Minería de textos, lógica y ontologías

Research in Computing Science

Series Editorial Board

Research in Computing Science es una publicación trimestral, de circulación internacional, editada por el Centro de Investigación en Computación del IPN, para dar a conocer los avances de investigación científica y desarrollo tecnológico de la comunidad científica internacional. **Volumen 95,** mayo 2015. Tiraje: 500 ejemplares. *Certificado de Reserva de Derechos al Uso Exclusivo del Título* No. : 04-2005-121611550100-102, expedido por el Instituto Nacional de Derecho de Autor. *Certificado de Licitud de Título* No. 12897, *Certificado de licitud de Contenido* No. 10470, expedidos por la Comisión Calificadora de Publicaciones y Revistas Ilustradas. El contenido de los artículos es responsabilidad exclusiva de sus respectivos autores. Impreso en la Ciudad de México, en los Talleres Gráficos del IPN – Dirección de Publicaciones, Tres Guerras 27, Centro Histórico, México, D.F. Distribuida por el Centro de Investigación en Computación, Av. Juan de Dios Bátiz S/N, Esq. Av. Miguel Othón de Mendizábal, Col. Nueva Industrial Vallejo, C.P. 07738, México, D.F. Tel. 57 29 60 00, ext. 56571.

Editor responsable: *Grigori Sidorov*, *RFC SIGR651028L69*

Research in Computing Science is published by the Center for Computing Research of IPN. **Volume 95**, May 2015. Printing 500. The authors are responsible for the contents of their articles. Printed in Mexico City, in the IPN Graphic Workshop – Publication Office.

Volume 95

Minería de textos, lógica y ontologías

Sabino Miranda Jiménez (ed.)

Instituto Politécnico Nacional, Centro de Investigación en Computación
México 2015

ISSN: 1870-4069

Instituto Politécnico Nacional (IPN)
Centro de Investigación en Computación (CIC)
Av. Juan de Dios Bátiz s/n esq. M. Othón de Mendizábal
Unidad Profesional "Adolfo López Mateos", Zacatenco
07738, México D.F., México

http://www.rcs.cic.ipn.mx
http://www.ipn.mx
http://www.cic.ipn.mx

Indexed in LATINDEX and Periodica / Indexada en LATINDEX y Periódica

Printing: 500 / Tiraje: 500

Printed in Mexico / Impreso en México

Editorial

El propósito de este volumen es reflejar las nuevas direcciones de investigación y aplicaciones de los métodos de la Inteligencia Artificial.

Los artículos de este volumen fueron seleccionados con base en un estricto proceso de revisión efectuada por los miembros del Comité de revisión, tomando en cuenta la originalidad, aportación y calidad técnica de los mismos. Cada artículo fue revisado por lo menos por dos miembros del Comité de revisión del volumen.

Este volumen contiene 13 artículos relacionados con varios aspectos del desarrollo de los métodos de Inteligencia Artificial y ejemplos de sus aplicaciones a varias tareas tales como:

- clasificación de la opinión pública generada en Twitter,
- detección de subjetividad en noticias en línea publicadas en español,
- sistema de reconocimiento multilenguaje del habla,
- técnicas de agrupamiento en la clasificación de estilos de aprendizaje,
- poblado de ontologías de perfiles académicos a partir de textos en español,
- metodologías para análisis político utilizando Web Scraping,
- clasificación semántica de textos, entre otras.

Este volumen puede ser interesante para los investigadores y estudiantes de las ciencias de la computación, especialmente en áreas relacionadas con la inteligencia artificial y su aplicación a los diferentes ámbitos de la vida cotidiana; así como, para el público en general interesado en estos fascinantes temas.

En este número especial de la revista RCS, a nombre de los catedráticos CONACYT y de la comunidad académica del Centro de Investigación e Innovación en Tecnologías de la Información y Comunicación (INFOTEC) expresamos nuestro agradecimiento al Dr. Sergio Carrera Riva Palacio, Director Ejecutivo, y Dr. Juan Carlos Téllez Mosqueda, Director Adjunto de Innovación y Conocimiento, por apoyar de manera ingente la investigación y el desarrollo de la ciencia y la tecnología, sustentado todo ello en la responsabilidad y el compromiso social.

El proceso de revisión y selección de artículos se llevó a cabo usando el sistema libremente disponible EasyChair, www.EasyChair.org.

Sabino Miranda Jiménez
Mayo 2015

Table of Contents

Page

Una herramienta visual para la búsqueda semántica RDF

Joanna Alvarado-Uribe[1], Miguel González-Mendoza[1], Neil Hernández-Gress[1], Carlos Eli Escobar-Ruiz[2] y Marcos Uriel Hernández-Camacho[2]

[1]Tecnológico de Monterrey, Campus Estado de México, México

[2]Universidad Politécnica de Chiapas, Chiapas, México

joanna.1890@gmail.com;{mgonza,ngress}@itesm.mx;carlosescobar@portaltuxtla.com;uriel.hdzc@gmail.com
http://www.itesm.mx
http://www.upchiapas.edu.mx

Resumen. La cantidad de información que uno o más usuarios de Internet generan para la Web Semántica está incrementando diariamente. Por esto, es necesario desarrollar herramientas que nos permitan mostrar esta información de una manera rápida, simple y fácil de entender. De acuerdo con esta premisa, hemos desarrollado una herramienta de visualización de datos semánticos, denominada DBPedia Search, capaz de: 1) consultar cualquier base de datos de tripletas que cuente con un *endpoint* de SPARQL y; 2) generar gráficos, mapas de calor y mapas de geolocalización de manera automática, con base en la información obtenida de la búsqueda realizada por el usuario. El objetivo principal es realizar una búsqueda y un análisis simplificados de los datos semánticos y presentarlos gráficamente.

Palabras clave: DBPedia search, visualización, *Endpoint* de SPARQL, tripletas.

1. Introducción

La Web Semántica es percibida como un área de investigación multidisciplinaria que combina campos científicos como la Inteligencia Artificial, Ciencias de la Información, Teoría de Algoritmo y de la Complejidad, Teoría de Base de datos, Redes de Computadoras, entre otros [1].

La Web Semántica se basa en la idea de agregar más semántica legible por la computadora a la información web a través de anotaciones escritas en *Resource Description Framework* (RDF) [2]. El modelo RDF se introdujo en 1999 como una recomendación del *World Wide Web Consortium* (W3C). Debido a esto,

la propuesta de la Web Semántica es la construcción de una infraestructura de semántica legible por la computadora para los datos en la Web [2].

Con base en la evolución del RDF, se están implementando en la red iniciativas mundiales tales como el *Open Directory Project*, *Dublin Core*, *Friend Of a Friend* (FOAF), *Simple Knowledge Organization System* (SKOS), *vCard Ontology*, y *Really Simple Syndication* (RSS) [2]. Este hecho es crucial para el desarrollo de la Web Semántica, porque RDF sigue los principios de diseño del W3C y algunas de las características principales de la Web Semántica como la interoperabilidad, extensibilidad, evolución y descentralización. Uno de los objetivos principales por el que el modelo RDF fue diseñado, es permitir que cualquier persona pueda hacer declaraciones sobre cualquier recurso. De esta manera, para la construcción de un modelo RDF únicamente es necesario disponer de un conjunto de recursos, básicamente cualquier cosa que tenga un *Universal Resource Identifier* (URI) [2]. Algunos ejemplos de recursos son: páginas web, imágenes, videos, computadoras, impresoras, etc. [3].

El lenguaje para representar los recursos está constituido por un conjunto de propiedades. Las descripciones de estas propiedades son enunciados estructurados en forma de tripletas sujeto-predicado-objeto o sujeto-propiedad-valor [2][4]. Mientras que el predicado y el objeto son recursos o cadenas, el sujeto y el objeto pueden ser objetos anónimos - también conocidos como *blank nodes* - [2]. Otra forma de explicar los componentes de las tripletas es: el sujeto es el recurso, el predicado es la característica que se describe y el objeto es el valor para esa característica [4]. Un aspecto interesante del modelo RDF es que el sujeto u objeto de una sentencia RDF puede ser otra declaración, esta característica es conocida como *reification* [2].

Gráficamente, el modelo RDF puede ser representado como un grafo de datos, Figura 1. La Figura 2 presenta parte de su codificación en RDF/XML [5].

Para trabajar con esta herramienta es necesario disponer de un *endpoint* de SPARQL de la base de datos de tripletas que se desea consultar. Un *endpoint* de SPARQL permite el procesamiento de consultas remotas [6].

En nuestra primera implementación, elegimos el *endpoint* de la versión en inglés de la ontología DBpedia [7]. La versión en inglés de la ontología DBpedia es parte del proyecto de DBpedia; este proyecto ha estado extrayendo información estructurada de Wikipedia en varios idiomas, como el inglés, el español, el japonés, entre otros; con la finalidad de generar información semántica disponible en la Web [7][8].

Este artículo está dividido en seis secciones. En la primera sección denominada Introducción, presentamos brevemente los temas que vamos a abordar en este documento. En la segunda sección llamada Trabajo relacionado, presentaremos algunas herramientas que trabajan con información semántica y/o *endpoints*

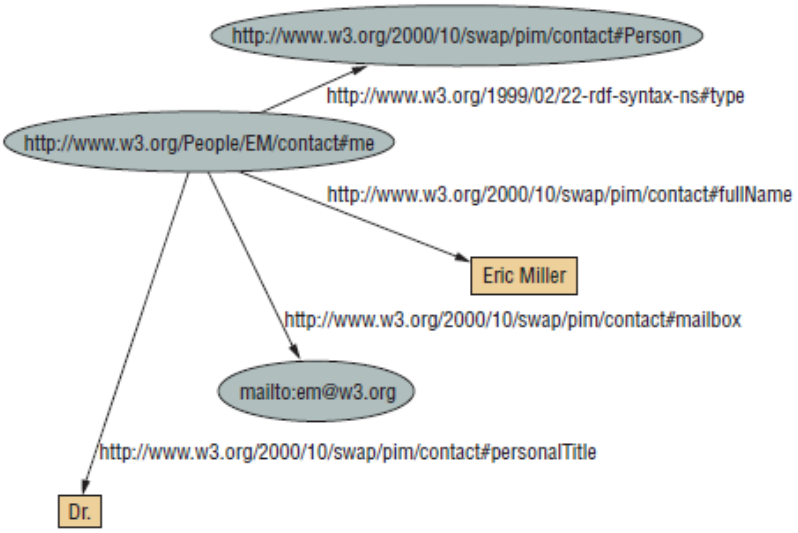


Fig. 1. Grafo de datos RDF, tomado de [5].

de SPARQL, y se mostrará una clasificación de las mismas de acuerdo con el motor de búsqueda que manejan. En la sección 3) Desarrollo y prototipo de DBPedia Search, presentaremos las fases de desarrollo de esta herramienta y el prototipo final; dentro de los aspectos que se abordarán están: tecnología utilizada, recopilación de datos, análisis de datos, entre otros. Para la sección 4) Experimentos y resultados, mostraremos el uso de la herramienta en diferentes Sistemas Operativos (S.O.) y explicaremos brevemente los resultados obtenidos en las pruebas de rendimiento. En la sección 5) Comparativo con otras herramientas, realizaremos un comparativo técnico y, de desarrollo y funcionamiento entre las herramientas presentadas en la Sección 2 y la nuestra. Y la última parte son 6) Conclusiones y trabajo futuro, en esta sección presentaremos nuestro punto de vista sobre la herramienta y describiremos algunas propuestas para mejorarla.

2. Trabajo relacionado

En esta sección introducimos como estado del arte, herramientas que también han abordado búsquedas semánticas. Para ello, nos centraremos en tres enfoques de búsqueda de las numerosas que hay, debido al impacto que tienen hacia nuestra herramienta, estos son:

1. Motores de búsqueda basados en forma. Estos motores se basan en formas complejas que toman ventaja de tener la información organizada en portales semánticos.

```
<?xml version=``1.0"?>
<rdf:RDF xmlns:rdf=``http://www.w3.org/1999/02/22-rdf-syntax-
    ns#"
xmlns:contact=``http://www.w3.org/2000/10/swap/pim/contact#">

 <contact:Person rdf:about=``http://www.w3.org/People/EM/
     contact#me">
   <contact:fullName>Eric Miller</contact:fullName>
   <contact:mailbox rdf:resource=``mailto:em@w3.org"/>
   <contact:personalTitle>Dr.</contact:personalTitle>
 </contact:Person>

</rdf:RDF>
```

Fig. 2. Parte de RDF en RDF/XML, tomado de [5].

2. Motores de búsqueda basados en palabra clave (como el descrito en este artículo). Estos motores se basan en una palabra o término dado, dando como resultado la visualización de información semántica.
3. Herramientas de pregunta/respuesta que utilizan datos semánticos. Estas herramientas permiten al usuario realizar una pregunta con el fin de extraer términos más específicos que les permitan buscar una respuesta directa en lugar de numerosos resultados.

La herramienta de búsqueda SHOE [9] muestra una serie de controles complejos en una forma. Esta forma permite al usuario construir consultas semánticas que se llevan a cabo a través de diferentes fuentes de información. Esta herramienta es un claro ejemplo del primer grupo, los motores de búsqueda basados en forma. La desventaja de este tipo de herramientas, es que el usuario necesita entender cómo trabajan las relaciones semánticas, para así poder construir una buena consulta desde la forma. La herramienta de búsqueda SHOE es impulsada por el lenguaje SHOE (*Simple HTML Ontology Extensions*), una alternativa a los estándares modernos como RDF y OML (*Ontology Markup Language*). Existen algunos ejemplos comerciales de esta categoría como Yummly [10], motor de búsqueda en la Web para comida, cocina y recetas; basado en datos de la Web Semántica.

TAP [10], construido sobre la interfaz de consulta GetData [11], permite al usuario buscar datos semánticos utilizando palabras clave. SemSearch [12] también introduce una interfaz en la que se teclean algunas palabras para realizar una búsqueda; esta herramienta fue construida para los usuarios denominados 'usuarios ingenuos', usuarios que no necesariamente conocen cómo está organizada la Web Semántica. Es importante destacar que este tipo de búsqueda (basada en palabras clave) es la que se realiza en la herramienta presentada en este artículo.

Evi [13], uno de los pocos productos comerciales basados en la Web Semántica, es la clara representación de una herramienta de pregunta/respuesta basada en datos semánticos. Fue desarrollado en Cambridge y se presenta como una aplicación móvil; utiliza el Procesamiento de Lenguaje Natural (PLN) y técnicas de búsquedas semánticas. AquaLog [14] es un ejemplo no comercial de esta categoría, es una solución portátil que puede ser adaptada a cualquier ontología dada, también utiliza tecnologías de PLN para formular tripletas ontológicas.

3. Desarrollo y prototipo de DBPedia Search

De acuerdo con la investigación realizada sobre la Web Semántica y el uso de la semántica con SPARQL, desarrollamos una herramienta cuyo objetivo principal es analizar los datos almacenados en las bases de datos de tripletas con el fin de realizar la clasificación de estos datos en categorías y de esta manera, construir gráficas de barras, mapas de calor y mapas de geolocalización, como resultado de la consulta realizada por el usuario. Las etapas de desarrollo y la construcción del prototipo se explicarán en 5 fases: tecnología utilizada, recopilación de datos, análisis de datos, visualización de las estadísticas y el prototipo.

Tecnología utilizada

Decidimos desarrollar la herramienta con PHP 5, por ser una tecnología de fácil instalación y porque es una tecnología en la que tenemos la experiencia suficiente para desarrollar aplicaciones para la Web. Se utilizó la librería ARC2 para conectar los *endpoints* con la herramienta, y la librería D3.js con JavaScript para permitir la visualización de los datos al usuario final.
A continuación, describiremos las librerías utilizadas:

- ARC2 es una librería de PHP 5.3 que funciona únicamente con triples semánticas (RDF), y un *endpoint* público utilizando SPARQL. También proporciona un almacenamiento en tripletas basado en MySQL con soporte para SPARQL [15].
- D3.js es una librería JavaScript que ayuda a manipular documentos basados en datos utilizando HTML, SVG y CSS. D3 combina componentes de visualización de gran alcance y un enfoque basado en datos para la manipulación DOM [16].

Recopilación de datos

Este es el primer paso para el desarrollo de la herramienta. En esta fase obtenemos la información que necesitamos para llevar a cabo el análisis (esta información será mencionada en las siguientes fases), como los tipos de datos. Los tipos de datos expresan la información contenida en las tripletas; por ejemplo, los tipos de datos en DBpedia son todas las categorías. Esta fase se realiza una única vez para cada *endpoint* de SPARQL.
Un aspecto relevante es que los datos están actualizados en todo momento, ya

que la herramienta trabaja directamente con el *endpoint* de SPARQL.

Análisis de datos

En este paso se analizan los datos obtenidos de la consulta realizada por el usuario, con el fin de encontrar una manera adecuada para mostrar la información resultante. Para cada consulta se mostrarán, en el mejor de los casos, cuatro elementos: una lista con los datos resultantes; a través de la información relacionada con los países, la herramienta mostrará un mapa de calor y geolocalización; y con la información en común, la herramienta construirá gráficas de barras.
El proceso completo se describe a continuación:

- Paso 1: encuentra el URI correcto.
  Este paso revisa ¿cuál es el URI más utilizado? Esto es útil si manejamos una gran base de datos de tripletas y la base de datos tiene información repetida.
- Paso 2: lista de los datos resultantes.
  En este paso se realiza una consulta sencilla para encontrar algunos ejemplos de datos que constituyen los resultados de la consulta.
- Paso 3: encontrar una propiedad relacionada con algún país, ciudad, estado o lugar.
  La herramienta realiza una búsqueda entre las propiedades para encontrar una o más propiedades que contengan información sobre los países o lugares más específicos. Esto es útil cuando la información recopilada es sobre personas y sus nacionalidades, nombre del país de nacimiento, ubicaciones de empresas, entre otros aspectos.
  En el caso de las ciudades, estados o lugares más específicos, buscamos su latitud y longitud para obtener una ubicación más exacta; y para los países, buscamos por sus nombres.
- Paso 4: conteo de datos para cada país o lugar.
  Una vez que la herramienta ha finalizado el paso 3, esta fase encuentra la mejor propiedad para describir el país y/o lugar, y de acuerdo con esta propiedad se realiza el conteo de los datos en cada país y/o lugar para visualizar el mapa de calor y su geolocalización.
- Paso 5: obtención de propiedades comunes para la palabra buscada.
  En el último paso la herramienta agrupa las propiedades comunes y, hace el conteo de los datos contenidos en estas propiedades para construir las gráficas de barras. Por ejemplo, si estamos buscando gente de México las propiedades comunes podrían ser Nombre, Apellido, Ciudad de nacimiento, Fecha de nacimiento, etc.

Es importante mencionar que las consultas, dentro de la herramienta, se realizan utilizando el lenguaje SPARQL, aunque para el usuario final, este hecho es transparente. Un ejemplo de un query en SPARQL, utilizado por la herramienta para esta fase, se muestra en la Figura 3.

Visualización de las estadísticas

En este paso se realiza un tratamiento de la información, en el que se analizan los tipos de datos en las tripletas para combinar los tipos de datos que tengan

```
SELECT ?res ?property ?value
WHERE {
        ?res pgn:tipo <http://mipagina.com/persona> .
        ?res ?property ?value
      }
ORDER BY ?res
LIMIT 500
```

Fig. 3. Filtrar y mostrar la lista de resultados de la palabra clave "personas".

el mismo nombre (aunque diferente URI).
Una vez que se realiza el tratamiento, por medio de la librería D3.js se visualizan los gráficos en Front-End. La herramienta construye un mapa de calor y geolocalización, y gráficas de barras para la visualización de la información resultante y además, presenta una lista de los resultados dentro de la búsqueda. El Front-End se presentará en la Sección de Experimentos y resultados.

Prototipo
Para explicar esta fase, construimos un diagrama de bloques para mostrar cómo se constituye el Back-End de la herramienta, el diagrama se muestra en la Figura 4.

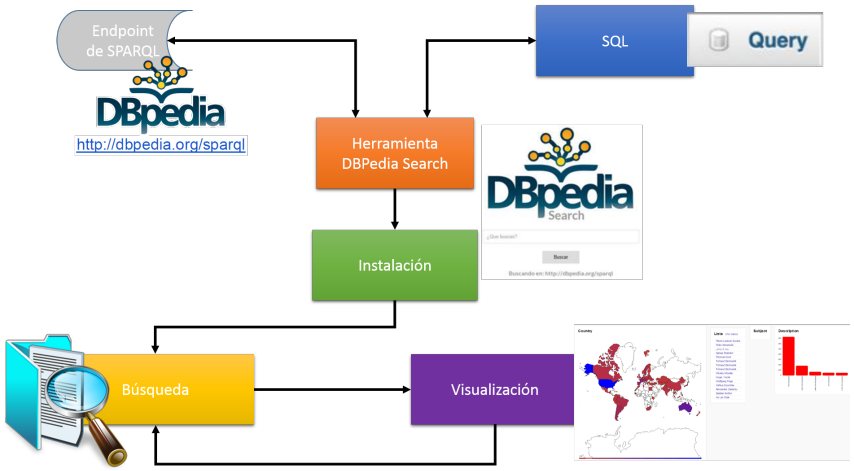


Fig. 4. Diagrama de bloques.

La Base de datos Relacional (SQL) y el *endpoint* de SPARQL son dos servicios independientes que están conectados a la herramienta. La Base de datos Relacional se utiliza para guardar información del *endpoint*; es decir,

información de las URIs y las propiedades encontradas. Esto con la finalidad de no realizar la fase de Análisis de datos cada vez que se realiza la misma consulta. Convirtiéndose en una pequeña caché que mejora la velocidad de la búsqueda y ahorra tiempo en la ejecución de la consulta. En el caso de que se agreguen, modifiquen o eliminen tripletas en la base de datos de tripletas (ontología), el usuario final tiene la seguridad de que la información que obtiene en su consulta está actualizada, ya que el programa se enlaza directamente con la base de datos de tripletas y obtiene todos los resultados en tiempo real.
La instalación de la herramienta sólo se realiza una vez, aunque se cambie de *endpoint*, lo único que se debe llevar a cabo es el borrado de los registros que se tienen almacenados en la Base de datos Relacional. Por lo que, la herramienta únicamente itera entre las actividades de búsqueda y visualización de la información consultada, como se muestra en la Figura 4. De esta manera, se forma un ciclo entre las fases de Análisis de datos y Visualización de las estadísticas.

4. Experimentos y resultados

Para llevar a cabo las pruebas de la herramienta, se insertó directamente la dirección del *endpoint* de la versión en inglés de la ontología DBpedia en la Base de datos Relacional; una vez que se realizaron varias consultas, decidimos cambiar el *endpoint* para trabajar con otras bases de datos de tripletas, con la finalidad de verificar que la herramienta funcione correctamente con diferentes *endpoints*.
En esta sección, únicamente mostramos tres capturas de pantalla de nuestra herramienta en diferentes Sistemas Operativos; una impresión de pantalla por Sistema Operativo.

Windows 8.1
En este Sistema Operativo fue instalado el *endpoint* de SPARQL de la versión en español de la DBpedia [8], como se muestra en la Figura 5.
Endpoint: http://es.dbpedia.org/sparql

Ubuntu 14.04
En este Sistema Operativo fue instalado el *endpoint* de SPARQL de la Biblioteca del Congreso Nacional de Chile/BCN [17], como se muestra en la Figura 6.
Endpoint: http://datos.bcn.cl/sparql

Mac OS
En este Sistema Operativo fue instalado el *endpoint* de SPARQL de *Serendipity* [18], como se muestra en la Figura 7.
Endpoint: http://serendipity.utpl.edu.ec/lod/sparql

Para obtener el rendimiento de nuestra herramienta entre los Sistemas Operativos, utilizamos el mismo endpoint (versión en inglés de la ontología DBpedia) y realizamos las mismas consultas en cada uno. Los resultados del Sistema Operativo Mac OS se indican en la Tabla 1, los del S.O. Windows en la Tabla 2 y los del S.O. Linux en la Tabla 3. [h]

Fig. 5. Búsqueda en español.

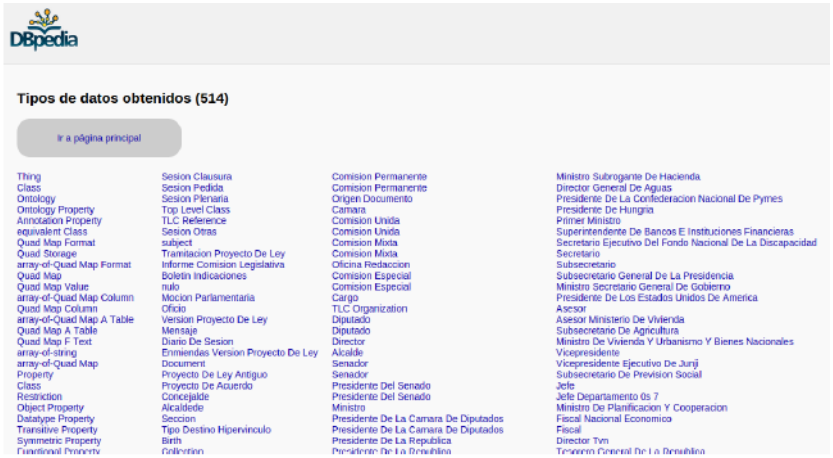


Fig. 6. Tipos de datos (categorías).

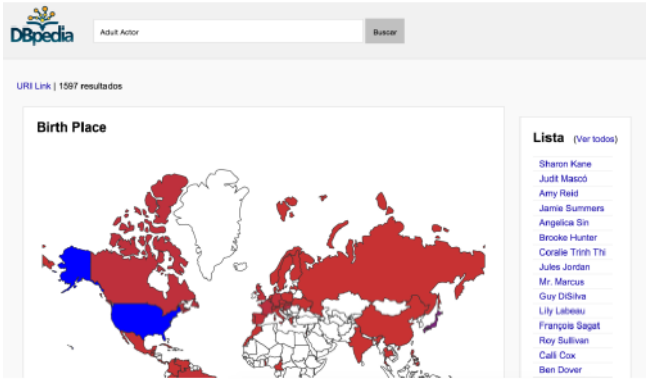


Fig. 7. Visualización de los datos de Serendipity.

Tabla 1. Rendimiento de la herramienta en OS X 10.10.3.

Query	Primera búsqueda	Siguientes búsquedas	No. Resultados
Australia International Soccer Players	39.7 seg.	10.7 seg.	329
Social Scientist	46.3 seg.	8.6 seg.	10,364
Computer Game Program	48.9 seg.	11.4 seg.	10,000
Visualización de la herramienta	0.0433 No. resultados/seg.	0.0114 No. resultados/seg.	

Tabla 2. Rendimiento de la herramienta en Windows 8.

Query	Primera búsqueda	Siguientes búsquedas	No. Resultados
Australia International Soccer Players	41 seg.	10.65 seg.	329
Social Scientist	47.1 seg.	8.52 seg.	10,364
Computer Game Program	48.2 seg.	11.7 seg.	10,000
Visualización de la herramienta	0.0446 No. resultados/seg.	0.01143 No. resultados/seg.	

La última fila de cada tabla (Visualización de la herramienta) representa el rendimiento de la herramienta al momento de ser visualizada en el navegador; para obtener estos valores se utilizó la herramienta *Page Speed Monitor* de [19].

De acuerdo con los resultados registrados en las Tablas 1, 2 y 3, el Sistema Operativo que ofrece un mejor tiempo de respuesta y rendimiento es Linux - Fedora 17, aunque la diferencia no es tan significativa en los otros S.O.

5. Comparativo con otras herramientas

Con la finalidad de identificar las ventajas y desventajas de nuestra herramienta respecto de las herramientas presentadas en la Sección de Trabajo rela-

Tabla 3. Rendimiento de la herramienta en Linux - Fedora 17.

Query	Primera búsqueda	Siguientes búsquedas	No. Resultados
Australia International Soccer Players	37.52 seg.	10.4 seg.	329
Social Scientist	44.1 seg.	11.1 seg.	10,364
Computer Game Program	45.3 seg.	10.9 seg.	10,000
Visualización de la herramienta	0.0409 No. resultados/seg.	0.01125 No. resultados/seg.	

cionado, construimos dos tablas comparativas. La Tabla 4 está enfocada en los aspectos técnicos, dentro de los cuales consideramos el lenguaje de programación utilizado para construir el motor de búsqueda, el nombre del framework/software de almacenamiento de la información en tripletas y mencionar si la herramienta utiliza un *endpoint* de SPARQL. La Tabla 5 contiene características centradas en el desarrollo y funcionamiento de la herramienta, estas características son: indicar si la herramienta presenta una interfaz amigable para el usuario y en qué plataformas funciona; si la herramienta es portátil; si se considera escalable; el tipo de enfoque (o grupo) al que pertenece el motor de búsqueda (de los mencionados en la Sección 2); si utiliza Lenguaje Natural en las consultas y; si es comercial.

Tabla 4. Comparativo: aspectos técnicos de las herramientas.

Motor de búsqueda	**Lenguaje de programación**	**Almacenamiento en tripletas**	**Endpoint**
SHOE	JAVA	Parka KB	No disponible
Yummly	NodeJS	No disponible	No disponible
TAP	No disponible	No disponible	GetData
SemSearch	JAVA	Repositorios de datos semánticos	Múltiples ontologías
Evi	JAVA y Objective C	No disponible	No disponible
AquaLog	No disponible	Sí	Sí
DBPedia Search	PHP	MySQL (aunque no es un almacenamiento en tripletas)	Intercambiable

6. Conclusiones y trabajo futuro

Un aspecto que consideramos importante mencionar es que nuestra herramienta funciona en diferentes Sistemas Operativos, aunque para cada uno de ellos fueron modificadas distintas características de las tecnologías utilizadas; estas modificaciones fueron debidamente documentadas para futuras implementaciones. Esto nos permite difundir rápidamente nuestra aplicación con el fin de validar nuestra herramienta con la mayor cantidad posible de usuarios finales.

El hecho de agregar gráficos en la visualización de los resultados en nuestra herramienta, marca una diferencia notable con las herramientas que se presentan en este documento, ya que ninguna de ellas expone sus resultados utilizando gráficos; lo que representa una característica importante en el análisis de la información para grandes cantidades de datos (Big Data).

Aunque, existe un problema que no podemos erradicar sin el apoyo de las organizaciones enfocadas en trabajar con la Web Semántica, SPARQL y los endpoints; este consiste en que existe una fuerte dependencia en el mantenimiento, disponiblidad y formato de los *endpoints*. Por esto, aunque la herramienta funcione correctamente, si los *endpoints* de SPARQL no se encuentran actualizados, la herramienta no presentará información útil para el usuario.

Como el trabajo futuro consideramos mejorar los aspectos que se enumeran a continuación:

1. Especificar las características de los mapas de calor; por ejemplo, la variación de la paleta de colores.
2. Realizar el tratamiento de las propiedades para combinar categorías comunes, aunque estas categorías contengan diferentes caracteres en sus nombres y/o se encuentren en otros idiomas. Por ejemplo, Lugar de nacimiento, Lugarnacimiento y *Birth Place*.
3. Construir de acuerdo con el tipo de información el (los) gráfico (s) más adecuado (s) para la visualización. De igual manera, permitir que el usuario valide los gráficos presentados por la herramienta, ya sea eliminando o agregando un gráfico.

Agradecimientos. A CONACYT por el apoyo de beca doctoral. A los estudiantes de Doctorado en Ciencias Computacionales del Tecnológico de Monterrey, por su apoyo en la etapa de pruebas.

Referencias

1. Spanos, D-E., Stavrou, P., Mitrou, N.: Bringing Relational Databases into the Semantic Web: A Survey. In: IOS Press, pp. 1–41 (2012)
2. Gutierrez, C., Hurtado, C., Mendelzon, A. O.: Foundations of Semantic Web Databases. In: ACM, PODS, pp. 95–106 (2004)
3. Recuperación y organización de la información a través de RDF usando SPARQL, `https://ggomez.files.wordpress.com/2008/09/informe-sparql.doc`
4. Sakr, S., Al-Naymat, G.: Relational Processing of RDF Queries: A Survey. In: SIGMOD Record, pp. 23–28 (2009)
5. Shadbolt, N., Hall, W., Berners-Lee, T.: The Semantic Web Revisited. In: IEEE Intelligent Systems, pp. 96–101 (2006)
6. Acosta, M., Vidal, M.-E., Lampo, T., Castillo, J., Ruckhaus, E.: ANAPSID: An Adaptive Query Processing Engine for SPARQL Endpoints. In: Lecture Notes in Computer Science, The Semantic Web – ISWC, vol. 7031, pp. 18–34 (2011)
7. DBpedia. `http://dbpedia.org/`
8. Spanish DBpedia. `http://es.dbpedia.org/index-en.html`
9. Heflin, J., Hendler, J.: Searching the Web with SHOE. In: Artificial Intelligence for Web Search, AAAI Workshop, WS-00-01, pp. 35–40 (2000)
10. Semantic Search. `http://www.willita.de/teaching/semweb14w/slides/4S_SemanticSearch.handout.pdf`

11. Guha R., McCool R.: TAP: A Semantic Web Platform. Computer Networks, vol. 42 (5), pp. 557–577 (2003)
12. Lei, Y., Uren V., Motta E.: SemSearch: A Search Engine for the Semantic Web. In: EKAW'06 Proceedings of the 15th international conference on Managing Knowledge in a World of Networks, vol. 4248, pp. 238–245 (2006)
13. Evi Technologies Ltd. `https://www.evi.com/`
14. Lopez, V., Pasin M., Motta E.: AquaLog: An Ontology-Portable Question Answering System for the Semantic Web. In: The Semantic Web: Research and Applications, vol. 3532, pp. 546–562 (2005)
15. ARC RDF Classes for PHP. `https://github.com/semsol/arc2`
16. D3 Data-Driven Documents. `http://d3js.org/`
17. Biblioteca del Congreso Nacional de Chile / BCN. `http://www.bcn.cl/`
18. Serendipity. `http://datahub.io/es/dataset/serendipity`
19. Page Speed Monitor. `https://chrome.google.com/webstore/detail/apptelemetry-page-speed-m/anlomjepbdgcgkebglgfpkinmdjgelhd?hl=en`

Tabla 5. Comparativo sobre los aspectos del desarrollo de las herramientas.

Motor de búsqueda	Amigable para el usuario	Portátil	Escalable	Enfoque	Lenguaje Natural	Comercial
SHOE	El plugin no se encuentra actualizado; por lo tanto, no se puede ver en los navegadores web actuales	No mencionado en el artículo	Sí	Basado en forma	No	No
Yummly	Sí, disponible en la Web y como una aplicación móvil	No, aunque tiene un API	No, está construido específicamente para ontologías de comida	Basado en forma	Sí	Sí
TAP	Las demostraciones ya no están disponibles	Sí	Sí, a través de GetData	Basado en palabra clave	No	No
SemSearch	Sí, aunque no con el detalle de otros motores de búsqueda	No mencionado en el artículo	Sí	Basado en palabra clave	No	No
Evi	Sí, disponible en aplicaciones móviles	No, sólo privado	Sí	Pregunta / Respuesta	Sí	Sí
AquaLog	Las demostraciones ya no están disponibles	Sí	Sí	Pregunta / Respuesta	Sí	No
DBPedia Search	Sí, a pesar de que solo puede ser visto en la Web, es la única herramienta que utiliza la visualización de datos por medio de gráficos	Sí, puede utilizar otras ontologías además de DBpedia	Puede utilizar sólo una ontología	Basado en palabra clave	No	No, por el momento

Sistema automático para la clasificación de la opinión pública generada en Twitter

José R. Gálvez-Pérez[1], Bárbara Gómez-Torrero[1], Raúl I. Ramírez-Chávez[1], Kathia M. Sánchez-Sandoval[1], Vicente Castellanos-Cerda[1], Roberto García-Madrid[2], and Héctor Jiménez-Salazar[1] y Esaú Villatoro-Tello[1]

[1] División de Ciencias de la Comunicación y Diseño, Universidad Autónoma Metropolitana Unidad Cuajimalpa, México D.F.

[2] División de Ciencias y Artes para el Diseño, Universidad Autónoma Metropolitana Azcapotzalco, México D.F.

{joseramon.galvezperez, barb.torrero, rich1983, kamissonce}@gmail.com
{vcatellanos, hjimenez, evillatoro}@correo.cua.uam.mx
gmra@correo.azc.uam.mx

Resumen. La facilidad de acceso a la diversidad de contenidos hace de Twitter un medio para pronunciarse sobre temas actuales o eventos en tiempo real. Los usuarios de esta red no cumplen únicamente una función pasiva de recepción y consumo de información, sino que al mismo tiempo se convierten en productores de contenidos. El estudio de tuits requiere de una disciplina que permita trabajar con el lenguaje en busca de patrones, los cuales hagan posible proponer una metodología con el fin de analizarlos automáticamente, considerando el contexto en el que han sido publicados. La presente investigación describe un sistema automático para el idioma español que permite conocer la polaridad de la opinión pública manifestada en Twitter respecto a temas políticos de la Ciudad de México.

Palabras clave: análisis de opinión, diccionarios afectivos, procesamiento de lenguaje natural, visualización de información, aplicaciones del procesamiento de lenguaje natural.

1. Introducción

Hoy en día es indiscutible el papel que están jugando las redes sociales en Internet, que en conjunto con la tecnología, el impacto potencial de la información sobre la vida política, económica y social transforma las dinámicas para comunicarse y adquirir información.

Los medios de comunicación tradicionales ya no son los únicos referentes sobre los acontecimientos del día a día y con el paso del tiempo, se ha incrementado la cantidad de fuentes a las que se tiene acceso para tener mayor conocimiento de lo que sucede en el mundo. Las personas y los usuarios con ayuda de la tecnología, difunden o dan a conocer parte de la información que surge a cada momento.

En esta línea, Twitter es una red para pronunciarse de forma inmediata sobre temas de cualquier índole que suceden en tiempo real o aquellos que cobran popularidad. Los usuarios de esta red no cumplen únicamente una función de consumo de información, sino que tienen la posibilidad de convertirse en difusores y/o productores de contenidos a través de sus experiencias, testimonios y opiniones personales.

La gran cantidad de información disponible y la velocidad en la que se publica, ha generado dificultades técnicas en el procesamiento para su análisis e interpretación, por lo que se requiere de herramientas que maximicen los beneficios potenciales en esta exploración.

En el presente trabajo, se desarrolló un sistema automático para la recuperación y clasificación de tuits, con el fin de identificar la postura del público de Twitter respecto a temas políticos nacionales. Se describe un sistema automático para el idioma español, que permite acercarse a la polaridad, en forma visual, de la opinión pública de la ciudad de México manifestada en Twitter, teniendo a la mano datos que soporten la toma de decisiones de un usuario especializado en la opinión pública.

El resto de este trabajo está organizado de la siguiente manera: la sección 2 describe brevemente la manera tradicional de atacar el problema de polaridad así como las características de algunos sistemas existentes actualmente. En la sección 3 se describe tanto el recurso léxico como el corpus empleado en el desarrollo de nuestro sistema, de igual forma se describen los resultados obtenidos durante la fase experimental. Posteriormente, en la sección 4 se describen las características del sistema desarrollado así como los resultados de una evaluación realizada con fines de definir el grado de usabilidad de la aplicación propuesta. Finalmente, en la sección 5 se derivan algunas conclusiones e ideas de trabajo futuro.

2. Antecedentes

Producto de la gran cantidad de publicaciones que se generan en Twitter, es notorio que el público tiende a bifurcarse entre aquellos que están a favor o en contra de algún acontecimiento. Es por esto que actualmente hay un fuerte interés por parte de especialistas relacionados en las áreas de la mercadotecnia, política, social y de comunicación en contar con herramientas capaces de recuperar mensajes escritos en las redes sociales para su análisis, con el fin de detectar opiniones de los usuarios que permitan mostrar tendencias respecto a determinados temas[3]. Este interés se extiende incluso al público en general que está atento en saber lo que sucede en redes sociales, debido a la inmersión que ha tenido la tecnología en la vida diaria.

En este contexto, existen intentos por desarrollar herramientas que hagan uso de datos obtenidos de Twitter con el fin de identificar la postura de los usuarios respecto a algún tema, los cuales están orientados a trabajar con el idioma inglés en su mayoría. Dentro de los intentos por resolver el problema

[3] Hasta el año 2014 el foro RepLab representaba el sitio más relevante donde este tipo de sistemas erán evaluados `http://nlp.uned.es/replab2014/`.

de clasificar opiniones publicadas en Twitter para el idioma español de forma automática, estas empresas se han enfocado a las búsquedas de productos con fines mercadológicos. En este caso, mediante ciertas características sintácticas o semánticas se realiza la clasificación de tuits en positivos, negativos o neutros [11]. También se ha estudiado si los métodos que han sido efectivos para la clasificación de opiniones de tuits en inglés, lo son para español [3]. Estos métodos consideran las palabras que se usan en los tuits para darles un valor semántico, las negaciones, e incluso el procesamiento de enlaces a otros sitios web. Sin embargo, este estudio concluyó que los métodos para analizar y clasificar automáticamente tuits en inglés no dieron buenos resultados al implementarlos en tuits escritos, puesto que no se puede establecer un contexto a cada tuit de forma automática.

Como estos trabajos existen muchas propuestas más, las cuales enfocan sus esfuerzos en la búsqueda de formas adecuadas de representación de los documentos así como en la identificación de los atributos que resultan más apropiados en la resolución de la tarea [9,6].

Con respecto a obtener resultados visuales provenientes del análisis de datos de Twitter, se encuentran disponibles algunas herramientas automáticas en línea que permiten realizar búsquedas de palabras clave presentando resultados cuantitativos sobre la emoción y sentimiento de cada tuit recuperado[4].

Twitter, ha hecho posible enfocarse en elementos particulares que por medio de una visualización geoespacial, muestra seguimiento de palabras o temas en tiempo real en determinada zona geográfica[5]. De esta forma se elige el punto geográfico a nivel mundial sobre el que hay interés en conocer sobre lo que hablan los usuarios.

Estos antecedentes ponen de manifiesto la inquietud por trabajar con información derivada de Twitter, y analizarla con el fin de estudiar las reacciones de la sociedad respecto a algún tema en particular, asimismo, que no hay sistemas para el idioma español que ofrezcan una aproximación a la polaridad de un tópico particular de Twitter. Y aunque pueden ser útiles los métodos empleados en otros sistemas desarrollados para lenguas diferentes al español, orientados a la clasificación de polaridad de tuits, no es viable adaptarlos al español. Por lo anterior, dentro de éste trabajo proponemos y describimos el desarrollo de un sistema que analice la opinión que generan los temas políticos nacionales, basado en un análisis contextual sobre el uso que se le da a Twitter en la Ciudad de México.

3. Método de clasificación automática de opinión

Uno de los elementos relevantes para proponer el método de clasificación de opinión fue la conformación de un diccionario afectivo de palabras, donde cada palabra tiene asociado un valor que determina su escala positiva o negativa. Agregado a esto, fue necesario también la construcción de un corpus de trabajo, sobre el cual fuera posible evaluar la pertinencia de método propuesto.

[4] `http://www.csc.ncsu.edu/faculty/healey/tweet_viz/`
[5] `http://trendsmap.com/topic/%23cnte`

En las siguientes secciones se detalla el proceso de construcción del diccionario como la recolección del corpus de trabajo.

3.1. Diccionario afectivo

Estudiar y conocer el contexto fue muy importante para elegir las palabras que integrarían un diccionario que sería la base del funcionamiento del sistema automático propuesto.

Para la conformación del diccionario de palabras que permitiera clasificar los tuits, se inició con la construcción de una lista de palabras obtenida del mismo corpus. Posteriormente se integraron palabras del diccionario Spanish Emotion Lexicon (SEL) proveniente del trabajo "Creación y evaluación de un diccionario marcado con emociones y ponderado para el español" [2]. En dicho trabajo se realizó una investigación a partir del interés de analizar opiniones en las redes sociales con atención a Twitter y presentaron una lista de 2036 palabras en español relacionadas con seis emociones básicas (alegría, sorpresa, repulsión, miedo, enojo, tristeza). Cada palabra de este diccionario tiene asignado un factor de probabilidad de uso afectivo (PFA) que indica el grado en que puede presentar su uso en relación con determinada emoción. Cabe señalar que la escala de valores del PFA es de 0 a 1, siendo 1 el valor máximo. Así entonces, el diccionario con el cual se trabajó está conformado por una lista de 1443 palabras calificadas con su correspondiente PFA. Es importante mencionar que nuestro diccionario resultó de menor tamaño al de [2] debido a que sólo consideramos las palabras con un PFA con valores entre 0.5 y 1.

3.2. Corpus

En el área de procesamiento de lenguaje natural, existen tareas para las cuales resulta relevante contar con colecciones de datos (textos), los cuales muestran de manera natural ejemplos del uso de la lengua. Este conjunto de documentos llamado "corpus" y a las aplicaciones que utiliza un corpus para obtener las reglas de interpretación, se le conoce como "lingüística basada en corpus".

El corpus de la presente investigación está formado por 2507 tuits recuperados en junio del 2013 y julio del 2014. Los temas recuperados fueron: #PVEM, Peña Nieto, Hugo Sánchez, CNTE, Reforma Energética, Chapo Guzman y #EPNvsInternet. Cada tuit fue clasificado de acuerdo a la opinión en consenso por parte de un grupo de cuatro expertos, tomando en cuenta el contexto social y costumbres expresivas de la Ciudad de México. Esto permitió asignar una etiqueta a cada tuit: negativo, positivo o neutro (estos últimos correspondían en su mayoría a las notas informativas de los medios de comunicación en Twitter). El corpus etiquetado fue tomado en cuenta para comparar los resultados obtenidos en los experimentos subsecuentes. Este proceso fue necesario para poder formar un marco de referencia que sirviera para evaluar el desempeño de los métodos propuestos.

Tabla 1. Muestra de los tuits recuperados y su clasificación asignada por los expertos.

Categoría de los tuits	**Entidad y/o tópico principal**				
	Peña_Nieto	*Hugo_Sánchez*	*#PVEM*	*CNTE*	***Total***
Positivos	7	1	7	27	**42**
Negativos	99	71	44	211	**425**
Neutral	75	29	67	262	**433**
Total	181	101	118	500	**900**

3.3. El método de clasificación

El método de clasificación de un tuit es determinado tomando en cuenta la ocurrencia de las palabras del diccionario afectivo (Sección 3.1) así como su ponderación afectiva. La polaridad de un tuit se determina por medio de realizar la combinación lineal de los pesos asignados a cada una de las palabras que aparecen dentro de un tuit y que ocurren dentro del diccionario afectivo. La Figura 1 muestra de manera esquemática el algoritmo de clasificación de opiniones.

El método comienza por hacer una comparación entre el tuit en revisión T y una serie de heurísticas que ayudan a determinar cuando un tuit carece de opinión, a las cuales llamamos T_I (*tuits informativos*). En caso de que T contenga características que pertenecen a los T_I se le asigna la etiqueta de T_{Winf} la cual indica que es un tuit informativo, mismo que lo define como un tuit de polaridad neutral. En el caso contrario, si T no tiene características de T_I, se etiqueta como T_{Wop} indicando ser un tuit de opinión. Es importante mencionar que el conjunto de heurísticas contenidas en T_I representa un conjunto de reglas que permiten distinguir cuando un tuit contiene URLs que refieren a sitios formales de información, por ejemplo, periódicos en línea.

Posteriormente se realiza una comparación con de cada palabra contenida en T con el diccionario afectivo D en busca de palabras (D_w) que pertenezcan a D. En caso de encontrarlas se identificará el D_{num} (valor afectivo de cada palabra) correspondiente de la(s) D_w, de lo contrario se asignará un valor cero (neutro) a la palabra. En este sentido, un valor afectivo de 0 significa que la palabra en revisión no tiene carga afectiva, *i.e.*, es neutra.

Una vez identificados los D_{num} en T, se realiza una sumatoria para obtener un resultado N (número natural), el cual dependiendo su valor final, indicará la clasificación que tendrá el tuit T en revisión. Si el resultado de la sumatoria es cero, clasificará al tuit como neutro; si es menor a cero se clasificará como negativo y con un valor mayor a cero se le asignará la etiqueta de positivo.

