 (2002)
14. Baeza-Yates, R., Ribeiro-Neto, B.: Modern Information Retrival, Addison Wesley (1999)