3.4. Medidas de evaluación

Para evaluar el método de clasificación propuesto se utilizaron las medidas de precisión (P), recuerdo (R) y medida F (F), que son medidas comunes en el

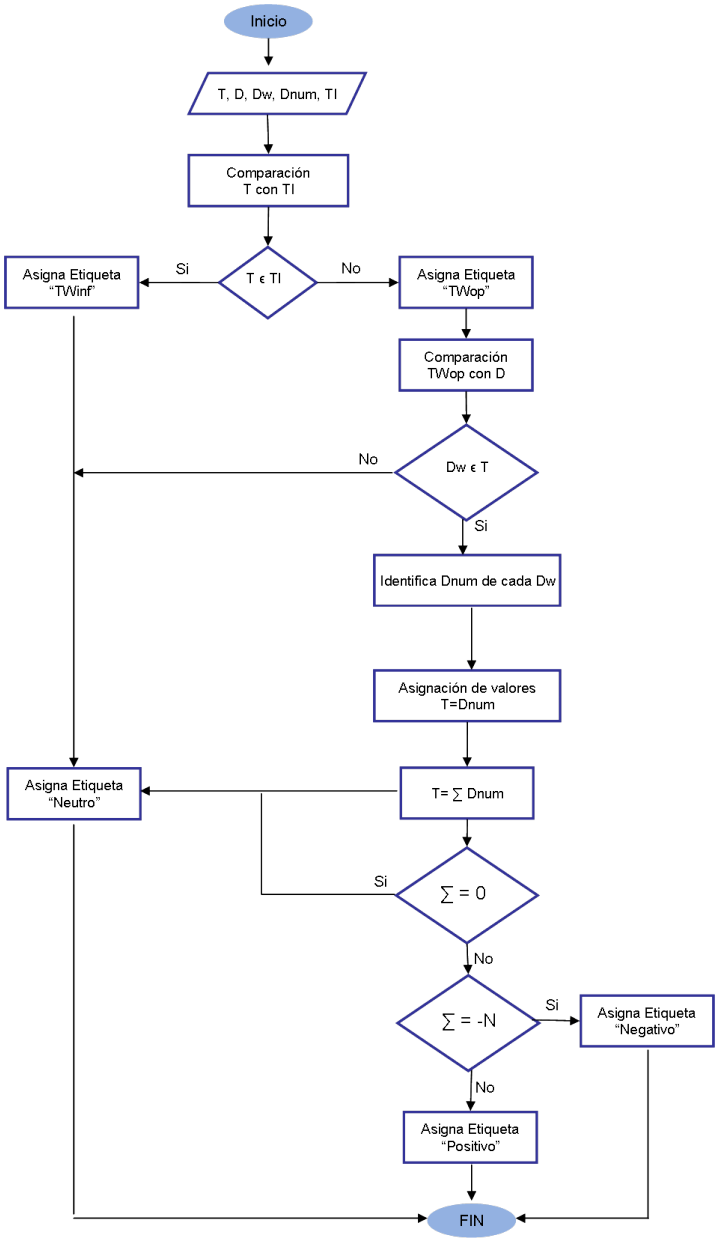


Fig. 1. Diagrama de flujo del algoritmo de clasificación de opinión en Tuiter. Note que de entrada se recibe al tuit T, el diccionario D, D_w y D_{num} refieren a las palabras del diccionario y su correspondiente peso afectivo respectivamente. Finalmente, T_I refiere a la ocurrencia de URLs dentro del tuit T.

área de Aprendizaje Automático. La precisión es la proporción de tuits que el sistema clasificó correctamente sobre el total de tuits que deben ser clasificados. El recuerdo es la proporción de los tuits que el sistema clasificó correctamente sobre el total de los tuits que el sistema clasificó [4,7]. Por último, la exactitud es la proporción del número total de predicciones que son correctas mientras que la medida F se considera como una medida armónica entre precisión y recuerdo.

La escala de valor que se maneja para cada medida va de 0 a 1, teniendo como valor máximo 1. Los resultados obtenidos haciendo uso de estas medidas, son los referentes para comparar el método propuesto y su aplicación en los diferentes experimentos realizados, identificando los casos de aciertos y errores. Para efectos de evaluación de resultados, se ha tomado en cuenta la media F y la *Precisión*.

3.5. Evaluación experimental

Para conocer el desempeño del método de clasificación propuesto, éste se implementó con cada conjunto de tuits recuperado. Teniendo el corpus etiquetado en su totalidad, fue posible comparar y analizar los resultados del método aritmético, con la clasificación de experto con el fin de mejorar el clasificador.

Es importante mencionar que el método propuesto contiene una serie de heurísticas que permiten determinar (con cierta confianza) cuando un tuit es un mensaje informativo y/o objetivo. De esta forma, dichos mensajes no son analizados por el método descrito en la Figura 1, y se evita introducir ruido al sistema de clasificación.

La Tabla 2 muestra los resultados obtenidos de la clasificación de la opinión en el conjunto de tuits descrito en la sección 3.2.

Mediante el proceso de experimentación del clasificador automático, se fueron tomando en cuenta adecuaciones al método propuesto para mejorar su efectividad. La mayor consideración fue en relación al diccionario de términos y su incremento con el fin de tener más elementos que permitan realizar una clasificación de tuits más apropiada.

Las modificaciones que Twitter ha hecho en los últimos años, ponen de manifiesto que trabajar con esta red conlleva ajustes constantes en la metodología de los sistemas de clasificación de opinión. Todo esto se ve reflejado en el uso que se le da a la red, y por consiguiente en la estructura de los tuits, sin embargo, la constante sigue siendo el uso que se le da al lenguaje para manifestar una opinión, lo cual sustenta el trabajo descrito en este artículo.

4. Sistema automático para la clasificación de la opinión pública

Una vez que se desarrolló el método para clasificar automáticamente las opiniones emitidas en los mensajes de Twitter, se dio paso al diseño de la parte visual e interactiva del sistema, que incluía la interfaz de uso, así como las distintas visualizaciones presentadas. El objetivo principal era lograr que el

Tabla 2. Evaluación del método de clasificación automática sobre el total de conjuntos recuperados, haciendo uso del método de clasificación definitivo en el cual se considera el total de los tuits recuperados sin separarlos en informativos y de opinión. El #EPNvsInternet generó mayor opinión por parte de los usuarios y fue con este conjunto donde se obtuvieron los valores más altos en la evaluación.

Tópico	**Medidas de Evaluación**		
	P	*R*	*F*
#PVEM	0.568	0.590	0.458
Hugo_Sánchez	0.571	0.858	0.569
Peña_Nieto	0.658	0.634	0.641
CNTE	0.443	0.484	0.425
Reforma_Energética	0.558	0.547	0.521
Chapo_Guzman	0.581	0.553	0.557
#EPNvsInternet	0.739	0.715	0.724

usuario conociera la polaridad de opinión respecto a diversas temáticas dentro del contexto político resultante de los mensajes publicados en Twitter. Asimismo, se le presentarían visualizaciones que le permitieran realizar acciones posteriores como el almacenamiento y análisis comparativo de resultados o toma de decisiones con base a ellos.

4.1. Diseño de la interfaz y las visualizaciones

La interfaz se refiere a la organización de elementos dispuestos en pantalla mediante los cuales el usuario hará uso del sistema. Su diseño debía enfocarse en lograr que fuera sencilla e intuitiva, sin elementos distractores que facilitaran que la atención del usuario estuviera puesta en los resultados que mostrara el sistema.

Dado lo anterior, la interfaz del sistema se orientó a dirigir al usuario de forma clara y concisa a las opciones que ofrece el sistema para acceder a las distintas gráficas, las cuales proveen distintos maneras de hacer una visualización de la información soportada por el propio sistema [1]. En ellas se despliegan cantidades medibles a través de puntos, líneas, sistema de coordenadas, números, símbolos, palabras y color [10]. Asimismo, las gráficas son presentaciones visuales - breves - que ilustran una o más relaciones entre números, y que nos permiten apreciar relaciones cuantitativas entre muchos elementos y darnos información precisa [5].

Así fue como se realizó una versión digital del prototipo en HTML y PHP para su implementación web. El trabajo en HTML se enfocó a la estructura formal del prototipo, y con PHP se programó el clasificador automático, permitiendo hacer adecuaciones relacionadas con la interfaz y las visualizaciones de información de manera independiente al clasificador. Por otra parte, las visualizaciones son creadas con la librería D3.js, la cual hace posible generar gráficos al momento de obtener los resultados obtenidos de la clasificación automática. La Figura 2

Fig. 2. Pantalla donde se muestran las "Categorías" de los tuits que están siendo analizados. Las categorias corresponden a tuits informativos o tuits de opinión. El sistema SCOPT sólo determina la polaridad de aquellos tuits que se consideran subjetivos, *i.e.,* tuits de opinión.

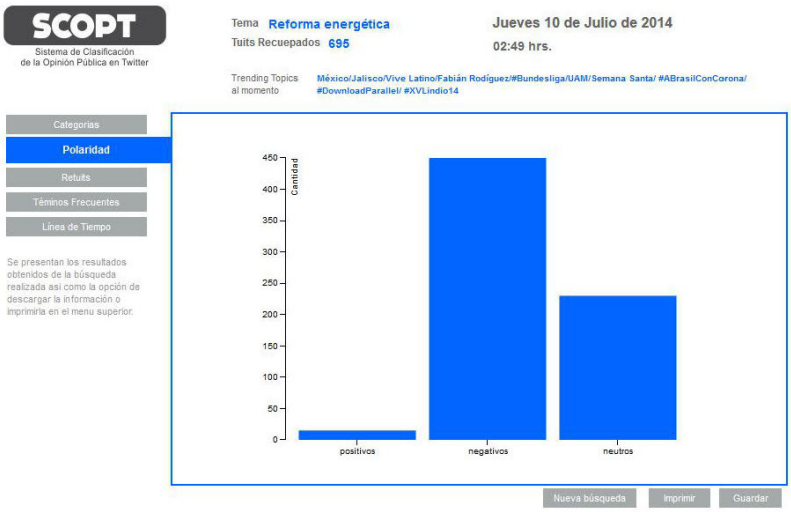


Fig. 3. Ejemplo de visualización "Polaridad". Aquí el usuario puede conocer rápidamente la cantidad de tuits positivos, negativos y/o neutrales existentes en la muestra de tuits que están siendo analizados.

muestra unos ejemplos de las pantallas principales del sistema SCOPT, mismo que actualmente está alojado en `http://lyr.cua.uam.mx`

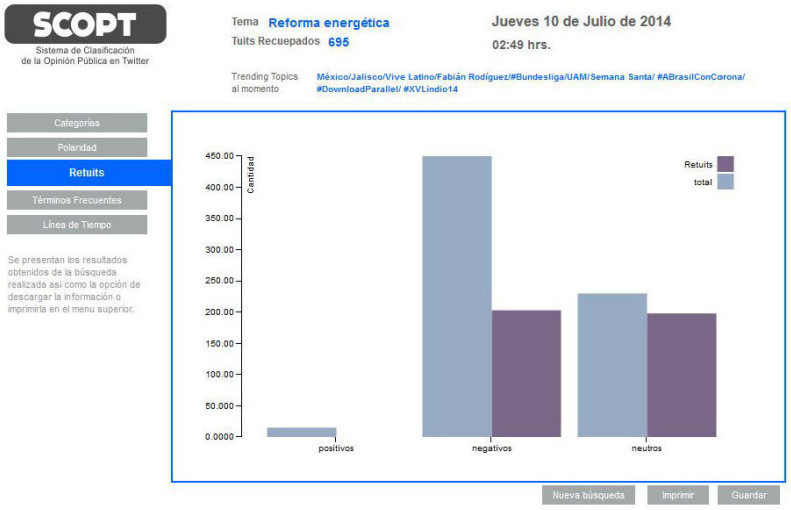


Fig. 4. Ejemplo de visualización "Retuits". En esta pantalla el usuario puede valorar la polaridad de los tuits que están siendo analizados conociendo cuántos de éstos son resultado de un *retuit*.

4.2. Comprobación y refinamiento

La evaluación tiene el propósito de conocer problemas de usabilidad y aplicar un proceso iterativo de ajustes al prototipo antes de la implementación del sistema. En esta etapa la propuesta se puso a prueba con usuarios potenciales con el fin de recabar información valiosa para su mejoramiento. La serie de ajustes y refinamientos se siguieron hasta lograr un nivel óptimo de eficiencia.

Las pruebas de "eye tracking" pueden resultar valiosas para constatar adónde se dirige la atención de las personas que están haciendo uso del sistema. "El concepto de eye-tracking hace referencia a un conjunto de tecnologías que permiten monitorizar y registrar la forma en que una persona mira una determinada escena o imagen, en concreto en qué áreas fija su atención, durante cuánto tiempo y qué orden sigue en su exploración visual" [8]. El eye-tracking es una tecnología de seguimiento ocular con mucho auge en el mundo de la usabilidad. Si bien los datos obtenidos mediante estas pruebas nos permiten saber dónde fija su atención el usuario, y qué zonas pasan desapercibidas, esta información puede resultar limitada porque no explica las causas por las que esto pasa, así que conviene establecer algún vínculo entre fijaciones y actividad cognitiva. Por ello, se pensó combinar la aplicación de un test de tareas de forma complementaria

Fig. 5. Ejemplo de visualización de los "Términos frecuentes". El objetivo de esta pantalla es proporcionar al usuario un vistazo rápido de los términos más comúnmente utilizados en la muestra de tuits en revisión. Idealmente darán una idea intuitiva de la temática de los mensajes.

Fig. 6. Ejemplo de visualización de la "Línea de Tiempo". En esta pantalla el usuario puede consultar el historial de sus búsquedas realizadas y comparar gráficamente los cambios de polaridad que han sucedido desde la primera búsqueda.

con la prueba de eye-tracking, pues cada uno aporta información exclusiva y facilitaría la interpretación de los datos obtenidos. En la Figura 3 se muestran las distintas pantallas que se presentan al usuario, con representaciones visuales de los recorridos de los cuatro participantes evaluados.

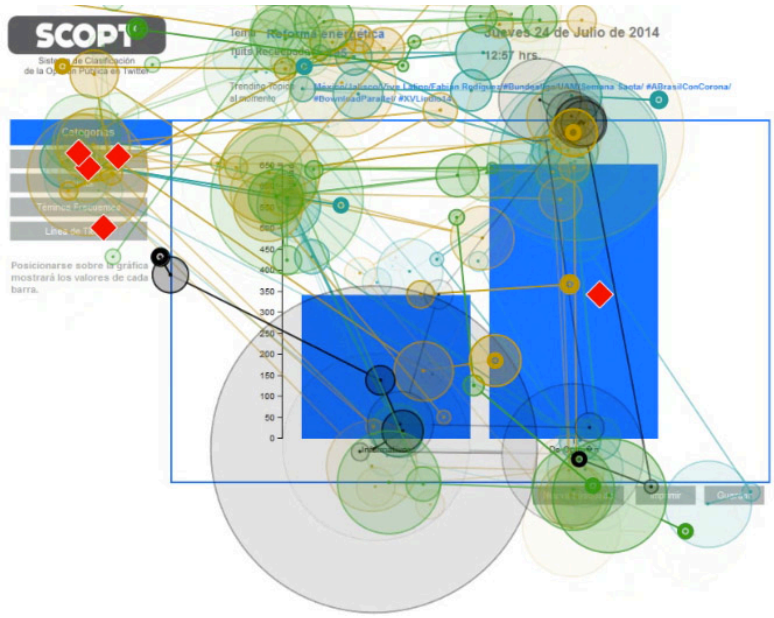


Fig. 7. Recorridos visuales de los participantes evaluados en la pantalla de "Categorías de Tuits (Informativo *VS.* Opinión)". Evaluación registrada por medio del sistema de eye-tracking.

Entre los resultados que arrojó esta etapa de evaluación dentro del proceso de diseño de la herramienta, encontramos que la interfaz cuenta con un diseño muy sintético y sin mayor problema para localizar, leer y entender la información mostrada, pues no se presentan elementos distractores dentro de la pantalla que desvíen la atención del usuario.

5. Conclusiones y trabajo futuro

En este trabajo, dentro de los temas políticos en la Ciudad de México, se pudo constatar que Twitter es un canal de opinión: sus usuarios a partir de las publicaciones que emite y hace lectura, estimula la continuidad de la discusión de un tema, nutre sus comentarios o refuerza su postura con la incorporación de material multimedia y enlaces web.

La exploración de los tuits generados ante eventos políticos nacionales, permitió descubrir posibles patrones en las formas de opinar y dependiendo del

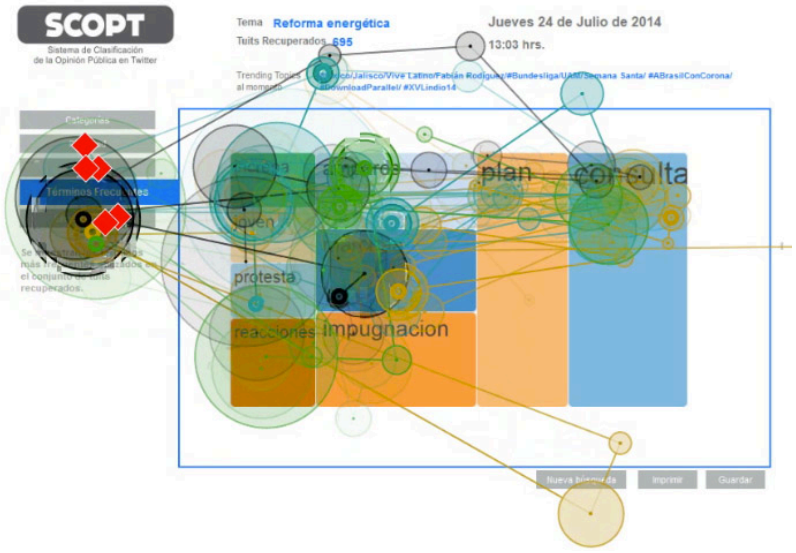


Fig. 8. Recorridos visuales de los participantes evaluados en la pantalla de "Términos Frecuentes". Evaluación registrada por medio del sistema de eye-tracking.

contexto y el tema se obtuvieron palabras que permitirían encontrar la polaridad de un tuit, permitiendo así la conformación del diccionario afectivo, mismo que es la base del sistema de clasificación propuesto. Entre mayor era la cantidad de palabras que integraban el diccionario de términos, mayor fue la precisión del mismo. Esto se hizo notorio conforme se fueron realizando los experimentos, ya que en cada uno de ellos, el diccionario iba incrementándose. Esta relación reforzó el método propuesto y propició continuar la búsqueda de palabras que hicieran posible determinar la polaridad de un tuit.

En este orden de ideas, para el diseño de interfaz del nuestro sistema, se retomaron los resultados arrojados mediante el cuestionario aplicado y la prueba de eye-tracking, permitiendo mostrar visualizaciones claras, limpias y muy sintéticas, por tanto una correcta lectura y comprensión de la información presentada. Si bien, aún podría pulirse la propuesta y aumentar la interactividad en las gráficas, el sistema arroja información general comprensible para los usuarios especializados, y sin que esto excluya al usuario general.

La generación de un método eficiente, derivado de los estudios y análisis previos, nos permitieron desarrollar SCOPT, bajo el propósito de tener un sistema sencillo de usar, presentando resultados de forma visual además de contar con la posibilidad de descargar la información obtenida para analizarla más puntualmente en caso de que así se requiera. Es importante mencionar que éste tipo de herramientas se vuelven fundamentales para el experto en análisis de la opinión publica, pues le permite de manera rápida y sencilla orientar su trabajo de investigación hacia aquellos temas que son de su interés al mismo tiempo

que le proporciona una aproximación sobre el sentir de la población hacia dicho tema.

Como trabajo futuro planeamos incorporar a la aplicación SCOPT técnicas más sofisticadas de clasificación de polaridad, como lo podría ser la inclusión de atributos estilísticos. De igual forma nos interesa incrementar las heurísticas que ayudan a determinar cuando un tuit es de opinión o informativo. Es conveniente mencionar que el problema de ironía aún no es soportado por el sistema desarrollado, consideramos que el desarrollo de más recursos lingüísticos en combinación con formas alternativas de representación podría ayudar a atacar éste problema.

Agradecimientos. El presente trabajo fue realizado con el apoyo de CONACyT (Becas: 373288, 373287, 373284, 373285). Agradecemos también al programa de Maestría en Diseño, Información y Comunicación (MADIC) de la Universidad Autónoma Metropolitana Unidad Cuajimalpa, así como al SNI-CONACyT.

Referencias

1. Card, S.K., Mackinlay, J.D., Shneiderman, B. (eds.): Readings in Information Visualization: Using Vision to Think. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1999)
2. Díaz-Rangel, I., Sidorov, G., Suárez-Guerra, S.: Creación y evaluación de un diccionario marcado con emociones y ponderado para el español (creation and evaluation of a dictionary tagged with emotions and weighted for spanish). Onomázein, Revista de Lingüística, Filología y Traducción (29), 1–26 (2014)
3. Fernández, A., Nuñez, L., Morere, P., Santos, A.: Sentiment analysis and topic detection of spanish tweets: A comparative study of nlp techniques. Revista de Procesamiento del Lenguaje Natural (50), 45–52 (2013)
4. Hernández, J., Ramírez, J., Ferri, C.: Introducción a la minería de datos. Prentice Hall, Pearson Educación, S.A. (2006)
5. Kosslyn, S.: Graph Design for the Eye and the Mind. Oxford University Press (2006)
6. Leon-Martagón, G., Villatoro-Tello, E., Jiménez-Salazar, H., Sánchez-Sánchez, C.: Análisis de polaridad en twitter. Research in Computing Science 62, 69–78 (2013)
7. Lewis, D.: Evaluating text categorization. In: Proceedings of Speech and Natural Language Workshop. pp. 312–318 (1991)
8. Page, W.: No solo usabilidad: Revista sobre personas, diseño y tecnología, uRL: http://www.nosolousabilidad.com/articulos/eye-tracking.htm#sthash.7N1RrSks.dpuf
9. Sidorov, G., Miranda-Jiménez, S., Viveros-Jiménez, F., Gelbukh, A., Castro-Sánchez, N., Velásquez, F., Díaz-Rangel, I.and Suárez-Guerra, S., Treviño, A., Gordon, J.: Empirical study of machine learning based approach for opinion mining in tweets. In: Lecture Notes in Artificial Intelligence LNAI. vol. 7629, pp. 1–14 (2012)
10. Tufte, E.: The Graphic Display of Quantitative Information. Graphics press (1983)
11. Valderrábanos, A., Torrejón, E.: Natural opinions: extracción de opinión basada en pln para contenidos generados por usuarios. In: CEUR Workshop Proceedings. pp. 339–346. No. 697 (2010)

Aplicación del patrón de transformación de síntesis para la comparación de los lenguajes ATL vs. QVT

Ana Karen Vega Maqueda, S. Gustavo Peláez Camarena, Ulises Juárez Martínez, Ma. Antonieta Abud Figueroa y Luis Ángel Reyes Hernández

Instituto Tecnológico de Orizaba, Departamento de Posgrado e Investigación, Orizaba, Veracruz, México

karen.vemaqueda@gmail.com, sgpelaez@yahoo.com.mx, ujuarez71@gmail.com, aabud@prodigy.net.mx, l_r_h01@hotmail.com

Resumen. La ingeniería dirigida por modelos (MDE) se caracteriza por asignar a los modelos el papel principal durante todas las etapas de desarrollo de software, aumentar la automatización en el proceso de desarrollo separando los aspectos de tecnología y promoviendo la productividad y mejora de calidad de los sistemas. El enfoque MDE ha surgido como un nuevo paso en el camino hacia una verdadera industrialización de la producción de software. Tras el éxito del paradigma orientado a objetos, el uso sistemático de modelos se presenta ahora como la forma apropiada para conseguir programar con un nivel más alto de abstracción y de aumentar el nivel de automatización. Se presenta un análisis de los lenguajes de transformación de modelos ATL y QVT donde se mencionan las características más relevantes de cada uno de los lenguajes y un caso de estudio.

Palabras clave: MDE, ATL, QVT, EMF, XMI.

1. Introducción

En la actualidad los modelos son parte importante dentro de la ingeniería de software, no obstante en la mayoría de los casos se encuentran plasmados en papel en lugar de incorporarse en el proceso de ingeniería, estos son considerados como la representación exacta o abstracción de las propiedades principales de un objeto, sistema o idea.

ATL es un lenguaje de transformación de modelos, el cual en el área de ingeniería dirigida por modelos, proporciona mecanismos para producir un conjunto de modelos de destino de un conjunto de modelos de origen. El enfoque basado en modelos supone proporcionar a los diseñadores y desarrolladores de modelo un conjunto de operaciones dedicadas a la manipulación de los modelos con propósito de obtener un alto nivel abstracción [1].

El objetivo del presente artículo es identificar las características principales de los lenguajes ATL y QVT y efectuar el proceso de transformación de modelo a modelo en un caso de estudio. La estructura del trabajo se describe a continuación: Sección 2: Información sobresaliente del lenguaje de transformación de modelos. Sección 3: Se presentan las características principales de los lenguajes ATL y QVT. Sección 4: Se

presenta la propuesta. Sección 5: Se presenta el empleo de los lenguajes de transformación de modelo ATL y QVT al caso de estudio. Sección 6: Se presentan los resultados obtenidos sobre el trabajo hasta el momento.

2. Trabajos relacionados

En [2] se comparó la propuesta del lenguaje consulta/vista/transformación (QVT) y el lenguaje de transformación (ATL) con el propósito de reunir conocimientos sobre los enfoques de transformación de modelos existentes, se describió la arquitectura, características del lenguaje y se identifican las categorías a comparar en los lenguajes. La atención se centra en los principales componentes del lenguaje y cómo se relacionan, mediante el análisis de diversas categorías (abstracción, paradigma, direccionalidad, cardinalidad, etc.). Se demostró cómo el lenguaje ATL es ejecutado en los motores de QVT y recíprocamente QVT en la máquina virtual de ATL. Por lo tanto es posible tener una interoperabilidad entre los lenguajes ATL y QVT a nivel conceptual, se espera que esta investigación sea útil en el análisis de interoperabilidad para otros lenguajes de transformación.

En [3] se describió el lenguaje de transformación ATLAS (ATL), las herramientas y el conjunto de documentación con ejemplos disponibles en el subproyecto ATL Eclipse / GMT, se analizó que ATL se apoya en un conjunto de herramientas de desarrollo integradas en la parte superior del entorno de Eclipse: un compilador, una máquina virtual, un editor y un depurador. El estado actual de las herramientas de ATL ya permite resolver problemas no triviales y hasta el momento ATL se utiliza y evalúa a través de diversos sitios de índole académica e industrial.

En [4] se describió el lenguaje de transformación de modelos ATL y su entorno de ejecución basado en la infraestructura de Eclipse, este tipo de entorno desarrollo proporciona apoyo en las tareas más relevantes que intervienen en el uso de lenguaje tales como: edición, compilación, ejecución y depuración, se comprobó que el lenguaje permite la aplicación de reglas de transformación imperativas y declarativas para la solución de problemas no triviales.

En [5] se presentaron transformaciones ATL basadas en las reglas de modelos de transformación, proporcionando una codificación intuitiva y versátil de ATL en OCL (lenguaje de restricción de objeto) utilizada para el análisis de diversas propiedades con respecto a las transformaciones, también se describió cómo generar automáticamente modelos de transformación partiendo de transformaciones ATL declarativas, específicamente este trabajo se enfocó en demostrar si un modelo de salida generado por una transformación ATL será válido para cualquier modelo de entrada.

En [6] se analizaron las características de modularidad en el lenguaje de transformación de modelos ATL, se analizaron dos casos donde las unidades modulares se identifican a través de las relaciones entre metamodelos de origen y destino en base a la funcionalidad de transformación genérica, también se aplicaron 3 técnicas de transformación: reglas explicitas, implícitas y de herencia para evaluar distintas implementaciones de los casos. Se concluyó que al aplicar la regla implícita se obtuvo un bajo acoplamiento, por lo tanto esta regla debe aplicarse cuando es primordial la reutilización y la adaptabilidad además la elección de diferentes descomposiciones en

el metamodelo destino llevo a diferentes conjuntos de reglas de transformación, por lo tanto se recomienda tener en cuenta más de una descomposición en los metamodelos.

En [7] menciona la existencia de 4 patrones de transformación de modelos para lenguajes HOTS (*Higher-Order Transformations*) ATL y QVT por mencionar, el primer patrón de transformación es nombrado de Síntesis en el deben definirse los metamodelos origen, destino y el modelo origen y resultado de la transformación es el modelo destino, en el patrón de transformación de análisis no es obligatoria la generación de un modelo destino y para cada mapeo se generan posibles variabilidades del sistema a transformar, el patrón de transformación de composición se define bajo 3 condiciones: Al menos uno de los modelos de origen debe ser una transformación, como mínimo uno de los modelos de salida debe ser una transformación y finalmente los modelos de origen y/o destino deben contener más de una transformación, por último el cuarto patrón es de transformación de modificación se tiene como elemento de entrada una transformación y se genera una versión actualizada de la misma transformación.

3. Lenguaje de transformación de modelos

3.1. Lenguaje de transformación ATL

ATL (ATL Lenguaje de Transformación) es un lenguaje de transformación de modelos y un conjunto de herramientas para el proceso de transformación de modelos. En el campo de la ingeniería dirigida por modelos (MDE), ATL proporciona formas para producir un modelo o un conjunto de modelos destino a partir del modelo o conjunto de modelos origen.

La sintaxis abstracta de ATL es especificada mediante un metamodelo MOF, y provee de lenguajes en modo textual y gráfico para representar las reglas de transformación del lenguaje [2], se muestra la figura 1.

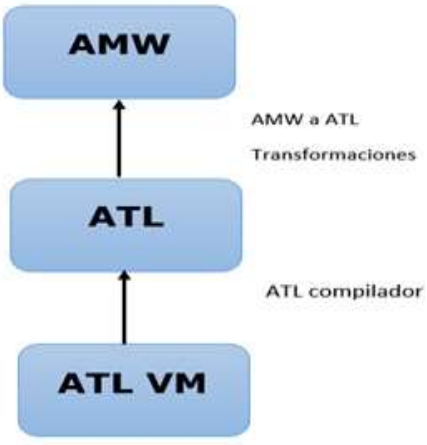


Fig. 1. Arquitectura del lenguaje.

Aquí la primera capa ATL VM significa máquina virtual de ATL, lenguaje ATL, AMW se refiere al modelo vista de ATL. Los programas ATL compilados son ejecutados mediante la ATL VM, que utiliza un conjunto de instrucciones orientadas al modelo. AMW (*ATLAS Model Weaver*) utilizado para establecer y representar las relaciones entre distintos modelos [8].

3.2. Lenguaje de transformación QVT

El lenguaje QVT (*Query, Views and Transformation*) es un estándar propuesto por la OMG (*Object Management Group*) para la definición del proceso de transformación de modelo a modelo, basado en el estándar MOF (Meta Object Facility) para la descripción de la estructura y sintaxis de los metamodelos.

En [9] menciona 2 tipos de transformaciones:

Relaciones: especificaciones de transformaciones multidireccionales. No son ejecutables en el sentido de que son incapaces de crear o modificar un modelo. Permiten comprobar la consistencia entre dos o más modelos relacionados. Se utilizan normalmente en la especificación del desarrollo de un sistema o para comprobar la validez de un mapeo.

Mapeo: implementaciones de transformaciones. A diferencia de las relaciones, los mapeos son unidireccionales y pueden devolver valores. Un mapeo puede refinar una o varias relaciones, en cuyo caso el mapeo debe ser consistente con las relaciones que refina. A continuación se muestra la fig. 2 [2].

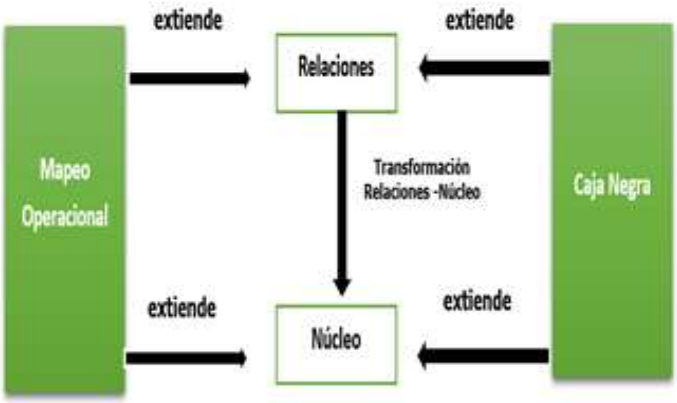


Fig. 2. Arquitectura del lenguaje QVT.

4. Propuesta

El objetivo es mostrar una transformación de modelo a modelo empleando el patrón de transformación de síntesis y utilizando los lenguajes de transformación de modelos ATL y QVT, se propone para ello un caso de estudio definido en el ambiente de desarrollo basado en eclipse (versión luna 4.4), recomendada por la comunidad de

eclipse para el desarrollo software dirigido por modelos, para esto es necesario la instalación de los plugin EMF (*Eclipse Modeling Framework*) para definir los modelos y metamodelos ECORE basados en el lenguaje XMI (*XML Metadata Interchange*), opción ATL en el framework, para la definición de las reglas de transformación en el lenguaje ATL y Model to Model Transformation para declarar las reglas de trasformación en QVT.

5. Justificación

El proceso de transformación de modelos basado en el campo laboral beneficia en la reducción de tiempo y costo durante el desarrollo de una aplicación, obteniendo como resultado una aplicación con un nivel de calidad sólido, por tal razón se pretende el promover la incorporación del enfoque en el proceso de ingeniería viendo más allá del modelado.

6. Aplicación del lenguaje de transformación de modelos ATL vs. QVT al caso de estudio

El caso de estudio basado en el patrón descrito, consiste en transformar un modelo que representa a una Agenda con el objetivo de obtener el modelo Cita, se define el modelo origen (modeloAgenda), el metamodelo origen (MMAgenda) y el meta-modelo destino (MMCita) ambos metamodelos son instancia del meta-metamodelo ECORE utilizado para la definición de metamodelos en Eclipse con el plugin EMF, el bloque llamado Agenda2Cita contiene las declaraciones y reglas para realizar el proceso de transformación ya sea en ATL y QVT, la siguiente figura muestra el esquema de transformación de modelo a modelo para el caso mencionado.

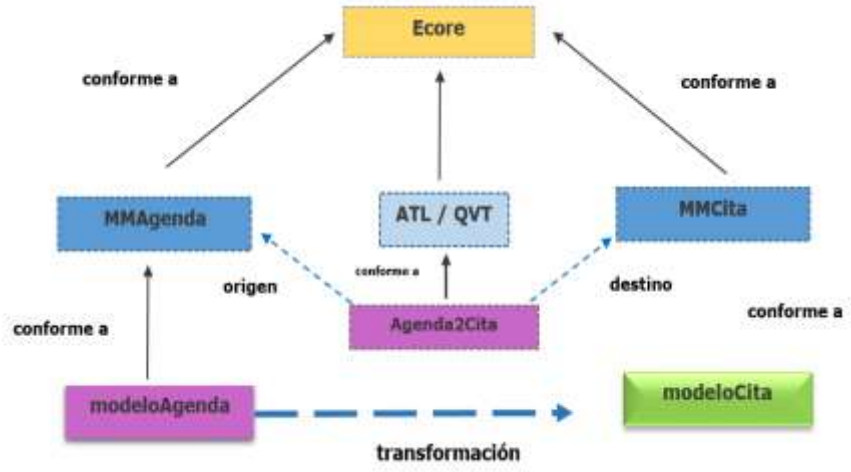


Fig. 3. Esquema de transformación de modelo a modelo.

A continuación se muestra en la figura 4 y 5 los elementos que componen al metamodelo Agenda y metamodelo Cita.

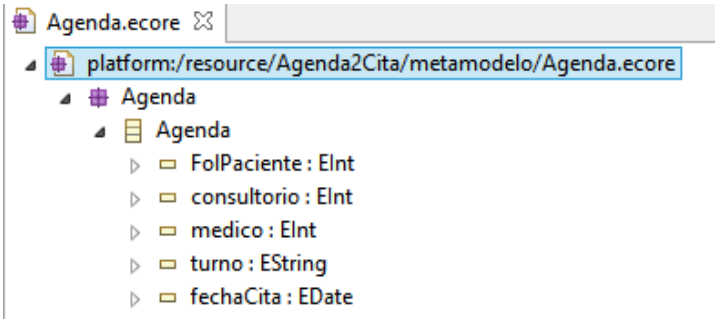


Fig. 4. Metamodelo Agenda.

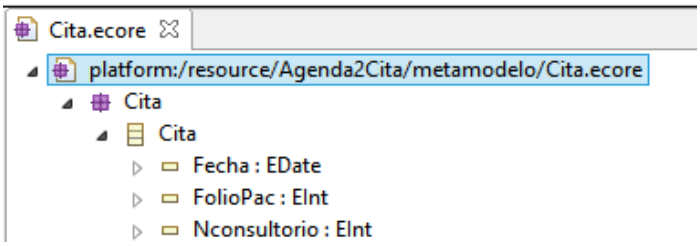


Fig. 5. Metamodelo Cita.

A partir de los metamodelos definidos se genera automáticamente código ECORE en formato XMI especificación utilizada para la definición y manipulación de metamodelos e intercambio de diagramas en UML. En el caso de la definición del modelo origen (modeloAgenda) se realiza de forma manual y de igual manera en formato XMI como se muestra en la siguiente figura.

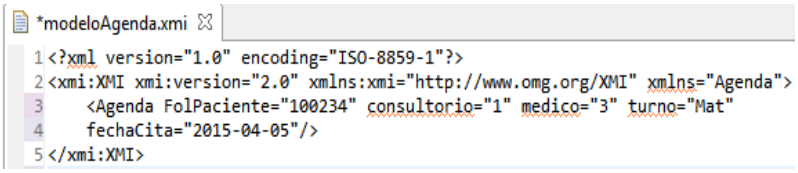


Fig. 6. Modelo Agenda.

El módulo en ATL permite la transformación entre modelos, declarando para ello las reglas de transformación basadas en sintaxis OCL (Object Constraint Language), como se observa en la siguiente figura en la línea 4 se define el nombre del módulo, posteriormente en la línea 7 se define el helper el cual corresponde a un método conforme a el paradigma orientada a objetos, este helper recupera un tipo de dato booleano para identificar si el elemento fechaCita de la clase Agenda se encuentra

definido, posteriormente en línea 14 se declara la regla para escribir los datos tales como fecha, folioPac y Nconsultorio en el modelo destino (modeloCita).

Agenda2Cita.atl

```
1  -- @path Agenda=/Agenda2Cita/Agenda.ecore
2  -- @path Cita=/Agenda2Cita/Cita.ecore
3
4  module Agenda2Cita;
5  create OUT : Cita from IN : Agenda;
6
7  helper context Agenda!Agenda def : getFecha() : Boolean =
8      if not self.fechaCita.oclIsUndefined() then
9          true
10     else
11         false
12     endif;
13
14 rule Agen2Cit {
15 from b : Agenda!Agenda (b.getFecha())
16 to
17 out : Cita!Cita (
18             Fecha <- b.fechaCita,
19             FolioPac <-b.FolPaciente,
20             Nconsultorio <- b.consultorio
21         )
22 }
```

Fig. 7. Módulo ATL.

El resultado de la transformación se muestra en la siguiente figura, dónde se obtuvo el modelo destino (modeloCita) con los datos recuperados como se estableció en la regla en el módulo ATL, generando un archivo en formato XMI.

*modeloCita.xmi

```
1 <?xml version="1.0" encoding="ISO-8859-1"?>
2 <xmi:XMI xmi:version="2.0" xmlns:xmi="http://www.omg.org/XMI" xmlns="Cita">
3   <Cita Fecha="2015-04-05T00:00:00.000-0600" FolioPac="100234" Nconsultorio="1"/>
4 </xmi:XMI>
```

Fig. 8. Modelo Cita.

Es posible obtener diferentes versiones del modelo destino (modeloCita) de acuerdo a las reglas establecidas en el módulo de cada uno de los lenguajes de transformación, para el siguiente ejemplo se restringe solo aquellos elementos de la clase Agenda asignados al turno matutino, partiendo del modelo origen (modeloAgenda) como se muestra en la siguiente figura.

*modeloAgenda.xmi

```
1 <?xml version="1.0" encoding="ISO-8859-1"?>
2 <xmi:XMI xmi:version="2.0" xmlns:xmi="http://www.omg.org/XMI" xmlns="Agenda">
3     <Agenda FolPaciente="100234" consultorio="1" medico="3" turno="Mat" fechaCita="2015-04-05"/>
4     <Agenda FolPaciente="100235" consultorio="2" medico="4" turno="Mat" fechaCita="2015-04-06"/>
5     <Agenda FolPaciente="100236" consultorio="3" medico="2" turno="Vesp" fechaCita="2015-04-07"/>
6     <Agenda FolPaciente="100237" consultorio="2" medico="4" turno="Vesp" fechaCita="2015-04-05"/>
7 </xmi:XMI>
```

Fig. 9. Modelo Agenda.

Por tanto se define en el módulo ATL, en línea 8 la declaración de un helper con el fin de validar el elemento turno de la clase Agenda con asignación a 'Mat' refiriéndose a Matutino y posteriormente en línea 18 se declara la regla para la escritura de los atributos del modelo destino (modeloCita) y así obtener la nueva versión del modelo destino.

*Agenda2Cita.atl

```
1  -- @path Agenda=/Agenda2Cita/Agenda.ecore
2  -- @path Cita=/Agenda2Cita/Cita.ecore
3
4  module Agenda2Cita;
5  create OUT: Cita from IN: Agenda;
6  helper context Agenda!Agenda def: OnlyMatutino(): Boolean =
7      if self.turno.equals('Mat') then
8          true
9      else
10         false
11     endif;
12
13 rule Agen2Cita {
14     from
15         b: Agenda!Agenda (
16             b.OnlyMatutino()
17         )
18     to
19         out: Cita!Cita (
20             Fecha <- b.fechaCita,
21             FolioPac <- b.FolPaciente,
22             Nconsultorio <- b.consultorio
23         )
24 }
```

Fig. 10. Módulo ATL.

La nueva versión del modeloCita se muestra en la siguiente figura, teniendo en cuenta que la escritura de un modelo destino, se basa de la existencia un modelo origen (modeloAgenda) de la figura 9.

modCita.xmi

```
1 <?xml version="1.0" encoding="ISO-8859-1"?>
2 <xmi:XMI xmi:version="2.0" xmlns:xmi="http://www.omg.org/XMI" xmlns="Cita">
3   <Cita Fecha="2015-04-05T00:00:00.000-0600" FolioPac="100234" Nconsultorio="1"/>
4   <Cita Fecha="2015-04-06T00:00:00.000-0500" FolioPac="100235" Nconsultorio="2"/>
5 </xmi:XMI>
```

Fig. 11. Modelo Cita.

A continuación se muestra en la siguiente figura el módulo en QVT en las líneas 1 y 2 se define el nombre y dirección de los metamodelos, después en la línea 4 se define el nombre del módulo y la correspondencia con los metamodelos, en las líneas 6 a la 8 se encuentra el main donde se recupera el conjunto de objetos de tipo Agenda que corresponde a la clase del metamodelo Agenda y finalmente en las líneas 9 a 13 se define el método Agend2Cita en el cual se indica la correspondencia del metamodeloAgendas tiene una clase Agenda y el metamodeloCitas su clase Cita y se recuperan los datos tales como fechacita, folpaciente y consultorio para ser escritos en el modeloCita.

QVTAgenda2Cita.qvto

```
1  modeltype Agendas uses 'http:///Agendas.ecore';
2  modeltype Citas uses 'http:///Citas.ecore';
3
4  transformation QVTAgenda2Cita(in agendaModelo:Agendas, out citaModelo:Citas);
5
6  main() {
7  agendaModelo.objects()[Agenda]->map Agend2Cita();
8  }
9  mapping Agendas::Agenda::Agend2Cita() : Citas::Cita {
10  Fecha:= self.fechaCita;
11  FolioPac := self.FolPaciente;
12  Nconsultorio :=self.consultorio;
13 }
```

Fig. 12. Módulo QVT.

El modelo destino (modeloCita) obtenido de la transformación se visualiza en estilo de árbol de navegación, semejante al de los metamodelos definidos en ECORE, indicando cual es la instancia del modelo destino (modeloCita) en este caso el metamodelo destino (metamodeloCita) y los datos correspondientes a las reglas establecidas en dicho módulo.

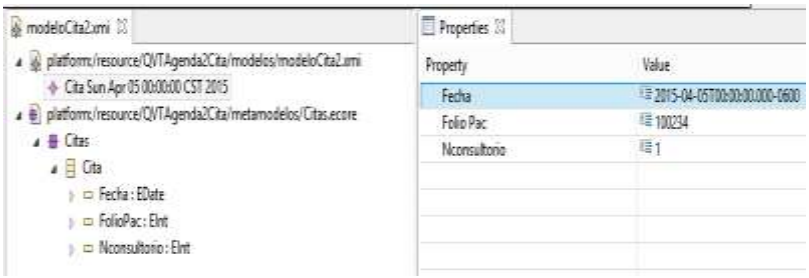


Fig. 13. Modelo Cita.

QVTAgenda2Cita.qvto

```
1  modeltype Agendas uses 'http:///Agendas.ecore';
2  modeltype Citas uses 'http:///Citas.ecore';
3
4  transformation QVTAgenda2Cita(in agendaModelo:Agendas, out citaModelo:Citas);
5
6  main()
7    {
8      agendaModelo.objects()[Agenda]->map Agend2Cita();
9    }
10 mapping Agendas::Agenda::Agend2Cita() : Citas::Cita {
11     if(self.turno.equalsIgnoreCase('Mat'))
12     {
13         Fecha:= self.fechaCita;
14         FolioPac := self.FolPaciente;
15         Nconsultorio :=self.consultorio;
16     }
17 }
```

Fig. 14. Módulo QVT.

El módulo en QVT que se muestra en la siguiente figura, es la continuidad al ejemplo planteado anteriormente, el cual requiere de la clase Agenda los elementos asignados al turno matutino, por tal razón en línea 11 se utiliza el método equalsIgnoreCase() para la comparación con la cadena 'Mat' y el contenido del elemento turno y así validar la escritura del modelo destino (modeloCita).

El nuevo modelo destino (modeloCita) contiene 4 elementos cita 2 de ellos son los que cumplieron con la restricción establecida en el módulo anterior, los 2 restantes tienen valores nulos por no cumplir con la condición definida y se puede visualizar que el modeloCita es una instancia del metamodelo destino (metamodeloCita).

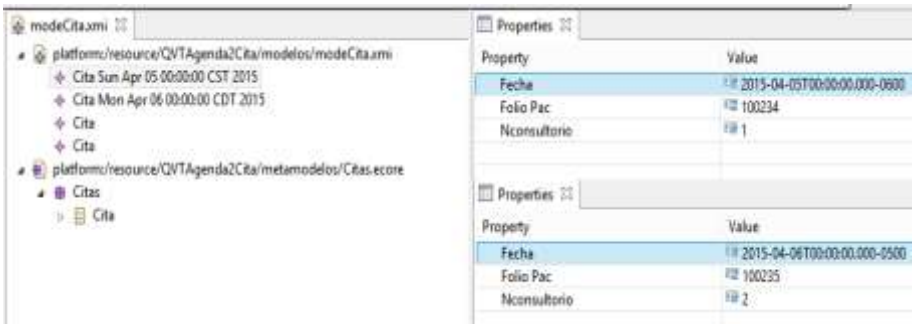


Fig. 15. Modelo Cita.

7. Conclusión

El análisis realizado revela algunas características de los lenguajes de transformación de modelos, el entorno de desarrollo para realizar la transformación de modelos, que el lenguaje ATL tiene su propia máquina virtual y compilador definidos por el lenguaje, que en QVT emplea dos maneras de realizar el proceso de transformación las cuales son; pruebas de caja negra y mapeo, la segunda se utilizó en el presente trabajo. A demás que la aplicación de los lenguajes de transformación a un caso de estudio en el campo laboral, beneficia en la reducción de tiempo y costo durante el desarrollo de una aplicación obteniendo como resultado una aplicación con un nivel de calidad solido por medio de la reutilización, también se menciona que el entorno de desarrollo en eclipse cuenta con un complemento llamado GMF el cual hoy en día resuelve problemas no triviales y una amplia documentación la cual incluye ejemplos básicos de transformación.

La mayoría de las herramientas para la transformación de modelos han sido desarrolladas como complementos del entorno de desarrollo en eclipse. Al desarrollarse sobre esta plataforma con varios años de utilización y desarrollo, se asegura cierta robustez, de manera global las herramientas hasta ahora disponibles otorgan características esenciales como el reconocimiento de la sintaxis y compilador, sin embargo aún existen funcionalidades sin implementar como por ejemplo la detección de errores en la codificación de transformación.

Finalmente en el proceso de transformación de modelo a modelo empleado al caso de estudio utilizando el patrón de transformación de síntesis, permite razonar que se obtuvo el modelo destino esperado en ambos lenguajes de transformación el cual incluyo un archivo en formato XMI, de ambos modelos destino el generado por el lenguaje QVT se obtuvo adicionalmente una vista tipo árbol de navegación, la cual facilita la visualización del resultado y especifica en la zona de espacio de nombres del archivo XMI a que metamodelo hace instancia el modelo generado ventaja no identificada en los resultados del lenguaje ATL.

Referencias

1. Trujillo, J. L., Espinoza, A. D.: Fundamental concepts of engineering headed by models and models of specific domain. Revista de Investigación de Sistemas e Informática, pp. 9–19 (2010)
2. Jouault, F., Kurtev, I.: On the Architectural Alignment of ATL and QVT. In Proceedings of the 2006 ACM symposium on Applied computing, pp. 1188–1195 (2006)
3. Jouault, F., Allilaire, F., Bézivin, J., Kurtev, I., Valduriez, P.: ATL: a QVT-like transformation language. In Companion to the 21st ACM SIGPLAN symposium on Object-oriented programming systems, languages, and applications, pp. 719–720 (2006)
4. Jouault, F., Allilaire, F., Bézivin, J., Kurtev, I.: ATL: A model transformation tool. Science of computer programming, vol. 72(1), pp. 31–39 (2008)
5. Büttner, F., Egea, M., Cabot, J., Gogolla, M.: Verification of ATL transformations using transformation models and model finders. In Formal Methods and Software Engineering, Springer Berlin Heidelberg, pp. 198–213 (2012)
6. Kurtev, I., van den Berg, K., Jouault, F.: Rule-based modularization in model transformation languages illustrated with ATL. Science of computer programming, vol. 68(3), pp. 138–154 (2007)
7. Tisi, M., Jouault, F., Fraternali, P., Ceri, S., Bézivin, J.: On the use of higher-order model transformations. In Model Driven Architecture-Foundations and Applications, Springer Berlin Heidelberg, pp. 18–33 (2009)
8. Jiménez, A., Vara, J. M., Bollati, V. A., Marcos, E.: Gestión de la trazabilidad en el desarrollo dirigido por modelos de Transformaciones de Modelos: una revisión de la literatura. In: XVI Jornadas de Ingeniería de Software y Bases de Datos–JISBD (2011)
9. Ferreira, M., García, F., Ruiz, F., Bertoa, M. F., Calero, C., Vallecillo, A., Mora, B.: Medición del software ontología y metamodelo. Departamento de Tecnologías y Sistemas de la Información, Castilla La Mancha (2006)

Clasificación semántica de textos no estructurados mediante un enfoque evolutivo

Eulalia T. Pacheco-Luz, Felipe Trujillo-Romero y Guillermo Juárez-López

Universidad Tecnológica de la Mixteca,
División de Estudios de Posgrado,
Huajuapan de León, Oaxaca, México

eulalia.pacheco@gmail.com,
{ftrujillo, gjuarezl}@mixteco.utm.mx

Resumen. En la actualidad, cerca del 90% de la información se encuentra plasmada tanto en documentos estructurados como no estructurados. Esto ha dado impulso a la investigación e implementación de diferentes algoritmos para el análisis y clasificación de textos de acuerdo a su orientación semántica. Por ello, en el presente trabajo se describe una manera de clasificación de textos no estructurados mediante el uso de algoritmos evolutivos. Esta técnica será utilizada en el análisis de documentos para determinar la clasificación de acuerdo al enfoque semántico de las palabras que contiene. Para este trabajo se analizaron textos pertenecientes a cuatro géneros literarios diferentes: ciencia-ficción, drama, comedia y terror. Se realizaron varias pruebas obteniendo un desempeño aceptable del sistema implementado.

Palabras clave: minería de textos, clasificación de textos, tesauro, algoritmos genéticos.

1. Introducción

La minería de textos es una disciplina que permite la extracción de información relevante de cantidades extensas de textos. Esto permite definir objetos y sus relaciones, revelando información semántica significativa. El tipo de texto puede ser obtenido de documentos estructurados, es decir que tengan un orden preestablecido en la organización de su contenido, o de no estructurados en los cuales el contenido o información no tiene ningún tipo de orden o estructura.

Adicionalmente la minería de textos se apoya en las técnicas de categorización de texto, procesamiento de lenguaje natural, aprendizaje automático, extracción y recuperación de la información. Existen diferentes opciones tanto comerciales como de software de código abierto. Ejemplo de ello es el software desarrollado por IBM SPSS [3], que es comercial pero ofrece una amplia y sólida variedad de soluciones en minería de textos. Dentro de las herramientas más populares de código abierto, se tiene al lenguaje de *R* [4] y a *RapidMiner* [5]. Este último posee una eficiente interfaz

de usuario, es altamente escalable debido a que maneja clústers y una programación orientada a bases de datos.

Los sistemas de minería de datos permiten el análisis léxico de los textos, especialmente la construcción automática de estructuras de clasificación y categorización que se codifica en forma de tesauros. Algunos ejemplos del uso de este tipo de sistemas se comentan a continuación. En [6] se utilizó RapidMiner (RM) para realizar el análisis de la similitud entre documentos de texto con los contenidos mínimos de los planes de estudio de las Licenciatura en Computación de la Universidad de San Juan. También se utilizó para procesar títulos bibliográficos de la biblioteca, midiendo la similitud sintáctica de los mismos con los contenidos de las diferentes carreras. Por su parte, en [7] se ha aplicado un coeficiente de legibilidad llamado Flesch-Kinkaid, para evaluar el contenido de los discursos del Rey de España y ha llegado a la conclusión de que la complejidad media de los mismos es bastante elevada. Siendo esta similar a la de un artículo científico, con un coeficiente en torno a 50. El estudio se realizó mediante el análisis de frecuencias de aparición de palabras, todo este estudio ha sido realizado con R y el uso de la librería de minería de textos *tm* [9]. En [10] Wei Zong *et al.* proponen un método para la categorización de texto el cual selecciona las características de los documentos basados en la medida de poder discriminativo y de la similitud entre las características usando para ello Maquina de Vectores de Soporte SVM (Support vector machine)

Por su parte, Yuen-Hsien Tseng [11] presenta una metodología de minería de textos especializados para el análisis de patentes mediante un enfoque de distribución de frecuencias de las palabras extraídas de los documentos analizados. En [12], se proponen dos nuevos algoritmos de agrupamiento de texto llamados: 1) Clustering Basado en Secuencias de Palabras Frecuentes (CFWS) y 2) Agrupación en Clústeres Basados en Significado de Secuencias de Palabras Frecuentes (CFWMS). En estos cada documento se reduce a sólo las palabras frecuentes para explorar la secuencia de palabra mediante la construcción de la estructura de un árbol del sufijo generalizado (GST). Finalmente comentamos el trabajo desarrollado por Zelai *et al.* [13] quienes proponen un sistema multiclasificador para categorización de documentos el cual utilizó el algoritmo de clasificación K-NN y un esquema de votación Bayesiano.

Por otro lado, los algoritmos genéticos (AGs) combinan las nociones de supervivencia, del más apto con un intercambio estructurado y aleatorio de características entre individuos de una población de posibles soluciones, conformando un algoritmo de búsqueda aplicado para resolver problemas de optimización en diversos campos [1, 8]. De tal forma, que los algoritmos genéticos se presentan como una herramienta de gran interés para extraer el significado de la información no estructurada de los datos de las organizaciones [2].

Además los algoritmos genéticos presentan ventajas con respecto a otras técnicas entre ellas: 1) no necesitan conocimientos específicos sobre el problema que intentan resolver, 2) operan de forma simultánea con varias soluciones en vez de trabajar de forma secuencial como las técnicas tradicionales, 3) cuando se usan para problemas de optimización-maximizar una función objetivo resultan menos afectados por los máximos locales que las técnicas tradicionales, 4) resulta sumamente fácil ejecutarlos

en las modernas arquitecturas masivas en paralelo y 5) usan operadores probabilísticos en vez de los típicos operadores determinísticos de las otras técnicas.

Entre las aplicaciones que tienen estos algoritmos relacionadas con la clasificación de documentos se puede mencionar a el agrupamiento de documentos y términos, la indización de documentos mediante el aprendizaje de los términos relevantes para describirlos por sus pesos y aprendizaje automático de los pesos de los términos proporcionados previamente por el usuario o de la composición completa de la consulta, incluyendo los términos y los operadores booleanos [14, 15, 16, 17, 18].

Si bien es cierto que los algoritmos genéticos están orientados a la optimización y pueden ser usados para la minería de textos, es necesario realizar las adecuadas modificaciones que permitan su empleo en la categorización de documento. Por ejemplo en el caso de clasificación de textos, cada cromosoma de la población está ligado a un tipo específico de clasificación de artículos como espacio de soluciones.

2. Metodología

Para lograr el descubrimiento de conocimiento, es necesario realizar un proceso que permita tratar la información y finalmente realizar la visualización de los resultados, en la figura 1, se muestra la metodología de la minería de textos que se desarrolló para ser utilizada en el presente trabajo.

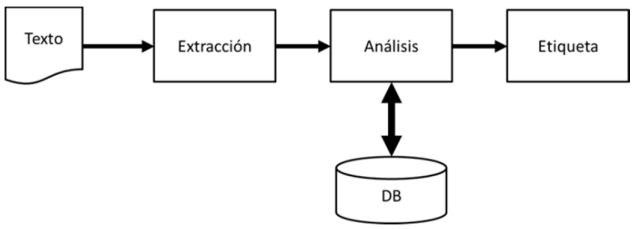


Fig. 1. Metodología de Minería de Textos.

En la misma figura 1, se puede visualizar como primera fase la definición del texto, es decir, se determinó el conjunto de documentos para el posterior análisis y clasificación, así también en esta fase se configuró el tesauro de términos especializados correspondiente a cuatro géneros literarios: ciencia-ficción, drama, comedia y terror. Todo el desarrollo de estudio en cuestión se basó en la norma ISO O 25964-1:2011, esta norma establece las directrices para el establecimiento y el desarrollo de tesauros monolingües [15], y define a tesauro como "*un vocabulario controlado y dinámico, compuesto por términos que tienen entre ellos relaciones semánticas y genéricas y que se aplica a un dominio particular del conocimiento*".

En la fase de extracción o pre-procesamiento se realizó operaciones de transformación sobre el documento, en información estructurada que facilitó su posterior análisis. El análisis de texto consistió en encontrar la secuencia de términos con el objetivo de encontrar patrones de lenguaje, las características que cumplieron dichos

términos se determinaron basándose en la técnica de algoritmos genéticos y la extracción de términos para su categorización.

Una vez terminada la fase de pre-procesamiento, se siguió con la fase de análisis, el cuál consistió en el descubrimiento de conocimiento, para lo cual se aplicó la fase de selección y mutación de algoritmos genéticos, para determinar cuáles son los cromosomas (términos) más representativos y que mayor información semántica proporcionan, a su vez se comparó con la información del tesauro predefinido.

La última fase, fue la visualización de los resultados, en la cual proporciona un ambiente para la exploración de los datos guiados para el usuario que sea lo más amigable posible. Las últimas tendencias presentan los resultados mediante graficas o páginas Web. Una vez obtenidos los conceptos, los términos o las tendencias, se pueden utilizar métodos automáticos de visualización o bien pueden interpretarse los resultados directamente. En este caso los resultados serán las gráficas de agrupación de términos para identificar la clasificación del documento analizado y determinar según la interpretación semántica a que área o campo de aplicación pertenece.

3. Análisis y discusión de resultados

Apoyándose en la metodología de minería de textos y el método de algoritmos genéticos, se desarrolló un programa en el entorno R, el cual permite analizar documentos en formato PDF (Portable Document Format), y clasificarlo en el campo de la literatura al cual pertenece, para la definición de la base de datos del tesauro se utilizó el gestor de base de datos MySql, en las secciones siguientes se describe el proceso realizado.

3.1. Pre-procesamiento

En primer lugar se delimitó el *corpus* a procesar que corresponde a cuatro géneros literarios; drama, ciencia ficción, terror y comedia, a modo de ejemplo se buscaron libros de cada uno de los géneros literarios. Seguidamente se realizó la definición del tesauro; esto incluye los nombres de los campos de especialización así como las palabras técnicas acordes a cada uno de los géneros que se están evaluando.

Para proceder a la extracción de la información, se procesó cada uno de los archivos a fin de convertirlo en archivo de texto plano, lo cual facilitara la extracción de cada uno de los términos.

Una vez que se cuenta con el archivo en texto plano, se procede a la aplicación de la minería de datos, para la limpieza y extracción de cada una de las palabras, para lo cual en primer lugar se instaló la librería *tm* en la herramienta R Studio, esta permitirá delimitar la matriz de términos a explorar, que al final es la matriz de cromosomas. Inmediatamente para realizar la limpieza de la matriz que contiene el texto en crudo, es necesario depurar términos, eliminar los números, los signos de puntuación, palabras auxiliares (artículos, pronombres, etc.), espacios en blanco y se convierte todo en minúsculas. Para realizar este proceso se guarda el texto extraído en un vector y

utilizando las palabras reservadas de la librería *tm* se realiza la limpieza de la información.

Consecuentemente se crea lo que se conoce como una matriz de términos del documento (*m x n*), donde *m* sería el número de descripciones a procesar y *n* sería el número de términos existentes en esos documentos. Los valores de la matriz sería el número de veces que cada fila contiene el término dado. Finalmente se guarda en un *dataframe* las palabras obtenidas para pasar a la etapa de procesamiento.

3.2. Procesamiento

En la etapa anterior, se llevó a cabo la depuración y delimitación de la población de palabras que son el conjunto de individuos a utilizar. Como siguiente paso se tiene que calcular la frecuencia de aparición de las palabras, conjuntamente se establece la conexión con la base de datos, con el objetivo de guardar en una tabla temporal, el conjunto de individuos que componen la población.

El tamaño de la población para este ejemplo es de 20 documentos, los cuales puede aumentar. Así también se definió un clúster de palabras, que es una colección de términos que semánticamente tienen relación entre sí, las cuales serán relacionadas a un campo de la literatura en específico.

El proceso de selección de la población inicial, se genera con los términos que se encuentran en el cuerpo del documento (Ec. 1), los cuales ya fueron guardados en la tabla temporal, cada registro es un cromosoma o individuo y cada uno de ellos está compuesto por un término, el valor del clúster al que pertenece el término, y una probabilidad de aparición asociada (Ec. 2). Es decir, se divide la adaptación de cada uno entre la suma de la de toda la población, y se asocia dicha distribución a una ruleta, dando más espacio en la misma a aquellos individuos que presenten mayor probabilidad de selección, en otras palabras, los mejor adaptados. La longitud el individuo siempre será variable, esto dependerá del texto que se esté analizando.

$$P_i = C_1, C_2, C_3, \dots C_n, \quad (1)$$

$$C_n = v_i, t_x, f, \quad (2)$$

donde:

P_i: Población inicial
$C_1 \dots C_n$: Cromosomas
v_i: Valor del clúster.
t_x: Término
f: Probabilidad.

Para obtener el campo *valor del clúster* (v_i), se realiza un comparación de cada una de las palabras del documento que resultaron del pre-procesamiento, con el tesauro especializado y se anota en el campo el valor del área al cual está asociado.

Pero para cumplir con el objetivo anterior, antes es necesario determinar la entropía de la población (Ec.3), ya que pueden existir diversas palabras con un elevado número de frecuencia, sin embargo semánticamente no aportan información relevante. Por ejemplo las palabras siguientes: que, luego, cuando, donde.

$$H(\mathrm{x}) = -x_i \log_2(x_i) \tag{3}$$

Una vez calculada la entropía se procede a le eliminación de aquellos términos superiores a la ponderación media de la entropía de la población (Ec. 4), así como los términos cuya frecuencia es igual a uno, puesto que tampoco proporcionan información relevante.

$$P = P_i - \left(\frac{H(x)}{n}\right) \qquad \left\{H(x) < \left(\frac{H(x)}{n}\right) \bigwedge P_i > 1\right\} \tag{4}$$

Para ejemplo se analizó el libro titulado "*Adiós Tierra*" del autor Álvaro Cotes Córdoba, del cual seleccionados los términos más representativos, la población para este ejemplo queda definida como se muestra en la Tabla 1.

Logrando de esta manera tener un cromosoma que está representado por los id de las áreas (v_i) a los que pertenece cada término. La representación de este cromosoma o individuo se puede observar en la Figura 2. Este individuo está compuesto por 18 genes.

Tabla 1. Términos representativos del libro "*Adiós Tierra*".

ID AREA (v_i)	TERMINO (t_x)	FRECUENCIA (f)
1	MERCURIO	4
1	TELESCOPIO	4
1	UNIVERSO	4
1	VIDA	4
4	PERSONAJE	4
1	ASTRO	3
1	DIOS	3
1	NASA	3
2	PROHIBIDO	3
1	AÑO	2
1	ATMOSFERA	2
1	DIGITAL	2
1	FISICA	2
1	FUERZA	2
1	PERIODO	2
2	SUDOR	2
3	RISAS	2
3	COLOR	2

1	2	.	.	.									14	15	16	17	18
1	1	1	1	4	1	1	1	2	1	1	1	1	1	1	2	3	3

Fig. 2. Representación del cromosoma a partir de la Tabla 1.

En este trabajo no se aplica el cruce ni la selección de los individuos debido a que todos pasan a la siguiente etapa que es la mutación del cromosoma. Para realizar la

mutación de los genes de los cromosomas, se utilizó el operador de mutación basado en el desplazamiento que describe Michalewizc [16]. Este proceso comienza seleccionando una subcadena de genes de un individuo al azar. Dicha subcadena se extrae del segmento y se inserta en un lugar aleatorio del individuo al cual se le extrae la subcadena. Por ejemplo, a partir del individuo de la figura 2 se toman los genes 14 al 17 que corresponden a la subcadena mostrada en la Figura 3.

1	1	2	3

Fig. 3. Subcadena tomada del individuo de la Fig. 2.

Después, se selecciona aleatoriamente un punto de inserción en el mismo individuo para insertar la subcadena extraída. En este caso fue el punto de inserción fue el gen número 6. Al insertar la subcadena esta reemplaza a los genes que existían anteriormente en el individuo quedando de la manera que se muestra en la Fig. 4.

También se analizó el libro "*El Sonido del Silencio*" del autor Heydee Cabrera, de su análisis se obtuvo la población que se muestra en la Tabla 2.

1	1	1	1	4	**1**	**1**	**2**	**3**	1	1	1	2	1	1	1	1	3

Fig. 4. Individuo mutado.

Tabla 2. Terminos representativos del libro "*El Sonido del Silencio*".

ID AREA	TERMINO	FRECUENCIA
2	SONIDO	56
2	LUNA	52
2	DEMONIOS	26
2	OSCURO	19
1	SILENCIO	19
1	DIOS	18
1	VIDA	14
2	MIEDO	13
1	TIEMPO	12
1	DIA	8
3	CONFUSIÓN	8
2	EXTRAÑO	6
1	GUSTO	4
1	SOL	4
2	CRIATURA	4
2	PELIGROSO	4

ID AREA	TERMINO	FRECUENCIA
1	AGUA	3
2	MUERTE	3
1	OIDO	3
2	SOMBRAS	3
1	SONAR	3
2	HORROR	3
2	PROHIBIDO	3
1	ESTRELLA	2
1	LUZ	2
2	PACTO	2
1	RAYOS	2
1	TIERRA	2
3	DISTURBIOS	2
1	VIDRIO	2
2	TRISTE	2

A partir de los datos obtenidos en la Tabla 2 se genera el cromosoma representativo que se muestra en la Fig. 5. Como se puede apreciar en la Fig. 2 este individuo es más grande (31 genes) que el generado para el ejemplo anterior.

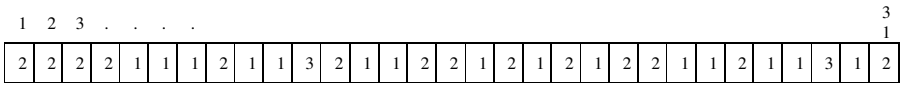


Fig. 5. Representación del cromosoma a partir de la Tabla 2.

A continuación se presenta la mutación del cromosoma mostrado en la Fig. 5. La subcadena extraída para realizar la mutación es [1, 1, 2, 1] y el cromosoma resultante es como se observa en la Fig. 6.

2	2	2	2	1	1	3	2	1	1	2	2	1	**1**	**1**	**2**	**1**	2	1	2	1	2	2	1	1	2	1	1	3	1	2

Fig. 6. Individuo mutado.

Esta mutación continuará hasta contar con un individuo cuyos elementos iguales sean continuos. Esto nos permitirá, posteriormente, dividir el individuo en subgrupos.

3.3. Análisis

Dentro de la información del conocimiento descubierto existen patrones que son más interesantes que otros, razón por la cual, se debe evaluar y determinar cuáles de ellos son los que le aportan conocimiento que le sean de mayor utilidad en la toma de decisiones. Por tal motivo, es necesario realizar la comparación semántica de términos para posteriormente identificar el género al cual pertenece, al contar con un tesauro predefinido es necesario realizar la comparación de términos para identificar cuáles son las palabras que van a definir el área o campo al cual pertenece el documento analizado, puesto que como anteriormente se explicó cada palabra está asociado a un área específica.

Sobre el último individuo cuya longitud siempre será variable para cada población, se determinará el género literario al que pertenece, esto será definido al formar subgrupos con los términos semejantes (v_i), así el subgrupo que mayor cantidad de términos contenga, se compara el número de v_i y se asocia al nombre del género literario que le pertenece, los cuales fueron determinados en el tesauro.

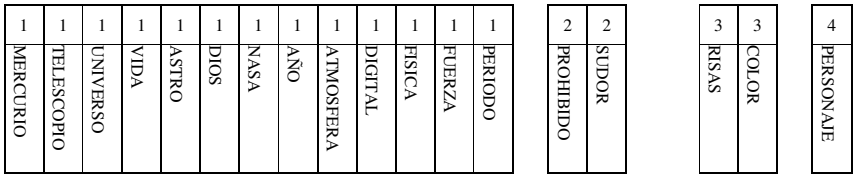


Fig. 7. Agrupación obtenida a partir de la evolución del individuo de la Fig. 2.

A partir de ello se obtuvo el resultado de la agrupación para el libro "*Adiós tierra*" el cual se muestra en la Fig. 7.

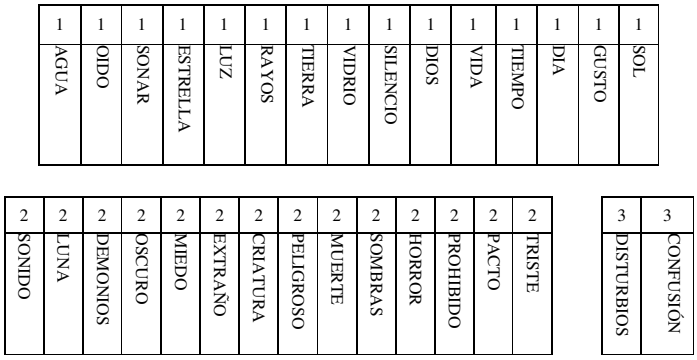


Fig. 8. Agrupación obtenida a partir de la evolución del individuo de la Fig. 5.

En el caso del análisis del libro "*El Sonido del Silencio*" la agrupación queda como se muestra en la Fig. 8.

De la agrupación obtenida se obtiene el nombre del área al cual pertenece el texto analizado mediante la extracción del máximo valor de los índices pertenecientes a los grupos. Esto se realiza mediante la expresión mostrada en la Ec. 5:

$$NA = \max_{i} \sum id\_Area. \tag{5}$$

3.4. Visualización de resultados

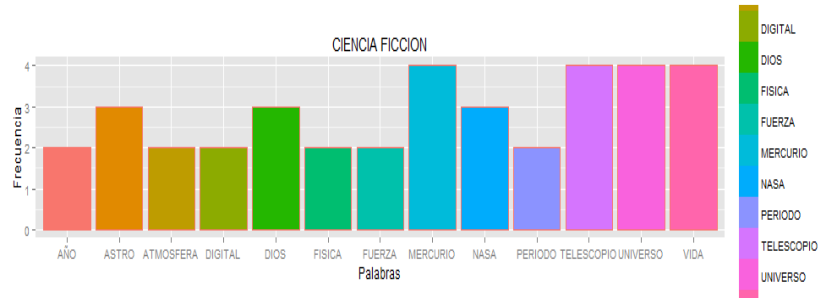
Finalmente, se presenta la visualización de resultados para el usuario final. Las gráficas que se muestran en la Fig. 9 contienen las palabras de mayor referencia semántica contenida en el tesauro, es decir, las palabras que se encontraron en los cromosomas y también se encontraron en el tesauro, así también se visualiza en el encabezado el nombre del campo al que pertenece, dicha información se obtiene seleccionando el id_area cuya mayor frecuencia se presenta en la tabla conceptos, puesto que pueden existir algunas palabras homógrafas.

Esta gráfica cambia de acuerdo al documento analizado puesto que el encabezado, contiene el nombre del área tomado de la base de datos.

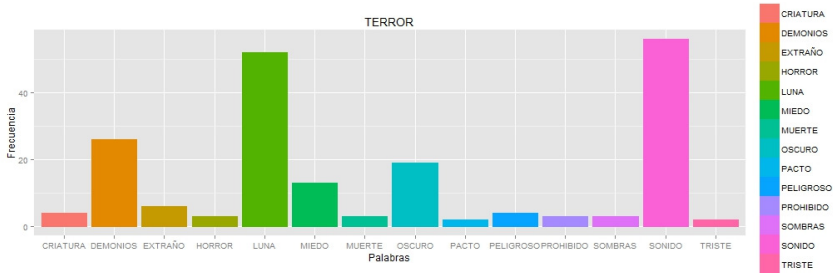
A continuación se muestra las gráficas de los dos ejemplos, donde se puede notar que las palabras que aparecen en cantidad no son elevadas pero sin embargo estos términos están asociados semánticamente y se puede notar la diferencia entre ambos géneros literarios. Además la cantidad de palabras, varía según el documento analizado, sin embargo el título de clasificación, que es el título de la gráfica, siempre se determinara en base al mayor grupo de palabras que se hayan encontrado.

Para la clasificación del documento y lo que nos da la certeza a que grupo pertenece al final es el grupo al que está asociado cada uno de los individuos que ya fue-

ron seleccionados, así por ejemplo, al final de todo el algoritmo tenemos la palabra "Peligroso" y dentro de su estructura de cromosoma está asociado al clúster 2, sabemos que la clasificación es de "Terror."



(a)



(b)

Fig. 9. Gráficas de conceptos.

4. Conclusiones

Actualmente existen varias herramientas para el análisis de documentos no estructurados, ninguno especializado en artículos literarios. Con el desarrollo de una herramienta utilizando el entorno de programación R y las técnicas de minería de datos con la lógica de algoritmos genéticos, se determinó que los algoritmos genéticos no pueden ser aplicados directamente en el proceso de análisis de textos, es necesario realizar algunos cambios al algoritmo para adaptarlo, como lo es el proceso de selección pues además de la probabilidad se toma en cuenta la frecuencia de aparición.

Además se descubrió que la mayor parte de los términos contenidos en los documentos lo podemos considerar basura puesto que no aportan información relevante, y el campo de clasificación es determinado por el mayor conjunto de palabras que se forman en base a su contenido semántico.

Como un trabajo futuro se pretende extender este trabajo para obtener un sistema que sea capaz de clasificar un mayor número de géneros orientándolo hacia el análisis de textos científicos.

Referencias

1. Holland, J. H.: Adaptation in Natural and Artificial Systems. University of Michigan Press, Ann Arbor. Republished by the MIT press (1992)
2. Moore, C.: Diving into data, Info world. http://www.infoworld.com/article/02/10/25/021028feundata_1.html
3. Sitio Web de Soluciones y software de analítica predictiva (Software SPSS). IBM, http://www-01.ibm.com/software/mx/analytics/spss
4. Venables, W. N., Smith, D. M.: the R Core Team. Introduction to R, version 3.1.3, R Core Team (2015)
5. Sitio web de RapidMiner. RapidMiner Studio. https://rapidminer.com
6. Gutiérrez Mag, L.: Pertinencias de planes de estudio de carreras de informática con normativas establecidas por CONEAU. Universidad Nacional de San Juan (2013)
7. Serrano Sánchez, A.: Minería de textos o cómo analizar los discursos del Rey. Universidad Francisco de Victoria. http://ti3.ceiec.es/mineria-de-textos-o-como-analizar-los-discursos-del-rey
8. Goldberg, D.: Genetics Algorithms in Search, Optimization and Machine Learning. Addison Wesley (1989)
9. Feinerer, I., Hornik, K., Meyer, D.: Text Mining Infrastructure in R, Journal of Statistical Software, vol. 25 (5), pp. 1548–7660 (2008)
10. Zong, W., Wu, F., Chu, L.K., Sculli, D.: Discriminative and Semantic Feature Selection Method for Text Categorization. International Journal of Production Economics, available online, ISSN 0925-5273 (2015)
11. Tseng, Y., Lin, C., Lin, Y.: Text Mining Techniques for Patent Analysis. Information Processing and Management, vol. 43 (5), pp. 1216–1247 (2007)
12. Li, Y., Chung, S. M., Holt, J. D.: Text Document Clustering based on Frequent Word Meaning Sequences. Data & Knowledge Engineering, vol. 64 (1), pp. 381–404 (2008)
13. Zelai, A., Alegria, I., Arregi, O., Sierra, B.: A Multiclass/Multilabel Document Categorization System: Combining Multiple Classifiers in a Reduced Dimensión. University of the Basque Country, UPV-EHU, Computer Science Faculty, Euskal-Herria, Spain (2011)
14. Mukherjee, I., Al-Fayoumi, M., Mahanti, P.K., Jha, R., Al-Bidewi, I.: Content Analysis based on Text Mining using Genetic Algorithm. 2nd International Conference on Computer Technology and Development (ICCTD 2010), pp. 432–436 (2010)
15. Khalessizadeh, S. M., Zaefarian, R., Nasseri, S.H., Ardil, E.: Genetic Mining: Using Genetic Algorithm for Topic based on Concept Distribution. Proceedings of World Academy of Science, Engineering and Technology, vol. 13, ISSN 1307-6884 (2006)
16. Yolis, E., Britos, P., Sicre, J., Servetto, A., García-Martínez, R., Perichinsky. G.: Algoritmos genéticos aplicados a la categorización automática de documentos. IX Congreso Argentino de Ciencias de la Computación (CACIC), La Plata, Argentina (2003)
17. Bharadwaj, D., Shukla, S.: Text Mining Technique using Genetic Algorithm. Proceedings on International Conference on Advances in Computer Application (ICACA) (2013)
18. Shivani, P., Gandhi, P.: A Detailed Study on Text Mining using Genetic Algorithm, International Journal of Engineering Development and Research (IJEDR), ISSN:2321-9939, vol. 1 (2), pp. 108–113 (2014)
19. Sitio Web de la Organización internacional para la estandarización. http://www.iso.org/iso/home/store/catalogue_ics/catalogue_detail_ics.htm?csnumber=53657
20. Michalewicz, Z.: Genetic Algorithms + DataStructures = Evolution Programs. Springer-Verlag, BerlinHeidelberg (1992)

Clasificación automática de la orientación semántica de opiniones mediante características lingüísticas

Alonso Palomino Garibay y Sofía N. Galicia-Haro

Facultad de Ciencias, UNAM, México, D.F.

alonsop@ciencias.unam.mx, sngh@fciencias.unam.mx

Resumen. En este trabajo examinamos el problema de clasificar la orientación semántica de opiniones de productos comerciales, en idioma Español. Exploramos las características de la colección de opiniones extraídas de Internet y estudiamos el uso de bigramas de afirmación como características de entrenamiento para un método supervisado. Analizamos la combinación de estas características y medimos su desempeño al utilizarse en el método de Máquinas de soporte vectorial. Los resultados se equiparan con el estado del arte para opiniones en español.

Palabras clave: máquinas de soporte vectorial, orientación semántica, bigramas morfosintácticos, función kernel.

1. Introducción

La enorme cantidad de comentarios de libre acceso en la Web, para productos y servicios, ha permitido que esas opiniones sean un recurso valioso para tomar decisiones. Las personas interesadas en dichos productos pueden realizar compras basadas en información de otros clientes. Por su volumen, resulta difícil su manejo para un análisis manual, ya que requiere mucho tiempo escoger y comparar opiniones sobre productos objetivo del interés de los compradores. Adicionalmente, dentro de las opiniones, puede haber oraciones que expresen tanto juicios positivos como negativos sobre características del producto, lo que hace más complicada su utilización.

La minería de opiniones intenta clasificar opiniones de forma automática, en función de lo que expresa cada autor. Esta área combina técnicas del Procesamiento de lenguaje natural y de la Minería de textos, y comprende actualmente una gran cantidad de tareas, unas con mayor desarrollo que otras. Por ejemplo, [1] consideraron: identificación de opiniones, polaridad del sentimiento, resumen de la orientación de la opinión; [2] consideró: análisis de sentimiento en oraciones de comparación, detección de SPAM, detección de opiniones que no evalúan, detección de opiniones engañosas.

Una de las principales tareas consiste en determinar la polaridad de las opiniones del documento completo, de oraciones, o de características consideradas. El resultado es clasificar las opiniones en positivas o negativas. Los principales enfoques para resolver este problema corresponden a la clasificación más amplia del aprendizaje mediante

computadora: métodos supervisados y no supervisados. Aunque este problema ha sido estudiado extensamente, la clasificación por polaridad sigue siendo un reto para el procesamiento de lenguaje natural.

En este trabajo aplicamos métodos supervisados a una colección de opiniones en español para lavadoras. Tratándose de productos tan básicos de la vida actual, clasificar estas opiniones es un reto porque los autores de estas opiniones utilizan lenguaje coloquial principalmente, incluyen pasajes anecdóticos y no ponen atención en introducir los signos de puntuación necesarios. Primero seguimos la idea de [3] que considera bigramas morfosintácticos como características para un método no supervisado. Aquí utilizamos bigramas morfosintácticos como características de métodos supervisados, principalmente Máquinas de soporte vectorial (SVM).

El artículo se organiza como sigue: en la segunda sección se presentan trabajos relacionados, enseguida se presenta una descripción de los materiales empleados y del conocimiento lingüístico considerado. En la cuarta sección se describe la herramienta utilizada para aplicar métodos supervisados y los parámetros que se consideraron. En la quinta sección se describen las condiciones de los experimentos y en la sexta sección presentamos los resultados correspondientes. Finalmente presentamos nuestras conclusiones y planteamos las líneas de trabajo futuro.

2. Trabajos relacionados

Desde que la Minería de opiniones se ha asumido como un reto en el Procesamiento de lenguaje natural, una de las tareas en la que más esfuerzos se ha aplicado es la clasificación de la polaridad de un texto, básicamente si es positiva o negativa. Como lo establecen [1], gran parte del trabajo sobre la clasificación de polaridad se ha llevado a cabo en el contexto de opiniones, pero la entrada a un clasificador de polaridad no es necesariamente siempre una opinión estrictamente.

En ese contexto, las opiniones positivas y negativas son a menudo evaluativas (*gustar*, *disgustar*) pero hay otros problemas en su interpretación, y aunque no proporcionan una definición clara de lo que debe considerarse problemas en la clasificación de polaridad, consideran que: (a) en la determinación de la polaridad de los textos de opiniones, donde los autores expresan explícitamente su sentimiento a través de frases como "*este portátil es grande"*, la información objetiva como "*larga duración de la batería*" se utiliza a menudo para ayudar a determinar el sentimiento completo; (b) la tarea de determinar si un pedazo de información objetiva es buena o mala todavía no es lo mismo que clasificarla en una de las varias clases; y (c) la distinción entre información subjetiva y objetiva puede ser sutil.

Desde el inicio se han considerado modificadores, como los adjetivos. El trabajo de [4] para predecir la orientación semántica de adjetivos considera un algoritmo de agrupamiento que separa los adjetivos en grupos de diferentes orientaciones, basado en la correlación entre características lingüísticas y la orientación semántica. Por ejemplo, adjetivos unidos con la conjunción "*y*" corresponden a adjetivos de la misma orientación como *justo y legítimo*, en oposición a *tranquilo pero flojo*.

Los principales enfoques que se han propuesto desde entonces para resolver la clasificación de la polaridad corresponden a la clasificación de aprendizaje automático: métodos no supervisados y métodos supervisados. Entre los primeros, un trabajo importante es el de [3], donde se determinó la orientación semántica de bigramas del texto de opiniones en inglés. Esta orientación se usó para calcular la orientación de oraciones y del texto completo de una colección de 410 opiniones de *Epinions*. En el segundo enfoque, el trabajo de [5] analiza los resultados de tres métodos: Naive Bayes, Entropía máxima y SVM en la clasificación de polaridad de opiniones de cine y concluyen que no funcionan tan bien como en categorización de textos por tema. Los autores incluyen distintas características lingüísticas de acuerdo al método; en general consideran: palabras solas, secuencias de dos palabras, categorías gramaticales, posición de la palabra (en las opiniones de películas, en particular, pueden empezar con una declaración general de sentimiento).

En cuanto a trabajos desarrollados para clasificar la orientación semántica de opiniones en español, el trabajo de [6] utilizó el corpus MuchoCine [7], de críticas de cine, para un estudio experimental de combinación de métodos supervisados y no supervisados en un corpus paralelo inglés-español. Tradujeron al inglés el corpus en español y combinaron los dos sistemas obtenidos al aplicar métodos supervisados a cada uno de los corpus con la aplicación de SentiWordNet[1] al corpus en inglés. En [8] los autores proponen un método no supervisado basado en el análisis sintáctico de dependencias para determinar la orientación semántica de textos, donde asignan un valor de orientación semántica a las siguientes construcciones sintácticas: negación, oraciones adversativas subordinadas e intensificadores, además de incluir otras características de las palabras mismas y del texto en general. En el trabajo de [7] compilaron un corpus de casi 4 mil opiniones de cine, y aplicaron el método no supervisado de [3] a una colección de 400 opiniones, 200 positivas y 200 negativas.

3. Materiales empleados y conocimiento lingüístico considerado

3.1. Corpus de opiniones

Para este trabajo usamos la colección de opiniones de [9]. La colección fue compilada automáticamente del sitio *ciao.e*s y consta de 2800 opiniones de lavadoras. El tamaño promedio por archivo en lexemas es de 345. El número total de lexemas de la colección es de 854,280. La colección total fue anotada con información de lema y categorías gramaticales (*part of speech* en Inglés) utilizando FreeLing [10], una biblioteca de código abierto.

De la colección total de opiniones en español, extrajimos un subconjunto significativo de opiniones diferentes: 2598 opiniones. No eliminamos las opiniones que claramente son anuncios de empresas de mantenimiento (SPAM) ya que tanto este tipo de textos como las opiniones pagadas por fabricantes aparecen en cualquier colección de opiniones de productos.

[1] SentiWordNet es un recurso léxico que asigna a cada uno de los sentidos de las palabras de WordNet tres valores de orientación (positiva, negativa y objetiva)

Utilizamos esta colección para entrenar un estimador cuyo objetivo es la predicción. Es decir, determinar qué tan bueno es un producto en base a la orientación semántica de las opiniones, y el puntaje de los usuarios que corresponden a: malo (una estrella), regular (dos estrellas), bueno (tres estrellas), muy bueno (cuatro estrellas) o excelente (5 estrellas).

Las opiniones tienen tanto errores gramaticales como ortográficos y de puntuación, pero debido a la diversidad de errores decidimos no aplicar métodos de corrección. Freeling es capaz de dar una categoría gramatical correcta aún con errores de ortografía aunque marca como nombre propio sustantivos comunes que no están precedidos por punto y comienzan con mayúscula.

Las características de esta colección en cuanto a número de opiniones por puntaje se presentan en la Tabla 1. Como se observa y como podría esperarse de opiniones de aparatos electrodomésticos cuyo uso es tan generalizado por su gran utilidad, las opiniones favorables son mayores en una proporción de 6:1 entre opiniones positivas y negativas.

Tabla 1. Corpus de reseñas de artículos comerciales.

Opiniones	**Detalles**	
Clase	*Numero de instancias*	*Estrellas*
Excelente	1190	5
Muy bueno	838	4
Bueno	239	3
Regular	127	2
Malo	204	1

En el área de aprendizaje automático se ha abordado el problema del desequilibrio de clases en la cantidad de ejemplos de entrenamiento para cada una de ellas y las soluciones que se han dado se han clasificado por [11] de la siguiente manera:

- Modificación del algoritmo. Este enfoque está orientado a la adaptación de métodos de aprendizaje para que sean más sensibles a los problemas de desequilibrio de clases, por ejemplo [12]
- Asignación de pesos distintos a los ejemplos de entrenamiento, introduciendo diferentes costos a ejemplos positivos y negativos. Este enfoque considera costos más altos para la clasificación errónea de la clase de mayoría respecto a la clase minoritaria durante el entrenamiento de clasificadores, por ejemplo: [13]
- Muestreo heterogéneo de datos, incluye bajo-muestreo, sobre-muestreo y métodos híbridos. El bajo-muestreo elimina instancias de clases mayoritarias mientras que el sobre-muestreo crea nuevas instancias de la clase minoritaria. Los métodos híbridos combinan los dos métodos anteriores, por ejemplo: [14]

La herramienta que empleamos en este trabajo permite hacer sobre-muestreo, sin embargo lo hemos dejado para un trabajo futuro, considerando que la proporción de desequilibrio de nuestra colección no es un caso que requiera métodos específicos de balanceo. Nos basamos en los resultados obtenidos por [15], donde para una colección

con similar proporción de desequilibrio no obtiene mejoras con diferentes métodos de balanceo.

3.2. Bigramas afirmativos

La clasificación del sentimiento o de polaridad en opiniones se ha trabajado en diferentes dominios, por ejemplo: reseñas de películas, comentarios de productos, para retroalimentar clientes, como opiniones de hoteles y servicios turísticos [1]. Mucho del trabajo se ha realizado con algoritmos de aprendizaje de máquina. El trabajo realizado con métodos no supervisados, es decir, métodos que no cuentan con ejemplos previamente anotados con la clasificación que se quiere aprender se ha basado en el conteo de términos positivos y negativos, determinando automáticamente si el término es positivo o negativo.

[3] determinó la orientación semántica de una opinión mediante un algoritmo que primero extrae bigramas, es decir, secuencias de dos palabras donde una de ellas es un modificador. Enseguida toma cada bigrama para realizar una búsqueda en la Web empleando el operador NEAR de AltaVista para encontrar cuántos documentos tienen ese bigrama cerca de un término positivo (*excellent*) y de un término negativo (*poor*). El puntaje para los dos conjuntos se realiza mediante la medida de información mutua puntual (Pointwise Mutual Information, PMI). La diferencia de puntuación para los dos conjuntos se utiliza para determinar la puntuación para la orientación semántica (SO-PMI), que da como resultado el grado en que cada (bigrama, término) es positivo o negativo.

El puntaje PMI de dos palabras w_1 y w_2 se obtiene mediante la probabilidad de que las dos palabras aparezcan juntas dividida por las probabilidades de cada palabra en forma individual:

$$\mathrm{PMI}(w_1, w_2) = log[P(w_1, w_2)/(P(w_1)P(w_2))] \quad (1)$$

Considerando el número de hits (resultados obtenidos de la Web para la búsqueda), el cálculo de la orientación semántica fue realizado de la siguiente manera:

$$\mathrm{SO(frase)} = \log_2 \frac{\mathrm{hits}(frase\ \mathrm{NEAR}\ excellent)\ \mathrm{hits}(poor)}{\mathrm{hits}(frase\ \mathrm{NEAR}\ poor)\ \mathrm{hits}(excellent)} \quad (2)$$

La orientación semántica de bigramas fue utilizada para determinar orientación semántica de oraciones y opiniones completas. Turney tomó 410 comentarios de *epinions.com*. Los resultados oscilaron entre el 84 % para las revisiones de automóviles y el más bajo de 66 % para las críticas de películas.

Si los bigramas morfosintácticos son una buena característica para métodos no supervisados, suponemos que para métodos supervisados incluso podrían ser mejores. Así que en este trabajo, consideramos los siguientes bigramas morfosintácticos como características para el entrenamiento del método supervisado:

1. Sustantivo - adjetivo
2. Verbo - adverbio
3. Adverbio - adjetivo
4. Adjetivo - adverbio

A diferencia de [3] no aplicamos restricción sobre la palabra que sigue a cada bigrama. En el trabajo de [7] aplicaron el método no supervisado de [3], y por la desproporción obtenida de sus resultados (35,5% para las negativas y 91,5% para las positivas) buscaron un umbral de forma supervisada, utilizando el 80% del corpus de 400 críticas para entrenamiento y el 20% restante para evaluación. En este trabajo empleamos las SVM.

Estos bigramas que llamamos morfosintácticos, no corresponden a compuestos obtenidos de un analizador sintáctico. Escribimos un programa que obtiene de la colección completa de opiniones de lavadoras todas las secuencias de dos palabras cuyas categorías gramaticales cumplen los patrones antes indicados. En el caso sustantivo–adjetivo el programa que extrae estos bigramas comprueba la concordancia en género y número. Para todos los bigramas se extraen no las palabras sino los lemas, esto permite agrupar diversas formas en una sola característica. Por ejemplo: *prenda vaquera* y *prendas vaqueras*, *lavadora nueva* y *lavadoras nuevas*, se agrupan en un solo bigrama para cada par.

Consideramos los bigramas adverbio-adjetivo y adjetivo–adverbio porque aun cuando en español es más común la forma adverbio-adjetivo encontramos que la forma inversa está presente en algunas opiniones de esta colección.

4. Herramienta de clasificación

Una ventaja de las SVM es la gran variedad de funciones kernel que pueden usarse para la clasificación. Es decir, se puede generalizar aun en presencia de muchas características con un amplio margen, usando funciones de nuestro espacio de hipótesis [16]. Lo anterior infiere el uso de heurísticas como *Grid search* para la optimización de un conjunto de hiper-parámetros para un estimador en el espacio de un algoritmo de aprendizaje.

Para una tarea de clasificación es necesario separar los datos entre un conjunto de entrenamiento y un conjunto de prueba, en nuestro caso separamos el corpus de opiniones en 70 % para entrenamiento y 30 % para prueba. Cada ejemplo o instancia se asocia a una clase, categoría o etiqueta, es decir el 70 % de los datos de entrenamiento fueron etiquetados con la clase correspondiente (i.e. *muy buena*) y el 30 % de los datos no tiene etiqueta.

4.1. Preprocesamiento de datos

Una de las ventajas de usar un lenguaje de programación de propósito general como *Python* y herramientas de aprendizaje automático como *scikit learn* es la gran cantidad de bibliotecas robustas para implementar distintos métodos y la manipulación de datos. Para obtener las estructuras lingüísticas definidas, se generaron distintas expresiones regulares para empatar esos patrones. Posteriormente, el programa desarrollado generó archivos del tipo *csv* con los bigramas propuestos en la sección anterior. Decidimos utilizar este formato porque es ampliamente usado por la comunidad de aprendizaje

automático; por ejemplo: la gran mayoría de los datos de UCI *machine learning repository* se encuentran en formato *csv*[2].

Una vez que se generaron los archivos *csv*, se procesaron con *pandas*, una biblioteca para el manejo y análisis de datos de forma eficiente. Pandas ofrece estructuras de datos y operaciones eficientes para tablas relacionales, grandes cantidades de datos y conjuntos de datos etiquetados [17]. En este trabajo utilizamos algunas funciones de esta biblioteca para leer el corpus y presentárselo al clasificador. Posteriormente estos datos fueron separados bajo un esquema de validación cruzada con la finalidad de prevenir el sobreajuste del modelo.

4.2. Sistema

Para resolver este problema de clasificación de opiniones en idioma Español decidimos usar un algoritmo supervisado. La clasificación se hizo mediante SVM para el caso multiclase, dadas sus fuertes bases teóricas y porque las SVM son algoritmos de aprendizaje que tienen la capacidad de aprender independientemente de la dimensionalidad del espacio de características. El objetivo de las SVM es producir un modelo basado en los datos de entrenamiento que prediga las clases o categorías de un conjunto nuevo de instancias, mediante la generación de un hiperplano en un espacio de *n*-dimensión.

Por [16] se sabe que las SVM funcionan bien para clasificar texto, básicamente porque cuando se clasifica texto se trabaja con espacios de dimensión alta, con pocas características irrelevantes, con representaciones vectoriales dispersas y porque la mayor parte de problemas de clasificación de texto son linealmente separables. Todos estos argumentos son fundamentos teóricos que nos permiten saber que es posible clasificar texto con esta clase de algoritmos supervisados.

5. Experimentos

En los siguientes experimentos comparamos el rendimiento de las máquinas de soporte vectorial para el caso multiclase usando distintas funciones kernel, luego comparamos contra un clasificador aleatorio que genera predicciones respetando la distribución de los datos de la clase del conjunto de entrenamiento. El entrenamiento de la máquina de soporte vectorial fue realizado empleando la herramienta *scikit-learn* una biblioteca de propósito general que implementa una gran variedad de algoritmos de aprendizaje automático y que al igual que otras bibliotecas incorpora o envuelve a la biblioteca de C++ LibSVM [18].

5.1. El truco del kernel

En aprendizaje automático los métodos kernel son una clase de algoritmos para analizar patrones, frecuentemente usados en máquinas de soporte vectorial. Las SVM

[2] http://archive.ics.uci.edu/ml/

necesitan funciones kernel debido a que en espacios de baja dimensión presentan dificultades para clasificar datos. Por ejemplo en R^2 la recta que separa dos conjuntos de vectores se vuelve más compleja si agregamos clases y vectores, por lo tanto necesitamos mapear estos datos a espacios de mayor dimensión donde podamos crear de forma más sencilla hiperplanos que separen instancias o vectores de diferentes clases.

Las funciones kernel se usan en muchos algoritmos de aprendizaje para dar una conexión entre linealidad y no linealidad, es decir para que los algoritmos clasifiquen *n* conjuntos de datos o vectores de forma eficiente. Si la cota entre C clases de vectores es muy cercana donde la cota es la distancia entre los vectores, el algoritmo tardaría mucho tiempo en converger a una solución y aunque converja a alguna, esta no sería la mejor.

En lugar de tomar la costosa y difícil ruta de clasificar los datos en dimensiones bajas generando una curva que separe las clases, las funciones kernel mandan o mapean los datos representados como vectores a espacios de dimensión mayor con la intención de que esos datos sean separados de forma más sencilla, más aun, estas funciones podrían mapear los datos a espacios de dimensión infinita, por lo que no hay restricciones en la dimensión a la cual se mapean los datos. Esto también es conocido como el truco del kernel.

Si los datos vienen estructurados adecuadamente, el algoritmo fácilmente genera una separación o hiperplano entre las distintas clases de datos. El truco del kernel es una herramienta útil que puede aplicarse a algún algoritmo de aprendizaje automático que dependa de un espacio con producto interno entre n vectores. Cuando mapeamos texto o datos a espacios de dimensión mayor o infinita, tenemos que usar algoritmos que usen el producto interno de vectores en espacios de dimensión alta y que sean capaces de generar un hiperplano. Para calcular el producto interno de vectores en espacios de dimensión infinita podemos usar una función kernel que calcule el producto interno directamente usando vectores que viven en espacios de baja dimensión.

Con el fin de probar estos algoritmos para el análisis de patrones usamos los kernels más usados en máquinas de soporte vectorial y evaluamos su comportamiento.

- Kernel de función de base radial. Este kernel mapea los datos a un espacio de dimensión infinita

$$k(x, y) = exp(\gamma|| x \neq y||^2) \quad (3)$$

- Kernel polinomial: Esta función kernel es útil para problemas en los cuales los datos están normalizados:

$$k(x, y) = (\alpha x^T y + c)^d \quad (4)$$

- Kernel lineal: Es la función kernel más simple. Es calculada por el producto interno Èx, yÍ más una c constante:

$$k(x, y) = x^T y + c \quad (5)$$

5.2. Evaluación

El sistema se configuró con tres funciones kernel distintas (i.e. lineal, polinomial, función de base radial) y se fueron agregando las características propuestas en la sección 3 para posteriormente evaluar la calidad de la clasificación mediante las siguientes métricas:

Exactitud: En clasificación multiclase, esta métrica calcula el subconjunto de la precisión del conjunto de etiquetas predichas para una muestra que exactamente corresponden al conjunto de etiquetas del conjunto de entrenamiento.

F1-score: Puede ser interpretado como un promedio balanceado entre la precisión y el recall, una F1-score alcanza su mejor valor en 1 y su peor valor en 0. La contribución relativa de precisión y recall al F1-score son iguales.

Score: Se refiere a la media de la precisión, dados los datos y etiquetas de prueba.

Recall: Es la capacidad que tiene un estimador de encontrar todas las muestras positivas. El recall es la relación $t_p/(t_p + f_n)$ donde t_p es el número de verdaderos positivos y f_n es el número de falsos negativos.

Precisión: Intuitivamente podemos decir que es la capacidad que tiene un estimador de no etiquetar como positiva una muestra que es negativa. Precisión es la relación $t_p/(t_p + f_p)$ donde f_p es el número de falsos positivos.

Pérdida de Hamming: Es la fracción promedio de etiquetas incorrectas en porcentaje. Nótese que la pérdida Hamming es una función de pérdida y que el valor perfecto es cero.

Similaridad de Jaccard: El índice de Jaccard o el coeficiente de similaridad de Jaccard, es el tamaño de la intersección dividida por el tamaño de la unión de dos conjuntos de etiquetas, esto es útil para comparar el conjunto de etiquetas predichas para una muestra correspondiente a un conjunto de etiquetas en los datos de entrenamiento.

F-Beta Score: Esta métrica es la media harmónica balanceada entre la precisión y el recall, alcanzando su óptimo valor en 1 y su peor valor en 0. El parámetro beta determina el peso de la precisión en el valor de la calificación.

5.3. Grid search

Las SVM particularmente son sensibles al conjunto de hiperparámetros con las que son entrenadas. Los hiper-parámetros o la configuración de los distintos estimadores no siempre pueden encontrarse directamente por un estimador, por eso es necesario hacer una Grid Search para encontrar la mejor configuración, es decir la que aporte la mejor exactitud, F1-score, score, recall, precisión, perdida de Hamming, similaridad de Jaccard, F-Beta score. Una Grid Search consiste en:

- Un estimador
- Un espacio de parámetros

- Un método para buscar o muestrear candidatos
- Un esquema de validación cruzada

Una Grid search es una búsqueda exhaustiva a través de un subconjunto del espacio de hiper-parámetros de un algoritmo de aprendizaje. Un algoritmo de Grid search debe ser guiado por una métrica de rendimiento, típicamente medido por validación cruzada en el subconjuto de entrenamiento [19]. Al ser una búsqueda exhaustiva la grid search sufre de *The curse of dimensionality*[3] pero generalmente es *embarrassingly parallel*[4] por lo que puede ocurrir una explosión combinatoria que retrase la búsqueda del mejor conjunto de hiperparametros en el espacio de un algoritmo de aprendizaje automático.

5.4. Evaluando el rendimiento base

Dado que muchas tareas de aprendizaje automático y minería de datos tratan de incrementar la tasa de éxito de resolución de un problema (i.e. tareas de clasificación). Evaluar la tasa base de éxito puede aportar un valor mínimo que otro estimador debe superar. Para comparar el resultado usamos un clasificador que usa estrategias muy simples (aleatorio y siempre predice la etiqueta más frecuente en el conjunto de entrenamiento). Este clasificador nos da una medida del rendimiento base del sistema (i.e. la tasa de éxito que uno debería esperar alcanzar aun cuando simplemente este adivinando).

Supongamos que queremos determinar si una opinión tiene o tiene alguna propiedad. Si hemos analizado un gran número de esas opiniones y hemos encontrado que el 90 % contiene la propiedad objetivo, entonces adivinar que cada futura instancia de la opinión objetivo la posea nos da un 90 % de probabilidad de adivinar correctamente.

Esto es equivalente a usar la estrategia de clasificación más frecuente que implementa la herramienta con la que se hizo el entrenamiento. Se obtuvieron los siguientes resultados con el sistema base: exactitud: 0.330767436436, F1 score: 0.33034110814, score: 0.321957945536, recall: 0.330767436436, precisión: 0.329920827829, perdidade Hamming: 0.669232563564, similaridad de Jaccard: 0.330767436436, F-Beta score: 0.200236094347. Los resultados de todos los experimentos se muestran en la Tabla 2, donde se observa que mejoran sobradamente los resultados del sistema base.

6. Resultados

A continuación presentamos los resultados de clasificar las opiniones en idioma

[3] En análisis numérico, muestreo, combinatoria, aprendizaje automático, minería de datos y bases de datos se refiere al fenómeno que surge al analizar y organizar datos en espacios de dimensiones muy altas

[4] Esto significa que se requiere poco o ningún esfuerzo para separar el problema en una serie de tareas paralelas

español mediante distintas configuraciones de máquinas de soporte vectorial con las características propuestas representadas con TF-idf.

En la figura 1 mostramos el rendimiento para un kernel lineal y un polinomial, debido a que el rendimiento del kernel RBF fue muy parecido al polinomial. En el análisis de resultados nos referimos a los valores obtenidos para el kernel lineal ya que fueron mejores aproximadamente en una relación de 2:1 a los otros dos.

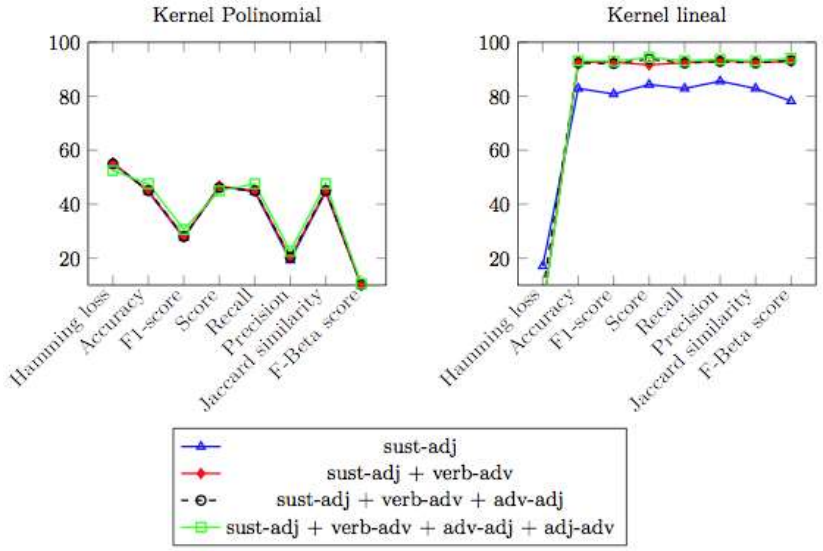


Fig. 1. Rendimiento del sistema con las distintas configuraciones y características propuestas.

Separamos la aplicación de los bigramas como características de entrenamiento en los siguientes conjuntos:

1. Sustantivo-adjetivo
2. Sustantivo-adjetivo y Verbo-adverbio
3. Sustantivo-adjetivo, Verbo-adverbio y Adverbio-adjetivo
4. Sustantivo-adjetivo, Verbo-adverbio, Adverbio-adjetivo y Adjetivo-adverbio

El conjunto 1 considera únicamente el bigrama Sustantivo-adjetivo que cumple una relación sintáctica, ya que como lo indicamos previamente se verificaron concordancias de género y número. La razón para iniciar con este bigrama es que expresa atributos de sustantivos que corresponderían a atributos de características del producto. La evaluación obtenida muestra resultados importantes: exactitud de 82.86 y F-Beta score de 78.22.

El conjunto 2 adiciona el bigrama Verbo-adverbio mediante el cual se puede expresar el modo en que se realiza la acción descrita por el verbo. La adición de este bigrama mejora la clasificación en casi un 10 %, la exactitid pasa de 82.86 a 92.65 y F-Beta score de 78.22 a 92.85.

El conjunto 3 adiciona a los anteriores bigramas el correspondiente a Adverbio-adjetivo. Este experimento muestra que este bigrama no es de utilidad. La exactitud

retrocede de 92.65 a 92.30 y F-Beta score pasa de 92.85 a 93.23 Consideramos este bigrama porque cuando el adverbio se une con un adjetivo, su función semántica es cualificadora o cuantificadora. Sin embargo, en estas opiniones su aportación a la clasificación es menor en relación a otros bigramas.

Tabla 2. Resultados de los experimentos.

Característica	**Máquina de soporte vectorial**			
Bigrama	*Métricas*	*RBF*	*Polinomial*	*Lineal*
Sustantivo-adjetivo	Exactitud:	46.96	44.63	82.86
	F1 score:	30.02	27.55	80.83
	Score:	45.22	46.37	84.31
	Recall:	46.96	44.37	82.86
	Precisión:	22.06	19.22	85.56
	Perdida de Hamming:	53.03	55.36	0.171
	Similaridad de Jaccard:	46.96	44.63	82.86
	F-Beta score:	10.50	10.03	78.22
Sustantivo - adjetivo Verbo - adverbio	Exactitud:	47.08	44.87	92.65
	F1 score:	30.14	27.79	92.44
	Score:	45.86	46.26	91.60
	Recall:	47.86	44.87	92.44
	Precisión:	22.17	20.13	93.08
	Perdida de Hamming:	52.91	55.12	0.073
	Similaridad de Jaccard:	47.08	44.87	92.65
	F-Beta score:	10.53	10.08	92.85
Sustantivo - adjetivo Verbo - adverbio Adverbio - adjetivo	Exactitud:	45.10	46.03	92.30
	F1 score:	28.04	29.02	92.08
	Score:	46.14	45.68	93.73
	Recall:	45.10	46.03	92.30
	Precision:	20.34	21.19	92.86
	Perdida de Hamming:	54.89	53.96	0.076
	Similaridad de Jaccard:	45.10	46.03	92.30
	F-Beta score:	10.13	10.32	93.23
Sustantivo - adjetivo Verbo - adverbio Adverbio -	Exactitud:	44.63	47.55	93.12
	F1 score:	27.55	30.64	92.99
	Score:	46.63	44.94	94.36
	Recall:	44.63	47.55	93.12
	Precision:	19.92	22.61	93.12

adjetivo Adjetivo - adverbio	Perdida de Hamming:	55.36	52.44	0.068
	Similaridad de Jaccard:	44.63	47.55	93.12
	F-Beta score:	10.03	10.62	94.07

El conjunto 4 adiciona el bigrama Adjetivo-adverbio, el cuál no corresponde a un orden muy común en el español. Sin embargo, algunas secuencias como *perfecto desde_luego*, *mejor claro*, *super bien*, mejoran los resultados de la clasificación. La exactitud pasa de 92.30 a 93.12 y F-Beta score de 93.23 a 94.07

Aunque la comparación con otros trabajos no puede ser directa por las colecciones empleadas, las características de entrenamiento, y los parámetros de los métodos supervisados, a continuación indicamos algunos resultados de clasificación de polaridad de opiniones en español.

En [8] utilizan una colección de 25 opiniones favorables y 25 opiniones desfavorables para lavadoras y con un método no supervisado obtienen 0.88 de *Accuracy* para opiniones negativas y 0.76 para opiniones negativas. En [6] aplican el método SVM a la colección de opiniones de cine de [7] y obtienen *Precision* de 0.8771, *Recall* de 0.8763, F1 de 0.8767 y *Accuracy* de 0.8766. Estos resultados muestran que los aquí obtenidos con bigramas se equiparan con el estado del arte de opiniones en español.

7. Conclusiones

En este artículo describimos experimentos realizados sobre un corpus de opiniones en español para obtener la orientación semántica de cada opinión. Analizamos el impacto de los bigramas morfosintácticos que definimos y extrajimos de la colección sobre el funcionamiento de un método supervisado para la clasificación de dicha orientación semántica. Los resultados muestran el impacto de cada bigrama, y especialmente el aporte menor del bigrama adverbio-adjetivo. En un trabajo futuro incluiremos bigramas de connotación negativa y haremos experimentos para balancear la colección.

Exploramos la utilidad de la biblioteca Scikit-learn de Python ya que este paquete es totalmente abierto y reutilizable. Debido a que Python se ha convertido en un lenguaje de programación ampliamente utilizado en Procesamiento de lenguaje natural consideramos importante explorar su funcionamiento para aprendizaje supervisado. Las funcionalidades implementadas nos permitieron hacer diversas evaluaciones y asignación de valores a parámetros del método SVM.

Referencias

1. Pang, B., Lee, L.: Opinion mining and sentiment analysis. Foundations and trends in information retrieval, vol. 2 (1-2), pp.1–135 (2008)

2. Liu, B.: Sentiment analysis and subjectivity. Handbook of natural language processing, pp. 627–666 (2010)
3. Turney, P. D.: Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In Proceedings of the 40th annual meeting on association for computational linguistics (ACL), pp. 417–424 (2002)
4. Hatzivassiloglou, V., Kathleen, R. M.: Predicting the Semantic Orientation of Adjectives. In EACL '97 Proceedings of the eighth conference on European chapter of the Association for Computational Linguistics, pp. 174–181 (1997)
5. Pang, B., Lee, L., Vaithyanathan, S.: Thumbs up? Sentiment Classification using Machine Learning Techniques. In Proceedings of EMNLP, pp. 79–86 (2002)
6. Martín-Valdivia, M. T., Martínez-Camara, E., Perea-Ortega, J.M., Ureña López, L.A.: Sentiment polarity detection in spanish reviews combining supervised and unsupervised approaches. Expert Systems with Applications Expert Systems with Applications, vol. 40 (10), pp. 3934–3942 (2013)
7. Cruz Mata, F., Troyano Jiménez, J. A., Enríquez de Salamanca, R., Ortega Rodríguez, F.J.: Clasificación de documentos basada en la opinión: experimentos con un corpus de críticas de cine en español. Procesamiento del lenguaje natural, vol. 41, pp. 73–80 (2008)
8. Vilares, D., Alonso, M. A., Gómez-Rodríguez, C.: A syntactic approach for opinion mining on Spanish reviews. Natural Language Engineering, vol. 1(1), pp. 1–26 (2013)
9. Galicia-Haro, S. N., Gelbukh, A.: Extraction of Semantic Relations from Opinion Reviews in Spanish. In Human-Inspired Computing and Its Applications. Lecture Notes in Computer Science, vol. 8856, pp. 175–190. Springer (2014)
10. Padró, L., Stanilovsky, E.: Freeling 3.0: Towards wider multilinguality. In Proceedings of the Language Resources and Evaluation Conference (LREC 2012), Istanbul, Turkey (2012)
11. López, V., Fernández, A., García, S., Palade, V., Herrera, F.: An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics. Information Sciences, vol. 250, pp.113–141 (2013)
12. Sun, Y., Kamel, M. S., Wong, A., Wang, Y.: Cost-sensitive boosting for classification of imbalanced data. Pattern Recognition, vol. 40 (12), pp. 3358–3378 (2007)
13. Pazzani, M., Merz, C., Murphy, P., Kamal, A., Hume, T., Brunk, C.: Reducing misclassification costs. In Proceedings of the Eleventh International Conference on Machine Learning, pp. 217–225 (1994)
14. Tang, Y., Yan-Qing, Z., Nitesh, V. C., Krasser, S.: Svms modeling for highly imbalanced classification. Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on, vol. 39(1), pp. 281–288 (2009)
15. Rehan, A., Kwek, S., Japkowicz, N.: Applying support vector machines to imbalanced datasets. In Machine Learning: ECML 2004, pp. 39–50 (2004)
16. Joachims, T.: Text categorization with support vector machines: Learning with many relevant features. Springer (1998)
17. McKinney, W.: Data structures for statistical computing in python. In Proceedings of the 9th Python in Science Conference, pp. 51–56 (2010)
18. Chih-Chung, C., Chih-Jen, L.: LibSVM: a Library for Support Vector Machines. ACM Transactions on Intelligent Systems and Technology (TIST), vol. 2(3), p. 27 (2011)
19. Chih-Wei, H., Chih-Chung, C., Chih-Jen, L.: A practical guide to support vector classification. http://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf

Sistema de reconocimiento multilenguaje del habla

Ali Montiel, Mario De Jesús, Raúl Hernández, Rubén Maldonado, Veronica Olvera, Yanette Morales y Leticia Flores-Pulido

Universidad Autónoma de Tlaxcala, Facultad de Ingeniería y Tecnología, Apizaco, Tlaxcala, México

{ilusion119,idmariodjc,chanbrawl,rumalo1791,vero.pink15,yanette_morales_salado,aicitel.flores}@gmail.com

Resumen. Este trabajo se comienza con la presentación de una serie de artículos relacionados con el Reconocimiento Automático del Habla. Se realiza un análisis de cada uno de ellos donde se obtienen datos relevantes y los que serán de gran ayuda para desarrollar la propuesta multilenguaje de un sistema de reconocimiento del habla aquí descrito. Existen varias técnicas que son aplicadas para lograr una efectividad más alta de los sistemas basados en Reconocimiento Automático del Habla. Entre las más utilizadas se encuentran los coeficientes cepstrales de Mel, el modelo oculto de Markov y Coeficientes Predictivos Lineales. Cada uno de los trabajos relacionados con el reconocimiento automático del habla presenta su propio modelo de lenguaje y un modelo acústico que permite tener un amplio porcentaje de efectividad. Las técnicas anteriormente mencionadas forman parte de la extracción de características de la prupuesta multilenguaje. El objetivo entonces es una propuesta de implementación que pueda reconocer diferentes clases de idiomas basado en una extracción de características bajo la combinación de técnicas como son los modelos ocultos de markov y los coeficientes de predicción lineal. En éste trabajo se muestra la etapa de extracción de formantes de tres corpus del habla de diferentes idiomas: PRESEEA, EUSTACE y DIMEX100.

Palabras clave: coeficientes de Mel, espectrograma, frecuencia, coeficientes predictivos lineales, modelo oculto de Markov.

1. Introducción

El proceso de reconocimiento automático del habla (RAH) dota a las máquinas de la capacidad de recibir mensajes orales. El reconocimiento automático del habla proporciona una nueva forma de interactuar con un computador, en este caso a través de la voz, este tipo de interfaces también son llamadas de usuario de voz, e interfaces basadas en el habla. Las tecnologías del habla son muy utilizadas en las aplicaciones de servicios telefónicos ofrecidos a los

usuarios para la realización de alguna operación bancaria. Las tecnologías del reconocimiento de voz se realizan bajo tres pilares, diseño de IVR/SIU, las ciencias del servicio y los factores humanos. Este tipo de tecnologías abre un gran abanico de aplicaciones prácticas como por ejemplo: (a) Sistemas de dictado, donde lo que se pretende es una transcripción textual lo más exacta posible de aquello que ha dicho un locutor. Y (b) Sistemas de diálogo, donde el objetivo es conceptualizar aquello que se ha captado por el sensor auditivo e inferir una respuesta. En definitivo, el reconocimiento automático del habla es un campo con gran interés práctico y que presenta problemas no precisamente triviales de resolver. Es por ello que se propone un sistema que reuna tres tipos de corpus del habla bajo diferentes idiomas: español de España, Inglés Británico y Español de México, que sea capaz de conformar tres clases de formantes que puedan ser discretizados por diferentes extractores de características y que además puedan ser reconocidos.

2. Estado del arte

En [1] se trabajó con un reconocedor que utilizó elementos independientes del contexto, denominadas "monófonos", como unidades básicas del modelo acústico. Para la creación de los modelos se emplearon modelos ocultos de Markov MOM de tres estados de izquierda a derecha del tipo semi-continuo asociados a cada uno de los 31 monófonos (30 fonemas + alófonos y un modelo de silencio). En [2] se presentan dos sistemas de análisis acústico del habla con aplicaciones a la descripción de segmentos de discurso espontáneo y un sistema de reconocimiento automático de habla espontánea orientado a la detección de palabras. En [3] se tiene como objetivo mejorar la interacción el hombre y la máquina, haciendo posible que un determinado dispositivo pueda rescatar información afectiva más que el contenido hablado por una persona. En [4] se menciona que el ruido de fondo está frecuentemente presente en ambientes donde se emplean sistemas de Reconocimiento Automático del Habla (RAH). Una señal ruidosa da lugar a una degradación en la tarea del reconocimiento debido al desajuste con el modelo acústico (MA). En [5] se plantea que la motivación principal es crear un sistema de reconocimiento automático del habla en el idioma español, el cual tiene como objetivo lograr altas tasas de reconocimiento en comparación con otros sistemas de su tipo. En [6] se considera también al ruido como uno de los principales factores a tener en cuenta en las aplicaciones reales del reconocimiento automático de voz. El rendimiento de los reconocedores se ve fuertemente afectado cuando la señal de voz es adquirida en un entorno ruidoso. En [7] se propone un algoritmo para el reconocimiento de personas en un canal telefónico. El algoritmo se basa en el comportamiento de las Redes Neuronales Artificiales (RNA), en particular, sobre el algoritmo Backpropagation. En [8] se presenta a Kaldi que es una herramienta que proporciona una biblioteca de módulos diseñados para acelerar la creación de sistemas automáticos de reconocimiento de voz para fines de investigación. Los efectos del modelado acústico y el conjunto de herramientas proporciona un marco para formantes bajo redes

neuronales mediante descenso de gradiente estocástico para el reconocimiento del habla.

3. Métodos de reconocimiento automático del habla

Existen varios métodos de reconocimiento del habla, los cuales no serán descritos a detalle, pero si serán mencionados a grandes rasgos para comprensión del lector.

3.1. Coeficientes predictivos lineales (Linear Predictive Coding)

Los CPL (coeficientes predictivos lineales) son un modelo para la producción de la señal de voz con la suposición inicial de que la señal de voz es producida bajo un modelo acústico muy especifico. Es un método para el modelado de la señal de voz y es de uso frecuente por los lingüistas como una herramienta de extracción de formantes. El análisis LPC es generalmente apropiado para modelar las vocales que son periódicas, salvo las vocales nasales. El LPC se basa en el modelo de fuente-filtro de la señal de voz.

El algoritmo consiste en lo siguiente:

- Pre énfasis: La señal de voz digitalizada, $s(n)$, se somete a un sistema digital de bajo orden, para espectralmente aplanar la señal y hacerla menos suceptible a efectos de presición finita posteriores en el procesamiento de la señal. La salida de la red de pre énfasis, está relacionada a la entrada de la red, $s(n)$, por la siguiete ecuación:

$$\tilde{s}(n) = s(n) - \tilde{s}(n-1) \tag{1}$$

- Empaquetado de marcos: La salida de la pre énfasis es empaquetada en marcos de N muestras, con marcos adyacentes los cuales son separados en muestras M. Si $x_i(n)$ es el l^{th} marco del habla, y hay L marcos con señal del habla entera, entonces

$$x_i(n) = \tilde{s}(Ml + n) \tag{2}$$

  donde $(n = 0, 1, ..., N)$ y $(l = 0, 1, ..., L-1)$
- Ventaneo: Después de empaquetar en marcos, el siguiente paso es que a cada marco se le minimizan las discontinuidades de la señal de principio a fin. Si definimos la ventana como $w(n), 0 \leq n \leq N-1$ entonces el resultado del ventaneo es la señal:

$$\tilde{x}(n) = x_i(n)w(n) \tag{3}$$

  donde $0 \leq n \leq N-1$

- Analisis de autocorrección: El siguiente paso es correlacionar cada marco de señal ventaneada en orden para dar

$$r_l(m) = \sum_{n=0}^{N-1-m} \tilde{x}_l(n)\tilde{x}_l(n+m) \tag{4}$$

donde el valor de autocorrección más alto, (p), es el orden del análisis CPL.

- Análisis CPL: El siguiente paso es el análisis CPL, donde se convierte cada marco de $(p+1)$ autocorrecciones a un conjunto de parámetros CPL usando el método de Durbin. Esto puede ser dado mediante el siguiente algoritmo:

$$E^{(0)} = r(0) \tag{5}$$

$$k_i = \frac{r(i) - \sum_{j=1}^{i-1} \alpha_j^{i-1} r(|i-j|)}{E^{i-1}} \tag{6}$$

$$\alpha_j^{(i)} = k_i \tag{7}$$

$$\alpha_j^{(i)} = \alpha_j^{(l-i)} - k_i \alpha_{i-j}^{(i-1)} \tag{8}$$

$$E^{(i)} = (l - k_i^2)E^{i-1} \tag{9}$$

Al resolver de 5 a 9 recursivamente para $i = 1, 2, ..., p$, el coeficiente CPL, a_m, es dado como

$$a_m = \alpha_m^{(p)} \tag{10}$$

- Conversión de parámetros CPL a coeficientes cepstrales: Los coeficientes cepstrales pueden ser derivados directamente del conjunto de coeficientes CPL. La recursión usada es
Para $1 \leq m \leq p$

$$c_m = a_m + \sum_{k=1}^{m_1} (\frac{k}{m}) * c_k * a_{m-k} \tag{11}$$

Para $m \geq p$

$$c_m = \sum_{k=m-p}^{m-1} (\frac{k}{m}) * c_k * a_{m-k} \tag{12}$$

3.2. Modelo oculto de Markov (MOM)

Es un modelo estadístico donde se asume que el sistema a modelar es un proceso de Markov de parámetros desconocidos. El objetivo es determinar los parámetros desconocidos de una cadena a partir de los parámetros observables. Los parámetros extraídos se pueden emplear para llevar a cabo análisis sucesivos. En un modelo oculto de Markov, el estado no es visible directamente, lo son las variables influidas por el estado. Cada estado tiene una distribución de probabilidad sobre los posibles símbolos de salida. Consecuentemente, la secuencia de símbolos generada por un MOM proporciona cierta información acerca de la secuencia de estados. Los modelos ocultos de Markov son aplicados a reconocimiento de formas temporales, como reconocimiento del habla, de escritura manual, de gestos, etiquetado gramatical o en bioinformática. En el reconocimiento de voz se emplea para modelar una frase completa, una palabra, un fonema o trifonema en el modelo acústico.

La Figura 1 muestra la arquitectura general de un MOM. Cada óvalo representa una variable aleatoria que puede tomar determinados valores. La variable aleatoria $x(t)$ es el valor de la variable oculta en el instante de tiempo t. La variable aleatoria $y(t)$ es el valor de la variable observada en el mismo instante de tiempo t., las flechas indican dependencias condicionales. El valor de la variable oculta $x(t)$ (en el instante t) solo depende del valor de la variable oculta $x(t-1)$ (en el instante $t-1$). A esto se le llama propiedad de Markov.

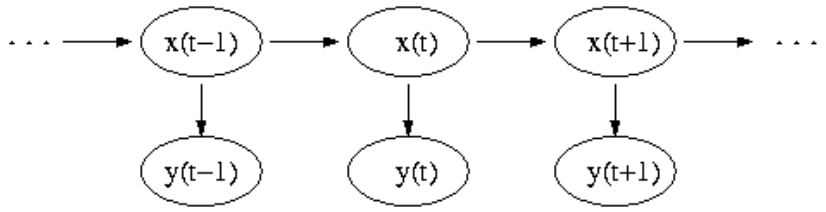


Fig. 1. Diagrama de la arquitectura general de un MOM.

Representación formal del modelo oculto de Markov
Una notación común del MOM es representarlo como una tupla:

$$(Q, V, \pi, A, B)$$

donde:

- El conjunto de estados $Q = 1, 2, ..., N$
  - El estado inicial se denota como q_t
  - En el caso de la etiquetación, cada valor de t hace referencia a la posición de la palabra en la oración.
- El conjunto V representa los posibles valores $_1, v_2, ..., v_M$ observables en cada estado

  - M es el número de palabras posibles y cada v_k hace referencia a una palablra diferente.
- $\pi = \pi_i$ son las probabilidades iniciales, donde:
  - π_i es la probabilidad de que el primer estado sea el estado Q_i
- El conjunto de probabilidades de transiciones entre estados se denota por $A = a_{ij}$

$$a_{ij} = P(q_t = j | q_{t-1=i}) \tag{13}$$

  donde, a_{ij} es la probabilidad de estar en el estado j es el instante t si en el instante anterior $t-1$ estaba en el instante i.
- El conjunto de probabilidades de las observaciones se representa por $B = b_j(v_k)$.
  $b_j(v_k) = P(o_t = v_k | q_t = j)$, es decir, la probabilidad de observar v_k cuando se está en el estado j en el instante t.
- La secuencia de observables se denota como un conjunto $O = (o_q, o_2, ..., o_T)$.

Los Modelos ocultos de Markov han demostrado ser una técnica efectiva en el procesamiento del Reconocimiento Automático del Habla. Para este trabajo se aplicará dicha técnica en el Modelo Acústico donde servirá de ayuda para la extracción de formantes de palabras, fonemas, o incluso de frases completas.

3.3. Coeficientes cepstrales en frecuencia MEL

Una técnica de extracción de parámetros de las más importantes y utilizadas actualmente en varios sistemas de reconocimiento de voz, es la obtención de los coeficientes de frecuencia Mel (CFM). Los coeficientes CFM son un tipo particular de coeficientes cepstrales derivados de la aplicación del Cepstrum sobre una ventana de tiempo de la señal de voz. El concepto de coeficientes CFM surge de hacer uso de una nueva escala de frecuencia no lineal denominada MEL para imitar el comportamiento psicoacústico a tonos puros de distinta frecuencia dentro del oído humano. De hecho, estudios dentro de esta ciencia han demostrado que el sistema auditivo humano procesa la señal de voz en el dominio espectral, caracterizándose por tener mayores resoluciones en bajas frecuencias y esto es precisamente lo que se consigue mediante la escala MEL, asignar mayor relevancia a las bajas frecuencias de forma análoga a como se hace en el sistema auditivo humano, en concreto en el oído interno. La obtención de los coeficientes MFCC ha sido considerada como una de las técnicas de parametrización de la voz más importante y utilizada dentro del área de verificación de interlocutor. El objetivo de esta transformación es obtener una representación compacta, robusta y apropiada para posteriormente poder obtener un modelo estadístico del locutor con un alto grado de precisión. Para obtener los coeficientes cepstrales en frecuencia MEL se aplica la Ecuación 14.

$$C_{MFCC}[m] = \sum_{k=0}^{N-1} log(E_k) cos\left(m\left(d - \frac{1}{2}\right)\frac{\pi}{N}\right) \tag{14}$$

donde:

- $m = m - esimo$ coeficiente MEL calculado.
- d = número de filtros utilizados en el banco de filtros MEL
- N = Tamaño de la Transformada Discreta de Fourier aplicada a la señal de voz enventanada.
- E_k = Energía correspondiente a cada uno de los F filtros

Particularmente, consideramos que ésta forma de parametrización de la señal de voz es muy conveniente y fácil de obtener. Sustentándonos en la teoría presentada, los coeficientes cepstrales en frecuencia MEL son parámetros que ofrecen información relevante de una señal de voz, además que permiten separar las dos componentes de información de la misma: la entonación y del tracto vocal.

4. Corpus de reconocimiento automático del habla

Los principales corpus a utilizar dentro de la propuesta multilenguaje, son mencionados a continuación:

- Corpus PRESEEA [17] el cual tiene como principal objetivo identificar los rasgos característicos del español hablado de Valencia. Este nace en 1996 por el equipo de investigación PRESEEA, coordinado por el Dr. José Ramón Gómez Molina. Las muestras recopiladas corresponden a 72 entrevistas semidirigidas con informantes de 3 niveles socioculturales y con un contenido aproximado de 425.000 palabras. Dicho Corpus, facilita la identificación de los rasgos característicos del castellano usado por los hablantes de dicha área metropolitana en un registro comunicativo semiformal o neutro.
- Corpus de Inglés de la Universidad de Edimburgo (EUSTACE). El Corpus EUSTACE [14] comprende 4608 oraciones habladas grabadas en el departamento de Lingüística Teórica Aplicadas de la Universidad de Edimburgo. Estas oraciones son mencionadas por seis hablantes del inglés británico, 3 mujeres y 3 hombres y fueron diseñadas para examinar el número de efectos duracionales en la voz y están controladas por su longitud y contenido fonético. En la Figura 2 se muestra la señal de voz y el espectrograma de una muestra de voz perteneciente al corpus EUSTACE.
- DIMEx100 y DIME (Diálogos Inteligentes Multimodales en Español). El Corpus DIMEx100 [15] tiene por objetivo hacer posible la construcción de modelos acústicos y diccionarios de pronunciación para la creación de sistemas computacionales para el reconocimiento del español hablado en México. Este tipo de sistemas permiten transcribir una señal de voz en su representación textual.

4.1. Tabla comparativa de los corpus utilizados

En la Tabla 1 que contiene los elementos principales de los tres corpus a utilizar en este trabajo, tomando como características principales el número de muestras, número de locutores que interfieren y el tipo de muestra.

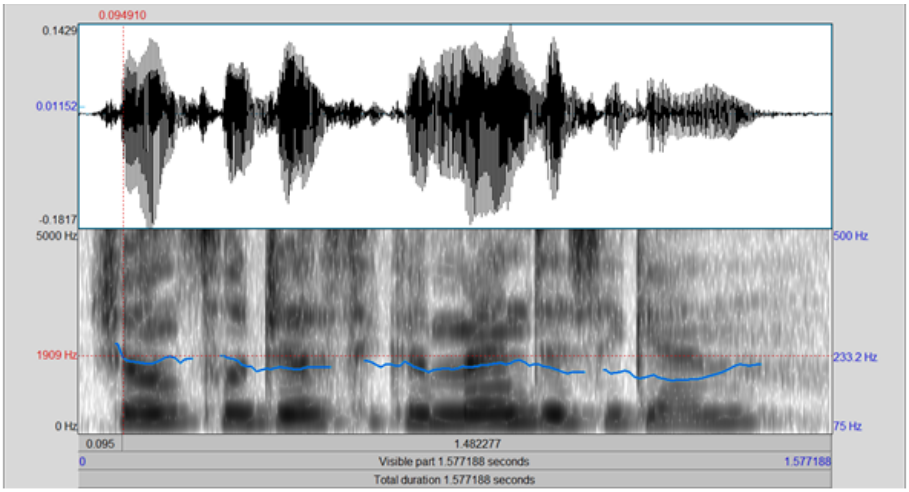


Fig. 2. Espectrograma de voz de la frase "John saw Jessica mend it again"pronunciada por una mujer en Inglés Británico.

Tabla 1. Tabla comparativa de corpus utilizados.

Nombre del Corpus	No. de muestras	No. de locutores	Tipo de muestra
PRESEEA	72 Entrevistas	4	Muestreo con extensión fija y exhaustiva.
EUSTACE	4608 oraciones	6	Formato ESPS y WAV, a una tasa de muestreo de 16 KHz y 24 dB de magnitud.
DIMEx100 y DIME	5010 oraciones	100	Formato mono a 16 bits y a 44.1 kHz, bajo *Wave Label*.

5. Sistema de reconocimiento automático del habla para corpus multilenguaje del locutor (RAHM)

La Figura 3 muestra cada uno de los corpus que sirven como entrada a ésta propuesta, los cuales llevan por nombre PRESEEA, EUSTACE y DIMEx100 respectivamente, previamente descritos. Cada uno de los componentes integrados en cada corpus deberán pasar por una extracción de características, donde podremos examinar más a fondo cada una de las partes resultantes de los corpus. Se obtendrán entonces ciertos formantes resultantes de cada corpus cada uno bajo diferentes métodos, es decir, MOM, CFM y CPL respectivamente para PRESEEA, EUSTACE y DIMEx100. La propuesta de Reconocimiento del Habla Mutilenguaje en la etapa de extracción de características, se puede apreciar en la Figura 3.

A continuación se muestra el avance de dicha propuesta, donde se ha realizado

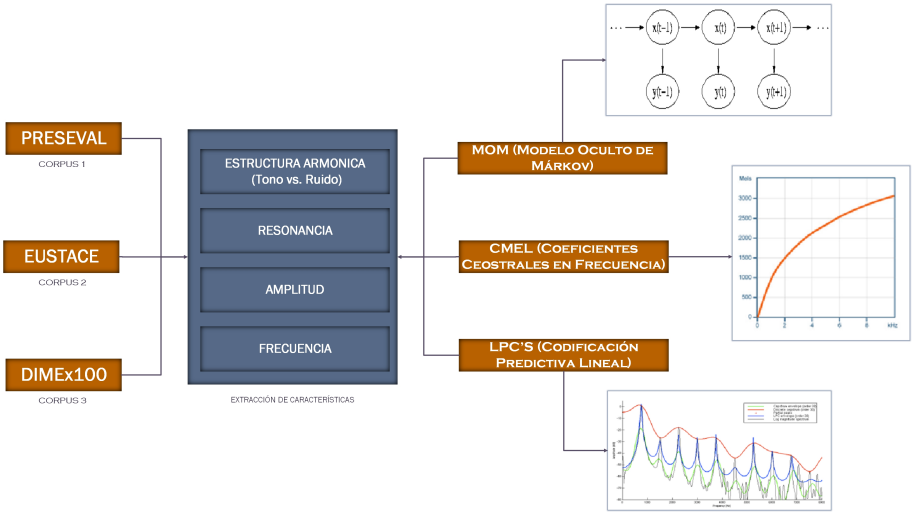


Fig. 3. Propuesta para el sistema de reconocimiento multilenguaje bajo MOM, CMEL y LPC's, en la etapa de extracción de formantes.

la extracción de las caracter—'isticas de los 3 corpus de voz anteriormente descritos.

5.1. Obtención de las características del corpus PRESEEA

El uso de la herramienta de PRAAT nos sirve para analizar u obtener características de audio, las cuales en ocasiones se obtienen a través de una señal o una voz, para posteriormente darle un uso específico dependiendo de nuestras necesidades. El objetivo es analizar el Corpus "PRESEEA" para posteriormente obtener la extracción de características que ayudará a la identificación del hablante. A continuación se enlistaran los pasos para la extracción de características de las muestras: (a) se seleccionan las 10 muestras desde la ventana Praat Objects, se visualiza el espectograma y se determina el rango de 3500Hz, (b) se obtiene el Pitch (Tono, Hz) y se obtiene la intensidad (db), (c) se hace el cálculo de sus formantes, de modo que se pueda visualizar su trayectoria a lo largo de la onda. En la Figura 4 se muestra la matriz de las 10 muestras de voz que fueron obtenidas por el Corpus PRESEEA y donde se muestra los valores correspondientes a las características obtenidas.

5.2. Obtención de las características del corpus DIMEx100

Para obtener las características de voz de dicho corpus fue necesario separar las palabras por cada oración dicha por el locutor, obtenidas por medio del programa Praat. Dentro del corpus se encuentran un total de 7 locutores diferentes donde intervienen 2 mujeres y 5 hombres. A cada palabra se le hizo la extracción

CORPUS PRESEEA

Muestras	Punto en el espectrograma	Segundos	(Análisis del tono Hz)	(Análisis de la intensidad dB)	Formantes			
Conv1	3500	3.609011	109.986399	78.931595	269.607769	1305.65853	2224.46005	3677.115703
Conv2	3500	1.128443	98.642629	86.079772	260.655081	1699.79315	2654.19012	3473.443557
Conv3	3500	1.003868	105.133992	82.792438	342.239126	1693.29933	2661.80493	3521.98262
Conv4	3500	1.203868	107.31849	77.291366	304.515062	1694.41563	2497.10435	3503.582736
Conv5	3500	1.375867	90.576321	89.471606	335.009811	1573.32754	2731.91205	3518.874236
Conv6	3500	0.83764	116.714949	82.724015	340.700786	1256.72135	2332.20479	3430.623548
Conv7	3500	0.157067	116.778331	81.547988	302.456841	1611.80108	2645.7048	3492.397091
Conv8	3500	0.132583	113.007869	81.794636	298.632725	1590.04476	2632.58269	3467.177862
Conv9	3500	1.715219	120.404686	84.052438	362.239126	1513.29933	2171.80493	35423.98262
Conv10	3500	1.219037	118.678473	85.026402	332.28554	1305.70486	2145.18913	3133.558223

Fig. 4. Tabla de características calculadas para el corpus PRESEEA.

del tono el cual es medido en hertz y así mismo la intensidad, la cual esta dada en decibeles, y por último se hizo la extracción de los 4 formantes, medidos en hertz. Aunque cada muestra varió en tiempos, se estableció una frecuencia de 3500 Hertz como condición inicial como en el corpus anterior. Para seguir con el procedimiento, fue necesario conocer por cada locutor el promedio del tono, intensidad y los formantes. Después de obtener cada una de las características para cada muestra se hizo el registro en una tabla y/o matriz de la cual en la Figura 5 se muestran los diez ejemplos tomados para realizar las gráficas correspondientes de cada una de las características como el tono, la intensidad y los formantes.

Muestra	Pitch (Hz)	Intensidad (dB)	Formante 1	Formante 2	Formante 3	Formante 4
locutor1_cual	166.771319	83.586892	523.202906	821.64902	2532.32265	3436.013358
locutor2fem_avancemos	286.335852	81.989834	577.37283	1954.711557	2909.773112	4267.363891
locutor3mas_estamos	164.312667	62.67601	491.08748	1869.265095	2579.405397	3975.68134
locutor3mas_puntos	164.412107	65.090599	433.702412	934.295444	2405.740231	3761.202684
locutor4fem_departamento	191.660023	70.958987	636.050739	1632.534055	2595.586095	4103.359404
Locutor5_explicar	155.082241	76.158301	394.5892706	2216.978903	3481.371294	3675.838238
Locutor6_41_herramienta	143.147655	71.41522	519.439371	1871.973696	2573.135455	3750.49384
Locutor7_31_competencia	90.656473	69.578603	430.189729	1487.581143	2317.03755	3576.601287
Locutor6_32_posible	148.743834	75.112057	404.245684	2113.509593	3460.688218	4070.694819
Locutor5_43_desarrollado	92.228238	76.894352	487.025495	1679.91227	2940.942573	3443.590818

Fig. 5. Matriz de las características obtenidas del corpus DIMEx100.

5.3. Obtención de las características de (EUSTACE)

El corpus de voz cuenta con 4608 oraciones y 6 locutores: 3 hombres y 3 mujeres. Debido a que el tamaño del corpus es excesivamente grande, sólo se ha tomado una pequeña parte para la obtención de características. La porción tomada involucra 50 frases mencionadas por cada locutor. De cada una de esas frases se obtuvieron las siguientes características: Análisis de Tono, Análisis de Intensidad y Análisis de Formantes. Dichas características fueron tomadas a un nivel de frecuencia estándar de 3500 Hz. Cada uno de los archivos de audio contiene alrededor de 15 frases, por lo que para efectuar un análisis fue necesario tomar sólo la señal comprendida por cada frase, lo que implica tomar en cuenta el instante en el que se tomó la muestra. La matriz de características se compone de 300 señales analizadas, 50 por cada locutor, y 8 valores característicos relacionados con los puntos anteriormente mencionados (Frecuencia (Hz), Tiempo (s), Tono (Hz), Intensidad (dB), Formante 1-4 (Hz)). En la Figura 6 se presenta un extracto de las primeras 10 muestras con sus respectivas características.

Características del Corpus de Voz "EUSTACE"

Muestra	Frec (Hz)	T(s)	Pitch (Hz)	Intensidad (dB)	Form_1 (Hz)	Form_2 (Hz)	Form_3 (Hz)	Form_4 (Hz)
Locutor 1 Masculino Grupo 1								
m1/capae_1	3500	1.019	120.80644	68.610449	470.629965	1154.53321	3074.46535	3696.3701
m1/capae_2	3500	4.323	128.72135	68.178712	458.268372	1204.96584	2080.47514	3885.94491
m1/capae_3	3500	7.561	122.66044	71.33897	421.043548	1105.42664	2206.74614	3354.31847
m1/capae_4	3500	11.45	139.80837	69.593722	421.02079	1454.85417	2234.0941	3881.95069
m1/capae_5	3500	15.82	126.64631	71.538076	459.873574	1211.60622	2024.20717	3445.60513
m1/capae_6	3500	20.08	125.68676	71.040469	436.916172	1180.34927	2290.04377	3573.48219
m1/capae_7	3500	23.76	121.88977	68.320104	443.776884	1389.25531	2027.13849	3585.39417
m1/capae_8	3500	27.64	129.33968	72.183424	495.92082	1220.31347	2058.79925	3640.42604
m1/capae_9	3500	30.45	136.42908	66.342916	380.507625	1815.16089	2458.08092	3606.50476
m1/capae_10	3500	35.23	137.15905	68.816431	425.159286	1666.58828	2393.36433	3454.65783

Fig. 6. Matriz de muestras obtenidas del corpus EUSTACE.

A continuación se muestran las características extraídas de 4 formantes para cada corpus analizado. En la Figura 7 se observa la extracción de los formantes de PRESEEA, en la Figura 8 se muestran los formantes extraídos de EUSTACE, y en la Figura 9 los formantes de DIMEx100.

6. Conclusiones

El método CPL (Coeficientes de Predicción Lineal) se implementó para las muestras y se obtuvo una gráfica donde se hacía comparación de la señal original con el CPL estimado. Éste proceso contiene filtros para mejorar la señal. El algoritmo que se describió en el estado del arte también se utiliza para poder calcular el CPL a las muestras correspondientes. El corpus que se utilizo fue el

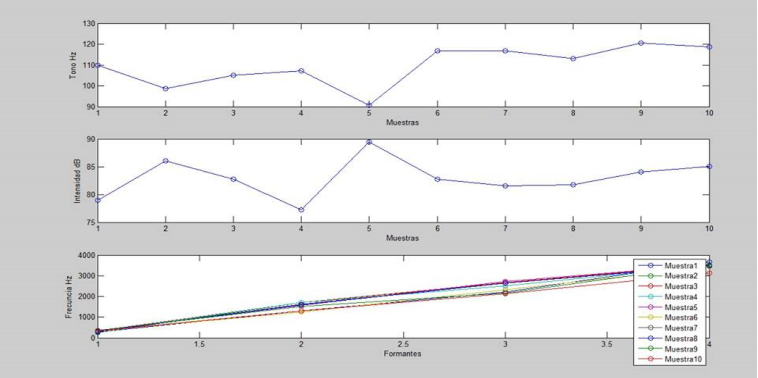


Fig. 7. Gráfica del extracción de formantes para PRESEEA.

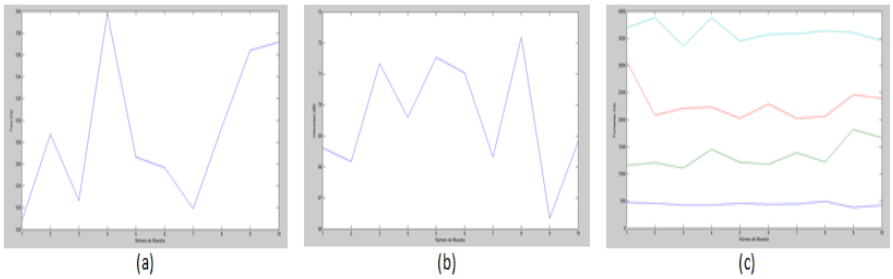


Fig. 8. Análisis de (a)Tono, (b)Intensidad y (c)Formantes de 10 frases contenidas en el corpus EUSTACE.

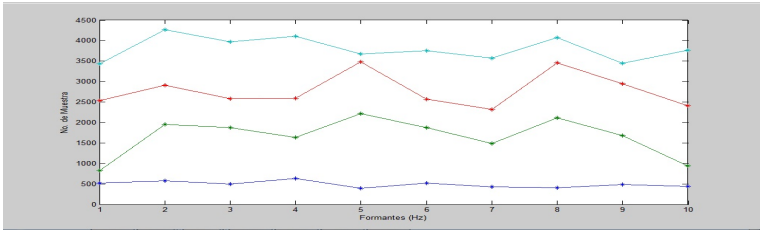


Fig. 9. Gráfica del extracción de formantes para DIMEx100.

de la Universidad de Valencia PRESEEA que se obtuvo de las respuestas de la pregunta? "Has hecho tu servicio militar?" Para las mujeres la pregunta fue "Si hubieras sido hombre / hubieras hecho el servicio militar?". De las respuestas obtenidas del hablante se extrajeron las características y se aplicó el método CPL. En el caso del Corpus DIMEx100 se encontraron diversas dificultades para su análisis. En primera, las muestras eran muy variadas en cuestión de locutores y de frases dichas por cada uno. Para su análisis fue necesario hacer el corte de las frases en palabras con el fin de lograr el reconocimiento de locutor, pero esto se dificultó al saber que las frases contenían diferentes palabras. Aun así fue relativamente sencillo obtener las características necesarias para su posterior análisis. En el caso de la técnica de los Modelos Ocultos de Markov es necesario mencionar que aunque es uno de los métodos más utilizados en el Reconocimiento Automático del Habla, era necesario adaptar nuestro corpus de diferente manera para poder hacer el procesamiento de extracción de formantes. De acuerdo al análisis que se realizó sobre la técnica antes mencionada, se puede concluir que es una de las mas eficaces para este tipo de trabajos. Particularmente, el corpus de voz EUSTACE es bastante robusto, así que sólo se consideró un 8 % aproximadamente de las señales de voz para el análisis y obtención de características. Las frases contenidas en cada archivo de audio analizado son sencillas y claras, lo que permite una fácil comparación entre las características obtenidas para cada una. El cálculo de las características fue relativamente simple, ya que son datos que se pueden obtener desde la herramienta utilizada de manera directa. Como resultado se obtuvieron 12 coeficientes en escala de mel, por cada serie de características y señal analizada, que representan aquellas frecuencias, las cuales proporcionan información relevante que puede ser útil en sistemas de Reconocimiento Automático del Habla. Es importante mencionar que la parte de la extracción de características para el resto del corpus será trabajo a futuro que resta por realizar bajo el esquema propuesto en la Figura 3.

Referencias

1. Univaso, P., Gurlekian, J. A., Evin, D.: Reconocimiento del habla continua independiente del contexto para el español de Argentina. Revista clepsidra, p. 11 (2009)
2. Grulekian, A. J., Evin, D., Torres, H., Renato, A.: Sistemas de Análisis Acústico y de Reconocimiento Automático en Habla Espontanea. Subjetividad y Procesos Cognitivos, vol. 14 (2), p. 10 (2010)
3. Solís Villarreal, J.F., Yáñez Márquez, C., Suárez Guerra, S.: Reconocimiento automático de voz emotiva con memorias asociativas Alfa-Beta SVM. Polibitis (2011)
4. Gomez, R., Tatsuya, K.: Denoising Using Optimized Wavelet Filtering for Automatic Speech Recognition. Academic Center for Computing and Media Studies (ACCMS), Kyoto University, Japan (2011)
5. Pérez, S., Pelle, P., Estienne, C., Messina, F.: Sistema de Reconocimiento de Habla en Español con adaptación al discurso. Universidad de Buenos Aires, p. 10 (2011)

6. De la Torre, A., Fohr, D., Paul, H.J.: Métodos Para Reconocimiento Robusto De Voz Adquirida En Automóviles. Universidad de Granada, Dpto. de Electrónica y Tec. Comp., España (2011)
7. Cruz-Beltrán, L., Acevedo-Mosqueda, M. A.: Reconocimiento de Voz usando Redes Neuronales Artificiales Backpropagation y Coeficientes LPC. SEPI Telecomunicaciones ESIME IPN (2011)
8. Edmons, C., Hu, S., Mandle, D.: Improvement of an Automatic Speech Recognition Toolkit (2012)
9. Thiang, Soryu Wijogo.: Speech recognition using linear predictive coding and artificial neural network for controlling movement of mobile robot. Electrical Engineearing Department, Petra Christian University, Indonesia (2011)
10. Antoniol, G., Rollo, V. F., Venturi, G.: Linear Predective Coding and Cepstrum coefficients for mining time variant information from software repositories. ACM SIGSOFT Software Engineering Notes, vol. 30 (4), pp. 1-5 (2005)
11. Makhoul, J.: Linear Prediction:A tutorial review. Proc. IEEE, pp. 561-580 (1975)
12. Colaboradores de Wikipedia, Modelos Ocultos de Markov. Wikipedia,La enciclopedia libre. http://es.wikipedia.org/wiki/Modelo_oculto_de_Markov
13. Extracción de Características. http://bibing.us.es/proyectos /abreproy/12054/fichero/MEMORIA %252F8.Cap %EDtulo+3.pdf
14. EUSTACE (Edinburgh University Speech Timing Archive and Corpus of English),CSTR (The Centre for Speech Technology Research), University of Edinburgh. http://www.cstr.ed.ac.uk/projects/eustace/index.html
15. DIMEx100 y DIME(Diálogos Inteligentes Multimodales en Español),Universidad Autónoma de México Centro de Ciencias Aplicadas y Desarrollo Tecnológico de la UNAM (CATED-UNAM). http:turing.iimas.unam.mx/ luis/DIME/DIMEx100/manualdimex100/index.html
16. Mel Frecuencial Cepstral Coeficients. http://es.wikipedia.org/wiki/MFCC
17. PRESEEA. http://www.uv.es/preseea/ppal.htm

Identificación de perfiles de usuario

P. Espinoza, D. Vilariño, D. Pinto, M. Tovar y B. Beltrán

Benemérita Universidad Autónoma de Puebla,
Facultad de Ciencias de la Computación,
Puebla, México

patricia.efong@gmail.mx
darnes,dpinto,mtovar,bbeltran@cs.buap.mx
http://nlp.cs.buap.mx

Resumen. En la presente investigación se propone un modelo para la identificación de perfiles de usuario. El modelo propuesto utiliza un conjunto de características extraídas de los textos. El modelo fue validado con 4 corpus en inglés: de Blogs, de Redes sociales, de Criticas y de Twitter y con 2 corpus en español: de Blogs y de Criticas. Se compara el desempeño de tres de los algoritmos más usados para clasificación: Naïve Bayes, Máquinas de Soporte Vectorial (SVM) y K Vecinos más cercanos (IBk).

Palabras clave: modelo de clasificación, perfil de usuario, patrones semánticos.

1. Introducción

Internet en la actualidad ofrece diversas herramientas para que los usuarios puedan expresarse libremente sin importar la edad, el sexo, el tema que traten y a que se dedican. La cantidad de conversaciones en línea (foros, salas de chats, redes sociales y blogs, entre otros medios) ha aumentado considerablemente. Dada esta situación es prácticamente imposible analizar manualmente una conversación y detectar el perfil del autor que la ha escrito.

La detección del perfil de un autor que puede ser edad, sexo, lenguaje nativo o tipo de personalidad es un problema que ha ganado importancia en aplicaciones forenses, de seguridad y de mercadotecnia. Hoy en día la comunidad de procesamiento de lenguaje natural desea estudiar la forma en que se comunican los diferentes grupos de edades y sexo, tratando de detectar los patrones de escritura comunes y diferentes entre estos grupos.

En la presente investigación se desea, dado un texto detectar la edad y el sexo de la persona que lo ha escrito. Dicho texto es un documento obtenido de los corpus de la Conferencia Internacional PAN 2014[1]. Lo que se pretende es encontrar patrones, características léxicas, sintácticas y semánticas propias de cada grupo de edad y género, para el desarrollo de modelos de aprendizaje que nos permitan clasificar adecuadamente dichas conversaciones.

[1] http://pan.webis.de

La estructura del artículo es la siguiente. En la sección 2 se presentan los trabajos desarrollados en la literatura con respecto a la identificación de perfiles de usuario. La sección 3 presenta la descripción de las características seleccionadas para desarrollar el modelo de clasificación. La discusión acerca de los resultados obtenidos se presenta en la sección 4. Finalmente la conclusión del presente trabajo de investigación se realiza en la sección 5.

2. Trabajo relacionado

Se realizó un estudio sobre los trabajos desarrollados en esta área, enfatizando sus avances, alcance, enfoques, ventajas y desventajas, así como sus aportaciones científicas, encontrando el siguiente panorama general:

En la propuesta presentada en [1] se desarrollan dos modelos uno para el idioma español y otro para el idioma inglés, ambos totalmente diferentes. Para el idioma inglés se extrajeron características léxicas y sintácticas, sin embargo para el idioma español, se realizó una representación mediante grafos de las conversaciones y se extrajeron los patrones de cada clase utilizando la herramienta SUBDUE[2]. Se reporta que los resultados para el idioma inglés superaron considerablemente los resultados obtenidos para el idioma español.

En la investigación desarrollada en [3] se proponen 2 tipos de características que pueden ser usadas para esta tarea. Características basadas en el contexto y características basadas en el estilo. Las características basadas en el estilo incluyen características léxicas y sintácticas utilizando Pos-tagger como etiquetador. Para las características relacionadas al contexto se extraen las 1000 palabras individuales con mayor frecuencia de un corpus que incluye 19 320 post extraídos de blogs escritos en inglés. Aplican además Información Mutua para detectar los pares de palabras que son colocaciones. Los resultados que obtuvieron muestran que las características estilográficas que más ayudan a discriminar el género son las preposiciones para el caso de los hombres y los pronombres para el caso de las mujeres. Y con respecto al contexto, los hombres utilizan palabras relacionadas con la tecnología y las mujeres utilizan más palabras relacionadas a la vida personal y a las relaciones.

El modelo propuesto en [8] se basa en el desarrollo de una variación del algoritmo *Exponential Gradient (EG)*, que permite detectar el género de un autor. Se propone una representación vectorial del conjunto de características que estudian y en cada paso bajo ciertos criterios de eliminación van reduciendo el espacio de representación, quitando aquellas características que aportan poco a la detección del género. Concluyen que las características más representativas son las palabras y las etiquetas de los textos.

En el trabajo desarrollado en [13] se estudia el comportamiento de hombres y mujeres blogueros, y mencionan que las características que mejores resultados ofrecieron son las palabras representativas de cada grupo, los hiperenlaces y palabras comúnmente usadas en los blogs (lol, haha, ur, entre otras).

[2] https://ailab.wsu.edu/subdue/

Los resultados que obtuvieron con estas características fueron del 80 % para el género y del 76 % para la edad. Se llegó a la conclusión de que las mujeres usan más pronombres y los hombres más preposiciones, también mencionan que se encontró que los blogs escritos por adolescentes son en su mayoría mujeres, que las mujeres hablan más sobre su vida privada y familia, mientras que los hombres hablan más sobre tecnología y política.

En otros trabajos precedentes para abordar esta tarea se puede observar, que las características más comúnmente utilizadas son:

- N gramas de palabras, [4],[9],[10] y [11].
- N gramas de caracteres, [4] y [11].
- Longitud de palabras, [5], [9] y [14].
- Longitud de oraciones, [6], [7] y [14].

En [10] y [9] se utiliza la herramienta *Linguistic Inquiry and Word Count (LIWC)*, la cual calcula el grado en que las personas usan diferentes categorías entre un conjunto de documentos, también se puede determinar el grado en el que un texto utiliza emociones positivas o negativas entre otras cosas. Además de las características mencionadas anteriormente, en el trabajo propuesto en [9] también se cuenta la frecuencia de uso de palabras en mayúscula, la frecuencia de uso de intensificadores y la longitud de las oraciones. En [12] las características que se usan son las mencionadas anteriormente y se agregan el uso de signos de puntuación, el uso de emoticones y el uso de las categorías gramaticales POS.

En el trabajo propuesto en [6] se utiliza la frecuencia de las clases a las que pertenecen las palabras. La clasificación de las palabras se realiza con la herramienta *RiTaWordNet* la cual establece la relación de una palabra con su clase mediante sinónimos e hiperónimos. Posteriormente se clasifican las palabras en positivas o negativas usando *SentiWordNet 3.0*, se cuenta los signos de puntuación usados, la frecuencia de las palabras cerradas, frecuencia de uso de pronombres, se reemplazan los emoticones por su palabra equivalente y se cuantifica una lista de palabras foráneas (meee, yesss, thy, u, urs, entre otras).

El modelo propuesto en [5] utiliza algunas de las siguientes características, la frecuencia de uso de palabras escritas en formato *CamelCase* y la frecuencia de uso de etiquetas POS. También comentan que las personas jóvenes utilizan más los pronombres en primera persona y que las personas que no son originarias de los Estados Unidos usan más las abreviaciones "u" y "ur". Mencionan que al igual que en [3] se aplica Información Mutua para detectar los pares de palabras que son colocaciones.

Otro trabajo que es importante destacar es el presentado en [5], donde se mencionan algunas características interesantes como son los determinantes (*a, the, that, these*) y cuantificadores (*one, two, more, some*), que sirven como indicadores para identificar a un hombre y una vez más se menciona que los pronombres (*I, you, she, her, their, myself, yourself, herself*) son indicadores para identificar a una mujer, ya que según los autores las mujeres tienden a personalizar más los textos que escriben, mientras que los hombres los generalizan.

En el trabajo desarrollado por [2] se presenta un metodología para detectar el perfil de un autor, en particular se considera edad y género. Las características

que utilizan son: categorías gramaticales, palabras cerradas, sufijos y signos. Los autores logran detectar solamente en un 55 % el género y a lo sumo un 45 % la edad. En este trabajo solo se presentan los resultados para el corpus de blogs en el idioma inglés y el idioma español ofrecido en la conferencia PAN 2013.

Entre las técnicas de clasificación más usadas se encuentran: Naive Bayes, que ha sido reportada en los trabajos desarrollados en [4],[14] y [7] y las máquinas de soporte vectorial (SMV) que han sido utilizadas en las investigaciones desarrolladas en [11], [4], [14] y [12].

Las características usadas fundamentalmente han sido léxicas, sintácticas y conteos de las frecuencias de uso de algunos elementos. En la presente investigación se pretende proponer características que de alguna manera permitan detectar el sentido del texto que se está estudiando y con ello analizar si es posible descubrir el perfil del autor.

Es importante destacar que es más simple detectar el género, que la edad, pues los hombres y las mujeres escriben o se interesan por temas diferentes independientemente de la edad que tienen. Un aspecto importante a estudiar es la técnica de clasificación que se debe usar.

3. Descripción del enfoque propuesto

Se ha desarrollado un modelo supervisado, donde se extraen un conjunto de características del corpus de entrenamiento considerando cuantas veces aparecen en este o su probabilidad de aparición. A continuación se detallan las 3 fases que permiten construir este modelo.

3.1. Preprocesamiento del corpus

Debido a que el corpus con el que se trabaja es descargado directamente de la página del PAN, es necesario hacer varias operaciones antes de poder trabajar con él, algunas de ellas son:

1. Separar el corpus por autor.
2. Separar el corpus por género.
3. Sustituir los símbolos HTML que pueda contener el texto, por su equivalente en utf8.

Para el último punto se desarrolló un diccionario de símbolos HTML.

3.2. Características seleccionadas

En el enfoque propuesto se emplea un modelo supervisado basado en máquinas de aprendizaje, para el cual se construye un modelo de clasificación usando los siguientes conjuntos de características, obtenidas de los documentos de cada autor del corpus de entrenamiento proporcionado para esta tarea:

1. Número de slangs.
2. Número de contracciones.
3. Número de prefijos.
4. Número de signos.
5. Número de links
6. Número de imágenes
7. Número de palabras mal escritas.
8. Longitud de la oración.
9. Número de números.
10. Número de palabras que empiezan con mayúscula.
11. Número de palabras escritas en mayúscula.
12. Longitud de la palabra más larga.
13. Número de palabras de longitud 1, 2,10 y 15.
14. 39 categorías gramaticales.
15. Conteo de las 200 palabras más frecuentes.
16. Probabilidad de cada palabra del vocabulario (unigrama).
17. Bolsa de palabras.

El primero conjunto está conformado por las primeras 13 características que se muestran en la lista, se realiza un conteo para determinar el número de veces que aparece cada característica en un documento. Se decide cuantificar las palabras de longitud 1, 2, 10 y 15 con el objetivo de detectar algún patrón que permitiera separar por grupos de edad y género, ya que son extremos, es decir, palabras muy cortas y muy largas. Se desarrollaron diferentes recursos léxicos para realizar estos conteos como son: un diccionario de *slangs*, un diccionario de signos, diccionario de contracciones, diccionario de prefijos y un diccionario que nos permite detectar si la palabra ha sido mal escrita.

Para el segundo conjunto de características, se creó un diccionario de categorías gramaticales las cuales se extrajeron del corpus después de ser etiquetado con la herramienta de Clips llamada pattern.en[3]. Posteriormente se utilizó el diccionario para contar el número de veces que aparece cada categoría gramatical en el documento.

Para la característica número 15 (conteo de las 200 palabras más frecuentes) se hizo un análisis del corpus de hombres y otro del corpus de mujeres para identificar cuáles son las 100 palabras más frecuentes de cada uno, descartando las palabras cerradas y las palabras que se repiten en ambos corpus. Algunas de las palabras que se extrajeron del corpus de blogs en inglés se pueden observar en la Tabla 1.

[3] www.clips.ua.ac.be/pages/pattern-en

Tabla 1. Palabras más frecuentes.

Corpus mujeres	Corpus hombres
Art	Banks
Design	Building
Diet	Development
Exercise	Economy
Food	Financial
Gallery	Goverment
Health	Information
Heart	Job
Home	Money
Personal	Payment

Para calcular la probabilidad de los unigramas, se eliminaron las palabras cerradas y símbolos y se calculó la probabilidad de aparición de cada palabra del corpus, las cuales son aproximadamente 20 mil palabras. Este conjunto de palabras se agregan al vector de la siguiente forma: si la palabra aparece en el documento, se pone la probabilidad calculada anteriormente para esa palabra, si la palabra no aparece en ese documento, el valor que se pone en el vector para esa palabra será cero.

Por último se agrega el texto de cada autor como una bolsa de palabras.

3.3. Representación de las características

Todas las características mencionadas en la sección 3.2 permiten construir un vector representativo de cada autor considerando ya sea la frecuencia o la probabilidad de aparición de cada una de las características seleccionadas.

Para la fase de entrenamiento, un ejemplo de dicho vector se muestra en la figura 1 donde el campo con el valor *Clase* al final del vector, es el atributo clasificador del documento que en el caso del género podría indicar si el documento pertenece a una mujer o a un hombre.

Fig. 1. Vector de entrenamiento.

Para la fase de prueba se utiliza un vector de características como se muestra en la en la figura 2, donde el atributo clasificador se sustituye por un signo de interrogación ya que se desconoce la clase a la que pertenece.

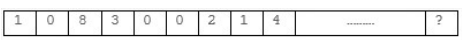


Fig. 2. Vector de prueba.

3.4. Proceso de clasificación

El modelo descrito anteriormente se puede observar en la figura 3, en ella se muestra el proceso que se sigue. Este se ha dividido en dos fases, en la primera fase se realiza el pre procesamiento descrito en la sección 3.1 y después se extraen las características descritas en la sección 3.2 en donde el atributo clasificador será el género del autor. Por último se envía a Weka el conjunto de vectores característicos que sirven para crear el *Modelo de clasificación por género*.

En la segunda fase se utiliza el mismo conjunto de vectores característicos que en la fase anterior, pero el atributo clasificador ahora será el rango de edad del autor. Aquí se crean dos modelos de clasificación diferentes, el *Modelo de clasificación de edadMujer* y el *Modelo de clasificación de edadHombre*. Como ya se sabe de antemano a que género pertenecen los documentos gracias a la fase anterior, se pueden discriminar los documentos para que a cada modelo solo entren vectores que correspondan a ese género.

Como paso final se evalúan los resultados de los modelos.

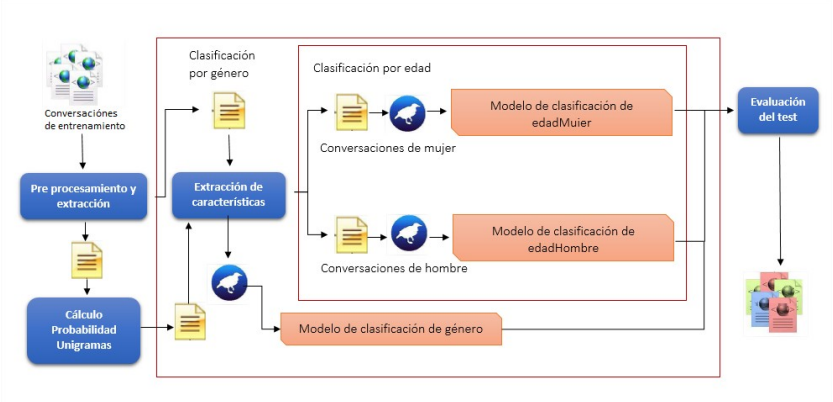


Fig. 3. Descripción del modelo.

4. Resultados

A continuación se muestra una descripción de los corpus que se utilizaron para el entrenamiento del modelo propuesto y posteriormente se muestran los experimentos realizados, así como los resultados de cada experimento.

4.1. Conjunto de datos

Se trabajó con cuatro corpus en el idioma inglés, los cuales contienen documentos de fuentes como Blogs, Criticas, Redes Sociales y Twitter. También se

contó con dos corpus en idioma español con documentos de Blogs y de Redes Sociales. Todos los corpus fueron obtenidos del sitio web del PAN 2014. Se proporciona una breve descripción en los siguientes tablas en donde se muestra el número de instancias con las que se cuenta en las diferentes corpus. La columna Autores representa el número de autores, las columnas 18-24, 25-34, 35-49, 50-64 y 65+ representan los rangos de edad de estos autores.

Tabla 2. Número de instancias por corpus y por clase para el idioma inglés.

Género	Autores	18-24	25-34	35-49	50-64	65+
Hombre(blog)	74	3	30	27	12	2
Mujer(blog)	73	3	30	27	11	2
Hombre(review)	2,080	180	500	500	500	400
Mujer(review)	2,080	180	500	500	500	400
Hombre(socialmedia)	3,529	693	945	1035	851	5
Mujer(socialmedia)	3,503	699	944	1025	828	7
Hombre(twitter)	149	9	44	63	29	4
Mujer(twitter)	152	10	43	65	30	4

Tabla 3. Número de instancias por corpus y por clase para el idioma español.

Género	Autores	18-24	25-34	35-49	50-64	65+
Hombre(blog)	44	2	13	21	6	2
Mujer(blog)	44	2	13	21	6	2
Hombre(socialmedia)	636	165	213	162	80	16
Mujer(socialmedia)	636	165	213	162	80	16

4.2. Experimentos

En las siguientes tablas se muestra un resumen de los mejores resultados obtenidos por corpus y por lenguaje. Todos los experimentos se hicieron aplicando validación cruzada de 10 pliegues y se utilizaron los algoritmos de clasificación vecinos mas cercanos (IBk), máquinas de soporte vectorial (SVM) y Naïve Bayes sobre el conjunto de características escogidas, las cuales fueron descritas en la sección 3.2. Se muestra en **negritas** los mejores resultados para cada idioma. Se puede observar que el algoritmo de clasificación que mejor comportamiento mostró fue la máquina de soporte vectorial, con polikernel . Los resultados varían de acuerdo al corpus. Esto nos indica que la forma en que se escribe en cada uno de ellos es diferente, ya que se han utilizado las mismas características. La detección del género en los corpus de Blogs, Twitter y Criticas superó el 80 %, sin embargo en las Redes Sociales las características escogidas no permitieron detectar fácilmente si el texto fue escrito por un hombre o una mujer.

Se realizaron diversos experimentos con el objetivo de analizar las características que nos permiten diferenciar cada una de las clases; llegando a la conclusión que la probabilidad de cada palabra le permite al clasificador poder detectar el género de la persona que ha escrito el texto. La detección de la edad es más compleja, debido a que el corpus no está balanceado y se dispone de 5 clases. Esto nos condujo a obtener resultados que no superan el 70 % de precisión.

Tabla 4. Resultados para los corpus en inglés.

Corpus	Tipo	Clase	Características	Clasificador	Resultado
BLOGS		Género	254 + Probabilidades + Texto	SMO	87.07
	Female	Edad	254 + Probabilidades + Texto	SMO	43.83
	Male	Edad	254 + Texto	SMO	**54.05**
CRITICAS		Género	254 + Probabilidades + Texto	SMO	**99.87**
	Female	Edad	254 + Probabilidades	SMO	30.76
	Male	Edad	54	SMO	29.61
REDES		Género	254	SMO	52.38
	Female	Edad	254	SMO	37.91
	Male	Edad	54	SMO	36.78
TWITTER		Género	254 + Texto	SMO	83.05
	Female	Edad	54	SMO	**48.29**
	Male	Edad	254 + Texto	SMO	49.48

Tabla 5. Resultados para los corpus en español.

Corpus	Tipo	Clase	Características	Clasificador	Resultado
BLOGS		Género	254 + Texto	SMO	**79.54**
	Female	Edad	254 + Texto	SMO	**68.18**
	Male	Edad	254	SMO	**45.45**
REDES		Género	254	SMO	60.61
	Female	Edad	254 + Texto	SMO	40.72
	Male	Edad	254 + Texto	SMO	38.36

5. Conclusiones

Se desarrolló un modelo para la detección del perfil de un autor (género y edad). Se pudo observar que el comportamiento del modelo propuesto fue similar para ambos idiomas y que los resultados obtenidos para varios corpus ofrecidos en la Conferencia Internacional PAN 2014 fueron satisfactorios para la detección del género, sin embargo es necesario incluir características diferentes para la detección de la edad.

Como trabajo futuro se propone desarrollar una representación de los textos mediante grafos de co-ocurrencia y calcular las medidas de centralidad que posee la herramienta gephi[4], con el objetivo de obtener las palabras representativas de cada clase y que puedan ser incluídas en el modelo de clasificación.

[4] http://gephi.github.io/

Referencias

1. Aleman, Y., Loya, N., Vilariño, D.: Two methodologies applied to the author. PAN 2013 (2014)
2. Aleman, Y., Vilariño, D., Pinto, D.: Avances en la ingeniería del lenguaje y del conocimiento. Research in Computer Science 85, 93–103 (2014)
3. Argamon, S., Koppel, M., J., P., Schler, J.: Automatically profiling the author of an anonymous text. Communications of the ACM - Inspiring Women in Computing 52(2), 119–123 (2009)
4. Burger, J.D., Henderson, J., Kim, G., G., Z.: Discriminating gender on twitter. In: EMNLP '11 Proceedings of the Conference on Empirical Methods in Natural Language Processing. pp. 1301–1309 (2011)
5. Estival, D., Gaustad, T., Pham, S.B., Radford, W., Hutchinson, B.: Author profiling for english emails. In: Proceedings of the 10th Conference of the Pacific Association for Computational Linguistics. pp. 263–272 (2007)
6. Gopal, P., Banerjee, S., Das, D.: Automatic author profiling based on linguistic and stylistic features. In: Proceedings of the 9th PAN at CLEF Conference (2013)
7. Goswami, S., Sarkar, S., Rustagi, M., Meder, T.: Stylometric analysis of bloggers age and gender. In: Proceedings of the Third International ICWSM Conference (2013)
8. Koppel, M., Argamon, S., A., S.: Automatically categorizing written texts by author gender. Literary and Linguistic Computing 17(4), 401–412 (2002)
9. Nguyen, D., Gravel, R., Trieschnigg, D., Meder, T.: "how old do you think i am?"; a study of language and age in twitter. In: Proceedings of the Seventh International AAAI Conference on Weblogs and Social Media (2013)
10. Nguyen, D., Smith, N., Rosé, C.: Author age prediction from text using linear regression. In: LaTeCH '11 Proceedings of the 5th ACL-HLT Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities. pp. 115–123 (2011)
11. Peersman, C., Daelemans, W., Vaerenbergh, L.V.: Predicting age and gender in online social networks. In: SMUC '11 Proceedings of the 3rd international workshop on Search and mining user-generated contents. pp. 37–44 (2011)
12. Rangel, F., Rosso, P.: Use of language and author profiling: Identification of gender and age. http://users.dsic.upv.es/ prosso/resources/RangelRosso_NLPCS13.pdf (2009)
13. Schler, J., Koppel, M., S., A., J., P.: Effects of age and gender on blogging. In: Proceedings of the AAAI Spring Symposium on Computational (2006)
14. Zhang, C., Zhang, P.: Predicting gender from blog posts. http://people.cs.umass.edu/ pyzhang/course/genderClassify.pdf (2010)

Detección de subjetividad en noticias en línea publicadas en español utilizando clasificadores probabilísticos

Noé Alejandro Castro-Sánchez[1], Sadher Abelardo Vázquez-Cámara[1] y Grigori Sidorov[2]

[1] Centro Nacional de Investigación y Desarrollo Tecnológico, México

[2] Instituto Politécnico Nacional, Centro de Investigación en Computación, México

{ncastro, sadhervazquez}@cenidet.edu.mx, sidorov@cic.ipn.mx

Resumen. Los textos periodísticos pueden clasificarse dentro del llamado género informativo si su contenido se orienta a la objetividad (descripción de los hechos ocurridos), o en el género de opinión, si incluye elementos subjetivos (como el punto de vista o ideología del autor de la nota). Uno de los problemas que se presenta en la redacción de noticias es que en las notas de tipo informativo se llegan a incorporar elementos subjetivos sin previa advertencia al lector. En este artículo se presenta un método para la detección automática de subjetividad en oraciones de noticias escritas en español. Se generó un corpus a partir de noticias publicadas en internet, las cuales contienen 8,108 oraciones que se clasificaron manualmente como objetivas (3,648) y subjetivas (4,460). Los mejores resultados obtenidos a partir de experimentos con diversos clasificadores automáticos arrojan un 76.3% de precisión, utilizando el clasificador *Bayes Net*.

Palabras clave: detección automática de subjetividad, detección de opinión, corpus de noticias, Naive Bayes, Bayes Net, Weka.

1. Introducción

La cantidad de usuarios de internet en México es de aproximadamente 51.2 millones de personas (más de la mitad de la población), las cuales en su mayoría oscilan en las edades entre 19 y 44 años. Entre los principales usos de internet en México se encuentra la recepción y envío de correos electrónicos, la búsqueda de información y el acceso a redes sociales [1].

Actualmente, México se encuentra en un cambio en cuanto a preferencias de uso de medios de comunicación, en donde dominaba el uso de la televisión y el uso de Internet se encuentra en constante aumento. El uso de internet es la principal fuente de noticias e información, seguida de periódicos, televisión y radio [2].

La subjetividad en cuanto al lenguaje natural hace referencia a aspectos del lenguaje utilizados para expresar opiniones y evaluaciones [3]. El lenguaje subjetivo es un tipo

de lenguaje utilizado para expresar estados privados, los cuales son términos que cubren opiniones, evaluaciones, emociones y especulaciones [4].

De manera general, los géneros periodísticos pueden ser clasificados como géneros informativos, en donde la información de algún hecho o dato se presenta tal y como ha ocurrido, dominando el uso impersonal y objetivo del lenguaje, y los géneros de opinión, en donde el escritor expresa su punto de vista acerca de un hecho o dato, dominando la subjetividad [5].

El papel que juegan los medios de comunicación en la política de México es reciente, y estos desempeñan dos papeles esenciales: Funcionan como diseminadores de información, lo cual es pieza clave en toda democracia y pueden movilizar la opinión pública, así como generar diferentes formas de actividad política [6].

Uno de los problemas que se presentan en el periodismo es que existen noticias que se publican dentro del género informativo, aunque en realidad incluyen opiniones de los escritores de la nota.

El objetivo de este trabajo, es desarrollar un método de detección de subjetividad en oraciones de noticias. Este método consiste en realizar una clasificación automática utilizando clasificadores probabilísticos y un corpus de noticias publicadas en español de México.

En este artículo, se presentan las pruebas y resultados obtenidos al utilizar los clasificadores probabilísticos con el corpus etiquetado de manera manual, realizando diversos experimentos, con el fin de determinar cuál es el mejor clasificador y el mejor conjunto de características.

El artículo se encuentra organizado de la siguiente manera: En la sección 2, se presentan los trabajos relacionados, en la sección 3 se presenta nuestra metodología de solución, describiendo el proceso realizado, presentando los resultados obtenidos en la sección 4. Para finalizar, se presentan las conclusiones y el trabajo a futuro en la sección 5.

2. Trabajos relacionados

Los trabajos que implementan diversas técnicas para la detección de subjetividad son tratados a continuación.

En el trabajo [7], se presenta un marco de trabajo utilizado para identificar declaraciones subjetivas en títulos de noticias, utilizando los sentidos de palabras para identificar el significado (objetividad) y sentido (subjetividad) de cada oración, además de determinar la emoción expresada. Los resultados de precisión de este trabajo muestran un 73% de precisión, aunque al combinarse con características extras, puede llegar a obtenerse un 99%.

En el trabajo [8], se propone realizar minería de opinión para detectar opiniones en columnas de noticias de género político publicadas en idioma tailandés. El trabajo se compone de 3 partes: Colección de datos (limpieza y almacenamiento de información), anotación (etiquetado manual por humanos) y clasificación (minería de datos con *Weka*). Los mejores resultados muestran una precisión del 80.7% al utilizar el clasificador *Naive Bayes*.

Otro trabajo es *OpinionFinder* [9], un sistema de análisis de subjetividad que identifica de manera automática cuando existen opiniones, sentimientos, especulaciones y otros estados privados en el texto. Opera en 2 partes: En la primera parte realiza un procesamiento del documento y en la segunda parte, se realiza el análisis de subjetividad, que consta de 4 componentes: clasificación de sentencias subjetivas, eventos del discurso y clasificación de expresión subjetiva directa, identificación del origen de opinión y por último, clasificación de la expresión del sentimiento.

En el trabajo [10], se presenta *SubjLDA*, un modelo jerárquico bayesiano basado en Asignación *Dirichlet* latente (*Latent Dirichlet Allocation*, *LDA*), para la detección de subjetividad a nivel de oraciones, el cual automáticamente identifica si una oración dada expresa opiniones o si expresa hechos. El proceso generativo involucra tres etiquetas de subjetividad para las sentencias, una etiqueta de sentimientos para cada palabra en la oración y las palabras en las oraciones. El algoritmo de *subjLDA* es un modelo bayesiano de cuatro capas y la clasificación de la subjetividad en la sentencia es determinada directamente desde la etiqueta de subjetividad de la sentencia. En los resultados obtenidos, se demostró que *subjLDA* obtuvo un porcentaje de precisión del 71.6% al analizar sentencias objetivas y un 71% al analizar sentencias subjetivas, dando un resultado final de precisión de precisión de 71.2%.

3. Método propuesto

Nuestro trabajo propone un método de detección de subjetividad de noticias utilizando un clasificador automático y tomando características a nivel de oración y n-gramas. La figura 1 muestra la arquitectura del método:

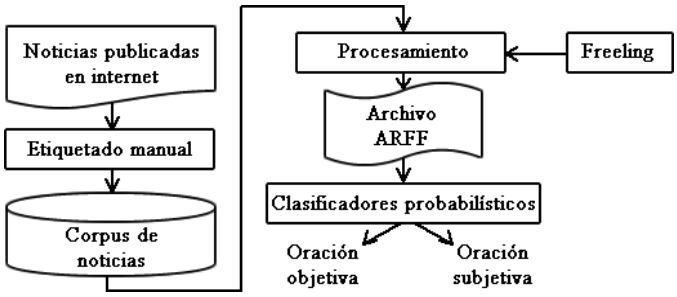


Fig. 1. Arquitectura del método de detección de subjetividad.

Primero, se obtienen noticias publicadas en internet, las cuales son etiquetadas de manera manual a nivel de oración por etiquetadores humanos, con el fin de generar el corpus de noticias. Después, se realiza el procesamiento de texto (generación de n-gramas, lematización) para crear el archivo ARFF que servirá de entrenamiento a los clasificadores probabilísticos. Finalmente, se realizan experimentos con diferentes clasificadores para determinar con cuál de estos se obtienen los mejores resultados de detección de objetividad y subjetividad en las oraciones.

3.1. Creación del corpus de noticias

Se creó un corpus de noticias escritas en español obtenidas de diversos sitios periodísticos mexicanos, con el fin de utilizarlas como datos de entrenamiento para los clasificadores probabilísticos. El corpus cuenta con un total de 1972 noticias, las cuales fueron divididas en 2 secciones: Noticias informativas, con un total de 1834 y noticias de opinión con un total de 138. Las notas informativas se conforman por noticias relacionadas a temas de interés sociopolítico (por ejemplo, política, seguridad y economía), descartando notas asociadas a entretenimiento (espectáculos, deportes, etc). Las oraciones de las noticias informativas fueron etiquetadas de manera manual a nivel de oración. Los etiquetadores recibieron una inducción acerca de objetividad y subjetividad, la cual se basó en un manual de etiquetado, desarrollado con base en el trabajo de [11] y [4]. Por otro lado, todas las oraciones de las noticias de opinión fueron etiquetadas como subjetivas. Algunas de las características que se tomaron en cuenta para determinar si una oración es subjetiva, son la modalidad oracional, léxico valorativo, signos de puntuación, entre otros.

Las oraciones de las noticias ya clasificadas arrojaron un total de 3648 oraciones objetivas y 4460 oraciones subjetivas.

3.2. Preprocesamiento de datos para experimentos

Para poder realizar los experimentos con clasificadores probabilísticos, se etiquetó el corpus de manera manual a nivel de oración, obteniendo un total de 8108 oraciones, de las cuales 3648 fueron clasificadas como objetivas y 4460 como subjetivas. Para realizar las pruebas, se generó un único archivo de extensión ARFF con el contenido de todas las notas etiquetadas, con el fin de utilizar los clasificadores que provee el software de minería de datos *Weka* [12]. El formato del archivo ARFF es el siguiente: el atributo "clasificación" indica si la oración fue etiquetada como objetiva (O) o subjetiva (S), mientras que el atributo "texto" contiene la oración o el n-grama, dependiendo del caso.

```
@relation OBJ_SUB
@attribute clasificacion {O,S}
@attribute texto String
@data
O, 'El caso inició el sábado 25 de octubre, cuando la joven
solicitó vía telefónica el apoyo de las autoridades'
S, 'Dicen que el funcionario público es el verdadero culpable
del incidente'
```

Durante el preprocesamiento de los datos, se utilizó el filtro *StringToWordVector*, el cual se encarga de convertir atributos de tipo cadena (*String*) en un conjunto de atributos, los cuales representan la ocurrencia de palabras dentro del texto contenido en la cadena. Los clasificadores utilizados en este trabajo de investigación fueron *Naive Bayes* y *Bayes Net*, pues según consta en la bibliografía son los que mejores resultados arrojan [13] [14]. Se realizaron diversos experimentos para identificar las mejores

características para los clasificadores, los cuales pueden dividirse de la siguiente manera según las diferentes combinaciones derivadas del tratamiento del texto:

1. Utilización de oraciones.
   (a) Inclusión de *stopwords* (palabras auxiliares) con texto sin lematizar,
   (b) Inclusión de *stopwords* con texto lematizado,
   (c) Eliminación de *stopwords* con texto sin lematizar,
   (d) Eliminación de *stopwords* con texto lematizado.

2. Segmentación de las oraciones en bigramas, trigramas y 4-gramas, a los cuales se les asigna automáticamente la etiqueta de la oración de donde son extraídos. El procesamiento realizado en el punto anterior también se aplicó para cada uno de los siguientes criterios:
   (a) Eliminación de n-gramas repetidos, dejando un único n-grama,
   (b) Eliminación de n-gramas etiquetados como subjetivos, que estuvieran etiquetados también como objetivos,
   (c) Eliminación de n-gramas etiquetados como objetivos, que estuvieran etiquetados también como subjetivos,
   (d) Eliminación de n-gramas etiquetados tanto como objetivos y subjetivos.

4. Resultados de experimentos

A continuación, se muestran los resultados obtenidos utilizando los criterios indicados a nivel oración:

En la tabla 1, podemos observar los resultados obtenidos al utilizar oraciones con el clasificador *Naive Bayes*. La mayor precisión al determinar objetividad fue 65%, en cuanto a subjetividad, la mayor precisión fue 74%. Las características que se consideraron fueron: Incluyendo *stopwords* (CSW), Eliminando *stopwords* (SSW), texto sin lematizar (TNL) y texto lematizado (TL).

Tabla 1. Resultados de experimentos utilizando oraciones con el clasificador *Naive Bayes*.

	Precisión		**Recall**		**Medida·F**	
	O	**S**	**O**	**S**	**O**	**S**
CSW, TNL	**0.65**	0.73	0.68	0.70	0.66	0.71
CSW, TL	0.60	0.71	0.68	0.62	0.64	0.66
SSW, TNL	0.63	**0.74**	0.71	0.67	0.67	0.70
SSW, TL	0.60	0.73	0.72	0.60	0.66	0.66

En la tabla 2, se observan los resultados obtenidos al utilizar oraciones con el clasificador *Bayes-Net*. La mayor precisión al determinar fue 63%, y en subjetividad fue 70%. Las características que se tomaron fueron: Incluyendo *stopwords* (CSW), Eliminando *stopwords* (SSW), texto sin lematizar (TNL) y texto lematizado (TL).

Tabla 2. Resultados de experimentos utilizando oraciones con el clasificador *Bayes Net*.

	Precisión		**Recall**		**Medida·F**	
	O	**S**	**O**	**S**	**O**	**S**
CSW, TNL	0.60	0.66	0.57	0.69	0.58	0.68
CSW, TL	**0.63**	0.59	0.28	0.86	0.39	0.70
SSW, TNL	0.61	**0.70**	0.66	0.65	0.63	0.67
SSW, TL	0.59	0.59	0.28	0.84	0.38	0.69

Después, se realizaron experimentos, eliminando n-gramas repetidos, dejando solamente un n-grama en el archivo ARFF.

En la tabla 3, se puede observar los resultados obtenidos al utilizar el clasificador *Naive Bayes.* La mayor precisión al determinar objetividad fue de 63% y al determinar subjetividad la mayor precisión fue de 67%. En este caso, las características fueron: Incluyendo *stopwords* (CSW), Eliminando *stopwords* (SSW), texto sin lematizar (TNL) y texto lematizado (TL).

Tabla 3. Resultados de experimentos utilizando n-gramas, eliminando n-gramas repetidos con el clasificador *NaiveBayes*.

	Precisión		**Recall**		**Medida·F**	
	O	**S**	**O**	**S**	**O**	**S**
Bigramas, CSW, TNL	0.52	0.53	0.36	0.68	0.42	0.60
Bigramas, CSW, TL	0.62	0.52	0.11	0.93	0.18	0.67
Bigramas, SSW, TNL	0.53	0.52	0.29	0.74	0.38	0.61
Bigramas, SSW, TL	**0.63**	0.57	0.42	0.76	0.50	0.65
Trigramas, CSW, TNL	0.54	0.55	0.52	0.56	0.53	0.55
Trigramas, CSW, TL	0.56	**0.67**	0.79	0.40	0.66	0.50
Trigramas, SSW, TNL	0.54	0.54	0.48	0.60	0.51	0.57
Trigramas, SSW, TL	0.56	0.63	0.73	0.45	0.63	0.52
4-gramas, CSW, TNL	0.56	0.56	0.58	0.54	0.57	0.55
4-gramas, CSW, TL	0.55	0.66	0.80	0.37	0.66	0.47
4-gramas, SSW, TNL	0.56	0.57	0.60	0.52	0.58	0.54
4-gramas, SSW, TL	0.57	0.62	0.69	0.49	0.62	0.55

En la tabla 4, se observan los resultados al utilizar el clasificador *Bayes Net*. La mayor precisión al determinar objetividad fue de 60%, en subjetividad, la mayor

precisión fue de 76%. Las características consideradas fueron: Incluyendo *stopwords* (CSW), Eliminando *stopwords* (SSW), texto sin lematizar (TNL) y texto lematizado (TL).

Tabla 4. Resultados de experimentos utilizando n-gramas, eliminando n-gramas repetidos con el clasificador *Bayes Net.*

	Precisión		**Recall**		**Medida·F**	
	O	**S**	**O**	**S**	**O**	**S**
Bigramas, CSW, TNL	0.58	0.51	0.07	0.95	0.12	0.67
Bigramas, CSW, TL	0.50	0.51	0.20	0.80	0.29	0.63
Bigramas, SSW, TNL	0.55	0.51	0.06	0.95	0.11	0.66
Bigramas, SSW, TL	0.50	**0.76**	0.98	0.04	0.66	0.09
Trigramas, CSW, TNL	0.52	0.58	0.75	0.33	0.62	0.42
Trigramas, CSW, TL	0.54	0.68	0.86	0.29	0.66	0.41
Trigramas, SSW, TNL	0.53	0.60	0.78	0.33	0.63	0.42
Trigramas, SSW, TL	0.55	0.64	0.76	0.40	0.64	0.50
4-gramas, CSW, TNL	0.55	0.65	0.81	0.34	0.66	0.45
4-gramas, CSW, TL	0.57	0.69	0.81	0.41	0.67	0.51
4-gramas, SSW, TNL	0.56	0.63	0.75	0.42	0.64	0.50
4-gramas, SSW, TL	**0.60**	0.65	0.70	0.54	0.65	0.59

Posteriormente, se realizaron dos clases de experimentos. En la primera, si algún n-grama se encontraba etiquetado, tanto objetivo como subjetivo, el n-grama objetivo fue eliminado, y en la segunda, si algún n-grama se encontraba etiquetado tanto objetivo como subjetivo, se eliminó el n-grama subjetivo.

A continuación, se muestran los resultados de los experimentos al eliminar los n-gramas clasificados como objetivos.

En la tabla 5, se muestran los resultados obtenidos con *Naive Bayes*. La mayor precisión determinando objetividad es de 61%, en cuanto a subjetividad, la mayor precisión fue de 68%.

Posteriormente, en la tabla 6, se observan los resultados obtenidos con el clasificador *Bayes Net*. La mayor precisión en objetividad fue de 73%, en subjetividad, la mayor precisión fue de 69%.

Las características a considerar en ambos experimentos fueron: Incluyendo *stopwords* (CSW), Eliminando *stopwords* (SSW), texto sin lematizar (TNL) y texto lematizado (TL).

Tabla 5. Resultados de experimentos utilizando n-gramas, eliminando n-gramas objetivos repetidos con el clasificador *Naive Bayes*.

	Precisión		**Recall**		**Medida·F**	
	O	**S**	**O**	**S**	**O**	**S**
Bigramas, CSW, TNL	0.50	0.56	0.13	0.89	0.21	0.68
Bigramas, CSW, TL	0.52	0.53	0.04	0.96	0.08	0.64
Bigramas, SSW, TNL	0.50	0.56	0.12	0.90	0.19	0.69
Bigramas, SSW, TL	**0.61**	0.53	0.04	0.97	0.07	0.69
Trigramas, CSW, TNL	0.53	0.54	0.33	0.73	0.40	0.62
Trigramas, CSW, TL	0.56	**0.68**	0.78	0.43	0.66	0.52
Trigramas, SSW, TNL	0.53	0.54	0.29	0.77	0.37	0.64
Trigramas, SSW, TL	0.56	0.64	0.72	0.47	0.63	0.54
4-gramas, CSW, TNL	0.56	0.56	0.53	0.59	0.55	0.58
4-gramas, CSW, TL	0.55	0.66	0.80	0.38	0.66	0.48
4-gramas, SSW, TNL	0.55	0.57	0.55	0.57	0.55	0.57
4-gramas, SSW, TL	0.57	0.62	0.68	0.50	0.62	0.55

Tabla 6. Resultados de experimentos utilizando n-gramas, eliminando n-gramas objetivos repetidos con el clasificador *Bayes Net*.

	Precisión		**Recall**		**Medida·F**	
	O	**S**	**O**	**S**	**O**	**S**
Bigramas, CSW, TNL	0.54	0.55	0.07	0.95	0.12	0.70
Bigramas, CSW, TL	**0.73**	0.53	0.01	0.99	0.03	0.69
Bigramas, SSW, TNL	0.51	0.56	0.06	0.95	0.11	0.70
Bigramas, SSW, TL	0.66	0.52	0.00	0.99	0.01	0.69
Trigramas, CSW, TNL	0.62	0.53	0.13	0.92	0.21	0.68
Trigramas, CSW, TL	0.54	0.68	0.86	0.30	0.66	0.42
Trigramas, SSW, TNL	0.62	0.54	0.14	0.92	0.23	0.68
Trigramas, SSW, TL	0.55	0.65	0.76	0.42	0.64	0.51
4-gramas, CSW, TNL	0.55	0.63	0.78	0.37	0.64	0.47
4-gramas, CSW, TL	0.57	**0.69**	0.81	0.41	0.67	0.52
4-gramas, SSW, TNL	0.55	0.63	0.74	0.43	0.63	0.51
4-gramas, SSW, TL	0.60	0.65	0.71	0.53	0.65	0.59

A continuación, se muestran los resultados de los experimentos al eliminar los n-gramas clasificados como subjetivos:

En la tabla 7, se presentan los resultados obtenidos con el clasificador *Naive Bayes*. La mayor precisión en objetividad fue de 60% y en subjetividad fue de 67%.

Las características fueron: Incluyendo *stopwords* (CSW), Eliminando *stopwords* (SSW), texto sin lematizar (TNL) y texto lematizado (TL).

Tabla 7. Resultados de experimentos utilizando n-gramas, eliminando n-gramas subjetivos repetidos con el clasificador *Naive Bayes*.

	Precisión		**Recall**		**Medida·F**	
	O	**S**	**O**	**S**	**O**	**S**
Bigramas, CSW, TNL	**0.60**	0.57	0.69	0.48	0.64	0.52
Bigramas, CSW, TL	0.56	0.66	0.82	0.35	0.67	0.46
Bigramas, SSW, TNL	0.60	0.57	0.79	0.34	0.68	0.43
Bigramas, SSW, TL	0.54	0.59	0.87	0.20	0.66	0.30
Trigramas, CSW, TNL	0.55	0.54	0.61	0.48	0.58	0.51
Trigramas, CSW, TL	0.55	**0.67**	0.81	0.37	0.66	0.48
Trigramas, SSW, TNL	0.56	0.54	0.62	0.48	0.59	0.48
Trigramas, SSW, TL	0.56	0.62	0.73	0.44	0.64	0.51
4-gramas, CSW, TNL	0.56	0.56	0.60	0.52	0.58	0.54
4-gramas, CSW, TL	0.55	0.66	0.81	0.36	0.66	0.47
4-gramas, SSW, TNL	0.56	0.57	0.64	0.49	0.60	0.53
4-gramas, SSW, TL	0.57	0.61	0.69	0.49	0.62	0.55

En la tabla 8, se muestran los resultados obtenidos con el clasificador *Bayes Net Bayes*. La mayor precisión en objetividad fue de 60% y en subjetividad de 74%.

Las características fueron: Incluyendo *stopwords* (CSW), Eliminando *stopwords* (SSW), texto sin lematizar (TNL) y texto lematizado (TL).

Tabla 8. Resultados de experimentos utilizando n-gramas, eliminando n-gramas subjetivos repetidos con el clasificador *Bayes Net.*

	Precisión		**Recall**		**Medida·F**	
	O	**S**	**O**	**S**	**O**	**S**
Bigramas, CSW, TNL	0.54	0.70	0.98	0.05	0.70	0.09
Bigramas, CSW, TL	0.51	0.72	0.97	0.07	0.67	0.14
Bigramas, SSW, TNL	0.56	0.73	0.98	0.04	0.71	0.08
Bigramas, SSW, TL	0.52	**0.74**	0.98	0.09	0.05	0.09
Trigramas, CSW, TNL	0.54	0.64	0.87	0.23	0.67	0.34

	Precisión		Recall		Medida·F	
	O	**S**	**O**	**S**	**O**	**S**
Trigramas, CSW, TL	0.54	0.68	0.86	0.29	0.66	0.40
Trigramas, SSW, TNL	0.56	0.65	0.87	0.26	0.68	0.37
Trigramas, SSW, TL	0.56	0.63	0.76	0.41	0.64	0.50
4-gramas, CSW, TNL	0.56	0.65	0.81	0.35	0.66	0.45
4-gramas, CSW, TL	0.57	0.69	0.81	0.41	0.67	0.51
4-gramas, SSW, TNL	0.58	0.63	0.76	0.43	0.66	0.51
4-gramas, SSW, TL	**0.60**	0.65	0.70	0.54	0.65	0.59

Tabla 9. Resultados de experimentos utilizando n-gramas, eliminando n-gramas objetivos y subjetivos repetidos con el clasificador *Naive Bayes*.

	Precisión		Recall		Medida·F	
	O	**S**	**O**	**S**	**O**	**S**
Bigramas, CSW, TNL	0.53	0.53	0.33	0.72	0.41	0.61
Bigramas, CSW, TL	0.65	0.52	0.10	0.94	0.17	0.67
Bigramas, SSW, TNL	0.54	0.53	0.32	0.73	0.40	0.61
Bigramas, SSW, TL	**0.66**	0.57	0.39	0.80	0.49	0.67
Trigramas, CSW, TNL	0.54	0.55	0.51	0.57	0.53	0.56
Trigramas, CSW, TL	0.56	**0.68**	0.80	0.40	0.66	0.50
Trigramas, SSW, TNL	0.55	0.55	0.5	0.61	0.52	0.54
Trigramas, SSW, TL	0.56	0.63	0.73	0.45	0.63	0.53
4-gramas, CSW, TNL	0.56	0.56	0.58	0.54	0.57	0.55
4-gramas, CSW, TL	0.55	0.66	0.80	0.37	0.66	0.47
4-gramas, SSW, TNL	0.56	0.57	0.60	0.53	0.58	0.55
4-gramas, SSW, TL	0.57	0.62	0.68	0.50	0.62	0.55

Las pruebas finales, se realizaron de la siguiente manera; Si algún n-grama se encontraba etiquetado tanto objetivo como subjetivo, el n-grama fue eliminado, tanto el objetivo como el subjetivo. A continuación, se presentan los resultados:

En la tabla 9, podemos observar los resultados con el clasificador *NaiveBayes*. La mayor precisión fue 66%, en subjetividad, la mayor precisión fue 68%.

Para finalizar, en la tabla 10, podemos observar los resultados obtenidos con el clasificador *Bayes Net*. La mayor precisión en objetividad fue de 60%. En subjetividad, la mayor precisión fue de 69%.

Las características fueron: Incluyendo *stopwords* (CSW), Eliminando *stopwords* (SSW), texto sin lematizar (TNL) y texto lematizado (TL).

Tabla 10. Resultados de experimentos utilizando n-gramas, eliminando n-gramas objetivos y subjetivos repetidos con el clasificador *Bayes Net*.

	Precisión		**Recall**		**Medida·F**	
	O	**S**	**O**	**S**	**O**	**S**
Bigramas, CSW, TNL	0.59	0.52	0.07	0.95	0.13	0.67
Bigramas, CSW, TL	0.51	0.51	0.21	0.81	0.29	0.63
Bigramas, SSW, TNL	0.58	0.51	0.07	0.95	0.12	0.66
Bigramas, SSW, TL	0.50	0.75	0.97	0.07	0.66	0.13
Trigramas, CSW, TNL	0.52	0.65	0.86	0.24	0.65	0.35
Trigramas, CSW, TL	0.54	0.69	0.86	0.29	0.66	0.41
Trigramas, SSW, TNL	0.54	0.59	0.71	0.40	0.61	0.48
Trigramas, SSW, TL	0.55	0.65	0.77	0.41	0.64	0.50
4-gramas, CSW, TNL	0.55	0.65	0.81	0.34	0.66	0.45
4-gramas, CSW, TL	0.57	**0.69**	0.81	0.41	0.67	0.51
4-gramas, SSW, TNL	0.57	0.63	0.75	0.43	0.64	0.51
4-gramas, SSW, TL	**0.60**	0.65	0.70	0.54	0.65	0.59

5. Conclusiones y trabajo futuro

En este artículo se presentó un método de detección de subjetividad en noticias en español, implementando clasificadores probabilísticos, utilizando un corpus de noticias etiquetado de manera manual y clasificadores probabilísticos mencionados en la bibliografía.

Se desarrolló un corpus de noticias en español, el cual contiene notas periodísticas publicadas en México, de diversos sitios web. Este corpus sirvió como datos de entrenamiento para los clasificadores probabilísticos.

Se realizaron diversos experimentos, modificando las características de los datos del corpus, con el fin de determinar con qué clase de características y con qué clasificador se obtendría el mejor rendimiento al detectar objetividad y subjetividad. Durante la creación del corpus, se observó que no se obtuvo alguna noticia que se encontrara libre de contener oraciones subjetivas.

Se pudo observar, que en cuanto a la detección de objetividad, la mayor precisión fue de 73%, al eliminar los n-gramas objetivos repetidos, utilizando bigramas, incluyendo *stopwords* y lematizando el texto, al utilizar el clasificador *Bayes Net*. En subjetividad, la mayor precisión fue de 76%, la cual se obtuvo al utilizar el clasificador

Bayes Net, eliminando los n-gramas repetidos, utilizando bigramas sin *stopwords* y lematizando el texto.

En cuanto al trabajo futuro, se planea implementar un módulo de análisis automático basado en reglas, con el objetivo de corregir posibles errores en la clasificación automática, además de tratar de mejorar los resultados obtenidos en este trabajo. Además, se planea tratar de identificar qué porcentaje de oraciones subjetivas puede contener una noticia para ser considerada como objetiva.

Referencias

1. Estudio sobre los hábitos de los usuarios de Internet en México. Asociación Mexicana de Internet. https://www.amipci.org.mx/es/estudios
2. Gómez, R., Sosa-Plata, G., Bravo, Téllez-Girón, P., Dragomir, M., Thompson, M.: Los medios digitales: México. Open Society Foundations. http://www.opensocietyfoundations.org
3. Wiebe, J. M.: Tracking point of view in narrative. Computational Linguistics, pp. 233-287 (1994)
4. Wiebe, J., Bruce, R., Martin, M., Wilson, T., Bell, M.: Learning subjective languaje. Computational Linguistics, pp. 277–308 (2004)
5. Salaverría, R., Cores, R. .: Géneros periodísticos en los cibermedios hispanos (2005)
6. Abundis, F.: Los medios de comunicación en México. Parametría, Investigación de opinión y mercados (2006)
7. Panicheva, P., Cardiff, J., Rosso, P.: Identifying Subjective Statements in News Titles Using a Personal Sense Annotation Framework. In: American Society for Information Science and Technology, pp. 1411–1422 (2013)
8. Sukhum, K., Nitsuwat, S., Choochart, H.: Opinion Detection in Thai Political News Columns Based on Subjectivity Analysis. In: 7th International Conference on Computing and Information Technology (2011)
9. Wilson, T., Hoffmann, P., Somasundaran, S., Kessler, J., Wiebe, J., Choi, C., Riloff, E., Patwardhan, S.: OpinionFinder, A system for subjectivity analysis. In: Conference on Empirical Methods in Natural Languaje Processing, pp. 34–35 (2005)
10. Lin, C., He, Y., Everson, R.: Sentence subjectivity detection with weakly-supervised learning. In 5th International Joint Conference on Natural Languaje Processing, pp. 1153–1161 (2011)
11. Bruce, R. F., Wiebe, J. M.: Recognizing subjectivity; A case of study of manual tagging. Natural Languaje Engineering, pp. 1–16 (1999)
12. Bouckaert, R. R., Eibe, F., Hall, M., Kirkby, R., Reutemann, P., Seewald, A., Scuse, D.: Weka manual for version 3-7-11, Waikato (2014)

13. Wang, S., Manning, C. D.: Baselines and Bigrams, Simple, good sentiment and topic classification. In: 50th Annual Meeting of the Association for Computaniotal Linguistics, pp. 90–94 (2012)
14. McCallum, A., Nigam, K.: A comparison of events model for Naive Bayes Text classification. In: Learning for text categorization (1998)

Metodologías para análisis político utilizando Web Scraping

Alexis Tadeo Hernández, Edy Gómez Vázquez, César Alejandro Berdejo Rincón, Jorge Montero García, Adrian Calderón Maldonado, y Rodolfo Ibarra Orozco

Universidad Politécnica de Chiapas,
Tuxtla Gutiérrez, Chiapas, México

Resumen. En este artículo se revisan distintas metodologías utilizadas para realizar un análisis político utilizando diversas fuentes de información disponibles en internet. En algunas sociedades, el uso de redes sociales tiene un impacto significativo en el ámbito político con la sociedad y se han empleado diversas metodologías para analizar diversos aspectos políticos y las estrategias a seguir. El propósito de este trabajo es entender estas metodologías para poder proporcionar información a los posibles votantes que les permitan tomar decisiones informadas. Primero, se revisa la terminología necesaria sobre web scraping, después, se presentan algunos ejemplos de proyectos para el análisis político que han empleado web scraping. Finalmente, se presentan nuestras conclusiones.

Palabras clave: Web scraping, text mining, análisis político.

1. Introducción

En Estados Unidos, las redes sociales han tomado un papel importante en el ambiente político: las usan para investigar a fondo a la oposición con equipos especializados para encontrar incoherencias en el adversario; aprovechan las diferencias y coyunturas, por ejemplo, [8], en los Estados Unidos, los republicanos en el Congreso se opusieron al recorte del impuesto sobre la nómina, que obligaría a los estadounidenses a que en cada pago de su sueldo le fueran recortados, en promedio, unos 40 dólares. Uno de los argumentos usados fue que 40 dólares no era mucho dinero.

En menos de 12 horas, la Casa Blanca reaccionó con una estrategia en la que invitaban a los ciudadanos a enviar por Twitter, Facebook, YouTube qué significaba para ellos 40 dólares y después Barack Obama tomó ejemplos de cómo 40 dólares menos al mes afectan a las familias estadounidenses logrando que el Congreso de los Estados Unidos rechazara el recorte del impuesto sobre la nómina [10].

Este es un ejemplo de cómo algunas sociedades aprovecha las redes sociales en el ámbito político. En México, por el contrario, la política no deja de ser una discusión eterna entre votados y votantes por medio de los spots que publican en fuentes convencionales (radio, televisión, incluso el cine), dando una "guerra" discursiva entre los votantes a favor de un partido o candidato y quienes no saben por quién votar [8].

Conocer a fondo a los políticos, consultar diversas fuentes y analizarlas es una labor complicada, éstas son las razones de desarrollar una herramienta que, utilizando técnicas de web scraping y text mining, permita a la población conocer de diversas fuentes (redes sociales, sitios web de periódicos, búsqueda en Google) a cualquier político, analizar los resultados de la búsqueda y mostrar indicadores de confianza en base a lo obtenido en un análisis de la información con el fin de dar al usuario una perspectiva distinta.

1.1. La política y el uso de las redes sociales

El uso de las redes sociales ha sido un factor importante para poder compartir opiniones sobre diversos temas que le interesan a la ciudadanía, conocer a candidatos que se postulan para un puesto, ya sea presidencia, senadores o demás personas que trabajen en el ámbito político.

En las elecciones del 2008, en los Estados Unidos, podemos ver un claro ejemplo acerca de cómo los candidatos a la presidencia de los Estados Unidos de América, Mitt Romney y Barack Obama, hacen uso de las redes sociales para darse a conocer con las personas, además de dar sus puntos de vista acerca de diversos temas sociales y así poder opinar junto con la sociedad.

La implementación de la tecnología en la política ha permitido participar en un nuevo nivel de conversación con los votantes, permitiendo así que una campaña donde los candidatos se dan a conocer se vuelva algo mucho más dinámico, algo más de un simple diálogo. Así Barack Obama, en las elecciones presidenciales de 2012, utilizó de forma más efectiva la web para poder establecer una campaña más insurgente y poder ganar el voto de las personas jóvenes, lo cual fue funcionando. Por ejemplo durante esta campaña (de junio 4 a junio 17), la campaña de Obama realizó 614 publicaciones mediante su plataforma, mientras que la Romney realizó sólo168, y la brecha en twitter fue aún mayor, promediando 29 mensajes diarios de Obama contra uno de Romney [7].

2. Web Sraping

También conocido como Web harvesting o Web data extraction, es el proceso de rastreo y descarga de sitios web de información y la extracción de datos no estructurados o poco estructurados a un formato estructurado. Para lograrlo, se simula la exploración humana de la World Wide Web, ya sea por implementación de bajo nivel del protocolo de transferencia de hipertexto, o la incorporación de ciertos navegadores web.

Para realizar scraping se utiliza un programa, conocido como orquestador, que organiza y ejecuta las peticiones al browser. Se deben tener bien definidos los elementos a buscar, y que se indique el estado de la búsqueda a realizar (búsqueda exitosa, errores en la búsqueda, sin resultados).

El proceso de web scraping se realiza en dos etapas, la primera es la etapa de extracción, en la cual se realiza una consulta de datos hacia un sitio y se guardan de

manera local y, después, en la segunda etapa, se realiza el análisis de estos datos para obtener información.

2.1. Extracción

Técnicas para la extracción de información

- Web bot, Spider, Crawler, Arañas y Rastreadores [11].

  Inspeccionan las páginas web de internet de forma metódica y automatizada. Se usan para rastrear la red. Lee la estructura de hipertexto y accede a todos los enlaces referidos en el sitio web. Son utilizadas la mayoría de las veces para poder crear una copia de todas las páginas web visitadas para que después puedan ser procesadas por un motor de búsqueda; esto hace que se puedan indexar las páginas, proporcionando un sistema de búsquedas rápido

- Plataformas de agregación verticales:

Existen plataformas que tienen el propósito de crear y controlar numerosos robots que están destinados para mercados verticales específicos. Mediante el uso de esta preparación técnica se realiza mediante el establecimiento de la base de conocimientos destinado a la totalidad de plataformas verticales y luego a crearla automáticamente. Medimos nuestras plataformas por la calidad de la información que se obtiene. Esto asegura que la robustez de nuestras plataformas utilizadas consiga la información de calidad y no sólo fragmentos de datos inútiles.

- Reorganización de anotación semántica.

El desarrollo de web scraping puede realizarse para páginas web que adoptan marcas y anotaciones que pueden ser destinadas a localizar fragmentos específicos semánticos o metadatos. Las anotaciones pueden ser incrustadas en las páginas y esto puede ser visto como análisis de la representación estructurada (DOM). Esto permite recuperar instrucciones de datos desde cualquier capa de páginas web.

Herramientas utilizadas en la extracción

- ScraperWiki. Es una plataforma web que permite crear scrapers de forma colaborativa entre programadores y periodistas para extraer y analizar datos públicos contenidos en la web.

- PHP. Cuenta con librerías para realizar web scraping como cURL, el cual permite la transferencia y descarga de datos, archivos y sitios completos a través de una amplia variedad de protocolos, y Crawl, que contiene varias opciones para especificar el comportamiento de la extracción como filtros Content-Type, manejo de cookies, manejo de robots y limitación de opciones.

- Guzzle: Es un framework que incluye las herramientas necesarias para crear un cliente robusto de servicios web. Incluye: descripciones de Servicio para definir las entradas y salidas de una API, iteradores para recorrer webs paginadas, procesamiento por lotes para el envío de un gran número de solicitudes de la manera más eficiente posible. Fué creado usando Symfony2 y emplea la librería cURL de PHP.

- Jsoup de Java: Es una librería para realizar web scraping. Proporciona una API muy conveniente para la extracción y manipulación de datos, utilizando lo mejor de DOM, CSS, y métodos de jQuery similares.

    - Raspa y analiza el código HTML de una URL, archivo o cadena
    - Encuentra y extrae los datos, utilizando el DOM o selectores CSS
    - Manipula los elementos HTML, atributos y texto.
    - Limpia el contenido enviado por los usuarios contra una lista blanca de seguridad, para evitar ataques XSS.
    - Salida HTML ordenada

- Beautifulsoup: Es una biblioteca de Python diseñada para proyectos de respuesta rápida como screen scraping o web scraping. Ofrece algunos métodos simples y modismos de Python para navegar, buscar y modificar un árbol de análisis: una herramienta para la disección de un documento y extraer lo que necesita, además de que no se necesita mucho código para escribir una aplicación. Beautiful Soup convierte automáticamente los documentos entrantes a Unicode y documentos salientes a UTF-8, también trabaja con analizadores de Python populares como lxml y html5lib y permite realizar el recorrido del DOM.

2.2. Análisis

Herramientas para análisis

Algunos ejemplos de herramientas utilizadas para el análisis de texto son:

- myTrama

Es un sistema web que aporta un lenguaje propio de consultas, similar a SQL. Cuenta con una interfaz visual que carga la web objetivo que permite seleccionar los datos mostrando en bloques de pantalla lo que se necesita. El proceso de selección se traduce en la construcción de una consulta en el lenguaje propio de la herramienta, al que denominan Trama-WQL (Web Query Language). Esta consulta puede ser gestionada en modo texto, incluso puede ser escrita desde cero sin tener en cuenta el seleccionador. Ambos, el editor WQL y el seleccionador están sincronizados, por lo que un cambio en uno de ellos repercute en el otro. Un sistema de recolección de la información permite que los tiempos de latencia entre myTrama y la web no afecten a las llamadas a las APIs, que devolverán muy rápido la información que hay en la caché. En caso de que el sistema detecte que los datos son obsoletos, refrescará los datos de

la caché en background. Permite trabajar con data web mining, como por ejemplo agregadores, comparadores o enlazado de datos.

- Gensim

Es una biblioteca de python que proporciona estadísticas escalables de semántica, analiza documentos de texto plano para la estructura semántica y recupera documentos semánticamente similares. Los algoritmos de Gensim, como análisis semántico latente y el de proyecciones aleatorias, descubren la estructura semántica de los documentos, mediante el examen de patrones de co-ocurrencia dentro del cuerpo de documentos de entrenamiento. Estos algoritmos son sin supervisión.

- Natural Language Toolkit (NLTK)

Es un conjunto de bibliotecas y programas para el procesamiento del leguaje natural (PNL) simbólico y estadístico para el lenguaje Python. Proporciona interfaces fáciles de usar para más de 50 cuerpos y recursos léxicos, como WordNet, junto con un conjunto de bibliotecas de procesamiento de texto para la clasificación, análisis y razonamiento semántico.

3. Trabajo relacionado

Proyecto de ley

Proyectosdeley.pe [9] es una aplicación web que muestra, en forma ordenada y accesible, los proyectos de ley presentados en el Congreso peruano. Es su primer intento de abrir la información estatal usando creativamente la tecnología para promover la transparencia. Este proyecto trata de presentar la información sobre los proyectos de ley producidos por el Congreso en una interfaz amigable e intuitiva. En "ProyectosDeLey" almacenan los datos que se extraigan con BeautifulSoup, principalmente los datos que se buscan almacenar son los títulos, autores, fechas de publicación, entre otros datos.

El software de Proyectosdeley se activa automáticamente cada 3 horas y empieza a buscar proyectos nuevos que hayan sido colgados en la web del Congreso. Si los hay, los descarga, parsea y los guarda indexados en la base de datos local. Cuando ya no hay más proyectos por descargar o procesar, empieza a generar los archivos HTML que puedes ver si visitas el sitio web. También genera las páginas web para cada congresista que haya sido autor de al menos un proyecto de ley.

6news Lawrence

6news Lawrence es un noticiero que mostraba en su programa las estadísticas de las elecciones del estado de Kansas en el 2006. Para poder tener la información antes que su competencia ellos realizaron scraping de datos. El proceso de scraping lo realizaron siguiendo los siguientes pasos:

1. Los votos contabilizados, en un principio, eran publicados en una página web privada a la que sólo podían acceder con una determinada IP.
2. Un script descarga la página de resultados cada vez que cambia y sube los resultados en una página pública.
3. Otro script raspa el html de la página publica (usando la librería de Python beautiful soup) e inserta los datos de la página en una base de datos
4. Un tercer script trae la información de la base de datos y escribe una hoja de cálculo en Excel en una url pública.
5. En 6news una ventana de Windows corre un archivo batch, el cual se encarga de descargar el archivo de Excel
6. Finalmente, el sistema on air-graphics (el cual se encarga de mostrar estadísticas y graficas en tiempo real) lee el archivo de Excel para posteriormente mostrarlo en el noticiero.

Con este procedimiento, 6news logró obtener los resultados inclusive hasta 30 minutos más rápido que su competencia.

Análisis de las tendencias políticas basadas en los patrones de web linking: el caso de Grupos políticos en el Parlamento Europeo

Con el fin de conocer la situación política de la Unión Europea (UE), en este proyecto [4] se recogieron diversos tipos de datos sobre enlaces web a sitios web de los 96 partidos que conforman la UE con el fin de encontrar patrones para su estudio. Se utilizaron 2 tipos de enlaces: los in-link que son hipervínculos incrustados en una página que apuntan a otra página; y los co-link que son enlaces incrustados en dos o más sitios que re-direccionan a una misma página.

Los datos Web co-link se visualizaron utilizando escalamiento multidimensional (MDS), mientras que los datos in-link se analizaron con un análisis de dos vías de varianza. Los resultados mostraron que los datos web de hipervínculo reflejaban algunos patrones políticos en la Unión Europea (UE). Los mapas MDS mostraron grupos de partidos políticos a lo largo de líneas ideológicas, históricas, lingüísticas y sociales.

El análisis estadístico basado en in-link confirmó además que había una diferencia significativa a lo largo de la línea de la historia política de un país, de manera que los partidos de izquierda en los antiguos países comunistas recibieron un número considerablemente menor de in-links a sus sitios web que los partidos de izquierda en los países sin una historia de comunismo.

Extracción de posiciones políticas desde textos políticos utilizando palabras como datos

Este artículo, [2], presenta una nueva manera de extraer posiciones políticas de textos políticos que, a los textos, no los ve como discursos sino como datos en forma de palabras. Se comparó este enfoque a los anteriores métodos de análisis de texto y se usó para hacer una replicación de las estimaciones publicadas sobre las posiciones políticas de los partidos en Gran Bretaña e Irlanda, en ambas dimensiones políticas, económicas y sociales.

A continuación se presentaran los pasos a seguir para la extracción y análisis de los textos.

- Paso 1: Se obtienen los textos de referencia con posiciones conocidas a priori.
- Paso 2: Se generan puntajes de palabras de textos de referencia (puntuación de palabras)
- Paso 3: Se obtiene la puntuación de cada texto virgen utilizando puntajes de palabras (textos básicos)
- Paso 4: (opcional) Se transforman las puntuaciones de texto vírgenes para una métrica original.

Para el proyecto se usaron técnicas del algoritmo "Word scoring" las cuales replican con éxito las publicaciones estimadas de política sin los costos sustanciales de tiempo y mano de obra que éstos requieren. Este algoritmo lee archivos de texto y calcula una puntuación en base a un sentido de palabras a partir de una intersección de ese conjunto de palabras y elige el sentido con las mejores puntuaciones.

El algoritmo toma una palabra de referencia y ve cuántas veces coincide en el conjunto de documentos o documento y le da una puntuación según sea la coincidencia, entre menor sea la coincidencia más alto es el puntaje que le dará y entre mayor sea la coincidencia menor puntaje dará. Si por la palabra evaluada hay varios significados en el documento, se considera la elección más cercana a ésta y es tomada como la mejor.

Midiendo opiniones políticas en blogs

En este proyecto, [3], se obtuvieron publicaciones de personas que están muy involucradas en la política, así como también de estadounidenses que normalmente bloguean cosas sobre otros temas, pero que por algún motivo deciden unirse a una conversación política en 1 o más publicaciones.

Se descargaron y analizaron todas las nuevas publicaciones de un blog cada día. La meta especifica es categorizar las publicaciones en 7 categorías únicas: extremadamente negativo (−2), negativo (−1), neutral (0), positivo (1), extremadamente positivo (2), no opinion (NA), y not a blog (NB). La metodología propuesta en este proyecto es:

Primero, se ignoran todas las publicaciones que estén en idiomas diferentes al inglés, lo mismo con las publicaciones que sean spam. Este proyecto se concentró en 4,303 publicaciones de blogs acerca del presidente Bush y 6,468 publicaciones acerca de la senadora Hillary Clinton.

Como segundo paso, se procesó el texto de cada documento, convirtiendo todo a minúscula, removiendo todo signo de puntuación, y derivando las palabras a su origen primitivo, por ejemplo, "consistir, "consistió", "consistencia", "consistiendo" se reduce a su palabra de origen primitivo que sería consistir, logrando reducir la complejidad de la información que se encuentra en el texto.

Por último se resumió el texto pre-procesado como variables dicotómicas, un tipo para la presencia o la ausencia de cada raíz de la palabra (o unigrama), un segundo tipo por cada par de palabras(o bigrama), y un tercero por cada palabra triplete (o trigrama), de esa forma hasta llegar a n-gramas, sólo se mide la presencia o ausencia

de la raíz de las palabras en vez de contarlas todas (la segunda aparición de la palabra "horroroso" en una publicación no provee tanta información como la primera aparición). Incluso así, el número de variables que restan es enorme, en el ejemplo que se tomó de 10771 publicaciones de blogs acerca del presidente Bush y la senadora Clinton incluía 201,676 unigramas únicos, 2,392,027 bigramas únicos, y 5,761,979 trigramas únicos. La forma usual para simplificar más las variables era considerar solo los unigramas dicotómicos que provengan de la raíz de las variables indicadoras.

Una investigación preliminar de análisis sentimental en el discurso político informal

Mullen [1] explica que dada la tendencia a la alza de que las publicaciones en línea se han convertido en comunicaciones estilo mensaje, el discurso político informal es ahora una característica importante dentro del internet, creando un área para la experimentación en técnicas de análisis sentimental. Menciona que, algunas de las preguntas que podríamos preguntar acerca de un texto, además de un simple juicio sobre un tópico, candidato o propuesta, son, por ejemplo:

a) Identificar la afiliación política del escritor,

b) Clasificar el punto de vista político del escritor, de acuerdo a una taxonomía, como izquierda o derecha, y

c) Evaluar el grado de confianza con el cual el escritor expresa su opinión.

En este artículo se realizó un análisis de la efectividad de métodos de clasificación estándar para predecir la afiliación política del blog evaluado. Los resultados obtenidos sugieren que los métodos tradicionales de clasificación de texto son inadecuados para la tarea de análisis sentimental político. Propone realizar un análisis de cómo un post interacciona con otro, esto con el fin de utilizar la información que se tenga de un post para ayudar a clasificar otros.

4. Conclusiones

En este trabajo se realizó una revisión del estado del arte de diferentes metodologías que se han propuesto para realizar análisis político en las redes sociales y en el internet en general. Esta es la primera etapa de un proyecto de investigación en el que se pretende obtener información de diferentes fuentes en internet mediante técnicas de web scraping y analizar lo obtenido mediante técnicas de text mining. Esta revisión nos será de gran utilidad para definir los indicadores necesarios, así como los resultados esperados de nuestro proyecto.

Nuestro objetivo es que cualquier persona pueda obtener información para poder dar una crítica fundamentada, tener una postura basada en diferentes fuentes y conocer noticias importantes con respecto a la política mexicana.

Referencias

1. Mullen, T., Malouf, R.: A Preliminary investigation into sentimental analysis of informal political discourse. AAAI Symposium on Computational Approaches of Analysing Weblogs, pp. 159–162 (2006)
2. Laver, M., Benoit, K., Garry, J.: Extracting Policy Positions from Political Texts Using Words as Data. The American Political Science Review, pp. 311–331 (2003)
3. Hopkins, D. J., King, G.: A method of automated nonparametric content analysis for social science. American Journal of Political Science, vol. 54 (1), pp. 229–247 (2010)
4. Romero-Frías, E., Liwen, V.: The analysis of political trends based on Web linking patterns: The Case of Political Groups in the European Parliament (2009)
5. Matt, T., Pang, B., Lillian, L.: Get out the vote: Determining support or opposition from Congressional floor-debate transcripts Proceedings of EMNLP, pp. 327–335 (2006)
6. Vasilevsky, D.: Parallelized web scraping using RollingCurl (2015)
7. How the Presidential Candidates Use the Web and Social Media, Pew Research Center: Journalism & Media Staff. http://www.journalism.org/2012/08/15/how-presidential-candidates-use-web-and-social-media
8. Ramos, D.: 5 usos electorales de las redes que EU hace y México desdeña (2012)
9. Organizando los proyectos de ley del congreso. http://aniversarioperu.utero.pe
10. What $40 means to Americans across the country. https://www.whitehouse.gov/40dollars
11. Schrenk, M.: Webbots, spiders, and screen scrapers, a guide to developing internet agent with PHP/CUR, 2nd edition (2012)

Designation of Situation Model in Twitter using Maximal Frequent Sequences

Anna Atyagina[1,2], Yulia Ledeneva[1], and René Arnulfo García-Hernández[1]

[1] Universidad Autónoma del Estado de México,
Unidad Académica Profesional Tianguistenco,
Toluca, Estado de México, Mexico

[2] Omsk F.M. Dostoevsky State University, Omsk,
Russia

atyagina@gmail.com, yledeneva@yahoo.com, renearnulfo@hotmail.com

Abstract. Hashtag is definitely one of the most significant features of Twitter which now is spread all over the social networking services. It can serve different functions, and one of the most important is the designation of situation models. Using the method of Maximal Frequent Sequences we proved that the main idea of all data of one hashtag can be described in two or three phrases as a summary processed using the given method. We demonstrate how the recognition of situation models can be done automatically and fast. Also this method can be used for analysis of hashtag combinations and reconstruction of concepts based on the results of 1-grams and 2-grams, as we presented in detailed example of analysis of the following hashtags: #GalaxyFamily, #RussianMeteor, #Grammys and #10Dec hashtags.

Keywords: Maximal Frequent Sequences, hashtag, Twitter, social media, situation model.

1 Introduction

More than 500 million messages appear on Twitter daily [1]. Although one of Twitter's main features is brevity of messages that are transmitted with it (no more than 140 characters), there is a real problem of organization and systematization of information on the service.

Hashtag is a feature that helps systematize process of communication. It has changed people's interaction and ways to find information within and outside of Twitter. Hashtags are key words or phrases that begin with a # symbol followed by any combination of Twitter permitted nonblank characters. They can occur in any part of tweets. Users simply may add # in front of any word. Hashtags can be used for searching messages, following a certain thread or topic, and therefore can mark a set of tweets focusing on a certain topic described by the hashtag.

At the same time there is still a problem of an immediate understanding of different hashtags, popular or not. Sometimes to get the main meaning of hashtag, user of

Twitter has to look through dozens of tweets. It can be almost impossible if someone wants to understand, for example, overall chronic of the day with its main trending hashtags or keywords.

The purposes of our work are to confirm the function of hashtags as indicators of models of situations using Maximal Frequent Sequences (MFSs); also demonstrate how the designation of situation models can be done automatically and fast; and present examples of analysis of hashtags data, based on the results of 1-grams and 2-grams Frequent Sequences (FSs).

2 Related Works

In the past years, methods of data mining were actively applied to the analysis of the information provided on Twitter, including use of hashtags. The researches are devoted to different topics, from automatic identification of diverse sentiment types using hashtags as long as some other Twitter features [2] to the most popular terms or topics of discussion on Twitter for better insight into the collective viewpoint and subjects of interest of the typical Twitter user [3]. The structure of gatekeeping in Twitter by means of statistical analysis of the political hashtags #FreeIran, #FreeVenezuela and #Jan25, each of which reached the top position in Twitter Trending Topics (list of the most tweeted topics ranked by Tweeter proprietary algorithm), is explored in [4].

Some of the researches are mostly practically oriented and are objected to recommend how to use hashtags more effective. Zangerle et al. [5] recommend an approach that aims at creating a more homogeneous set of hashtags for a particular message. At the same time, Ma et al. [6] propose methods to predict the popularity of new hashtags on Twitter by formulating the problem as a classification task. Page [7] presents the investigation on the role of hashtags as means of self-branding.

Although hashtags are a subject of big interest for researches, there are not many researches devoted to the functions of hashtags. There are some new terms as gametags, rhetorical hashtag, bashtag, hijacking or grasstag that serve different purposes [8]. The phenomenon of “hijacking on Twitter” is described in [9], where some certain jumps in polarity are shown, caused by "hijackers" engaged in a particular type of hashtag war. Another new definition of blacktag is described in [10] and connects Twitter studies with cultural and racial studies.

Data mining on Twitter hashtags provides interesting results when it comes to forecasting some situations, human decisions and actions. Big data gives an opportunity to automatize an analysis of users’ interaction with each other and referring to the important events. For example, [11] introduced the method of political elections forecasting in different countries based on Twitter messages. Another, more specific research forecasts the results of 2013 elections in Pakistan and 2014 elections in India [12].

Even the behavior of stock exchange can be predicted with Twitter data. Using behaviorist approach to the economy along with data mining methods, [13] demonstrate how collective mood influences on the decision making at the stock

exchange market. Authors discovered that the connection between Twitter users' mood and changes of Dow Jones Index exists in 86.7% of situations processed.

Big events are of particular interest for hashtags' researchers. Twitter's reaction to important sport and cultural events is analyzed in [14]. One of the examples is an analysis of American soccer fans tweets during the World Cup 2014 [15].

Atiagina [16] presents the classification of hashtag functions that we rely on. Hashtags can be used for different purposes, and carry different information. Five functions are included in the classification: designation of situation models in order of compression, inclusion in the overall context/trends, actualization and expression, self-presentation, promotion [16]. In this paper, we consider to the first and the most significant function which is designation of situation models in order of compression.

In the analysis of Twitter messages, it is appropriate to rely on the general principles of text compression mechanism. It allows to compress text in transmission and expand it in perception, without losing the most important and significant information. This is largely possible due to the fact that an adequate understanding between the author and the reader is based on a code system common to the sender and recipient and includes mental scenarios, concepts, categories, rules and strategies. On the basis of cognitive theory of language use, T.A. Van Dijk [17, 18] proposes the following idea: we understand the text only when we understand the situation referred to. Therefore the model of situation is necessary for us as the basis of interpretation of the text. The use of models is explained why listener understands implicit and unclear sections of code: in this case, they activate the corresponding fragments of situational model.

Hashtag serves as a model of situation that compresses the context of tweet but, generally, can be expanded by the reader in perception [16]. If hashtag exists and is widely used, at the same time it would describe the same situation for the most of the people that are using this hashtag. It means that they are likely to use similar words (or other hashtags) in messages with the same hashtag.

In this paper, we propose the method that is based on extraction of MFSs which helps us to show that among the large amount of messages with the same hashtag users are going to use the words with the same semantics and, in general, the main idea of all the messages can be described in one or two phrases as a summary.

3 Proposed Method

An ngram is a sequence of n words. We say that an ngram occurs in a text if these words appear in the text in the same order immediately one after another. We call an ngram frequent (more accurately, β-frequent) if it occurs more than β times in the text, where β is a predefined threshold. Frequent ngrams—we will also call them frequent sequences (FSs)—often bear important semantic meaning: they can be multiword expressions, idioms or otherwise refer to some idea important for the text [19, 20]. An ngram can be a part of another, longer ngram. FSs that are not parts of any other FS are called Maximal Frequent Sequences (MFSs) [21, 22].

Only MFSs should be considered as bearing important meaning, while non-maximal FSs (those that are parts of another FS) should not be considered. Other

additional motivation may be cost vs. benefit considerations: there are too many non-maximal FSs while their probability to bear important meaning is lower. In any case, MFSs represent all FSs in a compact way: all FSs can be obtained from all MFSs by bursting each MFS into a set of all its subsequences. FSs can express ideas both important and specific for the document [19].

Using data mining techniques and extraction of MFSs [21, 22] in accordance to the task involves the following steps:

1. Data collection using Twitter API.
2. Data preprocessing. Each Tweet contains a lot of information that needs to be removed or altered for adequate treatment and allocation of MFSs. Thus, functional words and phrases that are present in the interface of Twitter in each message (such as “expand”, “answer” etc.) are being automatically removed by the program Format Text. Links to the other resources are substituted by the word “@ liga” and hashtags are substituted by the word @hash. Also links to user names containing the @ symbol are replaced with the word “name”.
3. Extraction of MFSs. This is done as developed in [21, 22]. At the first stage, the program converts all data to binary files, which are then used to analyze the data. All documents (which are, in this case, Twitter messages) are loaded into the program so that each message is assigned its own line. Words are automatically highlighted in capital letters, separated by commas. The program automatically scans the text and extracts the words that are repeated often. Depending on what is specified maximum number of words within the sequence of the frequency (the average maximum frequency commonly used sequences containing from two to six words) can produce different results for later analysis.

4 Corpus

For this paper, four popular hashtags were processed in English, Russian or mixed languages. We've postulated models of situations to each of these hashtags that we're going to prove using method of MFS's lately.

The processed hashtags are:

#GalaxyFamily (English, 11133 tweets) — promo hashtag, community of Galaxy phones' users as a family.

#Grammys (English, 9890 tweets) — highlighting the famous musical ceremony, the most relevant artists' names and actions of event in 2014.

#RussianMeteorit (Russian and English, 3952 tweets) — meteorite fall in Chelyabinsk in February 2013.

#10dec (Russian and English, 2456 tweets) — protests against the results of State Duma elections on December, 10th, 2011.

We provide all results for all processed hashtags but more detailed description presented later will be devoted to #GalaxyFamily hashtag. It is a promoted hashtag that, at the same time, serves at least two functions: designation of model of situation and marketing. In this case, the situation itself is created by the company (Samsung) but still is a relevant objective as long as thousands of Twitter users tweeted with this

hashtag on their own, voluntarily. Our #GalaxyFamily corpus consists of 11133 tweets published during 2013 and contained hashtag #GalaxyFamily, mostly in English. Working specifically with this hashtag, we've processed 135448 words and got 30989 frequent sequences.

5 Experiments and Analysis

5.1 Indicating Meaning of Hashtags

Previously described method serves to identify the most popular MFS in the corpus which can help to make a quick summary of the chosen hashtag. In the Table 1, three types of results for each hashtag are provided: the MFSs, the longest FSs and the most frequent 1-gram (or in other words the most frequent words that are met in the corpus).

Generally, these results can demonstrate the main meanings and ideas of each hashtag and help to identify it to those Twitter users who are not aware of the model of situation. Both MFSs and 1-grams demonstrate connection with the model of situation that can be used then in different ways: quick and automatized recognition of models of situations, analysis and linguistic interpretation of users' behavior and opinions, etc. In the next part of the paper, we provide the detailed example of this analysis.

The proposed method is language independent: we see that results in different languages are valid and for the same hashtag they have the same meaning (#RussianMeteor, #10dec).

During our experiment we met a problem of a spam or commercial tweets that use popular hashtags to promote other information. Sometimes, for example, with hashtag #RussianMeteor, the most frequent MFSs can be considered as a spam. In other situations (#10dec, #Grammys) they are not a spam. Comparing these results to the longest MFSs found, we conclude that the most effective way of automatic summarization of hashtags data would be combination of the most frequent MFSs with the longest ones. It is also important because some of the results can be in the language that user doesn't understand. For example, the most frequent result for the hashtag #GalaxyFamily is in Turkish although the main language of the hashtag is English but other results serve to understand the model of situations.

Although MFSs are considered as the most significant, shorter n-grams also can help to explain or to analyze hashtags. For example, each hashtag has its own group of the most frequent 1-grams that can help to reconstruct the overall model of situation and confirm our hypothesis. Frequent 2-grams can be used both to reconstruct the model situation and to find the frequent hashtag combinations.

5.2 Example of Detailed Analysis: #GalaxyFamily Hashtag

5.2.1 Maximal Frequent Sequences

Usage of Maximal Frequent Sequences helps us not only to understand the main idea of the hashtag but also to provide different kind of analysis that can be useful for marketing specialist as well as for linguist.

Table 1. The results are the MFSs, the longest FSs and the most frequent 1-gram for hashtags #GalaxyFamily, #RussianMeteor, #Grammys and #10Dec.

The most frequent MFSs	The longest FSs	1-gram FSs
1. #GalaxyFamily [276] retweet uygulamas @liga @foto #galaxyfamily #milletineserikapatilamaz #benceeask #reklam #google [103] welcome to the #galaxyfamily you can learn all about your new phone in our guide to the galaxy at @liga [85] black friday deals on amazon now @liga #toysruskid #dwts thanksgiving #galaxyfamily	[2] i m a owner of the #s and i have to admit it s the best phone i ve had by far wouldn t change for the world thanks #galaxyfamily (30 words) [2] got my galaxy s saturday and i gotta say it was the best decision i made when i d comes to choosing a new cell phone #galaxyfamily(27) [2] to the #galaxyfamily i have had a galaxy phone for years and i have loved em all but the note might just be greatest phone ever (26)	[82] love [38] member [35] @samsungmobileus [32] shit [31] photo [27] fun [25] #apple [24] almost [24] lmao [24] her [24] his [23] apps [21] try [20] something [20] white [20] super [20] definitely [19] #follow [19] hi [19]#cybermondaymadness [19] nothing [19] wit [19] hello [18] @hashmtvstars [18] galaxys
2. #RussianMeteor 4 most frequent are spam not connected with the situation. after them there are the following: [15] rt @bbcbreaking #russianmeteor shower six of the most dramatic videos @liga #meteorit @liga [13] @liga ultimas imagenes del meteorito que cayo en rusia #meteorit @hashrussianmeteor [13] at least people injured in spectacular #russianmeteor shower interior ministry @liga @liga #meteorit	[5] coming up at pm @nasa experts discuss the #russianmeteor learn more about meteor listen to panel live at pm et @liga (21 words) [5] wow rt @wsj the explosion from the #russianmeteor is estimated to be as powerful as Hiroshima bombs @liga (18) [5] rt @nasa #russianmeteor is largest reported,meteor since Tunguska event impact was at utc still being @liga (17)	[11] kak (how) [10] jeto (this) [10] again [9] recorded [8] though [7] her [7] eshhe (again) [7] #meteorites [7] @neiltyson [7] window [7] loud [7] during [6] les [6] kilotons [6] answers

The most frequent MFSs	The longest FSs	1-gram FSs
3. #Grammys [19] see the best part of the #grammys also known as queen latifah marrying a bunch of happy couples via @upworthy @liga [18] fresh from the #grammys did you see what beyonce was wearing barely @liga [16] paul walker death tyrese gibson thanks fans during grammys #paulwalker #tyresegibson #grammys @liga	[2] the best part about watching the 7ashgrammys the day after is you re able to fast forward through the boring stuff which is most of it (26) [2] i m with trent on this one i tuned into the end because i was excited to see some of my fave artists collaborate left unhappy 7ashgrammys (27) [2] love how pharrell pointed to stevie to pick up his cue at the start when he missed it stevie s blind mate pointing won t do much 7ashgrammys (28)	[9] event [9] Ciara [9] JaredLeto [8] ridiculous [8] ni [8] loves [8] literally [8] exclusive [8] avec [8] apparently [7] noticed [7] lame [7] france [7] daftpunk [7] become [7] along [6] wit [6] water [6] themselves
4. #10dec [49] @liga #10dec #russia [40] rt @burmatoff voenkomaty segodnja perevypolnjat normu po prizyvu @liga soberut na bolotnoj vseh uklonistov #10dek #10dec (RUS to ENG: rt @burmatoff today military enlistment offices will exceed a plan @liga by finding all the objectors on Bolotnaya #10dek #10dec) [34] @liga #10dec #10dek #revolution	[3] nezavisimaja ocenka chislennosti akcij #10dec #10dec #feb v #omsk na osnove kart i fotografij @liga (15) (RUS to ENG: independent estimation of number of participans of the action based on maps and photos #10dec #10dec #feb in #omsk) [3] mitingi #10dec #10dec v #omsk ocenka chislennosti i jeffekta ubeditelnaja infografika na karte @liga (14) (RUS to ENG: protests #10dec #10dec in #omsk evaluation of number and effect, infographics on the map @liga (14)	[74] no [32] narod (people) [27] budet (will be) [24] net (no) [24] all [23] ru [21] ochen (very) [21] miting [20] with [20] policija (police) [20] policii (police) [20] est (is) [20] over [20] mitingujushhih (protesters) [19] rossija (Russia) [15] Navalnyj

In this part of our paper, we provide detailed analyses of the example of #GalaxyFamily hashtag that was described previously. All the other results of processing are in English (the main language of the #GALAXYFAMILY hashtag). Firstly, we look through the MFSs that appear in our list and give a brief analysis of the results.

The most popular MFSs as a result of processing #GalaxyFamily are a combination of hashtags, links and photos in Turkish that was used 276 times among our corpus:

(1) RETWEET UYGULAMAS @LIGA @FOTO #GALAXYFAMILY #MILLETINESERIKAPATILAMAZ #BENCEEASK #REKLAM #GOOGLE

It has its limits in automatically defining the model of the situation (researcher need to know Turkish). But our method provides more significant results. For example, the second frequent result can be widely used as a short description of this hashtag:

(2) WELCOME TO THE #GALAXYFAMILY YOU CAN LEARN ALL ABOUT YOUR NEW PHONE IN OUR GUIDE TO THE GALAXY AT @LIGA

We met this MFS 103 times in our corpus. It's the longest (with n = 20) and the most informative one. If we look further we can see some other MFSs that have almost the same meaning but consist of different words (examples 3-4):

(3) WELCOME TO THE #GALAXYFAMILY GET STARTED WITH GREAT TIPS, TRICKS, APP, RECOMMENDATIONS AT @LIGA (38 times)

(4) WELCOME TO THE #GALAXYFAMILY, WE'RE HERE TO HELP CHECK OUT OUR GUIDE TO THE GALAXY AT @LIGA (34 times)

Other popular messages in the list of MFS serve to connect product with different emotions (examples 5-9):

(5) WE'RE HAPPY YOU'RE PART OF OUR #GALAXYFAMILY (52 times)

(6) THANKS FOR THE LOVE #GALAXYFAMILY (42 times)

(7) WE LOVE HAVING YOU IN THE #GALAXYFAMILY (40 times)

(8) THANKS FOR BEING PART OF THE #GALAXYFAMILY (38 times)

(9) THANKS FOR BEING SUCH A LOYAL MEMBER OF THE #GALAXYFAMILY (38 times)

So we can conclude that emotions are an essential part of the concept. Hashtag #GalaxyFamily can be described as a marketing hashtag created to greet new clients and to create the image of Galaxy users and company itself as a kind family which is ready to welcome its new members.

Other important parts are some other brands than Galaxy products. Using method of MFSs we are able to identify the following: Amazon, Samsung, James Bond (examples 10-13).

(10) BLACK FRIDAY DEALS ON AMAZON NOW @LIGA #TOYSRUSKID #DWTS, THANKSGIVING #GALAXYFAMILY (85 times)

(11) CHECK OUT THE LATEST #GALAXYFAMILY PRODUCTS FROM @SAMSUNGMOBILEUS @LIGA (32 times)

(12) BOND, ALL JAMES BOND MOVIES ON BLU RAY FOR ONLY ON AMAZON TODAY ONLY @LIGA #DVD #BLURAY #GALAXYFAMILY (26 times)

(13) AMAZONS AFTER CHRISTMAS BLOWOUT SALE ON NOW @LIGA MGM GRAND #THEVOICE BOXING DAY UFC #UFC #THEGIFTER #GALAXYFAMILY (23 times)

Therefore we can conclude that using the proposed method, the main purpose of hashtag can be described in one or two MFSs. Although for better automatized results, as we've already mentioned, it is recommended to choose two or three MFSs out of issue just to elude the influence of spam information and one or two of the longest MFSs as well (see Table 1).

5.2.2 1-gram Frequent Sequences

1-grams can also be useful in understanding overall hashtag meaning. The frequency of the words can be useful as a basis of concept reconstruction both for marketing and scientific purposes. In this part we talk briefly about the analysis of results.

We divided the processed data into some groups with the exact significance. We didn't consider so called "stop words" which are pronounces, prepositions, articles etc. As a result, we have at least 6 significant groups – slots that are directly connected with the analyzed hashtag. The frequency of use of a particular word form is given in brackets.

Technology: photo (31), apps (23), smartphone (17), smartphones (15), #smartphone (12), software (14), data (12), pics (12), tech (11), videos (11), technology (9), droid (9), #photography (8), androids (7), android (5), #phone (6), content (4), etc.

Market: sold (11), business (11), money (10), service (9), credit (9), contract (8), purchase (8), prices (7), selling (7), etc.

Quality: small (11), cute (11), hot (11), quality (9), quick (9), nuevo (6), exclusive (6), special (6), #new (4), etc.

Emotions: love (82), shit (32), super (20), fucking (12), epic (8), gorgeous (6), etc.

Personalities: member (38), her (24), his (24), girl (18), anyone (17), myself (16), #together (16), mine (13), wife (12), families (11), owners (10), everybody (9), kids (8), somebody (6), men (6), etc.

Brands: SamsungMobileUs (35), #Apple (25), Galaxys (18), HTC (13), Facebook (12), Windows (11), Blackberry (9), #Verizon (9), #Galaxycampus (8), #iPhones (6), #Amazon (8), Verizon (5), etc.

Overall view on the most frequent words in the corpus helps us to reconstruct the model of situation and the image of the product. Predictably there are a lot of words connected with market and technology, emotions or qualities. At the same time there are surprisingly many words connected with Galaxy competitors. As long as tweet producing is uncontrolled, products of other companies can be promoted using competitor's hashtag which is highly undesirable and can be identified using the method of frequent sequences.

5.2.3 Combinations of Hashtags

Another significant feature is combinations of hashtags when different hashtags go together in one tweet.

The way hashtags are used in the phrase can also be important and significant for a research. As long as hashtags are concepts or models of situation, they have more significance than the other words that are frequently met in corpus. The combinations of different hashtags can demonstrate the attitude to the situation, show similar situations, etc. One of the ways of extracting combinations of hashtags is analysis of 2-grams MFSs. Here we provide some of the brief results.

According to data processing based on 2-grams, hashtags mostly used with #GalaxyFamily are the following (the frequency of use of a particular word form is given in brackets):

#BlackFriday (54)
#Note (44)
#Android (36)
#Apple (25)
#CyberMondayMadness (19)
#Samsung (18)
#TeamGalaxy (17)
#TeamIPhone (17)
#Christmas (17)
#Smartphone (12), etc.

The information on the word order can be provided too, if needed.

Among these results we can see references to Galaxy itself (Samsung, TeamGalaxy), competitors (Apple, TeamIPhone, Android), a situation when Galaxy products can be useful, probably as a gift (Christmas), an appropriate time to buy this products (Cyber Monday, Black Friday) and just key words for the hashtag as Smartphone. These combinations can be used, for example, by marketing specialist as long as they indicate the most significant features of a product or the main competitors as considered by Twitter users.

At the same time there are some hashtags that are used by people to promote their messages with another already popular hashtag #GalaxyFamily:

#RT (22 times)
#Follow (19 times)
#FollowMe (14 times)
#NowPlaying (13 times).

All of these hashtags can be used just to promote the message or the user itself: probably, the theme of the messages is not associated with Galaxy products neither with music, etc.

6 Conclusions

In this paper, we described the method of Maximal Frequent Sequences as applied to Twitter data processing. Using it we were able to restore the models of the situation described with four different hashtags: #RussianMeteor, #Grammys, #GalaxyFamily, #10dec.

We confirm the function of hashtags as an indicator of models of situations using the described method. The results demonstrate us the main meanings and ideas of

each hashtag. Both MFSs and 1-grams show connection with the model of situation and can be used for quick and automatized recognition of these models. Our method is language independent: the results in different languages are valid, and for the same hashtag they have the same meaning (#RussianMeteor, #10dec).

During our experiment we met a problem of a spam or commercial tweets that use popular hashtags to promote other information. Our experiment provided a possible way to avoid its influence on final results. Comparing these results to the longest found MFSs we can conclude that the most effective way of automatic summarization of hashtags data would be combination of the most frequent MFSs with the longest ones.

Although MFSs are considered as the most significant, shorter n-grams also help to explain and to analyze hashtags. Using example of #GalaxyFamily hashtag we demonstrate that the most frequent 1-grams can help to reconstruct the overall model of situation and also confirm our assumption of hashtag as a model of situation. Frequent 2-grams can be used both to reconstruct the model situation and to find the frequent hashtag combinations. The results we got for other hashtags (such as the political one, #10dec) show that the same method can be applied to non-commercial hashtags as well.

We consider the proposed method as valid for a rapid designation of the model of situation with any hashtag. Also this method is valid to reconstruct the overall concept of the situation and to determine the keywords or opinions connected with it for marketing or other purposes. As a future work, we use syntactic ngrams for extraction stage to the designation of situation model [23].

Acknowledgments. Work done under partial support of Mexican Government (CONACyT, SNI, UAEM). The authors thank Autonomous University of the State of Mexico for their assistance.

References

1. Krikorian, R.: New Tweets per second record, and how! Available at https://blog.twitter.com/2013/new-tweets-per-second-record-and-how (2013)
2. Davidov, D., Tsur, O., Rappoport, A.: Enhanced sentiment learning using Twitter hashtags and smileys. In: COLING '10 Proceedings of the 23rd International Conference on Computational Linguistics, pp. 241–249 (2010)
3. Cheong, M.: 'What are you tweeting about?': A survey of Trending Topics within Twitter. Clayton School of Information Technology, Australia.
4. Bastos, M.T., Raimundo, R. L. G., Travitzki, R.: Gatekeeping Twitter: message diffusion in political hashtags, Media, Culture & Society, vol. 35 (2), pp. 260–270 (2013)
5. Zangerle, E., Gassler, W., Specht, G.: On the impact of text similarity functions on hashtagrecommendations in microblogging environments. Springer-Verlag, available at: http://link.springer.com/article/10.1007%2Fs13278-013-0108-x (2013)
6. Ma, Z., Sun, A., Cong, G.: On predicting the popularity of newly emerging hashtags in Twitter. Journal of the American Society for Information Science and Technology, vol. 64 (7), pp. 1399–1410 (2013)
7. Page, R.: The linguistics of self-branding and micro-celebrity in Twitter: The role of hashtags. Discourse and Communication, vol. 6 (2), pp. 181–201 (2012)

8. Zimmer, B.: Hastag, You're It. Spectrum.IEEE.org. North American, vol. 50 (4), p. 24 (2013)
9. Hadgu, A.T., Garimella, K., Weber, I.: Political hashtag hijacking in the U.S. In: WWW '13 Companion Proceedings of the 22nd international conference on World Wide Web companion, pp. 55–56 (2013)
10. Sharma, S.: Black Twitter? Racial Hashtags, Networks and Contagion. New formations: a journal of culture/theory/politics, vol. 78, pp. 46–64 (2013)
11. Tsakalidis, A., Papadopoulos, S., Cristea, A.: Predicting elections for multiple countries using Twitter and polls. IEEE Intelligent Systems, vol. 30, pp. 10–17 (2015)
12. Kagan, V., Stevens, A., Subrahmanian, V.S.: Using Twitter Sentiment to Forecast the 2013 Pakistani Election and the 2014 Indian Election. IEEE Intelligent Systems, vol. 30, pp. 2–5 (2015)
13. Bollen, J., Mao, H., Zeng, X.: Twitter Mood Predicts the Stock Market. Journal of Computational Science, no. 2, pp. 1–8 (2011)
14. Popescu, A.-M., Pennacchiott, M.: Dancing with the Stars, NBA Games, Politics: An Exploration of Twitter Users' Response to Events. In: The Fifth International AAAI Conference on Weblogs and Social Media (2011)
15. Yu, Y., Wang, X.: World Cup 2014 in the Twitter World: A big data analysis of sentiments in U.S. sports fans' tweets. Computer in Human Behavior, vol. 48, pp. 392–400 (2015)
16. Atiagina, A.: Twitter as a new discoursive practice [Twitter kak novaja discoursivnaja practika (in Russian)]. Ph.D. Thesis, Omsk State University, Omsk (2014)
17. Dijk Van, T.A., Kinch, B.: Strategies and understanding of connected text [Strategii ponimaniia sviaznogo teksta (in Russian)]. In: New in foreign linguistics [Novoe v zarubezhnoi lingvistike (in Russian)], vol. 23, pp. 153–211 (1988)
18. Dijk Van, T.A.: Language, cognition, and communication [Iazyk, poznanie, kommunikatsiia (in Russian)], Moscow (1989)
19. Ledeneva, Y., Gelbukh, A., García-Hernández, R.A.: Terms Derived from Frequent Sequences for Extractive Text Summarization. LNCS 4919, pp. 593–604, Springer-Verlag (2008)
20. Ledeneva, Y., García-Hernández, R., Gelbukh, A.: Graph Ranking on Maximal Frequent Sequences for Single Extractive Text Summarization. LNCS 8404, pp. 466–480, DOI 10.1007/978-3-642-54903-8_39 (2014)
21. Garcia-Hernandez, R.A., Martínez-Trinidad, J. F., Carrasco-Ochoa, J.A.: A Fast Algorithm to Find All the Maximal Frequent Sequences in a Text. LNCS 3287, pp. 478–486 (2004)
22. García-Hernández, R.A., Martínez-Trinidad, J.F., Carrasco-Ochoa, J.A.: A New Algorithm for Fast Discovery of Maximal Sequential Patterns in a Document Collection. LNCS 3878, pp. 514–523 (2006)
23. Sidorov, G.: N-gramas sintácticos no-continuos. Polibits, vol. 48, pp. 67–75 (2013)

Uso de técnicas de agrupamiento en la clasificación de estilos de aprendizaje

Fernando Gudino-Penaloza[1], Miguel González-Mendoza[2] y Jaime Mora-Vargas[2]

[1] Universidad Nacional Autonoma de México, DF, México

[2] Tecnológico de Monterrey, Campus Estado de México, México

fernando.gudino@comunidad.unam.mx,{mgonza,jmora}@itesm.mx
http://www.unam.mx
http://www.itesm.mx

Resumen. El presente trabajo muestra la utilización de k-means y Fuzzy c-means para la determinación de estilos de aprendizaje en escuelas de educación básica, esto con el fin de establecer estrategias de enseñanza acordes con los perfiles de los alumnos. Dicho análisis de perfiles se basa en el modelo de programación neuro-lingüistica que divide a los estilos en tres, dependiendo de la forma en que favorecen la manera de percibir su entorno.

Palabras clave: algoritmos de agrupamiento, teoría neuro-lingüistica, k-means, fuzzy C-means.

1. Introducción

Los estilos de aprendizaje son los rasgos cognitivos, afectivos y fisiológicos que sirven como indicadores relativamente estables, de cómo el alumno percibe interacciones y responde a su ambiente de aprendizaje. Es decir, tienen que ver con la forma en que los estudiantes estructuran los contenidos, forman y utilizan conceptos, interpretan la información, resuelven los problemas, seleccionan los medios de representación (visual, auditivo, kinestésico), etc. [24]. Esto último basado en el modelo cognitivo de programación neurolingüistica(PLN) [23].

Cuando un profesor identifica el estilo de aprendizaje de sus alumnos tiene la oportunidad de saber por dónde y cómo encaminar su enseñanza; si por ejemplo; en un aula un 90 % de los alumnos son kinestésicos, un 6 % auditivos y 4 % visuales, no se debe solo basar en la mayoría pues el 10 % restante requieren de estrategias de enseñanza diferentes, al identificar a los alumnos y sus diferentes estilos de aprendizaje el profesor no basará su planeación solo en uno, si no en los tres pero ahora tiene la oportunidad de saber cómo cada uno de sus alumnos aprende y al desarrollar los tres estilos de aprendizaje en su clase

promueve también que todos desarrollen los tres, de esta manera estará formando alumnos kinestésicos, visuales y auditivos [23], entonces ellos podrán aprender posteriormente de cualquier modo que se les enseñe.

Una de las dificultades en el momento de la implementación de PLN es la complejidad del modelo para lograr una clasificación de un individuo o grupo de individuos dentro de los estilos principales:kinestésicos, visuales y auditivos [6]. La principal manera de hacerlo es el uso de encuestas [11,1,21]. Sin embargo la complejidad y diversidad de dichos test hacen difícil la interpretación de los resultados.

Es por ello que se decide utilizar un sistema basado en técnicas de agrupamiento - k-means, fuzzy C-means- , como clasificador para lograr una categorización de los perfiles de estudiantes de segundo nivel de educación secundaria, que permita determinar los estilos individuales y grupales de los alumnos para la implementación de estrategias adecuadas para el desempeño del grupo. Las técnicas de agrupamiento nos permiten utilizar parametros cualitativos de manera cuantitativa y de esta manera determinar de manera precisa la estrategia didáctica adecuada. El objetivo final de este proyecto es proporcionar un conjunto de herramientas a docentes cuyas nociones de las técnicas de Inteligencia Artificial no son de uso común, pero que tiene la necesidad de un instrumento útil para el desarrollo de sus clases. Así mismo se usa el perfil para ajustar la planeación y desarrollo de las clases durante el ciclo escolar, y finalmente medir la efectividad de la decisión tomada a priori.

2. Representación mental de la información según la programación neuro-lingüisitica

La Programación Neuro-Lingüisitica, también conocida como VAK(Visual-Auditivo-Kinestésico), toma en cuenta el criterio neurolingüístico, que considera que la vía de ingreso de la información resulta fundamental en las preferencias de quien aprende o enseña. Por ejemplo, en una clase de historia al presentar un video de la biografia de un personaje; ¿qué le es más fácil recordar: la cara, el nombre o la impresión producida por el personaje?

De manera sintetizada, la teoría maneja tres categorias para representar mentalmente la información: el visual(imágenes abstractas y concretas), el auditivo(voces, sonidos, música) y el kinestésico(sabores, o sentimientos y emociones). La mayoría de nosotros utilizamos los sistemas de representación de forma desigual, potenciando unos e infra-utilizando otros. Los sistemas de representación se desarrollan más cuanto más los utilicemos. Utilizar más un sistema implica que hay sistemas que se utilizan menos y, por lo tanto, que distintos sistemas de representación tendrán distinto grado de desarrollo [3].

El concepto de PNL hace referencia a la unión de tres términos Programación (aptitud para implementar programas de comportamiento), Neuro (percepciones sensoriales que marcan el estado emocional de un individuo) y Lingüística (medios verbales y no-verbales que utilizamos los seres humanos para comunicarnos).

Para implementar el estilos de aprendizaje adecuado es imprescindible realizar un diagnóstico adecuado. Para ello existen instrumentos y herramientas que posibiliten este diagnóstico. Una de las causas que ha impedido un mayor desarrollo y aplicaciones de este enfoque de la educación reside, precisamente, en la pluralidad de definiciones, enfoques y herramientas que se ponen a nuestra disposición[5,4,8].

Sin embargo podemos decir que existen 3 modelos básicos para clasificar las distintas herramientas y modelos [4]. El primer modelo se centra en las preferencias instruccionales y ambientes de aprendizaje[5].

El segundo modelo, se basa en las preferencias acerca de cómo se procesa la información. Esta teoría facilita al estudiante identificar sus preferencias vitales en el modo de aprendizaje en el aula [11]. El tercer nivel, se relaciona con las diferencias de aprendizaje debidas a la personalidad [17].

Después de analizar los tres modelos, se logra la conclusión de que es el segundo modelo es el más apropiado, toda vez que permite el acercamiento a la identificación de los estilos de aprendizaje de los estudiantes.

3. Detección automática de perfiles

En la literatura,podemos encontrar técnicas de modelado de perfiles de estudiantes en ambientes virtuales[25] y en ambientes reales,tales como: métodos basados en reglas[10], razonamiento basado en casos[19] o redes bayesianas[9].Estas obras pueden cubrir diferentes aspectos de la conducta del alumno y el conocimiento.Nuestro trabajo se puede colocar entre aquellos que modelan los estilos de aprendizaje de acuerdo a sus preferencias en los sistemas mentales de representación [10].

3.1. Algoritmos de agrupamiento

En las técnicas de agrupación el objetivo es dividir a los datos en clases o grupos homogéneos de modo que los elementos de la misma clase son tan similares como sea posible; mientras que elementos de diferentes clases son tan diferentes como sea posible. Aunque dichos algoritmos no estan diseñados para la clasificación. Podemos adaptarlo a los efectos de clasificación supervisada.

Dependiendo de los datos y de la aplicación, diferentes tipos de medidas de similitud pueden ser utilizados para identificar las clases. Algunos ejemplos de valores que pueden ser utilizados como medidas de similitud incluyen la distancia, la conectividad, y la intensidad. Los agrupamientos detectados dependen del algoritmo empleado, del valor dado a sus parámetros, de los datos utilizados y de la medida de similaridad adoptada [22]. Se han propuesto cientos de algoritmos de agrupamiento más o menos específicos. Según se use o no una función criterio se distinguen los algoritmos paramétricos y no paramétricos[2].

3.2. Formulación matemática del problema

Una definición del problema del agrupamiento, puede enunciarse de la siguiente manera:

Dado un conjunto S, de N elementos, se quiere encontrar la partición $S_1, S_2, ..., S_k$, tal que cada uno de los N elementos se encuentre sólo en un grupo S_i, y que cada elmento sea más similar a los elementos de su mismo grupo que a los elmentos asignados a los otros grupos[14].

- $S_i \neq \oslash$
- $S_i > 0$
- $i \neq j \longrightarrow S_i \bigcap S_j = \oslash$
- $N = \coprod S_i = \{X_1, X_2; ..., X_N\} | X_i \in \Re^n$

Para poder definir medidas de semejanza entre los objetos a agrupar, éstos se representan mediante vectores $v = (a_1, a_2, ..., a_m)$, donde cada componente del vector es el valor de un atributo del objeto. De esta forma, cada uno de los objetos a agrupar es un punto en un Espacio Euclideano de n dimensiones, $\Re^n$. Matemáticamente, el problema puede formularse como la minimización de:

$$f(W, Z, X) = \sum_{i=1}^{n} \sum_{j=1}^{k} w_{ij} ||X_i - Z_j||^2$$

$$\sum_{j=1}^{k} w_{ij} = 11 \leq i \leq n$$

$$w_{ij} \in \{1, 0\}$$

- X_i es el vector patrón correspondiente al ejemplo i-ésimo ($X_i \in \Re$).
- Z_j es el centro del j-ésimo cluster ($Z_j \in \Re$).
- W es la matriz de pertenencia ($n \times k$) tal que w_{ij} es 1 si $X_i \in S_j$ y 0 en caso contrario

La función $f$no es convexa, por lo que pueden existir mínimos locales. Así mismo la minimización de la función f requiere conocer a priori el número deseado de agrupamientos k (si no el problema sería trivial). No obstante, existen técnicas que permiten ajustar el número de agrupamientos (fusionando y dividiendo agrupamientos), así como tratar elementos discordantes debidos, por ejemplo, a ruido en la adquisición de datos [26].

Agrupamiento k-Means El algoritmo de las K- medias($k - means$) es probablemente el algoritmo de agrupamiento más conocido. Es un método de agrupamiento heurístico con número de clases conocido (K). El algoritmo está basado en la minimización de la distancia interna (la suma de las distancias de los patrones asignados a un agrupamiento al centroide de dicho agrupamiento). El algoritmo es sencillo y eficiente, lo cual lo coloca como una opción viable

para aquellos que no son expertos en la técnica. Además, procesa los patrones secuencialmente (por lo que requiere un almacenamiento mínimo). Aunque dicho algoritmo está sesgado por el orden de presentación de los patrones y su comportamiento depende enormemente del parámetro K si se selecciona adecuadamente el numero de agrupamientos el algoritmo se comporta como un buen clasificador, ya que los elementos internos son cercanos y los elementos externos se alejan. En este algoritmo la distancia cuadrática Euclideana es usada como medida discriminante:

$$d(x_i, x_{i'}) = \sum_{j=1}^{n} (x_{ij} - x_{i'j})^2 = \|x_i - xi'\|^2$$

De igual forma los puntos de dispersión pueden ser escritos como:

$$W(C) = \sum_{k=1}^{K} N_k \sum_{C(i)=k} \|x_i - xi'\|^2$$

donde $\bar{x_k} = (\bar{x}_{1k}, ..., \bar{x}_{nk})$, es el vector de medias asociado con el k-ésimo grupo, y $Nk = \sum_{i=1}^{N} I(C(i) = k)$.

Agrupamiento difuso C-Means En muchas situaciones cotidianas ocurre el caso que un dato está lo suficientemente cerca de dos grupos de tal manera que es difícil etiquetarlo en uno o en otro, esto se debe a la relativa frecuencia con la cuál un dato particular presenta características pertenecientes a grupos distintos y como consecuencia no es fácilmente clasificado.

En agrupamiento difuso, los puntos de datos pueden pertenecer a más de un grupo, y asociado con cada uno de los puntos son los grados de miembros que indican el grado en que los puntos de datos pertenecen a los diferentes grupos.

Fuzzy c-means (FCM) es un algoritmo que se desarrolló con el objetivo de solucionar los inconvenientes de la técnica de K-means. El algoritmo FCM asigna a cada dato un valor de pertenencia dentro de cada grupo y por consiguiente un dato específico puede pertenecer parcialmente a más de un grupo. A diferencia del algoritmo k-means clásico que trabaja con una partición dura, FCM realiza una partición suave del conjunto de datos, en tal partición los datos pertenecen en algún grado a todos los grupos; una partición suave se define formalmente como sigue: Sea X conjunto de datos y x_i un elemento perteneciente a X, se dice que una partición $P = (C_1, C_2, \ldots, C_c)$ es una partición suave de X si y solo si las siguientes condiciones se cumplen:

- $\forall x_i \in X \forall C_j \forall P 0 \leq \mu_{cj}(x_i) \leq 1$
- $\forall x_i \in X \exists C_j \forall P / 0 \leq \mu_{cj}(x_i) \leq 1$

Donde $\mu_{cj}(x_i)$ denota el grado en el cuál $x_i$pertenece al grupo C_j.[27] En contraste con k-Means, FCM puede asignar un caso a más de un grupo, con diferentes "grados de pertenencia". Como primer paso, Fuzzy c-Means calcula los centros de los conjuntos difusos para el número elegido de grupos. Entonces

calcula el grado de pertenencia de cada caso, respecto a cada conjunto, y para cada variable de entrada mediante:

$$\mu_{ci}(x) = \frac{1}{\sum_{j=1}^{k} \left(\frac{\|x_i - c_i\|^2}{\|x_i - c_j\|^2} \right)^{\frac{1}{m-1}}}$$

Fuzzy c-Means se basa en la minimización de la siguiente función objetiva:

$$J_m = \sum_{i=1}^{N} \sum_{J=1}^{C} u_{ij}^m \|x_i - c_j\|^2$$

donde m es cualquier número real mayor que 1 el cual es un peso que determina el grado en el cuál los miembros parciales de un conjunto afectan el resultado, x_i es el i-esimo caso de datos, $\mu_i j$ es el grado de pertenencia de x_i en cada conjunto j, c_j es el centro del conjunto j, y $||*||$ es cualquier norma que expresa la similitud entre un caso y el centro del conjunto difuso.

4. Determinación de estilos de aprendizje mediante técnicas de agrupamiento

La presente investigación la tipificamos según los siguientes criterios:

- Investigación descriptiva correlacional ya que se pretende establecer la relación de la programación neurolingüística con el aprendizaje estratégico.
- De diseño no experimental debido a que no se construye ninguna situación ni se van a manipular las variables de la investigación.
- Por el método de estudio de las variables: Es investigación mixta:cualitativa-cuantitativa, pues los datos consignados son tanto categoricas y numericas.

El proyecto de investigación trata las siguientes etapas:

1. Determinación del perfíl de estudiante; tanto individual como grupal mediante las diferentes técnicas.
2. Evaluación parcial del desempeño de los alumnos.
3. Aplicación de las estrategias de reforzamiento segun el perfíl indivudal y el grupal(trabajo en progreso).
4. Evaluación de los resultados a lo largo de un periodo escolar completo (trabajo en progreso).

4.1. Población del estudio

La población está constituida por 12 grupos con 348 estudiantes inscritos en el segundo grado de educación secundaria en México, dividido en 3 Escuelas ; dos publica y una privada distribuidos como se muestra en la Tabla 1.

Para lo cual de los 12 grupos se selecciono de manera aleatoria un grupo como control(22 individuos), cuyo rendimiento fue comparado a lo largo de cinco

Tabla 1. Distribución de alumnos por tipo de escuela.

Tipo de Escuela	No. Grupos	No. Alumnos
Publica	8	252
Privada	4	96

bimestres. a dicho grupo se le aplico la evaluación inicial; pero se le permitio a los docentes aplicar su planeacion escolar inicial. A los demas grupos se ajusto la estrategia de enseñanza inicial permitiendo a los encargados de cada grupo ajustar de acuerdo a los resultados del análisis de perfiles. Por último destacamos que para dicho estudio comparativo se seleccionaron las materias de : Español, Matemáticas, Ciencias e Historia, las cuales han sido objeto de criticas por diversas organizaciones, debido a los pobres resultados en pruebas como PISA [20] y ENLACE [7].

4.2. Recopilación de datos

Para la recopilacion y procesamiento de la información se utilizo una version del test Metts , mostrado en la figura 1, el cual a su vez esta basado en el modelo de Bandler y Grinder [1] e inspirado por Kolb[12].

El cuestionario está conformado por 24 reactivos, que miden tres dimensiones: visual, auditivo y kinestésico. La escala es aplicada en forma individual o colectiva y el tiempo de aplicación es aproximadamente de 10 a 15 minutos. Los ítems representan una dimensión y categoría de análisis como es: Canal perceptual: Vías de percepción de la información en lo Visual, Auditivo y Kinestésica (V. A. K).

Para la aplicación de la prueba se realizaron en las mismas aulas de aprendizaje de los estudiantes de 2 grado de secundaria. Se aplicó la prueba en forma grupal, con sesiones de 10 o 15 minutos; antes de la aplicación de la prueba se orientó a cada estudiante mediante ejemplos en la pizarra con un modelo de la prueba aplicada elegidos para que llenen el cuestionario de estilos de aprendizaje. Esto se llevó a cabo el primer bimestre del año escolar 2014.

Posteriormente se realiza un análisis de los datos grupo por grupo para determinar la tendencia de cada uno de ellos. Así mismo se diferencia entre las instituciones de tipo publico y privada para conocer las tendencias en ambos sectores, por último se realiza un análisis de acuerdo al genero del entrevistado, para comparar los estilos representativos de cada genero.

4.3. Procesamiento de datos

Reducción de la dimensionalidad Intuitivamente, la covarianza es la medida de variación mutua de dos variables aleatorias. Es decir, la covarianza tendrá un valor positivo más grande para cada pareja de valores que difieren del valor medio con el mismo signo (+ o -). Asimismo, la covarianza tendrá un valor negativo más

Nombre:..... **Fecha:**.....

Por favor, responda Ud. verdaderamente a cada pregunta. Responda Ud. según lo que hace actualmente, no según lo que piense que sea la respuesta correcta. Use Ud. la escala siguiente para responder a cada pregunta: Ponga un círculo sobre su respuesta.

1 = Nunca 2 = Raramente 3 = Ocasionalmente 4 = Usualmente 5 = Siempre

1	Me ayuda trazar o escribir a mano las palabras cuando tengo que aprenderlas de memoria	1	2	3	4	5
2	Recuerdo mejor un tema al escuchar una conferencia en vez de leer un libro de texto	1	2	3	4	5
3	Prefiero las clases que requieren una prueba sobre lo que se lee en el libro de texto	1	2	3	4	5
4	Me gusta comer bocados y mascar chicle, cuando estudio	1	2	3	4	5
5	Al prestar atención a una conferencia, puedo recordar las ideas principales sin anotarlas	1	2	3	4	5
6	Prefiero las instrucciones escritas sobre las orales	1	2	3	4	5
7	Yo resuelvo bien los rompecabezas y los laberintos	1	2	3	4	5
8	Prefiero las clases que requieran una prueba sobre lo que se presenta durante una conferencia	1	2	3	4	5
9	Me ayuda ver diapositivas y videos para comprender un tema	1	2	3	4	5
10	Recuerdo más cuando leo un libro que cuando escucho una conferencia	1	2	3	4	5
11	Por lo general, tengo que escribir los números del teléfono para recordarlos bien	1	2	3	4	5
12	Prefiero recibir las noticias escuchando la radio en vez de leerlas en un periódico	1	2	3	4	5
13	Me gusta tener algo como un bolígrafo o un lápiz en la mano cuando estudio	1	2	3	4	5
14	Necesito copiar los ejemplos de la pizarra del maestro para examinarlos más tarde	1	2	3	4	5
15	Prefiero las instrucciones orales del maestro a aquellas escritas en un examen o en la pizarra	1	2	3	4	5
16	Prefiero que un libro de texto tenga diagramas gráficos y cuadros porque me ayudan mejor a entender el material	1	2	3	4	5
17	Me gusta escuchar música al estudiar una obra, novela, etc.	1	2	3	4	5
18	Tengo que apuntar listas de cosas que quiero hacer para recordarlas	1	2	3	4	5
19	Puedo corregir mi tarea examinándola y encontrando la mayoría de los errores	1	2	3	4	5
20	Prefiero leer el periódico en vez de escuchar las noticias	1	2	3	4	5
21	Puedo recordar los números de teléfono cuando los oigo	1	2	3	4	5
22	Gozo el trabajo que me exige usar la mano o herramientas	1	2	3	4	5
23	Cuando escribo algo, necesito leerlo en voz alta para oír como suena	1	2	3	4	5
24	Puedo recordar mejor las cosas cuando puedo moverme mientras estoy aprendiéndolas, por ej. caminar al estudiar, o participar en una actividad que me permita moverme, etc.	1	2	3	4	5

Fig. 1. Cuestionario de Estilos de Aprendizaje, tomado de [16].

grande para cada pareja de valores que difieren del valor medio con signo distinto (+ o -). Si la covarianza mutua es cero entre dos variables aleatorias, esto indica que no existe una correlación entre ellas. Supongamos que existe una muestra de n pares de observaciones de dos variables$X$ e Y, $X : x_1, x_2, ..., x_n; Y : y_1, y_2, ...y_n$ definimos la covarianza como:

$$S_{xy} = \frac{1}{n}\sum_{i=1}^{n}(x_i - \acute{x})(y_i - \acute{y})$$

La covarianza ha demostrado ser un método sencillo para la reducción de variables, así como ser un mecanismo confiable para correlacionar y reducir el numero de variables en un estudio en el cual el numero de las mismmas se podria considerar como alto. El presente estudio utiliza dicho análisis para reducir variables, la tabla de resultados muestra los datos experimentales tanto con el total de las variables como con el conjunto reducido.

Método tradicional Para determinar el puntaje total por grupo primero se determinó el estilo individual para lo cual se procede a una suma algebraica de resultados totales tal como se muestra en la Tabla 2, aquel estilo que obtenga mayor puntuación determina el estilo individual. Para el cálculo grupal se suman los individuos por estilo y luego se convierte a frecuencia. La más alta frecuencia obtenida en cualquiera de los estilos indica la preferencia por un estilo de aprender. Los ítems representan una dimensión y categoría de análisis como es: Canal perceptual: Vías de percepción de la información en lo Visual, Auditivo y Kinestésica (V. A. K). El principal objetivo es identificar los estilos de aprendizaje, desde la preferencia de cada estudiante por medio del cuestionario.

Tabla 2. Relación de reactivos a agregar.

Tipo de Estilo	No. de pregunta a agregar
visual	1,3,6,9,10,11,14
Auditivo	2,5,12,15,17,21,23
Kinestesico	4,7,8,13,19,22,24

Fueron eliminadas las preguntas 16-18-20 para que quedaran la misma cantidad de preguntas por cada estilo. Una vez completado el test, deberán obtenerse tres puntajes, correspondientes a los tres estilos de aprendizaje, los que definirán el perfil del estilo del alumno.

4.4. Resultados

Análisis de covarianza Después de realizar el análisis de covarianza reducimos el número de variables a analizar tal como lo muestra la Tabla 3.

Podemos verificar que el numero de variables se redujo a solo cuatro por estilo de aprendizaje reduciendo de un total de 26 variables a 12.

Tabla 3. Relación de reactivos a agregar después del análisis de covarianza.

Tipo de Estilo	No. de pregunta a agregar
visual	1,6,9,14
Auditivo	2,5,15,21
Kinestesico	4,8,13,19

Efectividad de clasificación Uno de los problemas mostrados en la técnica tradicional de clasificación de perfiles es el hecho de su incapacidad de decidir el estilo cuando existe un numero igual de puntuación en 2 o 3 estilos. La Tabla 4.4 muestra la efectividad de los algoritmos para clasificar los ejemplos.

Tabla 4. Relación de ejemplos bien clasificados.

Técnica	Porcentaje de ejemplos bien clasificados
Tradicional	77.30
Tradicional reducido	90.90
k-means(ambas versiones)	100
Fuzy c-means(ambas versiones)	100

Clasificación por estilos La Tabla 5 muestra los resultados individuales del estudio de perfil de aprendizaje. Dicha tabla muestra los datos tanto originales como reducidos por covarianza.Solo se consideran los ejemplos bien clasificados.

Tabla 5. Relación de ejemplos clasificados por estilo de aprendizaje.Se uso k=3 para K-means y FCM, además de m=1.0 para FCM. Para el caso de FCM el número se determino por aquel agrupamiento con mayor función de pertenenecia.

	Estilos de aprendizaje(en porcentaje)		
Técnica	Visual	Auditivo	Kinestesico
Tradicional	20.69	31.03	48.28
Tradicional reducido	30.70	30.70	38.60
k-means	43.56	23.62	32.82
k-means reducido	42.04	21.25	36.71
Fuzy c-means	30.67	36.2	33.13
Fuzy c-means reducido	40.13	20.7	39.17

De acuerdo a los especialistas [1,15,12], el canal visual es el más utilizado por los estudiantes, con porcentajes que van del 40 al 50 porciento, el canal

kinestesico tiene valores similares entrew 35 y 40 porciento mientras que el canal auditivo es el menos usado con porcentajes de entre el 20 y 30 porciento. La tabla anterior demuestra que los datos son consistentes al usar las tecnicas de agrupamiento, por otro lado la técnica de conteo tradicional falla debido a que los datos son incompletos, esto debido a que existe un porcentaje de la población que no pudo ser clasificada dentro de algunas de las tres categorias. Adicionalmente podemos verificar que la reducción de variables mediante análisis de covarianza ayudo a mejorar el desempeño de las técnicas aplicadas y obtener resultados más cercanos a lo que se puede verificar por otros estudios.

5. Conclusiones

Determinar los estilos de aprendizaje presentes en un estudiante o grupo de estudiantes permite a los profesores establecer estatregias para que los alumnos sean capaces de desarrollar todas sus habilidades y no solo un canal de aprendizaje, Las técnicas tradicionales si bien han sido utiles para determinar a priori los estilos, tiene dificultad para determinar el mismo cuando los estudiantes presentan poca tendencia haia un estilo en particular.Es por ellos que las técnicas de agrupamiento nos ayudan a seleccionar el perfil de manera mas exacta. Con ello se pueden diseñar estrategias mas adecuadas. De las técnicas aplicadas, Fuzzy C-means nos permite perfilar mas detalladamente a los individuos y grupos debido a que nos indica los grados o tendencias en cada estilo y no solo los categoriza. Por otra parte la reduccion de variables ayudo a mejorar la capacidad de las técnicas para clasificar, logrando con ello que los resultados fueran mas cercanos a los reportados por los expertos en enseñanza. en un trabajo futuro se pretende utilizar covarianza difusa, para lograr un mejor desempeño en Fuzzy c-means. Asi mismo se buscara trabajr con otros niveles del parametro m en dicho algoritmo. Como trabajo futuro se propone analizar los resultados del desempeño global e individual de los grupos clasificados y compararlos con el grupo de control, con el fin de verificar la efectividad de las técnicas utilizadas, así mismo utilizar otras técnicas como SVM para comparar su desempeño.

Referencias

1. Alonso, C., Domingo, J., Honey. P.: Los estilos de aprendizaje: procedimientos de diagnóstico y mejora. Ediciones Mensajero, España (1994)
2. Berkhin, P. :A survey of clustering data mining techniques. In Grouping multidimensional data, Springer Berlin Heidelberg (2006)
3. Camacho, R.:¡Manos arriba! El proceso de enseñanza – aprendizaje. ST Editorial, México (2007)
4. Curry,L.:Integrating Concepts of Cognitive Or Learning Style: A Review with Attention to Psychometric Standards. Learning Styles Network (1987)
5. Dunn, R., Dunn K.,Price, G.E.:Learning Styles Inventory (LSI): An Inventory for the Identification of How Individuals in Grades 3 through 12 Prefer to Learn. Lawrence, KS, Price Systems (1985)

6. Einspruch, E.L., Forman, B.D.: Observations concerning research literature on neuro-linguistic programming.J. Counseling Psychology, vol. 32 (4), pp. 589–596 (1985)
7. Evaluación Nacional de Logros Académicos en Centros Escolares (ENLACE). http://www.enlace.sep.gob.mx/content/gr/docs/2013/ENLACE_InformacionBasica.pdf
8. Felder, R. M., Spurlin, J.: Applications, reliability and validity of the index of learning styles. International Journal of Engineering Education, vol. 21 (1), pp. 103–112 (2005)
9. García, P., Amandi, A., Schiaffino, S., Campo, M.:. Evaluating Bayesian networks' precision for detecting students' learning styles. Computers and Education, vol. 49 (3), pp. 794–808 (2003)
10. Graf, S., Viola.: Automatic Student Modelling for Detecting Learning Style Preferences in Learning Management Systems. In Proceedings of the IADIS international conference on cognition and exploratory learning in digital age (CELDA 2007), pp. 172–179 (2007)
11. Kolb, D.: Experiential learning: Experience as the source of learning and development. Prentice-Hall, USA (1984)
12. Kolb,D.: Learning Style Inventory: Self Scoring Inventory and Interpretation Booklet. McBer and Company, USA (1985)
13. Labinowicz, E.: Introduccion a Piaget Pensamiento, Aprendizaje, Enseñanza. Pearson Educación, México (1998)
14. Likas, A., Vlassis, N., Verbeek, J.J.: The global k-means clustering algorithm. Pattern Recognition, vol. 36, pp. 451–461 (2003)
15. Lozano, A.:Estilos de aprendizaje y enseñanza. Editorial Trillas, México (2000)
16. Metts, R.: Teorías y ejercicios. Derechos de propiedad literaria Ralph Metts S.J., Santiago de Chile, pp. 32 (1999)
17. Myers, I. B., McCaulley, M. H., Quenk, N. L., Hammer, A. L.: MBTI Manual (A guide to the development and use of the Myers Briggs type indicator). Consulting Psychologists Press (1998)
18. Nettleton, D., Baeza-Yates, R.: Web Retrieval: techniques for the aggregation and selection of queries and answers. In First Spanish Symposium on Fuzzy Logic and Soft Computing, Spain, pp. 183–190 (2005)
19. Peña, C., Narzo, J.,De la Rosa, J.: Intelligent agents in a teaching and learning environment on the web. In Proceedings of the international conference on advancedlearning technologies (2002)
20. Programme for International Student Assessment (PISA). http://www.oecd.org/pisa/keyfindings/PISA-2012-results-mexico-ESP.pdf
21. Román, J. M., Gallego, S.: Escalas de Estrategias de Aprendizaje,ACRA. TEA Ediciones, España (1994)
22. Rousseeuw, P.J.,Kaufman, L.: Finding Groups in Data: An Introduction to Clúster Analysis. Wiley (1990)
23. Stahl,T.: PNL, introducción a la programación neurolingüística : para qué sirve, cómo funciona y quién puede beneficiarse de ella. Paidós, México (2013)
24. Sternberg, R.: Thinking Styles, Cambridge University Press (2001)
25. Cassidy, S.: Learning styles: An overview of theories, models, and measures. In Educational Psychology, vol. 24 (4) (2004)
26. Xu, R., Wunsch, D.: Survey of clustering algorithms. In IEEE Transactions on Neural Networks, IEEE Press, USA, pp. 645–678 (2005)
27. Yen , J., Langari, R.: Fuzzy Logic Intelligence Control and Information. Prentice Hall, USA (1999)

Una propuesta de sistemas distribuidos con componentes autónomos definidos en SCEL e implementados en Erlang

Mónica García, Manuel Hernández, Ricardo Ruiz y Felipe Trujillo-Romero

Universidad Tecnológica de la Mixteca,
Huajuapan de Léon, Oaxaca, México

monicagg@mixteco.utm.mx
hg.manuel@gmail.com
rruiz,ftrujillo@mixteco.utm.mx

Resumen. *SCEL* es un formalismo para diseñar sistemas distribuidos con componentes autónomos. Erlang es un lenguaje de programación especializado en el tratamiento de problemas de cómputo distribuido. En este artículo, afirmamos que los programas escritos en Erlang son adecuados para proveer el soporte de software de *componentes autónomos* tal como están descritos por el formalismo SCEL. Damos una instancia mediante la descripción del núcleo de programación de un componente autónomo típico en un escenario robótico.

Palbaras clave: Erlang, sistema distribuido, sistema autónomo.

1. Introducción

SCEL (acrónimo del inglés *Software Component Ensemble Language*) [12] es un formalismo de modelado de sistemas distribuidos para el diseño, desarrollo e interacción de *componentes autónomos* y con un soporte teórico ahora bien establecido ([1,8]). La *programación en Erlang* (PE) [3,4,5] está apoyada principalmente por el lenguaje de programación funcional *Erlang*, y por la plataforma de desarrollo *OTP* (del inglés *Open Telecom Platform*). Este artículo presenta una propuesta para utilizar a los programas escritos en Erlang como una herramienta en la parte de software concerniente al manejo de comunicaciones de un componente autónomo según está definido por SCEL. Un objetivo similar fue planteado en [16] para el caso de agentes, en un contexto más amplio que el nuestro, al no utilizar Erlang solo para la parte de comunicaciones. Nosotros, y similarmente a [12], emplearemos un ejemplo de robots para describir el potencial de nuestra propuesta, afirmando que este esquema de trabajo se coordina bien con el formalismo de SCEL, manteniendo así una adecuada pureza teórica en una aplicación práctica.

Estructura de este trabajo

El panorama general de este trabajo es como sigue: Primero, daremos un recuento general del formalismo SCEL; segundo, presentaremos algunos fundamentos de la programación en Erlang (PE); finalmente, describiremos cómo Erlang es un buen candidato de algunas partes descritas en la teoría por el formalismo SCEL, dado su modelo computacional distribuido subyacente vía paso de mensajes sin recursos compartidos, anotando algunas conclusiones finales.

2. El formalismo SCEL

SCEL tiene como principal objetivo organizar algunas herramientas teóricas para el el diseño de sistemas distribuidos que consisten de *componentes autónomos*, mediante la identificación de *acciones* que tales componentes pueden realizar, y con la coordinación entre componentes para formar *grupos*, siendo el mecanismo principal de comunicación el intercambio o paso de mensajes. Cada componente autónomo (CA) está equipado con una *interface*, que a su vez consiste de una colección de *atributos*, tales como la identidad del CA, su memoria disponible, su poder computacional, su nivel de energía, su posición geográfica, su membresía de grupos, y otros más. Estos atributos permiten conformar grupos de CAs mediante la identificación de valores de atributos en común o por medio de la satisfacción de predicados. Cuando se forman grupos de CAs (GCA), tenemos posibilidad de realizar tareas coordinadas, que permiten una gran flexibilidad de desempeño y eficacia en el logro de los objetivos para los que fue diseñado el sistema.

Carecterizamos en SCEL un CA por lo siguiente:

1. *Conductas*, dadas por la ejecución secuencial o en paralelo de *acciones*, mismas que permiten un tratamiento teórico mediante sistemas con transiciones etiquetadas;
2. *conocimiento*, el cual permite que un CA almacene datos propios y en su caso, mediante *interfaces*, tenga acceso también a datos que no son los propios; se permite que este conocimiento de un CA sea manipulado internamente o por otros CAs (agregando, examinando, o eliminando la información);
3. *atributos*, los cuales se informan o se recuperan mediante las interfaces;
4. *políticas*, que permiten controlar un CA localmente y en su caso, globalmente.

Las propiedades auto-* (*self-* properties*) son una parte imprescindible de los modernos componentes autónomos. SCEL modela adecuadamente este tipo de propiedades, pues se catalogan como atributos que pueden ser visibles o no mediante interfaces y de acuerdo con las políticas vigentes. Ejempo de tales propiedades son los sensores de auto-evaluación (tales como los de nivel batería existente), de auto-protección (como los sensores de inclinación o temperatura, o bien la auto-detección de averías), o los de revisión de comunicaciones (detectando la intensidad de señal de comunicación, por ejemplo).

La siguiente es una seudo-ecuación que describe concisamente un CA:

$$CA = \mathcal{I}[\mathcal{K}, \Pi, P]$$

donde $\mathcal{I}$ representa la interface, $\mathcal{K}$ el conocimiento, y Π es la política que en un momento dado está vigente (se pueden manejar varias políticas para lidiar con condiciones cambiantes del ambiente). Aquí, también, P es un proceso que maneja la activación de acciones. Las *acciones*, precisamente, son parte de la instanciación que SCEL permite para que un sistema encaje en su paradigma. Identificar qué acciones son relevantes al sistema es una actividad fundamental del diseño de un sistema distribuido. SCEL tiene predefinido un conjunto básico de acciones relacionadas con el manejo de bases de conocimiento: **get**, **qry**, **put**, **fresh** y **new**. Cabe mencionar que una acción se describe completamente por medio de su ejecutor y del objetivo de la acción, con una posible referencia a otros CAs. Las políticas, en tanto, son utilizadas para reglamentar las acciones.

Ahora bien, la ejecución de las acciones no es trivial, al permitirse secuencialidad, concurrencia, compromiso (*commitment*) y posible monitoreo externo. Más información y definiciones formales extensas en [12]. En la siguiente sección mostramos los fundamentos de la programación en Erlang.

3. Erlang y sistemas distribuidos

Erlang y los sistemas distribuidos. Erlang es un lenguaje de programación de capacidad industrial orientado al tratamiento de problemas de sistemas distribuidos [3,5]. Erlang adopta un modelo de sistemas distribuidos basados en paso de mensajes sin recursos compartidos, con asincronía y concurrencia. Erlang concisamente puede ser descrito como sigue: *Erlang nodos* hospedan *procesos* (pensemos que a su vez los Erlang nodos están ejecutándose en una computadora determinada); tales procesos pueden comunicarse entre sí a través de *mensajes*. En principio, no existen canales de comunicación preestablecidos, y el flujo de información vía mensajes con remitentes es mediante el envío a *identificadores de procesos*, en tanto que del receptor se espera que el mensaje sea aceptado basándonos en casamiento de patrones (*pattern matching*) y en la identificación de los remitentes; de otra forma, los mensajes simplemente se ignoran. Los mensajes almacenan información heterogéna, desde bits hasta texto, o expresiones de tipo Erlang o inclusive procesos Erlang en sí.

Erlang y SCEL. En SCEL cada componente es diseñado siguiendo un enfoque de comunicación vía paso de mensajes, es equipado con una implementación de *conocimiento*, y rige sus acciones bajo algunas políticas de conducta. Con estas guías de diseño, el análisis de un sistema distribuido se facilita, haciendo incluso factible de forma inmediata la verificación por construcción de modelos (*model checking*) [11]. La programación en Erlang aquí mostrada fue realizada según el método incremental descrito informalmente en [3][p.148]. La idea básica de este tipo de programación es la creación de un proceso y el tratamiento gradual de los casos que se involucran en recibir y enviar mensajes. Siendo Erlang declarativo,

existen también posibilidades de modificar el código vía refactorización [14], transformación de programas [13], y evaluación parcial [10] (cuando se requieran de especializaciones de programas sobre computadoras con recursos limitados).

Para facilitar los diferentes dominios de posible aplicación, SCEL es *paramétrico* con respecto a algunas decisiones de posible implementación, tales como el lenguaje para expresar las políticas, los predicados regulando la interacción y agrupamiento (*ensembles*) entre componentes (con *coaliciones* cuando se realizan acciones con un objetivo en común), y la implementación de "conocimiento". SCEL no tiene un compromiso (por su alto nivel), en particular, con un lenguaje de programación específico. Queda a cargo del programador seleccionar las herramientas más adecuadas para el tratamiento de cada instancia. Por ejemplo, se podría utilizar un lenguaje tal como Python para programar las políticas, Prolog para el tratamiento del conocimiento, y Erlang para implementar la interacción distribuida entre componentes vía paso de mensajes. Más aún, Arduino podría proveer el acceso a los sensores y actuadores de los robots.

Aún más, como es descrito en [11], una componente podría tener algunas partes faltantes (o funcionalidades indicadas, pero no implementadas) y aún así ser completamente operacional como sistema. En escenarios donde los componentes podrían fallar y eventualmente quedar inoperantes, se podría delegar trabajo y funciones a otros componentes que cubrirían al componente faltante. En Erlang, esto es posible naturalmente vía el reemplazo de código en ejecución.

A continuación describimos por qué los programas en Erlang, basados en nociones de procesos infinitos en espera, son buenos candidatos para la formulación de la comunicación descrita en SCEL con *paso de mensajes*, *vectores de información*, y *satisfactibilidad de predicados*.

4. Erlang y los componentes autónomos de SCEL

De entre los lenguajes existentes actualmente, Erlang es un buen candidato para implementar el diseño teórico de un sistema y sus componentes siguiendo las directivas de SCEL. No obstante, debido al nicho de aplicación de otros lenguajes, no se excluye que exista partes de código escritas en su forma especializada para satisfacer requerimientos específicos. Para este fin, Erlang facilita la comunicación con otros lenguajes mediante filas de bits, y en particular, de utilizar, por ejemplo, Prolog, Prolog y Erlang interactuarían independiente en un solo nodo mediante un canal virtual construido mediante un puerto de comunicación serial (tal como ya hicimos en un experimento). Para el caso de comunicación remota la tecnología de *bluetooth* y *wi-fi* está suficientemente madura, y es económica, para la satisfacción de algunas necesidades de comunicación inalámbrica, aunque habría que valorar la distancia física cubierta contra el consumo de energía o la versatilidad en la estructuración y compartición de datos.

Cuando fuera necesario compartir recursos, se pueden programar agendas básicas ya sea locales o globales. En el caso en que SCEL enmarca el conjunto de políticas de uso de recursos, estas políticas están descritas mediante sistemas de transición etiquetados. La parte de depósitos de conocimiento (*knowledge*

repositories) podría tratarse con ETS y DETS (que son tecnologías de bases de datos pertenecientes al sistema OTP auspiciado por Erlang). En efecto, en OTP algunos módulos son específicos para el tratamiento intensivo de datos, tales como Amnesia. El tratamiento para estas bases de datos quedaría supervisada, nuevamente, por las políticas en boga dentro del marco teórico de SCEL.

A continuación describimos el conjunto de *abstracciones* que ligan a un nodo de Erlang (y sus procesos) con un componente autónomo descrito en SCEL.

1. Primero, describimos a nuestro componente autónomo como un *robot*: el robot está equipado con dispositivos para permitirle movilidad, relativa independencia energética, sensores, actuadores, y dispositivos de comunicación; bajo estas circunstancias, nuestro robot queda definido por la siguiente seudo-ecuación: robot = software (procesamiento, políticas, conductas, acciones, conocimiento) + hardware (sensores, actuadores)
2. Segundo, y específicamente, ahora consideramos que la parte esencial de procesamiento en software es llevado a cabo por un nodo de Erlang, con los procesos que sean necesarios poner en ejecución al robot. Para este fin, consideremos esencial que haya al menos un proceso inicial que bien puede apoyarse en automonitoreo o replicación para permitir la tolerancia a fallas; este proceso especial es nombrado `PSRobot`.
3. Adicionalmente al núcleo de programación en Erlang, es posible complementar las necesidades de programación con lenguajes específicos, tales como Prolog para el caso de la creación de planes, Arduino para las interfaces con los dispositivos sensoriales y actuadores, y Java si se requiere soporte vía bibliotecas para el manejo de comunicación mediante *wi-fi* o *bluetooth*.

La programación del núcleo de procesamiento del robot es por medio de la función `ac()`, y este núcleo es programado como ciclos eternos (*forever-loops*) para permanecer en un estado vigilante, y, en su caso y ya siendo parte de un proceso Erlang, activar las funciones asociadas al envío y recepción de mensajes. Los componentes autónomos pueden tener algunas propiedades auto-*; por ejemplo, el monitoreo propio del estado de energía, la auto-detección de fallas para el caso de auto-reparación (o al menos, para reportarse como inhabilitado), y la auto-preservación en ambientes riesgosos, adversos o peligrosos (por ejemplo, con un sensor de temperatura el robot se alejaría de una fuente de calor).

De las partes esenciales de estado de latencia de un robot, existe la construcción de Erlang `receive`, que trata diversos casos de recepción de mensajes mediante casamiento de patrones; este mecanismo es lo suficientemente versátil para permitir diversos tipos de entrada asociando *indicadores* (*tags*) a entradas numéricas, estructurando la información mediante vectores o bien diccionarios. Es importante en esta parte de programación tratar todos los casos exhaustivamente, con un uso bien planeado de las estructuras *atrapa-todo* (*catch-all*) que en ocasiones dificultan seriamente la depuración de programas en Erlang. En SCEL, el planteamiento teórico para una *interfaz* es mediante un conjunto de vectores, donde cada entrada señala un estado o bien un valor. La lectura de las entradas de este vector, como ya mencionamos, es manejada vía las políticas

en boga. Bajo una jerarquización armoniosa, es posible que un robot escriba también en la interfaz del vector de otro robot. Las entradas de los vectores también tienen otra función definida en SCEL: la de formación de grupos vía *publicaciones* (*broadcasting*). Esta formación de grupos mediante indicadores es una forma rápida y precisa de llevar a cabo planes mediante coaliciones: aquellos grupos que deberían trabajar en conjunto para lograr un objetivo. Por ejemplo, supongamos el diccionario: **{**activo:A, energía:E, tarea:T**}** señalando que el robot se halla activo (A puede ser activo o inactivo), con un nivel de energía E, y realizando en ese momento una tarea T, en donde la tarea *ocioso* (*idle*) se toma como no haciendo ninguna tarea. Supongamos que un robot requiere formar un grupo con aquellos elementos activos con un nivel de energía mayor del 50 % y estando ociosos. La petición se realiza mediante publicación de una solicitud y para cuando pasa un tiempo prudente los robots con las características solicitadas estarían listos para formar parte de un plan y llevarlo a cabo, con un objetivo común. El robot proactivo (quien convocó al grupo) puede decidir no llevar a cabo nada si, por ejemplo, no existen tantos robots como se esperaba. Dado que este mismo mecanismo de obtención de consenso podría llevarse a cabo concurrentemente, se puede optar por priorizar los objetivos y el mismo robot proactivo pasar a formar parte de un grupo.

Este mecanismo de interfaces es manejado por un conjunto de políticas. Dada una política, se estaría en posición de permitir que entre los robots se compartan información o no. El ejemplo a tratar sería el de un robot que esta *hackeado* e intenta confundir al sistema en su conjunto. Para esto, la política verificaría el número de accesos intentados de formación de grupos, un certificado de confianza por medio del robot proactivo, y un historial acerca de pautas de comportamiento para catalogar a un robot como estando en funcionamiento normal o anormal.

El diseño de una interfaz es particionada de acuerdo con un criterio basado en obtener información por cada uno de sus atributos de algunas posibles fuentes. En SCEL se utilizan predicados además de atributos para obtener grupos. Los predicados pueden a su vez formar fórmulas lógicas (o ecuaciones), y éstas deben de satisfacerse para el caso de formar grupos. Notemos también que la recepción de información no requiere necesariamente una respuesta como mensaje: tal información puede activar un actuador o, hasta en un caso extremo y por razones de seguridad, apagar al robot mismo.

Algunas acciones descritas directamente en SCEL están relacionadas con el envío y la recepción de mensajes. Esto es conveniente en términos de la contraparte de Erlang. Mediante tuplas (o diccionarios) y sus entradas, esta emisión y recepción de mensajes toma un papel preponderante. Mediante predicados, como fue ya comentado, las exigencias en la formación de grupos varía desde la comunicación entre pares (*peer-to-peer*) hasta la comunicación grupal mediante publicaciones (*broadcasting*) (orientada-a-grupo). Los filtros correspondientes se formulan mediante la inspección y verificación de los atributos así como la satisfacción de algunos predicados, para que la información llegue a un determinado número de robots. Si, aún más, existen permisos para explorar las bases de datos (conocimiento) de los robots, emergen complejas formas de interacción así como

objetivos de largo alcance que se verán a su vez reflejados en las secuencias de acciones que se planeen para lograr tales objetivos. En el trabajo en proceso que estamos realizando, por el momento, estamos enfocados más en el tratamiento reactivo de un robot, y hemos diferido el tratamiento de planes grupales (parte racional de un robot).

5. Nodos de Erlang y procesos en robots

En esta parte continuamos con una iteración de precisión de los anteriores conceptos por medio de "seudo-código" en Erlang. Afortunadamente, tal seudo-código está muy próximo a su implementación real. (Es necesario mencionar que "proceso" es una palabra que se utiliza bastante en Erlang pero también en los fundamentos teóricos de SCEL (cálculo de procesos).) Ahora analizamos el proceso `PSRobot`. El código mostrado en la Tabla 1 está destinado a casar muy cercanamente a las características de un robot en el escenario ejemplificado en [12], en hemos nombrado a una función `ac()` como la principal en el manejo de las funciones del robot. Notemos que hemos enfatizado el carácter *autónomo* en el diseño del robot, aunque un mensaje urgente de asistencia humana sería un buen último recurso.

Aparte de algunas funciones auxiliares definidas en otro lugar (y apoyadas quizás en otros lenguajes, tales como C, Python —para el caso de un micro-controlador Raspberry—, o Arduino). La pieza de código en Erlang debería casar bien en un robot en general a sus respectivas instancias de posibilidades, dispositivos, y actuadores, y de preferencia con un mínimo de asistencia humana.

Según la descripción de alto nivel de SCEL, los componentes autónomos deben ser lo suficientemente versátiles para permitir actividades de auto-configuración: desde jerarquías de diversa índole entre los componentes, agrupamientos de comunicación (*peer-to-peer*, comunicacion grupal), acuerdos consensados, enjambres (*swarms*), o la formación de coaliciones para cooperación o defensa. De manera importante, las conductas pueden redefinirse de acuerdo con los objetivos, y estos a su vez pueden cambiar, auto-adaptándose, de acuerdo con las circunstancias. A través de los atributos e interfaces, los componentes autónomos pueden alterar su configuración jerárquica (si existe) en aras de salvaguardar el objetivo principal del sistema como un todo. La decisión de si el mismo software estará en todos los robots es paramétrica de las posibilidades de que este software sea adaptable a diversas situaciones. No excluye la posibilidad, desde luego, de escribir programas que utilicen las capacidades especiales que un robot pudiera tener, aunque en nuestra simulación se está trabajando con robots que tienen exactamente los mismos componentes de hardware (con una suposición de homogeneidad en capacidades). En la parte restante de este trabajo queda por ver cómo Erlang brinda satisfacción concreta (con codificación asociada) a los requerimientos previos.

El programa mostrado en la Tabla 1 tiene una seudo-codificación en Erlang, con funcionalidades previsibles de acuerdo con el equipo de hardware que contamos, así como otras herramientas accesorias (varios procesos, microcontroladores

de Arduino y micro-computadoras Raspberrys). Desde este programa en seudo-código es posible obtener diversas instancias que permitirían una implementación efectiva de robots o inclusive de otras instancias de componentes autónomos (tales como agentes virtuales). En el caso de una simulación robótica, es posible crear incluso procesos que simulen el ambiente en donde los robots coexisten, así brindando un mayor realismo para mejor comprensión del sistema distribuido resultante.

Tabla 1. Listado de un programa en Erlang siguiendo directivas de SCEL.

```
1  ac() −> receive
2          {From, Qattributes} −> policy(From,attributes,Answer),
3                  if
4                          Answer==yes −> From ! attributes, ac();
5                          Answer==no −> From ! ‘‘Denied access’’, ac()
6                  end;
7          {From, Item} −> policy(From,Item,Answer),
8                  member(Item,attributes),
9                  if
10                         Answer==yes −> From ! yes, ac();
11                         Answer==no −> From ! ‘‘Denied access’’, ac()
12                 end;
13         {From, sensorsInput} −> policy(From, TypeofSencor,Answer),
14                 if
15                         Answer==yes −> From ! knowlege(sensors(type,input)), ac();
16                         Answer==no −> From ! ‘‘Denied access’’, ac()
17                 end;
18         {From, sensorsOutput} −> ....; %Idem;
19         {To, knowledge, acNew} −> knowledge(add(To,listTrust));
20         {To, knowledge,rmOld} −>
21                 knowledge(remove(To,listTrust)),
22                 knowledge(add(To,listDeny));
23         {gps,knowlege, gpsNew} −> knowlege(add(gps(Measures))); %Examples
24         {battery,knowlege, batteryLevelNew} −>
25                                 knowlege(add(batteryLevel(Measures)))
26         end.
27 policies (yes) −> listTrust ();
28 policies (no) −> listDeny().
29 knowledge(add,Items) −> add_list(Items);
30 knowledge(remove,Items) −> remove_list(Items);
31 knowledge(consult,Items) −> consult_list(Items).
32 attributes () −> list_of_attributes .
33 ensemble() −> .... %Some constraints to be satisfied to set up ensembles...
34 sensors(Input) −> obtain_values(Internal,Input);
35 sensors(Input) −> obtain_values(External,Input);
36 sensors(Output) −> send_value(Internal);
37 sensors(Output) −> send_value(External).
```

Cada CA tiene varias propiedades auto-*. Cada propiedad debería ser codificada como una función en Erlang, aunque con posible soporte de funciones externas para el manejo de periféricos, actuadores y otros dispositivos. De preferencia, existe una centralización del control del robot, aunque esto es preliminar en el diseño. Actualmente, por ejemplo, existen actuadores con sus propios procesadores (microcontroladores), y en este sentido, con un grado considerable de autonomía por sí mismos. Erlang tiene posibilidades de interactuar con los procesadores de estos actuadores debido a sus capacidades inherentes de concurrencia. Estamos investigando técnicas para que Erlang controle todos los procesadores disponibles mediante una programación de alto de nivel.

El componente de software aquí presentado es el resultado de varias iteraciones, como ya se ha mencionado, comenzando con las definiciones de SCEL, y siguiendo fielmente cada característica de un componente autónomo descrito por el formalisma. En cada iteración se ha llenado un "hueco" de precisión en código, haciendo eco de paradigmas de programación incremental y transformacional.

Nuestro escenario es una región planar con varios objetos diseminados para ser recolectados. Hay algunos obstáculos que sortear, y al menos dos sensores para este fin: uno ultrasónico y una cámara, los cuales son requeridos para la ubicación de los objetos. Los robots deambulan en la región de forma autónoma y aleatoria. Varias marcas son colocadas en el piso, y éstas son detectadas por los robots. Dado un grupo dedicado a la recolección (o al menos, detección) de estos objetos, la comunicación entre los elementos del grupo fluye rápidamente, para obtener consensos fortalecidos y evitar trabajo redundante.

6. Conclusiones

Obtenemos las siguientes conclusiones del trabajo presentado:

1. Seguimos la filosofía de desarrollo de programas de transitar tersamente de una especificación de alto nivel a una implementación orientada a un lenguaje de programación [13];
2. hemos instanciado uno de los parámetros del formalismo SCEL [12] en la parte de un lenguaje de programación seleccionado, ya que proponemos que Erlang, un lenguaje funcional con semántica bien identificada matemáticamente [6,15], sea el lenguaje de programación central en las aplicaciones distribuidas que un componente autónomo requiere;
3. aunque en etapa de diseño y con una propuesta todavía por perfeccionar, nuestro enfoque para la aplicación robótica es asequible yrealista, al utilizar tecnología de microprogramadores actuales así como la completa provisión de funciones de concurrencia preconstruidas en Erlang;
4. SCEL no tiene un compromiso explícito con un modelo computacional: aquí adoptamos el de concurrencia asíncrona con pasos de mensajes y sin recursos compartidos, modelo que es naturalmente dado en Erlang, con el conocimiento de que otros modelos computacionales podrían utilizarse también, cuando sea requerido.

En un trabajo a futuro, dotaremos con mayores capacidades sensoriales y de razonamiento a los CAs (tales como el razonamiento temporal [2,9]). Para ello, será necesario implementar algunos algoritmos de bases de datos y tratamiento de tiempo, sin menoscabo en los tiempos de reacción. La implementación en hardware tiene ya una factible realidad, pero es de notar que los CAs pueden tener otras aplicaciones a la aquí presentada; por ejemplo, pueden ser agentes que asistan de forma "personalizada" a estudiantes y profesores en un sistema de enseñanza/aprendizaje a distancia, como ya fue presentado, auxiliándose con varias otras herramientas, en [7], o bien en un sistema distribuido que gestione actividades colaborativas entre expertos.

Agradecimientos. Agradecemos las facilidades brindadas por la Universidad Tecnológica de la Mixteca para llevar a cabo la realización de este trabajo. El Dr. Manuel Hernández agradece al Dr. Andrés Fraguela Collar su aprecio y confianza a través de los años.

Referencias

1. Aceto, L., Ingólfsdóttir, A., Larsen, K.G., Srba, J.: Reactive systems. Cambridge (2007)
2. Aguilar-López, J.Y., Trujillo-Romero, F., Hernández, M.: Comunicación entre agentes inteligentes mediante cálculo de eventos usando Erlang. Research in Computing Science pp. 151–165 (2013)
3. Armstrong, J.: Programming Erlang: Software for a concurrent world. The pragmatic programmers (2007)
4. Bird, R., Wadler, P.: An introduction to Functional Programming. Prentice-Hall (1988)
5. Cesarini, F., Thompson, S.: Erlang programming. O'Reilly (2008)
6. Claessen, K., Svensson, H.: A semantics for distributed Erlang. In: ERLANG '05: Proc. of the 2005 ACM SIGPLAN workshop on Erlang. pp. 78–87. ACM (2005)
7. Cortés, H., García, M., Hernández, J., Hernández, M., Pérez-Cordoba, E., Ramos, E.: Development of a distributed system applied to teaching and learning. In: ERLANG '09: Proc. of the 8th ACM SIGPLAN workshop on ERLANG. pp. 41–50. ACM (2009)
8. Hennessy, M.: A distributed Pi-Calculus. Cambridge University Press (2007)
9. Hernández, M.: Event Calculus for Reasoning about Erlang Systems. 2013 12th Mexican International Conference on Artificial Intelligence 0, 29–35 (2011)
10. Jones, N.D.: An introduction to partial evaluation. Association for Computer Machinery (ACM) Computing Surveys 28(3), 480–503 (September 1996)
11. Nicola, R.d., Lluch-Lafuente, A., Loreti, M., Morichetta, A., Pugliese, R., Senni, V., Tiezzi, F.: Programming and verifying component ensembles (8415) (2014)
12. Nicola, R.D., Loreti, M., Pugliese, R., Tiezzi, F.: A formal approach to autonomic systems programming: The SCEL language. ACM Trans. Auton. Adapt. Syst. 9(2), 7:1–7:29 (Jul 2014), http://doi.acm.org/10.1145/2619998
13. Partsch, H.: Specification and Transformation of Programs. Texts and Monographs in Computer Science, Springer-Verlag (1990)

14. Sagonas, K., Avgerinos, T.: Automatic refactoring of Erlang programs. In: PPDP '09: Proc. of the 11th ACM SIGPLAN Conf. on Principles and practice of declarative programming. pp. 13–24. ACM (2009)
15. Svensson, H., Fredlund, L.Å.: A more accurate semantics for distributed Erlang. In: ERLANG '07: Proc. of the 2007 SIGPLAN workshop on ERLANG Workshop. pp. 43–54. ACM (2007)
16. Varela, C., Abalde, C., Castro, L., Gulías, J.: On modelling agent systems with Erlang. In: Proceedings of the 2004 ACM SIGPLAN workshop on Erlang. pp. 65–70. ERLANG '04, ACM, New York, NY, USA (2004), http://doi.acm.org/10.1145/1022471.1022481

Poblado automático de ontologías de perfiles académicos a partir de textos en español

José A. Reyes-Ortiz, Maricela Bravo, Oscar Herrera y Alejandro Gudiño

Universidad Autónoma Metropolitana, Unidad Azcapotzalco,
Departamento de Sistemas, Distrito Federal,
México

{jaro, mcbc, oha}@correo.azc.uam.mx, al209202106@alumnos.azc.uam.mx

Resumen. Este artículo propone un enfoque para el poblado automático de ontologías de perfiles académicos a partir de los textos, expedientes curriculares y resúmenes, de publicaciones científicas en español. El enfoque utiliza reglas semánticas y marcadores lingüísticos para extraer los individuos de clase, relaciones y valores de propiedad. Una evaluación ha sido realizada con un conjunto de individuos *gold standard*, en términos de precisión y exhaustividad para el poblado de elementos ontológicos.

Palabras clave: poblado automático de ontologías, reglas semánticas, marcadores lingüísticos, procesamiento de lenguaje natural (PLN).

1. Introducción

En los últimos años, ha surgido la necesidad de procesar la información de manera automática debido al crecimiento acelerado de la información electrónica disponible en Internet, empresas, organizaciones y repositorios en general. Para lograr este procesamiento ha sido necesario representar la información de tal manera que sea procesable con computadoras. Esta representación de conocimiento puede realizarse con ontologías, las cuales en los últimos años han ganado importancia [1].

Según [2], las ontologías poseen características significativas que las posicionan en una de las formas de representación más utilizadas inicialmente en la Web, y ahora en diversos medios electrónicos. Sus componentes básicos son: *conceptos*, *relaciones* (*propiedades de objeto* y *propiedades de dato*), *funciones*, *individuos de clase*, *relaciones entre individuos, valores de las propiedades* y *axiomas*. La construcción de ontologías puede realizarse de manera manual pero esto ocasiona diversos problemas de costo y tiempo. Como una alternativa a esto surge el aprendizaje automático de ontologías a partir de textos cuyo objetivo es identificar los elementos ontológicos de manera automática o semiautomática [3].

En el contexto del aprendizaje de elementos ontológicos, el poblado de ontologías hace referencia a extraer y representar, de manera automática o semiautomática, los *individuos de clase*, *relaciones entre los individuos* y *valores de las propiedades.* En la

literatura, existen enfoques ligeros para el poblado de ontologías basados en modelos estadísticos-sintácticos ([4, 5, 6, 7]) y enfoques semánticos basados en reglas ([8, 9, 10]), estos no satisfacen el poblado ya que desatienden el conocimiento lingüístico profundo de los textos.

Por lo tanto, en este artículo nos centramos la *extracción de individuos,* sus *relaciones* y los valores de las *propiedades de datos* para la representación semántica de publicaciones científicas a partir de textos de artículos en español. El enfoque propuesto se basa en reglas semánticas y marcadores lingüísticos que capturan la *información semántica* de los textos. La idea de proponer un enfoque para el español se debe, en gran medida, a la carencia de enfoques, herramientas o métodos de poblado de ontologías para este idioma.

El resto del artículo está organizado de la siguiente manera. En la Sección 2 se presenta las bases sobre el poblado de ontologías a partir de textos y el conjunto de tareas que intervienen. La Sección 3 expone un acercamiento al estado de arte de los trabajos relacionados al poblado de ontologías, particularmente, de dominio de los perfiles académicos o publicaciones científicas. El enfoque propuesto en este trabajo se muestra en la Sección 4, el cual consta de tres etapas: el etiquetado de los textos, la extracción de individuos basadas en reglas semánticas y el poblado de individuos en las ontologías. La Sección 5 presenta la experimentación y los resultados obtenidos en el enfoque propuesto. Finalmente, las conclusiones y trabajos futuros son presentados en la Sección 6.

2. Poblado de ontologías a partir de textos

Las ontologías han ganado gran popularidad en la comunidad científica al convertirse en una alternativa viable para la representación de conocimiento [1]. Este conocimiento se puede encontrar en los textos y necesita ser representado en una estructura que sea procesable por computadoras con la finalidad de lograr su manejo automático. Por ello, se plantea construir ontologías a partir de textos de manera cuidadosa para expresar con claridad y precisión el conocimiento. Éste es un proceso crítico, el cual puede realizarse de manera manual pero representa una tarea tediosa, costosa y que consume mucho tiempo [1].

En efecto, el proceso de construcción de ontologías se puede realizar de manera semiautomática o automática con *aprendizaje de ontologías*, tarea que consiste en construirlas a partir de un conjunto de datos iniciales, las cuales pueden ser un *corpus* de textos en lenguaje natural [1]. La idea consiste en partir de un conjunto de textos y relacionar los segmentos del texto (categorías gramaticales) con entidades ontológicas. Esta relación puede ser rígida y poco flexible; sin embargo, existen las excepciones, en las cuales intervienen técnicas de Procesamiento de Lenguaje Natural (PLN) para lograr una buena representación de los textos mediante ontologías.

El aprendizaje de ontologías incluye en una serie de tareas ordenadas [3]:

1. Adquisición de terminología relevante.
2. Adquisición de sinónimos.
3. Formación de conceptos.
4. Organización jerárquica de los elementos.

5. Aprendizaje de relaciones, propiedades, atributos, junto con su respectivo rango y dominio.
6. Organización jerárquica de las relaciones.
7. Instanciación de los axiomas del esquema.
8. Definición de los axiomas arbitrarios.

Este trabajo se centra en el *poblado de ontologías,* el cual corresponde a la tarea que [3] define como instanciación de los axiomas. El *poblado de ontologías* hace referencia al descubrimiento, a partir de los textos, de individuos de clases, relaciones entre los individuos y su representación en el modelo ontológico. Diversos trabajos han sido propuestos en este ámbito [11], [5] y [12] por ello se ha dedicado la Sección 3 de este trabajo a revisar el estado del arte relacionado al poblado de ontologías, particularmente del dominio académico y de perfiles profesionales.

3. Trabajos relacionados

El poblado de ontologías a partir de textos en forma automática o semiautomática es un área que ha sido considera por la comunidad de investigadores, un reto actual para el *Procesamiento de Lenguaje Natural* [3]. Por ello, en la literatura existe una amplia diversidad de enfoques sobre el poblado de ontologías a partir de textos y, particularmente, considerando ontologías de perfiles académicos.

En el aspecto del poblado de ontologías de diversos dominios, se han identificado características valiosas que han servido para organizar los enfoques, tales como: el idioma de los textos de entrada, poblados de elementos ontológicos, el tipo de enfoque de PLN utilizado para extraer el conocimiento y el nivel de análisis aplicado a los textos.

El idioma de los textos de entrada es una característica que define, entre otras cosas, el tipo de enfoque a aplicar y el nivel de análisis. Esto debido a la existencia de una diferencia de complejidad en las estructuras sintácticas y semánticas de los idiomas. De esta manera, existen enfoques que consideran al inglés como su dominio de los documentos de entrada como en [4, 6, 7, 8, 9, 10, 13, 14, 15, 16 y 17], el idioma francés considerado en los trabajos presentados en [18], enfoques que han sido propuestos para el español [19, 20, 21] y enfoques que son independientes del idioma en sus textos de entrada como en [22, 23].

Los elementos ontológicos que son poblados, a partir de textos, por los diversos enfoques son: *instanciación de clases*, *descubrir* y *representar individuos* [5, 6, 11, 20 y 23]; poblado de propiedades de objetos como en [4, 24]; poblado de valores de propiedades de dato [4, 8, 11]; finalmente, el poblado de relaciones entre individuos [4, 7, 9, 10, 11, 21].

El tipo de enfoque y nivel de análisis de PLN aplicado a los textos de entrada con la finalidad de extraer el conocimiento para el poblado de las ontologías depende del idioma del dominio. En este ámbito, se han identificado trabajos que utilizan un análisis ligero de los textos con un enfoque estadístico como en [4]; enfoques basados en dependencias sintáctica con un análisis superficial han sido propuestos en [5, 6, 7];

semántico o heurístico basado en reglas [8, 9, 10, 11, 20, 21, 23] y un enfoque lingüístico con un análisis ligero de los textos ha sido presentado en [24].

Por otro lado, el poblado automático de ontologías a partir de textos, en el dominio de perfiles profesionales o académicos, ha sido propuesto por trabajos como en [25] que presenta un marco de trabajo para la extensión semántica de documentos PDF escritos en LATEX, mediante la creación de metadatos para publicaciones científicas. Este marco de trabajo permite a los autores la creación de metadatos mientras se desarrolla el proceso de escritura de documentos; el trabajo presentado en [26] muestra un métodos para la ingeniería ontológica y poblado de ontologías a partir de textos, el cual fue probado con textos sobre experiencias académicas y de proyectos de investigación; el sistema descrito en [27] utiliza técnicas de extracción de información y aprendizaje automático para el reconocimiento y clasificación de eventos a partir de artículos de noticias electrónicas, las cuales describen la vida académica en el área de conocimiento multimedia, en particular, eventos como la adjudicación de proyectos, investigaciones y visitas; el trabajo expuesto en [28] ha presentado una herramienta para el poblado automatizado de ontologías a partir de textos utilizando extractores, es decir patrones, escritos por humanos con la finalidad de capturar el conocimiento de los textos. La herramienta ha sido utilizada en el dominio académico, donde se han poblado, entre otras clases, *Publicaciones*, *Revistas*, *Conferencias* y *Libros*, además de las relaciones *responsabilidad_de*, *publicado_en* y *autor_listado_en*; finalmente, el trabajo presentado en [29] propone una descripción ontológica del dominio de las ciencias computacionales y la aplicación de un enfoque de aprendizaje supervisado para la extracción de individuos a partir de artículos anotados manualmente.

El poblado automático de ontologías a partir de textos exige la intervención de técnicas de procesamiento de lenguaje natural. Además hemos identificado una desatención en la literatura para proponer enfoques de poblado de ontologías para el dominio académico en español. Por lo tanto, en este artículo se considera el idioma español como dominio de los textos de entrada, reglas semánticas basadas en marcadores lingüísticos, mediante un análisis profundo de los textos, con la finalidad de lograr el poblado automático de ontologías de perfiles académicos. El enfoque propuesto considera el poblado de individuos de clase, propiedades de objeto, propiedades de dato y relaciones entre los individuos.

4. Enfoque propuesto

El enfoque que se propone en este artículo para el poblado de ontologías de perfiles académicos, a partir del texto de los resúmenes, se centra en descubrir individuos de publicaciones y sus propiedades relacionadas con investigadores. A partir de los expedientes curriculares, se descubren las *propiedades de dato* de las publicaciones como título, autores listados, editor y año de la publicación. El contenido del artículo, particularmente el resumen, también es procesado para obtener información detallada de la publicación científica, tal como: solución aportada, evidencia de la solución y propósito del método.

El enfoque propuesto se compone de tres etapas: 1. Etiquetado de los textos, 2. Extracción de elementos ontológicos (*individuos*) basada en reglas semánticas y 3. El poblado de las ontologías.

4.1. Etiquetado de los textos

En la primera etapa, el texto de los expedientes curriculares y los resúmenes de las publicaciones científicas se divide en oraciones, y después en palabras para su etiquetado morfológico. Las tareas estándar de la segmentación de las palabras y el etiquetado morfológico se realizan con la interfaz de programación de aplicaciones proporcionada en el marco de desarrollo para el procesamiento de lenguaje natural llamado *GATE* [30].

El marco de desarrollo *GATE* proporciona los componentes necesarios para realizar la segmentación del texto en oraciones (*Sentence_Splitter*) y palabras (*Tokenizer*), además del etiquetado morfológico (*POS Tagger*). Estos componentes son fácilmente ensamblados para lograr una aplicación más compleja basada en tuberías, donde se agrega el componente de extracción de información.

El etiquetado de partes de la oración se encarga de asignar una categoría gramatical a cada palabra. Para esta tarea se recurre al componente llamado *POS Tagger* en el cual, para el caso de textos en español, utiliza el etiquetador *Spanish TreeTagger* [31].

4.2. Extracción de información

La segunda etapa consiste en realizar la extracción de individuos de las publicaciones, sus relaciones y sus propiedades de dato a partir de los textos etiquetados.

Para esta tarea de extracción de información, se utiliza el componente de *GATE* llamado *Jape Transducer*, el cual se encarga de compilar y ejecutar un conjunto de reglas basadas en la gramática *JAPE (Java Annotation Pattern Engine)* [32]. Un conjunto de *27* reglas semánticas fueron construidas y codificadas en la gramática *JAPE* con la finalidad de extraer los individuos, sus *propiedades de objeto* y sus *propiedades de dato*.

```
Rule: Propósito
(
     ({SpaceToken.kind == space}
            ({Token.string == "para"})
        {SpaceToken.kind == space}
     )
     (({Token.pos == VLinf})
       (NOUNPHRASE)
     ):prop
)--> :prop.Propósito = {rule = "Propósito", text =:prop@string}
```

Fig. 1. Regla en JAPE para la extracción de la relación *tiene_propósito.*

Los individuos de las publicaciones se encuentran relacionados mediante las siguientes propiedades de objeto: *publicada_en*, *es_listado_en* y *aporta_solución*. Además, se crearon reglas semánticas basadas en marcadores lingüísticos para la extracción de los valores de las siguientes propiedades de datos: *tiene_título*, *tiene_año_publicación, tiene_propósito*, *tiene_evidencia*. En la Figura 1 se muestra una regla semántica, codificada en la gramática *JAPE,* creada para la extracción del propósito de una publicación a partir del resumen (texto) del artículo científico.

Los individuos relacionados, mediante las propiedades de objeto llamadas *aporta_solución*, *tiene_propósito* y *tiene_evidencia*, se extraen mediante reglas semánticas basadas en marcadores lingüísticos. Estos marcadores ayudan a señalar la presencia de una relación entre segmentos del texto, en nuestro caso, apoyan la señalización de relaciones entre individuos y valores de propiedades de dato. En la Tabla 1 se muestra un conjunto marcadores lingüísticos para estas relaciones y propiedades.

Tabla 1. Marcadores lingüísticos para la extracción de relaciones.

Relación	Marcadores lingüísticos
Solución	*en este artículo, nuestra aportación*
Propósito	*para + verbo infinitivo, con la finalidad de, con el fin de*
Evidencia	*ya que, claramente*

Los individuos sobre las publicaciones, sus relaciones y las propiedades que fueron extraídas a partir de los textos se representan en un modelo ontológico, como individuos, el cual se describe en la sección 4.3.

4.3. Poblado ontológico

En la tercera etapa se diseñó un modelo ontológico para representar la información extraída de los expedientes curriculares y resúmenes de publicaciones como *individuos de clase*, junto con sus propiedades de objeto y datos.

El modelo está codificado en el Lenguaje Ontológico para la Web (OWL 2)[1] y se encuentra constituido por las siguientes clases: *Publicación*, la cual representa la descripción semántica de un artículo científico con sus relaciones correspondientes; la clase *Método* que describe semánticamente la forma de solución del problema planteado en la publicación; la clase *Autor* que representa la persona o lista de personas que colaboran en la publicación; la clase *Editor* que se utiliza para relacionar la revista, memoria o actas donde se publica el trabajo científico. En la Figura 2 se muestra este modelo ontológico con sus clases, propiedades de objeto y datos.

[1] http://www.w3.org/TR/owl2-overview/

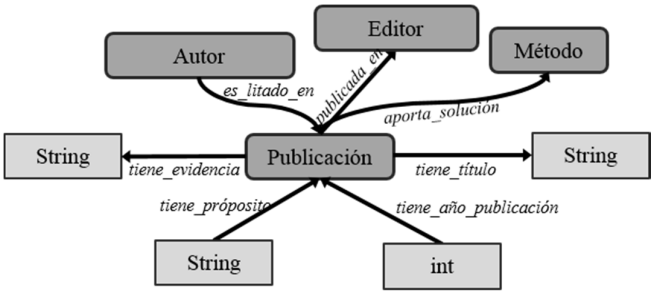


Fig. 2. Esquema ontológico para la representación semántica de publicaciones científicas.

El modelo ontológico también considera relaciones de tipo de dato, las cuales se utilizan para asignar valores a las propiedades de un individuo. De esta manera, en el modelo ontológico propuesto (Figura 2), también se consideran las siguientes propiedades: *tiene_título* y *tiene_año_publicación* que contienen el título y año de la publicación con sus valores respectivos; la relación de tipo de dato *tiene_propósito* que se encarga de relacionar la intención que existe detrás del método propuesto con las publicaciones científicas; finalmente la relación *tiene_evidencia* que contienen información destinada a aumentar la confianza del lector sobre el método y el propósito de la publicación.

La axiomatización del modelo ontológico se origina en la clase *Publicación*, en la cual los individuos extraídos son representados junto con las propiedades de objeto y datos. De esta manera, el modelo está compuesto por axiomas de clase y de relaciones, la Figura 3 muestra la instanciación de una publicación (*hasIndividual*) titulada: POBLADO_AUTOMÁTICO_DE_ONTOLOGÍAS_DE_PERFILES_ACADÉMICOS_A_PARTIR_DE_TEXTOS_EN_ESPAÑOL, la cual tiene una relación mediante la propiedad de objeto *aporta_solución* con el método REGLAS_SEMÁNTICAS-MARCADORES_LINGÜÍSTICOS y tiene como propósito al texto EXTRAER-REPRESENTAR_LAS_INSTANCIAS_EN_UN_MODELO_ONTOLÓGICO_DE_PUBLICACIONES_CIENTÍFICAS.

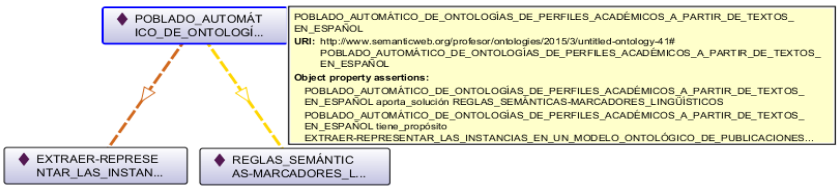


Fig. 3. Axiomatización de un individuo en la case *Publicación*.

En la Sección 5 se muestra la experimentación y los resultados obtenidos de la aplicación del enfoque con sus tres etapas, cuyo objetivo final es el poblado automático de ontologías a partir de textos.

5. Evaluación y resultados

La evaluación del enfoque propuesto fue realizada contra un conjunto *de individuos de clase*, *relaciones entre individuos* (propiedades de objeto) y valores de las *propiedades de dato*, las cuales fueron identificadas y representadas por expertos en el dominio científico-académico. Este conjunto incluye la identificación y representación manual de *568* individuos, los cuales se encuentran distribuidos de la siguiente manera: *337* individuos relacionados con la publicación y *231* individuos que están relacionados con el contenido del resumen de los artículos.

El procedimiento de evaluación consiste en comparar los elementos extraídos de manera automática con los individuos identificados y validados por los expertos. Las métricas de evaluación *precisión* y *exhaustividad* [33] fueron utilizadas con la finalidad de cuantificar los individuos identificados correctamente por el enfoque propuesto. En este contexto, se adaptan y definen las siguientes medidas: la *precisión* (*P*) que es el coeficiente entre el número de individuos relevantes identificados y el total de individuos extraídos, como se muestra en (1); y la *exhaustividad* (*E*) que es el coeficiente entre el número de individuos relevantes identificados y el número de individuos que deben ser extraídos, la cual se muestra en (2).

$$P = \frac{\#\ Individuos\ relevantes\ poblados}{\#\ Total\ de\ individuos\ poblados} \tag{1}$$

$$E = \frac{\#\ Individuos\ relevantes\ poblados}{\#\ Individuos\ que\ deberían\ ser\ poblados} \tag{2}$$

Una media armónica que combina los valores de *precisión* y *exhaustividad* llamada *medida F*, la cual se muestra en (3).

$$medida\ F = \frac{2*P*E}{P+E} \tag{3}$$

La Tabla 2 muestra los resultados del enfoque propuesto en términos de precisión, exhaustividad y medida F para la tarea de poblado automático de los 568 individuos de clase y relaciones (propiedad de datos y relaciones entre individuos).

Tabla 2. Resultados del poblado de la ontología de publicaciones científicas con instancias de clase y relaciones entre individuos.

Tipo de individuo	Precisión	Exhaustividad	Medida F
Publicación	0.894	0.931	0.912
Resumen	0.794	0.887	0.837
Promedio	0.844	0.909	0.875

La Tabla 3 presenta los resultados de los enfoques y propuestas presentadas en [4, 20, 24, 27, 34]. Ellos presentan propuestas para el descubrimiento y representación de instancias de clases, propiedades y relaciones entre las instancias. Los resultados están

expuestos en términos de un promedio de la medida F de los tipos de axiomas ontológicos que cada uno descubre.

Tabla 3. Resultados del poblado ontológico con individuos de clase, propiedades y relaciones entre individuos.

	Medida F
[4]	0.833
[20]	0.873
[24]	0.818
[27]	0.589
[33]	0.765
Nuestra propuesta	0.875

Los resultados demuestran una efectividad de nuestro enfoque propuesto, el cual logra una ligera ventaja de 0.002 sobre el trabajo presentado en [20]. Cabe mencionar que nuestro enfoque utiliza texto libre para la extracción de relaciones entre individuos a partir de los resúmenes de artículos científicos. Sin embargo, aun cuando nuestro enfoque se basa en textos semiestructurados, sólo para la extracción de propiedades de las publicaciones, el método utilizado puede ayudar a expertos del dominio a representar información sobre publicaciones científicas.

Los resultados del poblado de ontología de publicaciones científicas han demostrado que las reglas somáticas logran extraer información concerniente a las publicaciones como título, editor donde se publica, la lista de autores listados y el año de publicación. Mientras tanto, los marcadores lingüísticos son capaces de señalizar la información de solución, propósito y evidencia a partir de los resúmenes.

6. Conclusiones

En este artículo se ha presentado un enfoque basado en reglas semánticas y marcadores lingüísticos, con la finalidad de identificar y representar individuos de publicaciones científicas a partir de expedientes curriculares y resúmenes de artículos científicos en español.

El enfoque propuesto ha sido evaluado por un conjunto *gold standard* de *568* individuos que han sido identificados y representados por expertos. Los resultados han mostrado un mejor desempeño en el poblado de información propia de la publicación al poblar el *91.2%* de los individuos. Sin embargo, en el poblado de individuos extraídos a partir del resumen de la publicación se logra un desempeño promedio del *83.7%*. El bajo desempeño logrado en la precisión, apenas un *79.4%*, del poblado de individuos que provienen del resumen se debe a problemas del lenguaje natural como la polisemia. Este fenómeno del lenguaje está presente en los marcadores lingüísticos utilizados para la extracción, por ello un tratamiento de este fenómeno es necesario y deja una oportunidad para un trabajo futuro.

Los resultados de la evaluación del enfoque propuesto son prometedores y mejores (por encima del promedio) que los obtenidos por los trabajos relacionados del estado del arte. Todos ellos enfocados en la extracción y representación (poblado) de individuos de clases, propiedades y relaciones en un modelo ontológico.

Las principales contribuciones de este trabajo son (1) un modelo de representación semántica de publicaciones científicas, (2) un conjunto de reglas basadas en marcadores lingüísticos para extracción de información sobre publicaciones científicas y (3) un enfoque de poblado automático de ontologías del dominio académico para el idioma español.

La información extraída y representada en el modelo ontológico resulta de gran utilidad para tareas como la recuperación de información en el dominio académico sobre autores, año y editor, aportación, evidencia y propósito de las publicaciones científicas. Como trabajo futuro, se puede utilizar esta información por un sistema de pregunta-respuesta para conceder información precisa a las peticiones de los usuarios.

Agradecimientos. Este trabajo ha sido financiado por el Programa para el Desarrollo Profesional Docente (PRODEP) con el número de proyecto UAM-A-PTC-34. Agradecemos el apoyo otorgado por la Universidad Autónoma Metropolitana Unidad Azcapotzalco y el SNI-CONACyT.

Referencias

1. Maedche, A.: Ontology learning for the semantic web. Springer Science & Business Media, Massachusetts, USA (2002)
2. Gruber, T. R.: Toward principles for the design of ontologies used for knowledge sharing?. International journal of human-computer studies, vol. 43(5), pp. 907–928 (1995)
3. Cimiano, P.: Ontology Learning and Population from Text: Algorithms, Evaluation and Applications. Springer, New York, USA (2006)
4. Faria, C., Serra, I., Girardi, R.: A domain-independent process for automatic ontology population from text. Science of Computer Programming, vol. 95, pp. 26–43 (2014)
5. Oliveira, H., Lima, R., Gomes, J., Ferreira, R., Freitas, F., Costa E.: A confidence–weighted metric for unsupervised ontology population from web texts. Database and Expert Systems Applications, pp. 176–190 (2012)
6. Shen, W., Wang, J., Lou, P., Wang, M.: A graph-based approach for ontology population with named entities. In: Proceedings of the 21st ACM International Conference on Information and Knowledge Management, pp. 345–354 (2012)
7. Buitelaar, P., Eigner, T.: Topic Extraction from Scientific Literature for Competency Management. In: The 7th International Semantic Web Conference, Karlsruhe, Germany, (2008)
8. Cui, G., Lu, Q., Li, W., Chen, Y.: Automatic Acquisition of Attributes for Ontology Construction. Proceedings of the 22nd International Conference on Computer Processing of Oriental Languages, Language Technology for the Knowledge-based Economy, LNAI, vol. 5459, pp. 248–259 (2009)
9. Blomqvist, E.: Ontocase - A pattern-based ontology construction approach. In: Proccedings of On The Move to meaningful internet systems, ODBASE - The 6th International

Conference on Ontologies, DataBases, and Applications of Semantics, Trento, Italy, pp. 971–988 (2007)
10. Dahab, M. Y., Hassan, H. A., Rafea, A.: TextOntoEx: Automatic Ontology Construction from Natural English Text. Expert System with Applications: An International Journal, vol. 34(2), pp. 1474–1480 (2008)
11. Draicchio, F., Gangemi, A., Presutti, V., Nuzzolese, A. G.: Fred: From natural language text to RDF and owl in one click. In: The Semantic Web: ESWC 2013 Satellite Events Springer Berlin Heidelberg, pp. 263–267 (2013)
12. Wong, W.: Discovering Lightweight Ontologies using the Web. In Proceedings of the 9th Postgraduate Electrical Engineering & Computing Symposium, Perth, Australia (2008)
13. Fortuna, B., Lavrac, N., Velardi, P.: Advancing Topic Ontology through Term Extraction. In: Proceedings of the 10th Pacific Rim International Conference on Artificial Intelligence, Hanoi, Vietnam, pp. 626–635 (2008)
14. Cerbah, F.: Mining the Content of Relational Databases to Learn Ontologies with Deeper Taxonomies. In: IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, pp. 553–557 (2008)
15. Celjuska, D., Vargas-Vera, M.: Ontosophie: A Semi-Automatic System for Ontology Population from Text. In: Proceedings of the 3rd International Conference on Natural Language Processing (2004)
16. Waard, A., Buitelaar, P., Eigner, T.: Identifying the Epistemic Values of Discourse Segments in Biology Texts. International Workshop on Computational Semantics, Eindhoven, Netherlands, pp. 351–354 (2009)
17. Morante, R., Van-Asch, V., Daelemans, W.: A memory-based learning approach to event extraction in biomedical texts. In: Proceedings of the BioNLP 2009 Workshop Companion Volume for Shared Task, pp. 59–67 (2009)
18. Aussenac-Gilles, N., Despres, S., Szulman, S.: The TERMINAE method and platform for ontology engineering from texts. In Bridging the Gap between Text and Knowledge-Selected Contributions to Ontology Learning and Population from Text, pp. 199–223 (2008)
19. Sánchez, S., Llorens, J., Morato, J., Hurtado, J.: Extracción Automática de Relaciones Semánticas. En: Memorias de la 2da. Conferencia Iberoamericana en Sistemas, Cibernética e Informática, pp. 265–268 (2005)
20. Ruíz-Martínez, J.M.: Ontology Population: An Application For The E-Tourism Domain. International Journal of Innovative Computing, Information and Control, vol. 7 (11), pp. 6115-6134 (2011)
21. Lacasta, J., Lopez-Pellicer, F. J., Florczyk, A., Zarazaga-Soria, F. J., Nogueras-Iso, J.: Population of a spatio-temporal knowledge base for jurisdictional domains. International Journal of Geographical Information Science, vol. 28 (9), pp. 1964–1987 (2014)
22. Valencia, R.: Un Entorno para la Extracción Incremental de Conocimiento desde Texto en Lenguaje Natural. Tesis doctoral de la Universidad de Murcia, España (2005)
23. Llorens, H., Navarro, B., Saquete, E.: Detección de Expresiones Temporales TimeML en Catalán mediante Roles Semánticos y Redes Semánticas. Revista de Procesamiento del Lenguaje Natural, vol. 43, pp. 13–21 (2009)
24. Faria, C., Girardi, R. Novais, P.: Using domain specific generated rules for automatic ontology population. In: The International Conference on Intelligent Systems Design and Applications, Koshi, India, pp. 297–302 (2012)
25. Groza, T., Handschuh, S., Möller, K., Decker, S.: SALT-Semantically Annotated LaTeX for Scientific Publications. In: The Semantic Web: Research and Applications, pp. 518–532 (2007)

26. Aussenac-Gilles N., Jacques M. P.: Designing and Evaluating Patterns for Relation Acquisition from Texts with Caméléon. Terminology, Pattern-Based approaches to Semantic Relations, vol. 14 (1), pp. 45–73 (2008)
27. Vargas-Vera, M., Celjuska, D.: Event recognition on news stories and semi-automatic population of an ontology. In: Proceedings of the 2004 IEEE/WIC/ACM International Conference on Web Intelligence, pp. 615–618 (2004)
28. Aleman-Meza, B., Halaschek, C., Sheth, A. P., Arpinar, I. B., Sannapareddy, G.: SWETO: Large-scale semantic web test-bed. In: 16th International Conference on Software Engineering and Knowledge Engineering, Banff, Canada (2004)
29. Kröll, M., Klampfl, S., Kern, R.: Towards a Marketplace for the Scientific Community: Accessing Knowledge from the Computer Science Domain. D-Lib Magazine, vol. 20(11), pp. 10 (2014)
30. Cunningham, H., Maynard, D., Bontcheva, K.: Text processing with GATE. Gateway Press, California, USA (2011)
31. Schmid, H.: Probabilistic part-of-speech tagging using decision trees. In: Proceedings of the international conference on new methods in language processing, vol. 12, pp. 44–49 (1994)
32. Cunningham, H., Maynard, D., Tablan, V.: JAPE: a Java Annotation Patterns Engine (Second Edition). Technical Report, University of Sheffield, Department of Computer Science, United Kingdom (2000)
33. Baeza-Yates, R., Ribeiro-Neto, B.: Modern information retrieval, Association for Computing Machinery press, New York, USA (1999)
34. Navigli, R., Velardi, P.: From glossaries to ontologies: Extracting semantic structure from textual definitions. Ontology Learning and Population: Bridging the Gap between Text and Knowledge, pp. 71–87 (2008)

Reviewing Committee

Ricardo Acevedo
Moisés Alencastre Miranda
Roberto Alonso Rodriguez
Joanna Alvarado-Uribe
Gustavo Arroyo Figueroa
Christian Arzate
Ivonne Maricela Ávila Mora
Jorge Bautista López
Ivo Buzon
Maria Victoria Carreras
Felix Castro Espinoza
Noé Alejandro Castro Sánchez
Bárbara Cervantes
Jair Cervantes
Efren Chavez Ochoa
Gustavo Delgado Reyes
Sofia N. Galicia-Haro
Natalia Garcia
Alexander Gelbukh
David Gonzalez
Miguel Gonzalez-Mendoza
Hugo Gustavo González
Hernández Mario Graff
Fernando Gudiño
Pedro Guevara López
Yasmin Hernandez
Neil Hernandez Gress
Oscar Herrera
Rodolfo Ibarra
Yulia Ledeneva
Asdrubal Lopez Chau
Juan Carlos Lopez Pimentel
Ivan Adrian Lopez Sanchez
Antonio Marin-Hernandez
Lourdes Martínez
Martinez Medina
Miguel Angel Medina Perez
Efrén Mezura-Montes
Sabino Miranda-Jiménez
Daniela Moctezuma
Raul Monroy
Jaime Mora-Vargas
Saturnino Job Morales Escobar
Lourdes Muñoz Gómez
Antonio Neme
Alberto Oliart Ros
Mauricio Osorio
Elvia Palacios
Hiram Eredin Ponce Espinosa
Carlos Pérez Leguizamo
Maricela Quintana López
Carlos A. Reyes-Garcia
Carlos Alberto Rojas Hernández
Rafael Rojas Hernández
Dafne Rosso
Oscar S. Siordia
Grigori Sidorov
Abraham Sánchez López
Israel Tabarez Paz
Eric Sadit Tellez
Nestor Velasco-Bermeo
Elio Villaseñor
Alisa Zhila

Impreso en los Talleres Gráficos
de la Dirección de Publicaciones
del Instituto Politécnico Nacional
Tresguerras 27, Centro Histórico, México, D.F.
mayo de 2015
Printing 500 / Edición 500 ejemplares